General

Los Alamos
NATIONAL LABORATORY

Earth and Environmental Sciences

EES-15, Environmental Science Group, MS J495
Los Alamos, New Mexico 87545
(505) 667-0811, FAX 665-3866
EES15-97-102

May 19, 1997

Mr. R. Stewart Dinwiddie
NMED/HRMB
2044 Galisteo Street
Santa Fe, NM 87505

Dear Stewart:

Enclosed are copies of the User's Manual and the documentation for the ecosystem model, BIOTRAN 2, developed by Tony Gallegos. Please note that the name of the model has been changed to BIOTRAN 2 from ECOTRAN because of a potential trademark infringement with ECOTRAN, Inc. of Ohio, which uses the name ECOTRAN in conjunction with schoolbus routing and scheduling software. For this reason, please destroy the draft copy of the BIOTRAN 2 user's manual and documentation that you previously received in which the name ECOTRAN was used.

Thank you very much.

Sincerely yours,

Roger

Roger W. Ferenbaugh, Team Leader
Ecological Risk Assessment Team

Enclosures: a/s

Cy: CIC-10, A150
EES-15 Files

General

LA-UR- 97-1468

Approved for public release; distribution is unlimited.

Title: BIOTRAN.2 USER'S MANUAL

LIBRARY COPY

Author(s): Anthony F. Gallegos, EES-15
Kathleen P. Starmer, EES-15

Submitted to: ER Project Report

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. The Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (10/96)

Los Alamos
NATIONAL LABORATORY

Earth and Environmental Sciences

EES-15, Environmental Science Group, MS J495
Los Alamos, New Mexico 87545
(505) 667-0811, FAX 665-3866

EES15-97-102

May 19, 1997

Mr. R. Stewart Dinwiddie
NMED/HRMB
2044 Galisteo Street
Santa Fe, NM 87505

Dear Stewart:

Enclosed are copies of the User's Manual and the documentation for the ecosystem model, BIOTRAN 2, developed by Tony Gallegos. Please note that the name of the model has been changed to BIOTRAN 2 from ECOTRAN because of a potential trademark infringement with ECOTRAN, Inc. of Ohio, which uses the name ECOTRAN in conjunction with schoolbus routing and scheduling software. For this reason, please destroy the draft copy of the BIOTRAN 2 user's manual and documentation that you previously received in which the name ECOTRAN was used.

Thank you very much.

Sincerely yours,

Roger

Roger W. Ferenbaugh, Team Leader
Ecological Risk Assessment Team

Enclosures: a/s

Cy: CIC-10, A150
EES-15 Files

16.0 APPENDIX G

BIOTRAN.2 USER'S MANUAL

by

Anthony F. Gallegos
Kathleen P. Starmer

ABSTRACT....vii
16.1. INTRODUCTION....727
16.1.1. Purpose and Description of User's Manual....727
16.1.2 Background on BIOTRAN.2 Development....727
16.1.3 General BIOTRAN.2 Structure....729
16.1.4 Example Description....729
16.1.5. References....733
16.2. General Procedures for Initiating Input to BIOTRAN.2
16.2.1. Getting Started....735
16.2.2. Loading the BIOTRAN.2 Executable File....735
16.2.3. Following Initial Input Screen Instructions....735
16.2.4. Selecting Input Options....736

16.3. Main Program....737

16.3.1. Description....737
16.3.2 Input....737
16.4. Climate Simulation....739
16.4.1. General Description....739
16.4.2. Solar Radiation....739
16.4.3. Temperature....739
16.4.4. Precipitation and Precipitation Pattern....741
16.4.5. Input/Output....741
16.4.6 References....747
16.5. BIOPLT Subroutine - Variable Selector for Graphical Displays....751
16.5.1. Description....751
16.5.2. Input/Output....751
16.6. TRUPLT Subroutine - Data Selection For Plotting....753
16.6.1. Description....753
16.6.2. Input/Output....753
16.7. TPLOTR Subroutine - Plotting Instructions....755
16.7.1. Description....755
16.7.2. Input/Output....755
16.8. AIRAC Subroutine- Acute and Chronic Releases of Contaminants....757
16.8.1. Description....757
16.8.2. Input/Output....757
16.9. WATFLX, RUNOFF, EROSON, AQUIFER, AND GEOFLX Subroutines - Soil Moisture Flow, Erosion, Watershed Simulations....773
16.9.1 Description....773
16.9.2. Input/Output....775
16.9.3. References....775
16.10. PLTGRO Subroutine And Plant Type Simulation....813
16.10.1 Description....813
16.10.2. Input/Output....813
16.10.3. References....838
16.11. UPTAKE Subroutine, Contaminant Uptake and Transport....839
16.11.1. Description....839
16.11.2. Input/Output....839
16.11.3. References....858
16.12. ANIMAL and INVERT - Mammal and Invertebrate Simulations....859
16.12.1. Description....859
16.12.2. Input/Output....859
16.12.3. References....892
16.13. HUMTRN Subroutine - Human Simulation and Dose....893

16.13.1. Description....893
16.13.2. Input/Output....893
16.13.3. References....913
16.14. ZOOGRO Subroutine - Grazing, Harvest, Feed, and Food Storage....914
16.14.1. Description....914
16.14.2. Input/Output....914
16.15. FORMAN Subroutine - Forest Growth Model....927
16.15.1. Description....927
16.15.2. Input / Output....927
16.15.3. References....928
16.16. FORCUT Subroutine - Forest Management and Fire Events....945
16.16.1. Description....945
16.16.2. Input / Output....945
16.16.3. References....945
16.17. AQUAT Subroutine - Limnetic Zone....952
16.17.1. Description....952
16.17.2. Input / Output....952
16.17.3. References....952
16.18. EFFECTS Model - Population Dynamics and Cancer Risks....960
16.18.2. Input / Output....960
16.18.3. References....960

17.0. LIST OF TABLES FOR THE USERS MANUAL(16.0.)

1 List of Subroutines for the BIOTRAN.2 Model....733
2 MAIN Subroutine Input....738
3 CLIMAT Subroutine Input....742
4 Example 1(Amarillo, Texas): Minimum Input Required to Execute BIOTRAN.2.743
5 Output From Example 1, Amarillo, Texas, for first 20 Days in the Third Year of the Simulation....748
6 BIOPLT Subroutine....752
7 TRUPLT Subroutine Input....754
8 TPLOTR subroutine Input....756
9 AIRAC Subroutine Input....759
10 Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario....762
11 Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario....766
12 Output From Both the Acute and Chronic Simulations of Example 1, Amarillo, Texas, for Day 325-day 345of the first simulation year....770
13 WATFLX (I and II) and GEOFLX Subroutine Input....777
14 Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario....785
15 Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario....790
16 Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% slope....795
17 Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff scenario at 20% slope....800
18 PLTGRO Subroutine Input....816
19 Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration In Various Types of Plants....819
20 Examples 3,and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants Undergoing Succession and Niche Competition....828
21 UPTAKE Subroutine Input....840

22 Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (Corn) Utilized To Generate Uptake Output.....841
23 Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants Utilized To Generate UPTAKE output.....846
24 ANIMAL Subroutine Input.....863
25 Amarillo, Texas Crops and Beef and Dairy Cattle.....870
26 Amarillo, Texas Grasshopper Biomass Density.....881
27 HUMTRN Subroutine Input.....895
28 Plants, Cattle, and Human Input.....901
29 Amarillo, Texas Crops ZOOGRO Input.....915
30 Specific Gravity And Regression Coefficients For Selected Trees Simulated in FORMAN.....930
31 FORMAN Subroutine Input.....931
32 LANL Pinyon Pine Input.....932
33 Succession Input for Grasses, Shrubs, and Trees (Pinyon and Ponderosa) in an Ecotone Setting For Los Alamos.....936
34 FORCUT Subroutine Input.....946
35 LANL Pinyon Pine Fire INPUT.....947
36 AQUAT Subroutine Input.....955
37 Shallow Lake Input for Amarillo, Texas.....956
38 EFFECTS Subroutine Input Requirements.....958
39 Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Human Male Exposure to Uranium.....960

18.0. LIST OF FIGURES FOR THE USERS MANUAL (16.0.)

1 Structure of BIOTRAN.2.....731

2 Major Processes Simulated in the CLIMAT Subroutine. Temperature and Precipitation are Stochastically Simulated and Drive the Daily Climate Simulation, Bounding the Biomass and Moisture Transport Subroutine Simulations.....740
3 Simulated Rain (P) and Snow (F7) for the Amarillo, Texas, Climate Scenario For Four years.....749
4 Simulated Mean Daily Temperature for the Amarillo, Texas, Climate Scenario For Four Years.....750
5 Acute and Chronic Air Contaminant Concentrations Are Simulated Above the Soil Surface in AIRAC.....758
6 Simulated Acute Contaminating Event for the Amarillo, Texas, Air Contamination Scenario for Four Years.....771
7 Simulated Chronic Contaminating Event for the Amarillo, Texas, Air Contamination scenario for four years.....772
8 Illustration of the 1 m^2 Soil Profile for Each Region Simulated in WATFLX. The User Specifies the Depth and Soil Characteristics For Each Soil Layer.....774
9 Cumulative Precipitation Input From CLIMAT, P1, and Soil Surface Evaporation, EVAP, as Simulated in WATFLX for Amarillo, Texas Pullman Soils on Bare Ground.....805
10 Soil Contamination in pCi/gram for Four Simulation Years. scenario.....806
11 Buildup of Soil Water Content in the Second and Third Soil Layers from Bare Ground Conditions. Loss Through the Bottom of Soil Profile and Evaporation are not Sufficient to Remove all of the Added Soil Moisture. Transpiration of Moisture by Plants is Clearly Absent.....807
12 Soil Surface Water Dynamics When Irrigated Corn is Grown on Soils Illustrated in Figures 16.9., 16.10., and 16.11.....808
13 Soil Contamination Increasing in the First Layer After The Acute AIRAC Event on Day 195 of the Third year. The Initial and Subsequent Measure of Soil Surface Contaminants are Affected by Washoff of Uranium from Plant Surfaces.....809

14 Soil Water Content is Significantly Different in This scenario Than That Shown in Figure 16.11. The Effect of Plant Transpiration Throughout the Growing Season is Apparent....810
15 Surface Runoff and Aquifer Volume for Irrigated Corn Scenario for a 4000 Acre Square Watershed With a 20% slope....811
16 Surface Runoff and Aquifer Volume for Uncropped Warm Season Perennial Grass On the Same Watershed to Compare With Figure 16.15. for Irrigated Corn Which Has Been Furrowed, and Does Not Allow Surface Runoff....812
17 Biomass Growth Processes Simulated In PLTGRO For 22 Different Plant Types With Three Soil Layers....815
18 Irrigated Corn Biomass For Various Plant Components for the Amarillo, Texas Scenario....832
19 Irrigated Corn Transpiration For Amarillo, Texas Scenario....833
20 Live Above Ground Biomass for Amarillo, Texas Scenario for Garden Vegetables, Alfalfa, Warm and Cool Season Grasses, and Winter Wheat. All But the Grasses Have Specific Irrigation and Cropping Schedules. Alfalfa and Winter Wheat Span More than One Crowing Season. The Grasses are Niche Competitors....834
21 Transpiration For Grasses and Crops For the Amarillo, Texas Scenario. Compare These Curves With Those in Figure 16.19. for Irrigated Corn....835
22 Secondary Plant Succession in Pinyon-Juniper Woodland. Note the Accelerated Growth After Complete Dominance of Trees....836
23 Independent Growth of Plants in Pinyon-Juniper woodland. Succession and Niche Competition Strategies Have Not Been Applied To Any Plant Type or Grouping....837
24 Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine....853
25 Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine....855
26 Daily Surface Particle Concentrations of Uranium On Irrigated Corn Leaves and seeds for Amarillo, Texas....856
27 Internal Leaf Concentrations for Uranium for Crops and Range Grasses At Amarillo,Texas....857
28 Illustration of ANIMAL/ZOOGRO Operations With the Human Model HUMTRN. The Invertebrate Model INVERT Also Interacts in A Similar Manner....861
29 Contaminant Metabolism in a Mammal Based on Physiologically Based Pharmacokinetics Modeling in BIOTRAN.2....862
30 Beef Cattle Air, Soil, Feed, and Total Uranium Intakes For Amarillo, Texas....887
31 Dairy Cattle Air, Feed, and Total Uranium Intake for Amarillo, Texas....888
32 Beef Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas....889
33 Dairy Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas....890
34 Male and Female Grasshopper Biomass Densities for Amarillo, Texas....891
35 HUMTRN Model Integrates the Volume, Mass, and Contaminant Intake Into Human Male and Female Population. Humans are Simulated Based On User Diet Selection and Population Dynamics....894
36 Uranium Uptake From Various Foods by 25 Year Old Males....911
37 Radiation Doses From U-238 to Various Organs and Tissues of 25 Year Old Males....912
38 ZOOGRO Subroutine Simulates Daily forage, feed, or food to ANIMAL, INVERT, and HUMTRN According to Diet Schedules Supplied By the User....924
39 Crop Harvests For Beef Cattle in the Amarillo, Texas Scenario....925
40 Crop Harvests for Dairy Cattle in the Amarillo, Texas Scenario....926

41 FORMAN Subroutine Manages the Growth and Contaminant Transport for Forest Stands Based on Selections Made in PLTGRO....929
42 Timber Volume by Diameter Classification for Pinyon Pine Stand at Los Alamos....942
43 Biomass Densities for Mixed Pine Stand for Los Alamos....943
44 Timber volume for Ponderosa Pine in a Mixed Pine Stand for Los Alamos....944
45 Simulated Wood Volumes By Diameter Size From a Pure Stand of Pinyon-Juniper Before and After A Forest Fire.... 951
46 AQUAT Subroutine Input....953
47 AQUAT Simulation of Heat Flux Between Water Layers, Substrate, and Air; and Its Interaction With Other Biotic and Abiotic Components of The Aquatic System....954
48 Shallow Lake Phytoplankton and Zooplankton Densities for Amarillo, Texas....959
49 Predicted Cancer Mortalities in Males From Uranium Exposure by the Relative Risk Methodology....969

ABSTRACT

BIOTRAN.2 is an ecological transport model that is used to predict the flow of organic and inorganic contaminants, including radionuclides, through simulated ecosystems--which can include plant, animal, and human populations--using biomass as a vector. The model is written in FORTRAN 77 and consists of 22 coupled subroutines that utilize daily and yearly loops driven by a Monte Carlo simulated climate. These subroutines include variables within the disciplines of meteorology, soil physics, hydrology, geology, plant and animal physiology, limnology, radiation biology, ecology, epidemiology, risk analysis, and health physics. The flexibility of the model allows the user to develop simple scenarios using only a few subroutines or complex scenarios using all of the subroutines; furthermore, the model can simulate an ecosystem for whatever time period in years the user desires. The purpose of this manual is to assist the user in developing site-specific and generic simulations using user-selected nuclides, heavy metals, organics, and several nutrients in air, soil, aquifers, water, vegetation, animals, and man. Three input examples are used throughout the manual: One is based on a site in Amarillo, Texas, and the other two are based on the Los Alamos National Laboratory site. These examples illustrate most of the model's capabilities and will assist in verifying BIOTRAN.2 operation on a subroutine-by-subroutine basis.

16.0. APPENDIX G

16.1. INTRODUCTION

16.1.1. Purpose and Description of User's Manual

The BIOTRAN.2 simulation model is a collection of 22 different models that function together. The models are structured as "subroutines" to facilitate the exchange of information within daily and yearly simulation intervals (or "loops"). BIOTRAN.2 was developed to assist the user, through the use of simulations, in analyzing scenarios which incorporate aspects from widely different sources in terrestrial and aquatic environments and allows the user to investigate scenarios in a systematic manner. The purpose of this manual is to assist the new user in developing BIOTRAN.2 simulations.

The new user can be overwhelmed by the magnitude of the BIOTRAN.2 code. Thus, this manual uses an inductive approach, beginning with simple scenarios and proceeding to more complex simulations in subsequent chapters. Each subroutine will be thoroughly described as it is encountered in the manual, and the code logic will be reviewed. With each new chapter, the user should construct an input file utilizing all the subroutines described up to the current chapter, execute the simulation, and plot the output graphically. This approach will not only increase the user's familiarity with the model, but will also verify that the code performs to his expectations for solving specific problems using BIOTRAN.2.

Three input examples are included in the manual. One is for Pantex Site near in Amarillo, Texas, and the other two sites are for the Los Alamos National Laboratory in Los Alamos, New Mexico. It is expected that after the user employs these inputs in the verification exercises, he will proceed with the verification of a simulation of his own creation.

Each subroutine section in the manual is composed of two to four sub-sections
- The subroutine description addresses the major operations performed by the subroutine
- **references** are included when applicable
- **input** required to execute the code is described
- **output** is shown graphically.

Because example inputs are additive, the user should read all chapters prior to a specific subroutine before attempting a simulation using the subroutine in question.

The current version of BIOTRAN.2 contains plotting instructions as part of the input; consequently, at the end of each simulation, a graph is displayed. However, output is also created which can be imported into spreadsheet programs and graphical software packages. To create the graphical output examples shown in this manual, formatted output was imported into Microsoft Excel Version 5.0a (Microsoft Corporation, 1993).

16.1.2 Background on BIOTRAN.2 Development

BIOTRAN.2 development began in 1974 with the goal of establishing a regional ecosystem model to predict the fate of plutonium and other transuranic elements buried in trenches in the Los Alamos area. Emphasis was placed on modeling transport of plutonium within soils, roots, and above-ground portions of agricultural crops and surrounding natural ecosystems. An initial study was performed by New Mexico State University on New Mexico crop irrigation schedules, crop identification, and livestock distributions primarily in the Rio Grande Valley north of Belen, New Mexico (Creel and Lansford, 1975). This study resulted in an agricultural database used to perform early verification and validation studies with the soil and plant growth strategies of BIOTRAN.2 (formerly known as BIOTRAN). Refinement of

the climatic strategies and subsequent development of ruminant and irrigation components were then included in the model and reported in the first documentation of BIOTRAN (Gallegos, *et al.*, 1980).

Predictive modeling requires a detailing of soil, water, and biotic transport mechanisms over time frames between 10's and 1000's of years. Thus, a contract with Battelle at Hanford, Washington, resulted in several studies where BIOTRAN was coupled to several other models to simulate the detailed mesa top and canyon bottom hydrology at Los Alamos over large time frames, providing a comprehensive assessment of transuranic waste transport (Onishi, *et al.*, 1982; Whelan, *et al.*, 1983). During the same period, BIOTRAN was used to assess several farming and grazing scenarios at Los Alamos in a study of solid waste management alternatives (Walker, *et al.*, 1981). The model was then used in a risk analysis study for shallow land burial of transuranic-contaminated solid wastes (Hansen and Rodgers, 1983). The following year, BIOTRAN was coupled with the surface runoff, evaporation, and transpiration components of the CREAMS model (Knisel, 1980), a USDA sedimentation, fertilizer, and pesticide transport model, to perform validation studies at Los Alamos and other sites.

Concurrent with the use of BIOTRAN in radioactive waste management risk analyses, the model was used to evaluate the long-term risk of uranium and plutonium accidental releases in the Pantex (Amarillo, Texas) Environmental Impact Statement (EIS) (Wenzel and Gallegos, 1982; Wenzel, *et al.*, 1982a; Wenzel, *et al.*, 1982b; Buhl, *et al.*, 1982). After completion of the Pantex EIS, a human physiology and kinetics model (HUMTRN) was coupled to BIOTRAN to simulate age- and sex-specific human metabolism (Gallegos and Wenzel, 1984). Shortly thereafter, a risk assessment code called EFFECTS was developed to estimate radiation-induced cancer in dynamic human and animal populations from simultaneous acute and/or chronic exposures (Wenzel and Gallegos, 1985).

Further modifications to the model also occurred at this time. The macronutrients nitrogen (nitrate and ammonium anions), phosphate (soluble and labile), and potassium were incorporated into the plant growth portion of the model. A study conducted by Wenzel, *et al.* (1987), led to the incorporation of additional code which enabled BIOTRAN to simulate the transport of more than one nuclide simultaneously. In this study, the fate of low-level radioactive wastes from an abandoned waste pit at Los Alamos was determined by examining concentrations of the nuclides U^{238}, Pu^{239}, and Cs^{137} in roots, stems, leaves, and tree bark. A simulation of the waste site using BIOTRAN gave good agreement with field analyses, and the model is now capable of simulating up to contaminants simultaneously.

A need to make BIOTRAN simulate more than one sub-watershed region at a time arose from watershed radionuclide transport studies at Los Alamos involving surface runoff and erosion, channel deposition and scouring, and the movement of water through shallow aquifers (Gallegos and Wenzel, 1990). Four new subroutines, including RUNOFF, GEOFLX, EROSON, and AQUIFER, were developed to accomplish this goal, giving BIOTRAN multiregional capability.

In 1993, the model underwent extensive modification to incorporate a pharmacokinetic approach which made it possible to simulate the transport of inorganic and organic contaminants in plants and animals. It was this modification which precipitated the change of the model's name from BIOTRAN to BIOTRAN.2.

BIOTRAN.2 is considered a research and risk analysis tool at Los Alamos National Laboratory and is currently used for two purposes: to enhance the design and analysis of experimental research and to predict the short- and long-term consequences of contaminant release into natural and agrarian environments. The model is constantly being revised and expanded to address new programmatic goals; consequently, the user's manual is periodically updated. Formerly, the model was written to be executed on VAX/VMS and CRAY/CTSS systems at Los Alamos, but the current version of BIOTRAN.2 has been scaled for application on personal computers.

16.1.3 General BIOTRAN.2 Structure

BIOTRAN.2 is written in Fortran 77, and its current structure is illustrated in Figure 16.1. The connecting lines indicate the potential interactions between the various subroutines in the model. The CLIMAT subroutine models precipitation, insolation, temperature, potential evapotranspiration, and other climatic variables for use in other subroutines and consequently is the major driver of each simulation. Biomass, water, and contaminant transport are simulated within the subroutines as functions of time (daily loop) that are cycled in yearly time increments (yearly loop). With a few exceptions, the 24-hr day is the smallest time interval used in the large number of difference equations residing in the model. A short description of each subroutine is presented in Table 16.1.

The CLIMAT subroutine uses a Monte Carlo simulated climate which allows BIOTRAN.2 to be stochastically driven through the use of different random number seeds which produce both even and normal distributions. The use of different random number seeds will vary precipitation, precipitation pattern, and temperature. These variations in precipitation and temperature subsequently influence other variables in the model, such as the temperature of each water layer in AQUAT, the soil moisture in WATFLX, the amount of plant growth in PLTGRO, and the transport rates of contaminants in plants in UPTAKE. The short- and long-term effects of the simulated climate determine the eventual concentration of contaminants in air, soil, vegetation, aquifers, animals, and humans.
Because of the importance of the CLIMAT subroutine, the user should be especially certain that the CLIMAT input is reliable. However, in order to ensure reliable simulations, it is important that all input parameters be of good quality. Because BIOTRAN.2 simulates the most likely (mean) response of a process, most input values should be selected from mean values where sufficient data are available. Occasionally, the user may wish to simulate a region where reliable experimental data are not available. The user can still perform a simulation without this data, but the reliability of the output should not be extrapolated beyond the limitations of the input.
Shortcomings inherent in the input can be quantitatively investigated using sensitivity analyses, whereby input parameters are systematically varied within their known range of values. The resultant output from these analyses can be utilized to determine those input areas which require further research for model application purposes. Probabilistic methods are also available to determine the reliability of the output (Leggett and Williams, 1981).
The flexibility of BIOTRAN.2 allows a wide range of scenarios to be created. Detailed analyses can be performed by creating extensive input files using site-specific data; alternatively, simple analyses can be performed by bypassing many types of input (*i.e.*, using defaults) and utilizing only small portions of the code. Furthermore, simulations can encompass time intervals from one year to many thousands of years. The new user should carefully construct a scenario and then decide on the amount of input required to develop the simulation.

16.1.4 Example Description

Three examples are used in this manual:

- **Example 1** was developed during validation studies with BIOTRAN.2 (Wenzel and Gallegos, 1982; Wenzel, *et al.*, 1982a; Buhl, *et al.*, 1982; Gallegos and Wenzel, 1984) using environmental surveillance data for uranium and accident analyses for plutonium from the Amarillo, Texas, DOE Pantex site. Input utilizes 15 subroutines to develop simulated uranium transport in air, water, and the human food chain for range grasses and crops on Pullman soils. Beef cattle are grazed on the simulated range grass, dairy cattle consume the simulated crops, and humans inhale air from over the simulated crops, eat the crops and beef, and drink the milk. This results in contaminant uptake, radiation doses, and cancer mortalities in the human population. The above scenario illustrates and verifies the subroutines MAIN, AIRAC, CLIMAT, WATFLX, RUNOFF, PLTGRO, UPTAKE, ZOOGRO, ANIMAL, HUMTRN, EFFECTS, BIOPLT, TRUPLT, and TPLOTR.

Table 16.1.

List of Subroutines for the BIOTRAN.2 Model.

Subroutine Name	Description of Major Processes Modeled
MAIN*	Duration, main components, and subroutine access
CLIMAT*	Daily temperature, solar radiation, precipitation and precipitation pattern, potential evapotranspiration, and other related climatic Monte Carlo-driven variables used as "drivers" for BIOTRAN.2.
AIRAC	User-specified acute or chronic average daily air nuclide concentrations from accidental releases, fallout, or ambient air data selected by day, number of days, year, and number of years for a given simulation.
WATFLX	Daily soil profile hydrology, potential evaporation, and transpiration for a particular soil and plant cover type.
RUNOFF	Surface runoff simulator for specific precipitation events and soil moisture infiltration from such an event.
AQUIFER	Aquifer characterizer, aquifer influx, and aquifer outflux processor on a daily basis.
GEOFLX	Processor for surface runoff, subsurface flow, channel flow, and sediment transport from sheet and channel erosion events.
EROSON	Processor for differential erosion from surface and channel erosion events as a function of soil or alluvium particle diameter.
PLTGRO*	Simulator for above- and below-ground live and dead biomass dynamics for up to 22 model-specified and user-defined plant types (grasses, forbs, shrubs, trees, and agricultural crops).
UPTAKE*	Simulator of contaminant transport by wind, precipitation, and root uptake to estimate internal and surficial contamination of plants. Nutrient transport processes are included.

Table 16.1.

List of Subroutines for the BIOTRAN.2 Model (continued).

Subroutine Name	Description of Major Processes Modeled
ANIMAL	Simulator of animal metabolism and contaminant intake, uptake, and internal translocation based on a daily diet regime of simulated forage and feed, intake of contaminated water and inhalation of airborne contaminants. Insect growth simulation (INVERT) is included in this subroutine. forage and feed, intake of contaminated water and inhalation of airborne contaminants. Insect growth simulation (INVERT) is included in this subroutine. forage and feed, intake of contaminated water and inhalation of airborne contaminants. Insect growth simulation (INVERT) is included in this subroutine.
ZOOGRO	Harvest and storage simulator for native and agricultural crops used for animal and human consumption.
HUMTRN	Simulates age- and sex-specific human physiology. Contaminant intake, uptake, dose (for radionuclides), and/or concentration in internal organs is based on a daily diet regime of simulated milk, meat, vegetables, and grains. Individual doses and/or concentrations are estimated on an age- and sex-specific basis.
FORCUT	Simulates selective cutting events and forest fires in concert with the FORMAN subroutine.
AQUAT	Freshwater lake simulator which includes daily limnetic water characteristics, phytoplankton, and zooplankton population dynamics. Some processes in this subroutine use hourly time increments.
EFFECTS	Simulates age- and sex-specific human and animal cancer risk and deaths from the radiation exposure and body burdens estimated by the subroutines HUMTRN and ANIMAL. This subroutine can be used for stable nuclides and other substances if cancer induction rates are known and toxicological endpoints are similar.
BIOPLT*	Selector for graphical or formatted output.
TRUPLT*	Allows the user to select data from the subroutine BIOPLT for plotting.
TPLOTR*	Variable plotter subroutine.

*These subroutines are required in all simulations involving BIOTRAN.2. All other subroutines are user options and can be accessed by opening their respective gates.

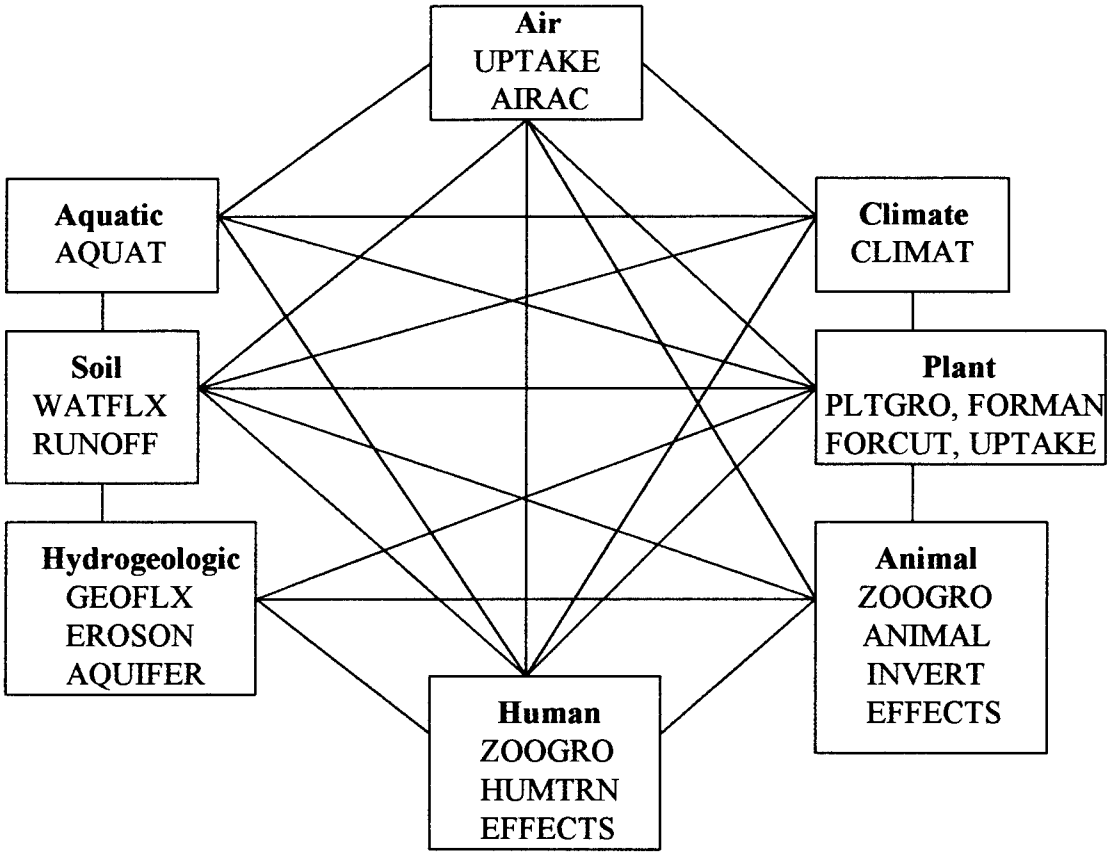


Figure 16.1.

Structure of BIOTRAN.2

note: Biomass, water, and contaminants are transported in daily and yearly intervals as driven by the simulated climate.

- **Example 2,** validated by Wenzel, *et al.* (1987), simulates a pinyon-juniper (*Pinus edulis, Juniperus monosperma*) forest at Los Alamos National Laboratory growing for 100 years over a low-level transuranic waste repository. This example uses ten subroutines in total, but only the subroutines FORMAN and FORCUT are described in detail.

- **Example 3** is also taken from the Los Alamos area. In this scenario, erosion, runoff, and contaminant transport processes are simulated in Mortandad Canyon watersheds and aquifers (Gallegos, *et al.*, 1990). This simulation illustrates the subroutines AQUIFER, GEOFLX, EROSON, and AQUAT.

16.1.5. References

Buhl, T., Dewart, J., Gunderson T., Talley, D., Wenzel, J., Romero, R., Salazar, J., and D. Van Etten. 1982. Supplementary documentation for an environmental impact statement regarding the Pantex Plant: radiation monitoring and radiological assessment of routine releases. Los Alamos National Laboratory report LA-9445-PNTX-C.

Creel, B.J. and R.R. Lansford. 1975. Agricultural resource base data for environmental evaluation of toxic waste burial sites and their ultimate disposition. Department of Agricultural Economics and Agricultural Business, Agricultural Experiment Station, New Mexico State University.

Gallegos, A.F., Garcia, B.J., and C.M. Sutton. 1980. Documentation of TRU biological transport model (BIOTRAN). Los Alamos National Laboratory report LA-8213-MS.

Gallegos, A.F. and W.J. Wenzel. 1984. ICRP-based age- and sex-specific human simulation model for radionuclide dose assessment. Los Alamos National Laboratory report LA-9994-MS.

Gallegos, A.F. and W.J. Wenzel. 1990. Runoff and sediment yield model for predicting nuclide transport in watersheds using BIOTRAN. Los Alamos National Laboratory report LA-11851-MS.

Hansen, W.R. and J.C. Rodgers. 1983. Risk analysis for shallow land burial and greater confinement of alpha-contaminated wastes. Nuclear and Chemical Waste Management 4: 81-94.

Knisel, W.G. (ed.) 1980. CREAMS: A field-scale model for chemicals, runoff, and erosion from agricultural management systems. USDA Science and Education Administration Conservation Report No. 26, USDA, Washington, D.C.

Leggett, R.W. and L.R. Williams. 1981. A reliability index for models. Ecological Modeling 13: 303-312.

Microsoft Corporation. 1993. User's Guide 1. Microsoft Corp., Redmond, Washington. 641 p.

Onishi, Y., G. Whelan, and R.G. Skaggs. 1982. Development of multimedia radionuclide exposure assessment methodology for low-level waste management. Pacific Northwest Laboratory report PNL-3370.

Walker, L.W., Hansen,W.R., Wilson, D.C., Maestas, G., Wenzel, W.J., and F.A. Guevarra. 1981. Alternative transuranic waste management strategies at Los Alamos National Laboratory. Los Alamos National Laboratory report LA-8982-MS.

Wenzel, W.J. and A.F. Gallegos. 1982. Supplementary documentation for an environmental impact statement regarding the Panex Plant: long-term radiological risk assessment for postulated accidents. Los Alamos National Laboratory report LA-9445-PNTX-O.

Wenzel, W.J., Wallwork-Barber, K.M., Horton, J.M., Rea, K.H., Hollis, L.C., Gladney, E.S., Mayfield, D.L., Gallegos, A.F., Rodgers, J.C., Thomas, R.G., and G. Trujillo. 1982a. Supplementary documentation for an environmental impact statement regarding the Pantex Plant: agricultural food chain radiological assessment, Los Alamos Nation Laboratory report LA-9445-PNTX-M.

Wenzel, W.J., Wallwork-Barber, K.M., Rodgers, J.C., and A.F. Gallegos. 1982b. The Texas panhandle soil-crop-beef food chain for uranium: a dynamic model validated by experimental data. Proceedings of the fourth DOE Environmental Protection Information Meeting, CONF-821215, report 6D, pp. 311-321. Denver, Colorado.

Wenzel, W.J. and A.F. Gallegos. 1985. EFFECTS: Documentation and verification for a BEIR III cancer risk model based on age, sex, and population dynamics for BIOTRAN. Los Alamos National Laboratory report LA-10371-MS.

Wenzel, W.J., Foxx, T.S., Gallegos, A.F.,Tierney, G., and J.C. Rodgers. 1987. Cs-137, Pu-239/240, total uranium, and scandium in trees and shrubs growing in transuranic waste at Area B. Los Alamos National Laboratory report LA-11126-MS.

Whelan, G., Thompson, F.L., and S.B. Yabushi. 1983. Multimedia contaminant environmental exposure assessment methodology as applied to Los Alamos, New Mexico. Pacific Northwest Laboratory report PNL-4546.

16.2. General Procedures for Initiating Input to BIOTRAN.2

16.2.1. Getting Started

Before beginning any BIOTRAN.2 simulation, the model requires that the files **input.dat**, **enter.dat**, **output.dat**, and **plot.dat** be created as dummy files. All dummy files created by the user are overwritten by the model when input is entered. The enter.dat file is what the code creates as a duplicate of input.dat, the output.dat file is the selected output from a simulation, and the plot.dat file contains only the output specified in the plotting subroutines.

There are two ways to create input for a simulation. One entails creating the input.dat file as a dummy file (mentioned in the preceding paragraph), then using the pop-up query windows in the model to overwrite the dummy file with actual input by entering answers to each question. The other entails entering input in a DOS editor and saving the file as input.dat, which is then run through the model. To this end, each subroutine chapter of the user's manual has a table which defines every line of input for the subroutine in question. Both methods accomplish the ultimate goal of creating a simulation, but using the model to create an input.dat file is recommended initially. Created input.dat files may be edited in DOS and run through the model without having to reenter input data using the query windows.

16.2.2. Loading the BIOTRAN.2 Executable File

Every BIOTRAN.2 simulation begins in the same manner, regardless of the input process. Because BIOTRAN.2 is written in FORTRAN 77, DBOS/486 Ver. 2.75 (Salford Software Ltd., 1990, 1991) must be loaded in order to run the model. Once DBOS is loaded, the command "run77 biotran" is typed at the prompt. The model's title screen will appear, followed by a license screen, a description of biotran.2, and a list of the model's subroutines.

16.2.3. Following Initial Input Screen Instructions

The **fifth** screen states that the model is now in the MAIN subroutine, but five more query screens follow before the input described in Chapter 3 (*i.e.*, the input entered into a DOS editor-created input.dat file) begins. This fifth screen makes the important point of notifying the user that <u>alphanumeric input characters must be entered in single quotes</u> (*e.g.*, 'b,' 'snow'). **Note that single quotes are <u>not</u> required when entering input in a DOS editor.**

The **sixth** screen serves two purposes. This is the screen that allows the user to select the random number seed which determines the simulated climate (discussed in detail in Chapter 4). If 'true' is entered, then the random number defaults to 1000, and if 'false' is entered, another query screen appears which asks the user to select the random seed of his choosing (*e.g.*, 21, 856, 9901). Alternatively, this screen can be used to exit a simulation by typing 'end.'

If a random number was selected in the previous step, either by the user or by default, the following screen gives the user the option of using an input.dat file of previous creation by typing 1.0 or of creating an input.dat file (and, simultaneously, the enter.dat file) with the model by typing 0.0. It should be noted that if the user has an incomplete input.dat file with complete <u>individual</u> subroutines, he may still choose option 1.0; when the model comes to a subroutine not yet in the input.dat file, he may switch to keyboard input and use the query screens to create the input for the subroutine in question.

The next screen asks the user if he would like an echo of his input variables. This is described in the following screen, which states that whole numbers can be entered as integers, which are then converted to real numbers as shown in the echo. This screen also gives the following information:

- Separate more than one input entry by commas or blanks (*e.g.*, 1.0,2.0e5,4.5,2). Repetitive numbers are permitted (*e.g.*, 1.0,4*5.0,2*0.0 2.0e4). If the next enter prompt does not appear, then not enough data has been entered.

- Continue to add data until the set is complete. If more data are required than can fit on one line, press return and continue as needed."

16.2.4. Selecting Input Options

The last screen gives the user the option of either entering input through the keyboard or of using data from his input.dat file for the first subroutine. At this point, the model begins asking for the information described in the following section. A final point to remember pertains to input errors. The user should check the model created **enter.dat** file for input errors after a simulation is completed before saving and archiving any files used in the simulation.

16.3. Main Program

16.3.1. Description

The MAIN program controls the input options, sequence, and execution strategy for each simulation as specified by the user. Variables are not processed in the MAIN program. At the beginning of a simulation, all the required subroutines are accessed for input. When all the required input has been entered, the simulation proceeds, beginning with the daily iteration and continuing with the yearly iteration. The subroutines FORMAN and FORCUT can be executed on *either* a daily or yearly basis, but in most applications the yearly iteration is selected because the daily iteration would be excessive. The user should have a detailed idea of the scenario he wishes to simulate before deciding which subroutines to execute. The plotting subroutines BIOPLT, TRUPLT, and TPLOTR are called directly by other subroutines to produce the desired output for a given run.

16.3.2 Input

The **ten** input lines required by MAIN are presented in Table 16.2. (***NOTE:*** Only the information asked for is entered when creating an input file; it is not necessary to enter the variable name). Variables ZOOX, ANIX, HUMX, BPX, AIRX, WATX, FRMNX, and AQUAX are gates to subroutines ZOOGRO, ANIMAL, HUMTRN, BIOPLT, AIRAC, WATFLX, FORMAN, and AQUAT, respectively. ANIX and HUMX are actually called from ZOOGRO; thus, to simulate animals and/or humans, the ZOOGRO gate must be opened in the MAIN subroutine. A value of 1.0 accesses a specified subroutine, whereas a value of 0.0 bypasses it. For the variables AIRX, WATX, FRMNX, and AQUAX, gate access must be specified for each region simulated. Since CLIMAT, PLTGRO, and UPTAKE are always called in a simulation, there are not any calling variables specified in MAIN for these subroutines as there are for the other subroutines requiring an access gate.
Three of the subroutines--WATFLX, FORMAN, and ANIMAL--have internal access to additional subroutines which may require input. Subroutine WATFLX will access subroutine GEOFLX, depending upon whether the input variable "CHNNU" (number of channels in a watershed) in WATFLX is greater than 0.0. Further input from GEOFLX may be required if the input variable "AQUGAT" (gate for aquifer model) in WATFLX is also greater than 0.0. Subroutine FORMAN will access subroutine FORCUT if the input array "TBRM" (number of cuttings or fire events) in FORMAN is greater than 0.0 for any region simulated. Similarly, subroutine ANIMAL will access INVERT for input if the input lifespans "ALSP" is set to 0.0. The input for each of these subroutines is described in more detail in their respective sections.

When entering values for variables DMD and YMD (Input Line 10, Table 16.2.), it must be remembered that there cannot be more than 8000 points per plot. These points include not only the number of days per year, but also the number of variables to be plotted. For example, if the user wishes to have output every day of each year for a four-year period, this results in a total of 1460 days being plotted. Now, if the user wishes to plot four variables on the same graph, 1460 x 4 results in 5840 points. Since this value is less than 8000, the proposed simulation can be successfully performed. Note that regardless of the number of points, a maximum of eight variables may be plotted per graph.

Finally, the sections to follow will describe in detail the specific inputs for subroutines mentioned above beginning with simple input instructions and proceeding with more complex simulation input requirements.

Table 16.2. MAIN Subroutine Input (REQUIRED).

Input Line	Variable Name	Definition
1	Y2	Number of years you wish to simulate
2	ANZ	Number of contaminants in your simulation
3	APX	Number of regions to be simulated
4	ZOOX	Open (1.0) or close (0.0) gate to ZOOGRO subroutine (no input for this subroutine)
	ANIX	Open (1.0) or close (0.0) gate to ANIMAL subroutine
	HUMX	Open (1.0) or close (0.0) gate to HUMTRN subroutine
	BPX	Open (1.0) or close (0.0) gate to BIOPLT subroutine
5	APL	Number of plant types simulated in each region
6	AIRX	Open (1.0) or close (0.0) gate to AIRAC subroutine for each region simulated
7	WATX	Open (1.0) or close (0.0) gate to WATFLX subroutine for each region simulated
8	FRMNX	Open (1.0) or close (0.0) gate to FORMAN subroutine for each region simulated
9	AQUAX	Open (1.0) or close (0.0) gate to AQUAT subroutine for each region simulated
10	DMD	Day modulus for output selection
	YMD	Year modulus for output selection

16.4. Climate Simulation

16.4.1. General Description

Major processes simulated in CLIMAT are presented in Figure 16.2., along with major transport routes, drivers, and important input parameters. Normal and evenly distributed random numbers (Monte Carlo drivers), derived from an internal random number generator, are used to stochastically generate temperature and precipitation patterns on a daily basis, described below. Specifically, the random numbers are used to determine the days during which precipitation events will occur, determine the amount of precipitation during each event, adjust the amount of precipitation using a coefficient of variation, and vary the temperature from the mean. The mean daily temperature and solar radiation for a site are generated using sine wave functions, also described below. If the estimated daily average temperature is greater than 2 °C, then precipitation is simulated as rain. If snow is present on a site when the temperature exceeds 2 °C, snowmelt is simulated. If a precipitation event occurs on a day when the temperature is less than or equal to 2 °C, it is simulated as snowfall. The initial evapotranspiration equation is very similar to the one used in the CREAMS model (Knisel, 1980) and will be discussed in a later section in conjunction with subroutine WATFLX.

Meteorological data, preferably from databases collected over 40 or more years (Baldwin, 1973; NOAA, 1973; NOAA, 1980), are required to develop statistics for realistic simulation of climate in a particular region. CLIMAT can also accept climatic data from reference sites in proximity to the site of interest if meteorological data for the area of interest is not available. This is made possible by certain CLIMAT input parameters (*e.g.*, latitude, adiabatic-lapse rate, elevation of the reference site) which allow the user to create a site-specific climate simulation by adjusting the input data from the reference site. Care should be taken to ensure the applicability of the reference data, however, if a realistic simulation is to be achieved.

16.4.2. Solar Radiation

Solar radiation (SOLRAD) is recorded in Langelys/day and is simulated by a sine function algorithm:

$$SOLRAD = ASOL*[\sin(2*\pi*DSOL) + 1] + T1$$
$$ASOL = (T2 - T1)/2$$
$$DSOL = (K3 - L1)/365,$$

where T1 (user input) is the minimum monthly solar radiation in Ly/d and T2 (user input) is the maximum monthly solar radiation. K3 is the Julian day number of the simulation, and L1 is the time lag in days to maximum solar radiation, estimated at 90 days from the vernal equinox in the northern hemisphere.

16.4.3. Temperature

Ambient daily average temperature (T), recorded in °F, is also simulated by a sine function algorithm; it is then converted to °C:

$$T = BTEMP*[\sin(2.0*\pi*E) + 1] + T3$$
$$BTEMP = (T4 - T3)/2$$
$$E = (K3 - L2)/365,$$

where T3 (user input) is the minimum average daily temperature (calculated on a per month basis; it is typically the January daily average) in °F and T4 (user input) is the maximum (usually the July daily average). L2 is the lag period in days to maximum temperature, estimated at 120 days after the vernal equinox in the northern hemisphere.

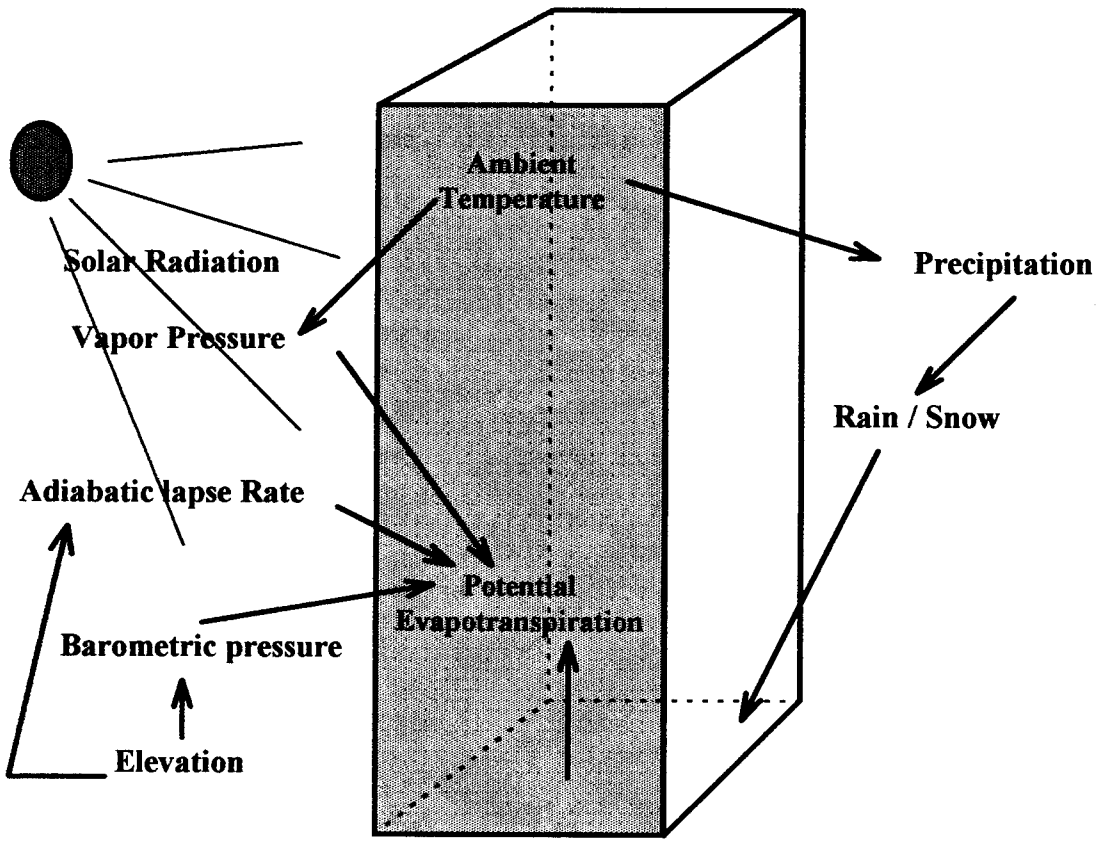


Figure 16.2.

Major Processes Simulated in the CLIMAT Subroutine. Temperature and Precipitation Are Stochastically Simulated and Drive the Daily Climate Simulation, Bounding the Biomass and Moisture Transport Subroutine Simulations.

Temperature (T) is stochastically incremented or decremented using the normal distribution of random numbers discussed at the beginning of this chapter:

$$T = T + G4*TMXV$$
$$T = (T - 32)*5/9,$$

where G4 is a randomly selected value from a normal distribution with a mean of 0.0 and a standard deviation (σ) of ± 1.0. TMXV (input value) is the mean daily temperature (calculated on a per month basis) standard deviation in °F, which is converted to °C.

16.4.4. Precipitation and Precipitation Pattern

Daily precipitation events (P) are recorded in mm and are simulated separately for each month of the year in a stochastic manner using both normal and even distributions:

$$P = (Q(nnx)*DRMM)/Q8$$
$$DRMM = X(mon)*R*25.4,$$

where the ratio [Q(nnx)/Q8] is the fraction of the total monthly rain falling on the nxxth day of the month. The fraction of yearly precipitation that occurs during a given month is an input parameter. The total mean monthly precipitation, DRMM, is estimated as the product of the fraction of total yearly precipitation per month [X(mon)] times total yearly precipitation (R). Both X(mon) and R are input parameters, with R entered in inches.

Each precipitation event (nxx) for a given month is incremented or decremented:

$$P = P + P*G5*PRCV,$$

where PRCV is the coefficient of variation for an individual rain event and G5 is a randomly selected value from a normal distribution with a mean of 0.0 and a standard deviation of ± 1.0.

16.4.5. Input/Output

The input required to execute the CLIMAT subroutine is described in Table 16.3. A portion of Example 1, containing data from Amarillo, Texas (Wenzel and Gallegos, 1982), is used to verify this subroutine (Table 16.4.). This example also contains the input for the subroutines MAIN, PLTGRO, UPTAKE, BIOPLT, TRUPLT, and TPLOTR, all of which are necessary to execute a simulation; thus, this example represents the smallest type of input file that is capable of generating output from BIOTRAN.2. It should be noted that duplicative statements in CLIMAT and PLTGRO would be required if more than one region was being simulated in this example, as watersheds usually contain several subwatershed regions with different microclimates and vegetative types. The inputs for the subroutines used in addition to CLIMAT are described in their respective chapters.

A portion of the output (specifically, a portion of the third simulation year) generated by Example 1 is shown in Table 16.5. BIOTRAN.2 is equipped with hundreds of output variables [see LANL publication "Documentation and Utilization of the Ecological Transport Model BIOTRAN.2" (in preparation) for a listing of these variables], and the user selects which variables he would like to examine as output in the plotting subroutines (described in detail in Chapter 5). In Example 1, the following variables are used: **P** (daily precipitation event, mm); **F7** (daily snow accumulation, mm); **SNM** (total snow accumulation, mm); **T** (mean daily temperature, °C); **E2** (potential evapotranspiration, mm/m^2/day); **T** (same as above); **TMN** (minimum daily temperature of site for a given day, °C); **TMX** (maximum daily temperature of site for a given day, °C); **SOLRAD** (solar radiation on ground surface, Ly/day); and **RADANG** (daily zenith angle for site, degrees). Of these output variables, precipitation (P, F7), and temperature (T) are shown graphically over a four-year period in Figures 4.3, 4.4, respectively.

Table 16.3.

CLIMAT Subroutine Input (REQUIRED).

Input Line	Variable Name	Definition
1		*Enter 555555 to access this subroutine.*
FOR EACH REGION SIMULATED:		
2	T1	Minimum monthly solar radiation, Ly/d
	T2	Maximum monthly solar radiation, Ly/d
3	L1	Lag period (days) to maximum insolation
4	T3	Minimum monthly average daily temperature (°F)
	T4	Maximum monthly average daily temperature (°F)
5	L2	Lag period (days) to maximum temperature
6	TMXV	Mean monthly temperature standard deviation (°F)
7	PRCV	Coefficient of variation for each precipitation event
8	RELV	Site elevation (ft) of reference site
9	ALPSR	Regional temperature/altitude lapse rate (°F/ft)
10	TRE	Number of rain events (>0.1 in) for each month (12 values)
11	X	Precipitation fraction for each month (12 values; must sum to 1.0)
12	R	Mean regional annual precipitation (in)
13	G1	Study site elevation (ft)
14	L3	Temperature correction (°F) for latitude difference between reference site and study site
15	FFP	Average number of frost-free days/year in the region
16	ALATD	Regional latitude (degrees)

Table 16.4.

Example 1 (Amarillo, Texas): Minimum Input Required to Execute BIOTRAN.2

MATN Subroutine

```
 5.00000
 1.00000
 1.00000
0.000000E+00   0.000000E+00   0.000000E+00   1.00000
 1.00000
0.000000E+00
0.000000E+00
0.000000E+00
0.000000E+00
1.00000        1.00000
```

CLIMAT Subroutine

```
 555555.
 277.000        625.000
 90.0000
 36.2000        78.0000
 120.000
 4.36000
0.200000
 3500.00
4.000000E-03
 4.00000        4.00000        4.00000        4.00000        4.00000
 4.00000        4.00000
 4.00000        4.00000        4.00000        4.00000        4.00000
2.300000E-02   2.600000E-02   3.700000E-02   6.200000E-02   0.149000
0.161000       0.143000
0.150000       9.300000E-02   8.900000E-02   3.900000E-02   2.900000E-02
 22.0000
 3538.00
0.000000E+00
 202.000
 36.0000
```

PLTGRO Subroutine

```
 121212.
0.000000E+00
0.000000E+00
0.000000E+00
0.000000E+00
```

UPTAKE Subroutine

```
 131313.
    0
```

BIOPLT Subroutine

```
 444444.
 1.00000
0.000000E+00   1.00000        0.000000E+00
 1.00000        1.00000
```

Table 16.4. (continued)

```
P
2.  16.
i       n
0.000000E+00
F7
1.   6.
n
0.000000E+00
SNM
1.  19.
n
0.000000E+00
T
1.  20.
n
0.000000E+00
E2
1.   5.
n
0.000000E+00
T
1.  20.
n
0.000000E+00
TMN
1.  20.
n
0.000000E+00
TMX
1.  20.
n
0.000000E+00
SOLRAD
0.  19.
RADANG
0.  18.
    1.00000
    3.00000
TRUPLT Subroutine
171717
2.00000
1 1
13.0000
1.00000       2.00000
   0.000000E+00   0.000000E+00
   0.000000E+00   0.000000E+00
   0.000000E+00   0.000000E+00
```

Table 16.4. (continued)

```
TPLOTR Subroutine
191919
4   2   20
        1        1
y
y
n
n
days
mm
PANTEX Precipitation
rain
snow
   -111.000      -111.000
   -111.000      -111.000
TRUPLT Subroutine
171717
1.00000
1
13.0000
4.00000
0.000000E+00
   0.000000E+00
   0.000000E+00
TPLOTR Subroutine
191919
4   7   18
        1
y
y
n
n
days
Celsius
PANTEX Temperature
Daily temp
   -111.000      -111.000
   -111.000      -111.000
TRUPLT Subroutine
171717
1.00000
1
13.0000
9.00000
   0.000000E+00
   0.000000E+00
   0.000000E+00
```

Table 16.4. (continued)

TPLOTR Subroutine

```
191919
4   4   17
        1
y
y
n
n
days
Ly/d
PANTEX Solar Rad.
Solar rad.
   -111.000      -111.000
   -111.000      -111.000
```

Note: The extra spacing between subroutines and the script notation of the individual subroutines is used for clarification purposes only: it is not part of the input

16.4.6 References

Baldwin, J.L. 1973. Climates of the United States. USDOC, NOAA, Environmental Data Service, Washington, D.C.

Knisel, W.G., Jr. 1980. CREAMS: a field scale model for chemicals, runoff, and erosion from agricultural management systems. USDA Conservation Research Report No. 26, USDA, Washington, D.C.

NOAA. 1973. New Mexico weather summary. USDOC, NOAA, National Climatic Center, Ashville, NC.

NOAA. 1980. Local climatological data for Amarillo, Texas. USDOC, NOAA, National Climatic Center, Ashville, NC.

Wenzel, W.J. and A.F. Gallegos. 1982. Supplementary documentation for an environmental impact statement regarding the Pantex Plant: long-term radiological risk assessment for postulated accidents. Los Alamos National Laboratory report LA-9445-PNTX-O.

Table 16.5.

Output From Example 1, Amarillo, Texas, for first 20 Days in the Third Year of the Simulation.

P	F7	SNM	T	E2	T	TMN	TMX
0.00E+01	0.00E+01	0.00E+01	3.68E+01	2.36E+01	3.68E+01	1.57E+01	6.15E+01
0.00E+01	0.00E+01	0.00E+01	3.59E+01	2.36E+01	3.59E+01	1.52E+01	6.01E+01
0.00E+01	0.00E+01	0.00E+01	3.50E+01	2.35E+01	3.50E+01	1.47E+01	5.88E+01
4.06E+01	0.00E+01	0.00E+01	3.41E+01	2.35E+01	3.41E+01	1.43E+01	5.75E+01
0.00E+01	0.00E+01	0.00E+01	3.33E+01	2.35E+01	3.33E+01	1.38E+01	5.63E+01
0.00E+01	0.00E+01	0.00E+01	3.25E+01	2.34E+01	3.25E+01	1.34E+01	5.51E+01
0.00E+01	0.00E+01	0.00E+01	3.18E+01	2.34E+01	3.18E+01	1.30E+01	5.40E+01
3.30E-01	0.00E+01	0.00E+01	3.11E+01	2.34E+01	3.11E+01	1.26E+01	5.29E+01
0.00E+01	0.00E+01	0.00E+01	3.04E+01	2.34E+01	3.04E+01	1.22E+01	5.18E+01
0.00E+01	0.00E+01	0.00E+01	2.98E+01	2.34E+01	2.98E+01	1.19E+01	5.08E+01
0.00E+01	0.00E+01	0.00E+01	2.91E+01	2.34E+01	2.91E+01	1.15E+01	4.99E+01
6.18E+01	0.00E+01	0.00E+01	2.86E+01	2.34E+01	2.86E+01	1.12E+01	4.90E+01
0.00E+01	0.00E+01	0.00E+01	2.80E+01	2.34E+01	2.80E+01	1.09E+01	4.81E+01
0.00E+01	0.00E+01	0.00E+01	2.75E+01	2.34E+01	2.75E+01	1.06E+01	4.73E+01
0.00E+01	0.00E+01	0.00E+01	2.70E+01	2.34E+01	2.70E+01	1.03E+01	4.66E+01
3.90E+01	0.00E+01	0.00E+01	2.65E+01	2.35E+01	2.65E+01	1.01E+01	4.59E+01
0.00E+01	0.00E+01	0.00E+01	2.61E+01	2.35E+01	2.61E+01	9.87E-00	4.53E+01
0.00E+01	0.00E+01	0.00E+01	2.57E+01	2.36E+01	2.57E+01	9.65E-00	4.47E+01
0.00E+01	0.00E+01	0.00E+01	2.54E+01	2.36E+01	2.54E+01	9.46E-00	4.41E+01
0.00E+01	0.00E+01	0.00E+01	2.51E+01	2.37E+01	2.51E+01	9.29E-00	4.37E+01

SOLRAD	RADANG
2.77E+03	3.05E+02
2.77E+03	3.05E+02
2.78E+03	3.06E+02
2.78E+03	3.06E+02
2.78E+03	3.06E+02
2.78E+03	3.07E+02
2.79E+03	3.07E+02
2.79E+03	3.08E+02
2.80E+03	3.09E+02
2.80E+03	3.09E+02
2.81E+03	3.10E+02
2.82E+03	3.11E+02
2.82E+03	3.12E+02
2.83E+03	3.13E+02
2.84E+03	3.14E+02
2.85E+03	3.15E+02
2.86E+03	3.17E+02
2.87E+03	3.18E+02
2.88E+03	3.19E+02
2.89E+03	3.21E+02

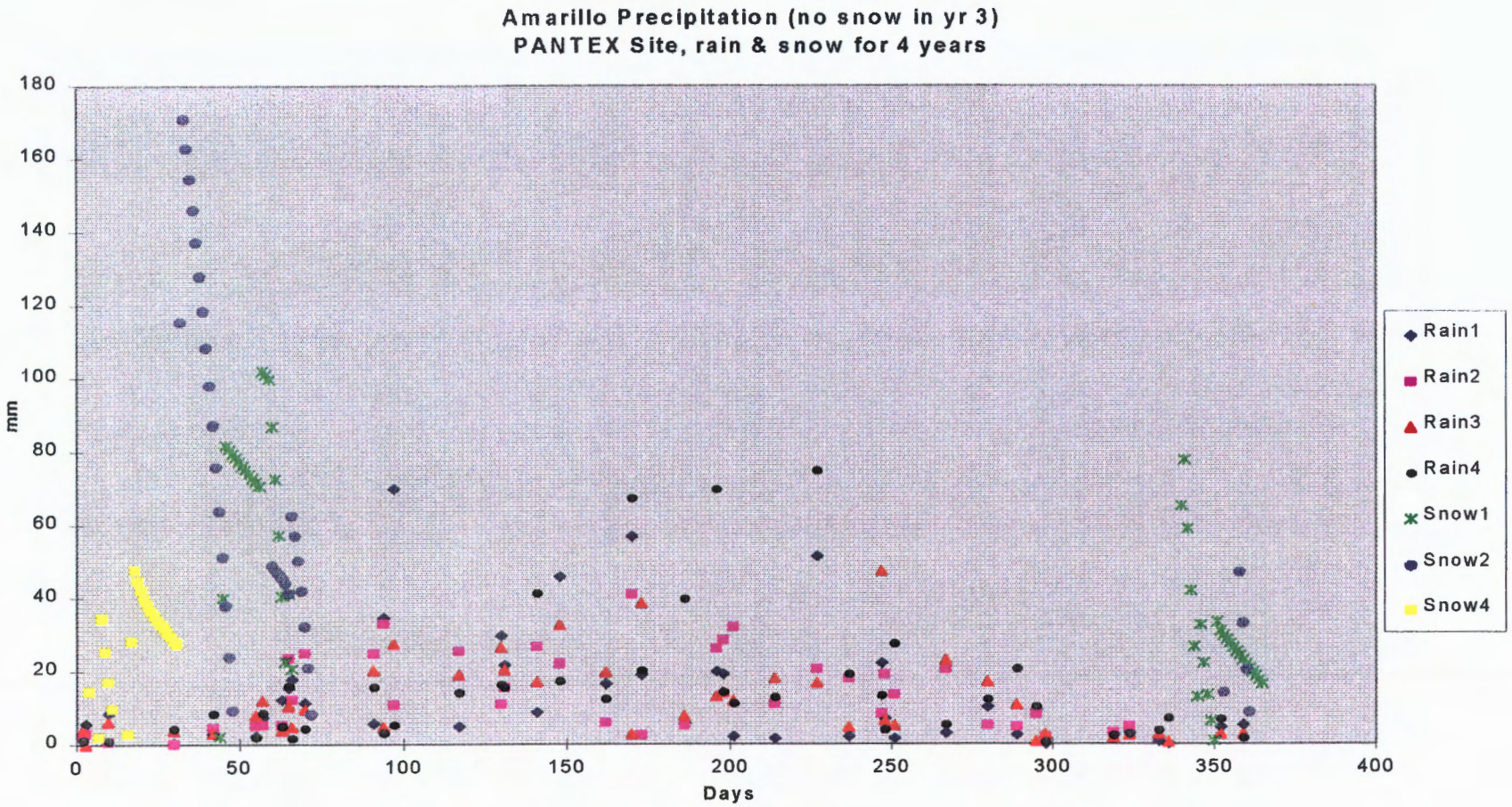


Figure 16.3.

Simulated Rain (P) and Snow (F7) for the Amarillo, Texas, Climate Scenario for Four Years.

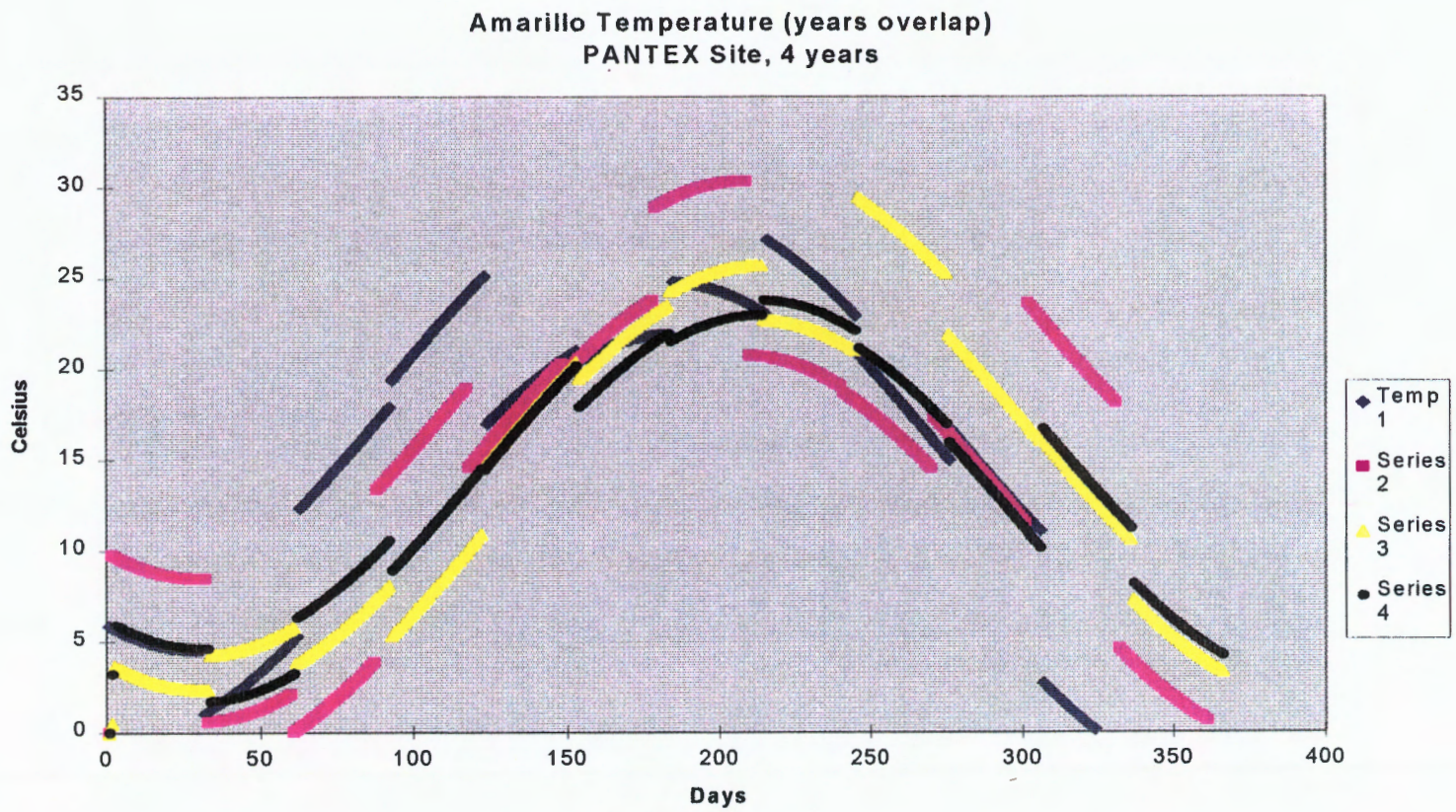


Figure 16.4.

Simulated Mean Daily Temperature for the Amarillo, Texas, Climate Scenario for Four Years

16.5. BIOPLT Subroutine - Variable Selector For Graphical Displays

16.5.1. Description

BIOPLT creates a database from which to choose when plotting output graphs. Arrayed variables may be printed in their entirety or partially, depending on BIOPLT input instructions. The unformatted output from this subroutine is used by the other plotting subroutines, TRUPLT and TPLOTR (see sections 16.6. and 16.7.).

Creating a graphical display of output greatly speeds the verification of the simulated scenario. By placing "spikes" or "tracers" in the simulation, the user can carefully and graphically follow the transport of contaminants throughout the simulation interval.

16.5.2. Input/Output

When the variable BPX is set to 1.0 in line 1 of the MAIN input, the user specifies that he will use BIOPLT for output. Alternatively, if BPX is set to 0.0, no access to BIOPLT is specified and no output file is generated.

BIOPLT always follows the end of the BIOTRAN.2 simulation set-up, after input is completed for all other accessed subroutines. It, in turn, is followed by the two other plotting subroutines discussed in sections 16.6. and 16.7.. The input statements required to execute BIOPLT are described in Table 16.6.

A **glossary** of variables which can be selected for output, as well as their attributes, are presented in Appendix D. These variables are available in their entirety only when all possible subroutines are accessed in the model. For example, variables pertaining to tree stands are available only when the FORCUT and/or FORMAN subroutines are used in a simulation.

Contained within each variable are "array elements," which can vary in number from zero (in which case no input is required for lines 7 or 8) to four per variable. The elements specific to each variable are listed in the glossary. Each array element represents a specific parameter; for example, the element "i" represents plant number, and the element "m" represents contaminant number.

When entering these elements into BIOPLT (input line 7, Table 16.6.) using a DOS editor, the user must be sure to place them in their appropriate fields. The first element is entered in column 1 of line 7, and the next element must be entered in column 11. Any subsequent elements must be entered 10 spaces from the beginning of the previous element entry (e.g., a third element would be entered starting with column 21 on line 7). Note that there is no field requirement on line 8 when using a DOS editor, and when entering data directly into the model, there are no field requirements at any time.

A final note regarding data entry concerns the use of alphanumeric characters. No special requirements exist when using a DOS editor, but when entering an alphanumeric character into the model itself (e.g., 'B1', 'n', 'Z6MAX'), these entries must be enclosed in **single quotes** as shown. To **verify** this subroutine, refer to the BIOPLT input section of Table 16.4..

Table 16.5.1.

BIOPLT Subroutine Input (REQUIRED).

Input Line	Variable Name	Definition
1		*Enter 444444 to access this subroutine.*
2	REC	Number of records desired as output. Ten variables are required for each record created.
	FOR EACH RECORD CREATED:	
3	GMM	Specify whether graphics are to be generated (0=yes; 1=no).
	REGRC	Specify whether regional output is to be generated (1=yes; 0=no); this must be "yes" if plants are to be plotted.
	PLTRC	Specify whether plant output is to be generated (1-yes; 0=no).
4	XPTC	Enter the record identification number (1.0 for record #1, etc.)
	XPTC	Specify daily (1.0) or yearly (0.0) output
	FOR EACH VARIABLE ENTERED:	
5	IPTC	Enter variable name (see glossary for list of available variables)
6	VARR	Enter number of array elements for variable selected on line 5 (see glossary for array elements within each variable)
	ALPH	Enter the numeric equivalent of the letter of the alphabet with which the variable selected on line 5 begins (e.g., "B1" begins with "B," which is the 2nd letter of the alphabet, so the value 2.0 would be entered here)
7	KPTC	Enter the array identifier(s) for the variable selected on line 5 (see glossary for array identifiers specific to each variable)
8	APTC	For each array identifier, enter specific output instructions. To have all values printed, enter 0.0. To have only part of an array printed, enter the highest array value for each identifier (e.g., 2.0 would cause the first two values of that particular identifier to be printed). To have only one value of an identifier printed, enter the negative value of that number. NOTE: If there are two array identifiers for the variable entered on line 5, then two values would be entered on this line; three array identifiers require three values, etc.
		AFTER ALL RECORDS HAVE BEEN CREATED:
9	STYR	Enter the year output is to begin
10	APLOTS	Enter the number of plots to be created (0.0 bypasses this option)

16.6. TRUPLT Subroutine - Data Selection For Plotting

16.6.1. Description

TRUPLT is used to select specific output, created in BIOPLT, for plotting on a per-plot basis. Thus, for each plot created, the user must enter a complete TRUPLT input, followed by a complete TPLOTR input (discussed in section 16.7.). Subsequent plots consist of their own set of TRUPLT/TPLOTR input.

16.6.2. Input/Output

The required input for the TRUPLT subroutine is presented in Table 16.7.. When creating input for TRUPLT, it is important not to exceed eight variables per plot (input line 2) or the model will terminate its run with an error statement. This also prevents the production of excessively "cluttered" graphs.

A shorthand method for entering multiple identical numbers on the same line of input, such as might be encountered in input line 3, 6, 7, or 8, entails the use of a multiplication statement. For example, if the user wishes to input 6 entries of "1.0" on input line 3, he would type "6*1.0."Lines 6, 7, and 8 are used as additional selection devices. Line 6 allows a specific plant type or animal age to be selected for plotting, line 7 dictates that a specific soil layer, tree diameter, water layer, contaminant, or plant organ dose be used for plotting, and line 8 is a selector for a specific region or animal type. In this simulation all selector options (SELCT1, SELCT2, and SELCT3) have been set to 0.0 to override these options. The user may wish to experiment with them, although they are not used in most cases.

Table 16.7.

TRUPLT Subroutine Input (REQUIRED).

Input Line	Variable Name	Definition
		FOR EACH PLOT CREATED:
1		*Enter 171717 to access this subroutine:*
2	VARPLT	Enter the number of variables to be plotted (not to exceed 8 variables per plot)
3	SREC	For each variable to be plotted, enter the record number (created in BIOPLT) in which it is stored (e.g., if 3 variables are to be plotted and they each come from record #1, enter "1.0 1.0 1.0")
4	XVAR	Enter a column value for the x-axis (entering 13.0 means the x-axis of the created graph will represent time)
5	XNO	Enter the number of each variable to be plotted [e.g., if the first three variables within a record (see line 3 for record number) are to be plotted, enter "1.0 2.0 3.0"]
6	SELCT1	For each variable to be plotted (i.e., if 3 variables are to be plotted, 3 entries are required on this line), an additional selector for plant type and animal age may be entered here. A default value of 0.0 for each variable indicates that no additional selection devices are being used for plotting.
7	SELCT2	For each variable to be plotted, an additional selector for soil layer, tree diameter, water layer, contaminant, or plant organ dose may be entered here. This is needed when selecting particular soil layers within particular plants. A default value of 0.0 for each variable indicates that no additional selection devices are being used for plotting.
8	SELCT3	For each variable to be plotted, an additional selector for a specific region or animal type may be entered here. A default value of 0.0 for each variable indicates that no additional selection devices are being used for plotting.

16.7. TPLOTR Subroutine - Plotting Instructions

16.7.1. Description

After the data to be plotted has been selected, TPLOTR is used to create the actual output graphs. For each plot to be created, a complete sequential set of input for TRUPLT and TPLOTR must exist.

16.7.2. Input/Output

Table 16.8. lists the input required for the TPLOTR subroutine. If color graphics are chosen (input line 4), the code will select the color for each plotted variable. If color graphics are not desired, input line 3 allows the user to distinguish between variables by selecting different data point markers for the individual variables. Other plotting options include the choice of using natural log values on either axis and the choice of entering maximum and minimum values for the axes or of having the code calculate these values. A default value of (- 111.0) for any minimum or maximum value for the axes will although code selection of these parameters.

A final note concerns the legend titles (input line 11). Each title is limited to 12 characters, and this includes any spaces or symbols used.

Table 16.8.

TPLOTR Subroutine Input (REQUIRED)

Input Line	Variable Name	Definition
		FOR EACH PLOT CREATED:
1		*Enter **191919** to access this subroutine.*
2	NXL	Enter the number of letters in the x-axis title (e.g., "time" has 4.0 letters)
	NYL	Enter the number of letters in the y-axis title
	NTIT	Enter the number of letters in the plot's title
3	ITYPE	For each variable to be plotted, enter the type of markers for each data point (0=none, 1=digits, 3=letters, 4=symbols)
4	IFLAG	Specify whether color graphics are desired (y or n)
5	IFLAG	Specify whether a legend is desired (y or n)
6	IFLAG	Specify whether log values are desired on the y-axis (y or n)
7	IFLAG	Specify whether log values are desired on the x-axis (y or n)
8	IXL	Enter the x-axis title
9	IYL	Enter the y-axis title
10	LTITLE	Enter the title of the plot
11	ISTL	For each variable, enter a legend title (one per line). A maximum of 12 letters are permitted per legend title.
12	XPP	Enter the maximum value on the x-axis (-111.0=code calculates value)
	XPP	Enter the minimum value on the x-axis (-111.0=code calculates value)
13	YPP	Enter the maximum value on the y-axis (-111.0=code calculates value)
	YPP	Enter the minimum value on the y-axis (-111.0=code calculates value)

Repeat input for TRUPLT and TPLOTR for each plot created.

16.8. AIRAC Subroutine- Acute and Chronic Releases of Contaminants

16.8.1. Description

The subroutine AIRAC is utilized to simulate both acute and chronic air contaminating events from accidental and continuous releases of radionuclides and other substances. Figure 16.5. illustrates this scenario, in which contaminants are considered to be homogeneously mixed in a space encompassing the air 30m above the soil surface over a ground area of $1m^2$.

Depending upon the input, the units attached to each value vary: pCi/m^3 for radionuclides, g/m^3 for stable nuclides, etc. All entered values are added to **A1**, the daily air contaminant concentration estimated in other subroutines from weathering, plant and animal mortality, and soil resuspension processes.

The simulation of an acute event (a spike) on day 325 is illustrated in Figure 16.6., and the simulation of chronic contamination is shown in Figure 16.7. Both of these inputs are taken from Example 1, set in Amarillo, Texas. Note that the air concentrations input to AIRAC are subsequently utilized in other subroutines, such as PLTGRO, UPTAKE, ANIMAL, and HUMTRN for simulations involving soil deposition, resuspension, uptake, and intakes specified by the user's scenario.

16.8.2. Input/Output

The subroutine AIRAC is capable of simulating both chronic and acute contamination events. Furthermore, each region simulated may have its own release structure (e.g., multiple acute events, more than one contaminant, etc.). Both chronic and acute releases can be modeled in the same scenario, and they can be structured to integrate *between* chronic and acute events. A description of input requirements for executing AIRAC is presented in Table 16.9..

Example 1 is used to illustrate both the acute and chronic capabilities of AIRAC, encompassing the subroutines MAIN, AIRAC, CLIMATE, WATFLX, BIOPLT, TRUPLT, and TPLOTR (Tables 16.10. and 16.11.). Since this chapter solely examines air contaminant concentrations, only one output variable was plotted: **AIRCA** (resuspendable contaminant concentration in air per m^3). A portion of this output for both the acute and chronic simulations is shown in Table 16.12. The first graph (Figure 16.6.) simulates a puff of uranium oxide particles with a concentration of 1×10^6 pCi/m^3 on the 325th day of the first simulation year. This type of acute event was common for the DOE firing and testing sites at Pantex in Amarillo, Texas. The second graph (Figure 16.7.) simulates a chronic release with an initial maximum air concentration of 1×10^6 pCi/m^3. This simulated release begins in year one and continues for the duration of the scenario (four years). A sine equation was chosen to simulate the release, and it is negatively damped with an exponential function. This simulation mimics the pattern observed for global radioactive fallout since the early 1960's.

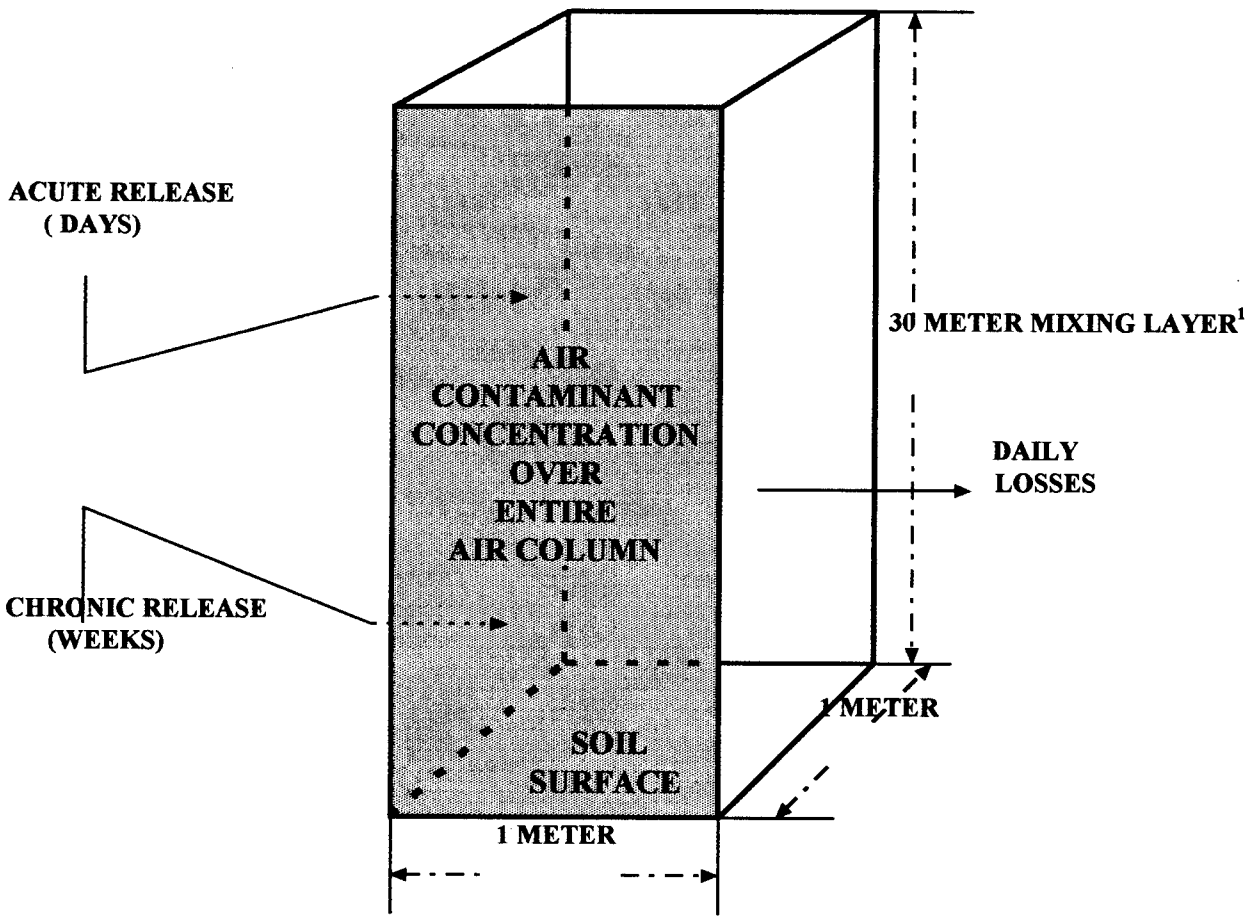


Figure16.5.

Acute and Chronic Air Contaminant Concentrations Are Simulated Above the Soil Surface in AIRAC

Table 16.9.

AIRAC Subroutine Input.

Input Line	Variable Name	Definition
1		*Enter 111111 to access this subroutine.*
2	ARCZ	Open (1.0) or close (0.0) contaminant event gates for each contaminant in the simulation
3		CHRAC Open (1.0) or close (0.0) the chronic particulate gate.
4		CHRAG Open (1.0) or close (0.0) initial chronic gaseous concentration gate.
		NOTE: If lines 3 and 4 are 0.0, go to *ACUTE EVENT INPUT* (line 18).
		CHRONIC EVENT INPUT:
5	CHRED	Enter Julian day number for the end of chronic exposure
6	CHREY	Enter year number for end of chronic exposure
7	CHRSD	Enter Julian day number for start of chronic exposure
8	CHRSY	Enter year number for start of chronic exposure
9	EQTYP	Enter type of chronic equation used (1=exponential, 2=linear, 3=ingrowth, 4=sine, 5=constant) DAMPE Enter of equation dampening (1=exponential, 2=linear, 3=ingrowth, 4=constant)
		If chronic equation 1 was chosen, go to line 10. **If chronic equation 2 was chosen, go to line 11.** **If chronic equation 3 was chosen, go to line 12.** **If chronic equation 4 was chosen, go to line 13.** **If chronic equation 5 was chosen, go to line 14.**
10	ARIN	Enter initial air concentration, mass, or activity/m^3
	AEXP	Enter decay/growth half-time (days) for exponential equation (see line 9)
11	ARIN	Enter initial air concentration, mass, or activity/m^3
	AEXP	Enter decay/growth half-time (days) for linear equation (see line 9)
12	ARIN	Enter maximum air concentration

Table 16.9.

AIRAC Subroutine Input (continued).

Input Line	Variable Name	Definition
	ARFL	Enter minimum air concentration
	AEXP	Enter decay/growth half-time (days) for ingrowth equation (see line 9)
13	ARIN	Enter maximum amplitude air concentrations
	ARFL	Enter minimum amplitude air concentrations
	AEXP	Enter lag/lead constant for determining maximum concentrations for sine equation (see line 9)
14	ARIN	Enter air concentration, mass, or activity units for soil for constant equation (see line 9)
		If dampening equation 1 was chosen, go to line 15. **If dampening equation 2 or 3 was chosen, go to line 16.** **If dampening equation 4 was chosen, go to line 17.**
15	AEXY	Enter decay/growth dampening/enhancement half-time for exponential function
16	AEXY	Enter proportionality constant (+ or -; do not exceed 1.0) for linear or ingrowth dampening/enhancement function
17	AEXY	Enter air concentration dampening constant (1.0 maintains repeatable variables)
		ACUTE EVENT INPUT:
18	CYR	Enter the number of acute events
19	DURC	Enter the duration (in days) of the longest acute event for contaminant *j* in region *l*
20	YRCNT	Enter the year number(s) of acute releases
21	DAIC	Enter the Julian day(s) of initial acute releases for year *x*
22	AMAC	Enter the particulate concentration(s) for duration of acute event(s) (one entry per day of acute release; 0.0 cancels)
23	AMAG	Enter the gaseous concentrations for duration of acute event(s) (one entry per day of acute release; 0.0 cancels)

Table 16.10.

Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario. Note That The Script Notation of the Individual Subroutines is Used for Clarification Purposes Only.

MATN Subroutine

```
 4.000000
 1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 1.000000
1.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000      1.000000
```

ATRAC Subroutine

```
 111111.0
 1.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000
 3.000000
 1.000000
 325.0000
 1000000.000      10000.00      500.0000
0.0000000E+00  0.0000000E+00  0.0000000E+00
```

CLTMAT Subroutine

```
 555555.0
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
 0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
```

Table 16.10.

Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario (continued)

WATFLX part 1 Subroutine

```
141414.
 0.000000E+00   0.000000E+00
 0.000000E+00
 0.000000E+00
 0.000000E+00
 0.000000E+00
```

WATFLX part 2 Subroutine

```
151515.
 92.0000      7.00000      7.00000      15.0000      15.0000      19.0000
 238.000      14.0000      14.0000      30.9800      30.9800      39.1000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 2.00000
 2.00000      0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
 1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 5.00000
 640.000
 20.0000      80.0000      200.000      210.000      130.000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.400000      0.400000      0.300000      0.200000      0.700000
0.400000      0.400000      0.350000      0.150000      0.200000
0.200000      0.200000      0.350000      0.650000      0.100000
 12.0000      12.0000      20.0000      29.0000      9.00000
 27.3000      27.3000      35.7000      44.1000      21.0000
 26.0000      26.0000      34.0000      42.0000      20.0000
 1.00000      1.00000      1.00000      1.00000      1.00000
 20.0000      80.0000      200.000      210.000      130.000
```

PLTGRO Subroutine

```
 121212.
0.000000E+00
0.000000E+00
0.000000E+00
0.000000E+00
```

UPTAKE Subroutine

```
131313.0
     0
```

BIOPLT Subroutine

```
444444.0
1.000000
0.0000000E+00  1.000000     0.0000000E+00
1.000000       1.000000
AIRCA
     2.   1.
```

Table 16.10.

Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario (continued)

```
m       n
 0.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
  1.000000
  1.000000
TRUPLT Subroutine
171717
  1.000000
  1.0
  13.00000
```

1.000000

Table 16.10.

Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario (continued)

```
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
TPLOTR Subroutine
191919
        4        6        18
l
y
y
n
n
year
pCi/m3
Pantex Acute Event
air radioact
 -111.0000     -111.0000
 -111.0000     -111.0000
```

Table 16.11.

Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario. Note That the Script Notation of the Individual Subroutines is Used for Clarification Purposes Only.

MAIN Subroutine

```
 4.000000
 1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 1.000000
1.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000      1.000000
```

ATRAC Subroutine

```
 111111.0
 1.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 1.00000
 0.0000000E+00
 365.000
 4.00000
 1.000000
 1.000000
 4.000000    1.000000
 1000.000      500.00      40.0000
 -1.00000
  0.0000000E+00
```

CLIMAT Subroutine

```
 555555.0
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
22.00000
 3538.000
```

Table 16.11.

Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario (continued)

```
0.0000000E+00
  202.0000
  36.00000
WATFLX part 1 Subroutine
  141414.
   0.000000E+00   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
WATFLX part 2 Subroutine
  151515.
   92.0000      7.00000      7.00000      15.0000      15.0000      19.0000
   238.000      14.0000      14.0000      30.9800      30.9800      39.1000
   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   2.00000
   2.00000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
   1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   5.00000
   640.000
   20.0000      80.0000      200.000      210.000      130.000
   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   0.400000     0.400000     0.300000     0.200000     0.700000
   0.400000     0.400000     0.350000     0.150000     0.200000
   0.200000     0.200000     0.350000     0.650000     0.100000
   12.0000      12.0000      20.0000      29.0000      9.00000
   27.3000      27.3000      35.7000      44.1000      21.0000
   26.0000      26.0000      34.0000      42.0000      20.0000
   1.00000      1.00000      1.00000      1.00000      1.00000
   20.0000      80.0000      200.000      210.000      130.000
PLTGRO Subroutine
   121212.
   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
UPTAKE Subroutine
  131313.0
      0
BIOPLT Subroutine
  444444.0
  1.000000
 0.0000000E+00  1.000000    0.0000000E+00
```

Table 16.11.

Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario (continued)

```
 1.000000      1.000000
AIRCA
      2.   1.
m       n
 0.000000E+00  0.0000000E+00

AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
AIRCA
      2.   1.
m       n
 -1.000000E+00  0.0000000E+00
  1.000000
  1.000000
TRUPLT Subroutine
171717
```

Table 16.11.

Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario (continued).

```
  1.000000
  1.0
  13.00000
  1.000000
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
TPLOTR Subroutine
191919
        4        6        18
l
y
y
n
n
year
pCi/m3
Pantex Acute Event
air radioact
 -111.0000     -111.0000
                                        -111.0000     -111.0000
```

Table 16.12.

Output From Both the Acute and Chronic Simulations of Example 1, Amarillo, Texas, for Day 325-day 345 of the First Simulation Year. Note That There Are Two Output Lines Per Day.

ACUTE	CHRONIC
1.00E+06	5.05E+02
1.00E+06	5.05E+02
1.00E+04	5.05E+02
1.00E+04	5.05E+02
5.00E+02	5.06E+02
5.00E+02	5.06E+02
0.00E+00	5.07E+02
0.00E+00	5.07E+02
0.00E+00	5.08E+02
0.00E+00	5.08E+02
0.00E+00	5.10E+02
0.00E+00	5.10E+02
0.00E+00	5.11E+02
0.00E+00	5.11E+02
0.00E+00	5.12E+02
0.00E+00	5.12E+02
0.00E+00	5.13E+02
0.00E+00	5.13E+02
0.00E+00	5.15E+02
0.00E+00	5.15E+02
0.00E+00	5.16E+02
0.00E+00	5.16E+02
0.00E+00	5.18E+02
0.00E+00	5.18E+02
0.00E+00	5.19E+02
0.00E+00	5.19E+02
0.00E+00	5.21E+02
0.00E+00	5.21E+02
0.00E+00	5.23E+02
0.00E+00	5.23E+02
0.00E+00	5.25E+02
0.00E+00	5.25E+02
0.00E+00	5.27E+02
0.00E+00	5.27E+02
0.00E+00	5.29E+02
0.00E+00	5.29E+02
0.00E+00	5.31E+02
0.00E+00	5.31E+02
0.00E+00	5.33E+02
0.00E+00	5.33E+02
0.00E+00	5.35E+02

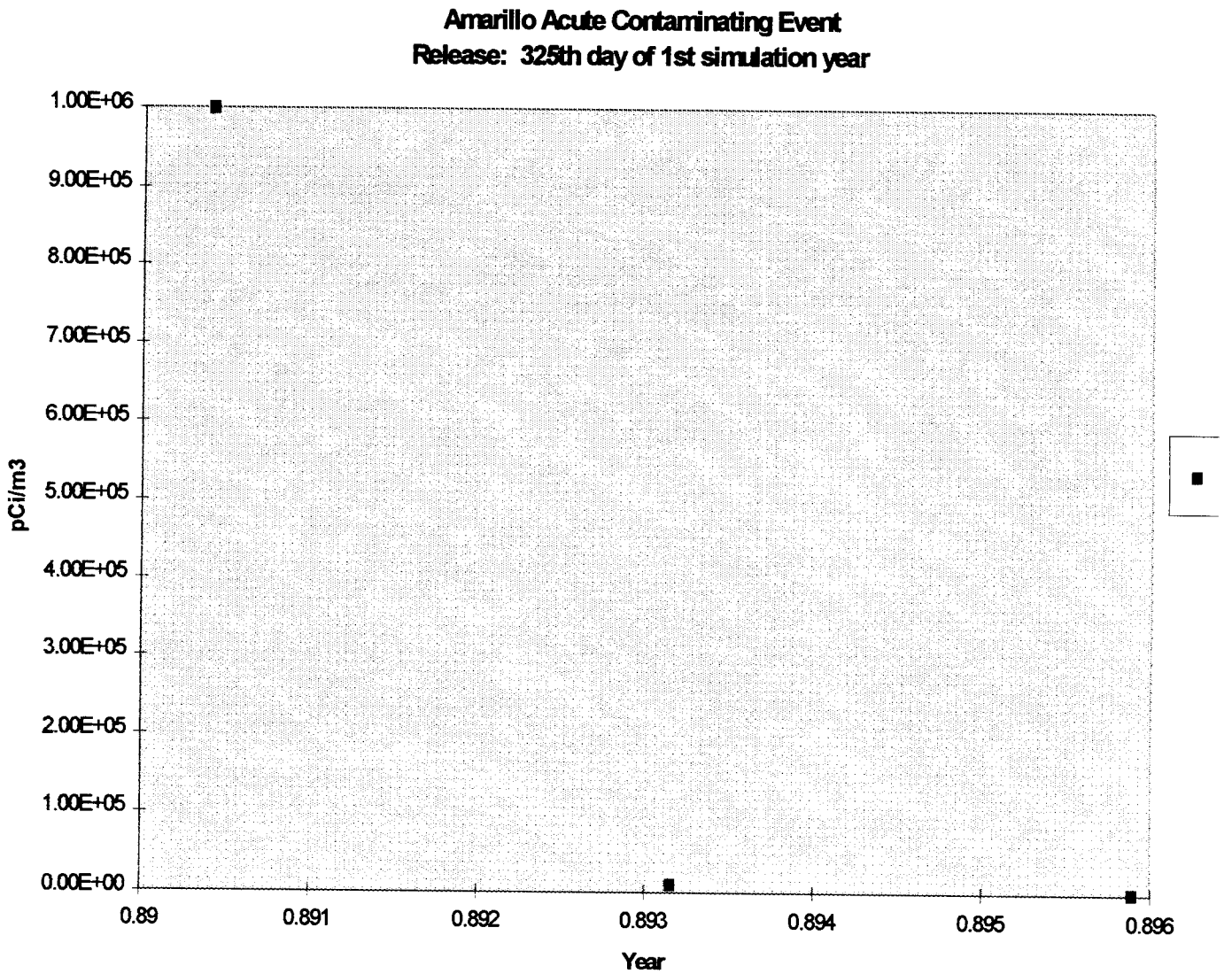


Figure 16.6.

Simulated Acute Contaminating Event for the Amarillo, Texas, Air Contamination Scenario for Four Years

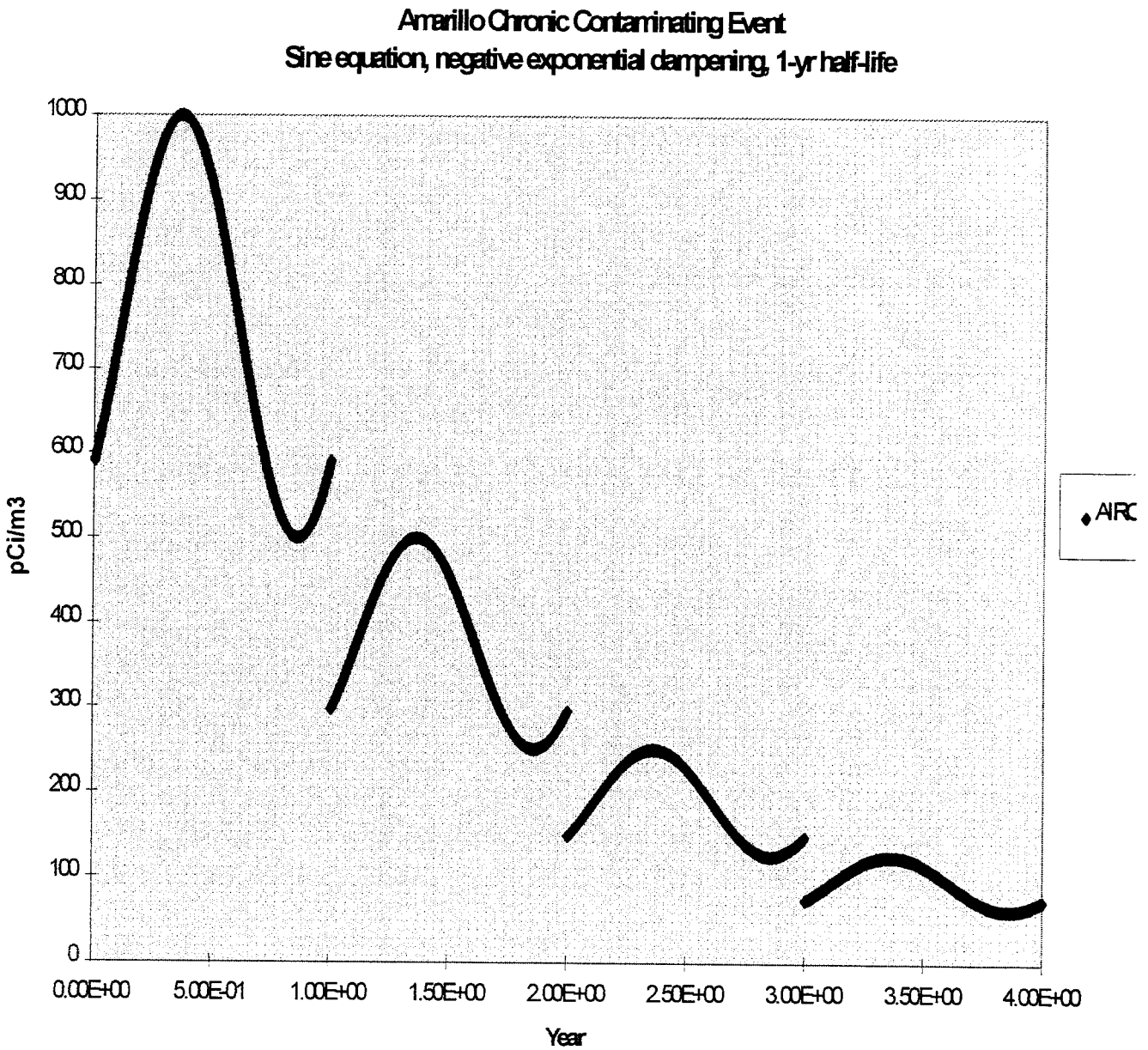


Figure 16.7.

Simulated Chronic Contaminating Event For the Amarillo, Texas, Air Contamination Scenario for Four Years.

16.9. WATFLX, RUNOFF, EROSON, AQUIFER, AND GEOFLX Subroutines - Soil Moisture Flow, Erosion, Watershed Simulations

16.9.1 Description

The development of subroutine WATFLX was based primarily on soil hydrology experiments conducted by Hillel and Von Bavel (1976) and Scholl (1976). Some verification studies of this subroutine were conducted using data from Amarillo, Texas, Pullman soils (Unger and Pringle, 1981). A schematic of the subroutine is presented in Figure 16.8. It is generalized for soil layers of varying thickness, and the user can specify how many soil layers are present up to the maximum in the PARAMETER statement of the code. The equations developed by Hillel and Von Bavel (1976) are based on sand, silt, and clay content of a given layer and thus, these values, in addition to rock content, are required as input by WATFLX on a per-layer basis.

WATFLX estimates the amount of soil moisture present in each layer on a daily basis. The movement of moisture between layers is simulated using a Darcy saturated-unsaturated flow relationship based on the initial holding capacity, field capacity, and wilting points entered by the user for each soil layer. If plants are to be simulated, then additional input is required to model transpiration losses and root uptake of contaminants and nutrients. These losses and uptakes are in turn based upon potential evapotranspiration estimates obtained from the CLIMAT subroutine. The CREAMS model (Knisel, 1980) was used as a template to model moisture evaporation from the top 20mm of soil in WATFLX, and this process is related to potential evapotranspiration and plant leaf area index (LAI).

Surface runoff and subsurface return flow are modeled in subroutine RUNOFF using a similar approach to the one taken in the USDA SPUR model (Wight, *et al.*, 1982). This approach enables BIOTRAN.2 to simulate several subwatershed regions simultaneously and allows the user to utilize detailed input, such as slope, vegetation type, and soil type, to create a detailed simulation of each subwatershed. Additionally, RUNOFF can interact with WATFLX's irrigation module, suppressing surface flow for agricultural crops and augmenting subsurface flow as a result of this constraint. However, WATFLX can operate without the RUNOFF subroutine, using surface runoff and subsurface flow as moisture sinks when these processes are not required by the user.

Subroutine EROSON, in conjunction with RUNOFF, interacts with WATFLX to simulate erosion events. The EROSON model is based on erosion studies in the Los Alamos, New Mexico, and Tucson, Arizona regions (Lane, *et al.*, 1985). Simulated erosion events are due to a differential rate of soil (or alluvium) particle detachment as a function of particle diameter. Subroutine AQUIFER governs the movement of soil moisture into an unconfined aquifer and its subsequent transport to the distal end of the watershed. It should be noted that this subroutine assumes trapezoidal interconnecting alluvial aquifer connections between channel segments (Abrahams, *et al.*, 1962; Purtymun, 1974; Purtymun, *et al.*, 1977).

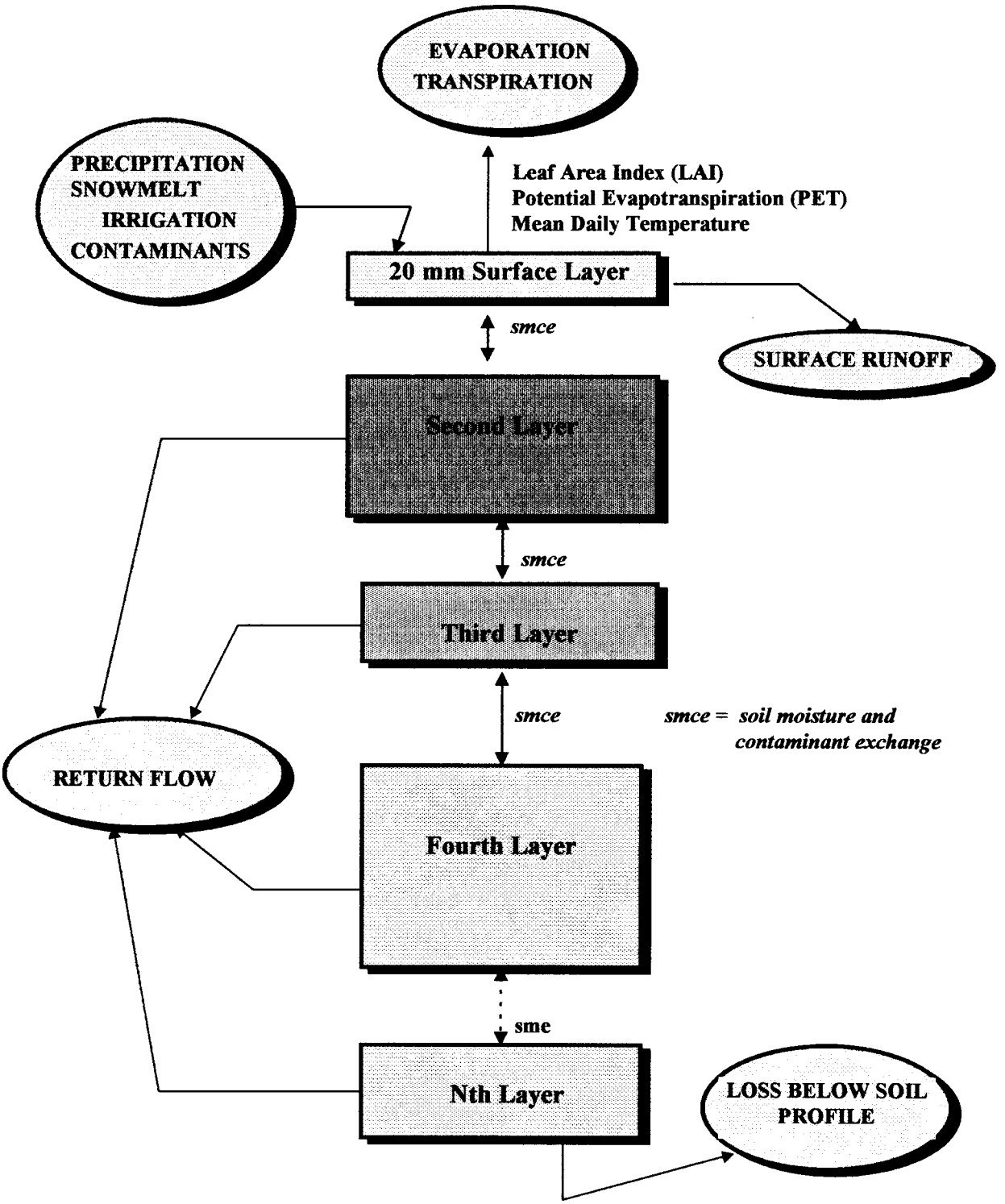


Figure 16.8.

Illustration of the 1 m^2 Soil Profile For Each Region Simulated in WATFLX. The User Specifies the Depth and Soil Characteristics For Each Soil Layer

Finally, subroutine GEOFLX processes runoff events by interfacing WATFLX with the other subroutines previously described in this chapter. A portion of GEOFLX is governed by the Modified Universal Soil Loss Equation (MUSLE) (Williams, 1975). This is almost identical to the USLE, except the R factor is replaced with a runoff erosivity factor. Use of this equation allows sediment yields from each watershed to be estimated for individual storm events.

WATFLX can be executed in the absence of plant growth simulation to create a bare soil scenario. This output can then be compared to a simulation of the same site with forested or cropped areas on it to observe how water dynamics are influenced by the presence of plants. Additionally, although nitrogen, potassium, and phosphorus cycling are automatically simulated in BIOTRAN.2 using default concentrations, the user may specify these concentrations if desired.

16.9.2. Input/Output

The input requirements for the operation of WATFLX and the associated subroutine GEOFLX are shown in Table 16.13.. The subroutines RUNOFF, AQUIFER, and EROSON do not require direct user input. These subroutines can operate with a large number of default conditions if the user does not have the actual input data available and/or does not require a high degree of scenario resolution.

Input from **Example 1**, which simulates five soil layers, is used to verify this subroutine. Table 16.14. contains input for bare soil simulation (with default input for PLTGRO), and Table 16.15. shows the same input with a simulated corn crop. The input for the other subroutines required to run the code are also shown for each table. Uranium oxide is the simulated contaminant.

Graphical output from WATFLX without plant growth simulation is shown in Figures 16.9., 16.10., and 16.11. using the input from Table 16.14.. Figure 16.9. depicts the variables **P1** (cumulative moisture addition to top layer of soil from all sources except ponded water, mm/day) and **EVAP** (evaporation rate, $mm/m^2/day$) over a four-year period. **Z1** (contaminant concentration in soil, pCi/gdwt) for the first and second soil layers is shown in Figure 16.10., and **WA** (soil moisture content in a given soil layer, mm) is presented in Figure 16.11. for the second and third soil layers.

By way of comparison, these same variables are plotted in Figures 16.12. (**P1** and **EVAP**), 16.13. (**Z1**), and 16.9.7. (**WA**) using the input shown in Table 16.15. to simulate an irrigated corn crop (see the next section for a description of the PLTGRO subroutine). Evaporation increases as the corn plants begin to grow (Figure 16.9. *vs.* 16.12.), and soil uranium concentration increases in the resuspension (*i.e.*, first) layer due to a washoff of uranium from plant surfaces (Figure 16.10. *vs.* 16.13.). Additionally, because of evapotranspiration by the plants, soil water content decreases in both the second and third soil layers (Figure 16.11. *vs.* 16.14.).

In addition to comparing bare soil to crops, an interesting comparison can be made between plant types' effects on soil moisture. In Figures 16.15. and 16.16., an irrigated corn crop's effects on runoff and aquifer volume are compared to the effects of a deep-rooted, non-irrigated perennial range grass. Specifically, the variables examined are **CFSPD** (channel peak flow, cfs), **VLAT** (lateral surface flow into channel, cfd), **VRFL** (subsurface return flow, cfd), **VAQF** (aquifer volume, cfd), and **VLOSS** (aquifer loss, cfd). Input for these scenarios is shown in Tables 16.16. and 16.17., respectively. In both scenarios, the land surface covers 4000 acres and has a 20% slope (approximately 11°). A shallow aquifer with an overlying channel slope of 20% is simulated at the base of the watershed.

16.9.3. References

Abrahams, J.H., Jr., Baltz, E.H. and W.D. Purtymun. 1962. Movement of perched ground water in alluvium near Los Alamos, New Mexico. U.S. Geological Survey. Prof. Paper 450-B.

Bras, R.L. 1990. Hydrology: an introduction to hydrologic science. Addison-Wesley, Reading, Massachusetts.

Hillel, D. and C.H.M. Von Bavel. 1976. Simulation of profile water storage as related to soil hydraulic properties. Soil Science J. 40: 807-815.

Knisel, W.G. (ed.) 1980. CREAMS: A field-scale model for chemicals, runoff, and erosion from agricultural management systems. USDA Science and Education Administration Conservation Report No. 26, USDA, Washington, D.C.

Lane, L.J., Purtymun,W.D., and N.M. Becker. 1985. New estimating procedures for surface runoff, sediment yield, and contaminate transport in Los Alamos County, New Mexico. Los Alamos National Laboratory report LA-10335-MS.

Purtymun, W.D. 1974. Dispersion and movement of tritium in a shallow aquifer in Mortandad Canyon at the Los Alamos Scientific Laboratory. Los Alamos Scientific Laboratory report LA-5716-MS.

Purtymun, W.D., Buchholz, J.R., and T.E. Hakonson. 1977. Chemical quality of effluents and their influence on water quality in a shallow aquifer. J. of Env. Quality 6: 29-32.

Scholl, D.G. 1976. Soil moisture flux and evapotranspiration determined from soil hydraulic properties in a chaparral stand. Soil Sci. 40: 14-18.

Unger, P.W., and F.B. Pringle. 1981. Pullman soils: distribution importance, variability, and management. Texas Ag. Experiment Station, USDA, ARS, and SCS, B-1372.

White, F.M. 1986. Fluid mechanics. McGraw-Hill, New York.

Wight, J.R., *et al.*, 1982. SPUR--simulation of production and utilization of rangelands: a rangeland model for management and research. USDA, Ag. Research Service, Miscellaneous Pub. 1431.

Williams, J.R. 1975. Sediment-yield prediction with universal equation using runoff energy factor. *In* Present and prospective technology for predicted sediment yields and sources. ARS-S-40, USDA Agricultural Research Service.

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input.

Input Line	Variable Name	Definition
WATFLX PART I:		
1		***Enter 141414 to access WATFLX part I.***
2	PHEFF	Open (1.0) or close (0.0) gate for simulating effect of soil pH on nuclide transport. If this gate is opened, input line 21 in WATFLX part II must be entered.
	CLAYEF	Open (1.0) or close (0.0) gate for simulating effect of clay content on nuclide transport.
3	CHNNU	Enter number of simulated watershed channels (0.0=not applicable; if >0, GEOFLX is accessed).
4	SEDGAT	Enter year that watershed simulation is initiated (0.0=not applicable).
5	AQUGAT	Open (1.0) or close (0.0) gate for aquifer portion of the model (if gate is opened, GEOFLX is accessed).
6	BDRK	Enter fraction of each region which is impervious to water (*e.g.*, if there are eight regions in the simulation, eight entries are required here).

If GEOFLX was accessed, the following section of input must be entered; otherwise, skip to WATFLX part II:

GEOFLX:		
1		***Enter 888888 to access GEOFLX.***
2	AKOND	Enter hydraulic conductivity for each channel (in/hr; 0.0=defaults to 3.0) (*e.g.*, if there are four channels in the simulation, four entries are required here).
3	C1HR	Enter flow duration constant for each channel (hr; 0.0 defaults to 2.53).
4	C2HR	Enter flow duration exponent for each channel (0.0 defaults to 0.2).
5	C5HR	Enter constant for determining peak flow for each channel (ft^3/sec; 0.0 defaults to 4.82).

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
Input lines 6-8 must be entered for each channel before proceeding to the next channel:		
6	DSZ	Enter 7 diameter classes (mm; a minimum permitted value of 0.062 and a maximum of 4.0) for this channel. Then, enter the median diameter (mm) of the alluvium. Eight entries are required on this line.
7	DSF	Enter 7 mass fractions for each diameter class above.
8	DZAS	Enter 7 concentrations (activity/g, ppm, *etc.*) for each diameter class above.
		Repeat lines 6-8 for each channel in the simulation.
9	ENTCHN	Enter the number of channels tributary to each channel in the simulation (enter 0.0 if there are none). *E.g.*, if there are four channels in the simulation, four entries are required on this line.
10	RNCHN	Enter the number of lateral flow surfaces attached to each of the channels in the simulation. *E.g.*, if there are four channels in the simulation, four entries are required on this channels, enter the identity number of these tributary channels. *E.g.*, if there are four channels in the simulation, and the third channel (say, channel 10.0) had two tributaries, enter the identity of these tributary channels (say, 9.0 and 11.0). If the fourth channel also had tributaries, the next line of input would ask for the identity of the fourth channel's tributaries.
12		CHLAT Enter the identity number of the region bordering the first channel in the simulation. Repeat this line of input for each subsequent channel.
13		CHNAQFor each channel in the simulation, enter the identity number of the those regions containing open aquifers (one entry per channel; enter 0.0 if none).
14	TCH	Enter the top posterior channel widths (ft) for each channel simulated. Enter 0.0 if no aquifer is present.
15	BCH	Enter the bottom posterior channel widths (ft) for each channel simulated. Enter 0.0 if no aquifer is present.
16	HTP	Enter posterior channel heights (ft) for each channel simulated. Enter 0.0 if no aquifer is present.

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
17	HBT	Enter anterior channel heights (ft) for each channel simulated. Enter 0.0 if no aquifer is present.
18	TBCH	Enter top anterior channel widths (ft) for each channel simulated. Enter 0.0 if no aquifer is present.
19	BBCH	Enter bottom anterior channel widths (ft) for each channel simulated. Enter 0.0 if no aquifer is present.
20	VAQF	Enter initial volume (ft^3) for each channel simulated. Enter 0.0 if no aquifer is present.
21	REGENT	Enter number of regions contributing sediment from runoff to each region (*e.g.*, if eight regions are simulated, eight entries are required here. Enter 0.0 if no sediment is being deposited in a particular region).
22	REGX	For each region that is receiving sediment (see line 21), enter the identity number of the region(s) which is/are contributing the sediment (*e.g.*, if the sixth region was receiving sediment from two other regions, two region identity numbers would be required here; repeat this input for each region receiving sediment).
23		RAREA Enter the area (acres) of each simulated region.
24	DSZ	Enter 7 soil diameter classes (mm; a minimum permitted value of 0.062 and maximum of 4.0) for this region. Then, enter the median soil diameter (mm). Eight entries are required on this line.
25	DSF	Enter 7 mass fractions for each diameter class above.
26	DZAS	Enter 7 soil concentrations (activity/g, ppm, *etc.*) for each diameter class above.
27	C1HR	Enter flow duration constant for each region (hr; 0.0 defaults to 2.53).
28	C2HR	Enter flow duration exponent for each region (0.0 defaults to 0.2).
29	C5HR	Enter constant for determining peak flow for each region (ft^3/sec; 0.0 defaults to 4.84).

Repeat lines 24-26 for each region in the simulation.

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
Input lines 24-26 must be entered for each region before proceeding to the next region:		
30	AKSOIL	Enter constant value "K" (K=soil erodability) for the MUSLE equation[1] for each region (0.0 defaults to 0.2).
31	CSOIL	Enter constant value "C" (C=cover and management) for the MUSLE equation for each region (0.0 defaults to 0.1).
32	PSOIL	Enter constant value "P" (P=erosion control practice) for the MUSLE equation for each region (0.0 defaults to 1.0).
33	ALSOIL	Enter constant value "L" (L=slope length) for the MUSLE equation for each region (0.0 defaults to 1.0).

[1] Modified Universal Soil Loss Equation = MUSLE = (RKLSCP), where R=runoff erosivity factor, K=soil erodibility, L=slope length, S=soil gradient or steepness, C=cover and management, and P=erosion control practice (Williams, 1975).

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
34	AKOND	Enter hydraulic conductivity (in/hr) for each region (0.0 defaults to 2.0).
35		SCHAN Enter slope fraction for each region (0.0 defaults to an internal setting). An additional entry is required after the regional values are entered: 0.0 if defaults were used for each region, or 1.0 if slopes were entered.
36		RCHANEnter Manning's roughness coefficient[2] for each region (0.0 defaults to an internal setting). An additional entry is required after the regional values are entered: 0.0 if defaults were used for each region, or 1.0 if coefficients were entered.
37		XCHANEnter channel length (mi) within each region or the length of the region's border (mi) with another region if no channel exists within that particular region. One entry required for each region simulated.
38	XCHAN	Enter channel length (mi) for each channel in the simulation (0.0 defaults to 1.0).
39	WCHAN	Enter channel width (ft) for each channel in the simulation (0.0 defaults to an internal setting). An additional entry is required after the channel values are entered: 0.0 if defaults were used for each channel, or 1.0 if widths were entered.
40	SCHAN	Enter slope fraction for each channel (0.0 defaults to an internal setting). An additional entry is required after the channel values are entered: 0.0 if defaults were used for each channel, or 1.0 if slopes were entered.
41	RCHAN	Enter Manning's roughness coefficient for each channel (0.0 defaults to an internal setting). An additional entry is required after channel values are entered: 0.0 if defaults were used, or 1.0 if coefficients were entered.
42	VTRNDP	Enter perennial flow rate (cfs) for each channel; enter 0.0 if there is no flow in a channel.

[2] Manning's roughness coefficient is a constant expressed as *n* in the Manning equation: $Q = (1/n)AR^{2/3}S^{1/2}$ (Bras, 1990). This equation is used to calculate streamflow, and *n* has been derived through experimentation for various channel surfaces (asphalt, gravel, etc.). A table of these *n* values can be found in most hydrology textbooks (e.g., White, 1986)

1		*Enter 151515 to access WATFLX part II.*
2	ATNO	At least 6 entries are required on this line. The first entry is the atomic number of the first contaminant in the simulation. Entries 2 through 6 are the atomic numbers of the nutrients necessary for plant simulation: 7.0=NO_3; 7.0=NH_3; 15.0=PO_4(l); 15.0=PO_4(s); and 19.0=K. Any additional contaminants in the simulation may be entered beginning with entry 7.
3	AWT	Enter the atomic weights of the contaminants and nutrients in input line 2 [14.0=NO_3; 14.0=NH_3; 31.0=PO_4(l); 31.0=PO_4(s); and 39.1=K].
4	HFLIF	Enter the half-life for the contaminants and nutrients in input line 2. 0.0 indicates stability or longevity.

Input lines 5-24 must be entered for each region before proceeding to the next region:

5	YZF	Open (1.0) or close (0.0) gate for entering root fractions in soil horizons for each plant simulated (*i.e.*, if 7 plants are to be simulated, 7 entries are required on this line). A value of 0.0 defaults to internal rooting depth estimations by the code. If 1.0 is entered here, input line 24 must also be entered; otherwise, input line 24 is skipped.
6	YAZ	Specify units for each contaminant and nutrient. 0=defaults to internal values for each soil horizon; 1=ppm input, corrected to g/gdwt soil for nutrients or pCi/g for radionuclides; 2=μg/g soil input which is <u>not</u> subsequently corrected; 3=same units as 2, except values are the same for all soil horizons; and 4=organic compound input, used as μg/g soil.
7	YSRT	Enter the year the source term takes effect for each contaminant and nutrient.
8	DSRT	Enter the day the source term takes effect for each contaminant and nutrient.
9	WATCN	Enter the water concentration of each contaminant and nutrient. If soil concentration input is in ppm, then units are in g/L, and if soil input is in μg/g, then units are in μg/L for nutrients or pCi/L for radionuclides.
10	YUP	Enter plant uptake ratios for each contaminant and nutrient, in "amount gdwt plant/amount gdwt soil." 0.0 defaults to UCRL uptake ratios. If values greater than 0.0 are entered, input line 23 must also be entered; otherwise, input line 23 is skipped.
11	RCODE	Enter the number of soil horizons available to each plant type in the simulation (*e.g.*, if there are 7 plants simulated, 7 values are required here). A value of 0.0 defaults to 250

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
		mm, which means that the total soil depth must be a multiple of 250. After the first value is entered, -1.0 can be entered for all subsequent values, indicating that each subsequent value is to be a duplicate of the first value. If each plant type has the identical number of soil horizons available, then plant-specific entry (input lines 13-21) **on a per plant basis** are not required. However, if the number of soil horizons vary between plant types, lines 13-21 must be entered for each plant type.
12	SDPTH	Enter total depth of the soil horizons available to each plant type.

If input line 11 had different soil horizons available to the various plant types in the simulation, input lines 13-24 must be repeated <u>for each plant type</u> in the simulation before proceeding to the next region's entry (which starts again at input line 5). The following lines use plant type #1 as an example, but each cycle through this section of input will address the subsequent plant in the simulation:

13	SLICE	Enter the individual depth (mm) of each soil horizon available to plant #1 (see line 11 for total number of soil horizons).
		Lines 14-17 must sum to 1.0 for each horizon:
14	ROCK	Enter the fraction of rock in each horizon for plant #1.
15	SAND	Enter the fraction of sand in each horizon for plant #1.
16	SILT	Enter the fraction of silt in each horizon for plant #1.
17	CLAY	Enter the fraction of clay in each horizon for plant #1.
18	W1	Enter the wilt point for each horizon (%) for plant #1.
19	H	Enter holding capacity for each horizon (%) for plant #1.
20	F1	Enter the field capacity for each horizon (%) for plant #1.
21	PH	(Enter this line if input line 2 in WATFLX part I opened the pH gate.) Enter the pH for each horizon for plant #1.
22	Z1	Enter the concentration of the first contaminant that is available to plant #1 in each horizon. If there are subsequent contaminants, their concentrations will be listed on lines 23, 24, etc.

Table 16.13.

WATFLX (I and II) and GEOFLX Subroutine Input (continued).

Input Line	Variable Name	Definition
23	ZU	(Enter this line if the input from line 10 was greater than 0.0.) Enter soil-plant uptake coefficients in each horizon for plant #1.
		Repeat input lines 13-23 for each plant type if necessary.
24	ZF	(Enter this line if the root fraction gate was opened in input line 5.) Enter the root fraction in each horizon for plant #1. Root fractions for any other plants in the simulation would be entered on subsequent lines (lines 25, 26, etc.).
		Repeat input lines 5-24 for each subsequent region in the simulation.

Table 16.14.

Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario. Note That The Script Notation of the Individual Subroutines is Used for Clarification Purposes Only.

```
MAIN Subroutine
4.000000
 1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 0.000000
 1.000000
0.0000000E+00
0.0000000E+00
4.0   1.0
AIRAC Subroutine
111111.
 1.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000
 1.000000
 3.000000
 195.0000
1.000000e+08
0.000000
CLIMAT Subroutine
   555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
Watflx Subroutine, Part I
   141414
0.0000000E+00  0.0000000E+00
```

Table 16.14.

Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario (continued)

```
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
Watflx Subroutine, Part II
   151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 1.000000
 2.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
 5.000000
 990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
 0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
 0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 8.8000000E-02  0.6200000     0.2000000     6.1000001E-02  2.1000000E-02
PLTGRO Subroutine
   121212
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.000000
 17.00000
 2.000000
 1.000000
 1.000000
 0.0000000E+00
 4000.000
 365.0000     365.0000
 0.0000000E+00  0.000000     0.000000     0.000000     0.000000
```

Table 16.14.

Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario (continued)

```
0.000000    0.000000    0.000000    0.000000    0.000000
 0.000000    0.000000    0.000000    0.000000    0.000000
 0.000000    0.000000
 150.0000    150.0000    157.0000    164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
UPTAKE Subroutine
 131313
 0
BIOPLT Subroutine
444444
 1.0
0.0  1.0  0.0

1.0  1.0
CZ
3. 3.
i      ks     n
0.0  0.0  0.0
EAVAL
3. 5.
i      ks     n
0.0  0.0  0.0
EVAP
2. 5.
i      n
-1.0  -1.0
P1
2. 16.
i      n
-1.0  -1.0
RNOFDG
2. 18.
i      n
0.0  0.0
RNOFDS
2. 18.
i      n
0.0  0.0
Z1
4. 26.
i      ks     m      n
-1.0  -1.0  -1.0   -1.0
WA
3. 23.
i      ks     n
-1.0  -2.0  -1.0
WA
```

Table 16.14.

Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario (continued)

```
 3. 23.
 i       ks      n
 -1.0  -3.0  -1.0
 Z1
 4. 26.
 i       ks      m       n
 -1.0  -2.0  -1.0  -1.0
  1.0
  3.0
 TRUPLT Subroutine
  171717
 2.0
  1.0 1.0
  13.0
  3.0 4.0
   0.0 0.0
  0.0 0.0
  0.0 0.0
 TPLOTR Subroutine
  191919
 4  4  30
      1     1
 y
 y
 n
 n
 year
 mm/d
 Soil Surface Moisture Dynamics
 evap'tion
 water in
   -111.0  -111.0
   -111.0  -111.0
  171717
  2.0
  1.0  1.0
  13.0
  7.0  10.0
  0.0  0.0
  0.0  0.0
  0.0  0.0
  191919
  4  5  18
  1  1
 y
 y
 n
 n
 year
```

Table 16.14.

Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario (continued)

```
pCi/g
Soil Radioactivity
1st layer
2nd layer
 -111.0  -111.0
 -111.0  -111.0
 171717
 2.0
 1.0  1.0
 13.0
8.0  9.0
 0.0  0.0
 0.0  0.0
 0.0  0.0
 191919
 4  2  19
      1  1
y
y
n
n
year
mm
Soil Layer Moisture
2nd layer
3rd layer
 -111.0  -111.0
 -111.0  -111.0
```

Table 16.15.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario. Note That the Script Notation of the Individual Subroutines is Used for Clarification Purposes Only.

```
MAIN Subroutine
4.000000
 1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 0.000000
 1.000000
0.0000000E+00
0.0000000E+00
4.0   1.0
AIRAC Subroutine
111111.
 1.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000
 1.000000
 3.000000
 195.0000
1.000000e+08
0.000000
CLIMAT Subroutine
   555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
WATFLX Subroutine, Part I
   141414
0.0000000E+00  0.0000000E+00
0.0000000E+00
```

Table 16.15.

Example 1(Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario (continued)

```
0.0000000E+00
 0.0000000E+00
 0.0000000E+00
WATFLX Subroutine, Part II
   151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
1.000000
 2.000000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 5.000000
 990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
8.8000000E-02 0.6200000     0.2000000     6.1000001E-02 2.1000000E-02
PLTGRO Subroutine
   121212
0.0000000E+00
0.0000000E+00
0.0000000E+00
1.0000000E+00
6.0
2.0
1.0
1.0
0.0
4000.
270. 270.
0.0 7.0 6.0 6.0 6.0 6.0
```

Table 16.15.

Example 1(Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario (continued)

```
 150. 150. 201. 229. 243. 257.
UPTAKE Subroutine
 131313
 0
BIOPLT Subroutine
 444444
 1.0
 0.0  1.0  0.0
 1.0  1.0
CZ
3. 3.
i      ks     n
0.0  0.0  0.0
EAVAL
3. 5.
i      ks     n
0.0  0.0  0.0
EVAP
2. 5.
i      n
-1.0  -1.0
P1
2. 16.
i      n
-1.0  -1.0
RNOFDG
2. 18.
i      n
0.0  0.0
RNOFDS
2. 18.
i      n
0.0  0.0
Z1
4. 26.
i      ks     m      n
-1.0  -1.0  -1.0   -1.0
WA
3. 23.
i      ks     n
-1.0  -2.0  -1.0
WA
3. 23.
i      ks     n
-1.0  -3.0  -1.0
Z1
4. 26.
i      ks     m      n
-1.0  -2.0  -1.0  -1.0
```

Table 16.15.

Example 1(Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario (continued)

```
 1.0
 3.0
IRUPLI Subroutine
 171717
 2.0
 1.0 1.0
 13.0
 3.0 4.0
 0.0 0.0
 0.0 0.0
 0.0 0.0
IPLOIR Subroutine
 191919
 4  4  30
      1     1
y
y
n
n
year
mm/d
Soil Surface Moisture Dynamics
evap'tion
water in
  -111.0  -111.0
  -111.0  -111.0
IRUPLI Subroutine
 171717
 2.0
 1.0  1.0
 13.0
 7.0  10.0
 0.0  0.0
 0.0  0.0
 0.0  0.0
IPLOIR Subroutine
 191919
 4  5  18
 1  1
y
y
n
n
year
pCi/g
Soil Radioactivity
1st layer
2nd layer
 -111.0  -111.0
```

Table 16.15.

Example 1(Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario (continued)

```
 -111.0  -111.0
TRUPLT Subroutine
 171717
 2.0
 1.0  1.0
 13.0
 8.0  9.0
 0.0  0.0
 0.0  0.0
 0.0  0.0
IPLOTR Subroutine
 191919
 4  2  19
     1  1
y
y
n
n
year
mm
Soil Layer Moisture
2nd layer
3rd layer
 -111.0  -111.0
 -111.0  -111.0
```

Table 16.16.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% Slope. Note That the Script Notation of the Individual Subroutines is Used for Clarification Purposes Only.

```
MAIN Subroutine
 2.000000
 1.000000
 1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 0.000000
 1.000000
 0.0000000E+00
 0.0000000E+00
 1.0  1.0
CLIMAT Subroutine
 555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
 0.2000000
 3500.000
 4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
 2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
 0.1610000     0.1430000     0.1500000     9.3000002E-02  8.9000002E-02
 3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
 0.0000000E+00
 202.0000
 36.00000
WATFLX Subroutine, Part I
 141414
 0.0000000E+00  0.0000000E+00
 1.000000
 1.000000
 1.000000
 0.0000000E+00
  888888
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 6.1999999E-02  0.1250000     0.2500000     0.5000000     1.000000
 2.000000      4.000000      0.9000000
 1.5000000E-02  4.9999999E-03  5.0000001E-02  0.1200000     0.3900000
```

Table 16.16.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn crop Runoff Scenario at 20% Slope (continued)

```
0.3700000     5.0000001E-02
 2.000000     2.000000     2.000000     2.000000     2.000000
 2.000000     2.000000
0.0000000E+00
 1.000000
 1.000000
0.0000000E+00
 3.000000
 1.500000
 2.000000
 6.000000
 24.00000
 14.00000
0.0000000E+00
0.0000000E+00
 4000.000
6.1999999E-02 0.1250000     0.2500000     0.5000000     1.000000
 2.000000     4.000000     0.1300000
0.4200000     9.0000004E-02 0.1000000     0.1200000     0.1600000
0.1000000     9.9999998E-03
 2.000000     2.000000     2.000000     2.000000     2.000000
 2.000000     2.000000
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.2000000     1.000000
2.9999999E-02  1.000000
 2.500000
 2.500000
 1.500000     1.000000
0.2000000     1.000000
2.9999999E-02  1.000000
0.0000000E+00
WATLX Subroutine, Part II
  151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000
 2.000000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
```

Table 16.16.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% Slope (continued)

```
0.0000000E+00
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 5.000000
 990.6000
 20.00000    132.4000    203.2000    304.8000    330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000    18.70000    18.90000    17.90000    16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 28.50000    28.50000    28.60000    25.80000    26.10000
 2.000000    2.000000    2.000000    2.000000    2.000000
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
PLTBRO Subroutine
  121212
0.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000
 17.00000
 2.000000
 1.000000
 1.000000
0.0000000E+00
 4000.000
 302.0000    302.0000
0.0000000E+00 7.000000    2.000000    2.000000    2.000000
 2.000000    2.000000    2.000000    2.000000    2.000000
 3.000000    3.000000    3.000000    3.000000    3.000000
 3.000000    3.000000
 150.0000    150.0000    157.0000    164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
UPTAXE Subroutine
  131313
  0
BIOPLT Subroutine
  444444
1.0
```

Table 16.16.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% Slope (continued)

```
0.0  1.0  0.0
 1.0  0.0
VTRND
1. 22.
30
-2.0
VLAT
1. 22.
30
-2.0
VRFL
1. 22.
30
-2.0
VAQF
1. 22.
30
-2.0
VLOSS
1. 22.
30
-2.0
CFSPD
1. 3.
30
-1.0
VLAT
1. 22.
30
-1.0
VRFL
1. 22.
30
-1.0
VAQF
1. 22.
30
-1.0
VLOSS
1. 22.
30
-1.0
 1.0
 1.0
TRUPLT Subroutine
171717
5.0
```

Table 16.16.

Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% Slope (continued)

```
1.0 1.0 1.0 1.0 1.0
 13.0
 1.0 2.0 3.0 4.0 5.0
 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0
IPLOTR Subroutine
 191919
 4 3 25
 1 1 1 1 1
y
y
y
n
days
cfd
Runoff and Aquifer Volume
vtrnd
vlat
vrfl
vaqf
vloss
 -111.0  -111.0
 -111.0  -111.0
```

Table 16.17.

Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff Scenario at 20% Slope. Note that the Extra Spacing Between Subroutines and the Script Notation of the Individual Subroutines is used for Clarification Purposes Only.

```
Main Subroutine
 2.000000
 1.000000
 1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 0.000000
 1.000000
 0.0000000E+00
 0.0000000E+00
 1.0  1.0
CLIMAT Subroutine
 555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
WATFLX Subroutine, Part I
 141414
 0.0000000E+00  0.0000000E+00
 1.000000
 1.000000
 1.000000
 0.0000000E+00
GEOFLX Subroutine
 888888
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
6.1999999E-02  0.1250000      0.2500000      0.5000000      1.000000
 2.000000      4.000000      0.9000000
```

Table 16.17.

Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff Scenario at 20% Slope (continued)

```
1.5000000E-02 4.9999999E-03 5.0000001E-02 0.1200000    0.3900000
0.3700000    5.0000001E-02
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
0.0000000E+00
 1.000000
 1.000000
0.0000000E+00
 3.000000
 1.500000
 2.000000
 6.000000
 24.00000
 14.00000
0.0000000E+00
0.0000000E+00
 4000.000
6.1999999E-02 0.1250000    0.2500000    0.5000000    1.000000
 2.000000    4.000000    0.1300000
0.4200000    9.0000004E-02 0.1000000    0.1200000    0.1600000
0.1000000    9.9999998E-03
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
0.2000000    1.000000
2.9999999E-02  1.000000
 2.500000
 2.500000
 1.500000    1.000000
0.2000000    1.000000
2.9999999E-02  1.000000
0.0000000E+00
WATFLX Subroutine, Part II
  151515
 92.00000    7.000000    7.000000    15.00000    15.00000
 19.00000
 238.0000    14.00000    14.00000    31.00000    31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000
```

Table 16.17.

Example 1 (Amarillo, Texas): Input used to execute the rangeland runoff scenario at 20% slope (continued)

```
2.000000    0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 5.000000
 990.6000
 20.00000    132.4000    203.2000    304.8000    330.2000
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000    18.70000    18.90000    17.90000    16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 28.50000    28.50000    28.60000    25.80000    26.10000
 2.000000    2.000000    2.000000    2.000000    2.000000
8.8000000E-02  0.6200000    0.2000000    6.1000001E-02  2.1000000E-02
PLTGRO Subroutine
   121212
0.0000000E+00
0.0000000E+00
0.0000000E+00
 8.000000
 0.00000
 0.000000
 1.000000
 0.000000
0.0000000E+00
 4000.000
UPTAKE Subroutine
131313
   0
BIOPLT Subroutine
   444444
1.0
0.0  1.0  0.0
1.0  0.0
VTRND
1. 22.
30
-2.0
VLAT
1. 22.
```

Table 16.17.

Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff Scenario at 20% Slope (continued)

```
30
-2.0
VRFL
1. 22.
30
-2.0
VAQF
1. 22.
30
-2.0
VLOSS
1. 22.
30
-2.0
CFSPD
1. 3.
30
-1.0
VLAT
1. 22.
30
-1.0
VRFL
1. 22.
30
-1.0
VAQF
1. 22.
30
-1.0
VLOSS
1. 22.
30
-1.0
 1.0
 1.0
IRUPLI Subroutine
 171717
 5.0
 1.0 1.0 1.0 1.0 1.0
 13.0
 1.0 2.0 3.0 4.0 5.0
 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0
IPLOIR Subroutine
 191919
 4 3 34
```

Table 16.17.

Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff Scenario at 20% Slope (continued)

```
1 1 1 1 1
y
y
y
n
days
cfd
Runoff and Aquifer Volume, grasses
vtrnd
vlat
vrfl
vaqf
vloss
-111.0 -111.0
-111.0 -111.0
```

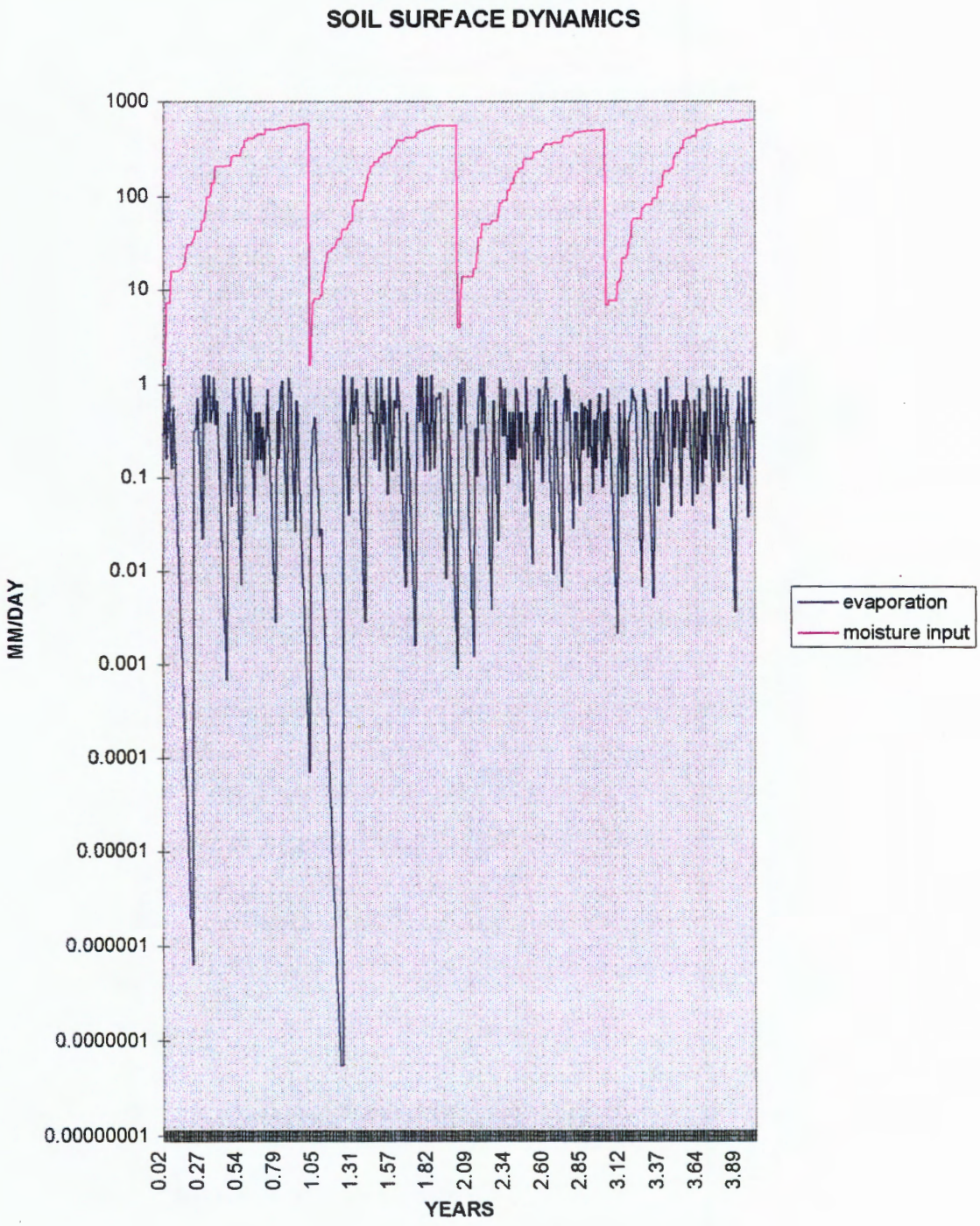


Figure 16.9.

Cumulative Precipitation Input from CLIMAT, P1, and Soil Surface Evaporation, EVAP, as Simulated in WATFLX for Amarillo, Texas Pullman Soils on bare ground.

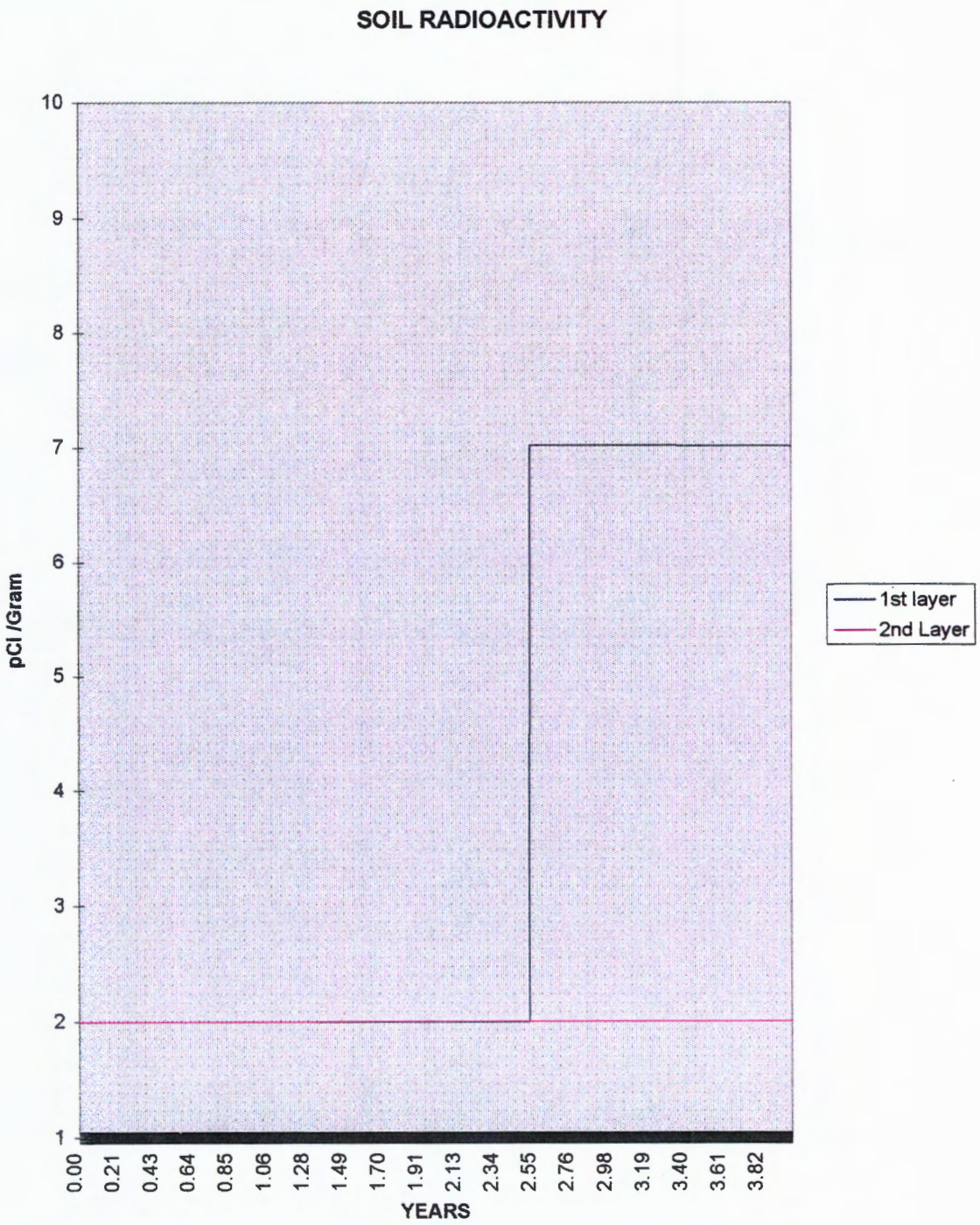


Figure 16.10.

Soil Contamination in pCi/gram for Four Simulation Years. Note That The Contaminating Spike on the Third Year for the Bare Soil Scenario.

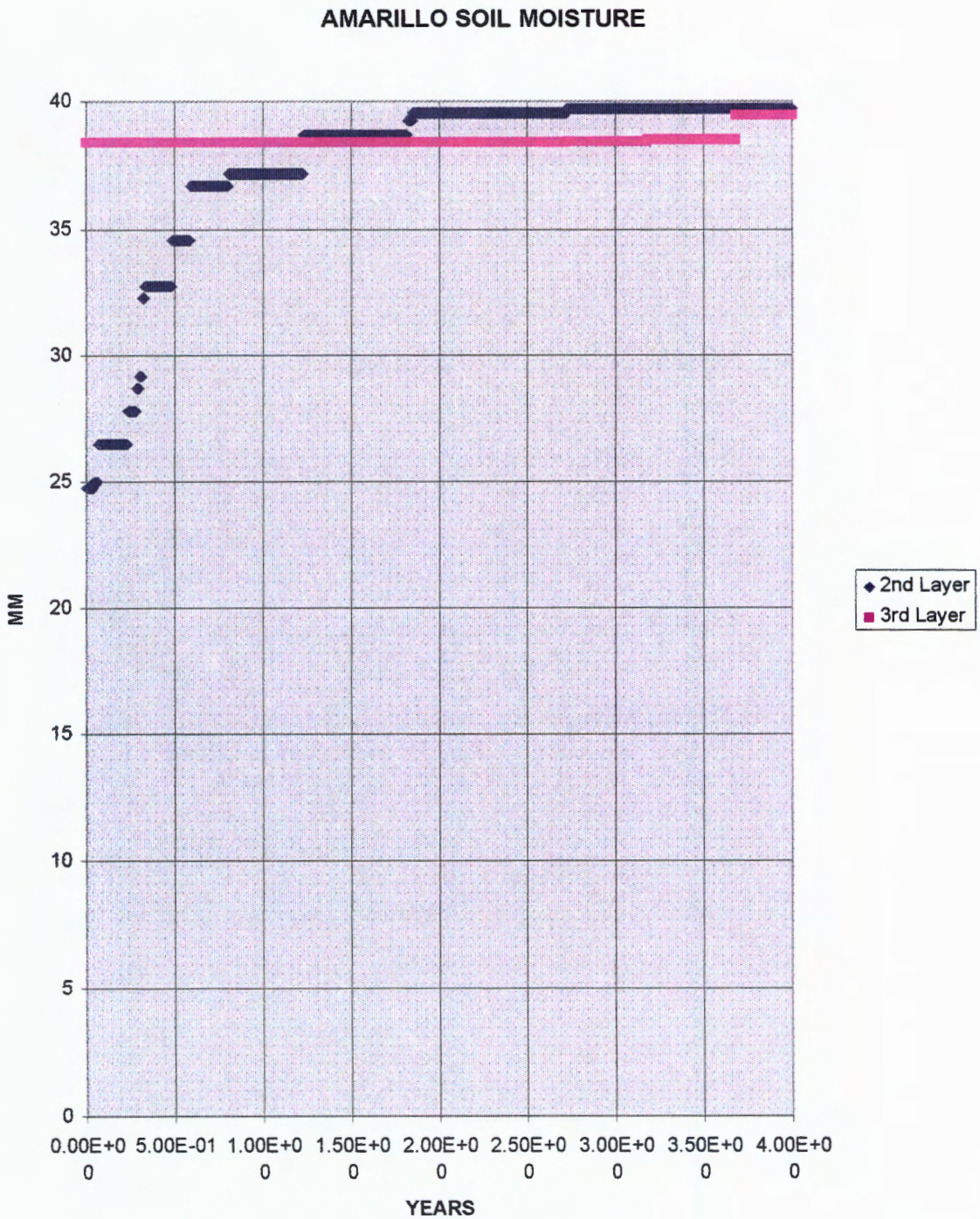


Figure 16.11.

Buildup of Soil Water Content in the Second and Third Soil Layers from Bare Ground Conditions. Loss Through the Bottom of Soil Profile and Evaporation are not Sufficient to Remove All of the Added Soil Moisture. Transpiration of Moisture by Plants is Clearly Absent.

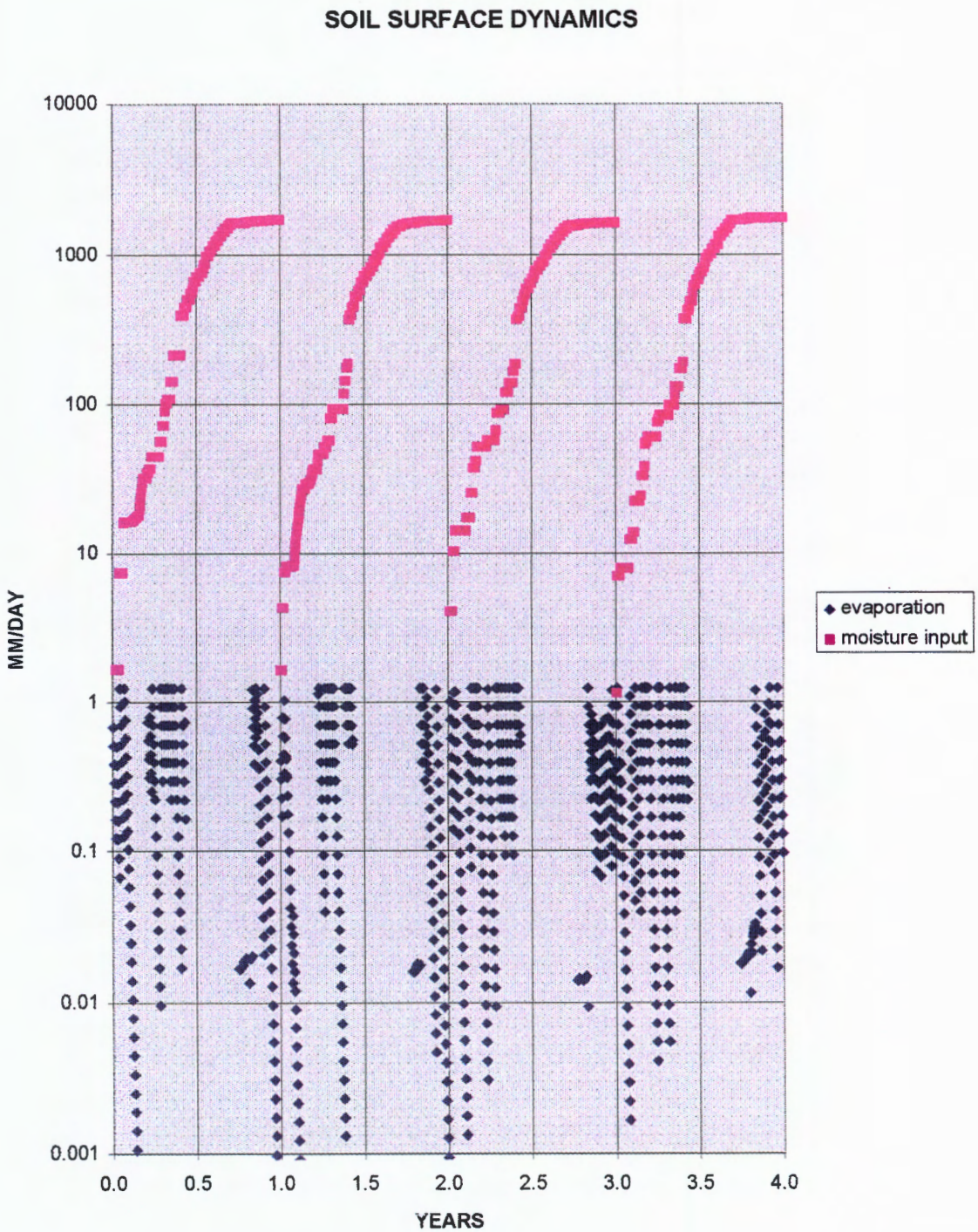


Figure 16.12.

Soil Surface Water Dynamics When Irrigated Corn is Grown on Soils Illustrated in Figures 16.9., 16.10., and 16.11.. Compare This Figure With Figure 16.9. Where Bare Soil is Simulated. Note the Decline in Evaporation During the Growing Season Due to Plant Crown Cover Increase.

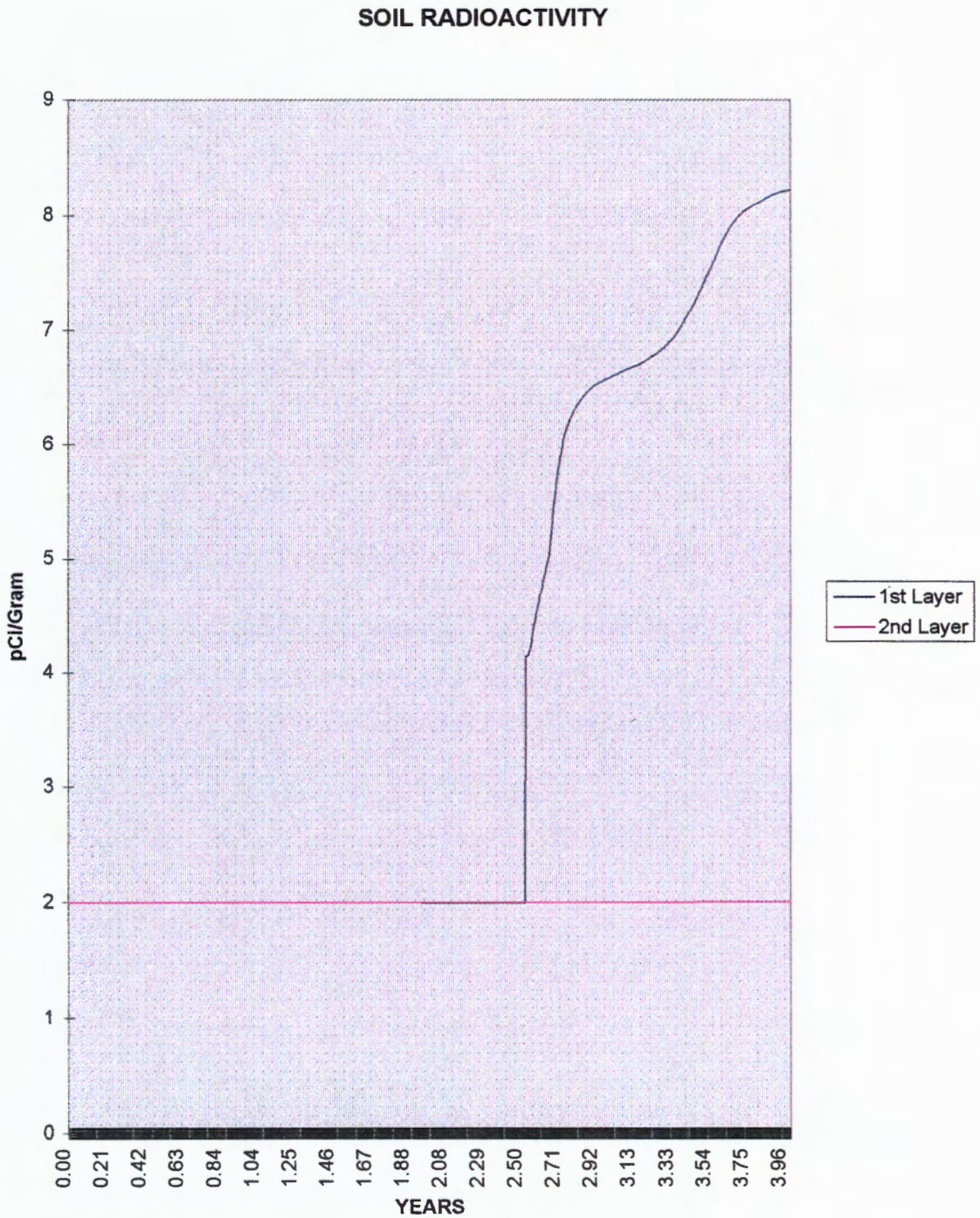


Figure 16.13.

Soil Contamination Increasing in the First layer After the Acute AIRAC Event on Day 195 of the Third Year. The Initial and Subsequent Measure of Soil Surface Contaminants Are Affected by Washoff of Uranium from Plant Surfaces.

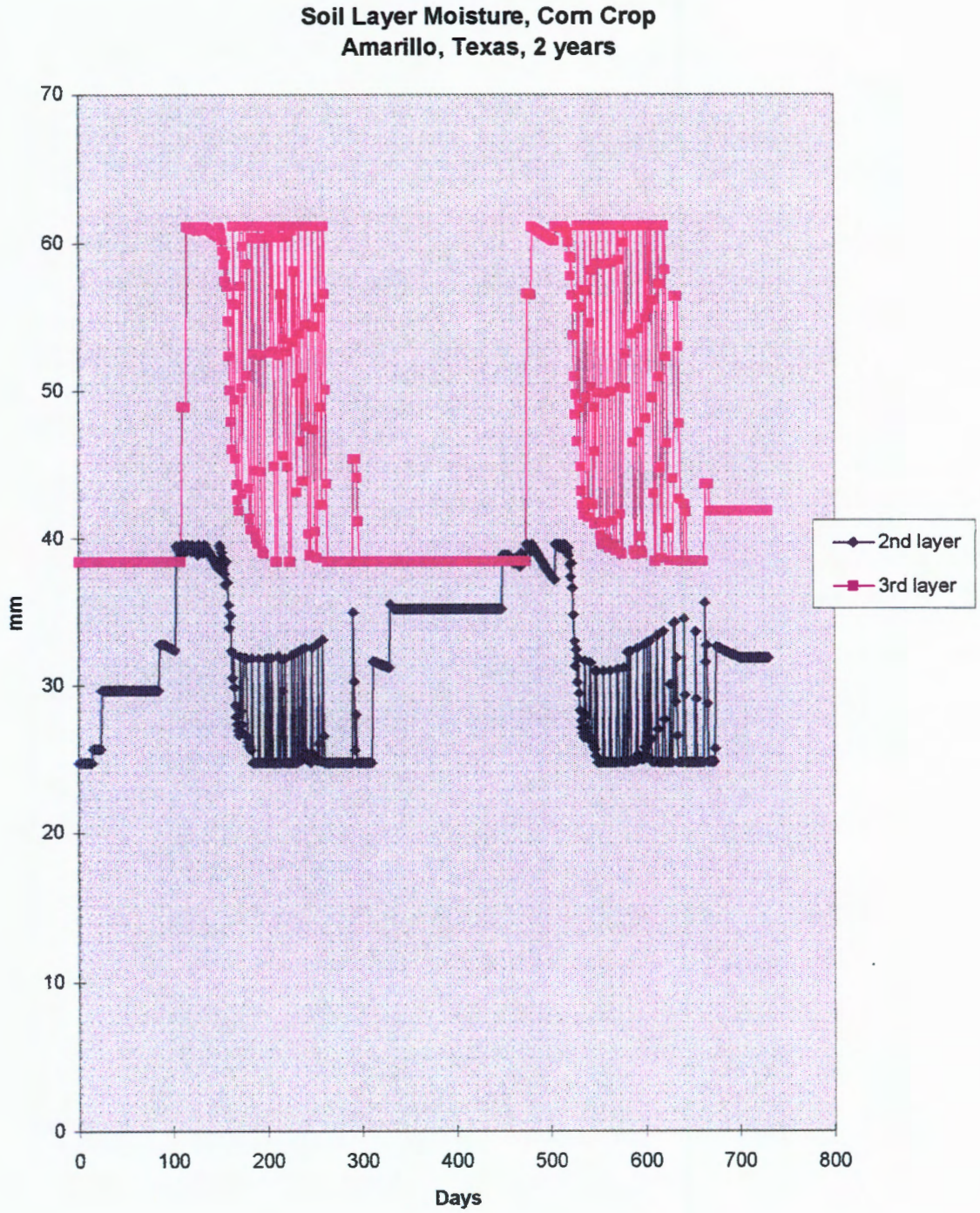


Figure 16.14.

Soil Water Content is Significantly Different in This Scenario Than That Shown in Figure 16.11. The Effect of Plant Transpiration Throughout The Growing Season is Apparent.

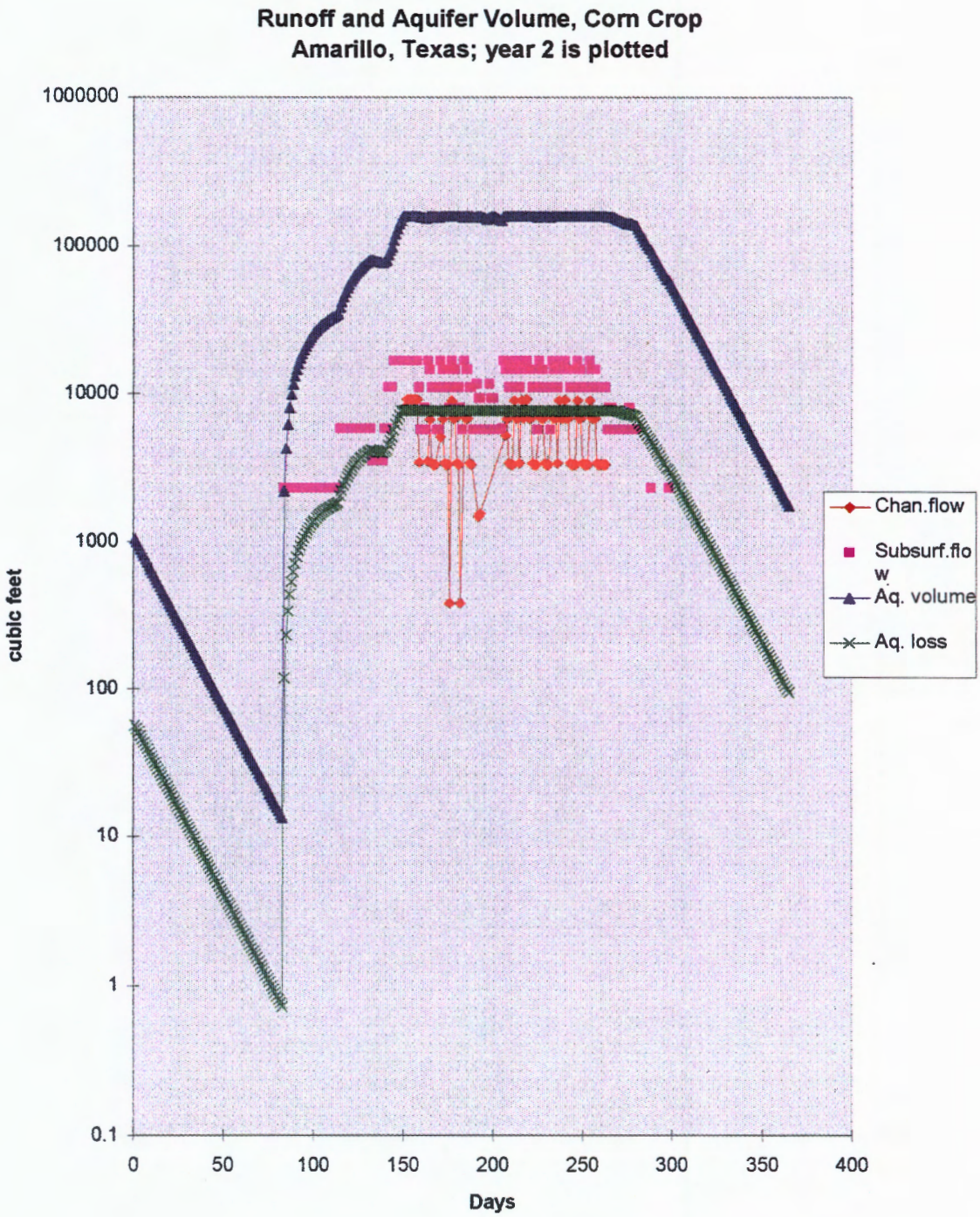


Figure 16.15.
Surface Runoff and Aquifer Volume for Irrigated Corn Scenario for a 4000 Acre Square Watershed With a 20% Slope. Note the Pattern of Aquifer Volume and Loss Compared With

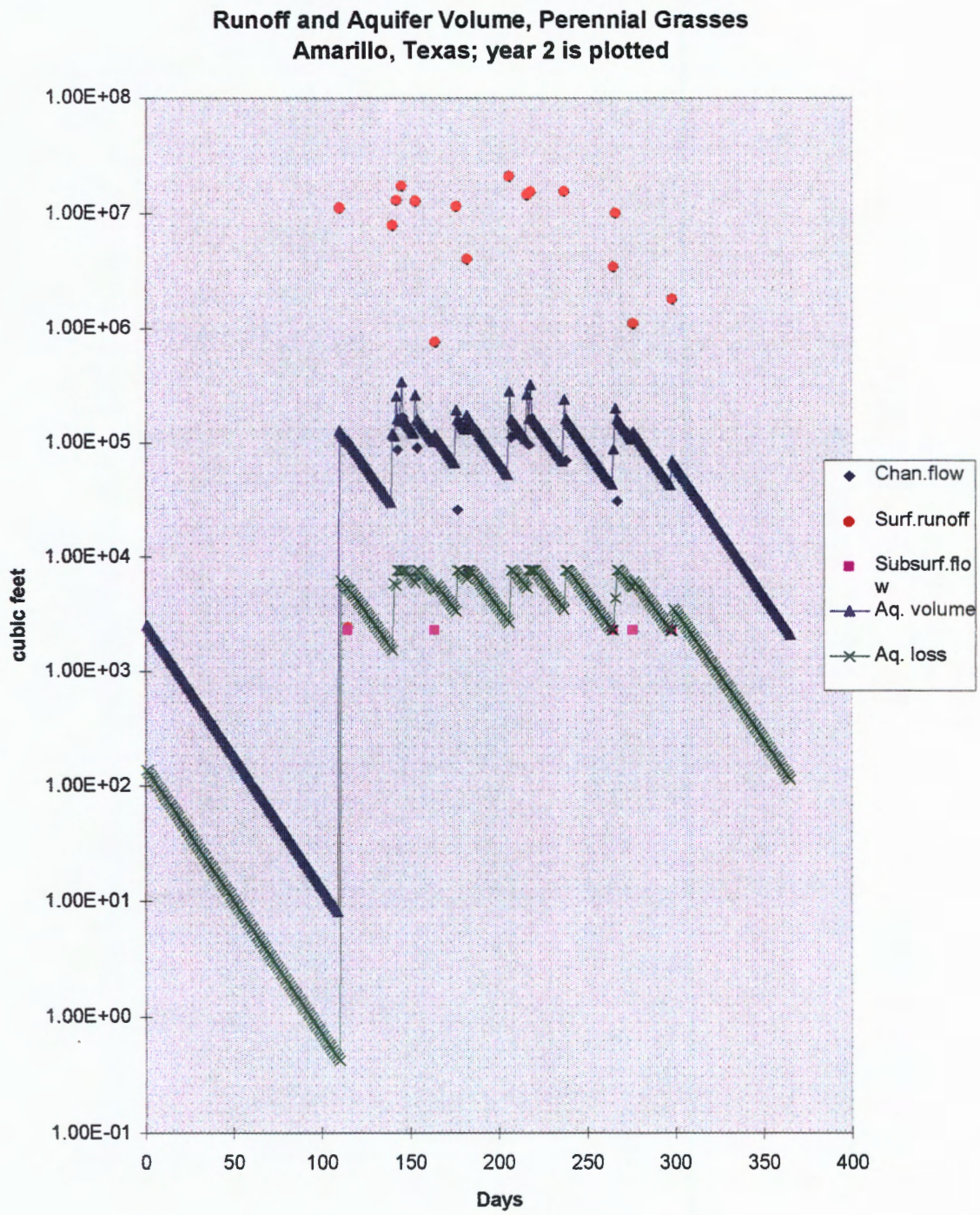


Figure 16.16.

Surface Runoff and Aquifer Volume For Uncropped Warm Season Perennial Grass on the Same Watershed to Compare with Figure 16.15. for Irrigated Corn Which Has Been Furrowed, and Does Not Allow Surface Runoff.

THIS PAGE LEFT BLANK INTENTIONALLY

16.10. PLTGRO Subroutine And Plant Type Simulation

16.10.1 Description

An illustration of simulated plant growth by PLTGRO is presented in Figure 16.17. There are 22 different plant types that can be automatically simulated by this subroutine which range from annual grasses to deciduous and evergreen trees and shrubs (see 15.1. Appendix F). The same plant type can be simulated more than once by setting the plant type number to double, triple..etc. of the original value: PLTID= 23,24,25,...44 for corn, vegetables, pasture grasses,..., cheatgrass, respectively. A third corn simulation would be plant type 45..etc. This allows PLTGRO to be used to simulate food for more than one consumer with the same plant type but differing contaminant sources. Additionally, the user can design new plant strategies by using negative plant type numbers between -1 to -22, however, the absolute value of the plant type cannot be simulated simultaneously...-22, and 22...etc. Finally, the user must provide as input coefficients and fractions required for this purpose as shown for automatically set plant types (see 15.2. Appendix F)

The number of plant types that can be simulated simultaneously in one run of the model is set by the PARAMETER statement in the code (currently set at 8, see Appendix 14.18.). , but any number which will not exceed the memory requirements of the computer in use may be specified in the Source Code, and recompiled. Each plant type simulated has its own soil profile associated with it (specified by WATFLX) which can be identical or different from the other plant types under consideration: a maximum of 10 soil layers/ plant type is currently set by the PARAMETER statement. Input can also be specified so that the same plant type may be simulated under different growth conditions within the same simulation, and provision is also made for the user to define a specific type if it is not available in PLTGRO.

As mentioned previously Growth specifications for given plant types in PLTGRO are stored in DATA statements in array format (See 15.2. Appendix F). They provide strategies and parameters for photosynthesis, and respiration, mortality coefficients, temperature ranges for optimum growth and tolerance, and other information utilized to specify plant growth strategies. This information in combination with simulated temperature and solar radiation obtained from CLIMAT, and soil moisture and evapotranspiration estimates from WATFLX is utilized to simulate both above and below ground growth of plant tissue. Photosynthate is shunted using strategies described in Sutcliffe (1974) to different plant components such as leaves, stems, seeds, and roots (see section 6.1.1.). Contaminant mass balance is conserved through mortality processes, grazing, humus formation, and harvesting strategies by animals and man.

PLTGRO has been previously documented and portions verified and validated (Gallegos 1980 and Wenzel 1982). Also, growth processes include the effects of nitrate-nitrogen, ammonia-nitrogen, and soluble phosphate using Michaelis-Menton type relationships (see section 6.1.4.). Potassium uptake and concentration is also modeled, but is currently not utilized in affecting plant growth. BIOTRAN.2 allows the user to simulate more than one region during a run (PARAMETER currently set at 8) . This makes it possible for the user to simulate a small watershed by partitioning the regions with different vegetative characteristics into sub-watersheds for this purpose. Finally, niche competition and succession of annual grasses and herbs, perennial grasses and herbs, deciduous and evergreen shrubs, and deciduous and evergreen trees can also be simulated for a given region. The plant types are placed into competition for water based on total structural biomass, and crown cover (see section 6.1.1.)

16.10.2. Input/Output

The development of input files for plant growth simulations requires careful planning. The input to PLTGRO specifies which plant type is to be simulated, and determines the planting, irrigation, and harvesting schedules for a plant if it is designated to be a managed crop. Input requirements for executing PLTGRO are presented in Table 16.18..

Example 1 input is presented in Table 16.19. as a portion of a more complex scenario. This example of corn growth is identical to that presented in the last section (Table 16.10.) with the exception of variable output instructions for this plant type, and that six other plant types are being simulated simultaneously in the given scenario. Output from this simulation shows the biomass density of the various plant components for corn, and also plant evapotranspiration for this plant type as presented in Figures 16.17. and 16.18. , respectively, for year three of the simulation.

Example 2 is also presented in Table 16.19. As mentioned above, this example simulates seven plant types, each with separate soil and crop management strategies as reflected in the WATFLX and PLTGRO portions of the input. The plant types simulated are designated as: 1=sorghum, 2=**warm season garden vegetables**, 8=**warm season perennial grasses**, 11=**alfalfa**, 12=**winter wheat**, 18=**cool season perennial grasses**, 23=corn. The highlighted plant types above are plotted for illustration of this example. Note that plant types 1 and 23 refer to the same plant type as described previously, but have different crop management strategies and soil characteristics. Also, note the different rooting depths assigned to each plant type. The input data for this scenario has been documented by Wenzel (1982). PLTGRO input for the Amarillo, Texas crop growth strategies because garden vegetables are annuals, whereas alfalfa, warm and cool season grasses are perennials, and winter wheat is a biennial. Only one harvest is specified for corn by CUTA. Note that the first field of input for the latter is the final harvest date, and that it is identical with the first harvest date entered in the second field for this variable (this also occurs with garden vegetables). For those crops which are not annuals, a value of 1.0 is placed in the first field of CUTA. Three harvests (or cuttings) are specified for alfalfa on extended Julian days 500, 531, and 577. By contrast winter wheat has only one harvesting on day 560. Dates specified in DIRA should not exceed the last harvest dates when converted to current Julian day numbers during the year of harvest

Output for this example is presented in Figures 16.20. and 16.21., and show daily biomass density and evapotranspiration, respectively, for the five of the plant types simulated.

Example 3 and 4, simulates succession at a LANL site as presented in Table 16.20. Succession and niche competition may be applied to all non-managed plants including annual and perennial grasses and herbs, and deciduous and evergreen shrubs and trees depending on user needs and interpretation. The transition from one square meter plots of homogeneous plants to large areas (acres) is readily assumed to be straightforward for crops through the use of ACRES, however, for mixed plant communities the use of succession and niche competition (separate or together) must be utilized to simulate the growth of two or more plants located on the same area using RSCDE, or it must be understood that the are growing in separate areas. In the second example described above plant types 1 and 8 (warm and cool season perennial grasses and herbs) are placed into niche competition through the use of RSS which adjusts daily photosynthate production for the competing plants growing on 4000 acres (not 8000 acres combined). Thus, when range plants such as these are grazed by simulated herbivores and omnivores using the ANIMAL subroutine, grazing preferences become important range management parameters. An accelerated version of secondary succession is also available in the code.

Output for succession involving grasses, shrubs, and trees on a LANL site is presented in Figure 16.22.. Some studies have been initiated at LANL to verify this type of simulation, particularly the timing pattern for the appearance of the different plant types involved. Finally, a plot of the same Pinyon-Juniper woodland showing growth of plants without either niche or succession competition strategies is presented in Figure 16.23.. as Example 4.

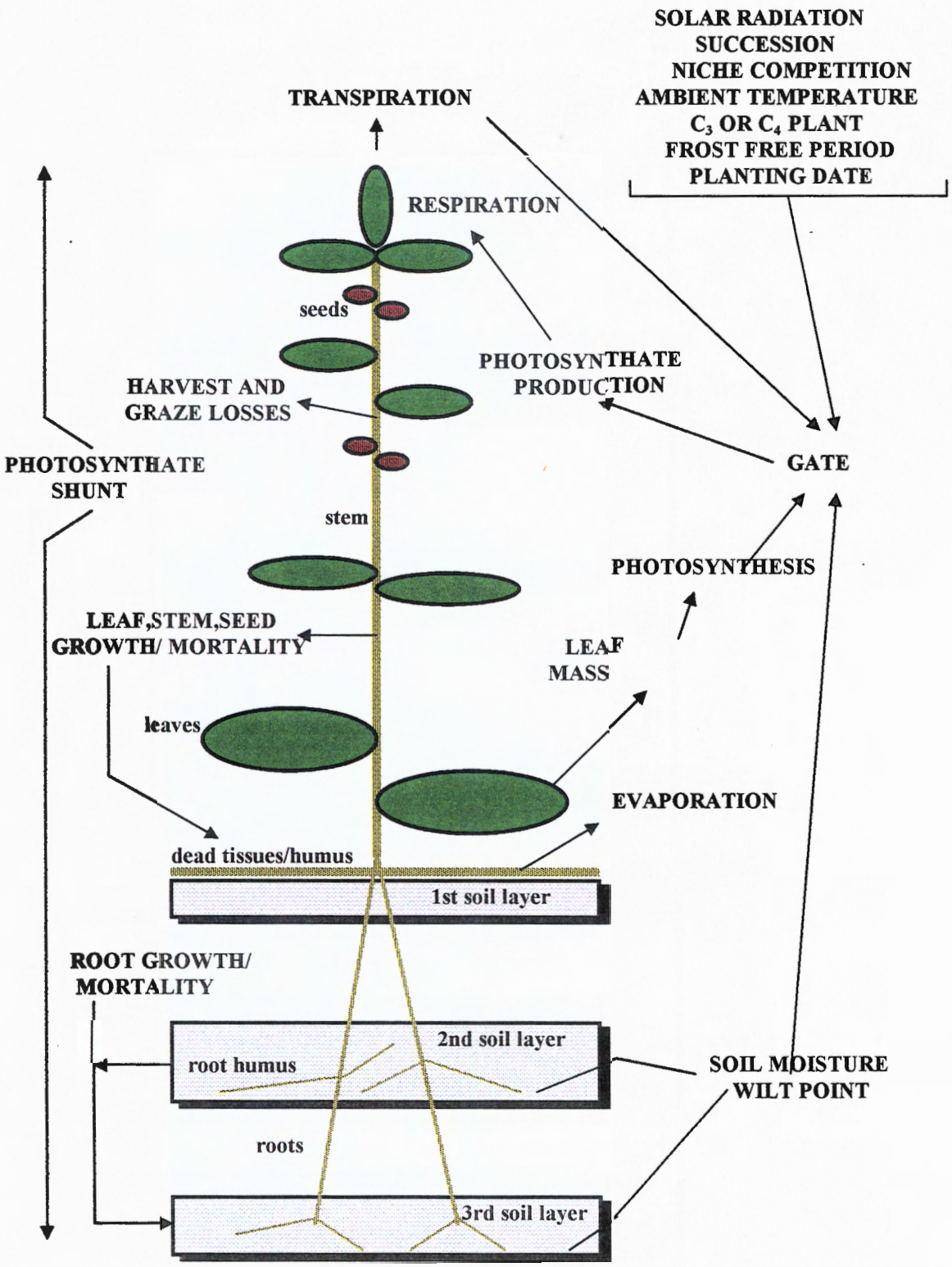


Figure 16.17.

Biomass Growth Processes Simulated In PLTGRO For 22 Different Plant Types With Three Soil Layers

Table 16.18.

PLTGRO Subroutine Input

Input Line	Variable Name	Definition
1.		*Enter 121212 to access PLTGRO*
2.	RSCDE	Enter plant succession gate (0=close, 1=open). An open gate indicates non-managed plants are to be placed under succession.
3.	AICHE	Enter plant niche competition gate (0=close, 1=open). An open gate indicates that non-managed plants are to be placed under niche competition.
4.	ADSUC	Enter advanced plant succession gate (0=close, 1=open). An open gate indicates that succession will proceed at an accelerated rate.
5.	PLTID	Enter the plant type numbers for APL (see Table 16.2.) plants. The plant types are presented in section 15.1., Appendix F. Review rules for multiple simulations of the same plant type or a user designed plant type in section 16.10.2..
6.	ARIG	Enter the number of water applications (irrigations) during the growing season for APL plants. Enter 0.0 if plant is not irrigated. The units are in acre-inches.
7.	AHRV	Enter the number of harvest dates for APL plants. Enter 0.0 if harvests are not modeled for a given plant type. If harvests are modeled, then a minimum of 2.0 harvests must be entered even if only one harvest is specified because the first entree is reserved for the final harvest date. That is, a specification of 2 harvests would require 3 columns of input: the 2nd harvest date, the 1st harvest date, and the 2nd harvest date again. Three harvest dates would require 4 columns of input..etc.. The reason for this requirement will be addressed later in this table (see CUTA below)
8.	RGC	Enter plant identification codes for APL plants (1,2,3). A code value of 1 = FORMAN subroutine not accessed (forest structure model), 2= FORMAN subroutine accessed on a yearly basis, 3= FORMAN model accessed on a daily basis. Code values 2.0, and 3.0 are used primarily for tree plant types, and code=3.0 is not recommended for tree plant types , but has been used to structure agricultural crops such as corn. If codes 2 or 3 are entered, then the user must be prepared to enter regression structural relationships in subroutine FORMAN. Code=1.0 is routinely used for non-tree plant types and for irrigated crops.
9.	RIRRC	Enter harvest/irrigation gate (0,1,2) for APL plants. A value of 0.0= no , 1.0= yes, and 2.0= yes with succession. This input will not allow irrigated plants to be placed into

Table 16.18.

PLTGRO Subroutine Input (continued)

		succession (1.0), unless it is desired to do so (2.0). A value of 0.0 or 2.0 always allows a given plant type to placed into succession provided the RSCDE gate has been opened (see 1.0 above).
10.	RSS	Enter niche competition matches for APL plants. A value of 0.0 indicates no match. Matched plants .. 1.0,1.0, 4.0,4.0,4.0 etc.. indicates that the plant types with matching RSS values are placed into niche competition. The above is valid only if the AICHE gate has been open (see 3.0 above), otherwise all values should be entered as 0.0.
11.	ACRES	Enter the plant coverage area, acres for APL plants. A default value of 0.0 sets the coverage area at 92162 acres (144 square miles). Remember to set all plants placed into niche competition and/or succession with the same acreage.

If RIRRC is set to 0.0 for any specific plant type of the APL plant types entered, then skip input instructions 12-14.

12.	CUTA	Enter the harvest dates (Julian Day numbers) for a specific plant type (RIRRC>0.0). The first column is reserved for the final harvest date; other dates including the final harvest date are then entered in ascending order. when a value of 1.0 is entered in the first column of input (in lieu of the final harvest date), then harvest day numbers greater than 365 may be used for multi-year plants. If a specific plant type is not harvested, then enter 0.0 twice in the first two columns of input.
13.	AWATA	Enter ARIG (see 6.0 above) irrigation water applications, inches, for a specific plant type beginning with the second column. Enter 0.0 in the first column (not used).
14.	DIRA	Enter ARIG water application dates (Julian day numbers) for a specific plant type beginning with the second column. The first column is reserved for the planting date of that plant type.

If the plant type specified is not to be designed by the user (PLTID ≥ 0.0, see 5.0 above), Then repeat lines 2 to 15 of the input instructions if more than one region is being modeled, otherwise input instructions are complete

for each user designed plant:

15.	HF	Enter mean life of plant, years
16.	WUTP	Enter the water utilization efficiency of the plant, ml-$H_2$0 /gdwt-photosynthate.
17.	SCTP	Enter type of plant, 0=woody, 1=herbaceous.
18.	ASTP	Enter type of plant, 0=deciduous, 1=evergreen.

Table 16.18.

PLTGRO Subroutine Input (continued)

19.	RXPP	root humus decomposition coefficient for the relationship gdwt/m^2-day = 0.1RFP RH1 $e^{RXPP\ temp}$., where temp=soil layer temperature (°C), RH1=rapidly decomposing fraction of root humus, gdwt/m^2 ; see below for RFP definition.
20.	RFP	Enter root humus decomposition coefficient (see above).
21.	PSEP	Enter type of plant, C_4 (Calvin)= 0.5, C_3 (Hatch-Slack)= 1.0.
22.	BCP	Enter fraction of photosynthate remaining after respiration losses are taken into account.
23.	STRANS	Enter mean fraction of photosynthate shunted to above ground biomass.
24.	RTRANS	Enter mean fraction of photosynthate shunted to below ground biomass.
25.	TOPT	Enter optimum temperature (°C) for photosynthesis for the plant.
26.	TOFL	enter absolute difference between optimum temperature for photosynthesis (TOPT) and lowest tolerated temperature (°C) for photosynthesis.
27.	TOFH	Enter absolute difference between optimum temperature for photosynthesis (TOPT) and the highest tolerated temperature (°C) for photosynthesis.
28.	FLEAF	Enter initial fraction of above ground biomass assigned to leaves of trees and shrubs. For other types of plants set value=0.0.
29.	ULEF	Enter running average net productivity fraction assigned to leaves of trees and shrubs. For other types of plants set value=0.
30.	RESPF	Enter live-dead stem and branch conversion coefficient: gdwt/m^2 = .0055*RESPF / HF, where HF is defined in line 15.
31	RSPF2	Enter live-to-dead leaf conversion coefficient for herbaceous plants: gdwt/m^2 = [1.1-1/(TOFL-5)] RESPF*WND (ambient temperature below the optimum temperature for photosynthesis). gdwt/m^2 = (RSPF2 (T-TOPT)+0.1) RESPF*WND (ambient temperature above the optimum temperature for photosynthesis). T= mean daily temperature(°C), WND= leaf biomass density, gdwt/m^2.

Repeat lines 2 to 31 of the input instructions as necessary if more than one region is being modeled, otherwise input instructions are complete.

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration In Various Types of Plants

Main Subroutine

```
3.000000
 1.000000
 1.000000
0.000000  0.000000  0.000000   1.000000
 7.000000
 0.000000
 1.000000
0.000000
0.000000
1.0  1.0
```

CLIMAT Subroutine

```
 555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1000001E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.000000
 202.0000
 36.00000
```

WATFLX Subroutine, Part I

```
 141414
0.000000  0.000000
0.000000
0.000000
0.000000
0.000000
```

WATFLX Subroutine, Part II

```
   151515
 92.00000      7.000000      7.000000      15.00000      15.00000
 19.00000
 238.0000      14.00000      14.00000      31.00000      31.00000
 39.10000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
 1.000000      1.000000      1.000000      1.000000      1.000000
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
1.000000     1.000000
 2.000000     0.000000  0.000000  0.000000  0.000000
0.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
 5.000000     5.000000     6.000000     7.000000     6.000000
 6.000000     5.000000
 990.6000     990.6000     1498.600     2006.600     2498.600
 2498.600     990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.000000  0.000000  0.000000  0.000000  0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.000000  0.000000  0.000000  0.000000  0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
 508.0000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.4200000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
0.4590000
 18.70000     18.70000     18.90000     17.90000     16.50000
 15.00000
 30.00000     30.00000     30.10000     27.20000     27.50000
 25.80000
 28.50000     28.50000     28.60000     25.80000     26.10000
 24.50000
 2.000000     2.000000     2.000000     2.000000     2.000000
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000    508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000   0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000   0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000   0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 15.00000    15.00000
 30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000    25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000    24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 15.00000
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
30.00000     30.00000     30.10000     27.20000     27.50000
 25.80000
 28.50000     28.50000     28.60000     25.80000     26.10000
 24.50000
 2.000000     2.000000     2.000000     2.000000     2.000000
 2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.000000 0.000000 0.000000 0.000000 0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
5.0099999E-02 0.5100000    0.3300000    0.1600000    7.9999998E-02
2.4000000E-02 7.8999996E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
PLTRO Subroutine
   121212
0.000000
1.000000
0.000000
 1.000000     2.000000     8.000000     11.00000     12.00000
 18.00000     23.00000
 6.000000     17.00000     0.000000  10.00000     5.000000
0.000000  17.00000
 2.000000     2.000000     0.000000     4.000000     2.000000
0.000000  2.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 1.000000     1.000000     0.000000  1.000000     1.000000
0.000000  1.000000
0.000000 0.000000  1.000000     0.000000 0.000000
 1.000000     0.000000
 4000.000     4000.000     4000.000     4000.000     4000.000
 4000.000     4000.000
 302.0000     302.0000
0.000000  7.000000     6.000000     6.000000     6.000000
 6.000000
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
150.0000    150.0000    201.0000    229.0000    243.0000
 257.0000
 302.0000    302.0000
 0.000000  1.000000    1.000000    1.000000    1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
 150.0000    150.0000    157.00001  164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
 1.000000    500.0000    531.0000    577.0000
 0.000000  7.000000    5.000000    5.000000    5.000000
 5.000000    5.000000    5.000000    5.000000    5.000000
 215.0000    215.0000    470.0000    485.0000    502.0000
 514.0000    527.0000    533.0000    544.0000    559.0000
 1.000000    560.0000
 0.000000  16.00000    5.000000    5.000000    5.000000
 250.0000    250.0000    285.0000    425.0000    500.0000
 302.0000    302.0000
 0.000000  7.000000    2.000000    2.000000    2.000000
 2.000000    2.000000    2.000000    2.000000    2.000000
 3.000000    3.000000    3.000000    3.000000    3.000000
 3.000000    3.000000
 150.0000    150.0000    157.0000    164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
UPTAKE Subroutine
   131313
   0
BIOPLT Subroutine
   444444
2.0000000E+00
0.0  1.0  1.0
0.0  1.0  1.0
 1.0  0.0
B1
2.0  2.0
i       n
-1.0  -1.0
B1
2.0  2.0
i       n
-2.0  -1.0
B1
2.0  2.0
i       n
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
-3.0  -1.0
B1
2.0  2.0
i       n
-4.0  -1.0
B1
2.0  2.0
i       n
-5.0  -1.0
B1
2.0  2.0
i       n
-6.0  -1.0
R1
2.0  18.0
i       n
-1.0  -1.0
WND
2.0  23.0
i       n
-1.0  -1.0
SD
2.0  19.0
i       n
-1.0  -1.0
EA
2.0  5.0
i       n
-1.0  -1.0
2.0  0.0
EA
2.0  5.0
i       n
-2.0  -1.0
EA
2.0  5.0
i       n
-3.0  -1.0
EA
2.0  5.0
i       n
-4.0  -1.0
EA
2.0  5.0
i       n
-5.0  -1.0
EA
2.0  5.0
i       n
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
-6.0  -1.0
B1
2.0  2.0
i       n
-7.0  -1.0
R1
2.0  18.0
i       n
-1.0  -1.0
WND
2.0  23.0
i       n
-1.0  -1.0
SD
2.0  19.0
i       n
-1.0  -1.0
EA
2.0  5.0
i       n
-1.0  -1.0
3.0
4.0
TRUPLT Subroutine
171717
4.0
1.0 1.0 1.0 1.0
13.0
1.0  7.0  8.0  9.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
TPLOTR Subroutine
191919
4  4  17
1 1 1 1
y
y
n
n
days
g/m2
Live Corn Biomass
above ground
root
leaf
fruit
-111.0  -111.0
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
-111.0  -111.0
TRUPLT Subroutine
171717
1.0
1.0
13.0
10.0
0.0
0.0
0.0
TPLOTR Subroutine
191919
4  6  24
1
y
y
n
n
days
mm/day
Evapotranspiration, Corn
ET
-111.0  -111.0
-111.0  -111.0
TRUPLT Subroutine
171717
5.0
1.0 1.0 1.0 1.0 1.0
13.0
2.0 3.0 4.0 5.0 6.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
TPLOTR Subroutine
191919
4  4  29
1 1 1 1 1
y
y
n
n
days
g/m2
Amarillo Above-Ground Biomass
garden veg
ws grass
alfalfa
wheat
```

Table 16.19.

Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration (continued)

```
cs grass
-111.0 -111.0
-111.0 -111.0
IRUPLI Subroutine
171717
5.0
2.0 2.0 2.0 2.0 2.0
13.0
1.0 2.0 3.0 4.0 5.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
IPLOIR Subroutine
191919
4  6  11
1 1 1 1 1
y
y
n
n
days
mm/day
Amarillo ET
garden veg
ws grass
alfalfa
wheat
cs grass
-111.0 -111.0
-111.0 -111.0
```

Table 16.20.

Examples 3 and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants Undergoing Succession and Niche Competition

```
MAIN Subroutine
100.000
1.00000
1.00000
0.000000E+00   0.000000E+00   0.000000E+00   1.00000
6.00000
0.000000E+00
1.00000
1.00000
0.000000E+00
250.000      1.00000
CLIMAT Subroutine
555555.
270.000      723.000
90.0000
29.9000      70.4000
120.000
2.00000
0.200000
6990.00
5.400000E-03
4.00000      4.00000      4.00000      4.00000      4.00000      3.00000      8.00000
9.00000      4.00000      4.00000      3.00000      4.00000
4.600000E-02   3.800000E-02   6.200000E-02   4.600000E-02   6.400000E-02      5.900000E-02
0.178000         0.216000         9.400000E-02   8.500000E-02   5.300000E-02   5.800000E-02
18.0000
6800.00
0.000000E+00
138.000
37.0000
WATLX Subroutine, Part I
141414.
0.000000E+00   0.000000E+00
0.000000E+00
0.000000E+00
0.000000E+00
0.100000
WATLX Subroutine, Part II
151515.
92.0000      7.00000      7.00000      15.0000      15.0000      19.0000
238.000      14.0000      14.0000      31.0000      31.0000      39.1000
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
2.00000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
```

Table 16.20.

Examples 3 and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants (continued)

```
 3.00000    -1.00000    -1.00000    -1.00000    -1.00000    -1.00000
 500.000     500.000     500.000     500.000     500.000     500.000
 20.0000     300.000     280.000
0.100000    0.100000    0.100000
0.300000    0.300000    0.300000
0.400000    0.400000    0.400000
0.200000    0.200000    0.200000
 27.0000     27.0000     27.0000
 36.0000     36.0000     36.0000
 34.0000     34.0000     34.0000
 1.00000     1.00000     1.00000
 1.00000     1.00000     1.00000
 1.00000     1.00000     1.00000
 1.00000     1.00000     1.00000
 1.00000     1.00000     1.00000
 1.00000     1.00000     1.00000
 PITRO Subroutine
 121212.
 1.00000  (enter 0.0) to by-pass succession for example 4)
 1.00000  (enter 0.0 to by-pass niche competition for example 4)
0.000000
 7.00000     19.0000     8.00000     18.0000     20.0000     5.00000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 1.00000     1.00000     1.00000     1.00000     1.00000     2.00000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 1.00000     1.00000     2.00000     2.00000     0.00000     0.000000E+00
 1000.00     1000.00     1000.00     1000.00     1000.00     1000.00
 UPTAKE Subroutine
 131313.
    0
 FORMAN Subroutine
 777777.
0.000000E+00
 2.00000
0.190000
0.540000
 1.50000
0.000000E+00
 BIOPLT Subroutine
 444444.
 1.00000
0.000000E+00   1.00000     1.00000
 1.00000     1.000000E+00
BSMAX
 2.00000     2.00000
i     n
-1.00000    0.000000E+00
BSMAX
```

Table 16.20.

Examples 3 and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants (continued)

```
     2.00000       2.00000
i       n
    -2.00000      0.000000E+00
BSMAX
     2.00000       2.00000
i       n
    -3.00000      0.000000E+00
BSMAX
     2.00000       2.00000
i       n
    -4.00000      0.000000E+00
BSMAX
     2.00000       2.00000
i       n
    -5.00000      0.000000E+00
BSMAX
     2.00000       2.00000
i       n
    -6.00000      0.000000E+00
TPGNW
     2.00000       20.00000
i       n
    -6.00000      0.000000E+00
WNDMAX
     2.00000       23.00000
i       n
    -6.00000      0.000000E+00
SFAC
     2.00000       19.00000
i       n
    -6.00000      0.000000E+00
SFAC
     2.00000       19.0000
i       n
    -5.00000      0.000000E+00
     1.00000
     1.00000
TRUPLT Subroutine
     171717.
     6.00000
     1.00000       1.00000       1.00000       1.00000       1.00000       1.00000
     13.0000
     1.00000       2.00000       3.00000       4.00000       5.00000       6.00000
    0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
    0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
    0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
TPLOTR Subroutine
     191919.
        4        4        25
```

Table 16.20.

Examples 3 and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants (continued)

```
0       0       0       0       0       0
y
y
y
n
year
g/m2
Maximum Biomass - PJ Site
ann grass W
ann grass C
per grass W
per grass C
evergreen C
pinyon-jun
   -111.000     -111.000
   -111.000     -111.000
```

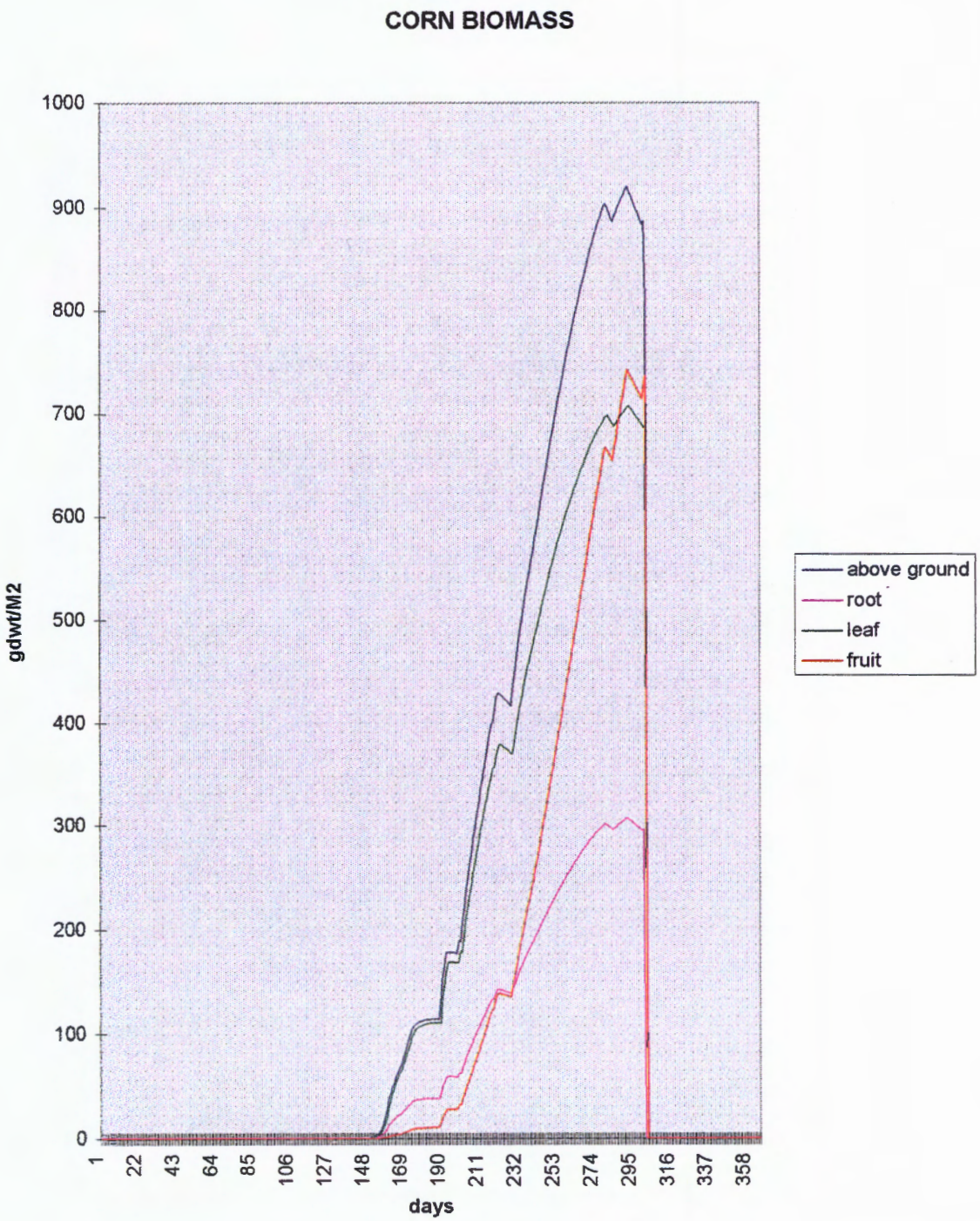


Figure 16.18.

Irrigated Corn Biomass For Various Plant Components for the Amarillo, Texas Scenario

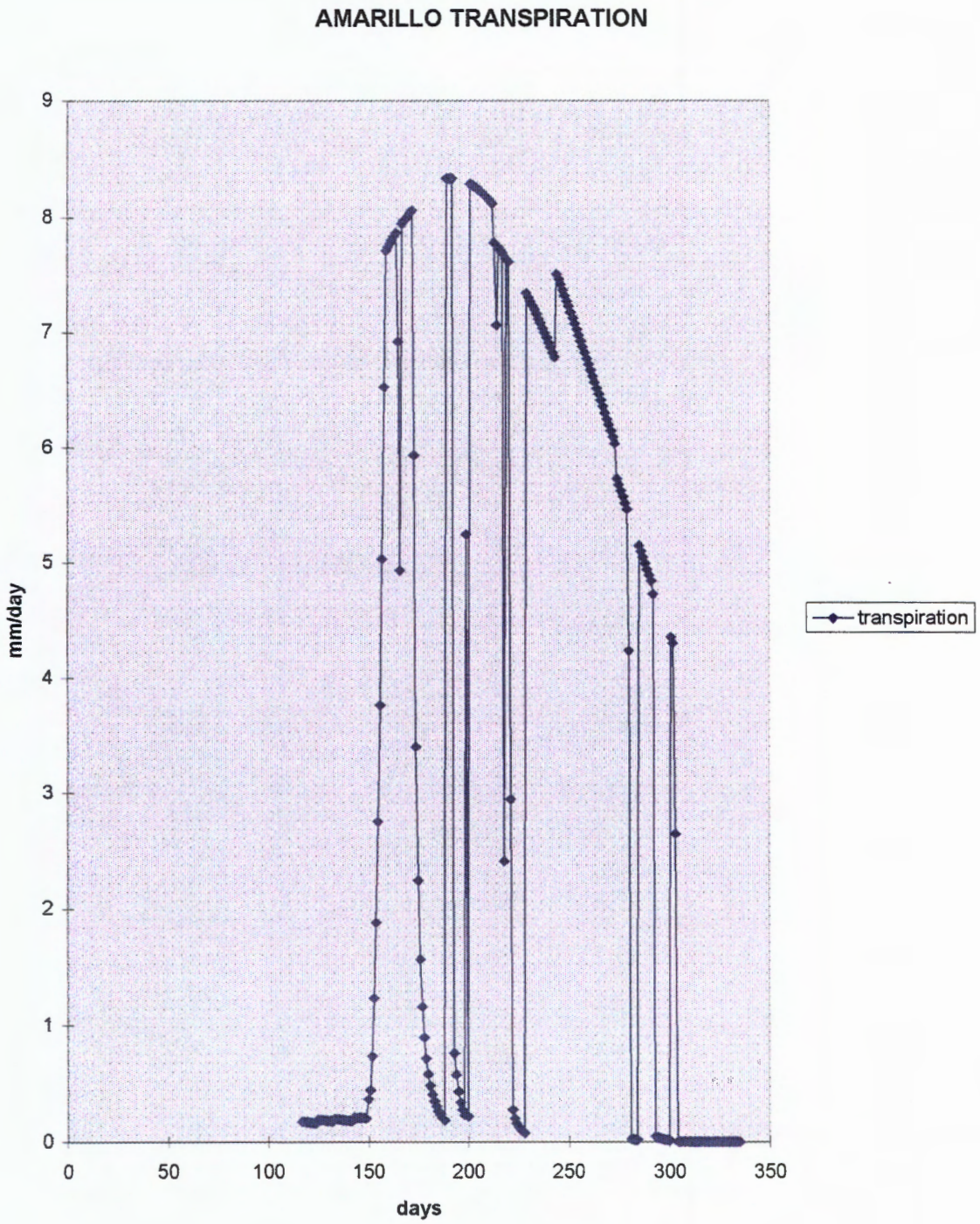


Figure 16.19.

Irrigated Corn Transpiration For Amarillo, Texas Scenario

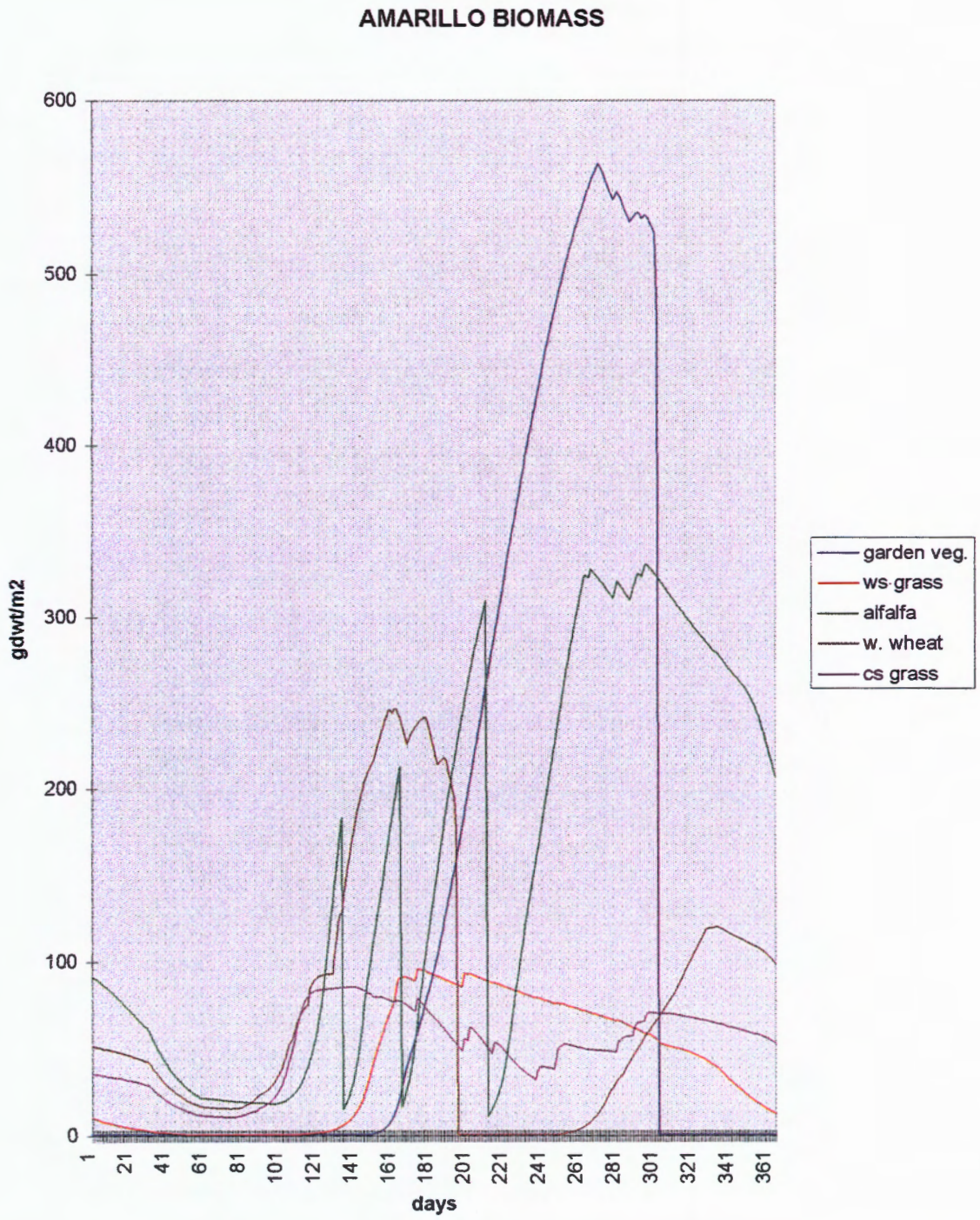


Figure 16.20.

Live Above Ground Biomass for Amarillo, Texas Scenario for Garden Vegetables, Alfalfa, Warm and Cool Season Grasses, and Winter Wheat. All But the Grasses Have Specific Irrigation and Cropping Schedules. Alfalfa and Winter Wheat Span More than One Crowing Season. The Grasses are Niche Competitors.

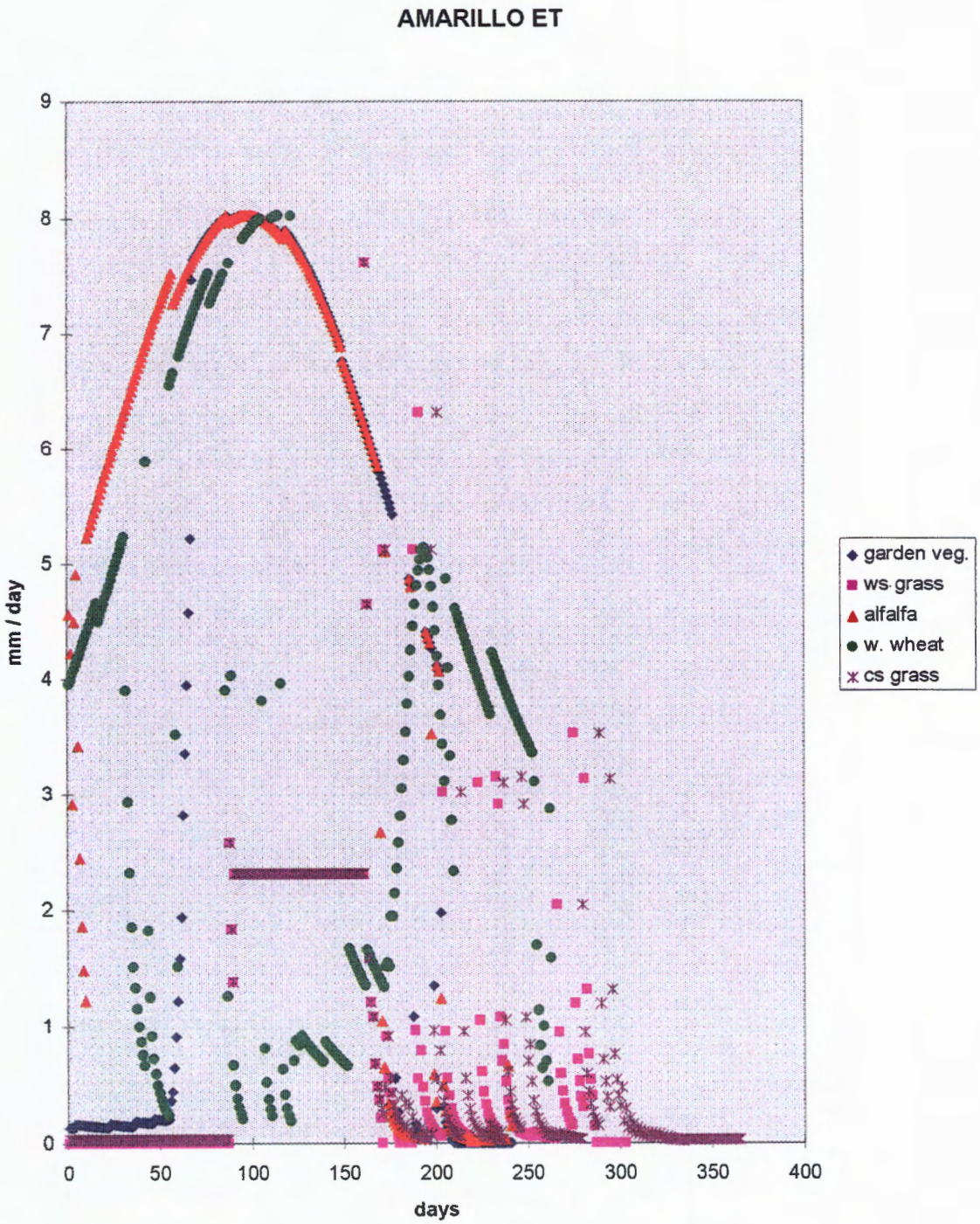


Figure 16.21.

Transpiration For Grasses and Crops For the Amarillo, Texas Scenario. Compare These Curves With Those in Figure 16.19. for Irrigated Corn.

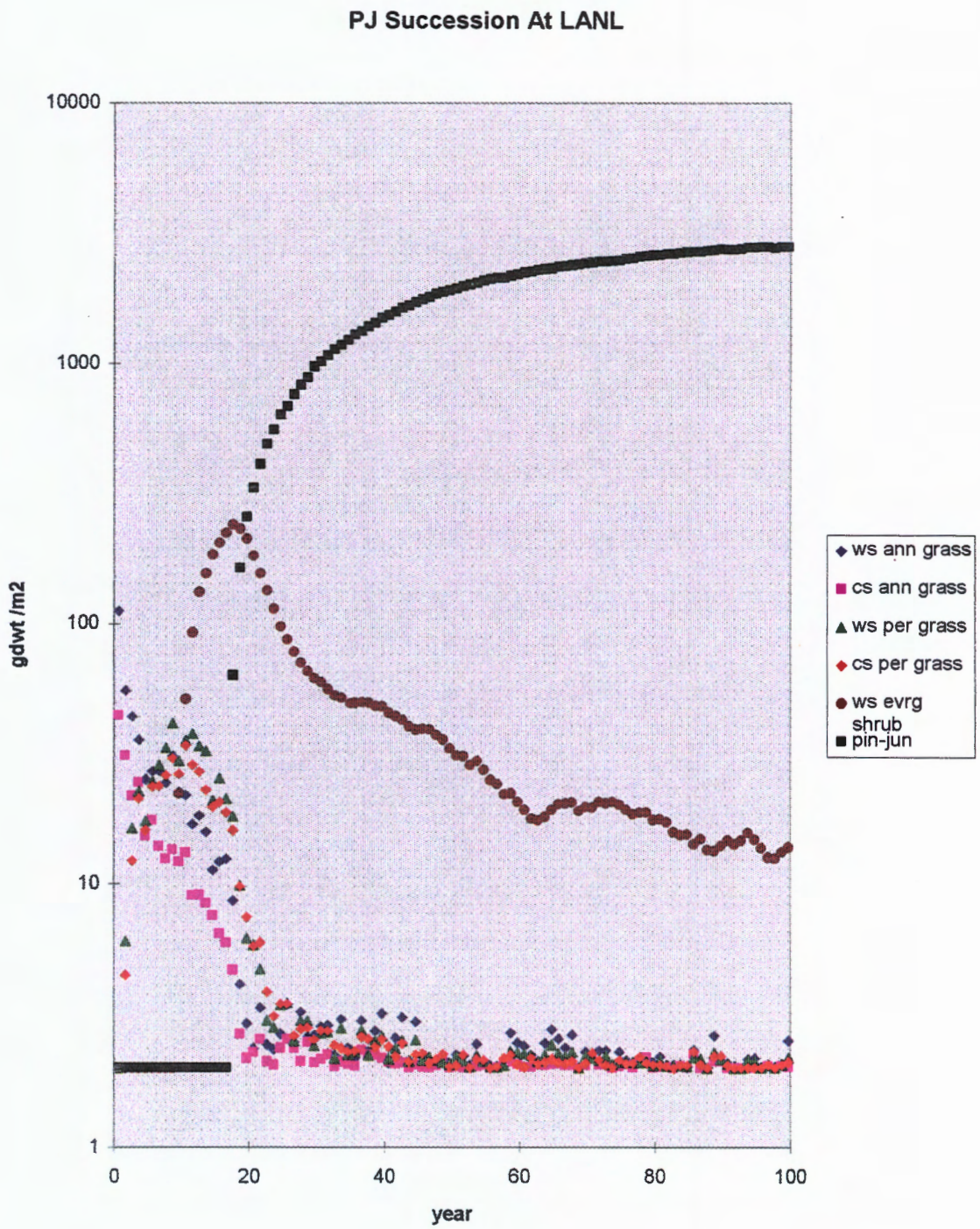


Figure 16.22.

Secondary Plant Succession in Pinyon-Juniper Woodland. Note the Accelerated Growth After Complete Dominance of Trees.

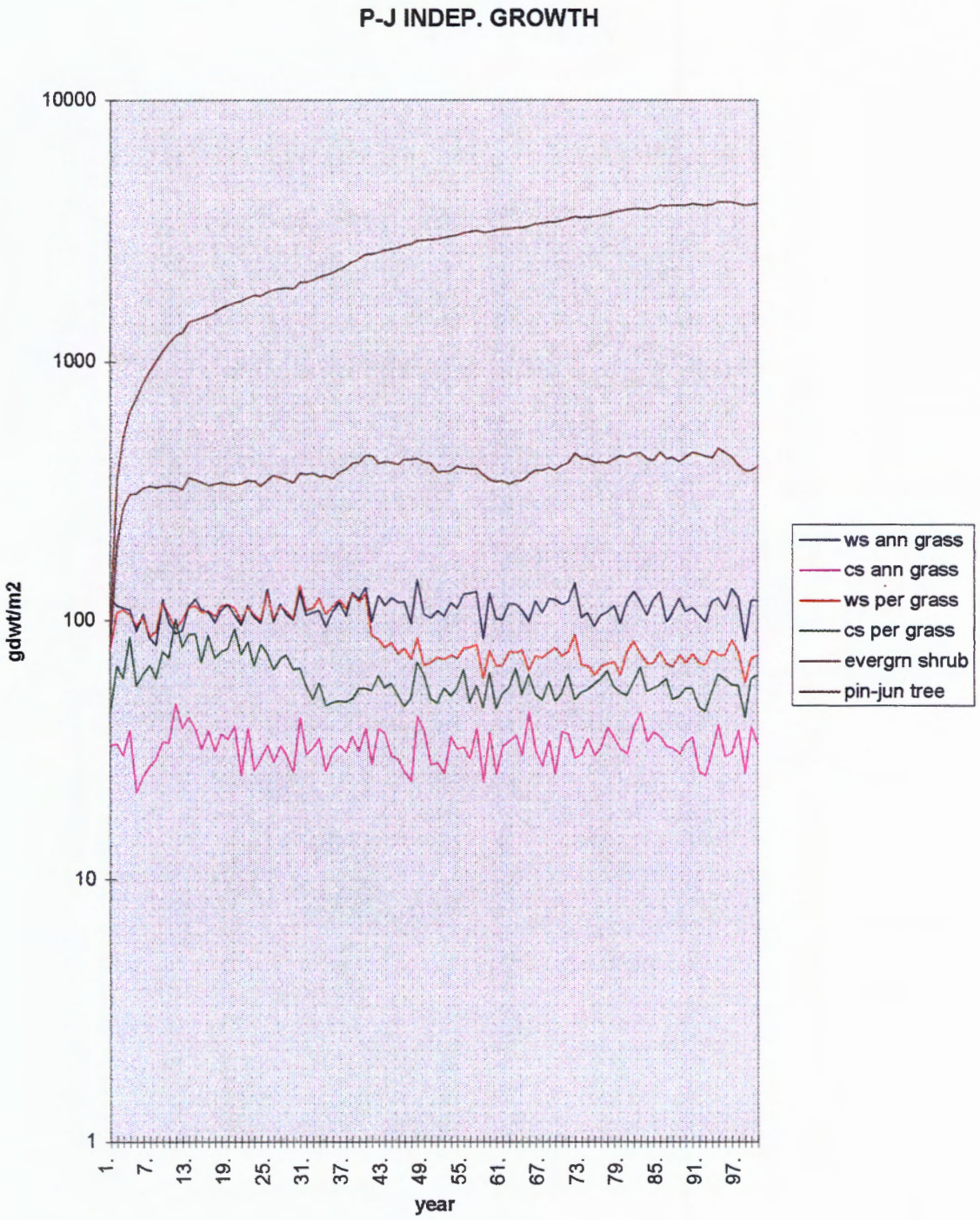


Figure 16.23.

Independent Growth of Plants in Pinyon-Juniper woodland. Succession and Niche Competition Strategies Have Not Been Applied To Any Plant Type or Grouping.

16.10.3. References

Gallegos, A.F., Garcia, B.J., and C.M. Sutton. 1980. Documentation of TRU Biological Transport Model (BIOTRAN). Los Alamos National Laboratory , Los Alamos, New Mexico. LA-8213-MS.

Sutcliffe, J.F., and D.A. Baker. 1974. Plants and mineral salts studies in biology. No. 48. Edward Arnold , Publisher, England. 60pp.

Wenzel, W.J., and A.F. Gallegos. 1982. Supplementary documentation for an environmental impact statement regarding the Pantex Plant, long term radiological risk assessment for postulated accidents. Los Alamos National Laboratory, Los Alamos, New Mexico. LA-9445-PNTX-O.

16.11. UPTAKE Subroutine, Contaminant Uptake and Transport

16.11.1. Description

Plant and soil profile contaminant transport processes modeled in the UPTAKE subroutine are presented in Figure 16.24.. The air above the soil surface is modeled to have a 30 meter mixing layer for resuspension processes, a one meter layer for saltation-creep and rainsplash process, and a 2 meter layer for volatile contaminants. UPTAKE estimates the daily air concentration of a contaminant by adding AIRAC chronic and/or acute releases to concentrations resulting from resuspension (Smith 1982), Saltation Creep (Gallegos 1978), rainsplash (Dreicer 1984), and volatile releases (EPA 1988). A mean wind velocity of 7 mph (US yearly average) is used for all simulated depositional and resuspension events. Resuspended air particulates are modeled to impact and adhere to plant and soil surfaces under both wet and dry atmospheric conditions from daily estimates of the contaminant concentration in the 30 meter mixing layer. Contaminant inventories for plants are compartmentalized into interior volumes and exterior surfaces. Root uptake of contaminants (Romney 1970 and Pimpl 1981) is modeled as a function of soil hydrology, root uptake and translocation, and root mortality. The resuspension layer (defined as the upper 20 mm of soil) receives contaminant in particulate form from plant washoff, and wet and dry deposition processes. Losses of contaminant from soil surface from resuspension processes are modeled as a daily sink and the non-deposited fraction is lost from the system. The same is true for volatile contaminants except that all except that inhaled by animals and humans is lost from the system. However, saltation-creep and rainsplash events are mass balanced. Because plants are compartmentalized into interior volumes and exterior surfaces, each physical process can be tracked separately and verified for different climatic regimes. The latter is of particular importance when comparing, for example, southwestern and southeastern U.S. terrestrial ecosystems. A large database from different ecosystems is required to verify the relative contributions of rainsplash, saltation-creep, resuspension, volatilization, root uptake, and ground cover by plants to the mean daily concentration of contaminant in the diet of animals and humans. A study of this type was performed at LANL on an abandoned radioactive waste pit site (Wenzel 1987) for non-volatile contaminants.

16.11.2. Input/Output

Only few lines of output are required to execute the UPTAKE subroutine as shown in Table 16.21.. The first line is generally set to zero if radiation dose measurements are not being simulated, otherwise three more lines of input are required to characterize the geometry of exposure, and the type of radiation being simulated.

UPTAKE follows PLTGRO in the execution of BIOTRAN.2. with the latter being updated by the former before returning to MAIN; hence UPTAKE information is utilized by other subroutines. Output from UPTAKE consists of contaminant concentrations in air, internal volumes of plants, surficial parts of plants, roots, soil surfaces , ..etc. **Example 1** is used (Table 16.22.) to predict daily resuspension and saltation-creep/rainsplash air concentrations of uranium as shown in Figure 16.25.., An acute release occurring during the growing season for corn (Julian day 280) is used to show the sensitivity of AIRAC acute/chronic events on contaminant concentrations in air. The effect of the acute air contamination effect is also illustrated for corn plant growth as shown in Figure 16.26.. Note that neither resuspension, nor saltation-creep contaminant air concentrations return to their original values after the event has taken place.

Example 2 has been used (Table 16.23.) to predict root uptake of uranium translocation to leaves by various cultivated plants and range grasses as presented in Figure 16.27.. Considerable variation is observed between different plants, and within the same plant depending on the stage of growth during the growing season for these plants.

Table 16.21.

UPTAKE Subroutine Input

Input Line	Variable Name	Definition
1.		***Enter 131313 to access UPTAKE***
2.	ANPPTS	Enter plant the number of plant parts to be included in radioactive contaminant dose measurements. Up to 6 plant parts allowed: 1=seeds, 2=leaves, 3=branches, 4=boles, 5=root stems, and 6=root boles. A value of 0.0 indicates that dose measurements are not simulated.
	End of input if ANPPTS	***set to 0.0, or if contaminant is not radioactive***
3.	EVSP	Enter plant soft tissue dose factor, MeV-rad/ dis for the contaminant.
	EGEOP	Enter exposure geometry (0.5 or 1.0)
	EXCP	Exposure fraction. This value is set by the user to indicate the fraction of the dose absorbed by the given tissue after geometrical considerations have been accounted for. Backscatter should be included for gamma-emitters if relevant.

Repeat for each radionuclide under consideration. If not estimating doses for a given radionuclide, then enter 0.0 for each choice in line 3 above.

Table 16.22.

Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (Corn) Utilized To Generate Uptake Output

```
MASM Subroutine
4.000000
 1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
 1.000000
 0.000000
 1.000000
0.0000000E+00
0.0000000E+00
4.0   1.0
AIRAC Subroutine
111111
1.00000 0.000000 0.000000 0.000000 0.00000
0.000000
0.000000
0.000000
1.000000
1.000000
3.000000
280.0000
1.0e06
0.000000
CLIMAT Subroutine
 555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
WATFLX Subroutine, Part 3
   141414
0.0000000E+00  0.0000000E+00
0.0000000E+00
```

Table 16.22.

Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (continued)

```
0.0000000E+00
 0.0000000E+00
 0.0000000E+00
WATFLX Subroutine, Part II
   151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
1.000000
 2.000000    0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
 5.000000
 990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
8.8000000E-02  0.6200000     0.2000000     6.1000001E-02  2.1000000E-02
PLTGRO Subroutine
   121212
0.0000000E+00
0.0000000E+00
0.0000000E+00
1.0000000E+00
6.0
2.0
1.0
1.0
0.0
4000.
270. 270.
0.0  7.0  6.0  6.0  6.0  6.0
```

Table 16.22.

Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (continued)

```
150. 150. 201. 229. 243. 257.
UPTAKE Subroutine
131313
0
BIOPLT Subroutine
444444
1.0
0.0  1.0  1.0
1.0  1.0
A1
3. 1.
i      m      n
0.0  -1.0  0.0
AS
3. 1.
i      m      n
0.0  -1.0 0.0
WZ42
3. 23.
i      m      n
0.0  -1.0 0.0
WZ43
3. 23.
i      m      n
0.0  -1.0 0.0
AFR
3. 1.
i      m      n
0.0  -1.0 0.0
AFS
3. 1.
i      m      n
0.0  -1.0 0.0
Z1
4. 26.
i      ks     m      n
-1.0  -1.0  -1.0   -1.0
WA
3. 23.
i      ks     n
-1.0  -2.0  -1.0
WA
3. 23.
i      ks     n
-1.0  -3.0  -1.0
Z1
4. 26.
i      ks     m      n
```

Table 16.22.

Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (continued)

```
-1.0  -2.0  -1.0  -1.0
 1.0
 2.0
TRUPLT Subroutine
 171717
 2.0
 1.0 1.0
 13.0
 1.0 2.0
 0.0 0.0
 0.0 0.0
 0.0 0.0
TPLOTR Subroutine
 191919
 4  6  25
      1     1
y
y
y
n
days
pCi/m3
Amarillo Air Concentration
resuspension
salta/creep
  -111.0  -111.0
  -111.0  -111.0
TRUPLT Subroutine
 171717
 4.0
 1.0  1.0 1.0 1.0
 13.0
 3.0 4.0 5.0 6.0
 0.0  0.0 0.0 0.0
 0.0  0.0 0.0 0.0
 0.0  0.0 0.0 0.0
TPLOTR Subroutine
 191919
 4 6 27
 1  1 1 1
y
y
y
n
days
pCi/m2
Plant Surface Contamination
leaf-resusp
```

Table 16.22.

Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (continued)

```
leaf-sal/crp
seed-resus
 -111.0  -111.0
 -111.0  -111.0
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants Utilized To Generate UPTAKE output

Main Subroutine

```
3.000000
 1.000000
 1.000000
0.000000  0.000000  0.000000   1.000000
 7.000000
 0.000000
 1.000000
0.000000
0.000000
1.0  1.0
```

CLIMAT Subroutine

```
 555555
 277.0000      625.0000
 90.00000
 36.20000      78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000      4.000000      4.000000      4.000000
 4.000000      4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1000001E-02  0.1490000
0.1610000      0.1430000      0.1500000      9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.000000
 202.0000
 36.00000
```

WATFLX Subroutine, Part I

```
 141414
0.000000  0.000000
0.000000
0.000000
0.000000
0.000000
```

WATFLX Subroutine, Part II

```
   151515
 92.00000      7.000000      7.000000      15.00000      15.00000
 19.00000
 238.0000      14.00000      14.00000      31.00000      31.00000
 39.10000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
 1.000000      1.000000      1.000000      1.000000      1.000000
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
1.000000    1.000000
 2.000000    0.000000  0.000000  0.000000  0.000000
0.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
 5.000000    5.000000    6.000000    7.000000    6.000000
 6.000000    5.000000
 990.6000    990.6000    1498.600    2006.600    1498.600
 1498.600    990.6000
 20.00000    132.4000    203.2000    304.8000    330.2000
0.000000  0.000000  0.000000  0.000000  0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000    18.70000    18.90000    17.90000    16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 28.50000    28.50000    28.60000    25.80000    26.10000
 2.000000    2.000000    2.000000    2.000000    2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
0.000000  0.000000  0.000000  0.000000  0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000    18.70000    18.90000    17.90000    16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 28.50000    28.50000    28.60000    25.80000    26.10000
 2.000000    2.000000    2.000000    2.000000    2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000
0.000000  0.000000  0.000000  0.000000  0.000000
0.000000
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.4200000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 15.00000
 30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000    508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000   0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000   0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000   0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 15.00000    15.00000
 30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000    25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000    24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
 508.0000
0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.4200000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
0.4590000
 18.70000    18.70000    18.90000    17.90000    16.50000
 15.00000
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
30.00000    30.00000    30.10000    27.20000    27.50000
 25.80000
 28.50000    28.50000    28.60000    25.80000    26.10000
 24.50000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000
 20.00000    132.4000    203.2000    304.8000    330.2000
0.000000 0.000000 0.000000 0.000000 0.000000
0.1800000   0.1800000   0.1200000   0.1200000   0.1200000
0.4900000   0.4900000   0.4100000   0.4020000   0.4030000
0.3300000   0.3300000   0.4700000   0.4780000   0.4780000
 18.70000    18.70000    18.90000    17.90000    16.50000
 30.00000    30.00000    30.10000    27.20000    27.50000
 28.50000    28.50000    28.60000    25.80000    26.10000
 2.000000    2.000000    2.000000    2.000000    2.000000
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
6.3000000E-03
5.0099999E-02 0.5100000   0.3300000   0.1600000   7.9999998E-02
2.4000000E-02 7.8999996E-03
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000   0.2000000   6.1000001E-02 2.1000000E-02
PLTGRO Subroutine
  121212
0.000000
1.000000
0.000000
 1.000000    2.000000    8.000000    11.00000    12.00000
 18.00000    23.00000
 6.000000    17.00000    0.000000  10.00000    5.000000
0.000000  17.00000
 2.000000    2.000000    0.000000    4.000000    2.000000
0.000000  2.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000    1.000000
 1.000000    1.000000    0.000000  1.000000    1.000000
0.000000  1.000000
0.000000 0.000000  1.000000   0.000000 0.000000
 1.000000    0.000000
 4000.000    4000.000    4000.000    4000.000    4000.000
 4000.000    4000.000
 302.0000    302.0000
0.000000  7.000000    6.000000    6.000000    6.000000
 6.000000
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
150.0000    150.0000    201.0000    229.0000    243.0000
 257.0000
 302.0000    302.0000
0.000000  1.000000    1.000000    1.000000    1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 2.000000    2.000000    2.000000    2.000000    2.000000
 2.000000    2.000000
 150.0000    150.0000    157.00001   164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
 1.000000    500.0000    531.0000    577.0000
0.000000  7.000000    5.000000    5.000000    5.000000
 5.000000    5.000000    5.000000    5.000000    5.000000
 215.0000    215.0000    470.0000    485.0000    502.0000
 514.0000    527.0000    533.0000    544.0000    559.0000
 1.000000    560.0000
0.000000  16.00000    5.000000    5.000000    5.000000
 250.0000    250.0000    285.0000    425.0000    500.0000
 302.0000    302.0000
0.000000  7.000000    2.000000    2.000000    2.000000
 2.000000    2.000000    2.000000    2.000000    2.000000
 3.000000    3.000000    3.000000    3.000000    3.000000
 3.000000    3.000000
 150.0000    150.0000    157.0000    164.0000    171.0000
 177.0000    184.0000    191.0000    198.0000    205.0000
 212.0000    219.0000    226.0000    233.0000    240.0000
 247.0000    254.0000
UPTAKE Subroutine
   131313
   0
BIOPLT Subroutine
   444444
1.0000000E+00
0.0  1.0  1.0
 1.0  1.0
WZ7
3.0  23.
i      m      n
-2.0 -1.0 0.0
WZ7
3.0  23.
i      m      n
-3.0 -1.0 0.0
WZ7
3.0  23.
i      m      n
-4.0 -1.0 0.0
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
WZ7
3.0  23.
i      m      n
-5.0 -1.0 0.0
WZ7
3.0  23.
i      m      n
-6.0 -1.0 0.0
WZ7MAX
3.0  23.
i      m      n
-2.0 -1.0 0.0
WZ7MAX
3.0  23.
i      m      n
-3.0 -1.0 0.0
WZ7MAX
3.0  23.
i      m      n
-4.0 -1.0 0.0
WZ7MAX
3.0  23.
i      m      n
-5.0 -1.0 0.0
WZ7MAX
3.0  23.
i      m      n
-6.0 -1.0 0.0
3.0
1.0
IRUPLT Subroutine
171717
5.0
1.0 1.0 1.0 1.0 1.0
13.0
1.0 2.0 3.0 4.0 5.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
IPLOTR Subroutine
191919
4  5  23
1 1 1 1 1
y
y
y
n
years
```

Table 16.23.

Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants (continued)

```
pCi/gdwt
AMARILLO PLANT U UPTAKE
garden veg.
ws per grass
alfalfa
w wheat
cs grass
-111.0  -111.0
-111.0  -111.0
```

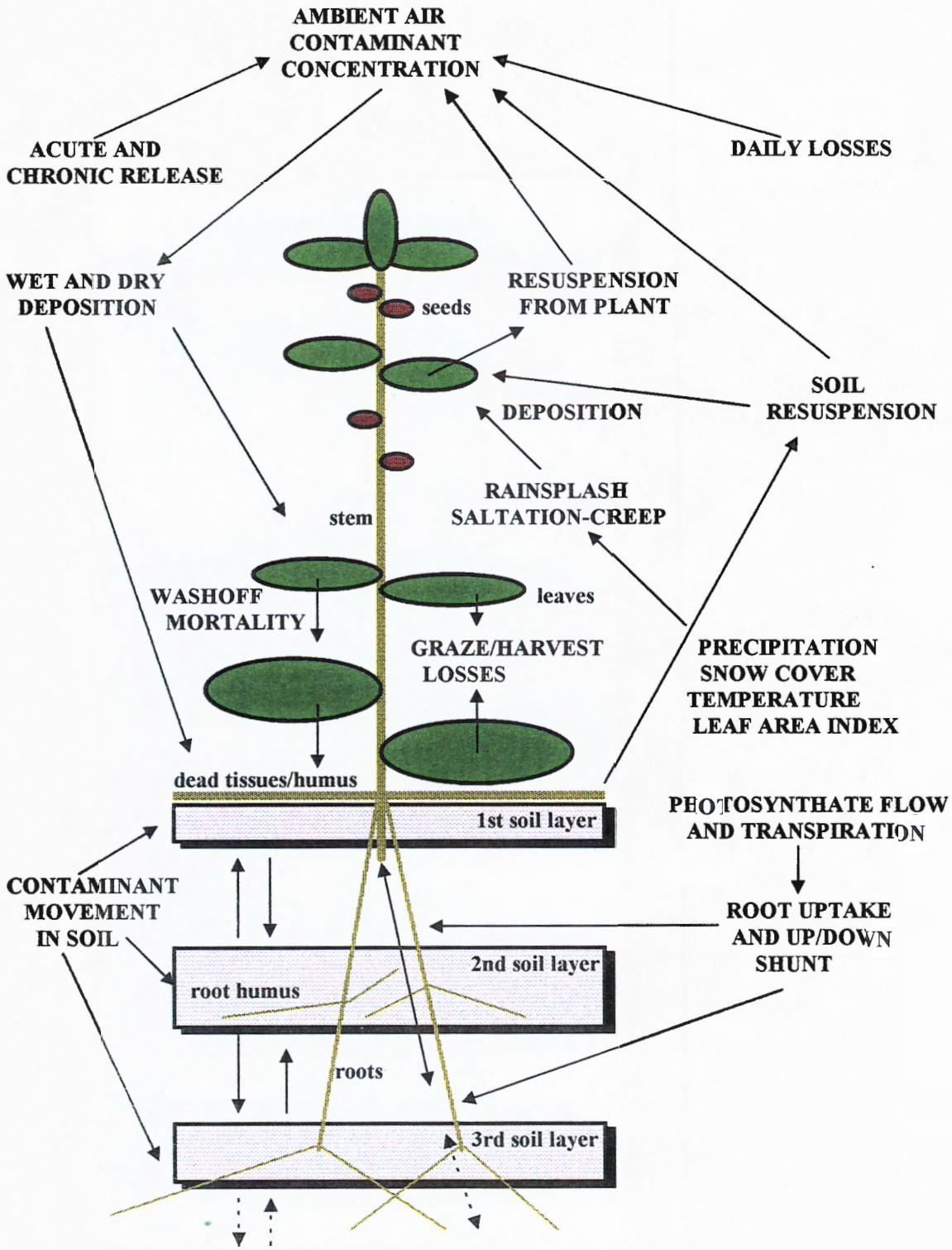


Figure 16.24.

Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine

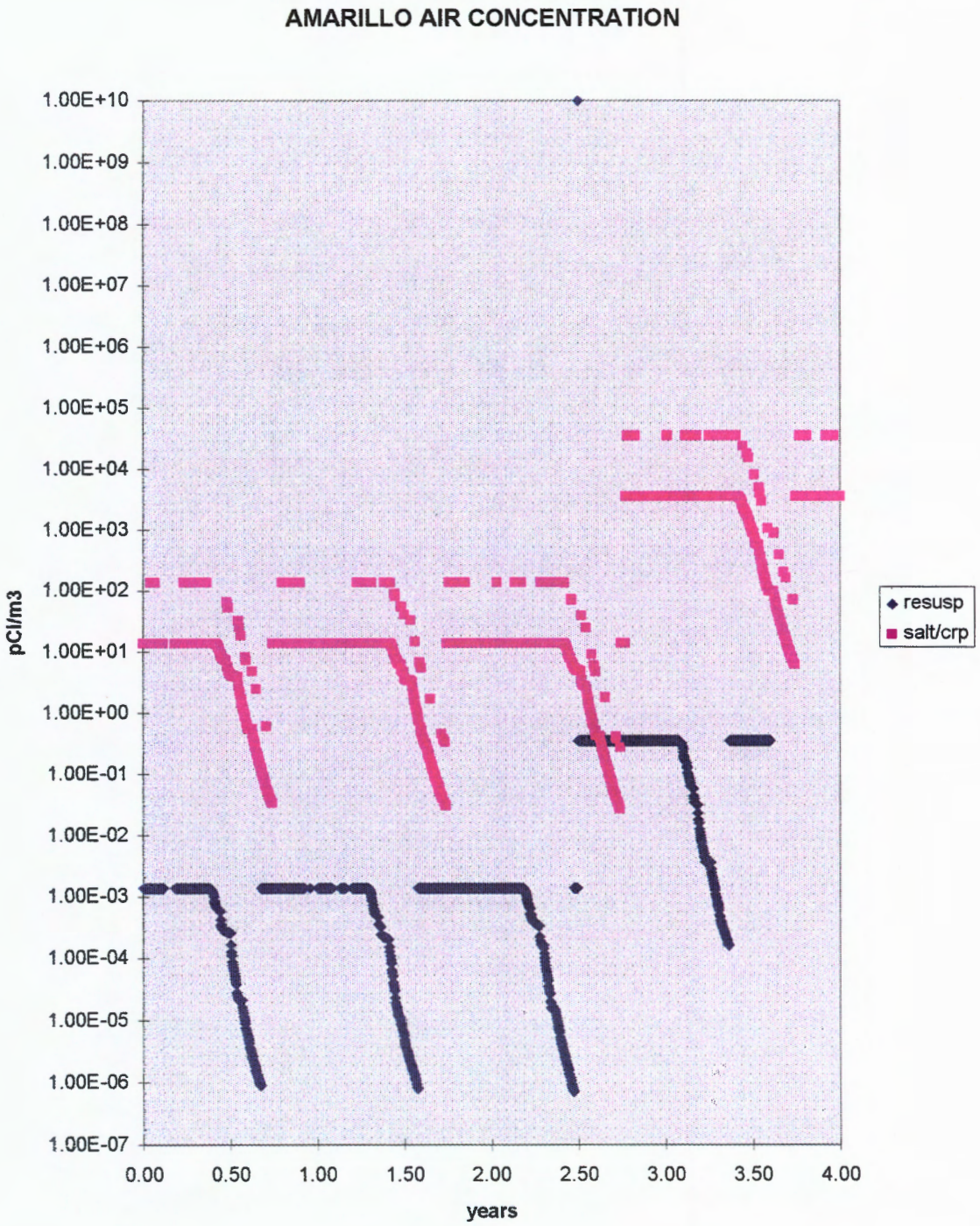


Figure 16.25.

Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine Note the Large Acute Contaminant event on Julian day 280 and its effect on ambient concentrations of Uranium.

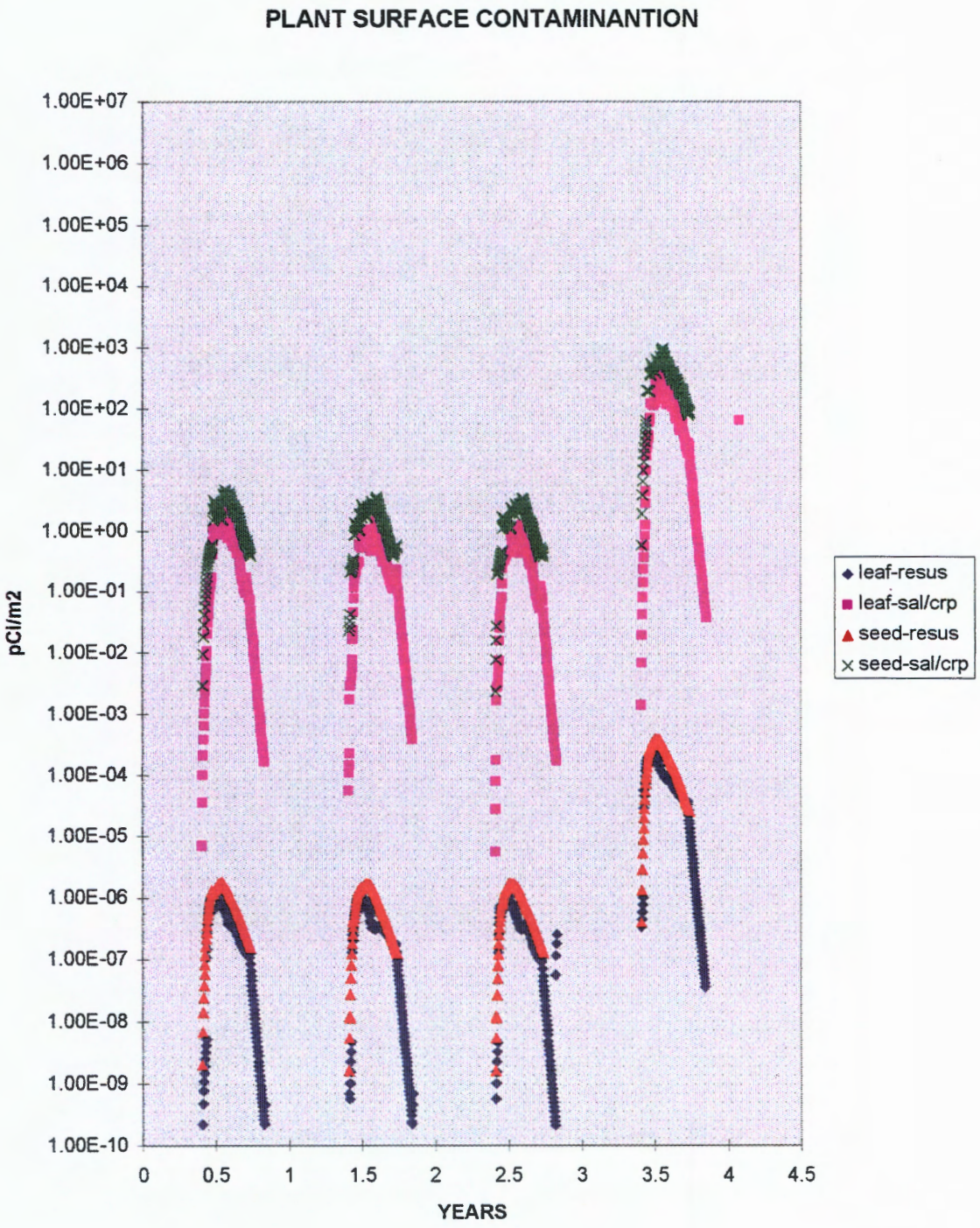


Figure 16.26.

Daily Surface Particle Concentrations of Uranium On Irrigated Corn Leaves and seeds for Amarillo, Texas. Note Effect of Uranium Spike During the 3rd Year

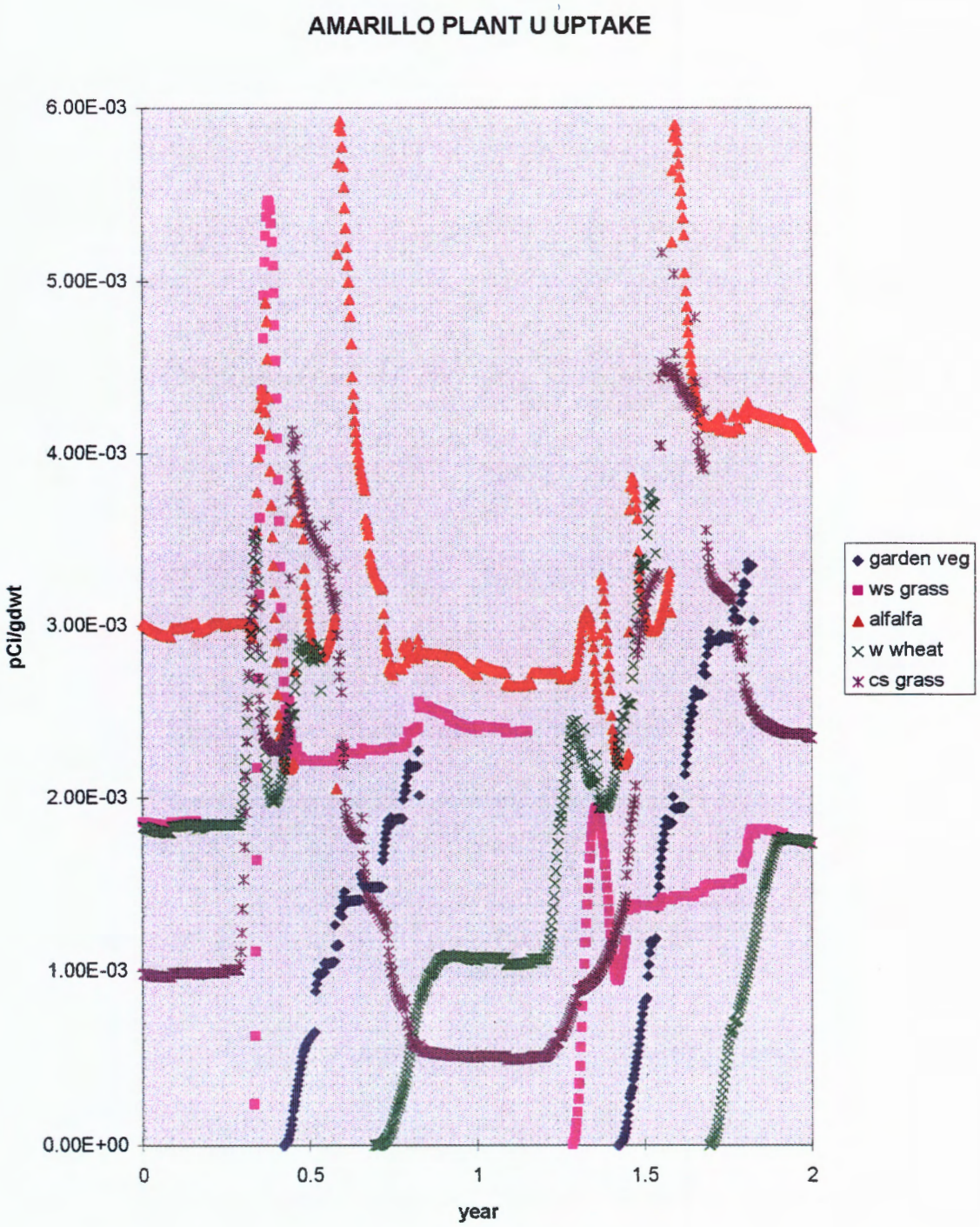


Figure 16.27.

Internal Leaf Concentrations for Uranium for Crops and Range Grasses At Amarillo, Texas

16.11.3. References

Dreicer, M., Hakonson, T.E., White, G.C., and F.W. Whicker. 1984. Rainsplash as a mechanism for soil contamination of plant surfaces. Health Physics 46: 177-187.

Gallegos, A.F.. 1978. Preliminary model of plutonium transport by wind at Trinity Site. In: Selected Environmental Plutonium Research Reports of the NAEG. USDOE. NVO-192. p681-695.

Pimpl, M., and H. Schuttelkopf. 1981. Transport of plutonium from soils into plants by root uptake. Nuclear Safety 22: 214-225.

Romney, E.M., Mork, H.M., and K.H. Larson. 1970. Persistence of plutonium in soil, plants, and small mammals. Health Physics 19: 487-489.

Smith, W.J., Whicker, F.W., and H.R. Meyer. 1982. Review and categorization of siltation, suspension, and resuspension models. Nuclear Safety 23: 685-699.

Wenzel, W.J., Foxx, T.S., Gallegos, A.F., Tierney, G., and J.C. Rodgers. 1987. Cesium-137, plutonium-239/ 240, total uranium, and scandium in trees and shrubs growing in transuranic waste at Area B. Los Alamos National Laboratory, Los Alamos, New Mexico, LA-11126-MS

EPA (U.S. Environmental Protection Agency), 1988. Superfund Exposure Assessment Manual, Environmental Protection Agency, Office of Remedial Response, EPA 1540/1-88/001.

16.12. ANIMAL and INVERT - Mammal and Invertebrate Simulations

16.12.1. Description

In the preceding sections BIOTRAN.2 subroutines have been used to simulate climate, soil transport and hydrology, plant growth, and contaminant transport and uptake in plants. The subroutines ANIMAL and ZOOGRO are used to simulate the growth and metabolism of mammals and invertebrates such as insects with BIOTRAN.2, although the invertebrate model (INVERT) has not been validated using field studies. Prior simulations were focused primarily on simulating ruminant digestion of feed and forage by both beef and dairy cows. Most recently, studies involving deermice, coyotes and deer have been addressed (Kelly 1995). An illustration of ANIMAL operation and its interactions in simulating mammals, particularly beef and dairy cows is presented in Figure 16.28.. A more precise representation of contaminant transport within a mammal via a physiologically based pharmacokinetics model (PB-PK) is presented in Figure 16.29..

Up to 10 different mammals, either wild or domestic, can be simulated simultaneously; the first two types are coded for beef and dairy cattle only if the human simulation model (HUMTRN) is being used, otherwise any order of animal types can be simulated. Also, if HUMTRN is being used, then the last two values of mammals must be for male and female humans, respectively. No restrictions as to order of appearance of specific animal type (including invertebrates) is required in BIOTRAN.2 simulations if HUMTRN is not used.

Simulated air, food, and water volumes, masses, and contaminant concentrations required for proper operation of ANIMAL are supplied from other BIOTRAN.2 subroutines in daily time steps as shown in Figure 16.28. noted earlier. Simulated food supplies must be adequate to ensure normal weight gain and maintenance energy by all age groups in a given animal population. The user should verify that the specified feed and/or forage schedules for domestic animals is adequate for the simulated population or herd. Verification should also include agreement between the onset of intake with organ uptake and retention of a specific contaminant. The lung portion of contaminant transport of the PB-PK model depicted in Figure 16.29. is based on an ICRP lung model, However, the transport of contaminants within the body, and exchanges of volatile contaminants with the ambient air are based on the use of partition coefficients for given contaminants between the blood plasma and the organ of interest, or the ambient air if alveolar exchange of a volatile contaminant is under consideration.

Portions of ANIMAL were validated using field data for uranium in cattle tissues from the proximity of the Pantex Site near Amarillo, Texas (Wenzel A and B 1982). Ruminants may be modeled in age groups according to herd management methods which includes feed lot, dairy herd, or wild ruminants. As noted earlier, different age distributions are used for each animal type simulated. In the first example, both age distributions for beef and dairy cows are simulated by assuming a parabolic age structure, and a maximum life-span of 15 years. Feedlot and dairy herd age distributions usually have a human determined life-span of about 8 years, however, this difference does not affect the transport of contaminants appreciably, although it does increase the amount of food to maintain the population. The user is required to set up realistic grazing and feed consumption schedules for domestic ruminants, and sufficient acreage of both types of feed must be specified , otherwise, a "starvation" warning is issued by BIOTRAN.2, and the simulation is terminated when the starvation weight is reached.

In the second example, an insect population based on the work of Randell 1970 is simulated using INVERT, and includes the growth and metabolism of three metamorphic stages, and the adult phase, together with the simulation of diapausing and non-diapausing egg stages. This simulation is included to illustrate the special features of INVERT.

16.12.2. Input/Output

The input instructions for subroutine ANIMAL are presented in Table 16.24. These instructions serve to set up the type of animal, the population size, age distribution, and forage and/or feed schedules. Although the input requirements for ANIMAL is rather lengthy, it is quite flexible. The user should verify that the diet intervals are read in and simulated as the schedule dictates. The onset of a specific diet becomes important for the scenario in question. Note the number of time that input must be repeated depending on the number of animal types , regions, contaminants, food schedules, ..etc. are used in the simulation. It is important to verify that a source of food is specified for a given animal type throughout the calendrical year, otherwise weight losses will be observed, and cautionary signals may appear on the screen of the PC. Once grazing preferences have been established from user input, then the ZOOGRO subroutine will simulate the grazing of specific plant types in order of preference. ZOOGRO will also manage the supplementary feed harvesting and rationing. The latter is fed to livestock when grazing becomes limited during certain portions of the year. Hence, when supplementary feed becomes limiting, then weight losses will occur rather rapidly as well. The user should plot intakes of feeds and follow weight gains to verify the diet dynamics.

Input for **Example 1** is presented in Table 16.25. Two ruminants, beef and dairy cattle, are simulated. Beef cattle have three different diets specified throughout the year, whereas, dairy cows have a constant diet. The beef cows graze on winter wheat for the first 73 days of the year, graze on open range on Julian days 74-224, and then a fattening regime on supplementary feed for the remainder of the year. Dairy cows are fed a constant diet of wheat grain and alfalfa throughout the year. Note that Example 1 is an extension of **Example 2** from the previous section (Table 16.23.). Sufficient grazing acreage (4000 acres) is specified for 99 beef cattle. There must also be enough harvested feeds for both beef and dairy cattle. Figures 16.30. and 16.31. show uranium intake for the two ruminants from lung deposition and transport to the gut, from accidental soil ingestion during grazing, and from that deposited externally and internally in the food from both range and harvested crops. Notice that accidental soil ingestion by dairy cattle is not present because these animals are maintained exclusively on harvested crops for this scenario. Figures 16.32. and 16.33. show uranium concentrations in various organs and tissues of 2 year-old beef and dairy cows, respectively. Note the sharp increase and consequent decline of uranium concentrations from the acute air contamination event involving 1 um AMAD particulates on Julian day 195 of the 3rd year of the simulation. Also note the sharp decline of lung burdens through time as clearance mechanism remove the particulates and transport them to the gut, or from absorption into the bloodstream as is the case with the pulmonary region of the lungs. Finally, it should also be mentioned that enough time (2 years in this scenario) must be allowed for harvested crops to accumulate before the cattle growth is initiated as is required for BIOTRAN.2 simulations. For both beef and dairy cows the drinking water uranium concentration was set a constant value of 6.91 pCi/liter.

Input for **Example 2** is presented in Table 16.26.; this example is also a modification of Example 2 from the previous section (Table 16.23.). Grasshoppers are simulated exclusively on range on a diet of warm and cool season perennial grasses and herbs. All stages of incomplete metamorphosis are modeled including diapause, and non-diapause eggs, young nymphs, old nymphs, young adults, and mature male and female adult grasshoppers. The latter are simulated to lay eggs to complete the cycle. A plot of adult grasshopper biomass densities as a function of time are presented in Figure 16.34. However, invertebrate simulations of this type have not been field verified, although the biomass densities are in reasonable agreement with published data (Randell 1970). Further research is required into the modeling aspects is also required to incorporate insects with complete metamorphosis (holometabolous), and possibly for invertebrates which have life-spans greater than one year. Finally, it was not possible to conveniently plot egg numbers (100 age groups), nymph stages (30 age groups), or young adults (30 age groups) to give a better understanding of the growth strategies employed in BIOTRAN.2 for these organisms, and it is left to the user to verify the metamorphic strategy.

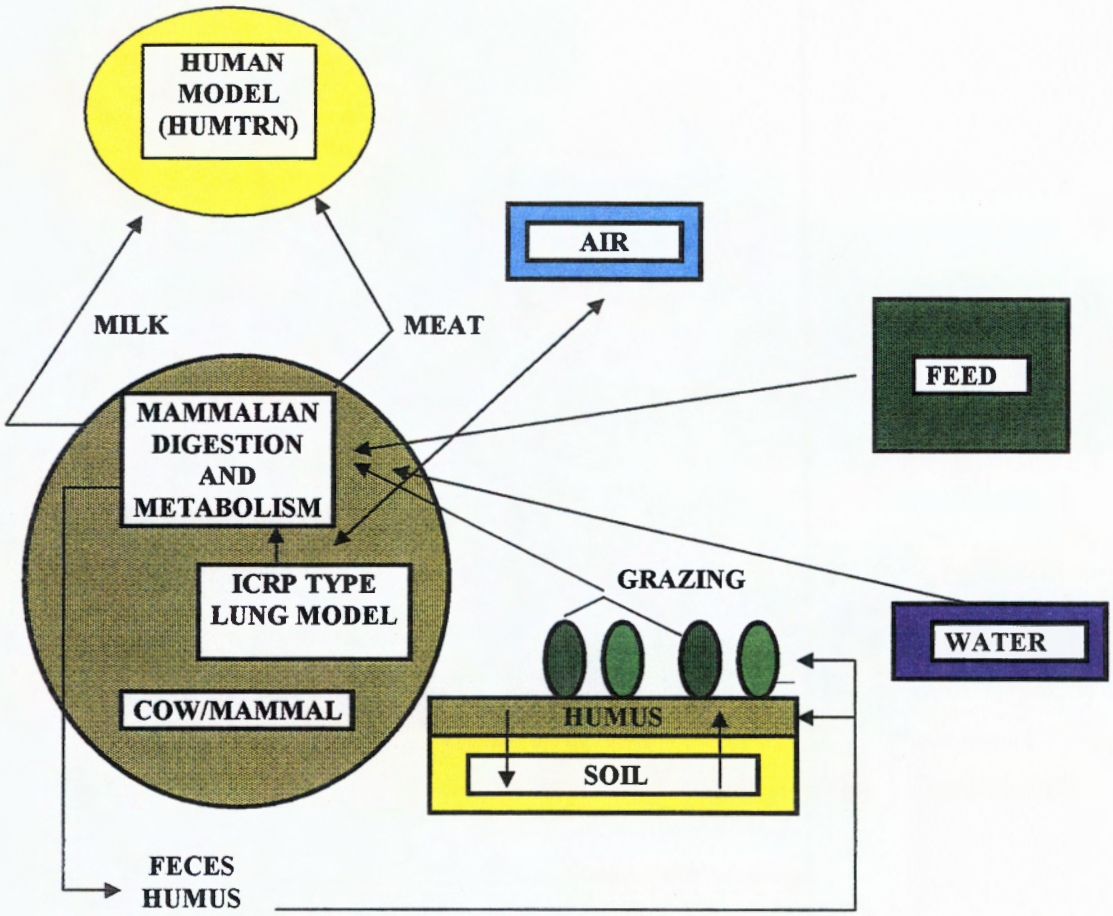
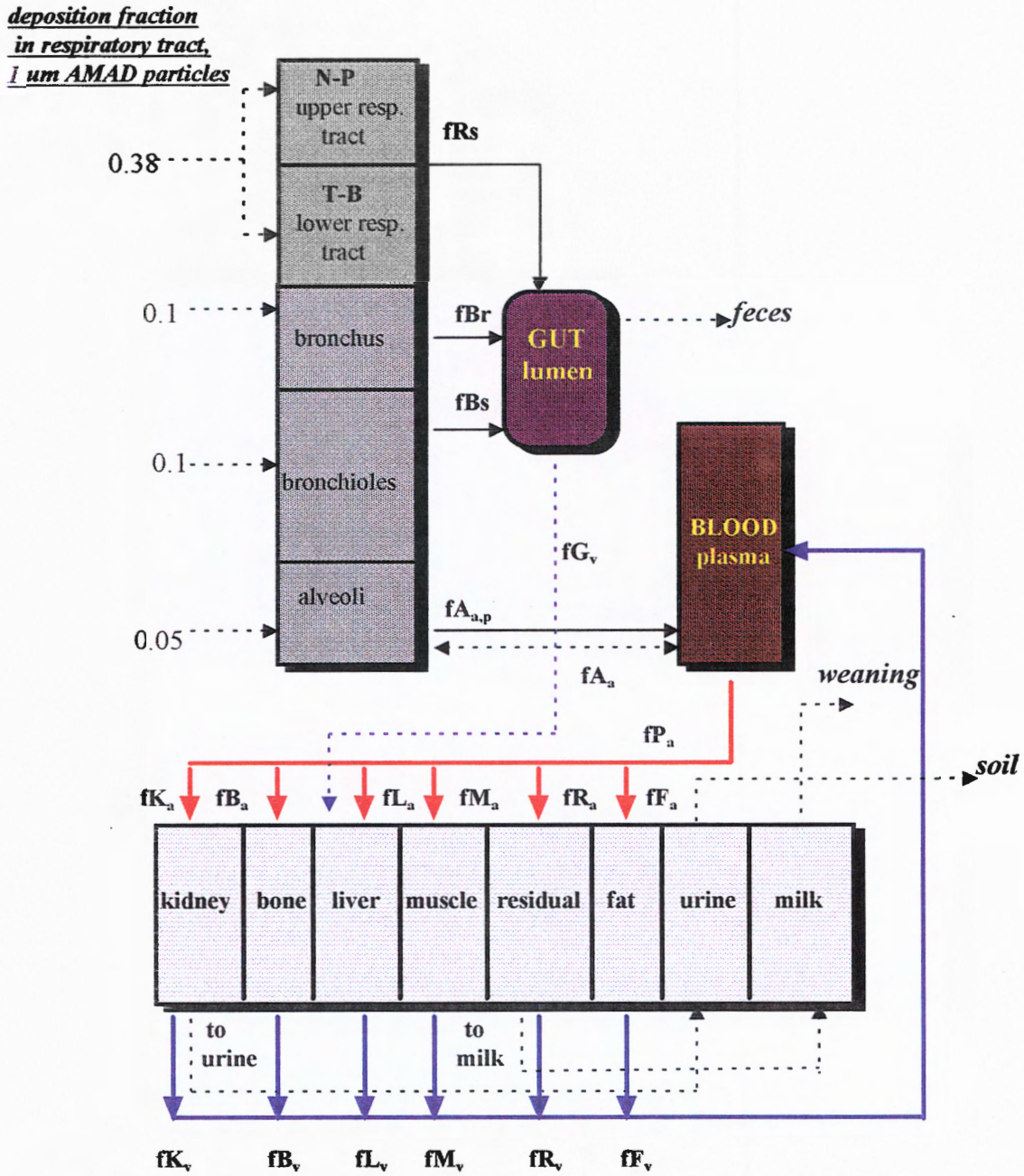


Figure 16.28.

Illustration of ANIMAL/ZOOGRO Operations With the Human Model HUMTRN. The Invertebrate Model INVERT Also Interacts in A Similar Manner



fRs, fBr, fBs = contaminant transferred from lung regions

fA$_{a,p}$, fA$_{a,g}$ = fraction of contaminant transported/exchanged into arterial plasma from the pulmonary region of the lung surface from particulates and vapors

fP$_a$, fP$_v$, fPk$_a$, fPk$_v$,..etc. = arterial and venous plasma flow, arterial and venous flow from kidneys, muscle, liver, bone...etc.

Figure 16.29.

Contaminant Metabolism in a Mammal Based on Physiologically Based Pharmacokinetics Modeling in BIOTRAN.2

Table 16.24.

ANIMAL Subroutine Input

Input Line	Variable Name	Definition
1.		*Enter 222222 to access ANIMAL*
2.	NAN	Enter the number of animal types to be simulated (including invertebrates
3.	RR	Enter mature weight, kgfwt, of each animal type simulated (set invertebrates to 1.0). If entering non-default partition coefficients, enter the negative of the mature weights of the animal types. If humans are to be simulated using HUMTRN, then beef and/or dairy cows must be simulated as animal types 1 and/or 2, respectively.
4.	WATCNR	Enter the concentation in drinking water , ppm, pCi...uCi..etc., for the first simulated contaminant for NAN animal types

Repeat line 4 four times to include the nutrients (nitrate, ammonium, soluble phosphate, labile phosphate, and potassium in that order). Enter these concentrations in ppm for NAN animal types. If more than one contaminant is being simulated simultaneously, then repeat line 4 for each additional contaminant in concentrations stated for the first contaminant for NAN animal types.

Repeat all lines of input noted for line 4 above for each region simulated.

5.	RZOON	Enter number of feeding schedules for NAN animal types. Wildlife are not generally specified for wildlife (enter 1.0 if such is the case).
6.	ALSP	Enter the mean life-span, years, of NAN animal types. For invertebrates such as insects enter (0.0). The latter value is used also as a gate for accessing invertebrate growth and metabolism in subroutine INVERT, and for by-passing algorithms that are specific for mammalian growth processes.
7.	EVS	Enter soft-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.
	EVB	Enter bone-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.
	EVE	Enter bone endosteal-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.

Table 16.24., ANIMAL Subroutine Input (continued)

Repeat for line 7 for each contaminant being simulated (ANZ-5) excluding nutrients

8.	YDOS	Enter the number of years over which effects of a given radionuclide will be performed if desired by the user. A value of (0.0) is entered for non-radionuclides, or if the user does not wish to simulate effects of radiation (fatal cancer production) on a population of mammals of a given type.
9.	P5	Enter the fraction of meat consumed from the total area(s) simulated by a given animal type. A value of (1.0) is generally entered unless the animal is not present on the site during portions of its growth cycle before slaughter and consumption. This value applies even if a multi-regional simulation is specified.
10.	ZOO	Enter the total population size (male and female) for a given animal type. For invertebrates such as insects enter (1.0), because their population size is estimated in biomass density, gdwt/m^2.
	SXRT	Enter the fraction of the total population that is composed of females for a given animal type.
	POPT	Enter the population structure desired for the given animal type. 1=none, 2=parabolic, 3=exponential, 4=hyperbolic. A value of (1.0) indicates that the user will enter the population structure below.
11.	AA2	Enter the fraction of the area modeled that is used by a given animal type. This parameter is generally set to (1.0). unless the animal is moved off-range, or another area is used for part of its life-cycle before slaughter and consumption. This value applies even if a multi-regional simulation is specified.
12.	FSUPF	Enter the fraction of supplementary feeds that are derived from the study area for a given animal type. This value is generally set to (1.0), unless the animal is fed supplementary feed that has not been grown in the study area for part of its life cycle before slaughter and consumption. this value applies even if a multi-regional simulation is specified.
13.	EATIN	Enter the number of yearly feeding intervals for a given animal type. The latter specifies the number of times throughout the simulation when a new feeding strategy is to be specified : 1-10, 11-20, as an example for 2 changes in strategy during the simulation..etc.. A value of (1.0) is generally set for wildlife and domestic animals in areas where a rather constant growth strategy is propagated through time.

Table 16.24., ANIMAL Subroutine Input (continued)

14.	FODIN	Enter the number feeding schedules (on a daily basis) specified during any specific yearly interval for a given animal type: 1-74, 75-365, as an example for 2 changes in the diet (different plant types and/or parts) during the simulation. An interval (1-365) is generally set for most wildlife.
15.	AGEA	Enter the maximum expected life-span (in whole years) for a given animal type. For example, an animal type with a specified mean life-span (ALSP) of 3.4 years would be given a maximum expected life-span of 4.0 years..etc.. **A value of (1.0) is required for invertebrates such as insects**. A value of (0.0) defaults to a maximum life span of 8 years which can be set for beef and dairy cattle if desired by the user.
	GST	Enter the initiation time of the gestation period: 1=Spring, 2=Fall, for a given animal type. Enter (0.0) for invertebrates.
	GSTN	Enter the number of gestation periods/year (limited to a maximum of 2). Enter (0.0) for invertebrates.
	FRPR	Enter the fraction of the off-spring produced during the first gestation period (may equal 1.0 only if one gestation period is specified). Enter (0.0) for invertebrates.
16.	YRFED	Enter the yearly feeding intervals that have different feed schedules for a given animal type: 1,10, 11,20, 21,50, as an example where EATIN=3..etc.., specified by the user over a 50 year simulation period. Enter 2 values/interval.
17.	ONFOD	Enter daily feeding interval that have different diets during a given yearly interval(s) for a given animal type: 1,74, 75,230, 76,365, as an example where EATIN=1, and FODIN=3..etc.., as specified by the user for a given simulation period. Enter 2 values/interval.
18.	RNFRC	Enter the regional utilization fraction in duplicate for a given animal type: 1,1, 0.5,0.5, as an example for a given region where FODIN=1. A duplicate value is used as a delimiter in the BIOTRAN.2 code.
19.	AUTZD	Enter the fraction of the diet of a given animal type which is of a specific prey chosen from NAN animal types. If the animal is not carnivorous, enter (0.0) for all prey types. Be careful to place a (0.0) next to the prey selection which is the animal type in question to avoid self-consumption or cannibalism.
20.	EATREC	Enter the duplicate feeding intervals for all yearly intervals specified: 3,3, 1,1, as an example where there are 2 yearly intervals (EATIN=2)..etc.. The latter would specify 3 food

Table 16.24., ANIMAL Subroutine Input (continued)

		schedules during the first yearly interval, and 1 food schedule for the second yearly interval of changed food schedules.
21.	PLTIND	Enter the number of changed feed schedules (RZOON) for a given animal type: 1,2,3, as an example where RZOON=3 for a given animal type. The latter is used in the BIOTRAN.2 code set up the specific plant types used in the diet of a given animal type throughout the simulation period.
22.	AISCH	Enter the number of plant type used for a specific food schedule of a given animal type of the number specified in RZOON above.
23.	PLTUSD	Enter the AISCH plant types for the specific food schedule in question for the given animal type: 8.0, 18.0, as an example where perennial cool and warm season grasses and herbs are specified for this yearly interval where AISCH=2.
24.	PLTRNK	Enter feeding preference (1,2,3) for grazed plants, or food parts (for supplementary feeds) for the plants identified in line 23. Food parts are identified as 4=all plant parts excluding roots, 5=seeds and grains, 6=green and dry forage, 7=temporarily assigned as seeds and grains.
25.	PLTF	Enter the utilization fraction of each plant and/or plant part in the diet of the specific animal type in question. The sum of the fractions should equal 1.0 even on multi-region basis. Also, if animal tissues are consumed, then the sum of the utilization fractions including animal tissues should not exceed 1.0

Repeat Lines 22-25 for each feeding schedule specified by RZOON

Repeat line 18-25 except for line20 for each of APX regions simulated.

If the user has not entered the negative of the mature weight of a given animal type (RR≥0.0), then skip lines 26-28.

26.	PLCM1	Enter deposition fraction for 1um AMAD particles in the naso-pharangeal (N-P) region of the lung for a given animal type.
27.	PLCM2	Enter deposition fraction for 1um AMAD particles in the tracheo-bronchiole (T-B) region of the lung for a given animal type.
28.	PLCM3	Enter deposition fraction for 1 um AMAD particles in the pulmonary region (P) of the lung for a given animal type.
29.	GTBL	Enter the gut-to-blood absorption coefficient. A value of (0.0) will default to an estimate based on UCRL data. If the

Table 16.24., ANIMAL Subroutine Input (continued)

		contaminant is organic, then a value of (0.0) will default to 0.5.
30.	BLTUR	Enter blood-to-urine clearance fraction, day^{-1}, for a given contaminant. A value of (0.0) will default to (0.693) for all contaminants, or a biological half-time of 1 day.
31.	BLTMK	Enter blood-to-milk transport fraction, day^{-1}, for a given contaminant. A value of (0.0) will default to the value set for GTBL above for all contaminants.
32.	PLTBL	Enter the lung pulmonary-to-blood fractional transport, day^{-1} for a given contaminant from particulates. A value of (0.0) will default to 0.1 for all contaminants.
33.	PLTBLG	Enter the lung pulmonary-to-blood partition coefficient for a given volatile contaminant. A value of (-1.0) is entered if the contaminant is non-volatile. A value of (0.0) will default to partition coefficient of 10.
34.	PLTG1	Enter the transport fraction, day-1, of 1 um AMAD particulates from the bronchus region of the lung to the gut. A value of 0.0 will default to 0.693 or a biological half-time of 1 day.
35.	PLTG2	Enter the transport fraction, day-1, of 1um AMAD particulates from the bronchiole region of the lung to the gut. A value of 0.0 will default to 0.0693, or a biological half-time of 10 days.
36.	FTTBL	Enter the fat-to-blood transport coefficient,day^{-1}, or partition coefficients. A positive value denote a transport coefficient, whereas, a negative number indicates that a partition coefficient has been entered. A value of (0.0) default to a transport coefficient of 0.17325, or a biological half-time of 4 days. A value of (-1000000) will default to a partition coefficient of 10. All transport coefficients are converted to partition coefficients internally in BIOTRAN.2.
37.	BNTBL1	Enter the 1st bone-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
38.	BNTBL2	Enter the 2nd bone-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
39.	LVTBL	Enter the liver-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.

Table 16.24., ANIMAL Subroutine Input (continued)

40.	KDTBL1	Enter the 1st kidney-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
41.	KDTBL2	Enter the 2nd kidney-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
42.	MSTBL	Enter the muscle-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
43.	RSTBL	Enter the residual tissue-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation. This value would represent the remaining tissues not included above.

If AGEA>0.0, and TPOPT > 1.0 as described above, then skip lines 44-47.

44.	A	Enter user specified population age structure for AGEA male age groups. These values are in fraction of the total male population for each age group.
45.	A	Enter user specified population age structure for AGEA female age groups. These values are in fraction of the total female population for each age group.
46.	TQX0	Enter user specified population mortality rates for AGEA male age groups. These values are in fraction year^{-1} of the total number of individuals in each male age group.
47.	TQX0	Enter user specified population mortality rates for AGEA female age groups. These values are in fraction year^{-1} of the total number of individuals in each female age group

If ALSP>0.0 as described above when the animal type is not an invertebrate, then skip lines 48-54.

48.	BLTGT	Enter absorption coefficient, day-1, from malphigian tubules to the gut of the specific invertebrate. A value of (0.0) will default to 0.693, or a 1day half-time.
49.	FDTMP	Enter the minimum invertebrate feeding temperature, °C, for a given invertebrate. A value of (0.0) will default to 16 °C.
50.	DVTMP	Enter the minimum developmental temperature, °C, for a given invertebrate. A value of (0.0) will default to 8 °C.

Table 16.24., ANIMAL Subroutine Input (continued)

51.	DEGDAY	Enter the minimum day accumulation temperature, degree-days, before egg hatching will occur. A value of (0.0) will default to 250 deg-days.
52.	COLDAY	Enter the minimum number of days below the minimum developmental temperature (DVTMP) required for cold treatment of developing diapause eggs. A value of (0.0) will default to 40 days.
53.	DIADAY	Enter Julian day at which 100% production of diapause eggs may develop at temperatures below DVTMP. A value of (0.0) will default to Julian day 240.
54.	EGGMRT	Enter the mortality rate, day-1, of regular, diapause. and post-diapause eggs averaged throughout the year. A value of (0.0) will default to 0.003 day^{-1}.

If ALSP =0.0, or the animal type is an invertebrate then skip lines 55-57.

55.	ACTI	Enter the activity factor above minimum metabolic need for a given animal type. Representative values of this factor are: 2=cows, 1.4=coyote, 2=rabbit, 1.4=mouse, 2=squirrel, 2=kangaroo rat, and 2=deer/elk.
56.	GASLOS	Enter digestion gas loss fraction from ingested food. Representative values of this fraction are: 0.06=cows, 0.01=coyote, 0.02=rabbit, 0.02=mouse/squirrel, 0.02=kangaroo rat, and 0.06=deer/elk.
57.	URNLOS	Enter loss fraction from ingested food from urine production. Representative values of this fraction are: 0.06=cows, 0.03=coyote, 0.02=rabbit, 0.05=mouse/squirrel, 0.05=kangaroo rat, and 0.06=deer/elk.

If YDOS= 0, as described above then skip EFFECTS input which will be described in another section of this report.

Repeat lines 7-57 for NAN animal types with some lines included/omitted based on the animal type being simulated.

Table 16.25.

Amarillo, Texas Crops and Beef and Dairy Cattle

MAIN subroutine

6.000000
1.000000
1.000000
1.000000 1.000000 0.0000000E+00 1.000000
7.000000
1.000000
1.000000
0.0000000E+00
0.0000000E+00
2.0 1.0

AIRAC Subroutine

111111
1.000000 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
1.000000
1.000000
3.000000
195.0000
1.000000e+08
0.000000

CLIMAT Subroutine

555555
277.0000 625.0000
90.00000
36.20000 78.00000
120.0000
4.360000
0.2000000
3500.000
4.0000002E-03
4.000000 4.000000 4.000000 4.000000 4.000000
4.000000 4.000000 4.000000 4.000000 4.000000
4.000000 4.000000
2.3000000E-02 2.6000001E-02 3.7000000E-02 6.1999999E-02 0.1490000
0.1610000 0.1430000 0.1500000 9.3000002E-02 8.9000002E-02
3.9000001E-02 2.8999999E-02
22.00000
3538.000
0.0000000E+00
202.0000
36.00000

WATFLX Subroutine, Part I

141414
0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
0.0000000E+00
 0.0000000E+00
WATFLX Subroutine, Part II
   151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 2.000000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 5.000000     5.000000     6.000000     7.000000     6.000000
 6.000000     5.000000
 990.6000     990.6000     1498.600     2006.600     1498.600
 1498.600     990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
 508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

0.4200000
0.3300000 0.3300000 0.4700000 0.4780000 0.4780000
0.4590000
18.70000 18.70000 18.90000 17.90000 16.50000
15.00000
30.00000 30.00000 30.10000 27.20000 27.50000
25.80000
28.50000 28.50000 28.60000 25.80000 26.10000
24.50000
2.000000 2.000000 2.000000 2.000000 2.000000
2.000000
20.00000 132.4000 203.2000 304.8000 330.2000
508.0000 508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00
0.1800000 0.1800000 0.1200000 0.1200000 0.1200000
0.1200000 0.1200000
0.4900000 0.4900000 0.4100000 0.4020000 0.4030000
0.4200000 0.4200000
0.3300000 0.3300000 0.4700000 0.4780000 0.4780000
0.4590000 0.4590000
18.70000 18.70000 18.90000 17.90000 16.50000
15.00000 15.00000
30.00000 30.00000 30.10000 27.20000 27.50000
25.80000 25.80000
28.50000 28.50000 28.60000 25.80000 26.10000
24.50000 24.50000
2.000000 2.000000 2.000000 2.000000 2.000000
2.000000 2.000000
20.00000 132.4000 203.2000 304.8000 330.2000
508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.1800000 0.1800000 0.1200000 0.1200000 0.1200000
0.1200000
0.4900000 0.4900000 0.4100000 0.4020000 0.4030000
0.4200000
0.3300000 0.3300000 0.4700000 0.4780000 0.4780000
0.4590000
18.70000 18.70000 18.90000 17.90000 16.50000
15.00000
30.00000 30.00000 30.10000 27.20000 27.50000
25.80000
28.50000 28.50000 28.60000 25.80000 26.10000
24.50000
2.000000 2.000000 2.000000 2.000000 2.000000
2.000000
20.00000 132.4000 203.2000 304.8000 330.2000
508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.1800000 0.1800000 0.1200000 0.1200000 0.1200000

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000
  28.50000     28.50000     28.50000     28.60000     25.80000
  26.10000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
  18.70000     18.70000     18.90000     17.90000     16.50000
  30.00000     30.00000     30.10000     27.20000     27.50000
  28.50000     28.50000     28.60000     25.80000     26.10000
  2.000000     2.000000     2.000000     2.000000     2.000000
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    2.1000000E-02 6.1000001E-02
6.3000000E-03
5.0099999E-02 0.5100000    0.3300000    0.1600000    7.9999998E-02
2.4000000E-02 7.8999996E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
PLTGRO Subroutine
   121212
0.0000000E+00
 1.000000
0.0000000E+00
 1.000000     2.000000     8.000000     11.00000     12.00000
 18.00000     23.00000
 6.000000     17.00000    0.0000000E+00  10.00000     5.000000
0.0000000E+00  17.00000
 2.000000     2.000000    0.0000000E+00  4.000000     2.000000
0.0000000E+00  2.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 1.000000     1.000000    0.0000000E+00  1.000000     1.000000
0.0000000E+00  1.000000
0.0000000E+00 0.0000000E+00  1.000000    0.0000000E+00 0.0000000E+00
 1.000000    0.0000000E+00
 4000.000     4000.000     4000.000     4000.000     4000.000
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
4000.000      4000.000
 302.0000      302.0000
0.0000000E+00  7.000000     6.000000     6.000000     6.000000
 6.000000
 150.0000      150.0000     201.0000     229.0000     243.0000
 257.0000
 302.0000      302.0000
0.0000000E+00  1.000000     1.000000     1.000000     1.000000
 1.000000      1.000000     1.000000     1.000000     1.000000
 2.000000      2.000000     2.000000     2.000000     2.000000
 2.000000      2.000000
 150.0000      150.0000     157.0000     164.0000     171.0000
 177.0000      184.0000     191.0000     198.0000     205.0000
 212.0000      219.0000     226.0000     233.0000     240.0000
 247.0000      254.0000
 1.000000      500.0000     531.0000     577.0000
0.0000000E+00  7.000000     5.000000     5.000000     5.000000
 5.000000      5.000000     5.000000     5.000000     5.000000
 215.0000      215.0000     470.0000     485.0000     502.0000
 514.0000      527.0000     533.0000     544.0000     559.0000
 1.000000      560.0000
0.0000000E+00  16.00000     5.000000     5.000000     5.000000
 250.0000      250.0000     285.0000     425.0000     500.0000
 302.0000      302.0000
0.0000000E+00  7.000000     2.000000     2.000000     2.000000
 2.000000      2.000000     2.000000     2.000000     2.000000
 3.000000      3.000000     3.000000     3.000000     3.000000
 3.000000      3.000000
 150.0000      150.0000     157.0000     164.0000     171.0000
 177.0000      184.0000     191.0000     198.0000     205.0000
 212.0000      219.0000     226.0000     233.0000     240.0000
 247.0000      254.0000
UPTAKE Subroutine
   131313
0
ANIMAL Subroutine
   222222
 2
550.0000      550.00000
 6.91 6.91
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 3.000000      1.000000
 15.000       15.000
 0.0 0.0 0.0
 0.0
 1.0
 50. 0.90 2.0
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
 1.000000
 1.000000
 1.000000
 3.0000
 15.0 1.0 1.0 1.0
 3.000000     8.000000
 1.000000     73.00000     74.00000     224.0000     225.0000
 365.0000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00  0.0000000E+00
3.00000 3.00000
 1.000000     2.000000     3.000000
 2.0
 11.00000     12.00000
 4.000000     1.000000
0.5000000     0.5000000
 3.0
 8.000000     11.00000     18.00000
 1.000000     4.000000     2.000000
0.3300000     0.3300000     0.3400000
 2.0
 1.000000     11.00000
 5.000000     4.000000
0.7800000     0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.0
0.0000000
0.0000000
0.000000
2.0
6.00000e-02
6.00000e-02
0.0 0.0 0.0
0.0
1.0
50.00000     0.90     2.000000
1.0
1.0
1.0
1.0
15.0 1.0 1.0 1.0
 3.000000     8.000000
 1.000000     365.0000
 1.000000    1.00000
 0.0 0.0
 1.0 1.0
 1.0
 2.0
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
1.000000      11.00000
 5.000000      4.000000
0.7800000     0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.000
0.000000
0.0000000
0.0
2.0
6.000e-02
6.000e-02
BIOPLT Subroutine
 444444
2.0
0.0 0.0 0.0
0.0 0.0 0.0
1.0 1.0
VLUG
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VLUG
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
VSOL
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VSOL
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
VPLN
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VPLN
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
VGUT
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VGUT
3.0 22.0
kr      ko      m
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

```
-2.0  -2.0 -1.0
CNBOD
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNBOD
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
2.0 1.0
CNBON
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNBON
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
CNMUS
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNMUS
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
CNLUN
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNLUN
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
CNKID
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNKID
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
CNLIV
3.0 3.0
kr      ko      m
-2.0  -1.0 -1.0
CNLIV
3.0 3.0
kr      ko      m
-2.0  -2.0 -1.0
3.0
4.0
```

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

TRUPLT Subroutine
171717
4.0
1.0 1.0 1.0 1.0
13.0
2.0 4.0 6.0 8.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
TPLOTR Subroutine
191919
4 3 25
1 1 1 1
y
y
y
n
year
pCi
U-238 DAIRY CATTLE INTAKE
lung
soil
feed
total
-111.0 -111.0
-111.0 -111.0
TRUPLT Subroutine
171717
4.0
1.0 1.0 1.0 1.0
13.0
1.0 3.0 5.0 7.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0
TPLOTR Subroutine
191919
4 3 24
1 1 1 1
y
y
y
n
year
pCi
U-238 BEEF CATTLE INTAKE
lung
soil
feed
total
-111.0 -111.0

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

-111.0 -111.0
TRUPLT Subroutine
171717
5.0
2.0 2.0 2.0 2.0 2.0
13.0
2.0 4.0 6.0 8.0 10.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
191919
TPLOTR Subroutine
4 9 33
1 1 1 1 1
y
y
y
n
year
pCi/kgfwt
U-238 DAIRY CATTLE CONCENTRATIONS
bone
muscle
lung
kidney
liver
-111. -111.
-111. -111.
TRUPLT Subroutine
171717
5.0
2.0 2.0 2.0 2.0 2.0
13.0
1.0 3.0 5.0 7.0 9.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
191919
TPLOTR Subroutine
4 9 32
1 1 1 1 1
y
y
y
n
year
pCi/kgfwt
U-238 BEEF CATTLE CONCENTRATIONS
bone
muscle
lung

Table 16.25., Amarillo, Texas Crops and Beef and Dairy Cattle (continued)

kidney
liver
-111. -111.
-111. -111.

Table 16.26.

Amarillo, Texas Grasshopper Biomass Density

```
MAIN Subroutine
 8.00000
 1.00000
 1.00000
 1.00000      1.00000      0.000000E+00    1.00000
 7.00000
 1.00000
 1.00000
0.000000E+00
0.000000E+00
 2.00000      1.00000
AIRAC Subroutine
 111111.
 1.00000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
0.000000E+00
 1.00000
 1.00000
 3.00000
 195.000
1.000000E+08
0.000000E+00
CLIMAT Subroutine
 555555.
 277.000      625.000
 90.0000
 36.2000      78.0000
 120.000
 4.36000
0.200000
 3500.00
4.000000E-03
 4.00000      4.00000      4.00000      4.00000      4.00000      4.00000      4.00000
 4.00000      4.00000      4.00000      4.00000      4.00000
 2.300000E-02   2.600000E-02   3.700000E-02   6.200000E-02   0.149000      0.161000
0.143000
 0.150000      9.300000E-02   8.900000E-02   3.900000E-02   2.900000E-02
 22.0000
 3538.00
0.000000E+00
 202.000
 36.0000
WATFLX Subroutine, Part I
 141414.
0.000000E+00   0.000000E+00
0.000000E+00
0.000000E+00
0.000000E+00
```

Table 16.12.3., Amarillo, Texas Grasshopper Biomass Density (continued)

```
0.000000E+00
WATLX Subroutine, Part II
  151515.
  92.0000     7.00000     7.00000     15.0000     15.0000     19.0000
  238.000     14.0000     14.0000     31.0000     31.0000     39.1000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
  1.00000     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
  2.00000     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
  1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
  1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
  5.00000     5.00000     6.00000     7.00000     6.00000     6.00000     5.00000
  990.600     990.600     1498.60     2006.60     1498.60     1498.60     990.600
  20.0000     132.400     203.200     304.800     330.200
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000    0.180000    0.120000    0.120000    0.120000
 0.490000    0.490000    0.410000    0.402000    0.402000
 0.330000    0.330000    0.470000    0.478000    0.478000
  18.7000     18.7000     18.9000     17.9000     16.5000
  30.0000     30.0000     30.1000     27.2000     27.5000
  28.5000     28.5000     28.6000     25.8000     26.1000
  2.00000     2.00000     2.00000     2.00000     2.00000
  20.0000     132.400     203.200     304.800     330.200
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000    0.180000    0.120000    0.120000    0.120000
 0.490000    0.490000    0.410000    0.402000    0.403000
 0.330000    0.330000    0.470000    0.478000    0.478000
  18.7000     18.7000     18.9000     17.9000     16.5000
  30.0000     30.0000     30.1000     27.2000     27.5000
  28.5000     28.5000     28.6000     25.8000     26.1000
  2.00000     2.00000     2.00000     2.00000     2.00000
  20.0000     132.400     203.200     304.800     330.200     508.000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000    0.180000    0.120000    0.120000    0.120000    0.120000
 0.490000    0.490000    0.410000    0.402000    0.403000    0.420000
 0.330000    0.330000    0.470000    0.478000    0.478000    0.459000
  18.7000     18.7000     18.9000     17.9000     16.5000     15.0000
  30.0000     30.0000     30.1000     27.2000     27.5000     25.8000
  28.5000     28.5000     28.6000     25.8000     26.1000     24.5000
  2.00000     2.00000     2.00000     2.00000     2.00000     2.00000
  20.0000     132.400     203.200     304.800     330.200     508.000     508.000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00
 0.180000    0.180000    0.120000    0.120000    0.120000    0.120000    0.120000
 0.490000    0.490000    0.410000    0.402000    0.403000    0.420000    0.420000
 0.330000    0.330000    0.470000    0.478000    0.478000    0.459000    0.459000
  18.7000     18.7000     18.9000     17.9000     16.5000     15.0000     15.0000
  30.0000     30.0000     30.1000     27.2000     27.5000     25.8000     25.8000
  28.5000     28.5000     28.6000     25.8000     26.1000     24.5000     24.5000
  2.00000     2.00000     2.00000     2.00000     2.00000     2.00000     2.00000
```

Table 16.26., Amarillo, Texas Grasshopper Biomass Density (continued)

```
 20.0000      132.400      203.200      304.800      330.200      508.000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000     0.180000     0.120000     0.120000     0.120000     0.120000
 0.490000     0.490000     0.410000     0.402000     0.403000     0.420000
 0.330000     0.330000     0.470000     0.478000     0.478000     0.459000
  18.7000      18.7000      18.9000      17.9000      16.5000      15.0000
  30.0000      30.0000      30.1000      27.2000      27.5000      25.8000
  28.5000      28.5000      28.6000      25.8000      26.1000      24.5000
  2.00000      2.00000      2.00000      2.00000      2.00000      2.00000
  20.0000      132.400      203.200      304.800      330.200      508.000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000     0.180000     0.120000     0.120000     0.120000     0.120000
 0.490000     0.490000     0.410000     0.402000     0.403000     0.420000
 0.330000     0.330000     0.470000     0.478000     0.478000     0.459000
  18.7000      18.7000      18.9000      17.9000      16.5000      15.0000
  30.0000      30.0000      30.1000      27.2000      27.5000      25.8000
  28.5000      28.5000      28.5000      28.6000      25.8000      26.1000
  2.00000      2.00000      2.00000      2.00000      2.00000      2.00000
  20.0000      132.400      203.200      304.800      330.200
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000     0.180000     0.120000     0.120000     0.120000
 0.490000     0.490000     0.410000     0.402000     0.403000
 0.330000     0.330000     0.470000     0.478000     0.478000
  18.7000      18.7000      18.9000      17.9000      16.5000
  30.0000      30.0000      30.1000      27.2000      27.5000
  28.5000      28.5000      28.6000      25.8000      26.1000
  2.00000      2.00000      2.00000      2.00000      2.00000
 8.800000E-02  0.620000     0.200000     6.100000E-02  2.100000E-02
 8.800000E-02  0.620000     0.200000     6.100000E-02  2.100000E-02
 8.800000E-02  0.620000     0.200000     2.100000E-02  6.100000E-02  6.300000E-03
 5.010000E-02  0.510000     0.330000     0.160000      8.000000E-02  2.400000E-02
7.900000E-03
 8.800000E-02  0.620000     0.200000     6.100000E-02  2.100000E-02  6.300000E-03
 8.800000E-02  0.620000     0.200000     6.100000E-02  2.100000E-02  6.300000E-03
 8.800000E-02  0.620000     0.200000     6.100000E-02  2.100000E-02
PLTGRO Subroutine
 121212.
 0.000000E+00
  1.00000
 0.000000E+00
  1.00000      2.00000      8.00000      11.0000      12.0000      18.0000      23.0000
  6.00000      17.0000      0.000000E+00  10.0000      5.00000      0.000000E+00  17.0000
  2.00000      2.00000      0.000000E+00  4.00000      2.00000      0.000000E+00  2.00000
  1.00000      1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
  1.00000      1.00000      0.000000E+00  1.00000      1.00000      0.000000E+00  1.00000
 0.000000E+00  0.000000E+00  1.00000      0.000000E+00  0.000000E+00  1.00000
0.000000E+00
  4000.00      4000.00      8000.00      4000.00      4000.00      8000.00      4000.00
  302.000      302.000
 0.000000E+00  7.00000      6.00000      6.00000      6.00000      6.00000
  150.000      150.000      201.000      229.000      243.000      257.000
```

Table 16.26., Amarillo, Texas Grasshopper Biomass Density (continued)

```
 302.000      302.000
 0.000000E+00   1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
  1.00000      1.00000      1.00000      2.00000      2.00000      2.00000      2.00000
  2.00000      2.00000      2.00000
  150.000      150.000      157.000      164.000      171.000      177.000      184.000
  191.000      198.000      205.000      212.000      219.000      226.000      233.000
  240.000      247.000      254.000
  1.00000      500.000      531.000      577.000
 0.000000E+00   7.00000      5.00000      5.00000      5.00000      5.00000      5.00000
  5.00000      5.00000      5.00000
  215.000      215.000      470.000      485.000      502.000      514.000      527.000
  533.000      544.000      559.000
  1.00000      560.000
 0.000000E+00   16.0000      5.00000      5.00000      5.00000
  250.000      250.000      285.000      425.000      500.000
  302.000      302.000
 0.000000E+00   7.00000      2.00000      2.00000      2.00000      2.00000      2.00000
  2.00000      2.00000      2.00000      3.00000      3.00000      3.00000      3.00000
  3.00000      3.00000      3.00000
  150.000      150.000      157.000      164.000      171.000      177.000      184.000
  191.000      198.000      205.000      212.000      219.000      226.000      233.000
  240.000      247.000      254.000
UPTAKE Subroutine
  131313.
     0
ANIMAL Subroutine
  222222.
     1
  1.00000
  6.91000
 0.000000E+00
 0.000000E+00
 0.000000E+00
 0.000000E+00
 0.000000E+00
 1.00000
 0.000000E+00
 0.000000E+00   0.000000E+00   0.000000E+00
 0.000000E+00
  1.00000
  1.00000      0.600000      4.00000
  1.00000
  1.00000
  1.00000
  1.00000
 1.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
 3.00000      8.00000
 1.00000      365.000
 1.00000      1.00000
 0.000000E+00   0.000000E+00
 1.00000      1.00000
```

Table 16.26., Amarillo, Texas Grasshopper Biomass Density (continued)

```
    1.00000
    2.00000
    8.00000        18.0000
    1.00000        2.00000
   0.750000       0.250000
   2.000000E-02
   0.231000
   0.000000E+00
   2.000000E-02
   -1.00000
   0.000000E+00
   0.000000E+00
   0.000000E+00
INVERT Subroutine
    181818.
   0.631000
   0.631000
   0.631000
   0.631000
   0.631000
   0.631000
    16.0000
    8.00000
    250.000
    40.0000
    240.000
   3.000000E-03
BIOPLT Subroutine
        444444.
    1.00000
   0.000000E+00   0.000000E+00   0.000000E+00
    1.00000        1.00000
ADLT
    2.00000        1.00000
2       ko
   -1.00000       -1.00000
ADLT
    2.00000        1.00000
2       ko
   -2.00000       -1.00000
ADLT
    2.00000        1.00000
2       ko
   -1.00000       -1.00000
ADLT
    2.00000        1.00000
2       ko
   -2.00000       -1.00000
ADLT
    2.00000        1.00000
2       ko
```

Table 16.26., Amarillo, Texas Grasshopper Biomass Density (continued)

```
   -1.00000      -1.00000
ADLT
    2.00000       1.00000
2       ko
   -2.00000      -1.00000
ADLT
    2.00000       1.00000
2       ko
   -1.00000      -1.00000
ADLT
    2.00000       1.00000
2       ko
   -2.00000      -1.00000
ADLT
    2.00000       1.00000
2       ko
   -1.00000      -1.00000
ADLT
    2.00000       1.00000
2       ko
   -2.00000      -1.00000
3.0
1.0
TRUPLT Subroutine
171717
2
1 1
13
1 2
0 0
0 0
0 0
TPLOTR Subroutine
191919
5 7 27
1 1
y
y
y
n
years
gdwt/m2
Grasshopper Biomass Density
male
female
-111.0 -111.0
-111.0 -111.0
```

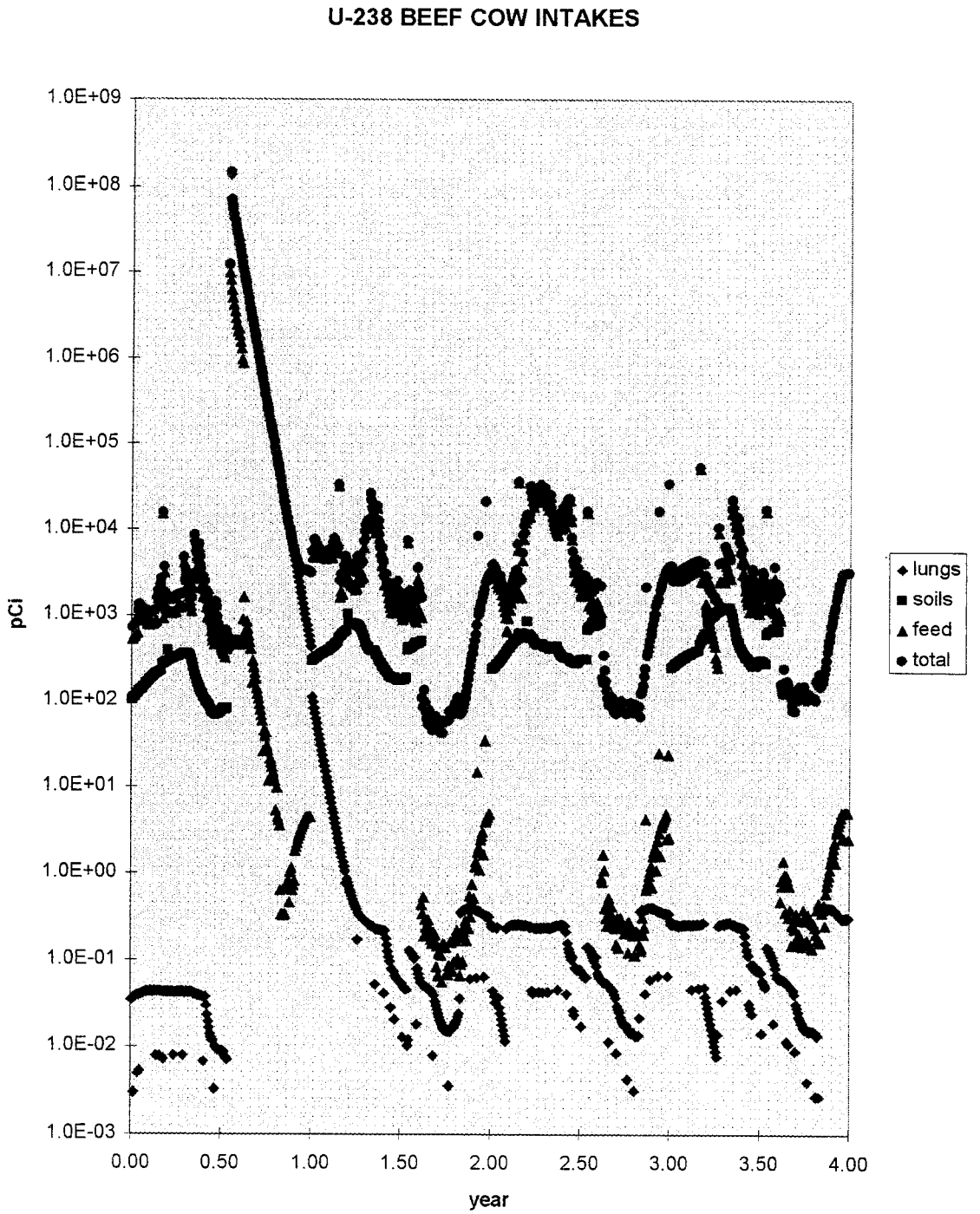


Figure 16.30.

Beef Cattle Air, Soil, Feed, and Total Uranium Intakes For Amarillo, Texas

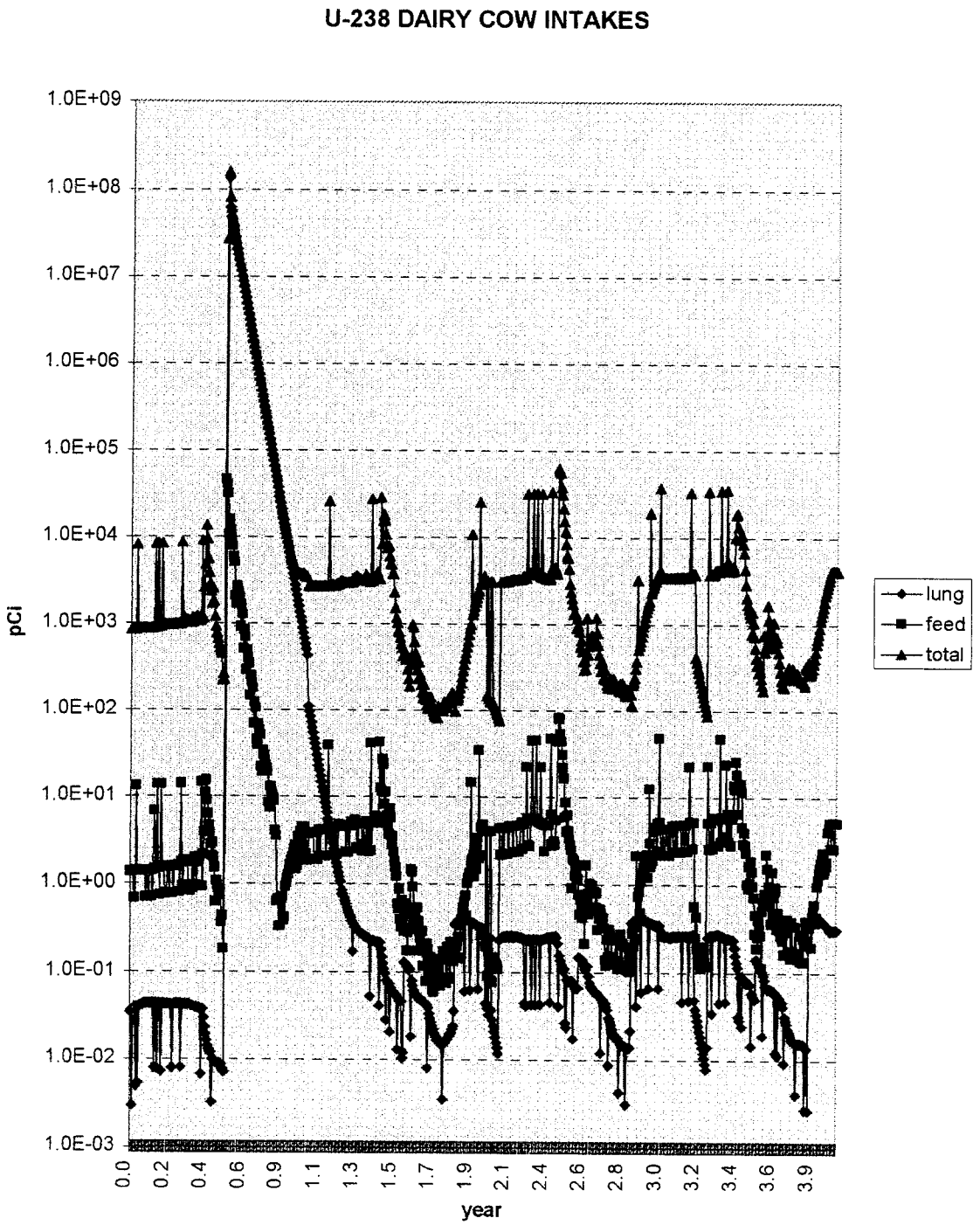


Figure 16.31.

Dairy Cattle Air, Feed, and Total Uranium Intake for Amarillo, Texas.

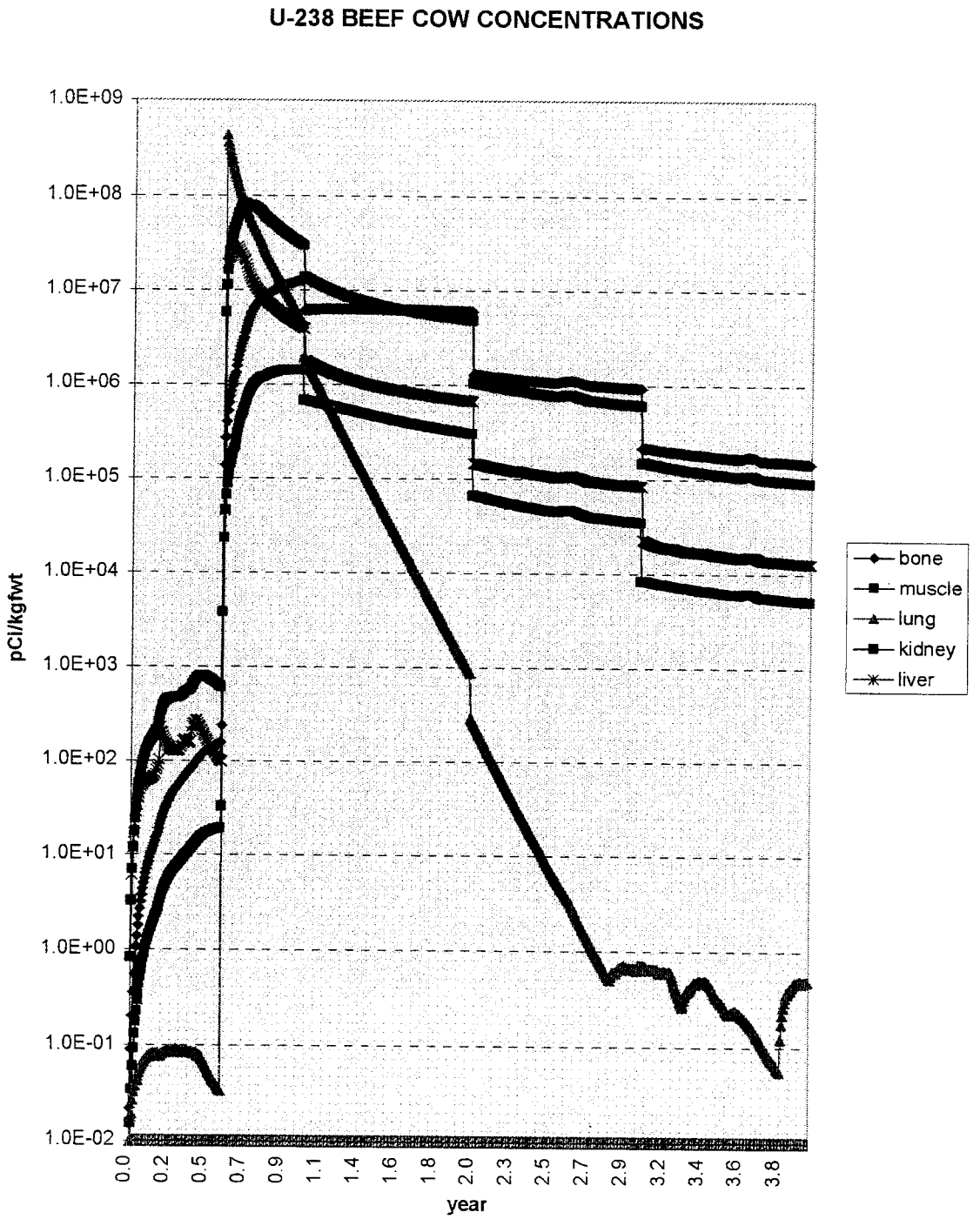


Figure 16.32.

Beef Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas.

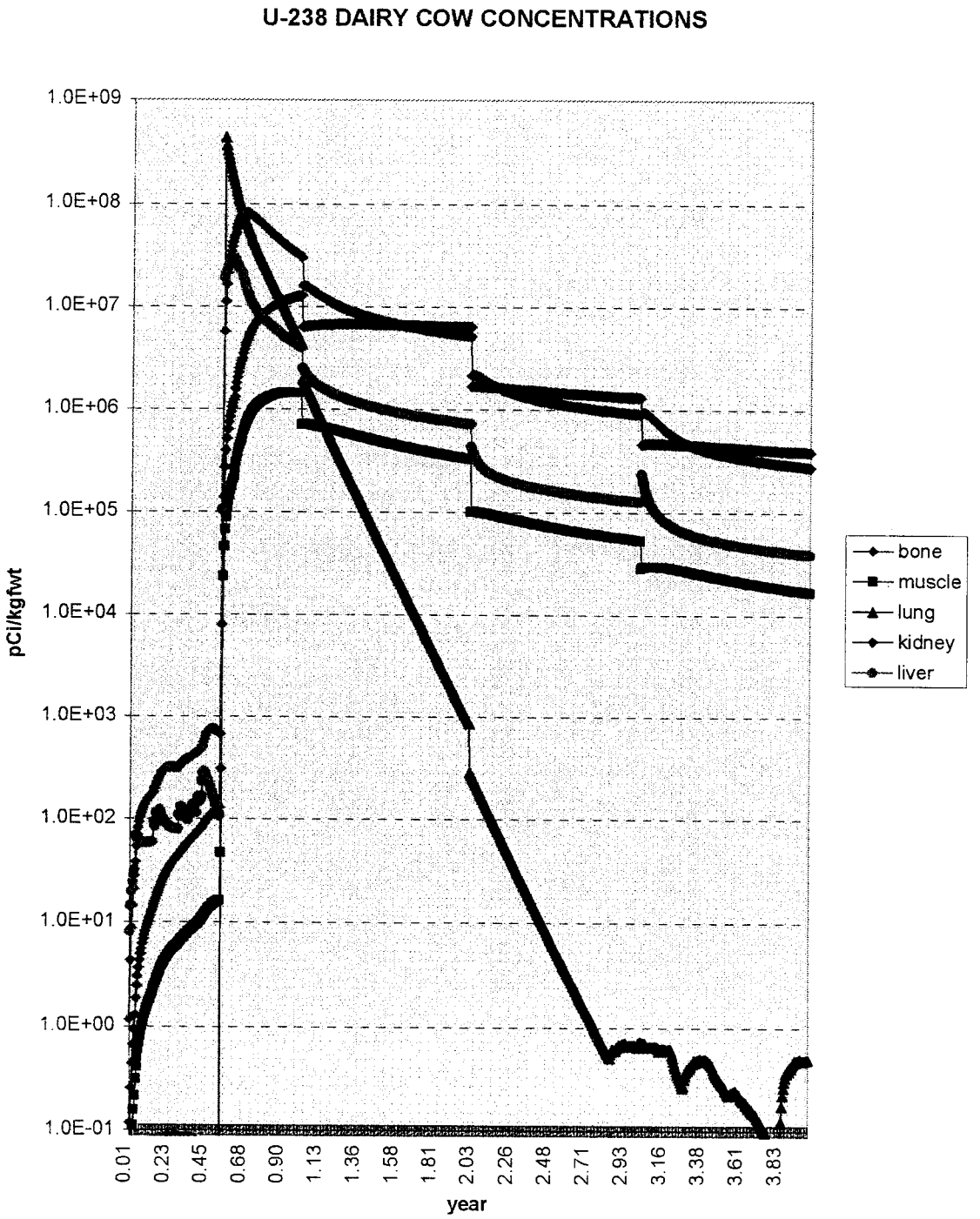


Figure 16.33.

Dairy Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas.

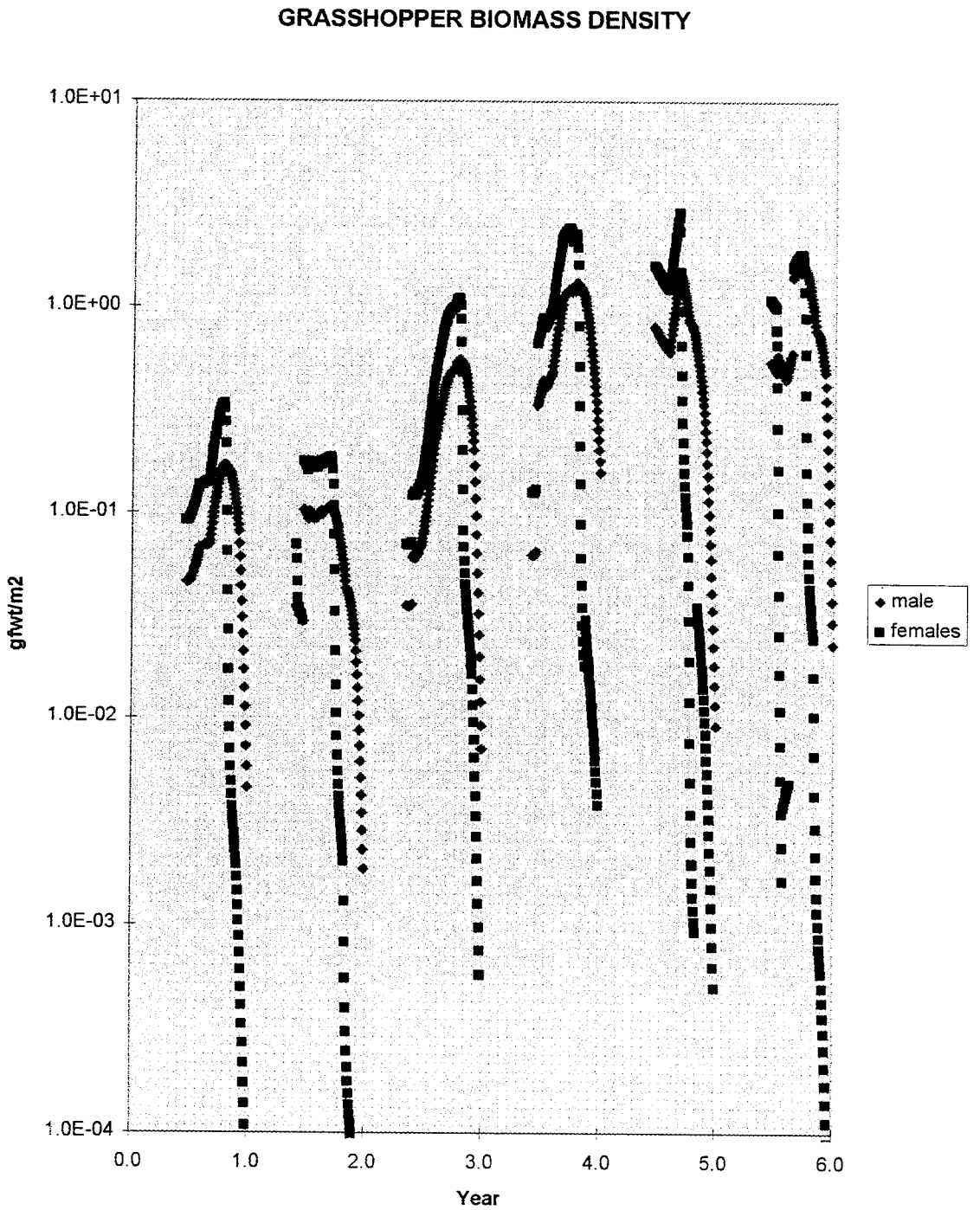


Figure 16.34.

Male and Female Grasshopper Biomass Densities for Amarillo, Texas

16.12.3. References

ICRP. 1966. Deposition and retention models for internal dosimetry of the human respiratory tract. International Commission on Radiological Protection. Health Physics (12), pp. 173-207.

Wenzel, W.J., Wallwork-Barber, K.M., Horton, J.M., Rea, K.H., Hollis, L.C., Gladney, E.S., Mayfield, D.L., Gallegos, A.F., Rodgers, J.C., Thomas, R.G, and G. Trujillo. 1982. Supplementary documentation for an environmental impact statement regarding the Pantex Site. Agricultural food chain radiological assessment. Los Alamos National Laboratory, Los Alamos, NM. LA-9445-PNTX-M.

Wenzel, W.J., Rodgers, J.C,, and A.F. Gallegos. 1982. The Texas Panhandle soil-crop-beef food chain for uranium. A dynamic model validated by experimental data. DOE. CONF-821215. pp311-321.

Randell, R.L. 1970. CARSGHOP/(26k). A simulation model of a grasshopper population. J. Canada Department of Agriculture. Research Branch. Research Station. Saskatoon, Sask.

Kelly, E., M. Kramer, C. McDaniel, P. Newell, K. Sasser, D. Stack, and A. Gallegos. 1995. Integrating ecological risk assessment and hazards analysis: a pilot project for the proposed Los Alamos Waste Treatment and Storage Facilities, Los Alamos National Laboratory (LA-UR-95-2555). 62p.

16.13. HUMTRN Subroutine - Human Simulation and Dose

16.13.1. Description

An illustration of the overall HUMTRN model as incorporated into BIOTRAN.2 is presented in Figure 16.35.., and a more detailed description of the ICRP lung model component of this subroutine was illustrated previously for animals (Figure 16.25.). With the exception of direct ingestion of soil by animals, and a scaling of air, water consumption, and metabolic consumption rates algorithms described for these organisms, HUMTRN resembles its animal counterpart ANIMAL described in the last section. This particularly true for given input requirements. The physiologic, metabolic, and risk analysis documentation and verification for HUMTRN was presented by Gallegos (1984). HUMTRN was developed from basic anatomical, physiological, and metabolic data obtained from ICRP 23, information from Brody (1945), and Adams (1975). Air, water, and food volume, mass and contaminant concentrations for the simulated human build up of contaminants in human tissues are supplied from various BIOTRAN.2 subroutines (in a manner similar to that described for subroutine ANIMAL). Male and female humans are simulated separately, although integrated through female reproduction, are derived from pertinent 1980 Census information included in the model, or a population can be created by the user by selecting a population structure (usually but not always a parabolic structure), in addition to specifying the life-span and other population parameters. The user can also enter a specific population structure if required. Food, water, and air intake, uptake, and retention of contaminants in the organs shown in Figure 16.36. in each age group can be tracked separately for each type of intake for each contaminant. Radiation doses from radionuclide contaminants can also be simulated for each intake such as for leafy vegetables, meat, milk, air, or drinking water can also be obtained separately for each organ or sum of organs. Man-rems from radionuclide exposures can be summed over age groups and/or sex. Output HUMTRN can also be coupled with subroutine EFFECTS to estimate health effects and cancer mortality from acute and chronic radionuclide contaminant exposures. for each age group and sex over any specified time period. This unique capability was developed from the work of Buhl (1984). Population dynamics and realistic health effects projections can be performed using EFFECTS (which will be described later in this manual) coupled to HUMTRN (Wenzel 1985). Using HUMTRN, the user can focus on specific details such as day-to-day diets and intakes for complex acute scenarios or long-term changes in radiation dose over hundreds of simulation years. The first PC version of HUMTRN and EFFECTS was described by Gallegos (1989), as it was not yet possible to execute all of BIOTRAN.2 on a PC.

It is not yet possible to use HUMTRN and EFFECTS to estimate human health risks from non-radionuclide contaminants using the methodology employed for radionuclides, however, it is possible to use model predictions of organ and tissue body burdens, and contaminant intake rates which can be used to make health effects estimates. The modification of EFFECTS to predict non-radionuclide health effects is currently under consideration.

16.13.2. Input/Output

Input requirements for execution of HUMTRN are very similar to input requirements for input to subroutine ANIMAL as shown in Table 16.27. A typical input set for BIOTRAN.2 involving HUMTRN access is presented in Table 16.28. for Amarillo, Texas. The latter is a modification and extension of the input required for the simulation of beef and dairy cattle (Table 16.25.). Output for human organ and tissue uptake, and resulting organ and tissue burdens for 25 year-old males are presented in Figures 16.36. and 16.37., respectively. The plots show the effect of an acute contamination event on Julian day 195 of the fourth year of simulation time.

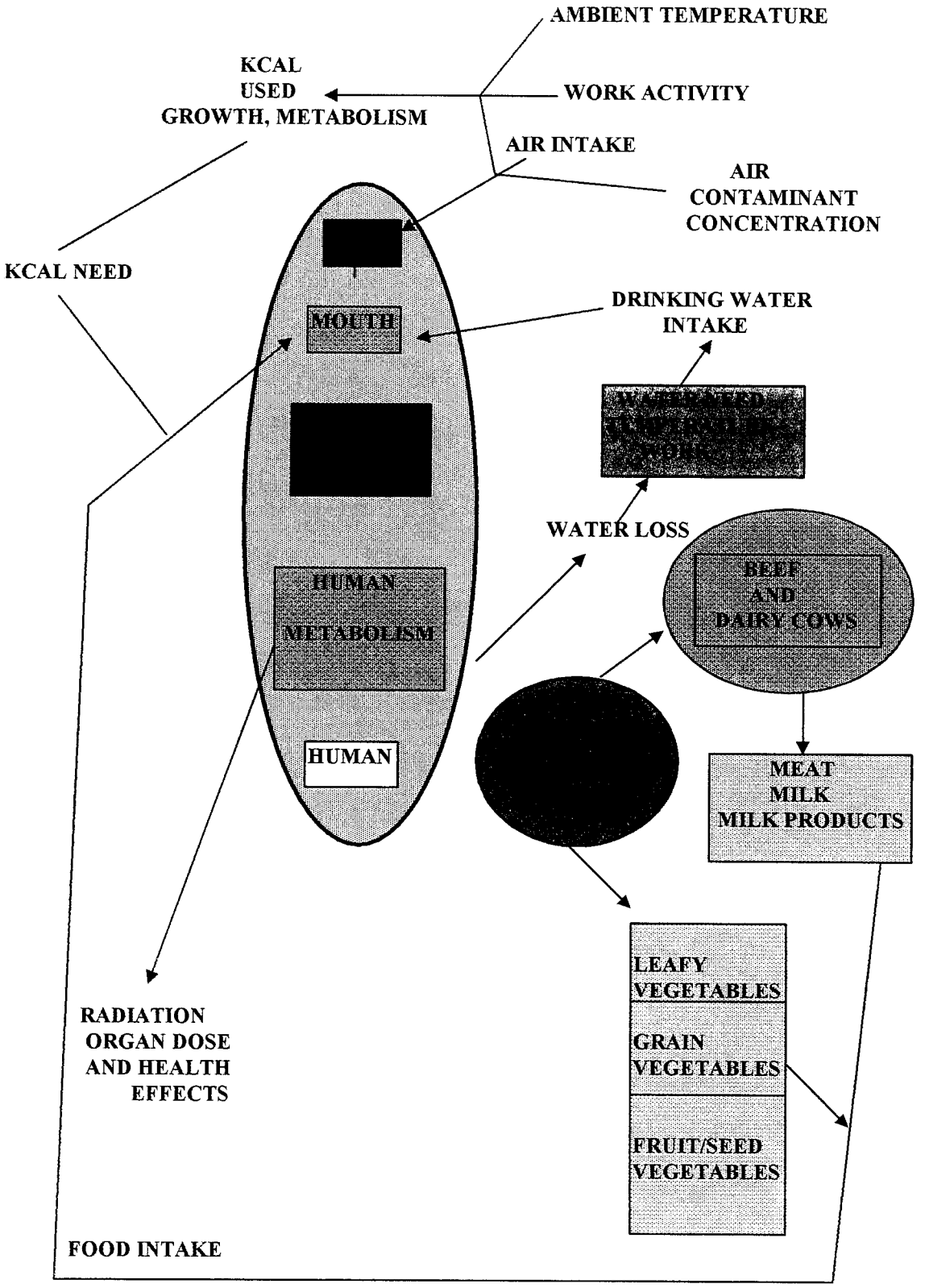


Figure 16.35.

HUMTRN Model Integrates the Volume, Mass, and Contaminant Intake Into Human Male and Female Population. Humans are Simulated Based On User Diet Selection and Population Dynamics

Table 16.27.

HUMTRN Subroutine Input

Input Line	Variable Name	Definition
1.		***Enter 999999 to access HUMTRN***
2.	RR	Enter mature weight, kgfwt, of males and females simulated If a value of (1.0) is entered, then default population parameters will be selected. If the negative of the mature weights are entered, then the user will enter population parameters, otherwise they will default to model estimates. The index value for humans are set to NAN, and NAN +1, respectively, for males and females. For example, if two animal populations are being simulated (NAN=2) in addition to male and female humans, then the index number for males=3, and for females=4. The latter information is used for plotting purposes.
3.	ACTI	Enter the activity factor above minimum metabolic need for the human males being simulated. This value is generally set to (1.0) for normal human activity.
4.	RZOON	Enter number of feeding schedules for males in the human population. A value of (1.0) is normally used for this purpose, and assumes that humans do not change their diet throughout the year, and over a period of years.
5.	EVS	Enter soft-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.
6.	EVB	Enter bone-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.
7.	EVE	Enter bone endosteal-tissue radiation deposition rate, MeV-rem/ disintegration for a given radionuclide contaminant. Enter (0.0) for non radionuclides, or if the user does not wish to calculate radiation doses in a simulation.
8.	YDOS	Enter the number of years over which effects of a given radionuclide will be performed if desired by the user. A value of (0.0) is entered for non-radionuclides, or if the user does not wish to simulate effects of radiation (fatal cancer production) on the specific human male population

Repeat Lines 3-8 for Female

Table 16.27., HUMTRN Subroutine Input (continued)

9.	WATCNI	Enter the concentation in drinking water , ppm, pCi...uCi..etc., for all contaminants and nutrients (ANZ)for the human males and females.

Repeat all lines of input noted for line 9 above for each region simulated.

10.	P5	Enter the fraction of meat consumed from the total area(s) simulated by human males in the population. A value of (1.0) is generally entered unless humans are not present on the site during portions of the year. This value applies even if a multi-regional simulation is specified.
11.	ZOO	Enter the total population size for males a given human population.
	POPT	Enter the population structure desired for males of a given population : 1=none, 2=parabolic, 3=exponential, 4=hyperbolic. A value of (1.0) indicates that the user will enter the population structure below or that a default population structure will be used. A value of (2.0) is normally entered to simulate a parabolic structure by the user, if a selection value of (1.0) is not desired.
12.	AA2	Enter the fraction of the area modeled that is used by human males. This parameter is generally set to (1.0), unless another area is used for a portion of the year. This value applies even if a multi-regional simulation is specified.
13.	FSUPF	Enter the fraction of non-meat food that is derived from the study area for human males. This value is generally set to (1.0), unless males consume plant foods that has not been grown in the study area for part of the year. this value applies even if a multi-regional simulation is specified.
14.	EATIN	Enter the number of yearly feeding intervals for a given for human males. The latter specifies the number of times throughout the simulation when a new diet is to be specified : 1-10, 11-20, as an example for 2 changes in strategy during the simulation..etc.. A value of (1.0) is generally set for humans in areas where a rather constant diet is propagated through time.
15.	FODIN	Enter the number feeding schedules (on a daily basis) specified during any specific yearly interval for a human male : 1-74, 75-365, as an example for 2 changes in the diet (different plant types and/or parts) during the simulation. An interval (1-365) is generally set for most human populations.

Table 16.27., HUMTRN Subroutine Input (continued)

16.	AGEA	Enter the maximum expected life-span (in whole years) for human males in a given population. For example, a male population with a specified mean life-span (ALSP) of 75.4 years would be given a maximum expected life-span of 76.0 years..etc... A value of (0.0) defaults to a maximum life span of 85 years for males and females.
	GST	Enter the initiation time of the gestation period: 1=Spring, 2=Fall, for a given female human (enter even if male).
	GSTN	Enter the number of gestation periods/year (limited to a maximum of 2). A value of (1.0) is used (enter even if male)
	FRPR	Enter the fraction of the off-spring produced during the first gestation period (may equal 1.0 only if one gestation period is specified). A value of (1.0) is generally used (enter even if male).
17.	YRFED	Enter the yearly feeding intervals that have different feed schedules for male humans: 1,10, 11,20, 21,50, as an example where EATIN=3..etc.., specified by the user over a 50 year simulation period. Enter 2 values/interval. Usually one interval is specified for humans throughout the simulation period.
18.	ONFOD	Enter daily feeding interval that have different diets during a given yearly interval(s) for a human males: 1,74, 75,230, 76,365, as an example where EATIN=1, and FODIN=3..etc.., as specified by the user for a given simulation period. Enter 2 values/interval. An interval of (1, 365) is generally used for humans.
19.	EATREC	Enter the duplicate feeding intervals for all yearly intervals specified: 3,3, 1,1, as an example where there are 2 yearly intervals (EATIN=2)..etc.. The latter would specify 3 food schedules during the first yearly interval, and 1 food schedule for the second yearly interval of changed food schedules.
20.	PLTIND	Enter the number of changed feed schedules (RZOON) for human males: 1,2,3, as an example where RZOON=3 for a given human population. The latter is used in the BIOTRAN.2 code set up the specific plant types used in the diet of human males throughout the simulation period.
21.	AISCH	Enter the number of plant types used for a specific food schedule for human males of the number specified in RZOON above.
22.	PLTUSD	Enter the AISCH plant types for the specific food schedule in question for human males: 1.0, 2.0, as an example where corn and vegetables are specified for this yearly interval where AISCH=2. Generally three types of foods are

Table 16.27., HUMTRN Subroutine Input (continued)

		selected for humans as described below for the next input requirement.
23.	PLTRNK	Enter feeding preferences for food parts (edible crops for humans) for the plants identified in line 22. Food parts are identified as 4=all plant parts excluding roots, 5=seeds and grains, 6=green and dry forage, 7=temporarily assigned as seeds and grains. Currently feeding preferences 4,5,and 7 are required by HUMTRN for all simulations since a human diet comparable to the typical American diet is contracted in this subroutine. If one or more of these types are missing, then the model will attempt to utilize those present, but a "starvation" warning may ensue if harvested crops are not adequate to sustain the specified population. Plant rankings 1,2,3, and 4 are currently ignored by the model as a source of human food.
24.	PLTF	Enter the utilization fraction of each plant and/or plant part in the diet of the human male. The sum of the fractions should equal 1.0 even on multi-region basis.

Repeat Lines 21-24 for each feeding schedule specified by RZOON

Repeat line 20-24 for each of APX regions simulated.

If the user has not entered the negative of the mature weight of a given animal type (RR≥0.0), then skip lines 25-27.

25.	PLCM1	Enter deposition fraction for 1um AMAD particles in the naso-pharangeal (N-P) region of the lung for a given animal type.
26.	PLCM2	Enter deposition fraction for 1um AMAD particles in the tracheo-bronchiole (T-B) region of the lung for a given animal type.
27.	PLCM3	Enter deposition fraction for 1 um AMAD particles in the pulmonary region (P) of the lung for a given animal type.
28.	GTBL	Enter the gut-to-blood absorption coefficient. A value of (0.0) will default to an estimate based on UCRL data. If the contaminant is organic, then a value of (0.0) will default to 0.5.
29.	BLTUR	Enter blood-to-urine clearance fraction, day^{-1}, for a given contaminant. A value of (0.0) will default to (0.693) for all contaminants, or a biological half-time of 1 day.
30.	BLTMK	Enter blood-to-milk transport fraction, day^{-1}, for a given contaminant. A value of (0.0) will default to the value set for GTBL above for all contaminants.

Table 16.27., HUMTRN Subroutine Input (continued)

31.	PLTBL	Enter the lung pulmonary-to-blood fractional transport, day^{-1} for a given contaminant from particulates. A value of (0.0) will default to 0.1 for all contaminants.
32.	PLTBLG	Enter the lung pulmonary-to-blood partition coefficient for a given volatile contaminant. A value of (-1.0) is entered if the contaminant is non-volatile. A value of (0.0) will default to partition coefficient of 10.
33.	PLTG1	Enter the transport fraction, day-1, of 1 um AMAD particulates from the bronchus region of the lung to the gut. A value of 0.0 will default to 0.693 or a biological half-time of 1 day.
34.	PLTG2	Enter the transport fraction, day-1, of 1um AMAD particulates from the bronchiole region of the lung to the gut. A value of 0.0 will default to 0.0693, or a biological half-time of 10 days.
35.	FTTBL	Enter the fat-to-blood transport coefficient,day^{-1}, or partition coefficients. A positive value denote a transport coefficient, whereas, a negative number indicates that a partition coefficient has been entered. A value of (0.0) default to a transport coefficient of 0.17325, or a biological half-time of 4 days. A value of (-1000000) will default to a partition coefficient of 10. All transport coefficients are converted to partition coefficients internally in BIOTRAN.2.
36.	BNTBL1	Enter the 1st bone-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
37.	BNTBL2	Enter the 2nd bone-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
38.	LVTBL	Enter the liver-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
39.	KDTBL1	Enter the 1st kidney-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
40.	KDTBL2	Enter the 2nd kidney-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.

Table 16.27., HUMTRN Subroutine Input (continued)

41.	MSTBL	Enter the muscle-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation.
42.	RSTBL	Enter the residual tissue-to-blood transport coefficient (+), or partition coefficient (-) for a given contaminant. A value of (0.0) will exclude this component from consideration in the simulation. This value would represent the remaining tissues not included above.

If AGEA>0.0, and TPOPT > 1.0 as described above, then skip lines 43-46.

43.	A	Enter user specified population age structure for AGEA male age groups. These values are in fraction of the total male population for each age group.
45.	TQX0	Enter user specified population mortality rates for AGEA male age groups. These values are in fraction year^{-1} of the total number of individuals in each male age group.

If entering human male data, then skip line 46.

46.	BRTY	Enter number of births/female for 8 age groups between 15% and 75% of the female lifespan. Trailing 0.0's may be added if not all intervals are need or desired by the user, depending on the differential reproductive rate with age.
47.	WT1	Enters mean weights, kgfwt, for male humans for AGEA age groups. Starting with 1 year-olds and proceeding to the age of the oldest individual in the population.

If YDOS= 0, as described above then skip EFFECTS input which will be described in another section of this report.

Repeat lines 10-47 for human females with some lines included/omitted based on which sex is being simulated.

Table 16.28.

Plants, Cattle, and Human Input

Main Subroutine

6.000000
1.000000
1.000000
1.000000 1.000000 1.0000000E+00 1.000000
7.000000
1.000000
1.000000
0.0000000E+00
0.0000000E+00
2.0 1.0

AIRAC Subroutine

111111
1.000000 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
1.000000
1.000000
4.000000
195.0000
1.000000e+08
0.000000

CLIMAT Subroutine

555555
277.0000 625.0000
90.00000
36.20000 78.00000
120.0000
4.360000
0.2000000
3500.000
4.0000002E-03
4.000000 4.000000 4.000000 4.000000 4.000000
4.000000 4.000000 4.000000 4.000000 4.000000
4.000000 4.000000
2.3000000E-02 2.6000001E-02 3.7000000E-02 6.1999999E-02 0.1490000
0.1610000 0.1430000 0.1500000 9.3000002E-02 8.9000002E-02
3.9000001E-02 2.8999999E-02
22.00000
3538.000
0.0000000E+00
202.0000
36.00000

WATLX Subroutine, Part I

141414
0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00

Table 16.28., Plants, Cattle, and Human Input (continued)

```
0.0000000E+00
0.0000000E+00
WATFLX Subroutine, Part II
  151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 2.000000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 5.000000     5.000000     6.000000     7.000000     6.000000
 6.000000     5.000000
 990.6000     990.6000     1498.600     2006.600     1498.600
 1498.600     990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
 508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
```

Table 16.28., Plants, Cattle, and Human Input (continued)

0.4200000					
0.3300000	0.3300000	0.4700000	0.4780000	0.4780000	
0.4590000					
18.70000	18.70000	18.90000	17.90000	16.50000	
15.00000					
30.00000	30.00000	30.10000	27.20000	27.50000	
25.80000					
28.50000	28.50000	28.60000	25.80000	26.10000	
24.50000					
2.000000	2.000000	2.000000	2.000000	2.000000	
2.000000					
20.00000	132.4000	203.2000	304.8000	330.2000	
508.0000	508.0000				
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	
0.0000000E+00	0.0000000E+00				
0.1800000	0.1800000	0.1200000	0.1200000	0.1200000	
0.1200000	0.1200000				
0.4900000	0.4900000	0.4100000	0.4020000	0.4030000	
0.4200000	0.4200000				
0.3300000	0.3300000	0.4700000	0.4780000	0.4780000	
0.4590000	0.4590000				
18.70000	18.70000	18.90000	17.90000	16.50000	
15.00000	15.00000				
30.00000	30.00000	30.10000	27.20000	27.50000	
25.80000	25.80000				
28.50000	28.50000	28.60000	25.80000	26.10000	
24.50000	24.50000				
2.000000	2.000000	2.000000	2.000000	2.000000	
2.000000	2.000000				
20.00000	132.4000	203.2000	304.8000	330.2000	
508.0000					
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	
0.0000000E+00					
0.1800000	0.1800000	0.1200000	0.1200000	0.1200000	
0.1200000					
0.4900000	0.4900000	0.4100000	0.4020000	0.4030000	
0.4200000					
0.3300000	0.3300000	0.4700000	0.4780000	0.4780000	
0.4590000					
18.70000	18.70000	18.90000	17.90000	16.50000	
15.00000					
30.00000	30.00000	30.10000	27.20000	27.50000	
25.80000					
28.50000	28.50000	28.60000	25.80000	26.10000	
24.50000					
2.000000	2.000000	2.000000	2.000000	2.000000	
2.000000					
20.00000	132.4000	203.2000	304.8000	330.2000	
508.0000					
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	
0.0000000E+00					

Table 16.28., Plants, Cattle, and Human Input (continued)

```
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
 0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000
  28.50000     28.50000     28.50000     28.60000     25.80000
  26.10000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
8.8000000E-02 0.6200000    0.2000000    2.1000000E-02 6.1000001E-02
6.3000000E-03
5.0099999E-02 0.5100000    0.3300000    0.1600000    7.9999998E-02
2.4000000E-02 7.8999996E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
6.3000000E-03
8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
PLTBRO Subroutine
   121212
0.0000000E+00
 1.000000
0.0000000E+00
 1.000000     2.000000     8.000000     11.00000     12.00000
 18.00000     23.00000
 6.000000     17.00000    0.0000000E+00  10.00000     5.000000
0.0000000E+00  17.00000
 2.000000     2.000000    0.0000000E+00  4.000000     2.000000
0.0000000E+00  2.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 1.000000     1.000000    0.0000000E+00  1.000000     1.000000
0.0000000E+00  1.000000
0.0000000E+00 0.0000000E+00  1.000000    0.0000000E+00 0.0000000E+00
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
 1.000000     0.0000000E+00
 4000.000     4000.000     4000.000     4000.000     4000.000
 4000.000     4000.000
 302.0000     302.0000
0.0000000E+00  7.000000     6.000000     6.000000     6.000000
 6.000000
 150.0000     150.0000     201.0000     229.0000     243.0000
 257.0000
 302.0000     302.0000
0.0000000E+00  1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 2.000000     2.000000     2.000000     2.000000     2.000000
 2.000000     2.000000
 150.0000     150.0000     157.0000     164.0000     171.0000
 177.0000     184.0000     191.0000     198.0000     205.0000
 212.0000     219.0000     226.0000     233.0000     240.0000
 247.0000     254.0000
 1.000000     500.0000     531.0000     577.0000
0.0000000E+00  7.000000     5.000000     5.000000     5.000000
 5.000000     5.000000     5.000000     5.000000     5.000000
 215.0000     215.0000     470.0000     485.0000     502.0000
 514.0000     527.0000     533.0000     544.0000     559.0000
 1.000000     560.0000
0.0000000E+00  16.00000     5.000000     5.000000     5.000000
 250.0000     250.0000     285.0000     425.0000     500.0000
 302.0000     302.0000
0.0000000E+00  7.000000     2.000000     2.000000     2.000000
 2.000000     2.000000     2.000000     2.000000     2.000000
 3.000000     3.000000     3.000000     3.000000     3.000000
 3.000000     3.000000
 150.0000     150.0000     157.0000     164.0000     171.0000
 177.0000     184.0000     191.0000     198.0000     205.0000
 212.0000     219.0000     226.0000     233.0000     240.0000
 247.0000     254.0000
UPTAKE Subroutine
   131313
0
Animal Subroutine
   222222
 2
550.0000     550.00000
 6.91 6.91
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 3.000000     1.000000
 15.000     15.000
 0.0 0.0 0.0
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
 0.0
 1.0
 99. 0.90 2.0
 1.000000
 1.000000
 1.000000
 3.0000
 15.0 1.0 1.0 1.0
 3.000000      8.000000
 1.000000      73.00000      74.00000      224.0000      225.0000
 365.0000
 1.000000      1.000000      1.000000      1.000000      1.000000
 1.000000
0.0000000E+00  0.0000000E+00
3.00000 3.00000
 1.000000      2.000000      3.000000
 2.0
 11.00000      12.00000
 4.000000      1.000000
0.5000000      0.5000000
 3.0
 8.000000      11.00000      18.00000
 1.000000      4.000000      2.000000
0.3300000      0.3300000      0.3400000
 2.0
 1.000000      11.00000
 5.000000      4.000000
0.7800000      0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.0
0.0000000
0.0000000
0.000000
2.0
6.00000e-02
6.00000e-02
0.0 0.0 0.0
0.0
1.0
50.00000      0.90          2.000000
1.0
1.0
1.0
1.0
15.0 1.0 1.0 1.0
 3.000000      8.000000
 1.000000      365.0000
 1.000000    1.00000
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
 0.0 0.0
 1.0 1.0
 1.0
 2.0
 1.000000      11.00000
 5.000000      4.000000
0.7800000      0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.000
0.000000
0.0000000
0.0
2.0
6.000e-02
6.000e-02
   999999.0
   1.0 1.0
   1.0
   1.0
   89.7
   89.7
   0.95
   0.00
   1.0
   1.0
   89.7
   89.7
   0.95
   0.0
   6.91 0 0 0 0 0
   1.0
   85.0 1.0
   1.0
   1.0
   1.0
   1.0
   0.0 1.0 1.0 1.0
   4.0 30.0
   1.0 365.0
   1.0 1.0
   1.0
   3.0
   2.0 12.0 23.0
   4.0 5.0 7.0
   0.33 0.33 0.34
   2.0E-02
   0.231
   0.0
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
    0.02
    -1.0
    0.0
    0.0
    0.0
    1.0
    85.0 1.0
    1.0
    1.0
    1.0
    1.0
    0.0 1.0 1.0 1.0
    4.0 30.0
    1.0 365.0
    1.0 1.0
    1.0
    3.0
    2.0 12.0 23.0
    4.0 5.0 7.0
    0.33 0.33 0.34
    2.0E-02
    0.231
    0.0
    0.02
    -1.0
    0.0
    0.0
    0.0
BIOPLT Subroutine
    444444.0
   1.000000
  0.0000000E+00   0.000000      0.0000000E+00
   1.000000       1.000000
ZLVI
3.   26.
kr      2       m
-25.0  -1.0  -1.
ZGRNI
3.   26.
kr      2       m
-25.0  -1.0  -1.
ZSVI
3.   26.
kr      2       m
-25.0  -1.0  -1.
ZMEATI
3.   26.
kr      2       m
-25.0  -1.0  -1.
RKD
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
3.   18.
kr      ko      m
-25.0  -3.0  -1.
RBN
3.   18.
kr      ko      m
-25.0  -3.0  -1.
RLV
3.   18.
kr      ko      m
-25.0  -3.0  -1.
RLU
3.   18.
kr      ko      m
-25.0  -3.0  -1.
RBN
3.   18.
kr      ko      m
-25.0  -3.0  -1.0
TPOP
3.   20.
kr      2       ko
-25.0  -1.0  -3.0
  4.000000
  2.00000
TRUPLT Subroutine
  171717.
  4.000000
  4*1
  13.00000
  1.000000      2.000000      3.000000      4.000000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
TPLOTR Subroutine
 191919.
        4        7        37
        1        1        1        1
y
y
y
n
year
pCi/day
U-238 Intake by 25 Yr Old Human Males
leafy veg
grains
seed veg
beef
 -111.0000     -111.0000
```

Table 16.28., Plants, Cattle, and Human Input (continued)

```
 -111.0000     -111.0000
TRUPLT Subroutine
171717.
5.0
1.0 1.0 1.0 1.0 1.0
13.0
5.0 6.0 7.0 8.0 9.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
TPLOTR Subroutine
191919.
4 4 45
1 1 1 1 1
y
y
y
n
year
Rems
U-238 Radiation Dose For 25 Yr Old Human Males
kidneys
bone
liver
lungs
muscle
-111. -111.
-111. -111.
```

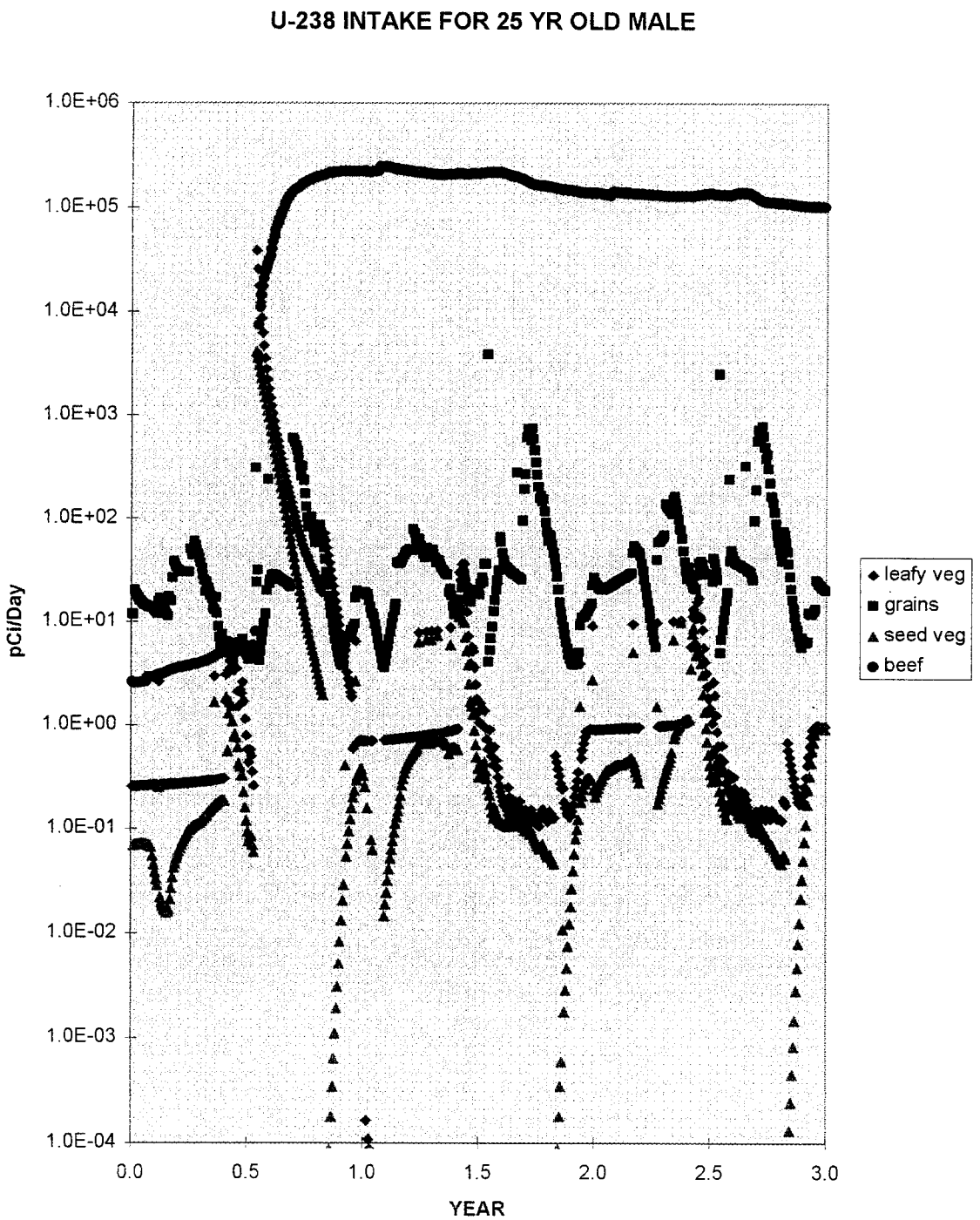


Figure 16.36.

Uranium Uptake From Various Foods by 25 Year Old Males.

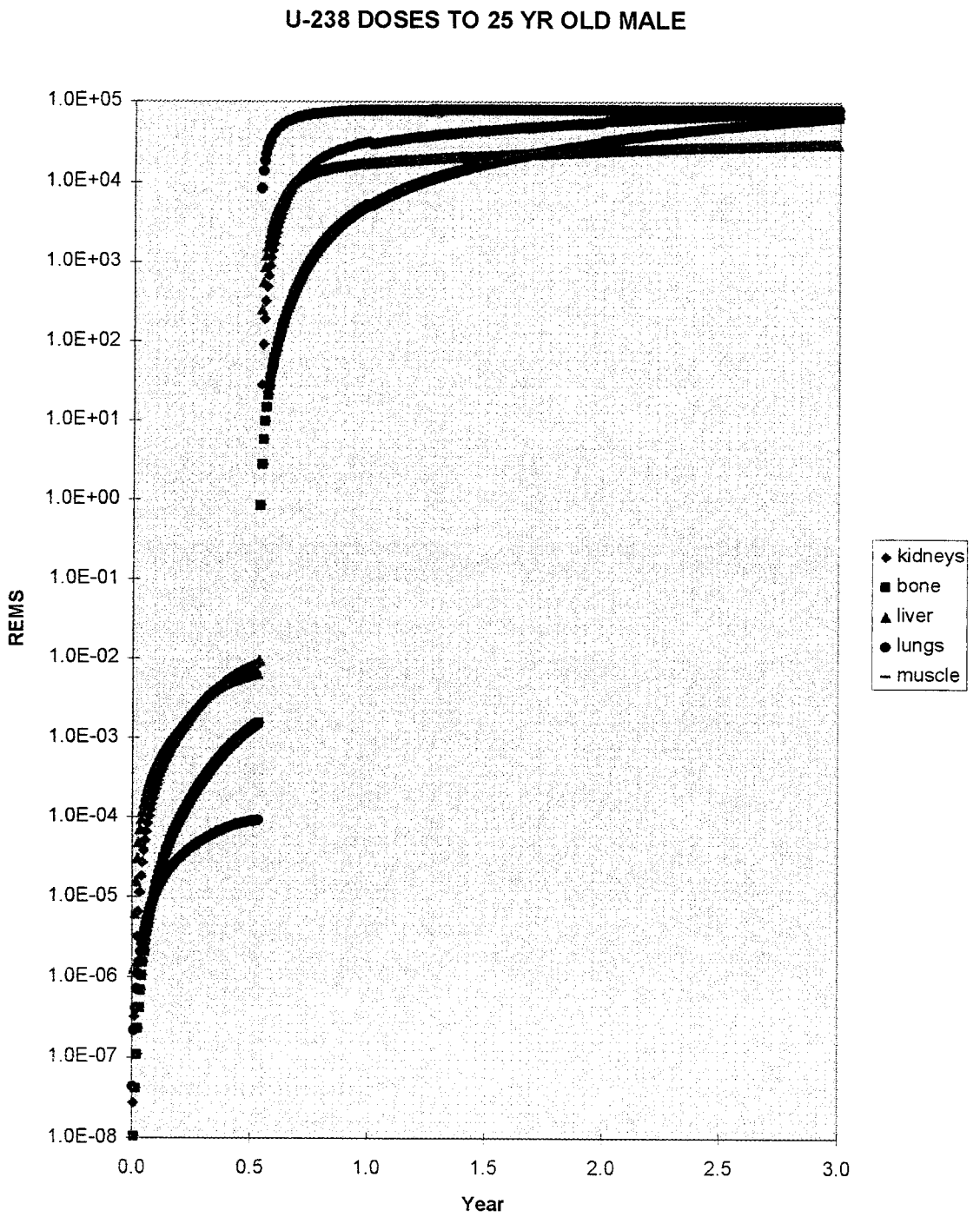


Figure 16.37.

Radiation Doses From U-238 to Various Organs and Tissues of 25 Year Old Males

16.13.3. References

Adams, C.F., 1975. Nutritive value of American foods in common units. Agricultural Handbook 456. USDA, Washington DC.

Brody, S., 1945. Bioenergetics and growth. Reinhold Publishing Corporation, New York. 1023pp.

Buhl, T.E., and W.R. Hansen. 1984. Estimating the risks of cancer mortality and genetic defects resulting from exposures to low levels of ionizing radiation. Los Alamos National Laboratory, Los Alamos, NM. LA-9893-MS.

Gallegos, A.F., and W.J. Wenzel. 1984. HUMTRN: ICRP based age-and-sex specific human simulation model for radionuclide dose assessment. Los Alamos National Laboratory, Los Alamos, NM. LA-9994-MS.

Gallegos, A.F. and W.J. Wenzel. 1989. HUMTRN and EFFECTS: Age and sex specific dosimetric and physiological human population dynamics models for dose assessment. Health Physics (57). 443-454.

International Commission on Radiological Protection. 1975. Report of the Task Force on Reference Man. ICRP-23. Pergamon Press.

Wenzel, W.J., and A.F. Gallegos. 1985. EFFECTS: Documentation and verification for a BEIR III cancer risk model based on age, sex, and population dynamics for BIOTRAN. Los Alamos National Laboratory, Los Alamos, NM. La-10371-MS.

16.14. ZOOGRO Subroutine - Grazing, Harvest, Feed, and Food Storage

16.14.1. Description

The ZOOGRO subroutine is illustrated in Figure 16.38.. ZOOGRO is called from MAIN, and calls both ANIMAL and HUMTRN on daily basis and by age group and sex (humans). The invertebrate subroutine INVERT is called from ANIMAL when they are being simulated. ZOOGRO simulates the daily forage, feed, or food intake according to the diet schedules specified by the user in the ANIMAL, and HUMTRN input. This subroutine rations the food and accounts for the losses from the feed or storage compartments (bins). Grazing preferences allows selective grazing to occur when more than one plant is specified for consumption on range.

Harvests are simulated in ZOOGRO according to harvest schedules input into PLTGRO. Contaminants on the surfaces of crops, feeds, and range plants as well as internal contamination are integrated and summed in ZOOGRO prior to passage to ANIMAL or HUMTRN subroutines.

16.14.2. Input/Output

There are no input requirements to ZOOGRO. ZOOGRO output involving harvested quantities, specific activities of contaminants, and nutrient contents are available through the use of subroutine BIOPLT. The input for **example 1** presented in Table 16.29. is identical to that in Table 16.25. for beef and dairy cow simulations except for input requirements in BIOPLT, TRUPLT, and TPLOTR subroutines. The harvest amounts for beef and dairy cattle supplementary feeds are presented in Figures 16.39. and 16.40., respectively. Note the gradual decline of feed as it is consumed, and the renewal at each harvest. This scenario clearly shows that feed supplies are more than adequate for the number of animal units simulated in each case (99 beef cows and 50 dairy cows). Contaminant concentrations in and on the plant components were presented in Figures 16.30. and 16.31. a sum total.

Table 16.29.

Amarillo, Texas Crops ZOOGRO Input

```
MAIN subroutine
6.000000
 1.000000
 1.000000
 1.000000     1.000000     0.0000000E+00   1.000000
 7.000000
 1.000000
 1.000000
0.0000000E+00
0.0000000E+00
2.0 1.0
AIRAC Subroutine
   111111
 1.000000     0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00
0.0000000E+00
 1.000000
 1.000000
 3.000000
 195.0000
1.000000e+08
0.000000
CLIMAT Subroutine
   555555
 277.0000     625.0000
 90.00000
 36.20000     78.00000
 120.0000
 4.360000
0.2000000
 3500.000
4.0000002E-03
 4.000000     4.000000     4.000000     4.000000     4.000000
 4.000000     4.000000     4.000000     4.000000     4.000000
 4.000000     4.000000
2.3000000E-02  2.6000001E-02  3.7000000E-02  6.1999999E-02  0.1490000
0.1610000     0.1430000     0.1500000     9.3000002E-02  8.9000002E-02
3.9000001E-02  2.8999999E-02
 22.00000
 3538.000
0.0000000E+00
 202.0000
 36.00000
WATFLX Subroutine, Part I
   141414
0.0000000E+00  0.0000000E+00
0.0000000E+00
0.0000000E+00
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
0.0000000E+00
 0.0000000E+00
WATLL Subroutine, Part II
   151515
 92.00000     7.000000     7.000000     15.00000     15.00000
 19.00000
 238.0000     14.00000     14.00000     31.00000     31.00000
 39.10000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000     1.000000
 2.000000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 5.000000     5.000000     6.000000     7.000000     6.000000
 6.000000     5.000000
 990.6000     990.6000     1498.600     2006.600     1498.600
 1498.600     990.6000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4020000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
0.3300000     0.3300000     0.4700000     0.4780000     0.4780000
 18.70000     18.70000     18.90000     17.90000     16.50000
 30.00000     30.00000     30.10000     27.20000     27.50000
 28.50000     28.50000     28.60000     25.80000     26.10000
 2.000000     2.000000     2.000000     2.000000     2.000000
 20.00000     132.4000     203.2000     304.8000     330.2000
 508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.1800000     0.1800000     0.1200000     0.1200000     0.1200000
0.1200000
0.4900000     0.4900000     0.4100000     0.4020000     0.4030000
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000
  28.50000     28.50000     28.60000     25.80000     26.10000
  24.50000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
  508.0000     508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00
 0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
 0.1200000    0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.4200000    0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000    0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000     15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000     25.80000
  28.50000     28.50000     28.60000     25.80000     26.10000
  24.50000     24.50000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000     2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
  508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
 0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000
  28.50000     28.50000     28.60000     25.80000     26.10000
  24.50000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
  508.0000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.4200000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
 0.4590000
  18.70000     18.70000     18.90000     17.90000     16.50000
  15.00000
  30.00000     30.00000     30.10000     27.20000     27.50000
  25.80000
  28.50000     28.50000     28.50000     28.60000     25.80000
  26.10000
  2.000000     2.000000     2.000000     2.000000     2.000000
  2.000000
  20.00000     132.4000     203.2000     304.8000     330.2000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.1800000    0.1800000    0.1200000    0.1200000    0.1200000
 0.4900000    0.4900000    0.4100000    0.4020000    0.4030000
 0.3300000    0.3300000    0.4700000    0.4780000    0.4780000
  18.70000     18.70000     18.90000     17.90000     16.50000
  30.00000     30.00000     30.10000     27.20000     27.50000
  28.50000     28.50000     28.60000     25.80000     26.10000
  2.000000     2.000000     2.000000     2.000000     2.000000
 8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
 8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
 8.8000000E-02 0.6200000    0.2000000    2.1000000E-02 6.1000001E-02
 6.3000000E-03
 5.0099999E-02 0.5100000    0.3300000    0.1600000    7.9999998E-02
 2.4000000E-02 7.8999996E-03
 8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
 6.3000000E-03
 8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
 6.3000000E-03
 8.8000000E-02 0.6200000    0.2000000    6.1000001E-02 2.1000000E-02
PLTGRO Subroutine
   121212
 0.0000000E+00
  1.000000
 0.0000000E+00
  1.000000     2.000000     8.000000     11.00000     12.00000
  18.00000     23.00000
  6.000000     17.00000     0.0000000E+00 10.00000     5.000000
 0.0000000E+00 17.00000
  2.000000     2.000000     0.0000000E+00 4.000000     2.000000
 0.0000000E+00 2.000000
  1.000000     1.000000     1.000000     1.000000     1.000000
  1.000000     1.000000
  1.000000     1.000000     0.0000000E+00 1.000000     1.000000
 0.0000000E+00 1.000000
 0.0000000E+00 0.0000000E+00 1.000000     0.0000000E+00 0.0000000E+00
  1.000000     0.0000000E+00
  4000.000     4000.000     4000.000     4000.000     4000.000
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
4000.000      4000.000
 302.0000      302.0000
0.0000000E+00  7.000000      6.000000      6.000000      6.000000
 6.000000
 150.0000      150.0000      201.0000      229.0000      243.0000
 257.0000
 302.0000      302.0000
0.0000000E+00  1.000000      1.000000      1.000000      1.000000
 1.000000      1.000000      1.000000      1.000000      1.000000
 2.000000      2.000000      2.000000      2.000000      2.000000
 2.000000      2.000000
 150.0000      150.0000      157.0000      164.0000      171.0000
 177.0000      184.0000      191.0000      198.0000      205.0000
 212.0000      219.0000      226.0000      233.0000      240.0000
 247.0000      254.0000
 1.000000      500.0000      531.0000      577.0000
0.0000000E+00  7.000000      5.000000      5.000000      5.000000
 5.000000      5.000000      5.000000      5.000000      5.000000
 215.0000      215.0000      470.0000      485.0000      502.0000
 514.0000      527.0000      533.0000      544.0000      559.0000
 1.000000      560.0000
0.0000000E+00  16.00000      5.000000      5.000000      5.000000
 250.0000      250.0000      285.0000      425.0000      500.0000
 302.0000      302.0000
0.0000000E+00  7.000000      2.000000      2.000000      2.000000
 2.000000      2.000000      2.000000      2.000000      2.000000
 3.000000      3.000000      3.000000      3.000000      3.000000
 3.000000      3.000000
 150.0000      150.0000      157.0000      164.0000      171.0000
 177.0000      184.0000      191.0000      198.0000      205.0000
 212.0000      219.0000      226.0000      233.0000      240.0000
 247.0000      254.0000
UPTAKE Subroutine
   131313
0
ANIMAL Subroutine
   222222
 2
550.0000      550.00000
 6.91 6.91
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 0.0000 0.0000
 3.000000      1.000000
 15.000      15.000
 0.0 0.0 0.0
 0.0
 1.0
 50. 0.90 2.0
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
 1.000000
 1.000000
 1.000000
 3.0000
 15.0 1.0 1.0 1.0
 3.000000     8.000000
 1.000000     73.00000     74.00000     224.0000     225.0000
 365.0000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00  0.0000000E+00
3.00000 3.00000
 1.000000     2.000000     3.000000
 2.0
 11.00000     12.00000
 4.000000     1.000000
0.5000000     0.5000000
 3.0
 8.000000     11.00000     18.00000
 1.000000     4.000000     2.000000
0.3300000     0.3300000     0.3400000
 2.0
 1.000000     11.00000
 5.000000     4.000000
0.7800000     0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.0
0.0000000
0.0000000
0.000000
2.0
6.00000e-02
6.00000e-02
0.0 0.0 0.0
0.0
1.0
50.00000     0.90     2.000000
1.0
1.0
1.0
1.0
15.0 1.0 1.0 1.0
 3.000000     8.000000
 1.000000     365.0000
 1.000000    1.00000
 0.0 0.0
 1.0 1.0
 1.0
 2.0
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
1.000000      11.00000
 5.000000      4.000000
0.7800000     0.2200000
2.0000001E-02
0.231
0.0
0.02
-1.000
0.000000
0.0000000
0.0
2.0
6.000e-02
6.000e-02
BIOPLT Subroutine
 444444.
1.0
0.0 1.0 1.0
1.0 1.0
HRVST
3.0  8.0
ko      i       n
-1.0  -1.0 -1.0
HRVST
3.0  8.0
ko      i       n
-1.0  -4.0 -1.0
HRVST
3.0  8.0
ko      i       n
-2.0  -1.0 -1.0
HRVST
3.0  8.0
ko      i       n
-2.0  -4.0 -1.0
VSOL
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VSOL
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
VPLN
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VPLN
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
VGUT
3.0 22.0
kr      ko      m
-2.0  -1.0 -1.0
VGUT
3.0 22.0
kr      ko      m
-2.0  -2.0 -1.0
3.0
2.0
TRUPLT Subroutine
171717
2.0
1.0 1.0
13.0
1.0 2.0
0.0 0.0
0.0 0.0
0.0 0.0
TPLOTR Subroutine
191919
4 5 26
1 1
y
y
y
n
year
gdwt
CROP HARVEST FOR BEEF COWS
sorghum
alfalfa
-111.0 -111.0
-111.0 -111.0
TRUPLT Subroutine
171717
2.0
1.0 1.0
13.0
1.0 2.0
0.0 0.0
0.0 0.0
0.0 0.0
TPLOTR Subroutine
191919
4 5 27
1 1
y
y
y
n
```

Table 16.29., Amarillo, Texas ZOOGRO Input (continued)

```
year
gdwt
CROP HARVEST FOR DAIRY COWS
sorghum
alfalfa
-111.0 -111.0
-111.0 -111.0
```

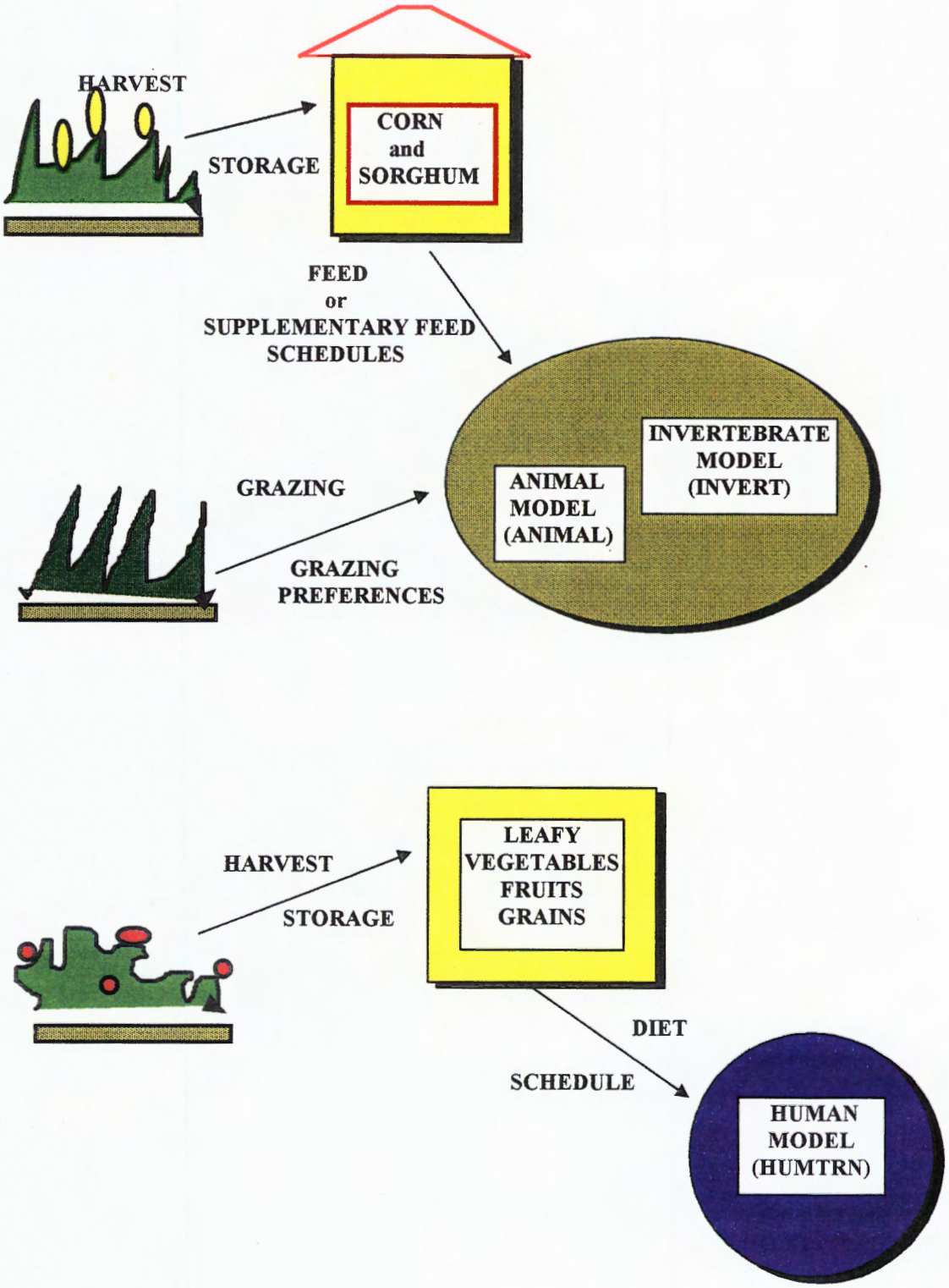


Figure 16.39.

ZOOGRO Subroutine Simulates Daily forage, feed, or food to ANIMAL, INVERT, and HUMTRN According to Diet Schedules Supplied By the User

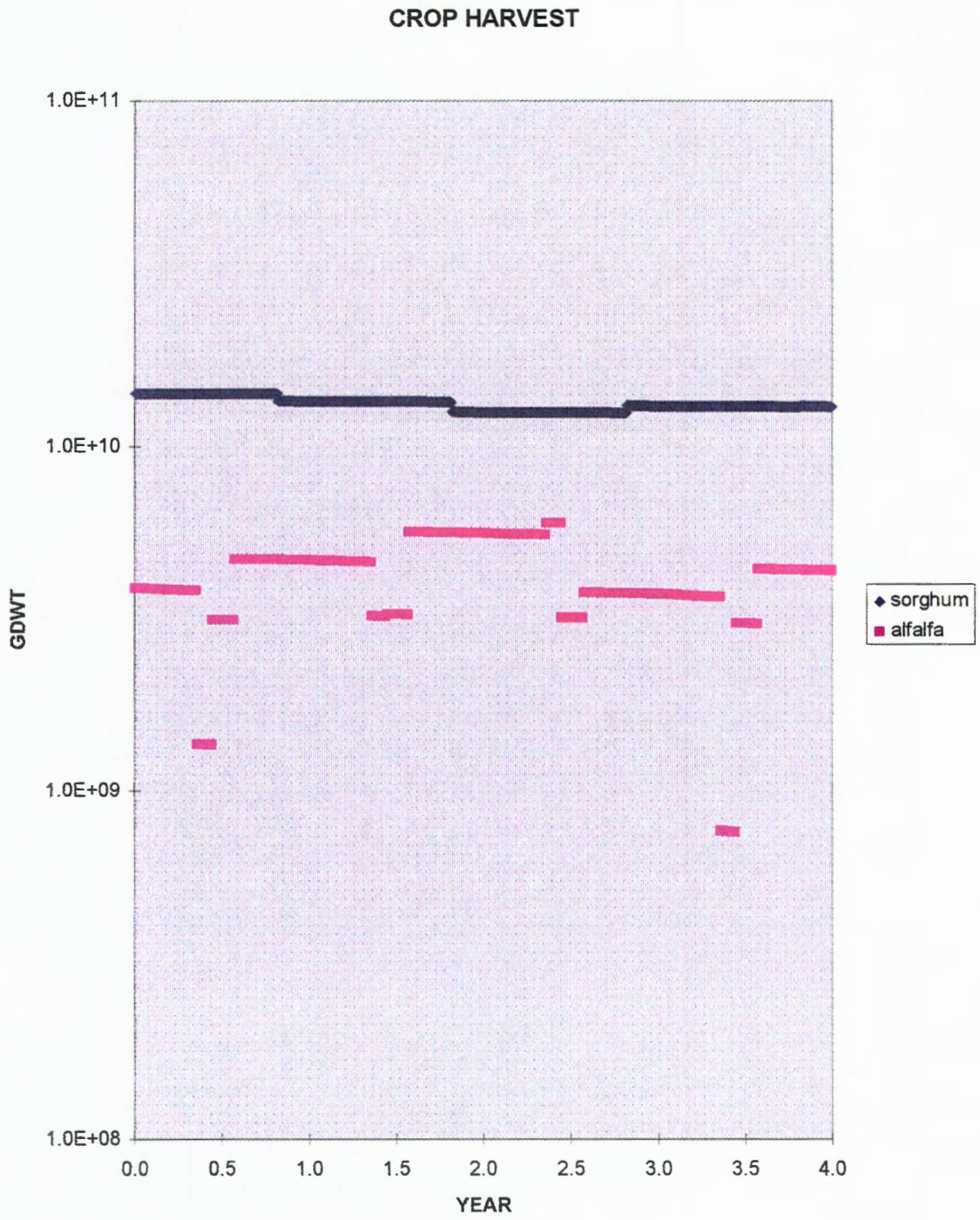


Figure 16.39.

Crop Harvests For Beef Cattle in the Amarillo, Texas Scenario

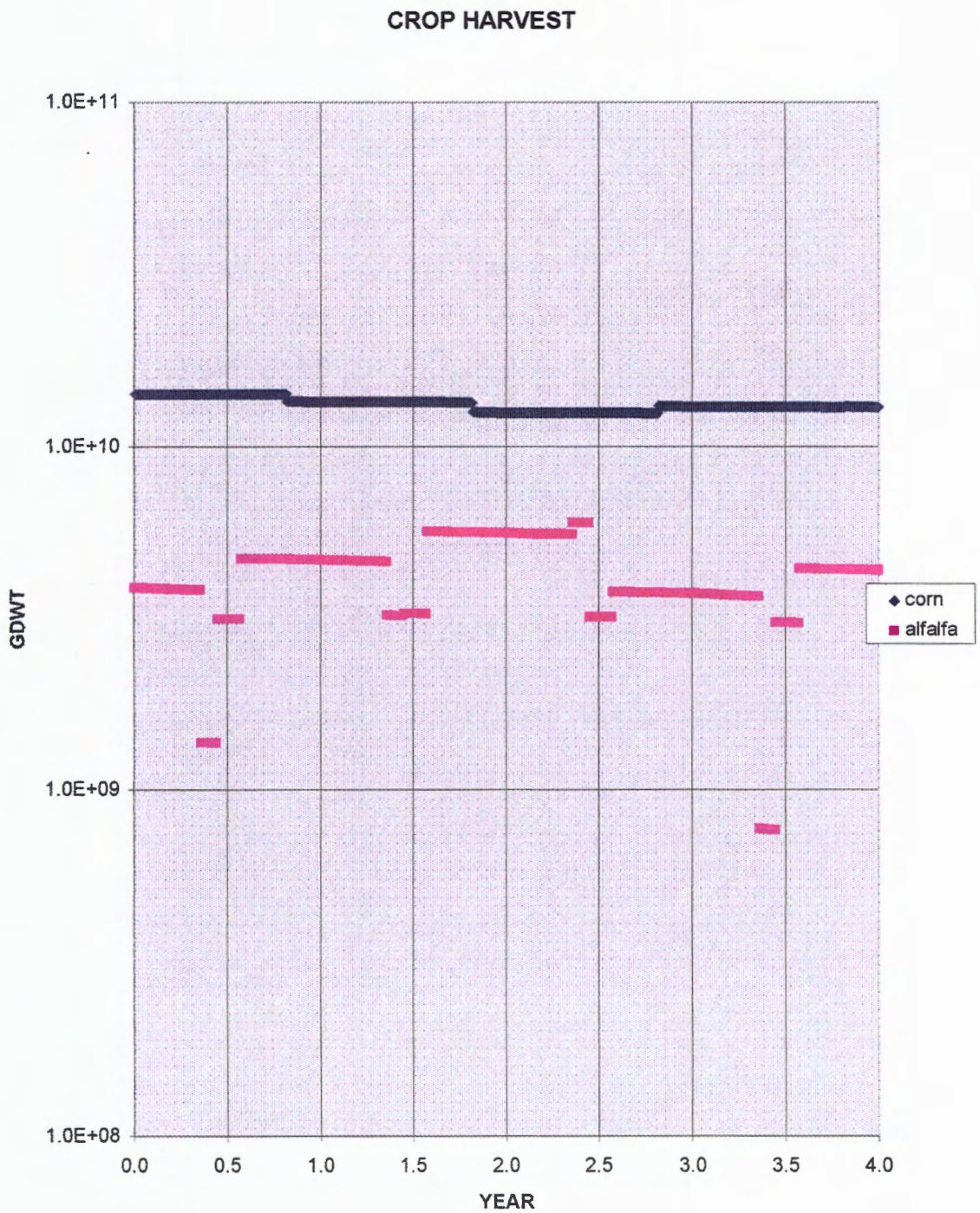


Figure 16.40.

Crop Harvests for Dairy Cattle in the Amarillo, Texas Scenario

16.15. FORMAN Subroutine - Forest Growth Model

16.15.1. Description

The FORMAN subroutine and its companion FORCUT are used mainly for the simulation of forest growth where the tree or shrub biomass density is over 1000 gdwt /m^2 . Also, the ground cover algorithm used for most plants in the PLTGRO subroutine become inaccurate at densities above the latter, and must be estimated in subroutine FORMAN. This subroutine is primarily designed to simulate forest structure by bole (stem) diameter classifications for output purposes, however, the output of contaminant concentrations in wood and leaves of trees is also of great importance. Entry into FORMAN is accomplished by specification of the variable RGC in the PLTGRO subroutine: a value of (1.0) does not permit access, a value of (2.0) permits access yearly, and a value of (3.0) permits access on a daily basis. The latter specification (3.0) is seldom used except where the structure of an herbaceous perennial or annual crop such as corn is desired, but proves wasteful if employed for trees. Shrub, grass, forbs, and crops do not ordinarily use FORMAN, but can be used to determine stem diameters, crown cover, biomass volumes, etc. in special applications. Forman and FORCUT were developed mostly from biomass density and forest mensuration data obtained from studies in New Mexico and Arizona (Gallegos 1978 and Garcia 1977).

FORMAN simulates tree stands. A stand may consist of more than one plant type: trees, shrubs, and grasses coexisting in one stand (with the area set in acres by PLTGRO) require both succession and niche competition strategies included in PLTGRO as part of the simulation. Niche competition places similar type plants into competition for resources (mainly water use) on a daily basis, whereas, succession places all plants into resource competition on a yearly basis. Thus, plant communities can be realistically simulated through the use of these strategies in conjunction with FORMAN and FORCUT subroutines. Example 2 for the Los Alamos area is used to illustrate succession, niche competition, and the FORMAN and FORCUT subroutines. The latter example is primarily and extension of earlier work by Rodgers and Walker (Walker 1981), and Gallegos and Smith (Gallegos 1978) with the addition of newer data from Tierney (1982), Abeele (1981), and Foxx (1984).

Tree Biomass is partitioned into stem (bole) diameter classes and growth is incremented within diameter classes as tree rings that are based on internally generated competition factors, but which interact with both succession and niche competition strategies in PLTGRO. Contaminants are also partitioned into biomass compartments in a similar fashion to that employed in the UPTAKE subroutine. FORCUT, the forest management and fire event simulator is called from FORMAN as an option depending on the users input specifications. A typical mixed stand of trees, shrubs, and grasses is shown in Figure 16.41..

16.15.2. Input / Output

Input to the FORMAN subroutine is relatively easy to set up, but requires tree specific data in order to operate. Three basic variables are required for any tree type simulated: C6, C9, and V8. The input variable C6 is defined as the linear regression coefficient for crown diameter (feet) to the bole diameter (feet). The latter is taken at breast height (DBH) for all trees except Pinyon and Juniper which was taken 1 foot above the ground. The input variable V8 is defined as the curvilinear regression coefficient (data forced through the origin) for bole diameter squared (cm^2), to the total tree biomass (kgdwt). Finally, C9 is defined as the specific gravity, gdwt /cm^3, of the wood component. Values of C6, C9, and V8 for selected trees are presented in Table 16.30. Note that the user specifies the time loop for FORMAN through the use of RGC in the PLTGRO subroutine. When RGC=1, FORMAN is not accessed for a given plant type, whereas, when RGC=2 or 3, this subroutine is accessed yearly or daily, respectively, as described earlier. This is clearly illustrated in MAIN when FORMAN is called or not called depending on the value of RGC for the plant types included in the simulation. FORMAN updates the major PLTGRO biomass compartments and returns these values back to MAIN for the tree types specified.

A description of input requirements for FORMAN is presented in Table 16.31.. The lines are repeated for each plant type in the simulation using FORMAN. Values for C6, C9, and V8 are taken from Table 16.30.. Input for Example 2 which simulates Pinyon Pine (PLTID=5) exclusively is presented in Table 16.32. and the simulated wood volume, VTMBR, in ft^3 / acre for several diameter classes is presented in Figure 16.42.. Input for a succession simulation involving grasses, shrubs, Pinyon-Juniper, and Ponderosa Pine (PLTID=15) in an ecotone setting is presented in Table 16.33. Output for the latter is presented in Figure 16.43. and shows the rapid replacement of plant growth by trees once they have become established. Also, this example shows that Ponderosa Pine shows greater dominance than Pinyon-Juniper for the particular site simulated. A study has been initiated to help verify the replacement process in both time and space.

The final example shown in Figure 16.44. depicts the growth of timber volume by diameter classification for Ponderosa Pine in the mixed conifer setting or ecotone. Input for this simulation is also taken from Table 16.33. with the only difference being the output instructions for these variables.

16.15.3. References

Abeele, W.V., Wheeler, M.L., and B.W. Burton. 1981. Geohydrology of Bandelier Tuff. Los Alamos National Laboratory, Los Alamos, NM. LA-8962-MS.

Foxx, T.S., and G.D. Tierney. 1984. Status of the flora of the Los Alamos National Laboratory Research Park, a historical perspective. Los Alamos National Laboratory, Los Alamos, NM. LA-8050-NERP, Vol II.

Gallegos, A.F., Smith, W.J., and L.J. Johnson. 1978. LASL models for environmental transport of radionuclides in forests. Los Alamos National Laboratory, Los Alamos , NM. LA-UR-78-1097.

Garcia, B.J. 1977. Selected characteristics of Pinon Pine. Masters Thesis. New Mexico Highlands University, Las Vegas, NM.

Tierney, G.D., and T.S. Foxx. 1982. Floristic composition and plant succession on near-surface radioactive waste disposal facilities in the Los Alamos National Laboratory. Los Alamos National Laboratory, Los Alamos, NM. LA-9219-MS.

Walker, L.W., Hansen, W.R., Wilson, D.W., Maestas, G., Wenzel, W.J., and F.A. Guevarra. 1981. Alternative transuranic waste management strategies at Los Alamos National Laboratory. Los Alamos National Laboratory, Los Alamos, NM. LA-8982-MS.

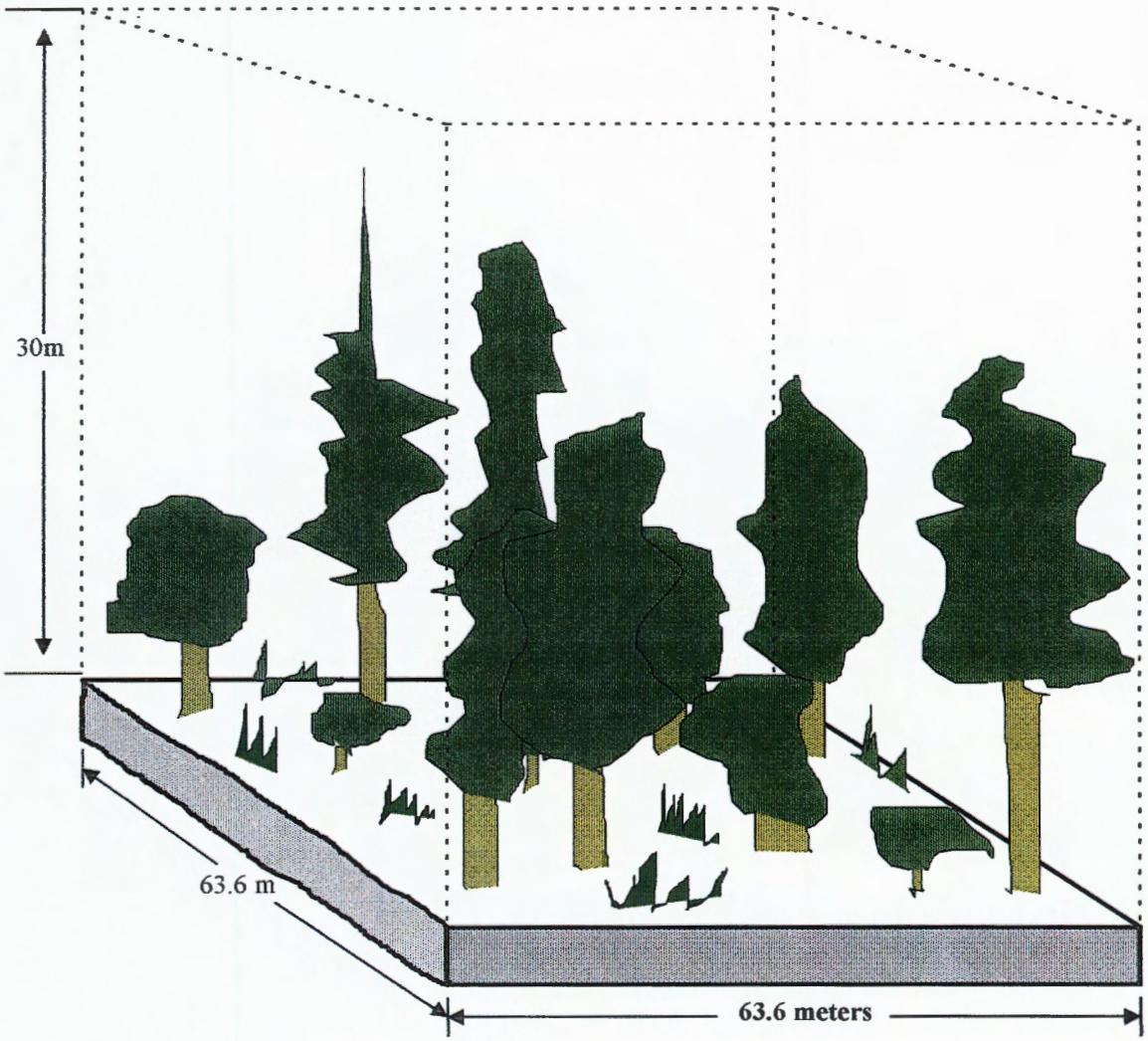


Figure 16.41.

FORMAN Subroutine Manages the Growth and Contaminant Transport for Forest Stands Based on Selections Made in PLTGRO

Table 16.30.

Specific Gravity And Regression Coefficients For Selected Trees Simulated in FORMAN

PLTID	Species	Regression Coefficients		
		Specific Gravity C9	Crown to Bole C6	Bole to Biomass V8
5	**Pinus edulis** (Pinyon Pine)	0.54	1.5	0.19
15	**Pinus ponderosa** (Ponderosa Pine)	0.39	1.0	0.30
17	**Abies concolor** (White Fir)	0.39	0.75*	0.31
14	**Psuedotsuga menziesii** (Douglas Fir)	0.46	0.76*	0.32
21	**Populus tremuloides** (Quaking Aspen)	0.41	0.95	0.28
5	**Juniperus monosperma** (juniper)	0.30	1.5	0.19*
10	**Quercus alba** (White Oak)	----	1.8	----

* Not field validated

Table 16.15.2.

FORMAN Subroutine Input

Input Line	Variable Name	Definition
1.		*Enter 777777 to access FORMAN*
2.	TBRM	Enter the number of timber cuttings or forest fire(s) for a given plant type in a given region. Also serves as a gate for entering subroutine FORCUT(0.0 = no access).
3.	XJI	Enter diameter class spacing , inches for a given plant in a given region. A value of 2.0 will simulate diameter classes as 1-2, 2-4, 4-6, ..etc. in inches. This input value (2.0) is suggested, because this diameter classification was used in developing the model from live forest stands.
4.	V8	Enter the curvilinear regression coefficient relating the mass a given tree type, kgfwt, to the square of the diameter at DBH (diameter at breast height), cm^2. For Pinyon and Juniper trees, the bole diameter measurement is made at 1 foot above the ground, all other types at DBH. The regression coefficient has been forced through zero. See Table 16.30..
5.	C9	Enter the specific gravity of oven-dried wood for a given plant type.
6.	C6	Enter the linear regression coefficient relating crown diameter, feet, to bole diameter, inches. the bole diameters are measured as described for V8 above.
7.	TNTR	Enter the initial tree density, trees /acre, for the stand of a given plant type. A value of (0.0) will default to natural growth, all other values refer to an even-aged stand.
8.	YDOS	Enter the number of years over which effects of a given radionuclide will be performed if desired by the user. A value of (0.0) is entered for non-radionuclides, or if the user does not wish to simulate effects of radiation (fatal cancer production) on the specific human male population

If TBRM has been set equal to (0.0), then subroutine FORCUT is not called, otherwise the user should be prepared to enter forest management and /or forest fire options. Input to FORCUT is described in the next section.

Repeat lines 2-8 for each plant where RGC ≥2.0 in a specific region

Repeat lines 1-8 for each region simulated if RGC ≥2.0 for any plant type in that region

Table 16.32.

LANL Pinyon Pine Input

MAIN Subroutine

```
100.00000
  1.000000
  1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
  1.000000
 0.0000000E+00
  1.000000
  1.000000
 0.0000000E+00
  365.0000      1.000000
```

CLIMAT Subroutine

```
  555555.0
  270.0000      723.0000
  90.00000
  29.90000      70.40000
  120.0000
  2.000000
 0.2000000
  6990.000
 5.4000001E-03
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000
 5.0000001E-02  5.0000001E-02  5.6000002E-02  6.1000001E-02  9.7999997E-02
 8.5000001E-02  0.1570000     0.1640000     0.1040000     7.9999998E-02
 4.3000001E-02  5.0999999E-02
  16.00000
  6800.000
 0.0000000E+00
  138.0000
  37.00000
```

WATFLX Subroutine, Part I

```
  141414
 0.0000000E+00  0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.1000000
```

WATFLX Subroutine, Part II

```
  151515.0
  94.00000      7.000000      7.000000      15.00000      15.00000
  19.00000
  239.0000      14.00000      14.00000      31.00000      31.00000
  39.00000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
  1.000000
  2.000000      0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
```

Table 16.32., LANL Pinyon Pine INPUT (continued)

```
0.0000000E+00
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
 1.000000     1.000000     1.000000     1.000000     1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 10.00000
 2775.000
 20.00000     60.00000     130.0000     50.00000     250.0000
 250.0000     250.0000     250.0000     250.0000     1265.000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.5500000     0.5500000     0.3300000     0.3300000     0.9000000
0.9000000     0.9000000     0.9000000     0.9000000     0.9000000
0.3000000     0.3000000     0.2200000     0.2200000     0.1000000
0.1000000     0.1000000     0.1000000     0.1000000     0.1000000
0.2000000     0.2000000     0.4500000     0.4500000     1.0000000E-03
1.0000000E-03 1.0000000E-03 1.0000000E-03 1.0000000E-03 1.0000000E-03
 10.00000     10.00000     10.00000     10.00000     10.00000
 10.00000     10.00000     10.00000     10.00000     10.00000
 35.00000     35.00000     35.00000     35.00000     31.00000
 31.00000     31.00000     31.00000     31.00000     31.00000
 28.00000     28.00000     28.00000     28.00000     29.50000
 29.50000     29.50000     29.50000     29.50000     29.50000
 1000.000     0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
8.2000002E-02 0.2200000     0.4600000     3.9999999E-02 0.2000000
5.7000000E-02 1.7999999E-02 5.5000000E-03 1.7000000E-03 9.8999997E-05
PLIGRO Subroutine
 121212.0
0.0000000E+00
0.0000000E+00
0.0000000E+00
 5.000000
 0.000000
 0.000000
 2.000000
0.0000000E+00
0.0000000E+00
 200.0000
UPIAXE Subroutine
 131313.0
      0
FORMAN Subroutine
 777777.0
0.0000000E+00
 2.000000
 0.1900000
```

Table 16.32., LANL Pinyon Pine INPUT (continued)

```
 0.5800000
  1.550000
 0.0000000E+00
BIOPLT Subroutine
  444444.0
  1.000000
 0.0000000E+00  1.000000     1.000000
  1.000000     1.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -1.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -2.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -3.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -4.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -5.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -6.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -7.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 0.0000000E+00  -8.000000     0.0000000E+00
B1
2.  2.
i      n
 0.0000000E+00  0.0000000E+00
BAR
2.  2.
kt      n
 -1.000000     0.0000000E+00
  1.000000
  1.000000
```

Table 16.32., LANL Pinyon Pine INPUT (continued)

```
IRUPLI Subroutine
 171717.
  8.000000
  8*1.0
  13.00000
  1.000000     2.000000     3.000000     4.000000     5.000000
  6.000000     7.000000     8.000000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00
IPLOIR Subroutine
 191919.
 5        8        19
 8*1
y
y
y
n
years
ft3/acre
LANL Timber Volume
diam 1
diam 3
diam 5
diam 7
diam 9
diam 11
diam 13
diam 15
 -111.0000    -111.0000
 -111.0000    -111.0000
```

Table 16.33.

Succession Input for Grasses, Shrubs, and Trees (Pinyon and Ponderosa) in an Ecotone Setting For Los Alamos

```
MAIN Subroutine
100.00000
  1.000000
  1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
  5.000000
 0.0000000E+00
  1.000000
  1.000000
 0.0000000E+00
  365.0000      1.000000
CLIMAT Subroutine
  555555.0
  270.0000      723.0000
  90.00000
  29.90000      70.40000
  120.0000
  2.000000
 0.2000000
  6990.000
 5.4000001E-03
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000
 5.0000001E-02  5.0000001E-02  5.6000002E-02  6.1000001E-02  9.7999997E-02
 8.5000001E-02  0.1570000     0.1630000     0.1040000     7.9999998E-02
 4.3000001E-02  5.0999999E-02
  16.00000
  6800.000
 0.0000000E+00
  138.0000
  37.00000
WATFLX Subroutine, Part I
  141414.0
 0.0000000E+00  0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.1
WATFLX Subroutine, Part II
  151515.0
  94.00000      7.000000      7.000000      15.00000      15.00000
  19.00000
  239.0000      14.00000      14.00000      31.00000      31.00000
  39.10000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
6*1.0
```

Table 16.33., Succession Input for Grasses, Shrubs, and Trees (continued)

```
2.000000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00
  1.000000    1.000000    1.000000    1.000000    1.000000
  1.000000
  1.000000    1.000000    1.000000    1.000000    1.000000
  1.000000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00
  10.00000    10.00000    10.00000    10.00000   10.000
  9144.000    9144.000    9144.000    9144.000   9144.0
  20.00000    60.00000    130.0000    50.00000    250.0000
  250.0000    250.0000    250.0000    250.0000    7634.000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
 0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
 0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
 0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
 0.2000000    0.2000000    0.4500000    0.4500000    0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
  10.00000    10.00000    10.00000    10.00000    10.00000
  10.00000    10.00000    10.00000    10.00000    10.00000
  35.00000    35.00000    35.00000    35.00000    31.00000
  31.00000    31.00000    31.00000    31.00000    31.00000
  28.00000    28.00000    28.00000    28.00000    29.50000
  29.50000    29.50000    29.50000    29.50000    29.50000
  1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
  20.00000    60.00000    130.0000    50.00000    250.0000
  250.0000    250.0000    250.0000    250.0000    7634.000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
 0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
 0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
 0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
 0.2000000    0.2000000    0.4500000    0.4500000    0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
  10.00000    10.00000    10.00000    10.00000    10.00000
  10.00000    10.00000    10.00000    10.00000    10.00000
  35.00000    35.00000    35.00000    35.00000    31.00000
  31.00000    31.00000    31.00000    31.00000    31.00000
  28.00000    28.00000    28.00000    28.00000    29.50000
  29.50000    29.50000    29.50000    29.50000    29.50000
  1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
  20.00000    60.00000    130.0000    50.00000    250.0000
  250.0000    250.0000    250.0000    250.0000    7634.000
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
```

```
 0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
 0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
 0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
 0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
 0.2000000    0.2000000    0.4500000    0.4500000    0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 10.00000     10.00000     10.00000     10.00000     10.00000
 10.00000     10.00000     10.00000     10.00000     10.00000
 35.00000     35.00000     35.00000     35.00000     31.00000
 31.00000     31.00000     31.00000     31.00000     31.00000
 28.00000     28.00000     28.00000     28.00000     29.50000
 29.50000     29.50000     29.50000     29.50000     29.50000
 1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 20.00000     60.00000     130.0000     50.00000     250.0000
 250.0000     250.0000     250.0000     250.0000     7634.000
 0.000000     0.000000     0.000000     0.000000     0.00000
 0.000000     0.000000     0.000000     0.000000     0.00000
 0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
 0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
 0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
 0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
 0.2000000    0.2000000    0.4500000    0.4500000    0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 10.00000     10.00000     10.00000     10.00000     10.00000
 10.00000     10.00000     10.00000     10.00000     10.00000
 35.00000     35.00000     35.00000     35.00000     31.00000
 31.00000     31.00000     31.00000     31.00000     31.00000
 28.00000     28.00000     28.00000     28.00000     29.50000
 29.50000     29.50000     29.50000     29.50000     29.50000
 1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 20.00000     60.00000     130.0000     50.00000     250.0000
 250.0000     250.0000     250.0000     250.0000     7634.000
 0.000000     0.000000     0.000000     0.000000     0.00000
 0.000000     0.000000     0.000000     0.000000     0.00000
 0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
 0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
 0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
 0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
 0.2000000    0.2000000    0.4500000    0.4500000    0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
 10.00000     10.00000     10.00000     10.00000     10.00000
 10.00000     10.00000     10.00000     10.00000     10.00000
 35.00000     35.00000     35.00000     35.00000     31.00000
 31.00000     31.00000     31.00000     31.00000     31.00000
 28.00000     28.00000     28.00000     28.00000     29.50000
 29.50000     29.50000     29.50000     29.50000     29.50000
 1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
8.2000002E-02 0.2200000    0.4600000    3.9999999E-02 0.2000000
5.7000000E-02 1.7999999E-02 5.5000000E-03 1.7000000E-03 9.9999997E-05
8.2000002E-02 0.2200000    0.4600000    3.9999999E-02 0.2000000
5.7000000E-02 1.7999999E-02 4.9999999E-03 1.7000000E-03 9.9999997E-05
```

Table 16.33., Succession Input for Grasses, Shrubs, and Trees (continued)

```
 8.2000002E-02  0.2200000     0.4600000     3.9999999E-02  0.2000000
 5.7000000E-02  1.7999999E-02  5.5000000E-03  1.7000000E-03  9.9999997E-05
 8.2000002E-02  0.2200000     0.4600000     3.9999999E-02  0.2000000
 5.7000000E-02  1.7999999E-02  5.5000000E-03  1.7000000E-03  9.9999997E-05
 8.2000002E-02  0.2200000     0.4600000     3.9999999E-02  0.2000000
 5.7000000E-02  1.7999999E-02  5.5000000E-03  1.7000000E-03  9.9999997E-05
PLTDRO Subroutine
  121212.0
  1.000000
  1.000000
  0.000000
  5.000000     6.000000     7.000000     8.00000    15.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
  2.000000     1.000000     1.000000     1.000000    2.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
 0.0000000E+00  0.0000000E+00   0.000000      0.000000   0.0
  200.0000     200.0000     200.0000     200.0000 200.0
UPTAXE Subroutine
  131313.0
       0
FORMAN Subroutine
  777777.0
 0.0000000E+00
  2.000000
 0.1900000
 0.5800000
  1.550000
 0.0
 0.0000000E+00
  2.000000
 0.300000
 0.3900000
  1.00000
 0.0
BIOPLT Subroutine
  444444.0
  1.000000
 0.0000000E+00   1.000000     1.000000
  1.000000     1.0000000E+00
BSMAX
2.   2.
i       n
 -1.000000     0.0000000E+00
BSMAX
2.   2.
i       n
 -2.000000     0.0000000E+00
BSMAX
2.   2.
i       n
```

Table 16.33., Succession Input for Grasses, Shrubs, and Trees (continued)

```
 -3.000000     0.0000000E+00
BSMAX
2.  2.
i      n
 -4.000000     0.0000000E+00
BSMAX
2.  2.
i      n
 -5.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 -5.000000  -1.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 -5.000000  -2.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 -5.000000  -3.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 -5.000000  -4.000000     0.0000000E+00
VTMBR
3.  22.
i      kt      n
 -5.000000  -5.000000     0.0000000E+00
  1.000000
  2.000000
IRUPLT Subroutine
171717.
  5.000000
  5*1.0
  13.00000
  1.000000      2.000000      3.000000      4.000000    5.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00 0.0
IPLOTR Subroutine
191919.
5        4        12
5*1
y
y
y
n
years
g/M2
LANL Biomass
```

Table 16.33., Succession Input for Grasses, Shrubs, and Trees (continued)

```
Pinyon
Ws Dec shrub
Ws An Grass
Ws Per Grass
Ponderosa
 -111.0000    -111.0000
 -111.0000    -111.0000
IRUPLI Subroutine
171717.
  5.000000
  5*1.0
  13.00000
  6.000000     7.000000     8.000000     9.000000    10.0
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0
 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0
IPLOIR Subroutine
191919.
5        10        18
5*1
y
y
y
n
years
ft3 / Acre
LANL Timber Volume
1" diam
3" diam
5" diam
7" diam
9" diam
 -111.0000    -111.0000
 -111.0000    -111.0000
```

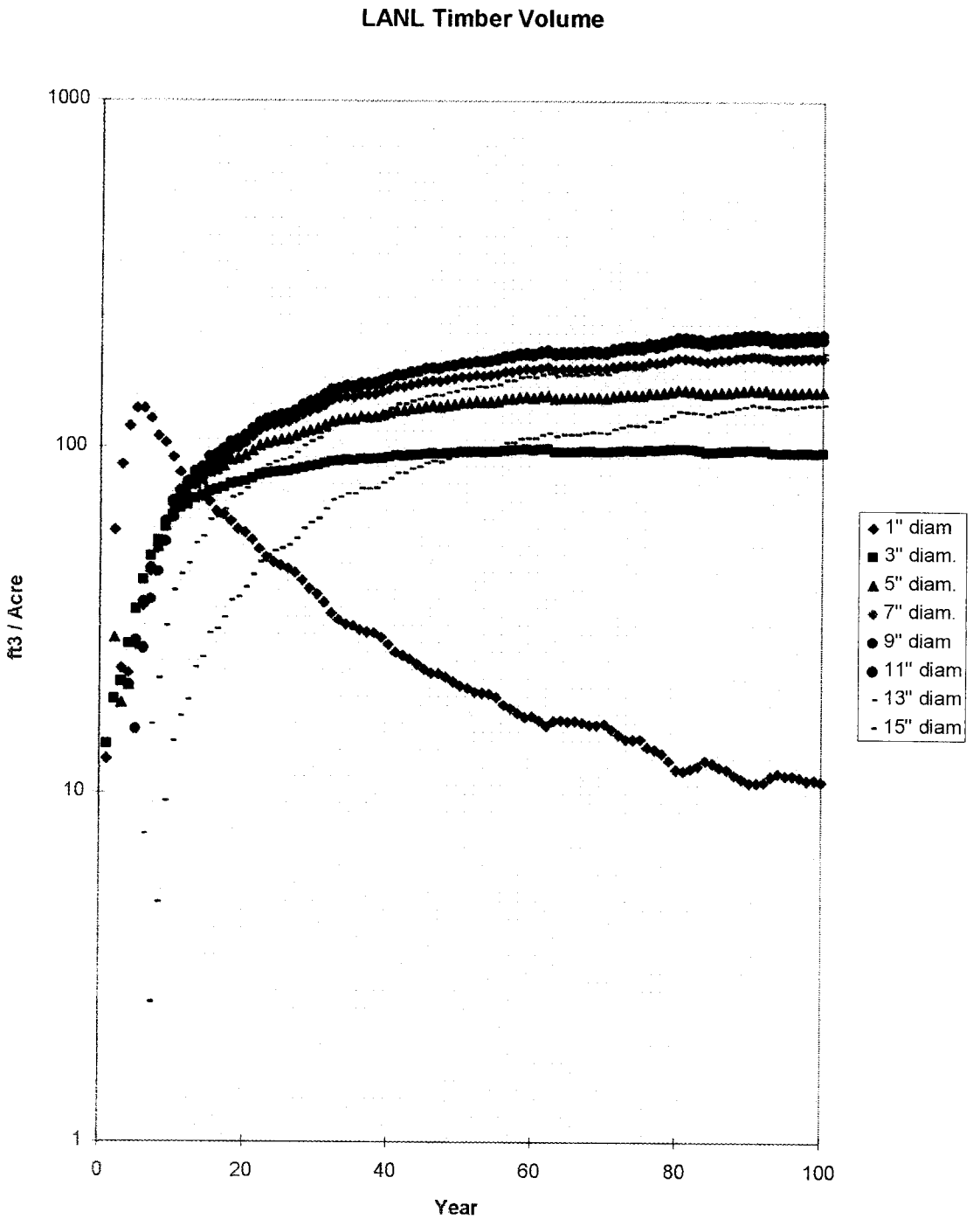


Figure 16.42.

Timber Volume by Diameter Classification for Pinyon Pine Stand at Los Alamos

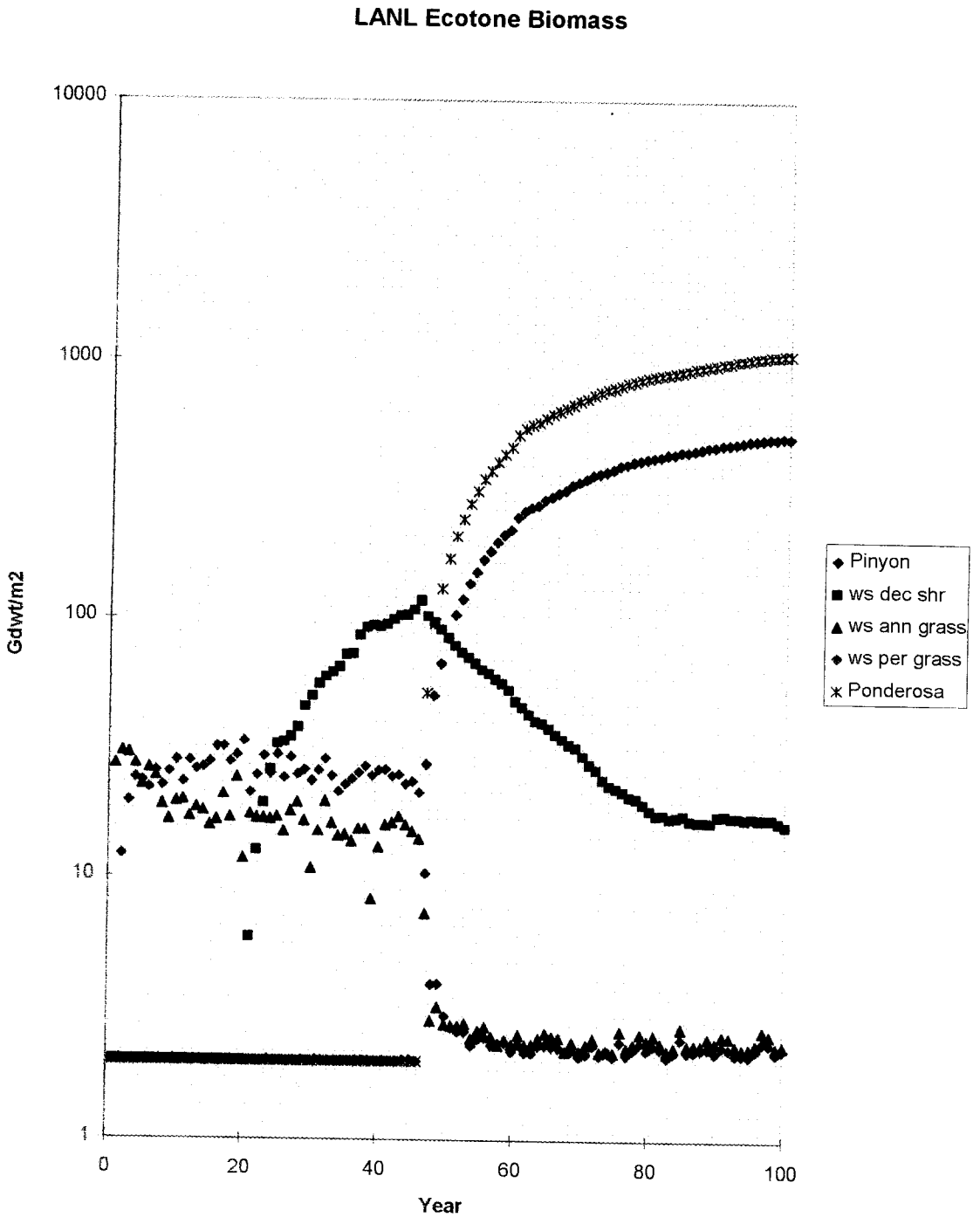


Figure 16.43.

Biomass Densities for Mixed Pine Stand for Los Alamos

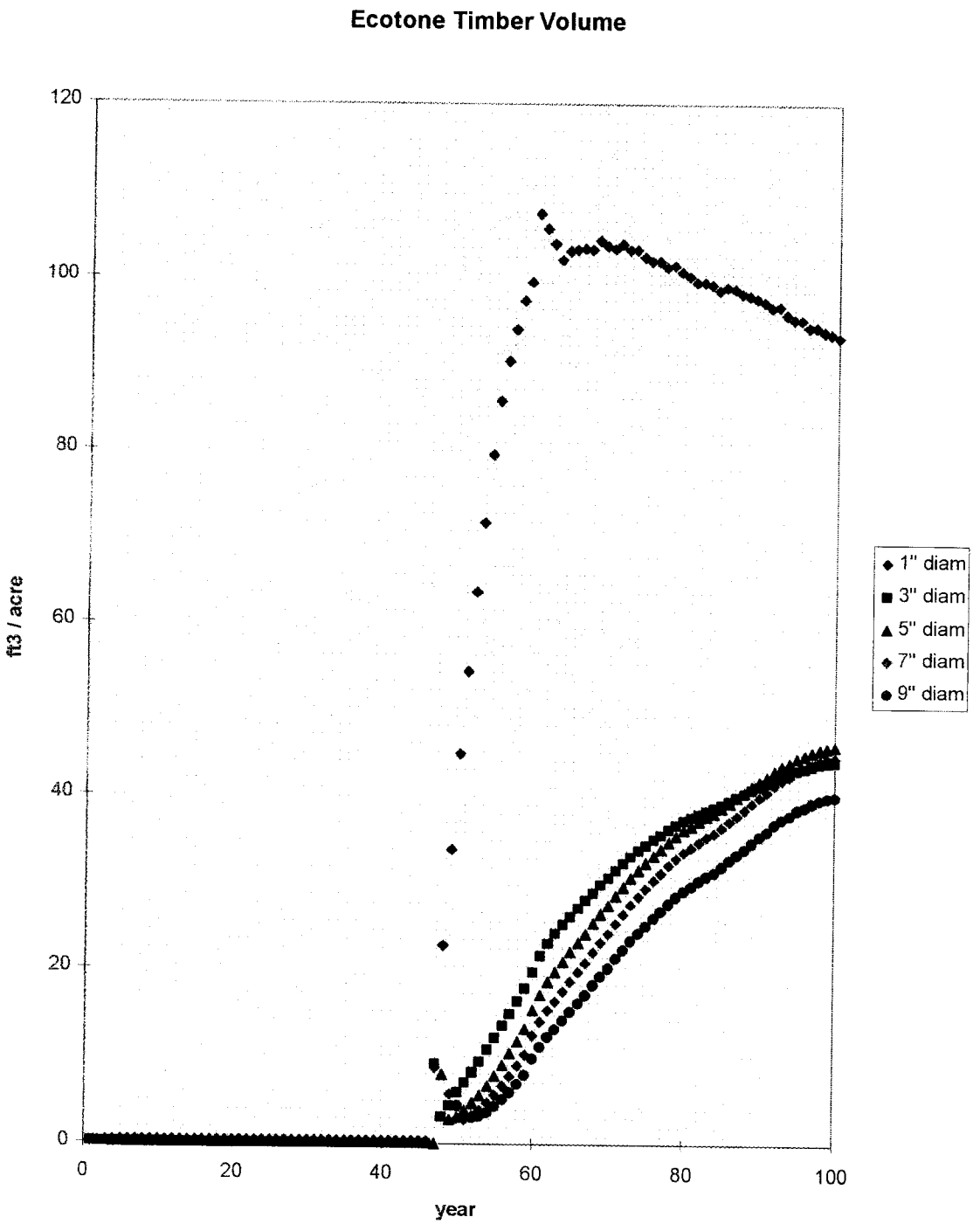


Figure 16.44.

Timber volume for Ponderosa Pine in a Mixed Pine Stand for Los Alamos

16.16. FORCUT Subroutine - Forest Management and Fire Events

16.16.1. Description

Subroutine FORCUT is called directly from FORMAN for input requirements during execution of BIOTRAN.2. FORCUT is used only in conjunction with FORMAN to simulate forest management by tree removal (type of cutting or logging or forest fire) of the type the user wishes to perform. One of the logging options includes a forest fire where the user specifies what fraction of specified diameter sizes of a tree stand are destroyed. Two types of cuttings are available: a choice cut where the user can choose up to three selective diameter groupings for removal in a given year(s). The user may also select a common removal scheme to optimize wood production (Reverse-J Cut) where trees are harvested based on an exponential cutting strategy (US Forest Service) that is a function of tree diameter, and removes all trees with diameters greater than 26" at breast Height (DBH). If a forest fire event is simulated such as the one described by Foxx (1984), then contaminant (and nutrient) losses via air transport are simulated in the variable AAA; with remainder entering the surface soil compartment (Z1) . The fire simulation affects all plants in the scenario including shrubs and grasses through the FORCUT subroutine.

16.16.2. Input / Output

FORCUT input instructions are presented in Table 16.34. Note that if a cutting or fire event is specified by FORCUT, then TBRM in FORMAN must be greater than or equal to (1.0). The **Example 2** input for FORCUT is presented in Table 16.35. and is identical to that of Table 16.32. of the previous section with exception of the additional lines of input required for FORCUT. A fire event is simulated to occur on the 50th year of the simulation of a 100 year run. The effects of a fire on timber volume (VTMBR) by diameter classification is illustrated by the plot in Figure 16.45.. Note the reappearance of the regenerative diameter class (1") after the simulated fire event. In this simulation 99% of the wood in all diameter classes was assumed to be destroyed.

16.16.3. References

Foxx, T.S. (Ed). 1984. La Mesa Fire Symposium. Los Alamos National Laboratory, Los Alamos, NM. LA-9236-NERP. 172pp.

Table 16.34.

FORCUT Subroutine Input

Input Line	Variable Name	Definition
1.		*Enter 666666 to access FORCUT*
2.	DATCT	Enter TBRM number of timber cuttings and/or forest fire(s) for a given plant type in a given region.
3.	TCUT	Enter TBRM cutting or removal types: 1=sawtimber (12-40",clear-cut ,1-4", pulpwood ,6-10", etc.), 2= Reverse-J Cut (US Forest service, above 26"), 3= fire (all diameters)
4.	ACUT	Enter the lower diameter range (eg. 5"-10" as an example) for the type of cut or forest fire. For Reverse-J enter (0.0, 0.0, default). The category may omitted by entering (0.0,0.0)
5.	BCUT	Enter the middle diameter range (eg. 12"-20" as an example) for the type of cut or forest fire. For Reverse-J enter (0.0, 0.0, default). The category may omitted by entering (0.0,0.0)
6.	CCUT	Enter the upper diameter range (eg. 25"-40" as an example) for the type of cut or forest fire. For Reverse-J enter (0.0, 0.0, default). The category may omitted by entering (0.0,0.0)
7.	RACUT	Enter the fraction of the total wood volume removed for the lower diameter range (ACUT). Enter (0.0) for RV-J cut or if omitting ACUT above.
	RBCUT	Enter the fraction of the total wood volume removed for the lower diameter range (BCUT). Enter (0.0) for RV-J cut or if omitting BCUT above.
	RCCUT	Enter the fraction of the total wood volume removed for the lower diameter range (CCUT). Enter (0.0) for RV-J cut or if omitting CCUT above.

Repeat lines 1-7 for each plant where RGC ≥2.0 in a specific region

Repeat lines 1-8 for each region simulated if RGC ≥2.0 for any plant type in that region

Table 16.35.

LANL Pinyon Pine Fire INPUT

```
MAIN Subroutine
100.00000
  1.000000
  1.000000
 0.0000000E+00  0.0000000E+00  0.0000000E+00   1.000000
  1.000000
 0.0000000E+00
  1.000000
  1.000000
 0.0000000E+00
  365.0000      1.000000
CLIMAT Subroutine
  555555.0
  270.0000      723.0000
  90.00000
  29.90000      70.40000
  120.0000
  2.000000
 0.2000000
  6990.000
 5.4000001E-03
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000      4.000000      4.000000      4.000000
  4.000000      4.000000
 5.0000001E-02  5.0000001E-02  5.6000002E-02  6.1000001E-02  9.7999997E-02
 8.5000001E-02  0.1570000     0.1640000     0.1040000     7.9999998E-02
 4.3000001E-02  5.0999999E-02
  16.00000
  6800.000
 0.0000000E+00
  138.0000
  37.00000
WATFLX Subroutine, Part I
  141414
 0.0000000E+00  0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.0000000E+00
 0.1000000
WATFLX Subroutine, Part II
  151515.0
  94.00000      7.000000      7.000000      15.00000      15.00000
  19.00000
  239.0000      14.00000      14.00000      31.00000      31.00000
  39.00000
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00
  1.000000
  2.000000      0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
```

Table 16.35., LANL Pinyon Pine Fire INPUT (continued)

```
0.0000000E+00
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
 1.000000    1.000000    1.000000    1.000000    1.000000
 1.000000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00
 10.00000
 2775.000
 20.00000    60.00000    130.0000    50.00000    250.0000
 250.0000    250.0000    250.0000    250.0000    1265.000
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.5500000    0.5500000    0.3300000    0.3300000    0.9000000
0.9000000    0.9000000    0.9000000    0.9000000    0.9000000
0.3000000    0.3000000    0.2200000    0.2200000    0.1000000
0.1000000    0.1000000    0.1000000    0.1000000    0.1000000
0.2000000    0.2000000    0.4500000    0.4500000    1.0000000E-03
1.0000000E-03 1.0000000E-03 1.0000000E-03 1.0000000E-03 1.0000000E-03
 10.00000    10.00000    10.00000    10.00000    10.00000
 10.00000    10.00000    10.00000    10.00000    10.00000
 35.00000    35.00000    35.00000    35.00000    31.00000
 31.00000    31.00000    31.00000    31.00000    31.00000
 28.00000    28.00000    28.00000    28.00000    29.50000
 29.50000    29.50000    29.50000    29.50000    29.50000
 1000.000    0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00
8.2000002E-02 0.2200000    0.4600000    3.9999999E-02 0.2000000
5.7000000E-02 1.7999999E-02 5.5000000E-03 1.7000000E-03 9.8999997E-05
PLIGRO Subroutine
 121212.0
0.0000000E+00
0.0000000E+00
0.0000000E+00
 5.000000
 0.000000
 0.000000
 2.000000
0.0000000E+00
0.0000000E+00
 200.0000
UPIAXE Subroutine
 131313.0
      0
FORMAN Subroutine
 777777.0
0.0000000E+00
 2.000000
0.1900000
```

Table 16.35., LANL Pinyon Pine Fire INPUT (continued)

```
 0.5800000
  1.550000
 0.0000000E+00
FORCUT Subroutine
666666
 50.00000
 3.000000
 1.000000      40.00000
0.0000000E+00  0.0000000E+00
0.0000000E+00  0.0000000E+00
0.01  0.01 0.01
BIOPLT Subroutine
  444444.0
  1.000000
 0.0000000E+00   1.000000      1.000000
  1.000000     1.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -1.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -2.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -3.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -4.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -5.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -6.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -7.000000     0.0000000E+00
VTMBR
3.  22.
i       kt      n
 0.0000000E+00  -8.000000     0.0000000E+00
B1
2.   2.
i       n
```

Table 16.35., LANL Pinyon Pine Fire INPUT (continued)

```
 0.0000000E+00  0.0000000E+00
BAR
2.   2.
kt      n
 -1.000000     0.0000000E+00
  1.000000
  1.000000
TRUPLT Subroutine
 171717.
  8.000000
  8*1.0
  13.00000
  1.000000      2.000000      3.000000      4.000000      5.000000
  6.000000      7.000000      8.000000
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00
TPLOTR Subroutine
 191919.
 5        8        19
 8*1
y
y
y
n
years
ft3/acre
LANL Timber Volume
diam 1
diam 3
diam 5
diam 7
diam 9
diam 11
diam 13
diam 15
 -111.0000     -111.0000
 -111.0000     -111.0000
```

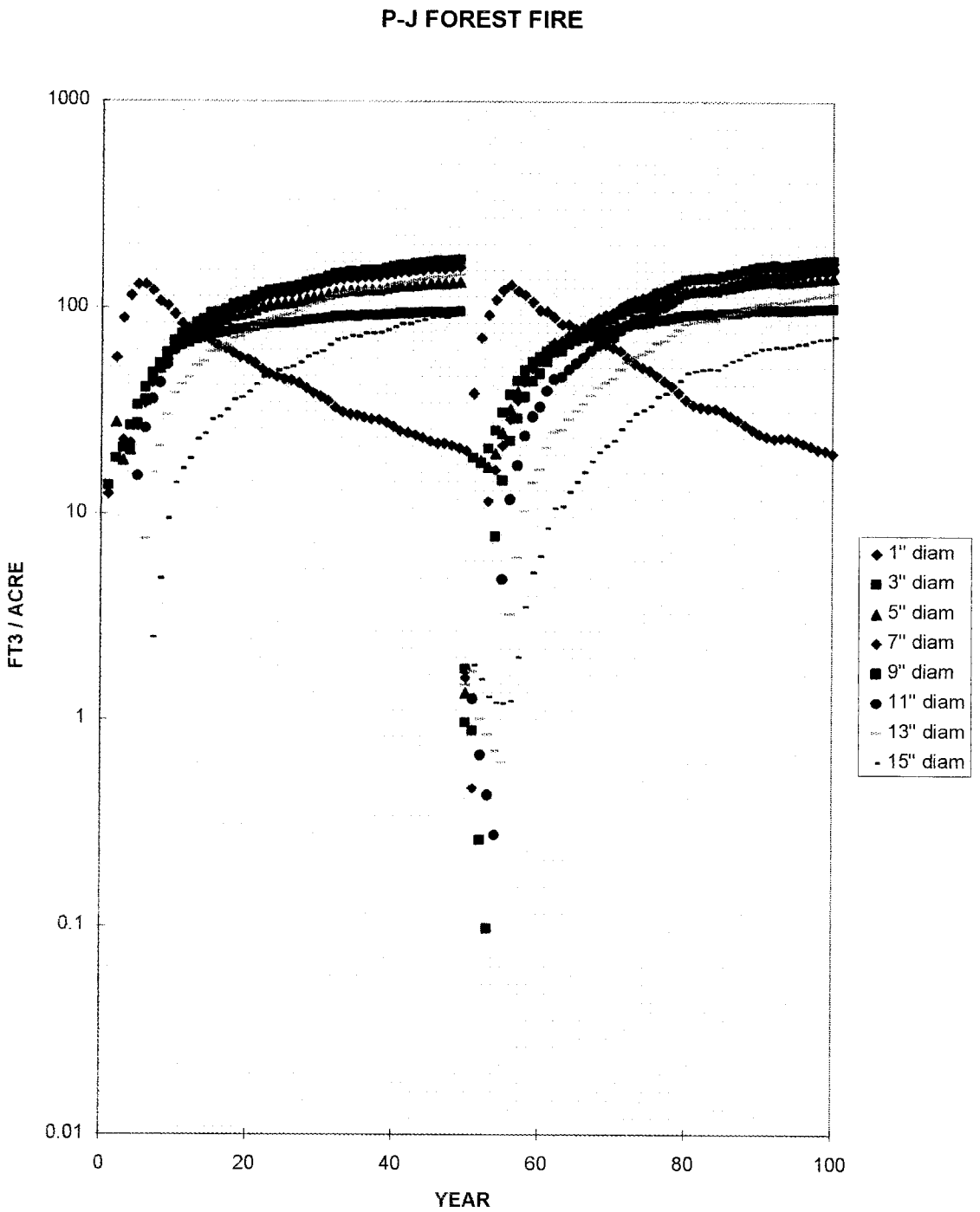


Figure 16.45.

Simulated Wood Volumes By Diameter Size From a Pure Stand of Pinyon-Juniper Before and After A Forest Fire

16.17. AQUAT Subroutine - Limnetic Zone

16.17.1. Description

The limnological processes simulated in AQUAT are presented in Figures 16.46. and 16.47. to illustrate the daily and hourly sunlight incidence estimated for each lake water layer using relationships for the northern hemisphere (Wetzell 1975). The amount of available light for photosynthesis that is converted to thermal energy is estimated for each layer modeled. Figure 16.47. shows the heat flux between layers, the air-water interface, and the bottom substrate of the lake. Water layers are physically transported along with associated nutrients, plankton, and contaminants (not yet modeled), in response to the relative water densities of the surrounding layers (Muller 1968). Five types of phytoplankton including green algae, blue-green algae, diatoms, etc. are modeled in association with two type of zooplankton (large and small) upon a food -web for each layer as well (Lehman 1975, Stross 1979, and Scavia 1976). The three major nutrients: P, N, and Si are exchanged between water and plankton as a function of water temperature, light intensity within a given year, and nutrient availability. Zooplankton filtration rates are also made dependent on water temperature.

AQUAT simulates heat, nutrient, contaminant (proposed), and biomass transport and exchange for a given volume of water in lakes of varying depths. It is planned to link AQUAT with a littoral zone model, with contaminant transport, and with other hydrological processes in BIOTRAN.2; and also to include fish as part of the food-web. Air temperature and solar radiation are daily variable obtained from subroutine CLIMAT in BIOTRAN.2. The development of a littoral zone model will allow coupling of AQUAT with many other subroutines in BIOTRAN.2 not currently possible for simulations of mineral and contaminant cycling as the hydrological cycle modeling become more complete for this subroutine.

16.17.2. Input / Output

A description of input required for AQUAT is presented in Table 16.36.. Table 16.37. presents AQUAT input for a shallow lake near Amarillo, Texas. The output in Figure 16.48. has not been verified, but is presented to illustrate the phytoplankton (5 types) and zooplankton (2 types) kinetics being simulated. The plot clearly shows several phytoplankton blooms during the late Spring and late Summer, followed by lagging zooplankton bursts. It also demonstrates that certain type of phytoplankton have different adaptations to the climatic conditions used in the simulation.

16.17.3. References

Lehman, J.T., Botkin, D.B., and G.E. Likens. 1975. The assumptions and rationales of a computer model of phytoplankton dynamics. Limnology and Oceanography (20): 343-364.

Muller, R.H., and J.A. Schufle. 1968. Shift in temperature of maximum density of water in capillaries. J. Geophysical Research (73): 3345-3347.

Scavia, D., and B.J. Eadie. 1976. The use of measureable coefficients in process formulation-zooplankton grazing. Ecological Modeling (2): 325-319.

Stross, R.G., Nobbs, P.A., and S.W. Chisholm. 1979. SUNDAY; a simulation model of an Arctic daphnia population. Oikos (32): 349-362.

Wetzell, R.G. 1976. Limnology. W.B. Saunders, Philadelphia, PA. 743pp.

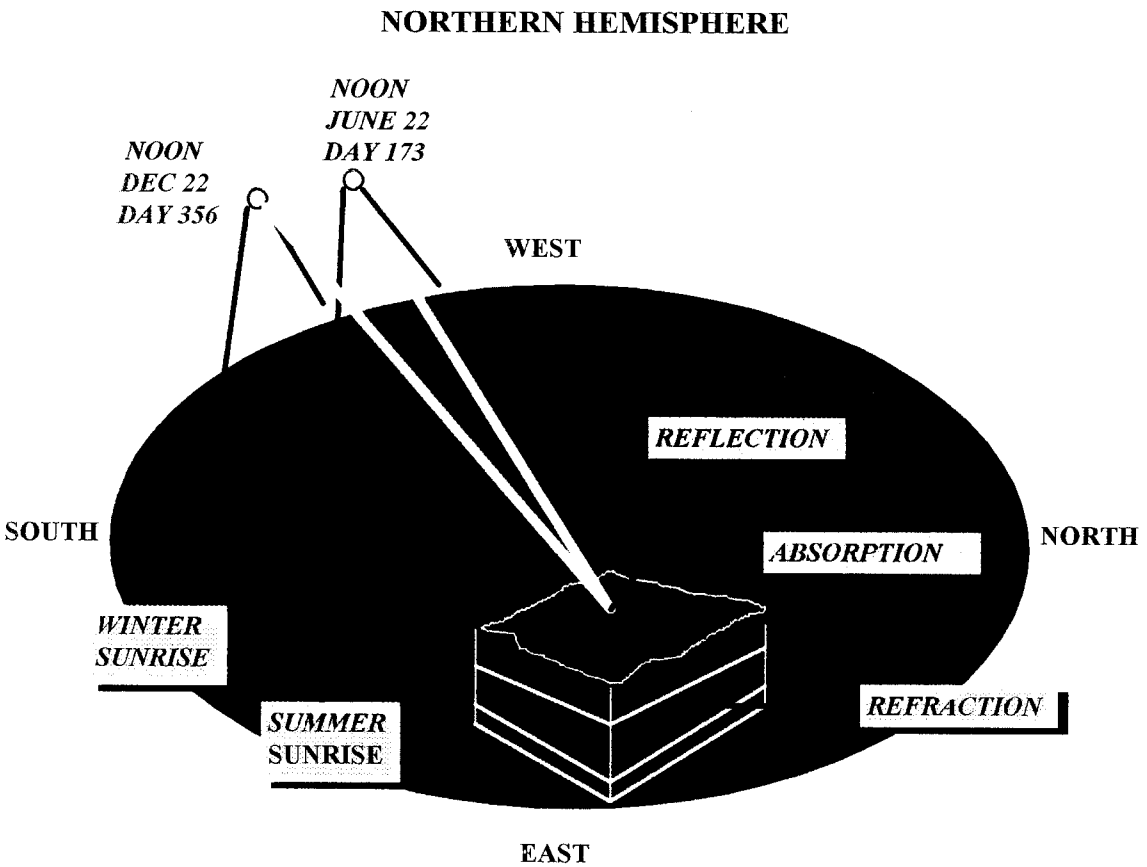


Figure 16.46.

AQUAT Subroutine Input

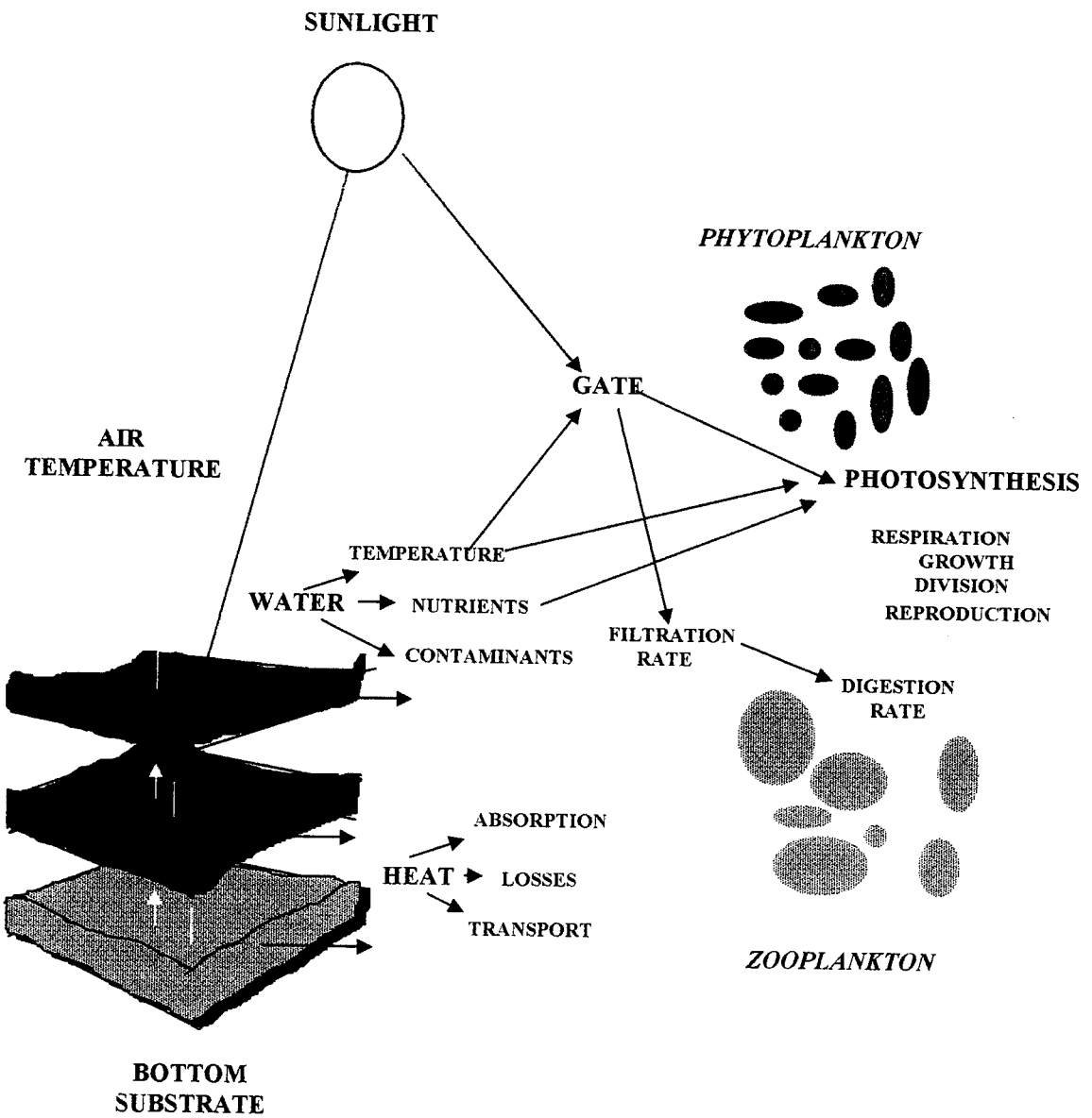


Figure 16.47.

AQUAT Simulation of Heat Flux Between Water Layers, Substrate, and Air; and Its Interaction With Other Biotic and Abiotic Components of The Aquatic System

Table 16.36.

AQUAT Subroutine Input

Input Line	Variable Name	Definition
1.		***Enter 333333 to access AQUAT***
2.	SKIP	Enter zooplankton feeding gate (0,1), 1=skip grazing on phytoplankton, 0=allow grazing. This gate is used to check phytoplankton growth without grazing effects.
3.	SNFRC	Enter fraction of light that is not backscattered from the water layers. Affects the amount of heat absorbed and phytosynthetic activity of phytoplankton. Enter for APX regions (see MAIN input).
4.	DEPTHI	Enter the total depth, meters, of the body of water. Enter for APX regions (see MAIN input).
5.	EXTC	Enter the mean light extinction coefficient, $meter^{-1}$, for the water medium including the effect of phytoplankton. Enter for APX regions (see MAIN input).
6.	DIT	Enter the number of water layers simulated for the body of water. Enter for APX regions (see MAIN input).
7.	EPLD	Enter the maximum depth, meters, to the eplimnion of a given body of water. If the lake does not stratify, then set value to (0.0). Enter for APX regions (see MAIN input).
8.	WATMP	Enter the initial temperature, °C, for DIT water layers
9.	PHOSP	Enter the initial phosphate concentration, ug/l, for DIT water layers.
10.	ANITR	Enter the initial nitrate nitrogen concentration, ug/l, for DIT water layers.
11.	SILCN	Enter the initial silicate silicon concentration, ug/l, for DIT water layers

Repeat lines 8-11 for each region simulated

Table 16.37.

Shallow Lake Input for Amarillo, Texas

```
MAIN Subroutine
4.00000
    1.00000
    1.00000
   0.000000E+00   0.000000E+00   0.000000E+00    1.00000
    1.00000
   0.000000E+00
   0.000000E+00
   0.000000E+00
    1.00000
    2.00000       1.00000
CLIMAT Subroutine
    555555.
    277.000       723.000
    90.0000
    36.2000       78.0000
    120.000
    4.36000
   0.200000
    3500.00
   4.000000E-03
    5.00000       5.00000       5.00000       5.00000       5.00000       5.00000       5.00000
    5.00000       5.00000       5.00000       5.00000       5.00000
   2.300000E-02   2.600000E-02   3.700000E-02   6.200000E-02   0.149000      0.161000
0.143000
   0.150000      9.300000E-02   8.900000E-02   3.900000E-02   2.900000E-02
    22.0000
    3538.00
   0.000000E+00
    280.000
    36.0000
    121212.
   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
UPTAKE Subroutine
    131313.
        0
A2UAT Subroutine
    333333.
   0.000000E+00
   0.750000
   0.250000
   0.400000
    10.0000
   0.000000E+00
    4.00000       4.00000       4.00000       4.00000       4.00000       4.00000       4.00000
    4.00000       4.00000       4.00000
```

Table 16.37., Shallow Lake Input for Amarillo, Texas (continued)

```
   3100.00    3100.00    3100.00    3100.00    3100.00    3100.00    3100.00
   3100.00    3100.00    3100.00
   5300.00    5300.00    5300.00    5300.00    5300.00    5300.00    5300.00
   5300.00    5300.00    5300.00
   3500.00    3500.00    3500.00    3500.00    3500.00    3500.00    3500.00
   3500.00    3500.00    3500.00
BIOPLT Subroutine
   444444.
   1.00000
  0.000000E+00   1.00000    0.000000E+00
   1.00000    0.000000E+00
PHYTO
3.0 16.
5      kw      n
-1.0 -1.0 0.0
PHYTO
3.0 16.
5      kw      n
-2.0 -1.0 0.0
PHYTO
3.0 16.
5      kw      n
-3.0 -1.0 0.0
PHYTO
3.0 16.
5      kw      n
-4.0 -1.0 0.0
PHYTO
3.0 16.
5      kw      n
-5.0 -1.0 0.0
ZOOPM
3.0 26.0
2      kw      n
-1.0 -1.0 0.0
ZOOPM
3.0 26.0
2      kw      n
-2.0 -1.0 0.0
ZOOPC
3.0 26.0
2      kw      n
-1.0 -1.0 0.0
ZOOPC
3.0 26.0
2      kw      n
-2.0 -1.0 0.0
WATMP
2.0 23.0
kw      n
-1.0 0.0
```

Table 16.37., Shallow Lake Input for Amarillo, Texas (continued)

```
1.0
1.0
IRUPLI Subroutine
171717
7.0
7*1.0
13.0
1.0 2.0 3.0 4.0 5.0 6.0 7.0
7*0
7*0
7*0
IPLOTR Subroutine
191919
3 8 15
7*1
y
y
y
n
day
ug/250 l
LAYER 1 BIOMASS
phyto 1
phyto 2
phyto 3
phyto 4
phyto 5
zooplk 1
zooplk 2
-111. -111.
-111. -111.
```

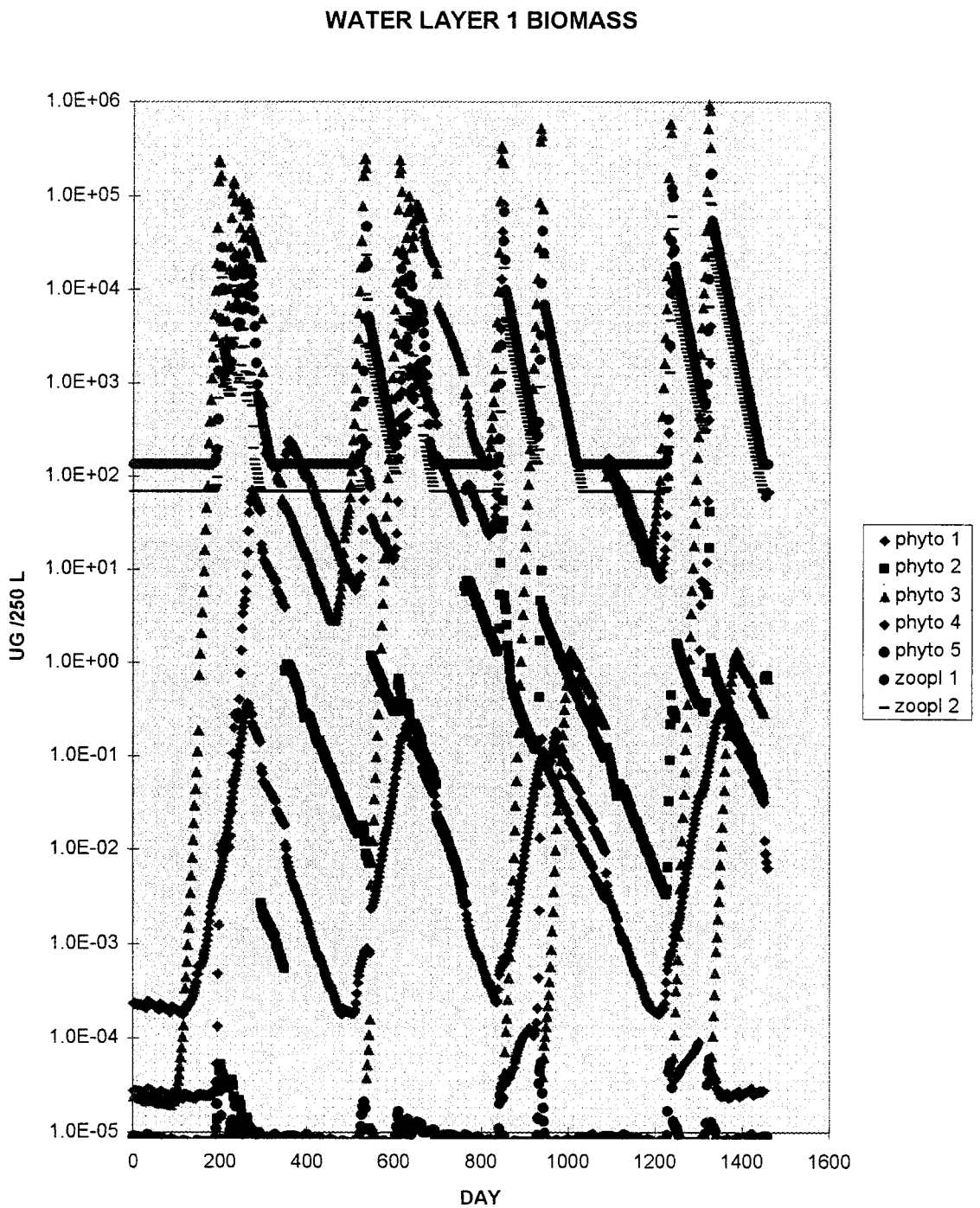


Figure 16.48.

Shallow Lake Phytoplankton and Zooplankton Densities for Amarillo, Texas

16.18. EFFECTS Model - Population Dynamics and Cancer Risks

The EFFECTS model uses dose parameters generated by either ANIMAL or HUMTRN subroutines when the input variable YDOS is set to a value greater than (0.0). EFFECTS processes yearly doses for specific organs and tissues and animals as a function of age and sex obtained from the above mentioned subroutines, and estimates the cancer risk in each age cohort separately for each sex and for any specified time interval. Thus, EFFECTS will estimate the total number of cancer mortalities as a result of radiation exposure on the animal or human population as a whole for both acute and chronic exposure scenarios developed in BIOTRAN.2. There is an upper boundary to the risk assessment depending on BIOTRAN.2 output used as input for EFFECTS. It is currently set to simulate ages 1-85 of both sexes for humans, and for the life-span of animals, yet it is a relatively small program of about 200 lines. However, it contains multi-arrayed population variables with three indices with a 6-stratified nested summation loop strategy to make possible the projection of cancer mortalities from a given age cohort to that grouping at various future times. EFFECTS (and HUMTRN) was adapted to run independently of BIOTRAN.2 in a special application (Gallegos 1989). It has also been executed using a main-frame computer system for a simulated time interval of 200 years (Wenzel 1985) to compare with earlier cohort analyses of this type reported by Buhl (1984). However, EFFECTS is not a cohort analysis model in the strictest sense, rather it is a dynamic population model simulated in either HUMTRN or ANIMAL subroutines that use population mortality and natality data for both humans and animals, respectively. It should be noted that animal cancer mortality rates have not been validated in the field or in the laboratory.

The 1980 US census data as a function of age and sex is part of the BIOTRAN.2 database. It also contains mortality and natality rates based on age and sex as well. These data statements can be updated or amended for specific applications. Also, BIOTRAN.2 can simulate a human or animal population structure to derive these parameters through user option. The maximum age set at 85 years for humans , and for the life-span of animals. Cancer risks are estimated for each age group and sex for specific organ or tissue type explicitly for every simulated year independently of previous or future exposures. Each year's dose has a specific risk for cancer induction in the future for a given age group and sex. The latter is projected into a future age where cancer mortalities will occur for this age cohort after the latent period, and during the projected expression period for the particular type of cancer under consideration. Both the linear and linear-quadratic models are used in EFFECTS to project cancer risks as reported by Buhl (1984) and the BEIR III task group (1980). Once a cancer risk is projected for a yearly exposure in an age group, it is stored, and when that age cohort arrives at the specific projected age, then the mortality rate of that group is increased during this time period to account for cancer mortalities. Hence , all exposures defined in the BIOTRAN.2 scenario are accumulated yearly for humans (varying time intervals for animals depending on the life-span) for each organ specified, and their expression as cancers are explicitly accounted for in EFFECTS. EFFECTS input must be carefully constructed and falls into three parts: BIOTRAN.2 organ/tissue doses, population dynamics, and adjustment of the latter from cancer mortalities.

16.18.2. Input / Output

A description of input parameters required to execute EFFECTS is presented is presented in Table 16.38.. **Example 1**, the Amarillo, Texas uranium uptake scenario for humans , shown in Table 16.39. is used to execute a 30 year BIOTRAN.2 simulation, except that only males are being considered. The plot shown in Figure 16.49. shows projected cancers for 15, 20, 25, 30, 40,50,60,and 70 year males for each year of the simulation. The expression period for these cancers is over the entire life-span, hence, the number of projections are quite numerous from one year to the next. The latent period has been set at 10 years so that cancers will not start to appear until this time interval has been equaled and exceeded. In the example noted the relative risk model has been used to make these projections. The array TCREL contains all relative risk projected cancer data by age group over time and can be used to construct a 3-D surface of cancer mortalities occurring in the given population over the entire simulated time period. Note that an acute air contaminating event has been simulated to occur in the 15th year of the simulation.

16.18.3. References

BEIR III. 1980. The effects on populations of exposure to low levels of radiation. National Research Council. National Academy of Sciences, Committee on the Biological Effects of Ionizing Radiation. National Academy Press, Washington, DC.

USNCI. 1981. Cancer incidence and mortality in the United States 1937-1977. USNCI Monograph (57). US Government Printing Office, Washington DC.

Buhl, T., and W.R. Hansen. 1984. Estimating the risks of cancer mortality and genetic defects resulting fro exposures to low levels of ionizing radiation. Los Alamos National Laboratory, Los Alamos NM. LA-9895-MS.

Wenzel, W.J., and A.F. Gallegos. 1985. EFFECTS: documentation and verification for a BEIR III cancer risk model based on Age, Sex, and Population dynamics. Los Alamos National Laboratory, Los Alamos, NM. LA-10731-MS.

Gallegos, A.F., and W.J. Wenzel. 1989. HUMTRN and EFFECTS: age and sex specific dosimetric and physiological human population dynamics models for dose assessment. Health Physics (57,1). 443-453.

Table 16,38.

EFFECTS Subroutine Input Requirements

Input Line	Variable Name	Definition
1		*Enter 919191 to access EFFECTS*
2	SRUN	Enter year of initial exposure to the human or animal population
3 used	DMDD	Enter the time interval ,days, (1-365) for calculation of risk effects. For early calculations, enter the value (365.0). The latter is always for human simulations.
4	ABSR	Enter type risk model to be used: 1=absolute risk model, 2=relative risk model.
5	POPFR	Enter the population multiplication number to adjust mortalities due to relative risk, absolute risk, and natural cancer and non-cancer rates. This value is generally set to 1.0 for humans and for most human applications.
6	PLTEA	Enter age specific expression period for cancer induction. Up to 8 entries can be made for a given cancer type. These values are presented for humans by Buhl (1984) ,Table III, and range from 24 years to lifetime for various cancer types. Current data in Table III are not age specific. Only one value covering all age groups in the population followed by 7 0.0's may be used in the absence of age specific information.
7	PLTEAY	Enter the upper age limit where a new expression period (or plateau) must be specified for a specific cancer type. Up to 8 entries can be included for each cancer type. Current data in Table III of Buhl (1984) does not present age specific expression periods. Hence, only one upper age limit followed by 7 0.0's may be used when age is not a factor. Values are in years
8	ORGRAD	Enter the gate (0,1) for identifying the type of cancer(s) to be simulated in a particular execution of BIOTRAN.2. There are currently 6 possible choices: 1=kidney,2=bone,3=liver,4=lung, 5=reserved for other cancer types identified by the user, and 6=endosteal cells of bone tissue. Enter the value (0.0) to deny access to a particular type of cancer, or (1.0) simulating . Six values must be entered.
9	ALTNT	Enter the latent period prior to cancer expression for each type of cancer simulated. Up to 8 age specific latent periods can be entered (see Table III, Buhl 1984). If latent period is not age specific, then one value followed by 7 0.0's may be entered.

10	ALTYR	Enter the upper age limit where a new latent period must be specified for a specific cancer type. Up to 8 entries possible (see Table III,

Table 16,38. EFFECTS Subroutine Input Requirements (continued)

		Buhl 1984). if latency is not age specific then the user may enter one latent period followed by 7 0.0's.
11	CMORTA	Enter age specific cancer risk rate, cancers 10^6person^{-1} year^{-1} rem^{-1}, for a specific cancer type from irradiation of a specific organ for a given sex for use with the linear model. 8 entries are possible. If risk rate is not age specific, then one rate followed by 7 0.0's may entered by the user.
12	CMORTB	Enter age specific cancer risk rate, cancers 10^6person^{-1} year^{-1} rem^{-1}, for a specific cancer type from irradiation of a specific organ for a given sex for use with the linear-quadratic model. 8 entries are possible. If risk rate is not age specific, then one rate followed by 7 0.0's may entered by the user.
13	CMORTY	Enter upper age limit by cancer type and sex where a new value for cancer risk must be specified (See Table III, Buhl 1984). If cancer rate is not age specific, then one rate followed by 7 0.0's may be entered by the user.
14	CNATR	Enter the natural organ specific cancer mortality rates, 10^5 person^{-1} year^{-1} by sex, for up to 16 age groups (see USNCI, 1981). values of 0.0 must be entered where not all possible age groupings are being used.
15	CNATY	Enter up age limit for each specific natural cancer rate as a function of age. trailing 0.0's must be added by user if they are not all used, or cancer are not age specific. 16 values must be entered.

Table 16.39.

Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Human Male Exposure to Uranium

```
MAIN Subroutine
30.0000
    1.00000
    1.00000
    1.00000      1.00000      1.00000      1.00000
    7.00000
    1.00000
    1.00000
   0.000000E+00
   0.000000E+00
    365.000      1.00000
AIRAC Subroutine
    111111.
    1.00000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   0.000000E+00
   0.000000E+00
    1.00000
    1.00000
    15.00000
    195.000
   1.000000E+08
   0.000000E+00
CLIMAT Subroutine
    555555.
    277.000      625.000
    90.0000
    36.2000      78.0000
    120.000
    4.36000
   0.200000
    3500.00
   4.000000E-03
    4.00000      4.00000      4.00000      4.00000      4.00000      4.00000      4.00000
    4.00000      4.00000      4.00000      4.00000      4.00000
   2.300000E-02   2.600000E-02   3.700000E-02   6.200000E-02   0.149000      0.161000
0.143000
   0.150000      9.300000E-02   8.900000E-02   3.900000E-02   2.900000E-02
    22.0000
    3538.00
   0.000000E+00
    202.000
    36.0000
WATFLX Subroutine, Part I
    141414.
   0.000000E+00   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
   0.000000E+00
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

WATLX Subroutine, Part II

```
151515.
92.0000     7.00000     7.00000     15.0000     15.0000     19.0000
238.000     14.0000     14.0000     31.0000     31.0000     39.1000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
1.00000     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
2.00000     0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
5.00000     5.00000     6.00000     7.00000     6.00000     6.00000     5.00000
990.600     990.600     1498.60     2006.60     1498.60     1498.60     990.600
20.0000     132.400     203.200     304.800     330.200
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.180000    0.180000    0.120000    0.120000    0.120000
0.490000    0.490000    0.410000    0.402000    0.402000
0.330000    0.330000    0.470000    0.478000    0.478000
18.7000     18.7000     18.9000     17.9000     16.5000
30.0000     30.0000     30.1000     27.2000     27.5000
28.5000     28.5000     28.6000     25.8000     26.1000
2.00000     2.00000     2.00000     2.00000     2.00000
20.0000     132.400     203.200     304.800     330.200
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.180000    0.180000    0.120000    0.120000    0.120000
0.490000    0.490000    0.410000    0.402000    0.403000
0.330000    0.330000    0.470000    0.478000    0.478000
18.7000     18.7000     18.9000     17.9000     16.5000
30.0000     30.0000     30.1000     27.2000     27.5000
28.5000     28.5000     28.6000     25.8000     26.1000
2.00000     2.00000     2.00000     2.00000     2.00000
20.0000     132.400     203.200     304.800     330.200     508.000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.180000    0.180000    0.120000    0.120000    0.120000    0.120000
0.490000    0.490000    0.410000    0.402000    0.403000    0.420000
0.330000    0.330000    0.470000    0.478000    0.478000    0.459000
18.7000     18.7000     18.9000     17.9000     16.5000     15.0000
30.0000     30.0000     30.1000     27.2000     27.5000     25.8000
28.5000     28.5000     28.6000     25.8000     26.1000     24.5000
2.00000     2.00000     2.00000     2.00000     2.00000     2.00000
20.0000     132.400     203.200     304.800     330.200     508.000     508.000
0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00
0.180000    0.180000    0.120000    0.120000    0.120000    0.120000    0.120000
0.490000    0.490000    0.410000    0.402000    0.403000    0.420000    0.420000
0.330000    0.330000    0.470000    0.478000    0.478000    0.459000    0.459000
18.7000     18.7000     18.9000     17.9000     16.5000     15.0000     15.0000
30.0000     30.0000     30.1000     27.2000     27.5000     25.8000     25.8000
28.5000     28.5000     28.6000     25.8000     26.1000     24.5000     24.5000
2.00000     2.00000     2.00000     2.00000     2.00000     2.00000     2.00000
20.0000     132.400     203.200     304.800     330.200     508.000
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000  0.180000  0.120000  0.120000  0.120000  0.120000
 0.490000  0.490000  0.410000  0.402000  0.403000  0.420000
 0.330000  0.330000  0.470000  0.478000  0.478000  0.459000
 18.7000  18.7000  18.9000  17.9000  16.5000  15.0000
 30.0000  30.0000  30.1000  27.2000  27.5000  25.8000
 28.5000  28.5000  28.6000  25.8000  26.1000  24.5000
 2.00000  2.00000  2.00000  2.00000  2.00000  2.00000
 20.0000  132.400  203.200  304.800  330.200  508.000
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000  0.180000  0.120000  0.120000  0.120000  0.120000
 0.490000  0.490000  0.410000  0.402000  0.403000  0.420000
 0.330000  0.330000  0.470000  0.478000  0.478000  0.459000
 18.7000  18.7000  18.9000  17.9000  16.5000  15.0000
 30.0000  30.0000  30.1000  27.2000  27.5000  25.8000
 28.5000  28.5000  28.5000  28.6000  25.8000  26.1000
 2.00000  2.00000  2.00000  2.00000  2.00000  2.00000
 20.0000  132.400  203.200  304.800  330.200
 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
 0.180000  0.180000  0.120000  0.120000  0.120000
 0.490000  0.490000  0.410000  0.402000  0.403000
 0.330000  0.330000  0.470000  0.478000  0.478000
 18.7000  18.7000  18.9000  17.9000  16.5000
 30.0000  30.0000  30.1000  27.2000  27.5000
 28.5000  28.5000  28.6000  25.8000  26.1000
 2.00000  2.00000  2.00000  2.00000  2.00000
 8.800000E-02  0.620000  0.200000  6.100000E-02  2.100000E-02
 8.800000E-02  0.620000  0.200000  6.100000E-02  2.100000E-02
 8.800000E-02  0.620000  0.200000  2.100000E-02  6.100000E-02  6.300000E-03
 5.010000E-02  0.510000  0.330000  0.160000  8.000000E-02  2.400000E-02
7.900000E-03
 8.800000E-02  0.620000  0.200000  6.100000E-02  2.100000E-02  6.300000E-03
 8.800000E-02  0.620000  0.200000  6.100000E-02  2.100000E-02  6.300000E-03
 8.800000E-02  0.620000  0.200000  6.100000E-02  2.100000E-02
PLTBRO Subroutine
 121212.
 0.000000E+00
 1.00000
 0.000000E+00
 1.00000  2.00000  8.00000  11.0000  12.0000  18.0000  23.0000
 6.00000  17.0000  0.000000E+00  10.0000  5.00000  0.000000E+00  17.0000
 2.00000  2.00000  0.000000E+00  4.00000  2.00000  0.000000E+00  2.00000
 1.00000  1.00000  1.00000  1.00000  1.00000  1.00000  1.00000
 1.00000  1.00000  0.000000E+00  1.00000  1.00000  0.000000E+00  1.00000
 0.000000E+00  0.000000E+00  1.00000  0.000000E+00  0.000000E+00  1.00000
0.000000E+00
 4000.00  4000.00  4000.00  4000.00  4000.00  4000.00  4000.00
 302.000  302.000
 0.000000E+00  7.00000  6.00000  6.00000  6.00000  6.00000
 150.000  150.000  201.000  229.000  243.000  257.000
 302.000  302.000
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
0.000000E+00   1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
1.00000      1.00000      1.00000      2.00000      2.00000      2.00000      2.00000
2.00000      2.00000      2.00000
150.000      150.000      157.000      164.000      171.000      177.000      184.000
191.000      198.000      205.000      212.000      219.000      226.000      233.000
240.000      247.000      254.000
1.00000      500.000      531.000      577.000
0.000000E+00   7.00000      5.00000      5.00000      5.00000      5.00000      5.00000
5.00000      5.00000      5.00000
215.000      215.000      470.000      485.000      502.000      514.000      527.000
533.000      544.000      559.000
1.00000      560.000
0.000000E+00   16.0000      5.00000      5.00000      5.00000
250.000      250.000      285.000      425.000      500.000
302.000      302.000
0.000000E+00   7.00000      2.00000      2.00000      2.00000      2.00000      2.00000
2.00000      2.00000      2.00000      3.00000      3.00000      3.00000      3.00000
3.00000      3.00000      3.00000
150.000      150.000      157.000      164.000      171.000      177.000      184.000
191.000      198.000      205.000      212.000      219.000      226.000      233.000
240.000      247.000      254.000
UPTAXE Subroutine
131313.
0
ANIMAL Subroutine
222222.
2
550.000      550.000
6.91000      6.91000
0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00
3.00000      1.00000
15.0000      15.0000
0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
1.00000
99.0000      0.900000      2.00000
1.00000
1.00000
1.00000
3.00000
15.0000      1.00000      1.00000      1.00000
3.00000      8.00000
1.00000      73.0000      74.0000      224.000      225.000      365.000
1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
0.000000E+00   0.000000E+00
3.00000      3.00000
1.00000      2.00000      3.00000
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
 2.00000
 11.0000      12.0000
 4.00000      1.00000
0.500000      0.500000
 3.00000
 8.00000      11.0000      18.0000
 1.00000      4.00000      2.00000
0.330000      0.330000      0.340000
 2.00000
 1.00000      11.0000
 5.00000      4.00000
0.780000      0.220000
2.000000E-02
0.231000
0.000000E+00
2.000000E-02
-1.00000
0.000000E+00
0.000000E+00
0.000000E+00
 2.00000
6.000000E-02
6.000000E-02
0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
 1.00000
 50.0000      0.900000      2.00000
 1.00000
 1.00000
 1.00000
 1.00000
 15.0000      1.00000      1.00000      1.00000
 3.00000      8.00000
 1.00000      365.000
 1.00000      1.00000
0.000000E+00   0.000000E+00
 1.00000      1.00000
 1.00000
 2.00000
 1.00000      11.0000
 5.00000      4.00000
0.780000      0.220000
2.000000E-02
0.231000
0.000000E+00
2.000000E-02
-1.00000
0.000000E+00
0.000000E+00
0.000000E+00
 2.00000
6.000000E-02
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
   6.000000E-02
HUMAN Subroutine
   999999.
   1.00000      0.000000E+00
   1.00000
   1.00000
   89.7000
   89.7000
   0.950000
   100.000
   6.91000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   1.00000
   85.0000      1.00000
   1.00000
   1.00000
   1.00000
   1.00000
   0.000000E+00   1.00000      1.00000      1.00000
   4.00000      30.0000
   1.00000      365.000
   1.00000      1.00000
   1.00000
   3.00000
   2.00000      12.0000      23.0000
   4.00000      5.00000      7.00000
   0.330000     0.330000     0.340000
   2.000000E-02
   0.231000
   0.000000E+00
   2.000000E-02
   -1.00000
   0.000000E+00
   0.000000E+00
   0.000000E+00
EFFECTS Subroutine
   919191.
   4.00000
   365.000
   10.00000
   1.00000
   200.000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
   100.000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
   1.00000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
   2.0000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
 100.000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
   0.700000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
    100.000      0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00
   0.000000E+00
   0.300000      0.100000      1.000000E-04   0.100000      0.200000      0.100000      0.500000
   0.700000      1.40000       4.00000       6.10000       10.9000       14.0000       19.0000
    23.7000      27.4000       27.4000       27.4000
    5.00000      9.00000       14.0000       19.0000       24.0000       29.0000       34.0000
    39.0000      44.0000       49.0000       54.0000       59.0000       64.0000       69.0000
    74.0000      79.0000       85.0000       85.0000
BIOPLT Subroutine
    444444.
    1.00000
   0.000000E+00   0.000000E+00   0.000000E+00
    1.00000      1.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -15.00000      -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -20.0000       -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -25.0000       -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -30.0000       -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -40.0000       -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -50.0000       -1.00000      -3.00000
TCREL
    4.00000      20.0000
200     kr     2      ko
   0.000000E+00   -60.0000       -1.00000      -3.00000
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

```
TCREL
   4.00000      20.0000
200    kr    2     ko
  0.000000E+00  -70.0000     -1.00000     -3.00000
TCREL
   4.00000      20.0000
200    kr    2     ko
  0.000000E+00  -80.0000     -1.00000     -3.00000
TCREL
   4.00000      20.0000
200    kr    2     ko
  0.000000E+00  -85.0000     -1.00000     -3.00000
   4.00000
   1.00000
IRUPLT Subroutine
   171717.
   8.00000
   1.00000      1.00000      1.00000      1.00000      1.00000      1.00000      1.00000
   1.00000
   13.0000
   1.00000      2.00000      3.00000      4.00000      5.00000      6.00000      7.00000
   8.00000
  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00
  0.000000E+00
  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00
  0.000000E+00
  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
0.000000E+00
  0.000000E+00
IPLOTR Subroutine
   191919.
      4     18     30
      1     1     1     1     1     1     1     1
y
y
y
n
year
Cancer Mortalities
Relative Risk Cancers In Males
15 yr-old
20 yr-old
25 yr-old
30 yr-old
40 yr-old
50 yr-old
60 yr-old
70 yr-old
   -111.000     -111.000
```

Table 16.39., Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Exposure (continued)

-111.000 -111.000

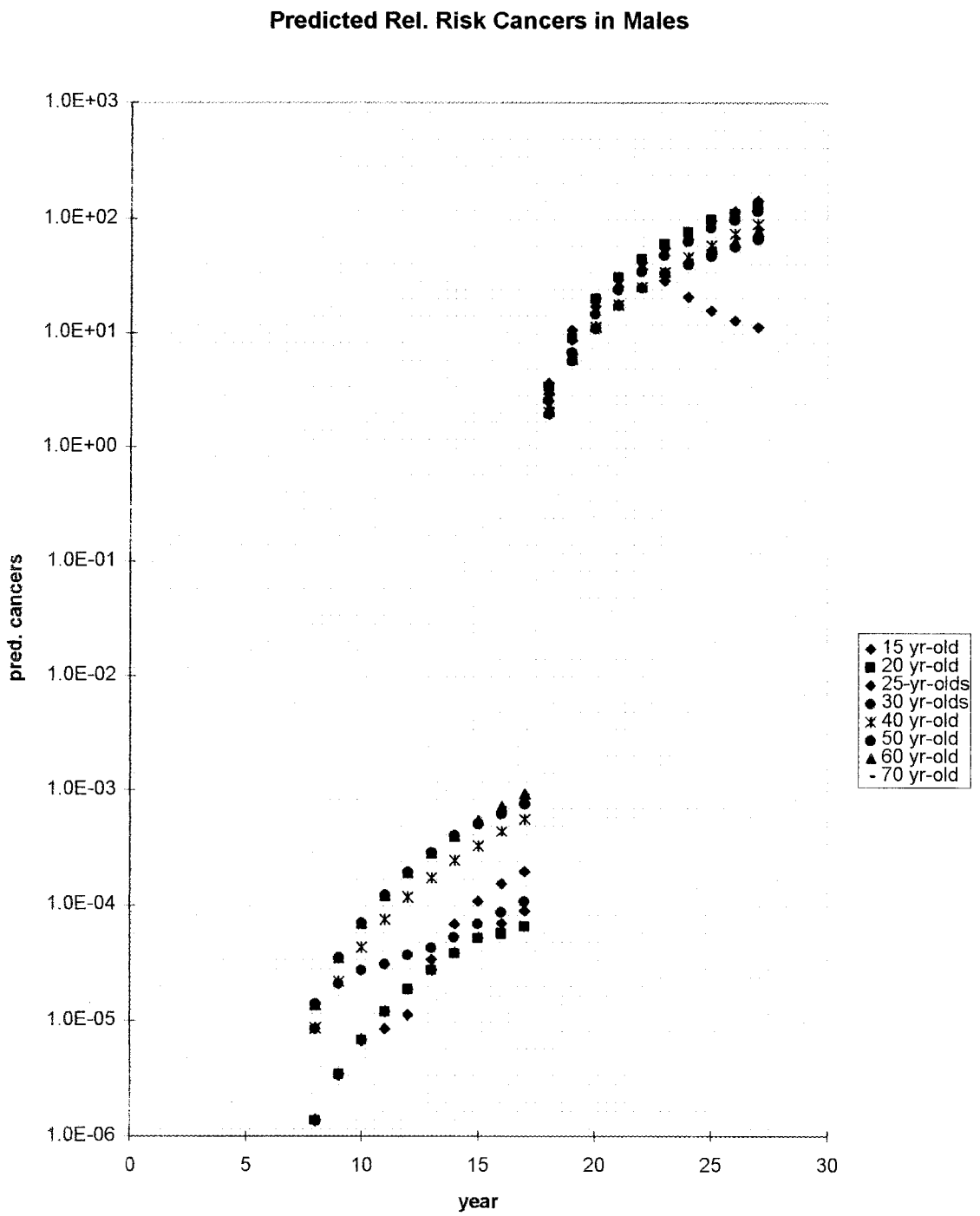


Figure 16.49.

Predicted Cancer Mortalities in Males From Uranium Exposure by the Relative Risk Methodology. (Note the Effect of the Acute Air Release)

General

LA-UR- 97-1469

Approved for public release; distribution is unlimited.

Title: DOCUMENTATION AND UTILIZATION OF THE ECOLOGICAL TRANSPORT MODEL BIOTRAN.2

LIBRARY COPY

Author(s): Anthony F. Gallegos, EES-15

Submitted to: ER Project Report

MAY 1997
RECEIVED

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. The Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (10/96)

DOCUMENTATION AND UTILIZATION OF THE ECOLOGICAL TRANSPORT MODEL (BIOTRAN.2)

PART 1 OF 2 PARTS

by

Anthony F. Gallegos
Environmental Science Group
EES-15

CONTENTS

ACKNOWLEDGMENTS....iv
ABSTRACT....v
1.0. INTRODUCTION....1
2.0. BACKGROUND ON MODEL DEVELOPMENT AND UTILIZATION....1
3.0. STRATEGY USED FOR MODEL ANALYSIS....6
4.0. MOISTURE FLOW AND CLIMATE....10
4.1. Precipitation....10
4.2. Temperature....12
4.3. Evaporation and Solar Radiation....13
4.4. Plant Transpiration....15
4.5. Soil Moisture Transport....16
4.6. Surface Runoff and Infiltration....18
4.7. Lateral Flow....22
4.8. Channel Flow....24
4.9. Channel Ice Formation....26
4.10. Shallow Aquifer Flow....27
5.0. SOIL AND SEDIMENT FLOW....33
5.1. Surface Runoff Events....33
5.2. Particle Size Effect....34
6.0. BIOMASS FLOW....39
6.1. Plants....39
6.1.1. Plant Niche and Succession Competition Factors....52
6.1.2. Forest Structure....55
6.1.3. Forest Management....60
6.1.4. Plant Nutrients....61
6.2. Animals....69
6.2.1. Food Preference and Feeding Strategies....69
6.2.2. Food Consumption, Metabolic Processes and Population Structure....82
6.2.3. Population Dynamics....89
6.2.3.1 Natural Mortality....90
6.2.3.2. Starvation....92
6.2.3.3. Prey Mortality....92
6.2.3.4. Invertebrate Prey Mortality....95
6.2.3.5. Natality....95
7.0. CONTAMINANT FLOW....98
7.1. plants....98
7.1.1. Resuspension: Air, Soils and Plants....103
7.1.2. Saltation-creep and Rainsplash: Air, Soils and Plants....106
7.1.3. Root Uptake....110
7.1.4. Translocation....111
7.2. Soils....112
7.2.1. Eluviation....112
7.2.2. Soil Moisture....113
7.2.3. Volatile Contaminants....113
7.3. Mammals....114
7.3.1. Pharmacokinetic Transport....114
7.3.2. Inhaled Resuspendable Particulates....124
7.3.3. Inhaled Volatiles....126
7.3.4. Soil Ingestion....126
7.3.5. Water Ingestion....132

8.0. **BIOMASS and CONTAMINANT FLOW** 133
8.1. Fetal Mammals 133
8.2. Weaning Mammals 134
9.0. **REFERENCES** 135
10.0. **APPENDIX A, FLOWS** 140
11.0. **APPENDIX B, EQUATIONS INVOLVING SECONDARY INTERMEDIATE VARIABLES** 164
12.0. **APPENDIX C, VARIABLES AND THEIR MNEMONIC EQUIVALENTS** 226
13.0. **APPENDIX D, GLOSSARY OF VARIABLES AND PARAMETERS THAT CAN BE SELECTED FOR OUTPUT IN BIOPLT SUBROUTINE FROM BIOTRAN.2 OPERATIONS** 243
14.0. **APPENDIX E, BIOTRAN.2 FORTRAN SOURCE CODE** 296
14.1. MAIN 296
14.2. AIRAC 309
14.3. ANIMAL 325
14.4. AQUAT 392
14.5. AQUIFER 408
14.6. BIOPLT 411
14.7. CLIMAT 441
14.8. EROSON 451
14.9. FORCUT 455
14.10. FORMAN 466
14.11. GEOFLX 479
14.12. HUMTRN 500
14.13. PLTGRO 562
14.14. RUNOFF 591
14.15. UPTAKE 594
14.16. WATFLX 630
14.17. ZOOGRO 657
14.18. BLOCK ARRAYS 674
14.19. INVERT 680
14.20. EFFECTS 695
14.21. TRUPLT 706
14.22. TPLOTR 711
14.23. MAIN.INC 723
15.0. **APPENDIX F, PLANT NUMBER POSITIONS FOR ECOLOGICAL TRANSPORT MODEL AND GROWTH SIMULATION DATA** 726

16.0. **APPENDIX G, BIOTRAN.2 USER'S MANUAL** 727

ABSTRACT 727
16.1. INTRODUCTION 728
16.1.1. Purpose and Description of User's Manual 728
16.1.2 Background on BIOTRAN.2 Development 728
16.1.3 General BIOTRAN.2 Structure 730
16.1.4 Example Description 730
16.1.5. References 734
16.2. General Procedures for Initiating Input to BIOTRAN.2
16.2.1. Getting Started 735
16.2.2. Loading the BIOTRAN.2 Executable File 735
16.2.3. Following Initial Input Screen Instructions 735
16.2.4. Selecting Input Options 736

16.3. Main Program....737

16.3.1. Description....737
16.3.2 Input....737
16.4. Climate Simulation....739
16.4.1. General Description....739
16.4.2. Solar Radiation....739
16.4.3. Temperature....739
16.4.4. Precipitation and Precipitation Pattern....741
16.4.5. Input/Output....741
16.4.6 References....747
16.5. BIOPLT Subroutine - Variable Selector for Graphical Displays....751
16.5.1. Description....751
16.5.2. Input/Output....751
16.6. TRUPLT Subroutine - Data Selection For Plotting....753
16.6.1. Description....753
16.6.2. Input/Output....753
16.7. TPLOTR Subroutine - Plotting Instructions....755
16.7.1. Description....755
16.7.2. Input/Output....755
16.8. AIRAC Subroutine- Acute and Chronic Releases of Contaminants....757
16.8.1. Description....757
16.8.2. Input/Output....757
16.9. WATFLX, RUNOFF, EROSON, AQUIFER, AND GEOFLX Subroutines - Soil Moisture Flow, Erosion, Watershed Simulations....773
16.9.1 Description....773
16.9.2. Input/Output....775
16.9.3. References....775
16.10. PLTGRO Subroutine And Plant Type Simulation....813
16.10.1 Description....813
16.10.2. Input/Output....813
16.10.3. References....838
16.11. UPTAKE Subroutine, Contaminant Uptake and Transport....839
16.11.1. Description....839
16.11.2. Input/Output....839
16.11.3. References....858
16.12. ANIMAL and INVERT - Mammal and Invertebrate Simulations....859
16.12.1. Description....859
16.12.2. Input/Output....859
16.12.3. References....892
16.13. HUMTRN Subroutine - Human Simulation and Dose....893
16.13.1. Description....893
16.13.2. Input/Output....893
16.13.3. References....913
16.14. ZOOGRO Subroutine - Grazing, Harvest, Feed, and Food Storage....914
16.14.1. Description....914
16.14.2. Input/Output....914
16.15. FORMAN Subroutine - Forest Growth Model....927
16.15.1. Description....927
16.15.2. Input / Output....927
16.15.3. References....928
16.16. FORCUT Subroutine - Forest Management and Fire Events....945
16.16.1. Description....945
16.16.2. Input / Output....945

16.16.3. References....945
16.17. AQUAT Subroutine - Limnetic Zone....952
16.17.1. Description....952
16.17.2. Input / Output....952
16.17.3. References....952
16.18. EFFECTS Model - Population Dynamics and Cancer Risks....960
16.18.2. Input / Output....960
16.18.3. References....960

17.0. LIST OF TABLES FOR THE USERS MANUAL(16.0.)

1 List of Subroutines for the BIOTRAN.2 Model....733
2 MAIN Subroutine Input....738
3 CLIMAT Subroutine Input....742
4 Example 1(Amarillo, Texas): Minimum Input Required to Execute BIOTRAN.2.743
5 Output From Example 1, Amarillo, Texas, for first 20 Days in the Third Year of the Simulation....748
6 BIOPLT Subroutine....752
7 TRUPLT Subroutine Input....754
8 TPLOTR subroutine Input....756
9 AIRAC Subroutine Input....759
10 Example 1 (Amarillo, Texas:) Input Used to Execute the Acute Air Contamination Scenario....762
11 Example 1 (Amarillo, Texas:) Input Used to Execute the Chronic Air Contamination Scenario....766
12 Output From Both the Acute and Chronic Simulations of Example 1, Amarillo, Texas, for Day 325-day 345of the first simulation year....770
13 WATFLX (I and II) and GEOFLX Subroutine Input....777
14 Example 1 (Amarillo, Texas): Input Used to Execute the Bare Soil WATFLX Scenario....785
15 Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop WATFLX Scenario....790
16 Example 1 (Amarillo, Texas): Input Used to Execute the Corn Crop Runoff Scenario at 20% slope....795
17 Example 1 (Amarillo, Texas): Input Used to Execute the Rangeland Runoff scenario at 20% slope....800
18 PLTGRO Subroutine Input....816
19 Examples 1 and 2, Input Used to Simulate Growth Dynamics and Transpiration In Various Types of Plants....819
20 Examples 3,and 4, Input Used to Simulate Growth Dynamics In Various Types of Plants Undergoing Succession and Niche Competition....828
21 UPTAKE Subroutine Input....840
22 Amarillo, Texas Example 1 WATFLX and PLTGRO Input For Single Soil Profile and Single Irrigated Plant (Corn) Utilized To Generate Uptake Output....841
23 Amarillo, Texas Example 2 WATFLX PLTGRO Input for Various Plants Utilized To Generate UPTAKE output....846
24 ANIMAL Subroutine Input....863
25 Amarillo, Texas Crops and Beef and Dairy Cattle....870
26 Amarillo, Texas Grasshopper Biomass Density....881
27 HUMTRN Subroutine Input....895
28 Plants, Cattle, and Human Input....901
29 Amarillo, Texas Crops ZOOGRO Input....915
30 Specific Gravity And Regression Coefficients For Selected Trees Simulated in FORMAN....930
31 FORMAN Subroutine Input....931
32 LANL Pinyon Pine Input....932

33 Succession Input for Grasses, Shrubs, and Trees (Pinyon and Ponderosa) in an Ecotone Setting For Los Alamos....936
34 FORCUT Subroutine Input....946
35 LANL Pinyon Pine Fire INPUT....947
36 AQUAT Subroutine Input....955
37 Shallow Lake Input for Amarillo, Texas....956
38 EFFECTS Subroutine Input Requirements....958
39 Amarillo, Texas BIOTRAN.2 Input for 30 Year Simulation of Human Male Exposure to Uranium....960

18.0. LIST OF FIGURES FOR THE USERS MANUAL (16.0.)

1 Structure of BIOTRAN.2....731

2 Major Processes Simulated in the CLIMAT Subroutine. Temperature and Precipitation are Stochastically Simulated and Drive the Daily Climate Simulation, Bounding the Biomass and Moisture Transport Subroutine Simulations....740
3 Simulated Rain (P) and Snow (F7) for the Amarillo, Texas, Climate Scenario For Four years....749
4 Simulated Mean Daily Temperature for the Amarillo, Texas, Climate Scenario For Four Years....750
5 Acute and Chronic Air Contaminant Concentrations Are Simulated Above the Soil Surface in AIRAC....758
6 Simulated Acute Contaminating Event for the Amarillo, Texas, Air Contamination Scenario for Four Years....771
7 Simulated Chronic Contaminating Event for the Amarillo, Texas, Air Contamination scenario for four years....772
8 Illustration of the 1 m^2 Soil Profile for Each Region Simulated in WATFLX. The User Specifies the Depth and Soil Characteristics For Each Soil Layer....774
9 Cumulative Precipitation Input From CLIMAT, P1, and Soil Surface Evaporation, EVAP, as Simulated in WATFLX for Amarillo, Texas Pullman Soils on Bare Ground....805
10 Soil Contamination in pCi/gram for Four Simulation Years. scenario....806
11 Buildup of Soil Water Content in the Second and Third Soil Layers from Bare Ground Conditions. Loss Through the Bottom of Soil Profile and Evaporation are not Sufficient to Remove all of the Added Soil Moisture. Transpiration of Moisture by Plants is Clearly Absent....807
12 Soil Surface Water Dynamics When Irrigated Corn is Grown on Soils Illustrated in Figures 16.9., 16.10., and 16.11....808
13 Soil Contamination Increasing in the First Layer After The Acute AIRAC Event on Day 195 of the Third year. The Initial and Subsequent Measure of Soil Surface Contaminants are Affected by Washoff of Uranium from Plant Surfaces....809
14 Soil Water Content is Significantly Different in This scenario Than That Shown in Figure 16.11. The Effect of Plant Transpiration Throughout the Growing Season is Apparent....810
15 Surface Runoff and Aquifer Volume for Irrigated Corn Scenario for a 4000 Acre Square Watershed With a 20% slope....811
16 Surface Runoff and Aquifer Volume for Uncropped Warm Season Perennial Grass On the Same Watershed to Compare With Figure 16.15. for Irrigated Corn Which Has Been Furrowed, and Does Not Allow Surface Runoff....812
17 Biomass Growth Processes Simulated In PLTGRO For 22 Different Plant Types With Three Soil Layers....815
18 Irrigated Corn Biomass For Various Plant Components for the Amarillo, Texas Scenario....832
19 Irrigated Corn Transpiration For Amarillo, Texas Scenario....833

20 Live Above Ground Biomass for Amarillo, Texas Scenario for Garden Vegetables, Alfalfa, Warm and Cool Season Grasses, and Winter Wheat. All But the Grasses Have Specific Irrigation and Cropping Schedules. Alfalfa and Winter Wheat Span More than One Crowing Season. The Grasses are Niche Competitors.....834
21 Transpiration For Grasses and Crops For the Amarillo, Texas Scenario. Compare These Curves With Those in Figure 16.19. for Irrigated Corn.....835
22 Secondary Plant Succession in Pinyon-Juniper Woodland. Note the Accelerated Growth After Complete Dominance of Trees.....836
23 Independent Growth of Plants in Pinyon-Juniper woodland. Succession and Niche Competition Strategies Have Not Been Applied To Any Plant Type or Grouping.....837
24 Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine.....853
25 Contaminant Transport Processes In and On Plants Modeled in UPTAKE Subroutine.....855
26 Daily Surface Particle Concentrations of Uranium On Irrigated Corn Leaves and seeds for Amarillo, Texas.....856
27 Internal Leaf Concentrations for Uranium for Crops and Range Grasses At Amarillo,Texas.....857
28 Illustration of ANIMAL/ZOOGRO Operations With the Human Model HUMTRN. The Invertebrate Model INVERT Also Interacts in A Similar Manner.....861
29 Contaminant Metabolism in a Mammal Based on Physiologically Based Pharmacokinetics Modeling in BIOTRAN.2.....862
30 Beef Cattle Air, Soil, Feed, and Total Uranium Intakes For Amarillo, Texas.....887
31 Dairy Cattle Air, Feed, and Total Uranium Intake for Amarillo, Texas.....888
32 Beef Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas.....889
33 Dairy Cattle Uranium Concentrations in Various Organs and Tissues for Amarillo, Texas.....890
34 Male and Female Grasshopper Biomass Densities for Amarillo, Texas.....891
35 HUMTRN Model Integrates the Volume, Mass, and Contaminant Intake Into Human Male and Female Population. Humans are Simulated Based On User Diet Selection and Population Dynamics.....894
36 Uranium Uptake From Various Foods by 25 Year Old Males.....911
37 Radiation Doses From U-238 to Various Organs and Tissues of 25 Year Old Males.....912
38 ZOOGRO Subroutine Simulates Daily forage, feed, or food to ANIMAL, INVERT, and HUMTRN According to Diet Schedules Supplied By the User.....924
39 Crop Harvests For Beef Cattle in the Amarillo, Texas Scenario.....925
40 Crop Harvests for Dairy Cattle in the Amarillo, Texas Scenario.....926
41 FORMAN Subroutine Manages the Growth and Contaminant Transport for Forest Stands Based on Selections Made in PLTGRO.....929
42 Timber Volume by Diameter Classification for Pinyon Pine Stand at Los Alamos.....942
43 Biomass Densities for Mixed Pine Stand for Los Alamos.....943
44 Timber volume for Ponderosa Pine in a Mixed Pine Stand for Los Alamos.....944
45 Simulated Wood Volumes By Diameter Size From a Pure Stand of Pinyon-Juniper Before and After A Forest Fire..... 951
46 AQUAT Subroutine Input.....953
47 AQUAT Simulation of Heat Flux Between Water Layers, Substrate, and Air; and Its Interaction With Other Biotic and Abiotic Components of The Aquatic System.....954
48 Shallow Lake Phytoplankton and Zooplankton Densities for Amarillo, Texas.....959
49 Predicted Cancer Mortalities in Males From Uranium Exposure by the Relative

Risk Methodology....969

19.0. LIST OF TABLES FOR THE BIOTRAN.2 DOCUMENTATION (8.0.)

1 Description of Transport Rates Ilustrated in Figure 4....9

20.0. LIST OF FIGURES FOR THE BIOTRAN.2 DOCUMENTATION (8.0.)

1 Ecological transport model....3
2 Intraregional and interregional integration routes of contaminants....4
3 Relationship between model components and model code....5
4 BIOTRAN.2 flow strategy....8
5 Moisture Flow Diagram....11
6 Moisture Flow Modeling Strategy....21
7 Moisture Runoff and Transmission Processes Below a Channel or Channel Segment....29
8 Illustration of Regional Channel and Aquifer Structure and Function....37
9 Biomass Flow and Peripheral Attributes....41
10 Biomass Flow in Forest Structure Model (FORMAN)....58
11 Ammonium-nitrate Transport and Exchange Strategies Employed in BIOTRAN.2....70
12 Phosphate Transport and Exchange Strategies Employed in BIOTRAN.2....71
13 Example of Food Selection Criteria for an Herbivore in BIOTRAN.2....75
14 Food Preference/ Selection Strategy From Available Food Sources....78
15 Mammalian Metabolic and Growth Strategies Used in BIOTRAN.2....94
16 Mammalian Population Dynamics Processes Used In BIOTRAN.2....101
17 Air, Soil, and Plant Interaction Strategies In BIOTRAN.2 For Contaminant Transport....102
18 Internal Contaminant Concentration In Biomass Compartments....116
19 Pharmacokinetics Transport Strategy for Contaminants Used in BIOTRAN.2....119
20 Transport of Contaminants From the Lung Surface to Organs and Tissues In BIOTRAN.2....127

ACKNOWLEDGMENTS

Many man-hours have gone into the development of BIOTRAN.2, parts of which were initiated about twenty-two years ago at Los Alamos National Laboratory (LANL). Other individuals who have helped to develop and support this effort are W.J. Wenzel (ESH-12), W.R. Hansen (EES-15), J.C. Rodgers (ESH-4), L.R. Soholt (EM/ER/D&D), R.W. Ferenbaugh (EES-15), and other past and present employees of LANL such as W.J. Smith (ERM), and M.L. Wheeler (LATA). The author's appreciation is extended to these and other individuals who contributed to the development of this model.

ABSTRACT

Inclusive of Appendices, this document describes the purpose, rationale, construction, and use of an ecological transport model (BIOTRAN.2). This model is used to predict the flow of organic and inorganic contaminants including radionuclides through specified plant and animal environments using biomass as a vector. The model also predicts the movement of contaminants through the physical environment though soil erosive processes, air, through the movement of soil moisture, and moisture movement in shallow aquifers along prescribed channel beds.

The main text is presented with a specific format which uses a minimum of space, yet is adequate for tracking most relationships from their first appearance in the text to their location in the code. The code itself is heavily commented to assist in this effort. Because relationships are treated individually in this manner, and rely strongly on Appendix material for understanding, the reader should become familiar with these materials as well.

1.0. INTRODUCTION

This document incorporates the documentation of a transuranic (TRU) biological transport model BIOTRAN[1] which was issued in 1980. Since that time the model has been refined and expanded to include a physiologically based human model[2], a human radiation dose effects model[3], a surface erosion model for contaminated soil particles[4] based on a multiregional or watershed approach, and an ability to simulate the transport of more than one contaminant per simulation run[5]. The model has been used in a number of applications during the last decade including an Environmental Impact Statement (EIS) at the Pantex site [6,7,8,9,10] near Amarillo, Texas. Many portions of the model have been refined, and made more user friendly with respect to input specification, and in the production of graphics output than was reported in a 1992 users manual version [11]. The name of the code was changed from BIOTRAN to BIOTRAN.2 in 1993 after the model had undergone intensive modification to incorporate a pharmacokinetic approach to make possible the transport of organic contaminants in plants and animals. The latter was necessary in order to include organic contaminants along with radionuclide and metal contaminants as part of the transport capabilities of this model. Finally, the new version of the model is capable of being executed on a PC which was not the case previously except for a small portion of the model[12].

The above mentioned citations as well as previous LANL progress reports [13,14,15], one other LANL report[16], and earlier LANL presentations [17,18,19,20] have provided limited documentation of model aspects including validation, stability, capabilities, and sensitivity analysis. This report will provide the rationale used in the development and utilization of the model itself. The databases external to the model which have been used in BIOTRAN.2 (and BIOTRAN) have also been reported in former citations of model use, and will not be reported unless they are part of an application included in this report. This report will put into perspective model capabilities, limitations, and the usefulness of the model in predicting the movement of contaminants and nutrients through the biotic components of an ecosytem using biomass as a vector.

2.0. BACKGROUND ON MODEL DEVELOPMENT AND UTILIZATION

A generalized description of BIOTRAN.2 is presented in Fig. 1 as it was generally conceived for application to a given grid of the study area[13] in 1974. This relationship was used as the basis for model development to the present time. The model has always been constrained to permit exchanges and transport of contaminants (originally transuranics (TRU) and other radionuclides, but now includes other inorganic and organic substances also) along what are considered the most significant pathways. The original modular approach design[1] was expanded to include a larger number of subroutines for realization of the transport processes indicated, however, some modifications, deletions, and consolidation of modules became necessary or more expedient as model development proceeded. All of the major components shown in Fig. 1 are currently represented in BIOTRAN.2, with exception of the contaminant transport modeling in the aquatic module of HYDROSPHERE. Other portions of HYDROSPHERE including transport of contaminants through water runoff and soil erosion caused by the latter are now complete[4]. The ATMOSPHERIC module remains , as in past model versions, dependent on user input of acute air concentrations and/or a chronic input function of such[1].

Early efforts were also directed toward modeling inter-regional (transport between gridded areas) transport routes for transuranic contaminants (TRU) as was described when an overall status report of module development was presented (Fig. 2); [14] however BIOTRAN.2 still does not have this capability except for the modeling of water runoff and soil erosion processes on this basis. The model does have the capability of simulating more than one region at a time, and can model the placement of animals and humans in a multi-regional mode. This early description[14] does indicate how the original conception

of the model has been changing to meet project needs over the last twenty years, and can be used to compare with the latest conceptual version of the model now that all the major modules have been completed as shown. Hence, the present model integrates transport of contaminants from biotic and abiotic components to humans as originally planned, and many of the module "lumpings" present in the first version[1] are now separated into subroutines , and are no longer embedded within larger groupings (Fig. 3). This has resulted in a simplification of the program coding arrangements and the complications which were present in earlier versions. For instance the main program (PLANTS) of an earlier version[1] carried all elements previously included in LITHOSPHERE, HYDROSPHERE, ATMOSPHERE, and most elements of PLANTS, whereas, the current model separates these elements into different subroutines, and the main program is assigned the role of data input, and integration of subroutine functions. However, imbedding of some module elements as sub-modules are still of necessity present in the code with varying degrees of of dependence on each other for proper operation as will be shown later in this report.

BIOTRAN.2 (inclusive of, and as an extension of BIOTRAN) was constructed primarily to fulfill several project objectives:[13]

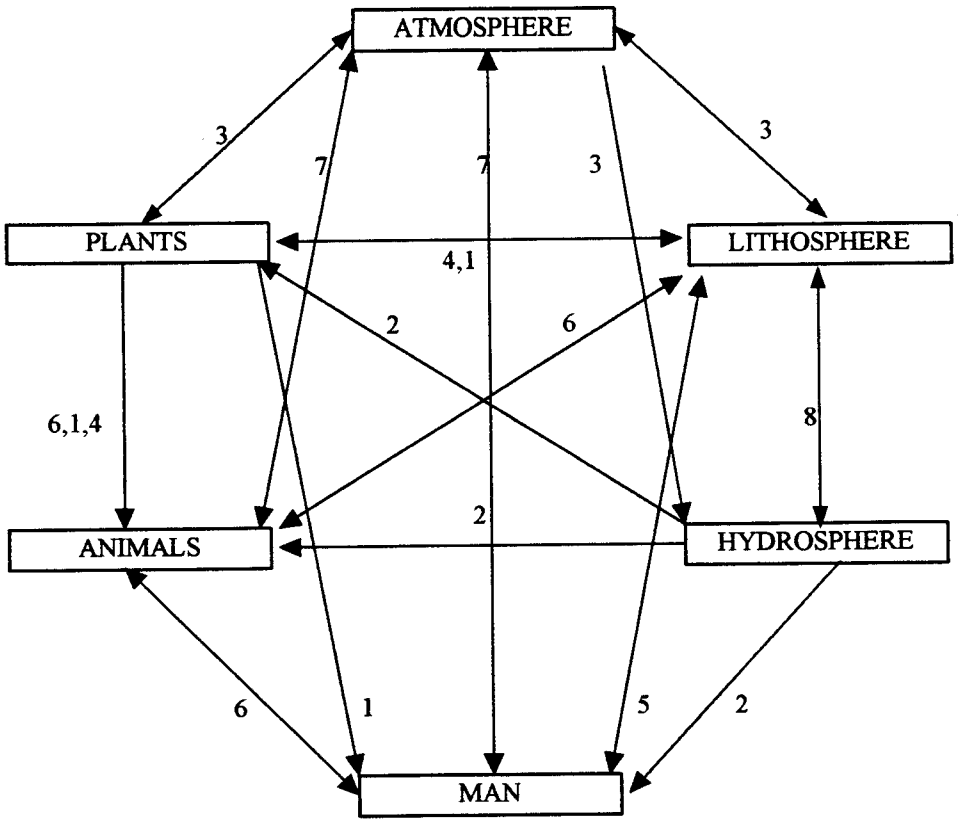
- to model the movement of contaminants through ecosystems within the study area.
- to provide enough generality for extending its use to other areas.
- to be capable of predicting both short and long range cycling of contaminants in a number of climatic regimes .
- to provide output and graphics information which could be used in evaluating potential hazards to man , animals, and plants from specific contaminants as a result of waste storage, and accidental releases to the environment.

A new project objective which has surfaced in conjunction with the Environmental Restoration Program (ER) at LANL is the need to perform effects at the population level with BIOTRAN.2. Although this version of the model is capable of predicting the human and animal cancer mortalities as a result of radionuclide transport to tissues and organs on the respective populations, it is not capable of predicting any other types of effects from other contaminants . The latter option is in the planning stage at this writing. (Please note that from this point of the discussion until the end of this document any reference to BIOTRAN.2 is inclusive of BIOTRAN unless so noted.) In the early modeling phases of BIOTRAN a decision was made to concomittantly model :

- ***flow of animal and plant biomass***
- ***flow of moisture***
- ***flow of contaminants***

Flows are here defined as the transport or exchange of substances between respective pools or reservoirs of those substances, or gains or losses from the system (sources or sinks, respectively). Substances which originate external to the system, but add to it are called sources. ***Flow of soil*** and ***flow of channel alluvium***, was added in 1990[4] when a mult-regional or watershed approach was developed. A mathematical model of intermediate complexity has been constructed consisting primarily of:

- ***State variables*** (X_i) - model elements that comprise the system structure (compartments)
- ***Driving variables*** (D_i) - model elements which vary with time, affect flow of compartmental (pool) material, but are largely independent of the system
- ***intermediate variables*** (I_i) - model elements which control or affect flow of pool material from one compartment (state) to another.
- ***Flows*** (F_i) - vectored transport of substances into or out of compartments or to sinks



LEGEND

1 = Agricultural growth module
2 = Aquatic and soil water modules
3 = Atmospheric, resuspension, and saltation-creep modules
4 = Forage and tree growth modules
5 = Human module
6 = Herbivore module
7 = Inhalation module
8 = Erosion and Surface runoff module

Fig. 1
Ecological transport model

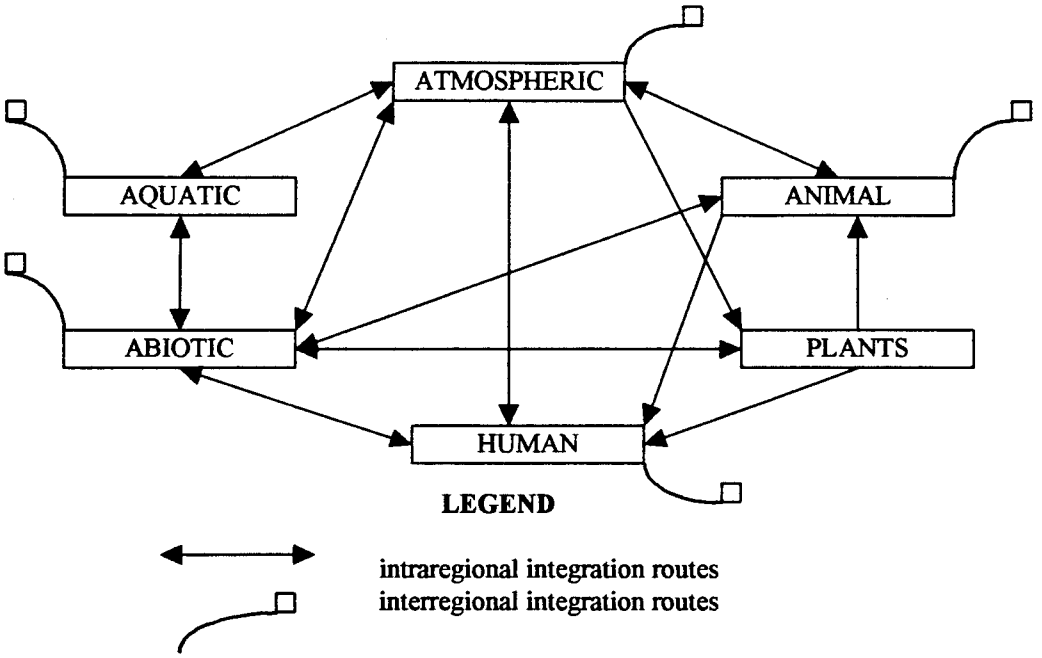


Fig. 2

Intraregional and interregional integration routes of contaminants

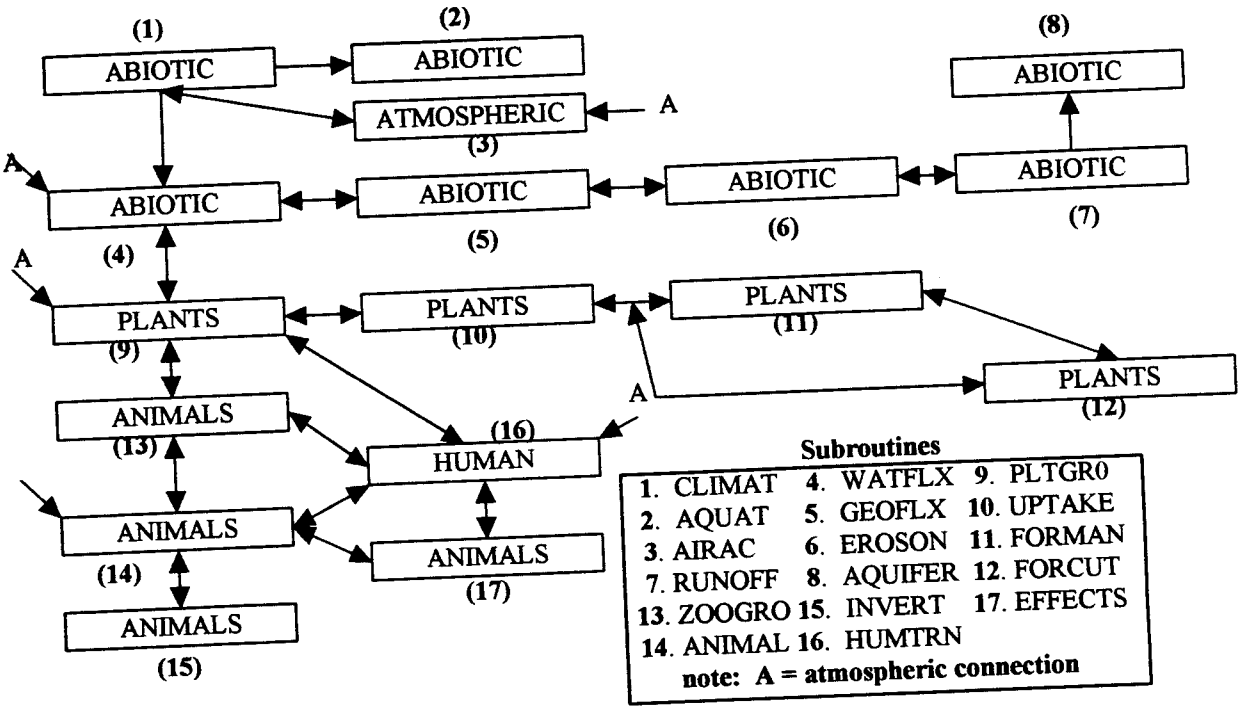


Fig. 3

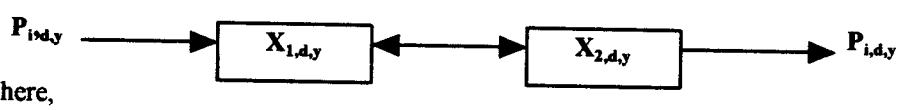
Relationship between model components and model code

which are common to many simulation models. It was also decided that the flow structure of the model would best be represented mathematically as a set of difference equations based primarily on daily time steps which would govern import and export of substances within and among compartments for the flows described previously. An illustration this type of strategy was reported earlier for transuranics[1,13] and is reproduced for all contaminants in Fig. 4 for forage plants where most of the important transport processes were identified. Subsequent development of this strategy to represent material flow has been employed throughout the development of this model inclusive of refinement, generalization, and sensitivity analysis which have been extended to all portions of the model during its development.

3.0. STRATEGY USED FOR MODEL ANALYSIS

This document is written with the view that there is not a single analysis strategy which can be used to adequately document a model. The early work of the United States International Biological Program (US /IBP) in conjunction with the Coniferous Forest Biome[21] suggests a mathematical approach for large simulation models alluded to earlier in this report which identifies flows, and catalogues variables of different types that affect transport mechanisms. However, not all model structure is amenable to this type of strategy, and may become confusing when operations other than flow are integral parts of a model such as BIOTRAN.2. Yet a strict analysis of relationships utilized in a model without emphasis on system construction may also be inadequate in putting into perspective all interrelationships of the system as a whole. Hence, both model construction and a mathematical approach were utilized in an earlier version of this model[1]. This approach has been continued for the current model, although clarifying material has been added when some relationships are being considered to help the reader to visualize the relationship under consideration.

Box-and-arrow diagrams indicating component arrangements will be disassembled into smaller and more manageable portions as necessary, rather than presenting the total module all at once. Also, certain conventions will be adopted: compartments (state variables) will be identified as ($\mathbf{X_i}$, **i=1,n**) for a given pool material (**P**); the units of flow will be specified. Flow diagrams will be represented as in the following example for a given Julian day number (**d**) and year (**y**):



where,

$X_{i,d,y}$ = ith compartment of system on the dth day, and yth year

(1, 2, ..i,d,y) = compartment indices, $X_{1,y}$ $X_{2,y}$ ...$X_{d,y}$, for the 1st, 2nd, and ith compartment on the dth day and the yth year

$P_{i,d,y}$ = gain or loss of pool material from the system for the ith compartment on the dth day and yth year.

Several types of flows can be derived from this example:

$F(P,X_1, d,y)$ = transport of pool material from outside the system into X_1 on the dth day of the yth year (**source term**)

$F(X_1,X_2, d,y)$ = transport of pool material from X_1 to X_2 on the dth day of the yth year

$F(X_2,X_1, d,y)$ = transport of pool material from X_2 to X_1 on the dth day of the yth year

$F(X_2,P,d,y)$ = transport of pool material from X_2 to outside the system on the dth day of the yth year (**sink**)

Assuming that a given day and year are implicit to these derivations, factors affecting flow can be indicated in functional form as:

$$F(X_i, X_m) = f(X_i, X_m, I_j, D_k)$$

where,

X_i = ith state variable (compartment)
X_m = mth state variable (compartment)
I_j = jth intermediate variable
D_k = kth driving variable

These variables will be identified and catalogued (Appendix C), and related to the model code glossary (Appendix D), and code documentation (Appendix H). In addition, justification for the use of any or all factors affecting a flow of pooled material will be presented as required throughout this report. Other variable types and their mnemonic equivalents used in this model:

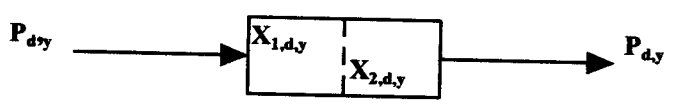
C_l = lth constant

O_m = mth option or user decision variable

L_n = nth logic or boundary variable

S_o = oth stochastic variable

complete the set of variable types and constants required for model documentation and are present in Appendix C. Finally, a flow or exchange which is considered instantaneous (relative to one day) between compartments, sources, or sinks will be represented in this document as:



where the dashed vertical line indicates this type of transport , and all other variables are the same as those previously defined. Furthermore , using the above example, the flow from X1 to X2 and visa versa will be designated as F(1:2) or F(2:1) rather than F(1,2) and F(2,1), respectively. Other conventions employed in this report will be described as they arise. Mneumonic symbols will not be used in describing the flows as they are considered in this manner under code documentation (Appendix G). This analysis will be performed first by identifying flows of pool materials, and then proceed to document the relationships as required. Water, biomass, and contaminant flows will be considered in that order, although parallel relationships between flows will be established as they occur. The order of the documentation will follow as closely as possible ,on a subroutine by subroutine basis, those subroutines which the least amount of dependence on other subroutine output(s), to those which are highly dependent on other subroutines for operation. Variables which are functions of other variables simulated in the code (secondary variables) will be developed within the test as necessary.

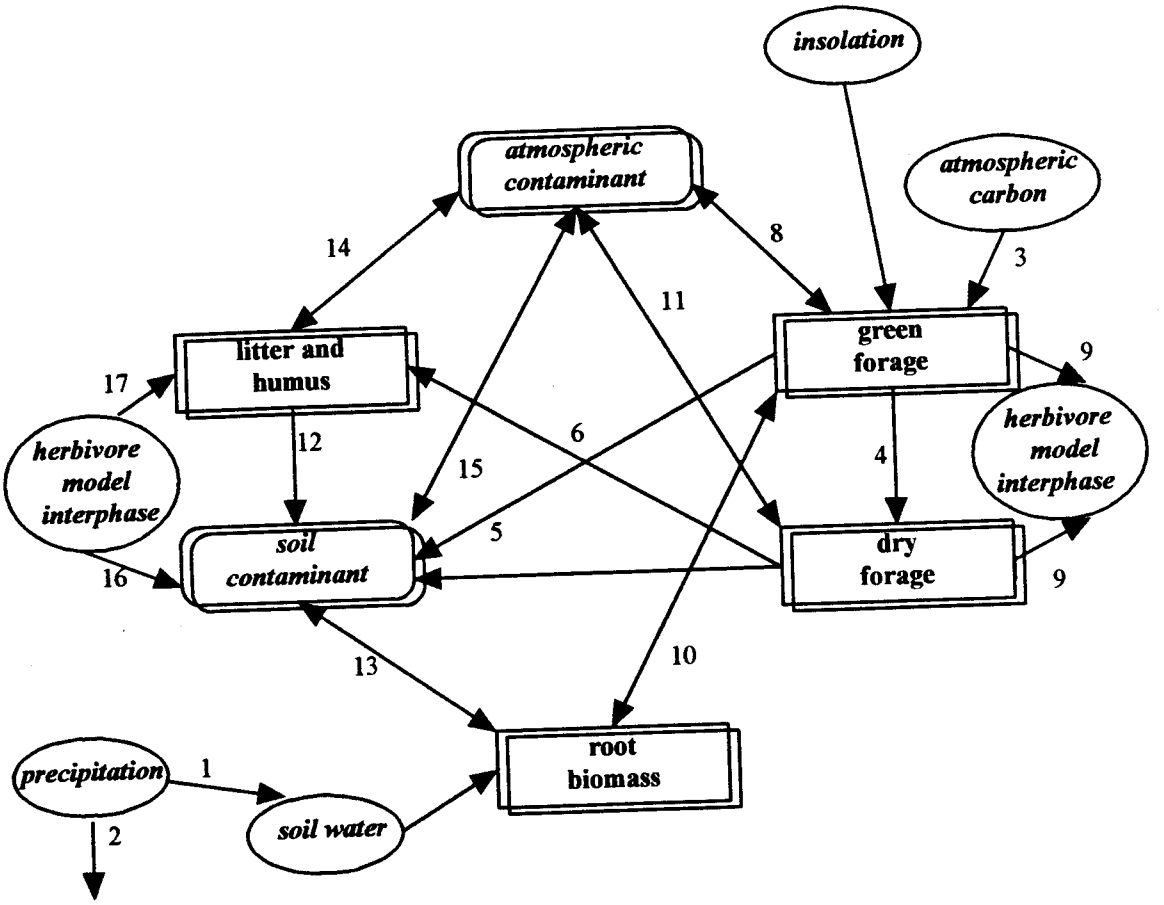


Fig. 4

BIOTRAN.2 flow strategy

note: Arrows indicate transport and exchange rates, numbers identify processes and boxes indicatecompartments or pools. The numbered processes are described in Table 1.

Table 1

Description of Transport Rates Ilustrated in Fig. 4

Transport Rate No.	Description of Process	Controlling Variables Used Determining the Specific Transport Rate
1	Precipitation to soil storage	Precipitation regulation. soil water
2	Surface run-off from soil	Soil water, temperature, live above ground biomass
3	Biomass production from photosynthesis	Soil water, live above ground biomass, temperature, and insolation (solar radiation)
4	Death of live above ground biomassand	Soil water, live above ground biomass, temperature
5	Loss of contaminant from above ground biomass (surface)	Precipitation regulation, time (loss rate)
6	Production of litter and humus from above ground biomass	Above ground dead standing biomass, soil water, temperature, precipitation regulation, above ground standing biomass
7	Loss of contaminant from above ground dead biomass (surface)	Precipitation regulation, time (loss rate)
8	Loss of contaminant from above ground standing biomass (surface) , and deposition from atmosphere	Live above ground biomass, atmospheric contaminant, time, precipitation regulation
9	Loss of above ground standing biomass to grazers	To be described in grazing module
10	Exchange of contaminant and biomass between roots and above ground standing biomass	Live above ground biomass, root biomass, temperature
11	Loss of contaminant from above dead standing biomass, and deposition from atmosphere	Dead above ground standing biomass, time, atmospheric contaminant, precipitation regulation
12	Transport of contaminant to soil from litter and humus	Litter and humus, temperature, and soil water
13	Transport of contaminant from soil to roots and roots to soil	Root biomass, soil contaminant, temperature (root death
14	Exchange of contaminant between litter and humus (surface)	Litter and humus, atmospheric contaminant, precipitation regulation, time
15	Exchange of contaminant between soil and atmosphere	Soil contaminant, atmospheric contaminant, precipitation regulation, time
16-18	Grazing model interphasing	Description in grazing module

note: Time enters as a variable as removal decay constant based on dry deposition or resuspension of a given contaminant for a given day of a given year.

The appendicies included in this report are as follows:

- Appendix A. - Flows of Pool Materials (Moisture, Biomass, Soils, Channel Alluvium, Contaminants)
- Appendix B. - Equations Involving Driving , Stochastic, and Intermediate Variables
- Appendix C - Variables and Their mnemonic Equivalents
- Appendix D - Variable Library (Glossary)
- Appendix E - Ecological Transport Model Code Listing (BIOTRAN.2)
- Appendix F - Plant Type Identification by Number
- Appendix G - BIOTRAN.2 Operating Instructions and examples of input./output.

As an example of how the appendicies may used, consider the first flow in the Moisture Flow Model. In parentheses there will be a reference made to the appendix where the flow is located and the number in the appendix, i.e., (App. A, No. 1). The flow F (P,X_1) or (F_1) can then be found in Appendix A, No. 1., and its mneumonic equivalent in the code can be found in Appendix C, under *flows variable listing*. For a secondary intermediate variable the same strategy will be used except that these variable will be found in Appendix B. In Appendicies A and B the equation involving a flow or secondary intermediate variable will be stated, and the variable used defined. Mneumonic names will not be used in these appendices.

Appendix C contains variables discussed in this document with their mnemonic equivalents in the code as well as the page number of the code listing (Appendix E) where specific variables are defined. Appendix D is a glossary listing of the most important arrayed and non-arrayed variables, together with plot selection criteria and instructions. Appendix F is a listing of plant types which can be internally selected by the user together with instructions for duplicating or creating particular plant type. Appendix G contains information which can be used to simulate sample scenarios, and enable the user to begin construction of new input data sets.

4.0. MOISTURE FLOW AND CLIMATE

4.1. Precipitation

The flow of moisture into the model system is in the form of snow or rain including irrigation water, from perennial stream channels, and from anthropogenic additions such as from waste treatment plants. The flow of moisture out of the system is from evaporation, plant transpiration, surface runoff, lateral flow beneath the root zone, and from stream channel flow. The flow of moisture into the system is illustrated in Fig. 5. The flow of pool material from precipitation $F(P_p,X_{1,1})$ and $F(P_p,X_2)$ are dependent on several variable types:

$$F(P_p,X_{1,1}) = F(P_p,X_2) = f(D_1,I_1,S_1,I_6,S_2,S_3,I_4) \qquad \text{(App. A, No. 1)}$$

whereThe flows simulate transport of moisture from precipitation events to soil and snow compartments during a given Julian day (I_5) of a given year:

$$F(P_p,X_{1,1}) = D_1 I_1 S_1 (1 + I_6 S_3) \qquad I_5 = S_2,\ D_2 > 2.0\,^\circ C \qquad \text{(App. A.,No. 2)}$$

$$F(P_p, X_2) = D_1 I_1 S_1 (1 + I_6 S_3) \qquad I_5 = S_2,\ D_2 \leq 2.0\,^\circ C$$

$$= 0 \qquad I_5 \neq S_2$$

$F(P_p,X_{1,1})$ = transport of moisture from rain events to soil, mm/day
$F(P_p,X_2)$ = transport of moisture from melting snow to soil, mm/day
D_1 = mean annual precipitation, inches
I_1 = January through December monthly precipitation fractions

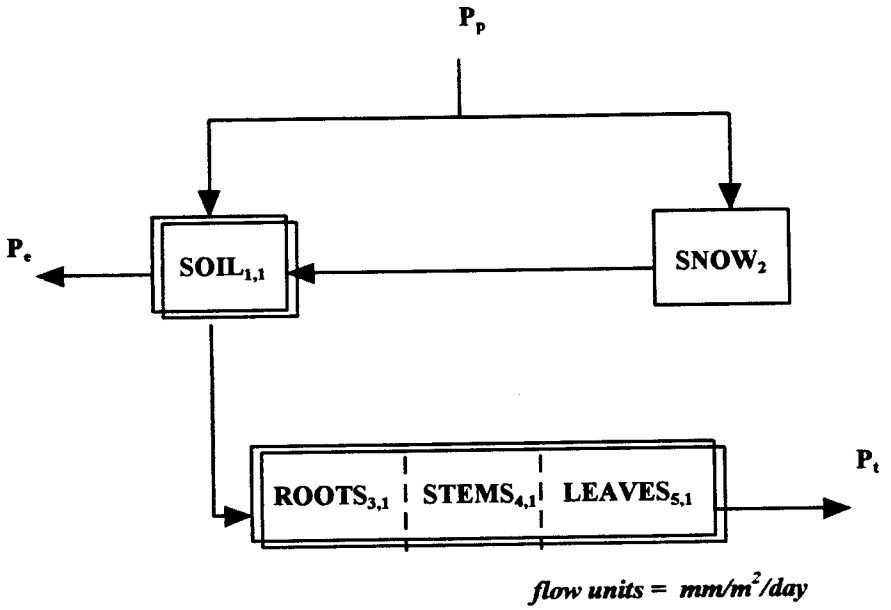


Fig. 5

Moisture Flow Diagram

S_1 = normalized random number fraction obtained from a normal distribution ($m = 0, \sigma = 1$) to assign precipitation amount to a given precipitation event for a given month (see App. B, IV, No.1)
I_6 = coefficient of variation for precipitation per event of reference site
S_2 = product of random number drawn from an even distribution (range 0 to 1) with the number of days in a given month to make a precipitation assignment (see App. B, IV, No.2)
I_5 = julian day number (1 - 365)
S_3 = random number drawn from a normal distribution ($m = 0, \sigma = 1$) to estimate precipitation increment/decrement for a given rainfall event
I_4 = mean number of rainfall events in a given month totalling more than 0.1 inches

The flow of moisture to the soil surface ($X_{1,1}$) or the snow compartment (X_2) is based on the total number of precipitation events occurring in a given area which exceed 0.1 inches (I_4) ,[22] a precipitation fraction by month (I_1), [23] and an estimated coefficient of variation [24] for the study area.. The stochastic intermediate variables S_1, S_2, and S_3 are derived elsewhere (App B., IV,1-3), and are used to derive precipitation pattern, precipitation amounts for each precipitation event, and to adjust the derived precipitation amount using the coefficent of variation for precipitation at the site (input). S_1 and S_3 utilize random number selection from a normal distribution ($m=0, \sigma =1$), whereas day selection for precipitation events utilizes random selection from an even distribution (range 0 to 1). S_3 is also used to vary temperature about the mean in random fashion

4.2. Temperature

The driving variable (D_2) or mean daily temperature for a site on given Julian day is based on a sinusoidal function; [25,26]

$$D_2 = [(I_7 - I_8)/2] \text{ Sin } (2\pi\{ I_5 - I_9\}/365) + 1] + I_8 \qquad \text{(App. B, II, No. 2)}$$

D_2 = mean daily temperature, °F
I_7 = maximum mean daily temperature on a yearly basis, °F (input)
I_8 = minimum mean daily temperature on a yearly basis, °F (input)
I_9 = lag period to maximim mean daily temperature on a yearly basis, days (input)
I_5 = julian day number (1 - 365)

The temperature is corrected for elevational and latitudinal differences between the reference and test site, and the temperature is varied stochastically on a monthly basis:[13]

$$D_2 = D_2 I_{10} (I_{11} - I_{13}) + S_3 I_{14} \qquad \text{(App. B, II, No. 3)}$$

I_{10} = environmental lapse rate, °F per foot from reference site elevation (input)
I_{11} = reference site elevation, feet amsl (input)
I_{12} = elevation of test site, feet amsl (input)
I_{13} = correction for latitude difference between reference and test site, °F (input)
I_{14} = standard deviation from mean monthly temperature for all months combined, °F (input)
S_3 = random number drawn from a normal distribution ($m = 0, \sigma = 1$)

and finally converted to degrees Celsius (App. B, II, No.4). The latitudinal correction (I_8) must be expressed in °F , and may be estimated by assuming that 1° latitude translates into about 69 miles north or south translation at 40° latitude. Furthermore, it is assumed that there is a change of about 1° F per 1° change in latitude between reference and study sites. This correction is not generally used since most references stations are generally closer than 1° latitude .

The decision to represent temperature trends on a monthly basis was based on observation of weather reports; As this analysis was not exhaustive, other trend lengths lesser or greater than one month could have been used but are not as easily incorporated into the model code.

The moisture flow, mm/day, from snow cover to the soil surface, $F(X_2, X_1)$ is based on an estimated snowmelt function: [27]

$$F(X_2, X_{1,1}) = C_0 D_2 + C_1 \qquad \text{(App. A, No. 3)}$$

$F(X_2, X_{1,1})$ = transport of moisture from snow to top layer of soil, mm/day
C_o = snowmelt regression coefficient, 11.28 mm $^\circ C^{-1}$
C_1 = snowmelt intercept, 1.1128 mm

The amount of snowmelt is limited by the amount of snow present, mm, in the snow compartment X_2, if the regression equation estimate exceeds this value.

4.3. Evaporation and Solar Radiation

Evaporation of moisture from the soil surface layer ($X_{1,1}$), mm/m^2 /day, and from transpiration losses of plant leaf compartment, gdwt/ m^2, (X_5) is governed by the potential evapotranspiration (PET) of the site [4]:

$$I_{17} = (1.28 I_3 I_{15}) / (I_3 + I_{16}) \qquad \text{(App. B, I, No. 1)}$$

I_{17} = potential evapotranspiration (PET) from soil surface, mm/day
I_3 = slope of psychrometric saturation line, mbars/ $^\circ$ C
I_{15} = net daily solar radiation absorbed by land surface at mean air temperature, Langelys
I_{16} = psychrometric constant, mbars/ $^\circ$ C

where the slope of the psychrometric saturation line (I_3), mbars/ $^\circ$ C, is:

$$I_3 = [5303 / (D_2 + 273)^2] \exp[21.255 - 5304 / (D_2 + 273)] \qquad \text{(App. B, I, No. 2)}$$

I_3 = slope of psychrometric saturation line, mbars / $^\circ$ C
D_2 = mean daily temperature, $^\circ$ C

and the net solar radiation, Langleys, absorbed by the land surface (I_{15}) at mean daily insolation, Ly, is:

$$I_{15} = (1 - C_2) D_3 / 58.3 \qquad \text{(App. B, I, No. 3)}$$

I_{15} = net solar radiation, ly, absorbed by the regional land surface
C_2 = albedo of site (set in code to 0.2)
D_3 = solar radiation at site, Langleys/ day

and the psychrometric constant (I_{16}), mbars/ $^\circ$ C, is estimated as:

$$I_{16} = 0.0006595 I_{18} \qquad \text{(App. B, I, No. 4)}$$

I_{16} = psychrometric constant, mbars / $^\circ$ C
I_{18} = site barometric pressure, mbars

The albedo (C_2) of the land surface has been set at 0.2 in the model, where as the solar radiation (D_3), Langleys/day, is estimated by the following relationship:

$$D_3 = [(I_{24} - I_{25})/2] \text{ Sin } (2\pi\{ I_5 - I_{26}\}/365) + 1] + I_{25} \qquad \text{(App. B, II, No. 3)}$$

D_3 = total solar radiation intensity at site, Langleys / day
I_5 = Julian day number
I_{24} = maximum yearly solar radiation intensity at site, Langleys / day (input)
I_{25} = minimum yearly solar radiation intensity at site, Langelys / day (input)
I_{26} = lag period to maximum insolation, days

and the barometric pressure, mbars, at the site (I_{18}) is estimated as:

$$I_{18} = 1013 - 0.1152 (0.3048I_{12} + 0.00000544 (0.3048I_{12})^2 \qquad \text{(App. B, I, No. 5)}$$

I_{12} = site elevation, feet, amsl

The major difference between the PET methodology used in this model when compared to the USDA's CREAMS model[21] methodology is that the psychrometric constant is based on the barometric pressure for the site has been related exclusively to site elevation (I_{12}) as a best estimate.

The evaporation or flow of moisture from the soil surface to the atmosphere F (1,P), mm/m^2 /day, is represented in the model by an adjusted PET that has allowed transpiration losses from plants , F (5,P), to satisfy PET requirements from soil layers beneath the surface layer. Evaporation of moisture is modeled to affect only the top soil layer which is always set a 20 mm in depth. This layer is also used for modeling resuspension and deposition of particulates to be described later in this report. Hence evaporation from the soil surface is modeled as:

$$F (X_{1,1} ,P_e) = I_{17a} \exp (-0.4I_{19}) \qquad \text{(App. A. No. 4)}$$
$$F (X_{1,1} ,P_e) = (X_{1,1} - 1) \exp (-0.4I_{19}) \qquad I_{17a} > (X_{1,1} - 1)$$
$$F (X_{1,1} ,P_e) = 0 \qquad (X_{1,1} - 1) \leq 0$$

$F (X_{1,1}, P_e)$ = transport of moisture from soil surface (evaporation), mm/ m^2 / day
I_{17a} = I_{17} (PET) adjusted for transpiration losses and low soil moisture in top layer
I_{19} = leaf area index, m^2/m^2, (LAI)
$X_{1,1}$ = soil moisture in surface layer, mm

where (I_{17a}) is the PET (I_{17}) of the site which has been adjusted for plant transpiration losses, and has been bounded so that the surface layer ($X_{1,1}$) does not fall below 5% soil moisture content (or 1 mm for the modeled 20 mm surface layer). The flow is also affected by the leaf area index , m^2/m^2, (I_{19}) which is estimated from the leaf biomass density, gdwt/ m^2, (X_5):

$$I_{19} = 0.013X_5 \qquad \text{(App. B., I, No. 6)}$$

I_{19} = total leaf area index (LAI), m^2/m^2
X_5 = total leaf biomass density, gdwt/ m^2

The modeling strategy is similar to the CREAMS model except for the boundary conditions imposed and the order of allocation of PET[21]. This adjustment was modeled because BIOTRAN.2 structures more than one soil layer for moisture allocation, and because of the steep moisture gradient created at the

surface by evaporation as described by Hillel[22]. Finally, the leaf area index (LAI) estimate described above uses only the live or transpirationally active leaf biomass present at the site.

4.4. Plant Transpiration

The flow of moisture through plant transpiration ,mm/m^2/ day, F (X_5,P_t) is modeled as the total amount of moisture evaporated from successive soil layers with exception of the first layer in the rooting zone of a given plant type:

$$F(X_5,P_t) = \sum_{i=2}^{I_{20}} [(X_{j,i} - I_{22}) / (I_{23} - I_{22})] (X_{j,i} - I_{22}) I_{21} I_{19} \qquad \text{(App. A, No. 5)}$$

$F(X_5,P_t)$ = total transpiration, mm/m^2/ day, for the jth plant
I_{20} = number of soil layers specified for the jth plant (input or estimated)
Xj,i = soil moisture content, mm, for the ith soil layer of the jth plant
I_{22} = soil moisture content, mm, for the ith soil layer of the jth plant at wilt point (input)
I_{23} = soil moisture content, mm, for the ith soil layer of the jth plant at holding capacity (input)
I_{19} = leaf area index, see above
I_{21} = root penetration fraction (0 - 1) of the jth plant into the ith soil layer

If the number of soil layers is estimated in the model, then 250 mm soil thicknesses (see I_{37}) are assumed. Furthermore, the first soil layer is divided into a surficial 20 mm layer, and a 230 mm layer. If a user wishes to use this default option (setting $I_{20} = 0$), then a soil depth value (see I_{56}) should be used as input which is a multiple of 250mm.

The upper boundary for the flow, F (X_5 ,P_t), is set by the PET of the site, and moisture removal is modeled from each successive layer until the PET is reached or until the last layer is reached. If a given layer does not satisfy demand , then deficit is provided by the next layer ..ie.. Also, the withdrawal is limited by the excess of moisture above the wilt point, and the degree of root penetration into the soil layer in question. This type of strategy is supported by studies on evapotranspiration rate studies of conifers in the Cascade Range[23]. It should be noted that the flow of moisture from all (n-1) soil layers into the roots of a plant, and ultimately through the leaves is numerically equal to F (X_5 ,P_t):

$$F\left(\sum_{i=2}^{n} X_{1,i} , X_3{:}X_4{:}X_5\right) = F(X_5,P_t) \qquad \text{(App. A., No. 6)}$$

n = number of soil layers
F ($X_{1,i}$,X_3:X_4:X_5) = flow of moisture from the ith soil layer to the leaf surface via the plant roots and stems

although they may not occur at the same time in nature. Transpiration losses for all plants under succession are normalized to the same land area because they are assumed compete for this resource. This modification to (F_6) above will be discussed and derived in the when plant succession is under consideration.

The simulation of the root penetration fraction influencing moisture removal from a given soil layer (I_{21}) is estimated from the ratio of twice the root penetration depth ,mm, (I_{27}) into the soil horizon to the bottom depth of a given soil layer on a given simulation day . The actual rate of root penetration varies with plant types simulated , time of year, ..etc.[5]. The root penetration fraction is estimated as:

$$I_{21} = 2I_{27} / I_{28} \qquad \text{(App. B.,I, No. 7)}$$

I_{21} = root penetration fraction for moisture withdrawal from a given soil layer

I_{27} = vertical root penetration into soil from the surface, mm
I_{28} = depth to the bottom of a given soil layer, mm

The extent of vertical root penetration into a given soil (I_{27}), mm, is estimated as:

$$I_{27} = \sum_{j=1}^{I_{29}} \sum_{i=1}^{365} C_4 \qquad \text{(App. B.,I, No. 8)}$$

I_{27} = vertical extent of root penetration into a given soil, mm
I_{29} = julian year number of specific year of simulation
C_4 = mean root growth rate in the vertical direction, mm/day

The vertical penetration into the soil is reset to 0.0 for annual plants 30 days after the end of the frost-free period, and remains at this value until 30 days prior to the beginning of the frost-free period. The same conditions apply to un-irrigated annual crops except that root growth remains at 0.0 until the first irrigation is applied. Root growth in irrigated annual crops ceases after the final harvesting and remains in this condition until the first irrigation application. The mean root growth rate in the vertical direction (C_4) has been estimated on the basis of the mean life-span of a given plant type (C_3) :

$C_4 = 12.7$	$C_3 \leq 1$	(App. B.,VI)
$C_4 = 7.62$	$1 < C_3 \leq 50$	
$C_4 = 2.54$	$50 < C_3 \leq 150$	
$C_4 = 1.0$	$C_3 > 150$	

C_4 = root growth in the vertical direction, mm /day
C_3 = mean life-span, years, of the plant type simulated

The four categories of root growth rates presented above roughly correspond to annuals grasses and herbs, perennial grasses and herbs, shrubs, and trees, respectively. Since these root growth rate estimates were derived from model tuning experiments from simulated growth experiments, and semi-empirically from root distribution studies[24], there is a need to validate such estimates for application in a generic fashion. Finally, the vertical root length remains constant when it equals the rooting depth specified as input by the user.

4.5. Soil Moisture Transport

Soil moisture remaining after adjustment for PET and surface runoff losses (to be described later in this report) is modeled as infiltrating from the top layer to each successsive layer taking into account matrics effects due to unsaturated conditions. The transport of moisture from the ith soil layer to the next successive soil layer is estimated as [22]:

$$F(X_{1,i}, X_{1,i+1}) = [\overset{i}{I}_{30} + (\overset{i}{I}_{30}/2)(\overset{i+1}{I}_{31} - \overset{i-1}{I}_{31})]\, I_{37i} \qquad i \neq 1 \qquad \text{(App. A., No. 7)}$$

$$F(X_{1,i}, X_{1,i+1}) = [\overset{i}{I}_{30} + \overset{i}{I}_{30}\overset{i+1}{I}_{31}/2]\, I_{37i} \qquad i = 1$$

$F(X_{1,i}, X_{1,i+1})$ = transport of moisture from the ith soil layer to next successive layer, mm /day

$\overset{i}{I}_{30}$ = weighted soil moisture conductivity (κ) , mm / day/ mm -soil, based on rock, sand, silt, and clay composition of the ith soil layer obtained by regression analysis of information in Hillel [22] as a functon of soil volumetric wetness (θ), $p = 0.01$, $r > 0.9$.
I_{37i} = thickness of ith soil layer, mm

$I_{31}^{i+1;\,i-1}$ = weighted soil matric potential (Ψ) , mm / mm-soil based on rock, sand, silt, and clay composition of the upper and lower layer adjacent to the ith soil layer obtained by regression of information in Hillel[22] as a function soil volumetric wetness (θ), p = 0.01, r > 0.9.

The weighted soil moisture hydraulic conductivity for (I_{30}^{i}, or κ) for the ith soil layer as a function of soil volumetric wetness (θ) is estimated as:

$$
\begin{aligned}
I_{30}^{i} &= 1.56e06\ I_{32}^{5.27} && \text{rock} \qquad \text{(App. B.,I, No. 9)}\\
&= 1.56e05\ I_{32}^{5.27} && \text{sand}\\
&= 6.99e04\ I_{32}^{6.4} && \text{silt}\\
&= 1.04e04\ I_{32}^{9.2} && \text{clay}
\end{aligned}
$$

where :

I_{32} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer.

The soil moisture hydraulic conductivity for rock is assumed to be ten times greater than that for sand. Similarly, the weighted soil moisture matric potential, mm/mm, (I31, or ψ) is also estimated as a function of soil volumetric wetness (θ) as:

$$
\begin{aligned}
I_{31}^{i} &= 50\ I_{32}^{0.5} && \text{rock} \qquad \text{(App. B.,I, No. 10)}\\
&= 50\ I_{32}^{1.5} && \text{sand}\\
&= 40\ I_{32}^{2.6} && \text{silt}\\
&= 41.4\ I_{32}^{4.1} && \text{clay}
\end{aligned}
$$

I_{32} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer

I_{30}^{i} = weighted soil moisture conductivity (κ) , mm / day/ mm -soil, based on rock, sand, silt, and clay composition of the ith soil layer obtained by regression analysis of information in Hillel [22] as a functon of soil volumetric wetness (θ), p = 0.01, r > 0.9.

The matric potential for rock has also been estimated to be considerably smaller than for sand from tuning experiments. finally, the weighted soil moisture conductivity ($\underline{\kappa}$, $\underline{I}_{30}^{i}$) and matric potential (ψ, $\underline{I}_{31}^{i}$) for ith soil layer is estimated as a linear combination based on rock, sand , and silt content for that layer:

$$\underline{I}_{30}^{i} = I_{33}\ 156e06\ I_{32}^{5.27} + I_{34}\ 1.56e05\ I_{32}^{5.27} + I_{35}\ 6.99e04\ I_{32}^{6.4} + I_{36}\ 1.04e04\ I_{32}^{9.2}$$

and:

$$\underline{I}_{31}^{i} = I_{33}\ 50\ I_{32}^{0.5} + I_{34}\ 50\ I_{32}^{1.5} + I_{35}\ 40 I_{32}^{2.6} + I_{36}\ 41.4\ I_{32}^{4.1} \qquad \text{(App. B.,I, Nos. 11,12)}$$

$\underline{I}_{31} = I_{33}\ 50\ I_{32}^{0.5} + I_{34}\ 50\ I_{32}^{1.5} + I_{35}\ 40 I_{32}^{2.6} + I_{36}\ 41.4\ I_{32}^{4.1}$
$\underline{I}_{30}$ = weighted mean soil hydraulic conductivity ($\underline{\kappa}$) , mm /day
I_{33} = rock fraction in ith soil layer (input)
I_{34} = sand fraction in ith soil layer (input)
I_{35} = silt fraction in ith soil layer (input)
I_{36} = clay fraction in ith soil layer (input)

The flow of moisture in either a downward or upward direction is the maximum potential flow assuming adequate soil moisture in a given soil layer. Beginning with the top layer and proceeding to to the bottom layer, a cummulative boundary moisture flow is estimated for each layer. Moisture flow is then adjusted to lower flow values if limiting soil moisture values are present. Also, moisture flow is adjusted to prevent any but the top layer from exceeding the holding capacity of that layer; this adjustment is accomplished by utillizing a sequential adjustment of moisture flow beginning with the bottom layer and proceeding to the second layer. Ponding of water in the top layer is simulated if all lower soil layers are saturated to holding capacity. Also, a lower moisture content of 5% moisture content is used as a boundary condition for all soil layers. A final adjustment on upward flow of moisture is simulated when the volumetric wetness of the upper layer exceeds the adjacent lower layer. In such cases, the upward flow of moisture is bounded so that the volumetric wetnesss is balanced under these conditions, or is excluded to prevent this condition from occurring in the simulation. This type of adjustment is the result of tuning experiments used in stabilizing model operation in this regard especially when relatively large volumetric wetness differences exist between adjacent soil layers.

The loss of soil moisture from the bottom soil layer is set at 0.5% / day of the available moisture in the bottom layer only at holding capacity; however, when subsurface return flow is activated in the simulation, then this parameter is set to 0.0%. Any losses of moisture by this process is modeled as a sink, whereas the presence of a a return flow process is integrated with watershed moisture flow to be discussed later in this report. The coupling between precipitation, evapotranspiration, soil moisture flow, surface runoff of un-infiltrated moisture, and lateral flow is show in Fig. 5. The latter two types of flow are discussed in the following sections of this report.

4.6. Surface Runoff and Infiltration

The simulation of moisture losses due to surface runoff and subsurface return (lateral) flow from a study site as a result of precipitation events is accomplished by closely coupling these events to soil moisture infiltration. Hence, the infiltration model used for this purpose is of necessity a mass-balance accounting of residual moisture after surface runoff losses have been taken into account. This approach was taken because because of the lack of storm structured data (intensity and duration) that would be required for modeling this infiltration in a more dynamic way. Thus, total precipitation per event independent of intensity and duration as well as other methodologies are used for simulating surface runoff events in this model [21,25,26]. Surface runoff is estimated by the use of *curve numbers*, a technique which takes into account soil type, plant cover, vegetative type, and antecedent soil moisture[25]. The data from this citation were used to formulate multiple and simple linear regression equations ($p = 0.01$, $R^2 \cong 0.8$) to estimate the normal soil moisture condition curve number:

$$I_{38} = I_{39} - I_{40}I_{41} - I_{42}C_3 \qquad \text{(App. B.,I, No. 13)}$$

I_{38} = the normal soil moisture condition curve number
I_{39} = plant crown cover factor
I_{40} = plant crown cover-conductivity interaction factor, hours/ inch
I_{41} = soil moisture infiltration rate, inches/ hour (input)
I_{42} = plant crown cover -vegetation type interaction factor, year^{-1}
C_3 = mean life-span of plant type, years

where,

$$I_{39} = 96 - 16I_{43}$$

$$I_{40} = 9 + 5.9I_{43}$$

and,

$$I_{42} = 0.8 + 0.04 I_{43} \qquad \text{(App. B., I, No. 14,15,16)}$$

I_{39} = plant crown cover factor
I_{40} = plant crown cover-conductivity interaction factor, hours/ inch
I_{42} = plant crown cover -vegetation type interaction factor, $year^{-1}$
I_{43} = plant fractional crown cover, m^2/m^2 (also see App. B.,I, No. 197 for trees)

The plant fractional crown cover (I_{43}) is estimated from a one year running average net productivity for annuals , and a 10 year running average net productivity for all other types of plants. However, if the forest management module FORMAN is accessed in the code, then this estimate is made from regression analysis of bole diameter to crown diameter, together with an estimate of the number plants (trees) in a stand. Since both productivity estimates for plants and tree stand structuring are treated in the next section of this report (Flow of Biomass), the reader should consult this section for a more complete descriiption of these parameters. The following relationships are used to estimate plant fractional crown cover when FORMAN access gate (L_2) is closed:

$$I_{43} = [\, 1 - \exp(-0.00154\, I_{44}\,)] \qquad L_2 \le 1 \qquad \text{(App. B.,I, No. 17)}$$

$$I_{45} = \sum_{i=1}^{n} 0.0929\pi\, (\, I_{46} I_{47i} \,/\, 2\,)^2\, (\, I_{48i} / 4047\,) \qquad L_2 > 1$$

I_{43} = plant fractional crown cover, m^2/m^2

I_{44} = 10 year running average net cummulative productivity, gdwt / m^2/ year (cummulative (daily productivity for annual plants (see App. B.,I, No. 82)
I_{45} = current number of tree diameter classifications simulated on a given year/day
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (input)
I_{47i} = mean of ith bole diameter class, inches (see App. B.,I, No. 181)
I_{48i} = number of trees in ith bole diameter class (see App. B., I, No. 182)

The plant fractional crown cover estimate when (L_2) is less than or equal to one is made on the basis of tuning experiments with range grasses and conifers combined with field observations from known leaf biomass densities of these plants, however, this estimate becomes less precise as leaf biomass densities approach 500 gdwt / m^2, and the tree structure method utilized in the FORMAN module is recommended for densities above this value[15].
Mean lifespans (C_3) for specific plant types are estimated in BIOTRAN.2 as are daily crown cover fractions (I_{43}) . Moisture infiltration rates (I_{41}) are supllied as input to the model. Also, plant types with mean lifespans less than or equal to 15 years are treated set equal to the latter for normal soil moisture condition curve number (I_{38}) determination. Hence, this strategy places all annual and perennial grasses and herbs in to one lifespan category, all shrubs into a second , all shorter lived-trees into a third, and all long-lived trees into a fourth category for curve number determination. More study of these relationships is needed to obtain more accurate formulation of the curve number by this methodology.
An estimate of the dry antecedent soil moisture condition *curve number* (I_{49}) from the normal soil moisture condition is estimated as is in the CREAMS model[21]:

$$I_{49} = -16.91 + 1.348\, I_{38} - 0.01379\, I_{38}^2 + 0.0001177\, I_{38}^3 \qquad \text{(App. B.,I, No. 18)}$$

I_{49} = the dry antecedent soil moisture condition curve number

I_{38} = the normal soil moisture condition curve number

An estimate of the dry antecedent soil moisture condition *curve number* (I_{49}) from the normal soil moisture condition is estimated as is in the CREAMS model[21]:

$$I_{49} = -16.91 + 1.348\, I_{38} - 0.01379\, I_{38}^2 + 0.0001177\, I_{38}^3 \qquad \text{(App. B.,I, No. 18)}$$

I_{49} = the dry antecedent soil moisture condition curve number
I_{38} = the normal soil moisture condition curve number

The present version uses the impervious fraction to implicitly adjust infiltration parameters in the WATFLX module, thus making the above correction unnecessary. More details of this adjustment will be described later in this report. The dry antecedent soil moisture curve number (I_{49}) is used to estimate a maximum soil retention parameter (I_{51}) ,mm :

$$I_{51} = 25.4\, [\, (1000/ I_{49}) - 10] \qquad \text{(App. B.,I, No. 20)}$$

I_{51} = maximum soil moisture retention parameter, mm
I_{49} = dry antecedent soil moisture curve number

The estimated soil moisture retention parameter (I_{52}), mm, over all soil layers for a given day at a given site for a given plant type can then be estimated as:

$$I_{52} = \sum_{i=2}^{I_{20}} (I_{51} I_{53i} [(I_{23i} - X_{1,i}) / I_{23i}] \qquad \text{(App. B.,I, No.21)}$$

I_{20} = total number of soil layers in the rooting zone of a given plant type (input or estimated)
I_{51} = maximum soil moisture retention parameter, mm
I_{52} = soil moisture retention parameter over all soil layers, mm
I_{53i} = ith soil layer weighting factor
I_{23i} = ith soil layer holding capacity, mm
$X_{1,i}$ = ith soil layer moisture content, mm

where,

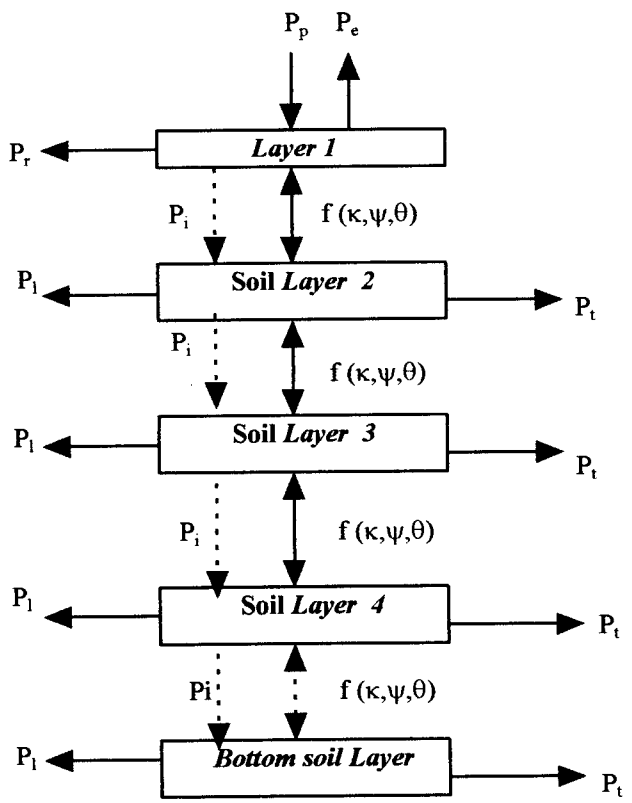
$$I_{53i} = I_{54} \exp(-4.16\, I_{55i} / I_{56}) \qquad \text{(App. B.,I, No. 22)}$$

and,

$$I_{54} = \sum_{i=2}^{I_{20}} [\exp (-4.16 I_{55i} / I_{56})]^{-1} \qquad \text{(App. B.,I, No. 23)}$$

I_{53i} = ith soil layer weighting factor
I_{54} = a parameter which normalizes the sum of all soil weighting factors to 1.0
I_{55i} = depth to bottom of the ith soil layer of the soil profile
I_{56} = total depth, mm, of soil profile for the rooting zone of a given plant type (input)

The SPUR model criteria[27] were used for estimating the soil-layer moisture weighting factors as well as the algorithm for estimating the weighted soil moisture parameter (I_{52}).



P_p = precipitation (source)
P_e = evaporation from soil surface (sink)
P_r = surface runoff (sink)
P_t = plant transpiration (sink)
P_l = subsurface lateral flow (sink)
P_i = infiltration (source)

Fig. 6

Moisture Flow Modeling Strategy

If the simulated precipitation event (P_p) on a given day exceeds 20% of the weighted soil moisture (I_{52}), then daily surface runoff (P_r, mm, (according to the Soil Conservation Services, SCS) is estimated as:

$$F(X_{1,1}, Pr) = (P_p - 0.2I_{52})^2 / (P_p + 0.8I_{52}) \qquad P_p > 0.2I_{52} \qquad \text{(App. A., No. 8)}$$
$$= 0.0 \qquad P_p \leq 0.2I_{52}$$

$F(X_{1,1}, P_r)$ = daily moisture runoff, mm, from the surface of soil
P_p = daily precipitation event, mm
I_{52} = weighted soil moisture parameter, mm

The difference between precipitation and surface runoff is treated as infiltration moisture (P_i), mm:

$$F(X_{1,1}, P_i) = P_p \qquad P_p \leq 0.2I_{52} \qquad \text{(App. A., No. 9)}$$
$$= P_p - P_r \qquad P_p > 0.2I_{52}$$

When infiltration is simulated, then beginning with the upper soil layer, each successive soil layer is increased to holding capacity if necessary until the amount of moisture added equals (P_i). Ponding of water is allowed to occur when the soil moisture content of the top soil layer is increased above holding capacity (I_{23}) should the lower layers become saturated to holding capacity before (P_i) is satisfied. Ponding is treated as additional surface runoff on the following day (which increases available water for transpiration, P_t) until mass balance has been achieved.

4.7. Lateral Flow

Sub-surface flow (P_l) which is also called return or lateral flow is modeled only when a watershed is being simulated ($I_{57} > 0$) using a simplified modification of Darcy's equation[22] : only those soil layers where the soil moisture content ($X_{1,i}$) exceeds field capacity (I_{58}) of that layer is modeled to contribute return flow from a given plant type. Return flow for the ith soil layer for the jth plant type in a given region is estimated as:

$$F(X_{1,i}, P_l) = 2\, I_{59j}\, I_{41}\, I_{60} \qquad I_{57} > 0;\ X_{1,i} > I_{58} \qquad \text{(App. A. No.10)}$$
$$= I_{66i} \qquad F(X_{1,i}, P_l) > I_{66i}$$
$$= 0 \qquad I_{57} = 0;\ X_{1,i} \leq I_{58}$$

$F(X_{1,i}, P_l)$ = potential sub-surface return flow from the ith soil layer with a jth plant type, ft^3 / day
2 = conversion factor: 1 inch/hr = 2 feet/day
I_{59j} = crossectional area, ft^2, for return flow into a channel from the jth plant type
I_{41} = soil moisture conductivity, inches / hr, for region (input)
I_{60} = fractional slope of regional land surface (input)
I_{66i} = total water available in ith soil layer with a jth plant type above field capacity, ft^3

where,

$$I_{59j} = 5280\, I_{61j}\, I_{62j}\, I_{63} \qquad \text{(App. B.,I, No. 24)}$$

I_{59j} = crossectional area, ft^2, for return flow into a channel from the jth plant type
I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{62j} = total depth, feet, of the soil horizon for the jth plant type in a given region
I_{63} = total length, miles, of a given region along a receiving channel (input)

and,

$$I_{61j} = (I_{43j} + 0.01 / I_{64}) / I_{65}$$ (App. B.,I, No. 25)

I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{43j} = crown cover fraction for jth plant type in a given region
I_{64} = number of plant types in a region excluding those where the area specified is not equal to the regional area such as for irrigated or non-irrigated plants under cultivation
I_{65} = total cover fraction for all plant types considered (I_{64}) in a given region

Finally, the amount of available soil moisture above field capacity , ft^3, from the ith soil layer (I_{66i}) of the jth plant type of a given region is estimated as:

$$I_{66i} = (X_{1,i} - I_{58i}) I_{67j} / 304.8$$ (App. B.,I, No. 26)

I_{66i} = total water available in ith soil layer with a jth plant type above field capacity, ft^3
$X_{1,i}$ = total soil moisture, mm, in the ith soil layer of the jth plant type in a given region
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of the jth plant type in a given region (input)
I_{67j} = estimated crown cover area, ft^2, of the jth plant in a given region

The estimated crown cover (I_{67j}) of a given plant in a given region is estimated as:

$$I_{67j} = 43562\, I_{61j}\, I_{68}$$ (App. B.,I, No. 27)

I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{68} = area, acres, of the region containing the jth plant type (input)

Thus, the total lateral or subsurface later flow from a given region into a receiving channel (I_{69}) is the sum of all contributing soil layers (I_{20}) from all plant types (I_{68}) in a given region:

$$\Sigma F_a (X_{1,i}, P_l) = \sum_{j=1}^{I_{69}} \sum_{i=2}^{I_{20}} F (X_{1,i} , P_l)$$ (App. A., No.11)

$\Sigma F_a (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, ft^3 /day, into a receiving channel
I_{69} = total number of plants in a given region (input)
I_{20} = number of soil layers specified for the jth plant (input or estimated)
$F (X_{1,i} , P_l)$ = sub-surface return flow from the ith soil layer with a jth plant type, ft^3 / day , (see App. A. No. 10)

Similarly, the total lateral or subsurface return flow from a given region expressed in inches/ day is:

$$\Sigma F_b (X_{1,i}, P_l) = 12\, \Sigma F_a (X_{1,i}, P_l) / (43564\, I_{68})$$ (App. A., No. 12)

$\Sigma F_b (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, inches/ day, into a receiving region
$\Sigma F_a (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, ft^3 /day, into a receiving channel
I_{68} = area, acres, of the given region (input)

4.8. Channel Flow

After having established the flow of moisture from a given region(s) by surface runoff and return flow processes , the next process which is modeled is the channelization of this moisture to a specified drainage system; the latter is specified by user input instructions. The corresponding interconnections of specific channels or channel segments to form the drainage pattern are also supplied as input. If a given lower region is receiving surface runoff from another upper region(s), then the additional runoff imcrement is treated as additional precipitation and processed as potential surface runoff described previously. Although this regional apportionment structure appears cumbersome from a mass-balance perspective, the model appears to operate correctly in this respect. Similarly, return flow increments are added to the soil layers (starting with the bottom layer and proceeding upward) below each vegetation type in proportion to the total number of vegetation types in the receiving region. Each consecutive soil layer is allowed to reach holding capacity before allowing moisture to enter upper soil layers withoin the same soil profile. Residual moisture remaining after all soil layers for each vegetation type are at holding capacity is allocated to the surface layer or to pond and be treated as surface runoff on the next consecutive day. Other parameters such as mean temperature,precipitation, and the total area combined of all regions are also utilized in making channel flow estimates.

The volume of moisture entering specific channels from specified regions, from other channels, or channel segments, must be adjusted for transmission losses before downslope channels or channel segments are incremented. The outflow of moisture from the ith channel or channel segment (X_{8i}), ft^3 , is modeled after Lane[27] as:

$$F(X_{8,i}, P_{c,i}) = \{ I_{70i} + I_{71i} [\sum_{j \neq i} (P_{c,j}) + P_{m,i} + P_{a,i}] + I_{72i} (P_{r,i} + P_{l,i}) / I_{63i} \} 43654 \qquad \text{(App. A., No. 13)}$$

$F(X_{8,i}, P_{c,i})$ = daily flow or tranmission rate, ft^3 , from the ith channel or channel segment
$X_{8,i}$ = amount of moisture, ft^3, in the ith channel or channel segment
$P_{c,i}$ = flow of moisture, acre-ft , from the ith channel or channel segment
I_{70i} = regression intercept, acre-ft, for the ith channel or channel segment
I_{71i} = regression slope factor for the ith channel or channel segment
$P_{c,j}$ = flow of moisture, acre-ft, from jth connecting upslope channel or channel segment
$P_{m,i}$ = flow of moisture from icemelt,acre-ft, from the ith channel or channel segment
$P_{a,i}$ = flow of moisture from underlying aquifer, acre-ft, from below the ith channel or channel segment (value = 0.0 if aquifer not present under channel or channel segment)
I_{72i} = lateral flow parameter for transmission, mi , for the ith channel or channel segment
$P_{r,i}$ = return flow, acre-inch, entering ith channel or channel segment
$P_{l,i}$ = lateral flow, acre-inch, entering ith channel or channel segment (value = 0.0 if aquifer present under channel or channel segment)
I_{63i} = length of ith channel or channel segment , miles (input)

where,

$$I_{71i} = \exp(-I_{73i} I_{63i} I_{74i}) \qquad \text{(App. B,I, No. 28)}$$

I_{71i} = regression slope factor for the ith channel or channel segment
I_{63i} = length of ith channel or channel segment , miles (input)
I_{74i} = width of ith channel or channel segment, feet (input)

and ,

$$I_{73i} = -1.09 \, ln(1 - I_{79i}) \qquad \text{(App. B.,I, No. 29)}$$

I_{73i} = decay factor for transmission losses, $(mi\text{-}ft)^{-1}$ for the ith channel or channel segment
I_{79i} = decay factor coefficient, in/ acre-ft, for the ith channel or channel segment

where,

$$I_{79i} = 0.00545\ I_{41i}\ I_{75i} / (\ I_{76i}\ I_{68i}\ /12) \qquad \text{(App. B.,I, No. 30)}$$

I_{79i} = decay factor coefficient, in/ acre-ft, for the ith channel or channel segment
I_{41i} = channel hydraulic conductivity, inches/ hour, for the ith channel or channel segment
I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)
I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours
I_{76i} = potential mean runoff volume into the ith channel or channel segment, inches

where,

$$I_{75i} = I_{96i}\ (\ I_{68i} / 640\ )^{I_{97i}} \qquad \text{(App. B.,I, No. 31)}$$

I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours
I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)
I_{96i} = duration equation coefficient for ith channel or channel segment (input)
I_{97i} = duration equation exponent for ith channel or channel segment (input)

and,

$$I_{76i} = 0.05\ (\ I_{68i} / 640\ )^{-2} \qquad \text{(App. B.,I, No. 32)}$$

I_{76i} = potential mean runoff volume into the ith channel or channel segment, inches
I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)

The regression intercept (I_{70i}) is thus estimated as:

$$I_{70i} = [\ I_{77i} / (\ 1 - I_{78i})\ ]\ (\ 1 - I_{71i}\ ) \qquad \text{(App. B.,I, No. 33)}$$

I_{70i} = regression intercept, acre-ft, for the ith channel or channel segment
I_{71i} = regression slope factor for the ith channel or channel segment
I_{77i} = unit channel intercept for transmission loss, acre-ft, for the ith channel or channel segment
I_{78i} = unit channel regression slope for transmission loss for the ith channel or channel segment

where,

$$I_{77i} = 0.00465\ I_{41i}\ I_{75i} \qquad \text{(App. B.,I, No. 34)}$$

and,

$$I_{78i} = \exp\ (\ -\ I_{71i}\ ) \qquad \text{(App. B.,I, No. 35)}$$

I_{77i} = unit channel intercept for transmission loss, acre-ft, for the ith channel or channel segment
I_{78i} = unit channel regression slope for transmission loss for the ith channel or channel segment
I_{41i} = channel hydraulic conductivity, inches/ hour, for the ith channel or channel segment
I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours

I_{71i} = regression slope factor for the ith channel or channel segment

Finally, the lateral flow parameter (I_{72i}) can be estimated as:

$$I_{72i} = (1 - I_{71i}) / (I_{73i} I_{74i}) \qquad \text{(App. B.,I, No. 36)}$$

I_{72i} = lateral flow parameter for transmission, mi , for the ith channel or channel segment
I_{71i} = regression slope factor for the ith channel or channel segment
I_{73i} = decay factor for transmission losses, $(\text{mi-ft})^{-1}$ for the ith channel or channel segment
I_{74i} = width of ith channel or channel segment, feet (input)

These runoff and transmission processes are illustrated in Fig. 7, together with losses to and from an aquifer if one is located below a channel or channel segment.

The upper boundary of moisture flow along a channel or channel segment is set by the total inflow from all sources. If an aquifer is present below the channel or channel segment, then the moisture surplus is added to the aquifer, otherwise it is lost from the system (sink). In the event that an ephemeral or perennial flow of moisture enters a given channel, then ice formation decrements and icemelt increments to channel flow are taken into account in the model: if the mean daily temperature is below the freezing point , ice formation is estimated as:

$$F (P_{t,i}, X_{9i}) = 304.8 \, [(P_{r,i} + P_{l,i} + \sum_{j \neq i} (P_{c,j}) + P_{a,i})/ I_{80i}] \qquad D_2 < 0^\circ C \qquad \text{(App. A., No. 14)}$$

$$= 0 \qquad D_2 \geq 0^\circ C$$

$F (P_{t,i}, X_{9,i})$ = ice thickness, mm, deposited on ith channel or channel segment on a given day
$P_{t,i}$ = total amount of moisture added from all sources to the ith channel or channel segment, ft^3, on a given day
$X_{9,i}$ = ith channel or channel segment moisture content, ft^3, stored as ice
$P_{r,i}$ = return flow, ft^3, entering ith channel or channel segment
$P_{l,i}$ = lateral flow, ft^3, entering ith channel or channel segment (value = 0.0 if aquifer)
$P_{c,j}$ = flow of moisture, ft^3, from jth connecting upslope channel or channel segment
$P_{a,i}$ = flow of moisture from underlying aquifer, ft^3, from below the ith channel or channel segment (value = 0.0 if aquifer not present under channel or channel segment)
I_{80i} = surface area of channel or channel segment, ft^2

where,

$$I_{80i} = 2640 (I_{81i} + I_{82i}) I_{63i} \qquad \text{(App. B.,I, No. 37)}$$

I_{80i} = surface area of channel or channel segment, ft^2
I_{63i} = length of ith channel or channel segment , miles (input)
I_{81i} = width of aquifer, ft, at proximal end of ith channel or channel segment (input)
I_{82i} = width of aquifer, ft, at distal end of ith channel or channel segment (input)

4.9. Channel Ice Formation

The strategy employed in the model is to allow ice formation over the surface area of the aquifer directly below, and including the channel bed assuming a trapezoidal area. Similarly, icemelt is modeled at temperatures above freezing:

$$F (X_{9,i} , X_{8,i}) = (1.11280 + 11.28 D_{4i}) I_{80i} / 304.8 \qquad D_4 > 0^\circ C \qquad \text{(App. A., No.15)}$$

$$= X_{9,i} \qquad F (X_{9,i} , X_{8,i}) > X_{9,i}$$

$= 0$ $\qquad D_{4i} \leq 0° C$

$= 0$ $\qquad X_{9,i} = 0$

$F(X_{9,i}, X_{8,i})$ = flow of moisture, ft^3, from channel icemelt to the ith channel or channel segment on a given day

$X_{8,i}$ = amount of moisture, ft^3, in the ith channel or channel segment

$X_{9,i}$ = ith channel or channel segment moisture content, ft^3, stored as ice

I_{80i} = surface area of ith channel or channel segment, ft^2

D_{4i} = mean daily ith channel or channel segment drainage temperature, ° C

where,

$$D_{4i} = \sum_{j=1}^{n} D_{3j} / I_{83i} \qquad \text{(App. B.,II, No. 6)}$$

D_{4i} = mean daily channel drainage temperature, ° C for the ith channel or channel segment

D_{3j} = mean daily regional temperature of the jth region contributing surface runoff to the ith channel or channel segment

I_{83i} = number of regions supplying the ith channel or channel segment with surface runoff

Thus, the moisture resulting from icemelt is used to increment other surface flow additions to the stream channel, however, as the magnitude of surface additions increase, not all moisture would be expected to produce ice under freezing conditions, and the model becomes less accurate. It would be expected that some fraction of the moisture flow under these conditions would not form ice, but this relationship is not currently available.

4.10. Shallow Aquifer Flow

The flow of moisture into a confined alluvial aquifer(s) from transmission losses of the overlying channel, and gains from up-slope subsurface return flow is in the AQUIFER module of the code. The flow is assumed to flow through vertical cross sections that are derived from user input. The maximum aquifer volume itself is represented as an irregular trapezoidal solid that incorporates two trapezoidal cross sections at the proximal (up-slope) and distal (down-slope) ends. The different aquifers are then articulated into a drainage system using input instructions with each channel or channel segment having a specified length (I_{63}) ,miles, described previously. Furthermore, each aquifer is tilted from the proximal to distal end at a user specified slope (I_{60}) . Consequently, a given aquifer is modeled as a "tipped-bucket" to accomodate vertical height variations at both ends as a function of total moisture content at any given time. Also, it is possible to model a riparian community by specifying a region of this type on the non-channel portion of the surface area of aquifer (I_{80}) . This surface is then modeled to intercept moisture drainage in the soil horizon before water enters the associated channel as runoff or subsurface flow if excess moisture is available for either or both of these process to occur (F_8, F_{11}, respectively).

The maximum volume, ft^3, of specific aquifer (transverse slice method) is estimated as:

$$I_{102} = I_{84} I_{85}^{3} + I_{86} I_{85}^{2} + I_{87} I_{85} \qquad \text{(App. B.,I, No.38)}$$

I_{102} = maximum aquifer volume, ft^3

I_{84} = cubic coefficient

I_{85} = linear distance between proximal and distal ends of aquifer, ft

I_{86} = quadratic coefficient, ft

I_{87} = linear coefficient, ft^2

where,

$$I_{85} = (I_{88}^{2} + I_{89}^{2})^{1/2}$$ (App. B.,I, No.39)

I_{85} = linear distance between proximal and distal ends of aquifer, ft
I_{88} = vertical distance, ft, to the bottom of the aquifer at the proximal end relative to the bottom of the aquifer at the distal end of a channel or channel segment
I_{89} = horizontal distance, ft, between proximal and distal ends of a channel or channel segment

where ,

$$I_{89} = [(5280\ I_{63})^{2} - I_{90}^{2}]^{1/2}$$ (App. B,I, No. 40)

I_{89} = horizontal distance, ft, between proximal and distal ends of a channel or channel segment
I_{63} = length of the given channel or channel segment , miles (input)
I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment

and,

$$I_{90} = 5280\ I_{63}\ I_{60}$$ (App. B.,I, No. 41)

I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment
I_{63} = length of the given channel or channel segment , miles (input)
I_{60} = fractional slope of a given channel or channel segment (input)

The vertical distance (I_{88}) can now be estimated as:

$$I_{88} = I_{91} - I_{92}$$ (App. B.,I, No. 42)

I_{88} = vertical distance, ft, to the bottom of the aquifer at the proximal end relative to the bottom of the aquifer at the distal end of a channel or channel segment
I_{91} = vertical distance, ft, between bottom of aquifer at distal end relative to the surface elevation of the aquifer at the proximal end
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)

where,

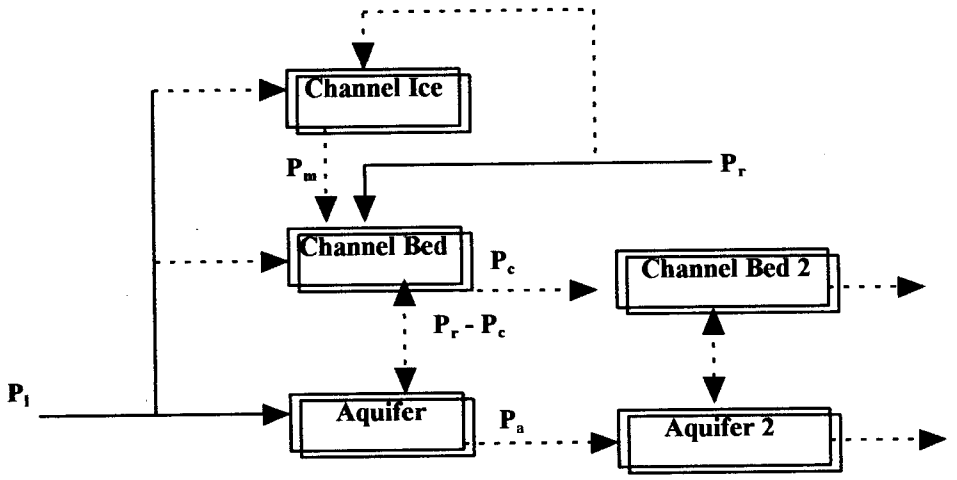
$$I_{91} = I_{93} + I_{90}$$ (App. B.,I, No. 43)

I_{91} = vertical distance, ft, between bottom of aquifer at distal end relative to the surface elevation of the aquifer at the proximal end
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment

The cubic coefficient (I_{84}) is derived as:

$$I_{84} = (I_{93} - I_{92}) [(I_{82} - I_{81}) + (I_{95} - I_{94})] / (6\ I_{85}^{2})$$ (App. B.,I, No. 44)

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)



P_r = moisture runoff from adjacent regional surfaces
P_m = moisture addition to channel bed from melting ice
P_l = subsurface or lateral moisture flow from adjacent regions
P_c = stream flow of moisture from channel bed
P_a = flow of moisture downslope from aquifer

note 1: dotted lines indicate conditional flows of moisture
note 2: (P_r) flows to channel bed or forms channel ice at sub-freezing temperatures

Fig. 7

Moisture Runoff and Transmission Processes Below a Channel or Channel Segment

I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input) oisture: (P_I) flows to aquifer if present,
flows to channel bed if aquifer not present , or forms channel ice at sub-freezing temperatures if aquifer not present

I_{81} = width of aquifer, ft, at proximal end below a given channel or channel segment (input)

I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)

I_{85} = linear distance between proximal and distal ends of aquifer, ft

I_{94} = width, ft, of the bottom of the aquifer at the proximal end below a given channel or channel segment (input)

I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

The quadratic coefficient (I_{86}) , ft, is derived as:

$$I_{86} = \{ I_{93} [I_{81} + I_{94} - 2 (I_{95} + I_{82})] + I_{92} (I_{82} + I_{95}) \} / (4 I_{85}) \qquad \text{(App. B.,I, No. 45)}$$

I_{86} = quadratic coefficient, ft

I_{81} = width of aquifer, ft, at proximal end below a given channel or channel segment (input)

I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)

I_{85} = linear distance between proximal and distal ends of aquifer, ft

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)

I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)

I_{94} = width, ft, of the bottom of the aquifer at the proximal end below a given channel or channel segment (input)

I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

The linear coefficient (I_{87}), ft^2, is derived as:

$$I_{87} = I_{93} (I_{82} + I_{95}) / 2 \qquad \text{(App. B.,I, No. 46)}$$

I_{87} = linear coefficient, ft^2

I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)

I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

The aquifer volume of a specific aquifer (I_{98}), ft^3, at a given height, above the bottom (I_{99}), ft, is estimated (vertical slice method) as:

$$I_{98} = I_{85} (I_{100} I_{99}^{\ 2} + I_{101} I_{99}) \qquad \text{(App. B.,I, No. 47)}$$

I_{98} = aquifer volume, ft3, at a given height, ft, above the bottom

I_{85} = linear distance between proximal and distal ends of aquifer, ft

I_{99} = estimated aquifer height, ft, for a given aquifer volume estimated using Newton's iterative approximation method from a known volume (I_{98}) , ft^3

I_{100} = quadratic coefficient

I_{101} = linear coeficient, ft

where,

$$I_{100} = (I_{82} - I_{95}) / (4 I_{93}) + (I_{81} - I_{94}) / (4 I_{92})$$ (App. B.,I, No. 48)

and,

$$I_{101} = (I_{95} + I_{94}) / 2$$ (App. B.,I, No. 49)

I_{100} = quadratic coefficient
I_{101} = linear coefficient, ft
I_{81} = width of aquifer, ft, at proximal end of a given channel or channel segment (input)
I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{94} = width, ft, of the bottom of the aquifer at the proximal end below a given channel or channel segment (input)
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

The flow of moisture from a channel or channel segment to an underlying aquifer F (X_8,X_{10}) is estimated as the difference between the total moisture inflow to the channel F (P_t,X_8) and that which leaves as streamflow , F (X_8, P_c) as defined previously (App. A., No. 13). Two other possible sources of moisture additions to an aquifer are from subsurface return flow, F (P_1,X_{10}), and transport from an upslope aquifer or aquifer segment, F (X_{10i-1} , X_{10i}) as shown in Fig. 7. The flow of moisture into an aquifer from subsurface return flow is articulated to the initial flow to the soil moisture compartment (X_1) overlying the aquifer which initially receives this flow from adjoining regions (see App. A, No. 11). The flow of moisture from a given aquifer or aquifer segment is dependent on the moisture pool in the aquifer compartment (X_{10}) which is used to estimate the cross section over which the flow out of the aquifer is modeled.
The distal vertical cross section through which moisture exits depends on the aquifer porosity (set in the code to 0.25), the maximum vertical crossection, and the estimated moisture volume (X_{10}) within the aquifer itself. If the moisture volume in the aquifer after adjustment for porosity does not exceed the maximum potential volume (I_{102}) derived earlier in this report (App. B.,I, No. 38), then Newton's approximation method is applied to formulation of the aquifer volume estimate ($_{I98}$) to solve for the unknown height (I_{99}):

$$I_{99n} = I_{99n} - I_{103} / I_{104}$$ (App. B.,I, No. 50)

I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of the above equation is close to 0.0
I_{103} = difference between the estimated aquifer volume using I_{99n} as the estimated height of the aquifer at the distal end and the known volume (I_{98})
I_{104} = first derivative estimate of volume equation (I_{98}) using aquifer estimated height at the distal end (I_{99n})

where,

$$I_{103} = I_{98n} - I_{98}$$ (App. B.,I, No. 51)

and,

$$I_{104} = 2 I_{85} (I_{100} I_{99n} + I_{101})$$ (App. B.,I, No. 52)

I_{103} = difference between the estimated aquifer volume using I_{99n} as the estimated height of the aquifer at the distal end and the known volume (I_{98})

I_{104} = first derivative estimate of volume equation (I_{98}) using aquifer estimated height at the distal end (I_{99n})

I_{98n} = aquifer volume estimate, ft^3, for the nth iteration at which convergence of equation (50) above is close to 0.0 using (I_{99n}) as the estimated height

I_{98} = aquifer volume, ft^3, at an unknown height, ft, above the bottom

I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0

I_{85} = linear distance between proximal and distal ends of aquifer, ft

Surplus moisture in excess of the maximum aquifer volume (I_{102}) is treated as regional runoff provided all soil layers above the aquifer are at the saturation point. If overlying soil layers are not at saturation, then moisture flow, $F(X_{10}, X_2)$, to the soil layers beginning with the lowest soil layer and proceeding upward until the excess aquifer moisture materials balance is attained. Finally, the loss of soil moisture from the distal vertical cross sectional area of the aquifer in a down slope direction is estimated by a modification of Darcy's equation described previously:

$$F(X_{10i-1}, X_{10i}) = 2\, I_{105}\, I_{41}\, I_{60} \qquad \text{(App. A., No. 16)}$$

$F(X_{10i-1}, X_{10i})$ = flow of moisture, ft^3 /day, from (i-1)th up slope aquifer compartment to ith down slope aquifer compartment

X_{10i-1} = aquifer moisture content, ft^3, of (i-1)th up slope compartment

X_{10i} = aquifer moisture content , ft^3, of ith down slope compartment

I_{41} = soil moisture infiltration rate, inches/ hour (input)

I_{60} = fractional slope of regional land surface (input)

I_{105} = cross sectional area, ft^2, of the distal vertical cross section of the aquifer at moisture level

where,

$$I_{105} = I_{99n}\,(I_{95} + I_{106})\,/\,2 \qquad \text{(App. B.,I, No. 53)}$$

I_{105} = cross sectional area, ft^2, of the distal vertical cross section of the aquifer at moisture level

I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0

I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

I_{106} = width of distal end of aquifer, ft, at the estimated height (I_{99n}) above the bottom of the aquifer

where,

$$I_{106} = I_{95} + (I_{82} - I_{95})\,/\,(I_{93}\, I_{99n}) \qquad \text{(App. B.,I, No. 54)}$$

I_{106} = width of distal end of aquifer, ft, at the estimated height (I_{99n}) above the bottom of the aquifer

I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0

I_{82} = width of aquifer, ft, at distal end of the given channel or channel segment (input)

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)

The modeling strategy for construction of the aquifer volumes was taken from the work of investigators studying Mortandad Canyon on the LANL reservation [28]. A diagram of the structure of an aquifer segment together with accompanying channel and regional characteristics is presented in Fig. 8. Other relationships related to moisture flow will be discussed other sections of this report in conjunction with other flows considered in this report.

5.0. SOIL AND SEDIMENT FLOW

5.1. Surface Runoff Events

The flow or transport of soil and sediment particles from one region to another is modeled using surface runoff and channel flows as vectors or carrying agents. The transport of soil particles to plant and animal surfaces , and internal deposition and ingestion by the latter are topics which will be discussed under the biomass flow section as they represent intra-regional transport phenomenon. The transport of soil and sediments are modeled using the strategies employed in the SPUR model for estimating total sediment yield from a given land surface as a result of surface runoff events [29]. These strategies are then coupled with differential erosion rate strategies as a function of particle diameter developed by Lane[27] to produce the overall modeling of particle transport into specific channel beds described previously. Themovement of particles within and between channels, channel segments, and regions are also adaptations from the study by Lane. These strategies are highly integrated with moisture flow modeling described above. Particle erosion is particulary sensitive to peak discharge rates from channels:

$$F(X_8, P_c)_{pk} = (I_{107} / I_{96})(F_8 / 25.4)(I_{68} / 640)^{(1 - I_{97})} \quad \text{(for a region)} \qquad \text{(App. A., No. 17)}$$
$$= I_{107} F_{13} / I_{75} \quad \text{(for a channel)}$$

$F(X_8, P_c)_{pk}$ = peak discharge rate, ft^3/s, for a given region or channel as a result of a runoff event
I_{96} = duration equation coefficient for a given channel or region (input)
F_8 = runoff event , mm $/m^2$ /day, from a given region
F_{13} = channel runoff, ft^3 / day
I_{68} = area, acres, of the region containing plant(s) where runoff has occurred (input)
I_{75} = estimated duration of a runoff event in a given channel or region, hours
I_{97} = duration equation exponent for a given channel or region (input)
I_{107} = regional or channel peak flow parameter (input)

It should be noted that the total runoff event from each region (F_8) is summed over all plant types present, whereas channel runoff (F_{13}) used for erosion purposes is the volume transmitted to the next channel or channel segment. Regional soil or sediment erosion is modeled as a function peak discharge (F_{17}) estimated above in combination with the Universal Soil Loss Equation (USLE) equation derived by the US Agricultural Research Service (ARS)[30] as:

$$F(X_{1,1}, P_s) = 95 (F_8 F_{17} / 25.4)^{0.56} I_{108} I_{109} I_{110} I_{111} I_{61} \qquad \text{(App. A., No. 18)}$$

$F(X_{1,1}, P_s)$ = weight of regional surface soil , tons /acre /day, eroded from a given moisture runoff event
F_8 = runoff event , mm $/m^2$ /day, from a given region
F_{17} = peak discharge rate, ft^3/s, for a given region as a result of a runoff event
I_{61} = ratio of crown cover fraction of a given plant type to the total cover of all plant types in a given region
I_{108} = estimate of (K) in the USLE equation (input)
I_{109} = estimate of (C) in the USLE equation (input)
I_{110} = estimate of (P) in the USLE equation (input)

I_{111} = estimate of (L) in the USLE equation (input)

This estimate must also be obtained by summing the contributions from each vegetation type specified for the given region. Estimates of differential particle erosion by diameter class from regional surfaces are modeled by coupling overall sediment yield from all vegetation types (F_{18}) described above with a weighting factor specific to a given particle diameter. These particles are then transported either to other regional surfaces or to a channel bed in a receptor region. However, estimates of differential particle in channel beds as bedload are modeled by using soil weighting factors and other runoff characteristics peculiar to a given runoff event [27].

5.2. Particle Size Effect

The following relationships are used in this model to estimate differential erosion rates, lbs /day, by particle diameter class:

$$F(X_{1,1}, P_{s,i}) = 2000\, F_{18}\, I_{68}\, I_{112i}\, I_{129} / I_{113} \quad \text{(for regions)} \qquad \text{(App. A., No. 19)}$$
$$= 3600\, I_{75}\, I_{74}\, I_{112i}\, I_{129} \quad \text{(for channels)}$$

$F(X_{1,1}, P_{s,i})$ = loss of particles of the ith diameter class, lbs /day, from a region or channel bed
F_{18} = weight of regional surface soil , tons /acre /day, eroded from a given moisture runoff event
I_{68} = area, acres, of the region containing a given number of plant type(s) (input)
I_{74} = width of a given channel or channel segment, feet (input)
I_{75} = estimated duration of a runoff event for a given channel or channel segment, hours
I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed
I_{113} = sum of all particle transport capacities, lbs /s-ft, for all diameter classes from a given region or channel bed normalized to the median particle diameter for a given soil or sediment
I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

where,

$$I_{112i} = 2\, I_{114i}\, I_{115}\, I_{116}^{\,2} \quad (i = 1, \text{ suspended load}) \qquad \text{(App. B.,I, No. 55)}$$
$$= I_{114i}\, I_{115}\, I_{117i}\, (I_{115} - I_{118i}) \quad (i > 1, \text{ bedload})$$

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed
I_{114i} = mass fraction of ith particle diameter classification for soil or sediment from a given region or channel bed (initial input)
I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed
I_{116} = The mean velocity of surface runoff, ft /s, for a given region or channel
I_{117i} = soil or sediment transport coefficient, ft^3 /lb-s, for the ith diameter class for soil or sediment from a given region or channel bed
I_{118i} = critical shear stress, lbs /ft^2, for the ith diameter class for soil or sediment from a given region or channel bed

where,

$$I_{117i} = 40 / I_{128i}^{\,1.5} \qquad \text{(App. B.,I, No. 56)}$$

I_{117i} = soil or sediment transport coefficient, ft^3 /lb-s, for the ith diameter class for soil or sediment from a given region or channel bed

I_{128i} = particle diameter, mm, for the ith diameter class of soil or channel sediment for a given region or channel bed (input)

and,

$$I_{118i} = 0.0022 + 0.01\, I_{128i} \qquad I_{128i} \le 1 \text{ mm} \qquad \text{(App. B.,I, No. 57)}$$
$$= 0.0078 + 0.02\, I_{128i} \qquad I_{128i} > 1 \text{ mm}$$

I_{118i} = critical shear stress, lbs /ft^2, for the ith diameter class for soil or sediment from a given region or channel bed

I_{128i} = diameter, mm, of ith diameter class for a given regional or channel surface (input)

The effective shear stress (I_{115}) is estimated as:

$$I_{115} = 62.43\, I_{119}\, I_{60} \qquad \text{(App. B.,I, No. 58)}$$

I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed

I_{60} = fractional slope of regional land surface or channel (input)

I_{119} = hydraulic radius, ft, for resistance of particles on the surface of regions or channels

where,

$$I_{119} = I_{120}\,(I_{121} / I_{122})^{1.5} \qquad \text{(App. B.,I, No. 59)}$$

I_{119} = hydraulic radius, ft, for resistance of particles on the surface of regions or channels

I_{120} = hydraulic radius for bed resistance, ft, for a given region or channel

I_{121} = grain or particle resistance coefficient for particles on the surface of regions or channels

I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels

and,

$$I_{121} = 0.132\, I_{127}^{\,0.166} \qquad \text{(App. B.,I, No. 60)}$$

I_{121} = grain or particle resistance coefficient for particles on the surface of regions or channels

I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

and also,

$$I_{122} = [\, I_{124}^{\,0.66} + (2\, I_{125} / I_{74})\,(I_{124}^{\,0.66} - 1.32\, I_{124}^{\,1.5})]^{0.66} \qquad \text{(App. B.,I, No. 61)}$$
$$= 0.63\, I_{124} \qquad I_{122} < 0.63\, I_{124}$$
$$= I_{126} \qquad I_{122} > I_{126}$$

where,

$$I_{126} = I_{124}\,[\,(I_{74} + 4\, I_{125}) / (4\, I_{125})]^{0.66} \qquad \text{(App. B.,I, No. 62)}$$

I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels

I_{126} = hydraulic roughness of the area of the channel or regional surface perpendicular to the direction of flow

I_{74} = width of a given channel or channel segment, feet (input)

I_{124} = roughness or Manning coefficient for a given region or channel (input)

I_{125} = depth of surface runoff flow, ft, from a given region or channel

and,

$$I_{125} = (F_8 / 25.4) / 12 \quad \text{(for regions)} \qquad \text{(App. B.,I, No. 63)}$$
$$= F_{13} / (43200\ I_{75}) \quad \text{(for channels)}$$

I_{125} = depth of surface runoff flow, ft, from a given region or channel
I_{75} = estimated duration of a runoff event for a given channel or channel segment. hours
F_8 = runoff event , mm /m^2 /day, from a given region
F_{13} = channel runoff, ft^3 / day

The hydraulic radius for bed resisitance for a given channel (I_{120}) can then be estimated as:

$$I_{120} = I_{123} (I_{122} / I_{124})^{1.5} \qquad \text{(App. B.,I, No. 64)}$$

I_{120} = hydraulic radius, ft, for bed resistance for a given region or channel
I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels
I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels
I_{124} = roughness or Manning coefficient for a given region or channel (input)

Where,

$$I_{123} = I_{74} I_{125} / (I_{74} + 2\ I_{125}) \qquad \text{(App. B.,I, No. 65)}$$

I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels
I_{125} = depth of surface runoff flow, ft, from a given region or channel
I_{74} = width of a given channel or channel segment, feet (input)

Also, the mean velocity of surface flow (I_{116}) can now be estimated as:

$$I_{116} = (1.49 / I_{124})\ I_{60}^{0.5}\ I_{123}^{0.66} \qquad \text{(App. B.,I, No. 66)}$$

I_{116} = The mean velocity of surface runoff, ft /s, for a given region or channel
I_{124} = roughness or Manning coefficient for a given region or channel (input)
I_{60} = fractional slope of regional land surface (input)
I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels

Both the individual diameter transport capacities (I_{112}), and the sum of all transport capacities (I_{113}) must be adjusted by the normalization factor (I_{129}) because this model of particle transport is based on the assumption that the sum of the transport capacities of the different particle size fractions is equal to the transport capacity estimated by using the median particle diameter size (I_{127}). However, this normalization factor applies to bedload particle diameters (> 0.063mm), and not to suspended load (≤ 0.063mm) particle diameters [27]. This factor is estimated as:

$$I_{129} = I_{130} / (\sum_{i=2}^{7} I_{112i}) \qquad \text{(App. B.,I, No. 67)}$$

I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed

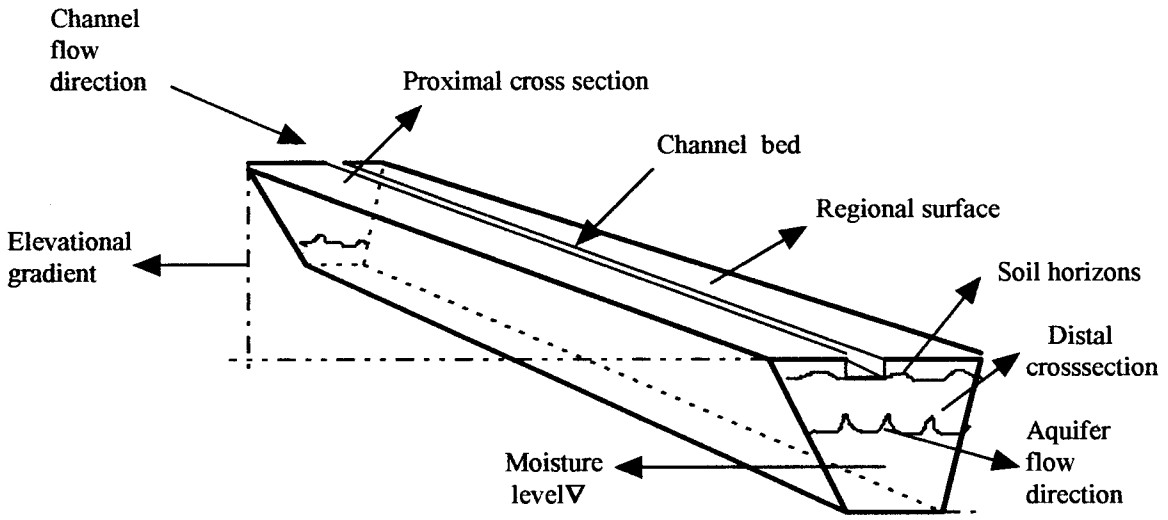


Fig. 8

Illustration of Regional Channel and Aquifer Structure and Function

I_{130} = transport capacity , lbs /ft-s, for the median particle diameter, mm (I_{127})

and,

$$I_{113} = I_{112,1} + I_{129} \sum_{i=2}^{7} I_{112i} \qquad \text{(App. B.,I, No. 68)}$$

I_{113} = sum of all particle transport capacities, lbs /s-ft, for all diameter classes from a given region or channel bed normalized to the median particle diameter for a given soil or sediment

$I_{112,1}$ = transport capacity , lbs /s-ft, of the suspended load particles ($\leq$ 0.063 mm)

I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed

The transport capacity for the median particle diameter size, mm, (I_{130}) is estimated as:

$$I_{130} = I_{131}\, I_{115}\, (I_{115} - I_{132}) \qquad \text{(App. B.,I, No. 69)}$$

I_{130} = transport capacity , lbs /ft-s, for the median particle diameter, mm (I_{127})

I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed

I_{131} = particle transport coefficient, ft^3 / lb-s, for the median particle diameter, mm (I_{127})

I_{132} = critical shear stress, lbs /ft^2, for the median for the median particle diameter, mm (I_{127})

where,

$$I_{131} = 40 / I_{127}^{\,1.5} \qquad \text{(App. B.,I, No. 70)}$$

I_{131} = particle transport coefficient, ft^3 / lb-s, for the median particle diameter, mm (I_{127})

I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

and,

$$\begin{aligned} I_{132} &= 0.0022 + 0.01\, I_{127} && I_{127} < 1.0 \text{ mm} \qquad \text{(App. B.,I, No. 71)} \\ &= -0.0078 + 0.02\, I_{127} && I_{127} \geq 1.0 \text{ mm} \end{aligned}$$

I_{132} = critical shear stress, lbs /ft^2, for the median for the median particle diameter, mm (I_{127})

I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

Finally, the total erosion of soil from a given region or sediment from a given channel over all particle diameter classes including bedload are estimated as the sum of (F_{19}) over all particle diameters in the medium:

$$F (X_1, P_s) = \sum_{i=1}^{7} F19_i \qquad \text{(App. A. No. 20)}$$

$F(X_1, P_s)$ = total soil or sediment loss , lbs /day, over all diameter classifications from a region or channel bed

F_{19i} = loss of particles of the ith diameter class, lbs /day, from a region or channel bed

The model uses a maximum of seven diameter classifications with the first diameter modeled as the suspended load of fines (≤ 0.063 mm in diameter), whereas the eighth diameter classification is reserved for the median particle diameter in the media.

6.0. BIOMASS FLOW

6.1. Plants

The flow of photosynthetically derived organic matter or biomass is central to all other modeled flows in that it simulates the principal vector for transporting contaminants to animals and humans through their respective food chains. Certain attributes of plant biomass flow are also integrated in this model such as food management, forest structure ; i.e., because biomass flow generally parallels contaminant flow which occurs concomitantly with it during feeding activities by both animals and humans. A diagram illustrating the interactions between plant, animal, and human biomass transports is presented in Fig. 9, together with other options available for use in the model. Internal flows of biomass always occur during during program operation, wheras, peripheral biomass flow requires user option. Attribute flow may affect biomass flow of various competing plants such as when the succession or niche competition options are exercised. The different strategies for plant growth require that internal flows adapt simultaneously to any group of plants under consideration. This type of adaptation will become more apparent as biomass flow characteristics are described in terms of intermediate variables and associated constants. Nutrients are also part of the growth process, and their affect will also be described in this section.

The potential gross photosynthesis of a given plant type is represented as:

$F(P_b, X_{5,1}) = X_{5,1} L_3 I_{133} I_{134}$ (App. A., No. 21)

$F(P_b, X_{5,1})$ = potential gross photosynthesis, $g/m^2/day$, of a given plant type
$X_{5,1}$ = plant leaf compartment, g / m^2
L_3 = photosynthetic gate (0,1)
I_{133} = photosynthetic efficiency, g-photosynthate /ly -day-g-leaf
I_{134} = effect of temperature on photosynthesis

where,

$L_3 = 1$ (App. C.,IV, No. 3)
$= 0$ $I_5 < I_{135} - 14$; $I_5 > I_{136} - 30$ (warm season annual plants)
$= 0$ $_5 < I_{135} - 20$; $I_5 > I_{136} - 14$ (cool season annual plants)
$= 0$ $I_{137} < I_{138}$; $I_{137} > I_{139}$ (agricultural annuals and biennials)

I_5 = julian day number (1 - 365)
I_{135} = beginning of frost-free period for the region, julian day number
I_{136} = end of frost-free period for the region, julian day number
I_{137} = day counter during the growing season, day number (may exceed 365 for some crops)
I_{138} = planting date , julian day number (input)
I_{139} = last harvest date, day number (may exceed 365 for some crops) (input)

and the frost-free period is used to estimate the beginning and end of the period:

$I_{135} = 210. - I_{140} / 2$ (App. B.,I, No. 72)
$I_{136} = 210 + I_{140} / 2$ (App. B.,I, No. 73)

I_{135} = beginning of frost-free period for the region, julian day number
I_{136} = end of frost-free period for the region, julian day number
I_{140} = frost-free period, days (input)

The photosynthetic efficiency is estimated for all plant type plants is estimated as:

$$I_{133} = 0.001\, D_3 \qquad \text{(App. B.,I, No. 74)}$$
$$= 0.45 \qquad I_{133} > 0.45$$

I_{133} = photosynthetic efficiency, g-photosynthate /ly -day-g-leaf
D_3 = total solar radiation intensity at site, Langleys / day

The photosynthetic efficiency is set to saturate at 450 ly /day insolation in agreement with the United States International Biological Program (US /IBP) for light saturation of Calvin type plants[31] which have maximum efficiencies of about 0.45 g of $[CH_2O]$ /g leaf biomass in the temperate zone. An adjustment in the model for Hatch-Slack plants is considered when net plant productivity is discussed.

The effect of temperature on photosynthesis is modeled as[32,33] :

$$I_{134} = 1 - (D_2 - C_5)^2 / C_6^2 \qquad \text{(App. B.,I, No. 75)}$$
$$= 1 - (D_2 - C_5)^2 / C_7^2 \qquad D_2 > C_5$$
$$= 0 \qquad I_{134} \leq 0$$

I_{134} = effect of temperature on photosynthesis
D_2 = mean daily temperature, ° C
C_5 = optimum temperature for photosynthesis for a given plant type, ° C (input)
C_6 = difference between optimum and lowest temperature tolerance for photosynthesis of a given plant type, ° C (input)
C_7 = difference between highest temperature tolerance and the optimum temperature for photosynthesis for a given plant type (input)

The potential gross photosynthesis estimated above is then used to estimate the water requirement for such production which is further adjusted by the availability of nitrate and phosphate in the metabolically active portion of root biomass[34,35] :

$$I_{141} = C_8 F_{21} / I_{142} \qquad \text{(App. B.,I, No. 76)}$$

I_{141} = moisture requirement, ml /m^2 /day, to realize potential gross photosynthesis
F_{21} = potential gross photosynthesis, g/m^2/day, of a given plant type
C_8 = water utilization requirement, ml-moisture /g-photosynthate production (input)
I_{142} = effect of root soluble nitrate and phosphate concentrations on photosynthesis

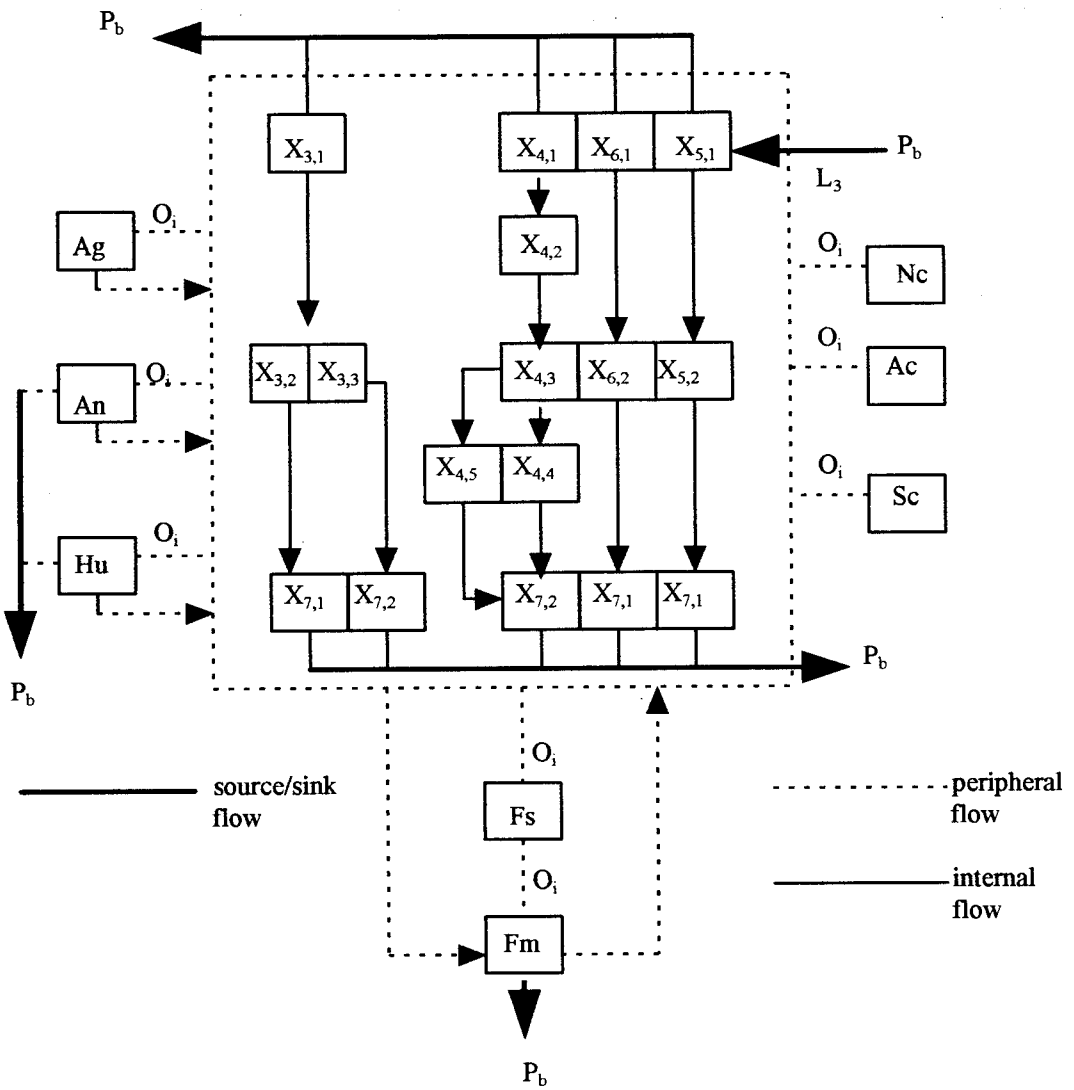
where,

$$I_{142} = 1.32 \{ 1 - \exp[(- 0.693 / 0.014) I_{143}]\}\{ 1 - \exp[(- 0.693 / 0.0014) I_{144}]\} \qquad \text{(App. B.,I, No. 77)}$$

I_{142} = effect of soluble root nitrate and phosphate concentrations on photosynthesis
I_{143} = nitrate concentration, g-N / gdwt-root in metabolically active tissues
I_{144} = phosphate concentration, g-P / gdwt-root in metabolically active tissues

The derivation of nitrate and phosphate concentrations in root tissues are presented in another section of this report where nutrient transport processes are addressed (see Section X), however, the user has the option of not using (setting $I_{142} = 1.0$) nutrient effects on plant growth if desired.

The volume of moisture available for photosynthesis is governed by the tramspiration rate and other factors:



Legend

Ag = agriculture
An = animal module
Hu = human module
Nc = niche competition
Sc = plant succession
Fs = forest structure module
Fm = forest management module
Ac = acute/chronic air module
O_i = ith user option, i=1,n
L_3 = photosynthesis gate
P_b = biomass source/sink
$X_{3,1}$ = live root
$X_{3,2}$ = dry root twig biomass
$X_{3,3}$ = dry root bole-branch biomass
$X_{7,1}$ = root twig humus
$X_{7,2}$ = root bole-stem humus
$X_{4,1}$ =live above standing biomass
$X_{4,2}$ = dry standing biomass
$X_{4,3}$ = dry lying biomass
$X_{4,4}$ = dry lying branch-twig biomass
$X_{4,5}$ = dry lying bole biomass
$X_{7,2}$ = branch-twig humus
$X_{7,2}$ = bole humus
$X_{5,1}$ = live leaf biomass
$X_{5,2}$ = dry leaf biomass
$X_{7,1}$ = leaf humus
$X_{6,1}$ = seed biomass
$X_{6,2}$ = dry seed
$X_{7,1}$ = seed humus

Fig. 9

Biomass Flow and Peripheral Attributes

$I_{145} = 1000 F_5 I_{146} I_{147} I_{148}$ (App. B.,I, No. 78)

I_{145} = adjusted amount of moisture available for photosynthesis, ml /m^2 /day
F_5 = total transpiration, mm/m^2/ day, for the a given plant type
I_{146} = adjustment factor for leaf areas indicies (LAI) below 1 m^2 /m^2
I_{147} = niche competition factor (> 0, ≤ 1, see App. B.,I, No. 106)
I_{148} = successsion competition factor (> 0, ≤ 1, see App. B.,I, No.107)

where ,

$I_{146} = 1 \quad I_{19} \geq 1$ (App. B.,I, No. 79)
$= 1/I_{19} \quad I_{19} < 1$

I_{146} = adjustment factor leaf areas indicies below 1 m^2 /m^2
I_{19} = leaf area index, m^2/m^2, (LAI)

The estimation of both the niche competetion and succession factors will be discussed later in this section. The adjustment factor (I_{146}) is based on the assumption that the availability of moisture to a plant type with low biomass density is proportional on a per m^2 basis. Hence, gross photosynthesis can be estimated as the potential (F_{21}), or may be adjusted for available moisture:

$F(P_b, X_{5,1})_a = F_{21}$ (App. A., No. 22)

$F(P_b, X_{5,1})_a = F_{21} I_{145} / I_{141} \quad I_{141} > I_{145}$

$F(P_b, X_{5,1})_a$ = adjusted potential gross photosynthesis, g/m^2/day
F_{21} = potential gross photosynthesis, g/m^2/day, of a given plant type
I_{145} = adjusted amount of moisture available for photosynthesis, ml /m^2 /day
I_{141} = moisture requirement, ml /m^2 /day, to realize potential gross photosynthesis

Net photosynthesis is then estimated from the potential adjusted or non-adjusted photosynthesis by removing the respiration fraction, and correcting for photosynthetic efficiency with respect to whether the specific plant is a a Calvin cycle (C^4) or a Hatch-Slack (C^3) type:

$F(P_b, X_{5,1})_n = F_{22} C_9/C_{10}$ (App. A., No. 23)

$F(P_b, X_{5,1})_n$ = net photosynthesis, g/m^2/day, for a given plant type
F_{22} = adjusted potential gross photosynthesis, g/m^2/day, for a given plant type
C_9 = respiratory maintenance fraction (input)
C_{10} = photosynthetic efficiency factor (input)

The transport of photosynthate from stems and leaves to roots is estimated as :

$F(X_{5,1} : X_{4,1} : X_{3,1}) = (1 - C_{11})(1 - I_{150}) F_{23}$ (App. A., No. 24)

$F(X_{5,1} : X_{4,1} : X_{3,1})$ = photosynthate transport from leaves-stems to root, g /m^2 /day
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
C_{11} = photosynthate transport fraction to stems (input)
I_{150} = adjusted photosynthate transport fraction to fruiting body for a given plant type
F_{23} = net photosynthesis, g/m^2/day, for a given plant type

Similarly, the transport of photosynthate from leaves to the stem is estimated as:

$$F(X_{5,1} : X_{4,1}) = C_{11}(1 - I_{150}) F_{23} \qquad \text{(App. A., No. 25)}$$

$F(X_{5,1} : X_{4,1})$ = photosynthate transport from leaves to stems, $g/m^2/$ day
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2
I_{150} = adjusted photosynthate transport fraction to fruiting body for a given plant type
F_{23} = net photosynthesis, $g/m^2/$day, for a given plant type
C_{11} = photosynthate transport fraction to stems (input)

and the the transport of photosynthate to developing fruits or seeds is:

$$F(X_{5,1} : X_{4,1} : X_{6,1}) = F_{23} I_{150} \qquad \text{(App. A., No.26)}$$

$F(X_{5,1} : X_{4,1} : X_{6,1})$ = photosynthate transport to developing fruits and seeds, g/m^2 /day
F_{23} = net photosynthesis, $g/m^2/$day, for a given plant type
I_{150} = adjusted photosynthate transport fraction to fruiting body for a given plant type
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2

where,

$$\begin{aligned} I_{150} &= I_{149} \\ &= I_{149}/2 && C_3 > 2 \\ &= 2 I_{149} && L_4 = 1 \end{aligned} \qquad \text{(App. B.,I, No. 80)}$$

I_{150} = adjusted photosynthate fraction to fruiting body of a specific plant type
I_{149} = photosynthate transport fraction to fruiting body of a specific plant type
C_3 = mean life-span, years, of the plant type simulated
L_4 = evergreen tree and shrub gate (1,0) (input)

and,

$$\begin{aligned} I_{149} &= 0.0 && L_3 = 0;\ \text{or } I_{149} \le 0.0 \text{ (all plant types)} \\ & && I_{137} > I_{139} \text{ (irrigated plants)} \\ & && \text{(fruit trees, deciduous and evergreen)} \\ & && \text{(shrubs less than 5 years old; and)} \\ & && \text{(other deciduous and evergreen trees)} \\ & && \text{(less than 10 years old} \\ &= 0.95 && I_{149} \ge 0.95 \\ &= \sum_{i=I_{135}+1}^{239} I_{149} + 0.004 && \text{(warm season plants)} \\ &= \sum_{i=240}^{270} I_{149} - 0.004 && \text{(warm season plants)} \\ &= \sum_{i=I_{135}-30}^{179} I_{149} + 0.004 && \text{(cool season plants)} \\ &= \sum^{209} I_{149} - 0.004 && \text{(cool season plants)} \end{aligned} \qquad \text{(App. B,I, No. 81)}$$

$$= \sum_{i=I_{138}+1}^{I=180 \; \Pi 39} I_{149} + 0.004 \qquad \text{(irrigated plants)}$$

I_{149} = photosynthate transport fraction to fruiting body of a specific plant type
L_3 = photosynthetic gate (0,1)
I_{137} = day counter during the growing season, day number (may exceed 365 for some crops)
I_{139} = last harvest date, day number (may exceed 365 for some crops) (input)
I_{135} = beginning of frost-free period for the region, julian day number
I_{138} = planting date , julian day number (input)

The above cummulative relationship (I_{149}) shows a gradually increasing fraction of photsynthate to the fruiting body of a plant which reaches a maximum value before declining toward the end of the growing season[35,36]. The final adjustment on the cummulative transport function (I_{150}) is made for perennial plants (excluding biennials) , and for evergreen shrubs and trees as shown previously.

The standing crop of live leaf biomass ($X_{5,1}$) as shown above is central to the production of photosynthate for translocation purposes. This model estimates the leaf biomass density as a fraction of the above live above ground biomass ($X_{4,1}$) as an adaptation from the RANGE IV model[32] to include woody plants into the strategy in addition to grasses and herbs:

$$\begin{aligned} X_{5,1} &= 928\,(X_{4,1} / 1000)\,\{72 + \exp[-3.11\,(X_{4,1} /1000)]\} && \text{(App. B.,V, No. 1)} \\ &= C_{12}\,X_{4,1} && L_5 = 0 \\ &= C_{13}\,I_{44} && L_5 = 0;\ X_{5,1} > C_{13}\,I_{44} \end{aligned}$$

$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{4,1}$ = live above ground biomass (stem), g /m^2
L_5 = woody or deciduous plant gate (0,1)
C_{12} = initial fraction of above ground biomass density assigned to leaf biomass density
C_{13} = mean fraction of above ground biomass density assigned to leaf biomass density as a function of a 10 year running average net productivity estimate
I_{44} = 10 year running average net productivity, g /m^2/ year, for a given plant type

where,

$$\begin{aligned} I_{44} &= I_{151} && C_3 = 1 && \text{(App. B., No. 82)} \\ &= I_{152} / I_{153} && C_3 > 1 \end{aligned}$$

I_{44} = 10 year running average net cummulative productivity, gdwt / m^2/ year (cummulative (daily productivity for annual plants)
F_{23i} = net photosynthesis, g/m^2/day, for a given plant type on the i<u>th</u> day
I_{135} = beginning of frost-free period for the region, julian day number
I_{136} = end of frost-free period for the region, julian day number
C_3 = mean life-span, years, of the plant type simulated
I_{151} = cummulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
I_{152} = cummulative net productivity for 1 year, g /m2/ year: the sum of current and previous year totals (previous year is running average between 1 - 10 years)
I_{153} = number of years of net productivity used in estimating (I_{153}) above (1 - 10 years)

where,

$$I_{151} = \sum_{i=1}^{365} F_{23i} \qquad \text{(App. B.,I, No. 83)}$$

I_{151} = cummulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
F_{23i} = net photosynthesis, g/m^2/day, for a given plant type on the ith day

and,

$$I_{152} = \sum_{i=1}^{n} I_{151} \qquad 1 \le n \le 10 \qquad \text{(App. B.,I, No. 84)}$$

$$I_{153} = \sum_{i=1}^{n} i \qquad 1 \le n \le 10 \qquad \text{(App. B.,I, No. 85)}$$

I_{152} = cummulative net productivity for 1 year, g /m2/ year: the sum of current and previous year totals (previous year is running average between 1 - 10 years)
I_{151} = cummulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
I_{153} = number of years of net productivity used in estimating (I_{153}) above (1 - 10 years)

A ten year cummulative productivity average for estimating mature leaf biomass densities for shrubs and trees was found by experiment to dampen the otherwise wide variablity seen from extreme climatic conditions in some years. The conversion of leaf biomass density to dead or dry leaf biomass is estimated as:

(App. A., No. 27)

$$\begin{aligned}
F(X_{5,1}, X_{5,2}) &= 0.005\, X_{5,1} \\
&= 0.693\, X_{5,1} / 365 && L_4 = 1;\ L_5 = 0 \\
&= [2 / (365\, C_3)]\, C_{14}\, X_{5,1} && L_4 = 0;\ L_5 = 0 \\
&= X_{5,1} && L_4 = 0;\ L_5 = 0;\ I_5 = I_{136} \\
&= \{1.1 - [1 / (C_6 - 5)]\, D_2\}\, C_{14}\, X_{5,1} && L_4 = 0;\ L_5 = 1;\ D_2 < C_6 - 5 \\
&= [\, C_{15}\, (D_2 - C_5) + 0.1]\, C_{14}\, X_{5,1} && L_4 = 0;\ L_5 = 1;\ D_2 > C_5 \\
&= X_{5,1} && L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1 \\
&= 0.1\, X_{5,1} && O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \le 2;\ I_{139} = I_{137} \\
&= 0.0 && O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \le 2; \\
& && I_{139} \neq I_{137};\ I_{137} < I_{138};\ I_{137} \ge I_{139} \\
&= 0.0 && F_{27} \le 0
\end{aligned}$$

The conversion of live to dry stem biomass density follows most of the conditions outlined above:

(App. A., No. 28)

$$\begin{aligned}
F(X_{4,1}, X_{4,2}) &= 0.005\, X_{4,1}\, (1 - X_{5,1} / X_{4,1}) \\
&= (X_{4,1} - X_{5,1})\, [2 / (365\, C_3)]\, C_{14} && L_4 = 1;\ L_5 = 0 \\
&= [2 / (365\, C_3)]\, C_{14}\, (1 - X_{5,1} / X_{4,1})\, X_{4,1} && L_4 = 0;\ L_5 = 0 \\
&= F_{27}\, (X_{4,1} - X_{5,1}) / X_{4,1} && L_4 = 0;\ L_5 = 1;\ D_2 < C_6 - 5 \\
&= F_{27}\, (X_{4,1} - X_{5,1}) / X_{4,1} && L_4 = 0;\ L_5 = 1;\ D_2 > C_5 \\
&= X_{4,1} - X_{5,1} && L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1 \\
&= 0.1\, (1 - X_{5,1} / X_{4,1})\, X_{4,1} && O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \le 2;\ I_{139} = I_{137} \\
&= 0.0 && O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \le 2; \\
& && I_{139} \neq I_{137};\ I_{137} < I_{138};\ I_{137} \ge I_{139} \\
&= 0.0 && F_{28} \le 0
\end{aligned}$$

The conversion of live seeds to dry seeds is estimated as:

(App. A., No. 29)

$$\begin{aligned}
F(X_{6,1}, X_{6,2}) &= 0.005\, X_{6,1} \\
&= X_{6,1} && L_4 = 0;\ L_5 = 0;\ I_5 = I_{136}
\end{aligned}$$

$= X_{6,1}$ $\quad L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1$

$= 0$ $\quad F_{29} \leq 0$

Finally, the conversion of live roots to dry roots is divided into a readily decomposable fraction and a resistant fraction given as:

$F(X_{3,1}, X_{3,2}) = 0.005\, X_{3,1}\, (X_{5,1} / X_{4,1})$ (App. A., No. 30)

$= 8[2 / (365\, C_3)]\, X_{3,1}\, (X_{5,1} / X_{4,1})$ $\quad L_4 = 1;\ L_5 = 0$

$= 15\, [2 / (365\, C_3)]\, X_{3,1}\, (X_{5,1} / X_{4,1})$ $\quad L_4 = 0;\ L_5 = 0$

$= X_{3,1}\, (X_{5,1} / X_{4,1})$ $\quad L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1$

$= X_{3,1}\, (X_{5,1} / X_{4,1})$ $\quad O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \leq 2;\ I_{139} = I_{137}$

$= 0.0$ $\quad F_{30} \leq 0$

$F(X_{3,1}, X_{3,3}) = 0.005\, X_{3,1}\, (1 - X_{5,1} / X_{4,1})$ (App. A., No. 31)

$= 8[2 / (365\, C_3)]\, X_{3,1}\, (1 - X_{5,1} / X_{4,1})$ $\quad L_4 = 1;\ L_5 = 0$

$= 15\, [2 / (365\, C_3)]\, X_{3,1}\, (1 - X_{5,1} / X_{4,1})$ $\quad L_4 = 0;\ L_5 = 0$

$= X_{3,1}\, (1 - X_{5,1} / X_{4,1})$ $\quad L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1$

$= X_{3,1}\, (1 - X_{5,1} / X_{4,1})$ $\quad O_5 = 1;\ I_{139} \neq I_{154};\ C_3 \leq 2;\ I_{139} = I_{137}$

$= 0.0$ $\quad F_{30} \leq 0$

$F(X_{5,1}, X_{5,2})$ = conversion of live leaf biomass to dry leaf biomass, g /m^2 /day
$F(X_{4,1}, X_{4,2})$ = conversion of live stem biomass to dry stem biomass, g /m^2 /day
$F(X_{6,1}, X_{6,2})$ = conversion of live seed biomass to dry seed biomass, g /m^2 /day
$F(X_{3,1}, X_{3,2})$ = conversion of live root biomass density to readily decomposable dry root biomass g /m^2 /day
$F(X_{3,1}, X_{3,3})$ = conversion of live root biomass density to resistant to decomposition dry root biomass, g /m^2 /day
$X_{3,1}$ = live root biomass density, g /m^2
$X_{3,2}$ = dry root biomass, g /m^2, which is readily decomposed
$X_{3,3}$ = dry root biomass, g /m^2, which is not readily decomposed
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2
L_4 = evergreen tree and shrub gate (1,0)
L_5 = woody or deciduous plant gate (0,1)
O_5 = irrigation option (1,0) (input)
C_3 = mean life-span, years, of the plant type simulated
C_5 = optimum temperature for photosynthesis for a given plant type, $^\circ$C (input)
C_6 = difference between optimum and lowest temperature tolerance for photosynthesis of a given plant type, $^\circ$C (input)
C_{14} = live-to-dry stem and leaf conversion factor (input)
C_{15} = live-to-dry leaf conversion factor for herbaceous plants (input)
D_2 = mean daily temperature, $^\circ$C
I_5 = julian day number (1 - 365)
I_{136} = end of frost-free period for the region, julian day number
I_{137} = day counter during the growing season, day number (may exceed 365 for some crops)
I_{138} = planting date , julian day number (input)
I_{139} = last harvest date, julian day number (may exceed 365 for some crops) (input)
F_{27} = conversion of live leaf biomass to dry leaf biomass, g /m^2 /day
I_{154} = last irrigation date, julian day number (input)

Another conversion of live biomass density to dry biomass density occurs when fruit trees are regenerated: the code performs this function when the age of the these trees equals 25 years of simulated age. At this time 0.1, 0.1, and 0.5 of live leaves, stems, and root biomass densities are converted to their dry counterparts.

The conversion of dry biomass density to humus density for above ground plant components is accumulated in the first soil layer, whereas, the conversion of dry root biomass density is accumulated in all layers occupied by the plant type in question depending on its degree of penetration into the soil. The flow of biomass from dry leaves and seeds to humus biomass density are estimated as:

$$F(X_{5,2}, X_{7,1}) = I_{155}\, I_{156}\, X_{5,2} \qquad \text{(App. A., No. 32)}$$

$$F(X_{6,2}, X_{7,1}) = I_{155}\, I_{156}\, X_{6,2} \qquad \text{(App. A., No. 33)}$$

$F(X_{5,2}, X_{7,1})$ = conversion of dry leaf litter to humus, g /m^2 /day
$F(X_{6,2}, X_{7,1})$ = conversion of dry seed litter to humus, g /m^2 /day
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{7,1}$ = leaf, seed, and root-twig humus combined biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2
I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{156} = effect of soil moisture on litter decomposition

where,

$$I_{155} = 0.004 + 0.0032\, I_{157} \qquad \text{(App. B.,I, No. 86)}$$

I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{157} = estimated temperature of soil surface, ° C

and,

$$\begin{aligned} I_{157} &= D_2 - 4\,(X_{5,1} - 300)\,/\,600 && X_{5,1} > 300 \\ &= D_2 + 15\, I_{158}\,(1 - X_{5,1}\,/\,300) && X_{5,1} \le 300 \end{aligned} \qquad \text{(App. B.,I, No. 87)}$$

I_{157} = estimated temperature of soil surface, ° C
D_2 = mean daily temperature, ° C
$X_{5,1}$ = live leaf biomass density, g /m^2
I_{158} = actual to potential evaporation ratio

and,

$$\begin{aligned} I_{158} &= 1 - (F_5\, I_{19} + F_4\,/\,I_{17}) && I_{19} \le 1 \\ &= 1 - (F_5 + F_4\,/I_{17}) && I_{19} > 1 \end{aligned} \qquad \text{(App. B.,I, No. 88)}$$

I_{158} = actual to potential evaporation ratio
I_{17} = potential evapotranspiration (PET) from soil surface, mm/day
F_5 = total transpiration, mm/m^2/ day, for the jth plant
F_4 = transport of moisture from soil surface (evaporation), mm/ m^2 / day
I_{19} = leaf area index, m^2/m^2, (LAI)

Finally, the effect of soil moisture on litter decomposition is estimated as:

$$I_{156} = [1\,/\,(I_{58} - I_{22})]\,(X_{1,1} - I_{22}) \qquad \text{(App. B.,I, No. 89)}$$

I_{156} = effect of soil moisture on litter decomposition

I_{58} = total soil moisture present at field capacity, mm, for the 1st soil layer of the jth plant type in a given region (input)
I_{22} = soil moisture content, mm, for the 1st soil layer of the jth plant at wilt point (input)
$X_{1,1}$ = soil moisture in surface layer, mm

The conversion of dry stems to humus requires an intermediate step of converting dead standing stems, branches and twigs to dry material on the ground surface:

$$F(X_{4,2}, X_{4,4}) = 0.693\, X_{4,2}\, [100\, X_{4,4}) / (365\, C_3)]\, X_{4,6} / (X_{4,6} + X_{4,7})$$ (App. A., No. 34)

$$F(X_{4,2}, X_{4,5}) = 0.693\, X_{4,2}\, [10\, X_{4,5}) / (365\, C_3)]\, X_{4,7} / (X_{4,6} + X_{4,7})$$ (App. A., No. 35)

$$F(X_{4,2}, X_{4,3}) = [0.693\, X_{4,2} / (365\, C_3)]\, I_{159}$$ (App. A., No. 36)

$F(X_{4,2}, X_{4,3})$ = conversion of dry standing stem to dry lying stem biomass, g /m^2 /day
$F(X_{4,2}, X_{4,4})$ = amount of dry standing stem converted to lying branch-twig litter, g /m^2 /day
$F(X_{4,2}, X_{4,5})$ = amount of dry standing stem converted to lying bole litter, g /m^2 /day
$X_{4,2}$ = standing dry stem biomass density, g /m^2
$X_{4,3}$ = dry stem biomass on ground surface, g /m^2
$X_{4,4}$ = branch-twig component of dry lying stem biomass, g /m^2
$X_{4,5}$ = bole component of dry lying stem biomass, g /m^2
$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2
$X_{4,7}$ = bole component of live above ground biomass, g /m^2
C_3 = mean life-span, years, of the plant type simulated
I_{159} = weighting factor for estimating litter formation from stems (boles) and branches

where branches and twigs are assumed to form litter ten times faster than boles:

$$I_{159} = (100\, X_{4,6} + 10\, X_{4,7}) / (X_{4,6} + X_{,4,7})$$ (App. B.,I, No. 90)

I_{159} = weighting factor for estimating litter formation from stems (boles) and branches
$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2
$X_{4,7}$ = bole component of live above ground biomass, g /m^2

where,

$$X_{4,7} = I_{160}\, I_{161} = 1 \qquad X_{4,7} \le 1$$ (App. B.,V, No. 2)

$$X_{4,6} = (X_{4,1} - X_{5,1}) - X_{4,7} = 1 \qquad X_{4,6} \le 1$$ (App. B.,V, No. 3)

$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2
$X_{4,7}$ = bole component of live above ground biomass, g /m^2
I_{160} = estimated biomass volume density, cm^3 / m^2 for a given plant at a given time
I_{161} = estimated density of dry plant tissue, g / m^3, for a given plant
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2

where,

$$I_{160} = \pi\, I_{162}\, I_{163}^{\,2}\, I_{164} / 3$$ (App. B.,I, No. 91)

I_{162} = estimated plant areal density, number / m2 for a given plant type at a given time

I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{164} = estimated height of a given plant type, cm, at a given time

The estimated plant areal density (I_{162}) is set to 1.0 for all plants when the model is not accesssing the forest structure module (FORMAN) or is estimated from the simulated number of trees within the code itself. If an even-aged stand is being simulated using FORMAN, then the areal density may be estimated from input of the number of trees. A similar situation arises when making mean stem diameter (I_{163}), plant height (I_{164}), and wood (cellulosic) density estimates:

$$I_{162} = I_{165} / 4047 \qquad L_2 > 1 \qquad \text{(App. B.,I, No. 92)}$$
$$= 1 \qquad L_2 \leq 1$$

$$I_{163} = 2.54\, I_{166} \qquad L_2 > 1 \qquad \text{(App. B.,I, No. 93)}$$
$$= 0.003\, X_{4,1} \qquad L_2 \leq 1$$
$$= 0.1 \qquad I_{163} \leq 0.1$$

$$I_{164} = 308.4\, I_{167} \qquad L_2 > 1 \qquad \text{(App. B.,I, No. 94)}$$
$$= 0.1\, X_{4,1} \qquad L_2 \leq 1$$
$$= 7 \qquad I_{164} < 7$$

$$I_{161} = I_{168} \qquad L_2 > 1 \qquad \text{(App. B.,I, No. 95)}$$
$$= 0.2 \qquad L_2 \leq 1$$

I_{161} = estimated density of dry plant tissue, g / m^3, for a given plant
I_{162} = estimated plant areal density, number / m^2 for a given plant type at a given time
I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{164} = estimated height of a given plant type, cm, at a given time
L_2 = forest structure access gate (1,2,3) (input)
I_{167} = mean height, ft, for a stand of a given plant (see App. B., I, No. 120)
I_{165} = tree density, trees /acre, for a given plant type (input, or see App. B,I, No. 121)
$X_{4,1}$ = live stem biomass density, g /m^2
I_{166} = mean bole diameter, inches, for a stand of a given plant (see App. B.,I, No. 123)
I_{168} = wood density, g /cm3, of a given plant type (input or estimated)

The conversion or flow of both bole and branch-twig litter into humus is estimated as:

$$F(X_{4,4}, X_{7,2}) = I_{155}\, I_{156}\, X_{4,3} / 10 \qquad \text{(App. A., No. 37)}$$

$$F(X_{4,5}, X_{7,2}) = I_{155}\, I_{156}\, X_{4,5} / 10 \qquad \text{(App. A., No. 38)}$$

$F(X_{4,4}, X_{7,2})$ = conversion of dry branch-twig biomass density to humus, g /m^2 /day
$F(X_{4,5}, X_{7,2})$ = conversion of dry bole biomass density to humus, g /m^2 /day
I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{156} = effect of soil moisture on litter decomposition
$X_{4,4}$ = dry branch and twig component of lying stem biomass, g /m^2
$X_{4,5}$ = dry bole component of lying stem biomass, g /m^2
$X_{7,2}$ = branch, bole, twig, and root bole-branch humus combined biomass density for a given plant, g /m^2

The conversion or flow of both readily decomposable and resistant to decomposition dry roots into humus is estimated as:

$$F(X_{3,2}, X_{7,1}) = I_{169i}\, I_{170i}\, I_{171i}\, X_{3,2} \qquad \text{(App. A., No. 39)}$$

$$F(X_{3,3}, X_{7,2}) = I_{169i}\, I_{170i}\, I_{171i}\, X_{3,3} / 10 \qquad \text{(App. A., No. 40)}$$

$F(X_{3,2}, X_{7,1})$ = conversion of readily decomposable root fraction to humus, g /m^2 / day
$F(X_{3,3}, X_{7,2})$ = conversion of resistant to decomposition root fraction to humus, g /m^2 /day
I_{169i} = effect of soil temperature of the ith soil on root litter decomposition
I_{170i} = effect of soil moisture of the ith soil layer on root litter decomposition
I_{171i} = root penetration fraction into the ith soil layer
$X_{7,1}$ = leaf, seed, and root-twig humus combined biomass density, g /m^2
$X_{7,2}$ = branch, bole, twig, and root bole-branch humus combined biomass density for a given plant, g /m^2

where,

$$I_{169i} = 0.004 + 0.0032\, I_{172i} \qquad \text{(App. B.,I, No. 96)}$$

I_{169i} = effect of soil temperature of the ith soil on root litter decomposition
I_{172i} = estimated temperature of ith soil layer, °C

and,

$$I_{172i} = 12.78 + (D_2 - 12.78) \exp[(0.693 / 396)(I_{173i} - 0.5\, I_{37i})] \qquad \text{(App. B.,I, No. 97)}$$

I_{172i} = estimated temperature of ith soil layer, °C
D_2 = mean daily temperature, °C
I_{37i} = thickness of ith soil layer, mm
I_{173i} = depth to the bottom of the ith soil layer, mm

and,

$$I_{173i} = \sum_{j=1}^{i} I_{37j} \qquad \text{(App. B.,I, No. 98)}$$

I_{173i} = depth to the bottom of the ith soil layer, mm
I_{37j} = thickness of jth soil layer, mm

The effect of soil moisture on root litter decomposition is estimated as:

$$I_{170i} = 1 + [1 / (I_{58i} - I_{22i})](X_{1,i} - I_{22i}) \qquad \text{(App. B.,I, No. 99)}$$
$$= 1 \qquad X_{1,i} < I_{22i}$$

I_{170i} = effect of soil moisture of the ith soil layer on root litter decomposition
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of a given plant type (input)
I_{22i} = soil moisture content, mm, for the ith soil layer of a given plant at wilt point (input)
$X_{1,i}$ = soil moisture in the ith layer, mm

$$I_{171i} = I_{21i}\, I_{174i} \qquad \text{(App. B.,I, No. 100)}$$
$$= 1 \qquad C_3 = 1$$

I_{171i} = root penetration fraction into the ith soil layer
I_{21i} = root penetration fraction (0 - 1) of a given plant into the ith soil layer
I_{174i} = fraction of root biomass in ith soil layer of a given plant (input or estimated)

C_3 = mean life-span, years, of the plant type simulated

when the default value of root biomass fraction is used, it is estimated as [32,37]:

$$I_{174i} = (I_{175i} - I_{175i-1}) / I_{176} \qquad \text{(App. B.,I, No. 101)}$$

I_{174i} = fraction of root biomass in ith soil layer of a given plant (estimated)
I_{175i} = cummulative root fraction up to the ith soil layer
I_{175i-1} = cummulative root fraction up to the (i-1)th soil layer
I_{176} = maximum soil penetration factor for a given soil depth (I_{56})

where,

$$I_{175i} = [1 - \exp(- 0.004\, I_{178i})] / 0.632 \qquad \text{(App. B.,I, No. 102)}$$

I_{175i} = cummulative root fraction up to the ith soil layer
I_{177i} = depth to bottom of ith soil layer, mm

where,

$$I_{177i} = \sum_{j=1}^{i} I_{37j} \qquad \text{(App. B.,I, No. 103)}$$

I_{177i} = depth to bottom of ith soil layer, mm
I_{37j} = thickness of the jth soil layer, mm

and,

$$I_{176} = [1 - \exp(-0.004\, I_{56})] /0.632 \qquad \text{(App. B.,I, No. 104)}$$

I_{176} = maximum soil penetration factor for a given soil depth (I_{56})
I_{56} = total depth, mm, of soil profile for the rooting zone of a given plant type (input)

The mineralization of humus and consequent recycling of biomass components is divided into a readily mineralized portion, and another portion which is resistant to this process. The resistant humus is assumed to mineralize about 20 times slower than the readily decomposable humus:

$$F (X_{7,1i} ,P_b) = I_{178}\, X_{7,1} \qquad \text{(App. A., No. 41)}$$

$$F (X_{7,2i} ,P_b) = 0.05\, I_{178}\, X_{7,2} \qquad \text{(App. A., No. 42)}$$

$F (X_{7i,1} ,P_b)$ = humus recycling losses, g /m^2 /day, for readily mineralized humus
$F (X_{7i,2} ,P_b)$ = humus recycling losses, g / m$_2$ /day, for resistant humus
$X_{7,1i}$ = readily mineralized portion of humus, g /m^2, in the ith soil layer
$X_{7,2i}$ = resistant portion of humus, g /m^2 , in the ith soil layer
I_{178} = mineralization factor for readily mineralized humus
P_b = biomass source/sink

where,

$$I_{178} = 0.1\, C_{16} \exp(C_{17}\, I_{172i}) \qquad \text{(App. B.,I, No. 105)}$$

I_{178} = mineralization factor factor for readily mineralized humus
I_{172i} = estimated temperature of ith soil layer, °C
C_{16} = mineralization coefficient

C_{17} = mineralization constant, $^\circ C^{-1}$

Humus is also assumed to eluviate into the soil profile at about 0.2mm year based on field observations of soil profiles:

$$F(X_{7,1i}, X_{7,1i+1}) = (0.00055 / I_{37i}) X_{7,1i} \qquad \text{(App. A., No. 43)}$$
$$F(X_{7,2i}, X_{7,2i+1}) = (0.00055 / I_{37i}) X_{7,2i} \qquad \text{(App. A., No. 44)}$$

$F(X_{7,1i}, X_{7,1i+1})$ = transport of readily decomposable humus from the ith soil layer to the (i=1)th soil layer, g /m^2 /day
$F(X_{7,2i}, X_{7,2i+1})$ = transport of resistant humus from the ith soil layer to the (i+1)th soil layer, g /m^2 /day
$X_{7,1i}$ = readily mineralized portion of humus, g /m^2, in the ith soil layer
$X_{7,2i}$ = resistant portion of humus, g /m^2 , in the ith soil layer
I_{37j} = thickness of the jth soil layer, mm

6.1.1. Plant Niche and Succession Competition Factors

When plants of similar growth habit such as warm and cool perennial season grasses may be placed into competition for moisture throughout the growing season on the assumption that they share the same surface area for growth. The competition factor developed in this model from tuning experiments is based on the ratio of net photosynthesis to the current leaf biomass density:

$$I_{147} = F_{23} / X_{5,1} \qquad O_1 = 1;\ I_{179} > 0 \qquad \text{(App. B.,I, No. 106)}$$
$$= 1.0 \qquad X_{5,1} = 1$$
$$= 0.1 \qquad I_{147} \le 0.1$$

where,

I_{147} = moisture or niche competition factor, day^{-1}
F_{23} = net photosynthesis, g/m^2/day, for a given plant type
$X_{5,1}$ = live leaf compartment, g/ m^2
I_{179} = niche competition match number (input)
O_1 = niche competition option (1,0) (input)

In this modeling strategy the competition factors (I_{147}) for all plant types with matching numbers (I_{179}) are summed , and each competition factor is adjusted so that the sum of all competition factors of the matched set is normalized to equal one. There may be several or more sets of matched numbers in any given simulation. Furthermore, for all plants undergoing niche competition on a daily basis (having equal matching numbers , I_{179}), a weighted total transpiration rate for all plants combined is estimated on the basis of standing live leaf biomass densities at any given time:

$$F(X_{5w}, P_t) = \left[\sum_{j=1}^{n} \sum_{i=2}^{I_{20j}} F_{5j,i}\ X_{5,1j} \right] / \sum_{j=1}^{n} X_{5,1j} \qquad I_{179,1} = I_{179,2} = \ldots I_{179,n} \qquad \text{(App. A, No. 45)}$$

$F(X_{5w}, Pt)$ = weighted total transpiration, mm /m^2 /day, for the niche matched plant group
n = number of plants in a niche matching group of plants
$F_{5j,i}$ = total transpiration, mm/m^2/ day, for the ith soil layer of the jth plant (see App. A., No. 5)
I_{20j} = number of soil layers specified for the jth plant (input or estimated)
$X_{5,1j}$ = live leaf biomass density, g /m^2, of the jth plant in the niche matching group
$I_{179,j}$ = niche competition match number for the jth plant in the matched plant group (input)

Hence, the availability of moisture for transpiration by a specific plant in the niche competition grouping is limited by F_{45} above, and the soil moisture available for photosynthesis adjusted by both the niche competition factor and the succession factor (see App. B,I, No. 145). Finally, this strategy assumes that the soil structure (depth and composition) for each plant in any niche grouping is identical, as it does in the development of the succession competition factor to be descibed below.

The competition of plants of differing growth habits results in a differential change in plant abundances over many growing seasons with the development of seral stages. The present capability of BIOTRAN.2 to simulate this succession is the result of a significant increase in model complexity and a simplification of earlier attempts to model this phenomenon [20]. The new version operates similarly to the niche competition model by limiting the soil moisture available for transpiration among competing plants, but is estimated on a yearly rather than on a daily basis. In addition, niche competition can also be modeled incorporated into the succession scheme. The current succession modeling strategy differs significantly from other models of this type [38,39,40], which use Markovian replacement processes for estimating plant succession. The strategy employed in this model uses dynamically generated parameters that are ultimately dependent on the stochastic driving variables utilized in BIOTRAN.2 simulations and is affected by climatic variations simulated in the model:

$$
\begin{aligned}
I_{148} &= 1 && O_5 = 1 \qquad \text{(App. B.,I, No. 107)} \\
&= 1 && O_2 = 0;\ O_{2a} = 0 \\
&= 1 && I_5 = 1, I_{29} = 1, C_3 < 30;\ O_2 = 1; O_{2a} = 0 \\
&= 0 && I_5 = 1, I_{29} = 1, C_3 \geq 30;\ O_2 = 1; O_{2a} = 0 \\
&= \mathit{ln}\,(X_{4,1\,m} + X_{3,1} + 0.7183)\, I_{43\,m}^{\ 2}\ I_{167m}^{\ 0.6} && I_5 = 1, I_{29} > 1;\ O_2 = 1 \text{ or } O_{2a} = 1
\end{aligned}
$$

where,

$$
\begin{aligned}
I_{43m} &= 0.3163 && I_{43m} < 0.3163 \\
I_{167m} &= I_{167} && L_2 > 1 \\
&= X_{4,1m} / 360 && L_2 = 1 \\
&= 0.022 && I_{167m} < 0.022
\end{aligned}
$$

I_{148} = successsion competition factor ($> 0, \leq 1$)
$X_{4,1\,m}$ = maximum live above ground biomass (stem) compartment, g/ m^2 for a given year
$X_{3,1}$ = live root compartment, g/ m^2 on Julian day 365
$I_{43\,m}$ = maximum plant fractional crown cover, m^2/m^2, for a given year
I_{43} = plant fractional crown cover, m^2/m^2
I_{29} = julian year number of specific year of simulation
I_{167m} = mean or maximum height, ft, for a stand of a given plant
I_{167} = mean height, ft, for a stand of a given plant (see App. B., I, No. 120)
O_5 = irrigation access option (1,0) (input)
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)
L_2 = forest structure access gate (1,2,3) (input)

The modeling strategy used incorporates a logarithmically related structural biomass term factor involving live above and below ground plant portions, a ground or crown cover factor, and a height factor. The ultimate dominance of perennial over annual grasses is primarily the effect of the presence of live root biomass at the end of the growing season. The dominance of shrubs over grasses and ,finally, the dominance of trees over shrubs is due to the crown cover and height factors in the model. This type of strategy has been tested and tuned primarily to comparisons of model output to field observations from Pinyon-Jumiper and Ponderosa Pine forests[41], but has also been compared to other

types of evergreen and deciduous forests with favorable results. The succession factor estimated for all plants under going succession are combined and normalized as was done for niche competition factors:

$$I_{148in} = I_{148j} / \sum_{j=1}^{n} I_{148j} \qquad \text{(App. B.,I, No. 108)}$$

$= 0.97$ $\qquad I_{148in} > 0.97$
$= 0.01$ $\qquad I_{148in} < 0.01$
$= 1$ $\qquad X_{4,1} = 1$

I_{148jn} = normalized succession factor for the jth plant type
I_{148j} = un-adjusted successsion competition factor ($> 0, \leq 1$) for the jth plant type
n = number of plants undergoing succession

The are two types of timing strategies employed in this model to estimate the appearance of a given seral stage of development: the first strategy assumes that only the first seral stage (annual grasses and herbs) is present initially (I_{148ja}), whereas the second st rategy assumes incomplete dominance of the last seral stage of development (trees, I_{148jt}). The latter strategy is used to accelerate the establishment of the climax or pseudo-stable last seral stage of development. The timing strategy for development of the plant community from the first seral stage of development is estimated as:

$I_{148ja} = 1$ $\qquad I_{29}=1;\ O_2=1; O_{2a}=0$ $\qquad$ App.B.,I, No. 109)
$I_{148jp} = 0$ $\qquad 30 \leq C_3 < 70;\ I_{29}= 1 O_2=1; O_{2a}=0$
$I_{148js} = 0$ $\qquad X_{4,1}=1;\ 70 \leq C_3 < 150;\ I_{148ja} > I_{148jp} / 3\ O_2=1; O_{2a}=0$
$I_{148j}t = 0$ $\qquad X_{4,1}=1; 150 \leq C_3;\ I_{148jp} > I_{148js} / 3 O_2=1; O_{2a}=0$

I_{148ja} = composite normalized succession factor for the jth annual plant type
I_{148jp} = composite normalized succession factor for the jth perennial plant type
I_{148js} = composite normalized succession factor for the jth shrub plant type
I_{148jt} = composite normalized succession factor for the jth tree plant type
C_3 = mean life-span, years, of the plant type simulated
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
I_{29} = julian year number of specific year of simulation
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)

The timing strategy for the development of all seral stages discussed shows a timed appearance of shrub and tree seral stages when their preceding seral stages have a succession factors which exceed by a factor of three the next earliest seral stages: annual grasses and herbs, perennial grasses and herbs, evergreen and deciduous shrubs, and evergreen and deciduous trees. This timing strategy has been estimated from tuning experiments with the model as the best fit to field observations of this type. However, the entire strategy needs more field validation and study under various climatic conditions , and with different plant communities than those used for tuning purposes. Finally, if more than one plant is simulated for any of the above seral stages, then these plants are subjected to niche competition in order to estimate composite succession factors for each stage as shown above.

The conditions for the timing strategy of the accelerated development of the climatic climax seral stage are modeled as:

$I_{148jt} = 0.7$ $\qquad I_5=1; I_{29}=1; O_2=0;\ O_{2a}=1; O_5=0$ $\qquad$ (App. B.,I, No. 110)
$= 0.7$ $\qquad I_5=1;\ I_{148jt} < 0.7$
$I_{148js} = 0.2$ $\qquad I_5=1; I_{29}=1; O_2=0;\ O_{2a}=1;\ C_3 \leq 70; O_5=0$
$= 0.2$ $\qquad I_5=1;\ I_{148js} < 0.2$
$I_{148jp} = 0.1$ $\qquad I_5=1; I_{29}=1; O_2=0;\ O_{2a}=1;\ C_3 \leq 30; O_5=0$

$$
\begin{aligned}
&= 0.1 && I_5=1;\ I_{148jp} < 0.1 \\
I_{148ja} &= 0.1 && I_5=1; I_{29}=1; O_2=0;\ O_{2a}=1;\ C_3 \leq 30; O_5=0 \\
&= 0.1 && I_5=1;\ I_{148ja} > 0.1
\end{aligned}
$$

O_5 = irrigation access option (1,0) (input)
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)
I_5 = julian day number (1 - 365)
I_{29} = julian year number of specific year of simulation
I_{148ja} = composite normalized succession factor for the jth annual plant type
I_{148jp} = composite normalized succession factor for the jth perennial plant type
I_{148js} = composite normalized succession factor for the jth shrub plant type
I_{148jt} = composite normalized succession factor for the jth tree plant type

This type of semi-empirical strategy results in a final seral stage of development which is similar to a simulation initated with the first seral stage (annual grasses and) in a much shorter simulation run. As can be observed from the above relationships all seral stage succession factors except the first are allowed to exceed the initially assigned factors with exception of the first. Additionally, the first seral stage of development is not allowed to become larger than the initially assigned succession factor, whereas, all other seral stages are not allowed to become smaller than their initial values. Finally, the initial values for each seral stage are based on simulation runs with the more comprehensive version discussed earlier.

6.1.2. Forest Structure

The forest structure model and its companion forest management model are used for simulations where tree or shrub biomass densities over 1000 m^2/gdwt are attained. An additional need for the the forest structure is necessary for estimating ground cover more accurately because the algorithm used for estimating ground cover for grasses and herbs (I_{43}) becomes inaccurate at biomass densities of the type described above. This model is primarily designed to simulate forest structure by bole diameter classification but other important aspects such as contaminant transport within the structure itself , and as a result for forest cutting options or forest fires are also important outputs from this model [18,20]. The forest structure model (FORMAN) was developed initially using a semi-empirical strategy using biomass density and mensuration data obtained from studies in New Mexico and Arizona [13,14,20,42,43].

The finally accepted strategy is hypothetical in that the specific attributes of a Ponderosa Pine type forest was used in its initial development [43] by comparison of diameter class evolution as a dynamic modified log-normal distribution based on total volume of stand of known age in a given study area. Tuning experiments with this type of model yielded satisfactory predictive capabilities, but it was found to be insensitive or capable of coping with perturbations of the type required for forest management operations, nor for negative biomass excursions from one year to the next which is common to the plant growth portions of BIOTRAN.2. Consequently, an attempt was made to couple a log-normal type tree structure generator for the initial diameter class with a compartmental strategy with a compartmental growth strategy that would yield results similar to a modified log-normal distribution generator when applied to the whole stand. Subsequent development led to complete replacement of distribution generaztors for all diameter classes , and to the refinement of the compartmental approach which yielded satisfactory forest structure operation for Ponderosa Pine stands and Pinyon woodlands [42,43,44]. Since that time the model has been extended for use on other evergreen and deciduous forest s stands with some successs, but require further study for validation purposes.

A diagram of the forest structure is presented in Fig. 10 to illustrate the allocation of net wood production to different diameter classes for a given year from an annual growth increment of a specific stand. Also, it can be observed that the model includes both intra-class wood volume transport or growth of a specific diameter grouping, and inter-class transport of wood volume from the lower diameter classes to the larger ones. Hence, larger diameter class groupings are created through time as total wood

volume increases. The diameter class competition factors which determine the fraction of annual growth/mortality which is allocated to a specific diameter class is estimated as follows:

$$I_{180i} = 0.2212\, I_{46}^{2}\, I_{48} \qquad I_{47i} = 1 \qquad \text{(App. B.,I, No. 111)}$$
$$= I_{46}\, I_{47i}^{2}\, I_{48i}\, I_{181} \qquad I_{47i} > 1$$

where,

I_{180i} = diameter class competition factor for the ith diameter class
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (input)
I_{47i} = mean of ith bole diameter class, inches
I_{48i} = number of trees in ith bole diameter class

I_{181} = diameter class ingrowth factor for the ith diameter class

where,

$$I_{47i} = (2i - 1.0)\, I_{182} / 2 \qquad \text{(App. B.,I, No. 112)}$$

$$I_{181} = 1 - \exp[-0.5(2i - 1)] \qquad i > 1 \qquad \text{(App. B.,I, No. 113)}$$

I_{47i} = mean of ith bole diameter class, inches
I_{182} = diameter range, inches, within a given diameter class (input)
i = diameter class, 1,2,3,...n
I_{181} = diameter class ingrowth factor for the ith diameter class

and,

$$I_{48i} = X_{11i} / I_{183i} \qquad \text{(App. B.,I, No. 114)}$$

I_{48i} = number of trees in ith bole diameter class
X_{11i} = total volume of trees in the ith diameter class, ft^3
I_{183i} = volume of a unit tree in the ith diameter class, ft^3

where,

$$I_{183i} = [\,(0.035 I_{184} / I_{168})\,(2.54\, I_{182})^2\,] / 2 \qquad i = 1 \qquad \text{(App. B.,I, No. 115)}$$

$$= [0.035\, I_{184} / I_{168}\,(\{2.54\, I_{182}(i - 1)\} + \{2.54\, i\, I_{182}\}^2)] / 2 \qquad i > 1$$

I_{183i} = volume of a unit tree in the ith diameter class, ft^3
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{182} = diameter range, inches, within a given diameter class (input)
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)

The total volume of all trees in the first diameter class is estimated for either an even or un-even aged stand as follows:

$$X_{11i} = I_{185} \exp(-0.0037\, I_{185}) \qquad I_{147i} = 1 \qquad \text{(App.B.,V, No. 4)}$$

$$= (I_{186}\, I_{183i})^3 / I_{185}^{2} \qquad I_{147i} = 1,\ I_{186} > 1$$

X_{11i} = total volume of trees in the ith diameter class, ft^3, i=1
I_{183i} = volume of a unit tree in the ith diameter class, ft^3
I_{185} = total volume of tree stand ,ft^3 /acre
I_{186} = number of trees in even-aged stand, otherwise set to 0.0 (input)

where,

$$I_{185} = 0.142\, X_{4,1} / I_{168} \qquad \text{(App. B.,I, No. 116)}$$

I_{185} = total volume of tree stand ,ft^3 /acre
I_{168} = wood density, g /cm3, of a given plant type (input)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2

The volume for the first diameter class is predicted to decline exponentially for uneven-aged stands as total volume of the stand increases, whereas, for an even-aged stand this diameter class is predicted to decline in proportion to increasing stand volume. The diameter class competition factors described above (I_{180}) are applied to all diameter classes except the first to partition the net gain in wood volume by the stand:

$$F(P_{n,v}, X_{11i}) = P_{n,v}\,(I_{180i-1} - I_{180i}) / I_{188} \qquad i > 1 \qquad \text{(App. A., No. 46)}$$

$$= 0.0 \qquad X_{11i} \leq 0.01\, I_{183i}$$

$$F(X_{11i}, P_{m,v}) = X_{11i}\, I_{187} \qquad P_{n,v} < 0.0 \qquad \text{(App. A. No. 47)}$$

$F(P_{n,v}, X_{11i})$ = transport of annual net wood production to the ith diameter class, ft^3 /yr
$F(X_{11i}, P_{m,v})$ = net annual mortality loss from the ith diameter class, ft^3 /year
$P_{n,y}$ = net wood increment, ft^3 / yr
$P_{m,v}$ = net wood losses, ft^3 / yr
X_{11i} = total volume of trees in the ith diameter class, ft^3
I_{180i} = intra-class competition factor for the ith diameter class
I_{187} = mortality factor for wood volume losses
I_{188} = sum of all intra-class competition factors

where,

$$I_{187} = I_{185m} / I_{185\,m-1} \qquad P_{n,v} < 0.0 \qquad \text{(App. B.,I, No. 117)}$$

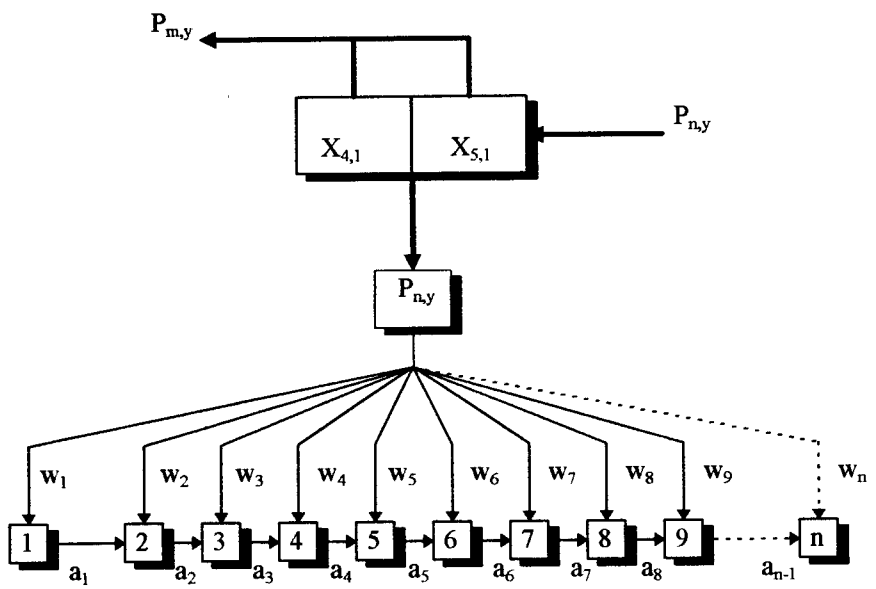
I_{187} = mortality factor for wood volume losses
I_{185m}, I_{m-1} = total volume of tree stand ,ft^3 /acre, for current (mth) and previous year(m-1th)

and,

$$I_{188} = \sum_{j=1}^{k} I_{180j} \qquad \text{(App. B.,I, No. 118)}$$

I_{188} = sum of all intra-class competition factors
I_{180j} = intra-class competition factor for the jth diameter class
k = total number of diameter classes (value varies during a simulation)

This type of strategy provides for a replacement of smaller diameter classes in the stand by larger class diameters in several steps: The diameter class competition factor of the previous diameter class is used to simulate intra-class transport of wood , whereas, the current diameter class competition is used to simulate inter-class diameter transport of wood between the different diameter classes. A uneven-aged is transformed into a even-aged stand by this mechanism at stand maturity, whereas, an even-aged stand



Legend

$X_{4,1}$ = standing live bole biomass, g / m^2 , at the end of the year
$X_{5,1}$ = standing live leaf leaf biomass, g /m^2 , at the end of the year
$P_{m,y}$ = net annual wood losses, ft^3 / year
$P_{n,y}$ = net annual wood increment, ft^3 / yr
$w_{1,2,..n}$ = intraclass diameter competition factors
$a_{1,2,n}$ = interclass diameter transport , ft^3 /year
X_{11i} = volume compartments by diameter classification: i = 1,2,3,..k , ft^3

Fig. 10

Biomass Flow in Forest Structure Model (FORMAN)

initially is forced to remain as such by transporting excess wood increments from lower to higher diameter classes beginning with the higher diameter classes until the number of trees initially specified is equal to the simulated number. A reversed strategy to that mentioned above is applied to an even-aged stand when the simulated number of trees falls below that initially specified.

The prediction of wood volumes and number of trees in different diameter classes allows other secondary predictions to be made of a given stand of trees for both specific diameter classes and for the stand as a whole. These predicted parameters include mean tree heights , basal (bole) areas , crown covers, mean ages, ... etc., including variances, for both specific diameter classes, and for the stand in general. Most of these predictions do not involve uncommon relationships, however, estimates of mean tree heights does involve a power function relationship described below[42] :

$$I_{189i} = (307\, I_{184} / I_{168})\,(I_{47i} / 24)^{0.9} \qquad \text{(App. B.,I, No. 119)}$$

I_{189i} = mean tree height, ft, in the ith diameter class
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{47i} = mean of ith bole diameter class, inches

Other studies [41,43] have been used to show the application of this relationship to Pinyon Pine, Douglas Fir, Spruce, and Aspen. The stand tree geometric mean height excluding the first diameter class is estimated as:

$$I_{167} = \exp(I_{190} / I_{165}) \qquad \text{(App. B.,I, No. 120)}$$

$$I_{165} = \sum_{i=2}^{n} I_{48i} \qquad \text{(App. B.,I, No. 121)}$$

I_{167} = mean geometric height, ft, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
I_{190} = logarithmic sum of tree heights

where,

$$I_{190} = \sum_{i=2}^{n} I_{48i}\; ln\,(I_{189i}) \qquad \text{(App. B.,I, No. 122)}$$

I_{190} = logarithmic sum of tree heights
I_{48i} = number of trees in ith bole diameter class
I_{189i} = mean tree height, ft, in the ith diameter class

Similarly, the geometric mean bole diameter for all diameter classes except the first can be estimated as:

$$I_{166} = \exp(I_{191} / I_{165}) \qquad \text{(App. B., I, No. 123)}$$

I_{166} = geometric mean bole diameter, inches, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
I_{191} = logarithmic sum of tree diameters

where,

$$I_{191} = \sum^{n} I_{48i}\; ln\,(I_{47i}) \qquad \text{(App. B.,I, No. 124)}$$

i=2

I_{191} = logarithmic sum of tree diameters
I_{48i} = number of trees in ith bole diameter class
I_{47i} = mean of ith bole diameter class, inches

The stand mean tree height can also be estimated using the geometric mean bole diameter to compare with the geometric mean estimate (see App. B.,I, No. 189):

$$I_{192} = (307\, I_{184} / I_{168})\,(I_{166} / 24)^{0.9} \qquad \text{(App. B.,I, No. 125)}$$

I_{192} = mean tree height, ft, using the geometric mean bole diameter, inches
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{166} = geometric mean bole diameter. inches, for a stand of a given plant

Finally, the crown cover fraction for the entire stand excluding the first diameter class which is used in other sections of the model can be estimated as:

$$I_{43} = (0.0929\pi / 4047) \sum_{i=2}^{n} I_{48i}\,(I_{47i}\, I_{46} / 2)^2 \qquad \text{(App. B.,I, No. 126)}$$

I_{43} = plant fractional crown cover, m^2/m^2
I_{48i} = number of trees in ith bole diameter class
I_{47i} = mean of ith bole diameter class, inches
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (input)

Other predicted forest stand characteristics not derived in this document that can be used for output purposes are found in Appendix D of this document.

6.1.3. Forest Management

The forest management options in this model are companion to the forest structure described previously, and cannot be used independently of the latter ($O_4 \geq 1$, see Fig. 9). The management options are in the form of clearing or cutting schemes and/ or the simulation of a forest fire. Basically, the strategy is performed by dividing the diameter classes generated in the forest structure model into three groupings, and then specifying through user input removal fractions for these groupings which have also been specified as input. For example, the user may wish to group the first 5 diameter classes into the first group, the 6 - 10 diameter classes into the second grouping, and the 11 - 20 diameter classes into the third grouping. The user then specifies what fraction of the wood from each of these groupings is removed, and when this event is to occur in simulation time (year). The strategy is the same for a forest fire which may be a scheduled controlled or uncontrolled event. The model removes wood (ft^3) using these specifications and then adjusts the stand biomass density to a lower value. A type of cutting option employed by the U.S. Forest Service[42] which differs from the previously mentioned cutting strategies is also available, and is presented below as a Reverse-J cut:

$$\begin{aligned} F(X_{11i}, P_{rji}) &= X_{11i} - 126 \exp(-0.394i)\, I_{183i} && \text{(App. A., No. 48)} \\ &= 0.0 && X_{11i} < 126 \exp(-0.394i)\, I_{183i} \\ &= X_{11i} && i \geq 13 \end{aligned}$$

P_{rji} = wood removed, ft^3, from the ith diameter class using Reverse-J cutting option
X_{11i} = total volume of trees in the ith diameter class, ft^3

I_{183i} = volume of a unit tree in the ith diameter class, ft^3

The second term of the generalized equation above represents the desired wood volume, ft3, allowed to remain un-cut; if the amount present is less than this preset value, then cutting is not allowed. However for trees with diameters equal to or greater than 26 inches (a 2 inch class interval as reference), then all of the wood volume from that diameter class is removed. The total amount of wood removed is thus:

$$F(X11_t, P_{rj}) = \sum_{i=1}^{n} P_{rji} \qquad \text{(App. A., No. 49)}$$

P_{rji} = wood removed, ft^3, from the ith diameter class using Reverse-J cutting option
P_{rj} = wood removed, ft3, from all diameter class combined using Reverse-J cutting option
X_{11t} = total volume of trees from all diameter classes combined, ft^3

where ,

$$X_{11t} = \sum_{i=1}^{n} X_{11i} \qquad \text{(App. B.,V, No. 5)}$$

11_t = total volume of trees from all diameter classes combined, ft^3
X_{11i} = total volume of trees in the ith diameter class, ft^3

Up to ten forest management events may be simulated in the same run of the model including a forest fire(s).

6.1.4. Plant Nutrients

The ions NH_4^{+1} , NO_3^{-1} , and PO_4^{-3} , and K^{+1} are currently used in the model as macronutrients, but only the first three ions are modeled to affect plant growth at the present time. In addition, the uptake dynamics of potassium cations are modeled with the same strategy as other metals and metal contaminants, whereas, the uptake dynamics of ammonium cations, and nitrate and phospate anions are modeled with very different strategies[35]. Hence, only the later three ions will be considered in this section, and potassium uptake dynamics will be considered in the section on contaminant transport in plants to be presented later in this report. The effect of nitrate and phosphate concentrations on plant growth have already been presented (see Section VI), and this discussion will be limited to completing the documentation on the transport of these nutrients. Phosphate concentrations in the model are compartmentalized into labile (relatively insoluble) and soluble fractions with the former being the source of the latter. The phosphate exchange fraction from the labile to the soluble form is affected by the clay content of the soil:

$$I_{193i} = 1 / \{15 + 185 [1 - \exp(- 0.693\, I_{36i} / 3)] \} \qquad \text{(App. B.,I, No. 127)}$$

I_{193i} = phosphate exchange fraction from labile to soluble form in the ith soil layer
I_{36i} = clay fraction in ith soil layer (input)

The concentration of soluble phosphate in the soil is estimated on a daily basis as:

$$I_{194i} = I_{193}\, I_{195i}\, I_{32i} \qquad \text{(App. B.,I, No. 128)}$$

I_{194i} = soluble phosphate concentration, g-P /gdwt-soil, in the ith soil layer
I_{195i} = labile phosphate concentration, g-P / gdwt-soil in the ith soil layer
I_{193} = phosphate exchange fraction from labile to soluble form in the ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer.

The un-adjusted plant root uptake rate, of inorganic phosphate is, however, estimated using the labile phosphate fraction as:

$$I_{196i} = \{ [1.635e\text{-}04/ (1 + 1.43 / I_{197i})]\}+\{ [3.27e\text{-}05/ (1 + 0.084 / I_{197i})]\} \qquad \text{(App. B.,I, No. 129)}$$

I_{196i} = un-adjusted plant root uptake rate, g-P / g-root / m^2/day of inorganic phosphate in the ith soil layer
I_{197i} = ratio of labile phosphate in 1 m^3 of soil to the volumetric moisture ratio

where,

$$\begin{aligned} I_{197i} &= 1.4e06\ I_{195i} / I_{32} && && \text{(App. B.,I, No. 130)} \\ &= 0 && I_{32} = 0 \\ &= 0.00001 && I_{32} > 0,\ I_{197i} \le 0.00001 \end{aligned}$$

I_{197i} = ratio of labile phosphate in 1 m^3 of soil to volumetric moisture ratio in the ith soil layer
I_{195i} = labile phosphate concentration, g / gdwt-soil in the ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer

The un-adjusted root uptake is adjusted for effects of soil temperature and moisture, phosphate leakage from roots, soil matric potential, concentration of nitrate in roots, and for fraction of transpiration :

$$\begin{aligned} I_{198i} &= I_{196i}\ I_{199} (1 - I_{200})\ I_{201}\ I_{202}\ I_{203} && && \text{(App. B.,I, No. 131)} \\ &= 0 && X_{3,1} \le 1.0 \\ &= 0 && I_{198i} < 0 \end{aligned}$$

I_{198i} = adjusted plant root uptake g-P/ g -root/ m^2 /day of inorganic phosphate for ith soil layer
I_{196i} = unadjusted plant root uptake rate, g-P / g-root / m^2/day of inorganic phosphate in the ith soil layer
I_{199i} = effect of soil temperature on uptake of phosphate for ith soil layer
I_{200i} = leakage fraction of phosphate from roots for ith soil layer
I_{201i} = effect of soil moisture on uptake of phosphate for ith soil layer
I_{202i} = evapotranspiration ratio effect on uptake of phosphate for ith soil layer
I_{203i} = effect of nitrate concentration in roots on uptake of phosphate for ith soil layer

where,

$$\begin{aligned} I_{199i} &= 0.19 + 0.045\ I_{172i} && && \text{(App. B.,I, No. 132)} \\ &= 0.41 + 0.0175\ I_{172i} && I_{172i} \le 10 \\ &= 0.25\ I_{172i} && I_{172i} \le 2 \\ &= 1 && I_{172i} > 22 \end{aligned}$$

I_{199i} = effect of soil temperature on uptake of phosphate for ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

and,

$$\begin{aligned} I_{200i} &= \exp \{ [X_{13,3,4}\ X_{4,1} /(X_{5,1}\ X_{3,1}) - 0.001]\ 173.29\} - 1 && \text{(App. B.,I, No. 133)} \\ &= 0 && X_{13,3,4}\ X_{4,1} /(X_{5,1}\ X_{3,1}) < 0.001 \end{aligned}$$

I_{200i} = phosphate leakage fraction from roots for ith soil layer
$X_{13,3,4}$ = phosphate content of root compartment, g-P/ m^2 (see App. A., No. 50)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2

and,

$$I_{201i} = 10 (I_{32} - 0.073) \qquad \text{(App. B.,I, No. 134)}$$
$$= 1 \qquad I_{32} > 0.173$$
$$= 0 \qquad I_{32} < 0.073$$

I_{201i} = effect of soil moisture on uptake of phosphate for ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer

and,

$$I_{202i} = F_{6i} / F_5 \qquad \text{(App. B.,I, No. 135)}$$
$$= 0 \qquad F_{6a} = 0$$

I_{202i} = transpiration ratio effect on uptake of phosphate for ith soil layer
F_5 = flow of moisture from all soil layers combined, mm/m^2/day, to the leaf surface via roots and stems (see App. A., No. 5)
F_{6i} = flow of moisture from the ith soil layer to the leaf surface via the plant roots and stems , mm/m^2/day (see App. A., No. 6)

and finally,

$$I_{203i} = 1.21 \{ 1 - \exp[-0.693 X_{13,3,3} / (0.014 X_{3,1})]\} \qquad \text{(App. B.,I, No. 136)}$$
$$= 0.001 \qquad I_{203i} < 0.001$$

I_{203i} = effect of nitrate-N concentration in roots on uptake of phosphate for ith soil layer
$X_{3,1}$ = live root compartment, g/ m^2
$X_{13,3,3}$ = live root nitrate and ammonium content , g-N /m^2 (see App. A., No. 51)

Hence, the daily flow or transport of labile phosphate (via soluble phosphate) from soil compartment to the root compartment can be estimated as:

$$F (X_{12,i,5} : X_{12.i,4} , X_{13,3,4}) = I_{198i} I_{204} X_{3,1} X_{5,1} /X_{4,1} \qquad \text{(App. A., No. 50)}$$

$F (X_{12,i,5} : X_{12.i,4} , X_{13,3,4})$ = phosphate transport from the ith soil layer to plant roots, g-PO_4 / m^2 /day
I_{198i} = adjusted plant root uptake g-PO_4 / g -root/ m^2 /day of inorganic phosphate for ith soil layer
$X_{12,i,5}$ = labile soil phosphate content (compartment), g-PO_4 /m^2, in the ith soil layer
$X_{12,i,4}$ = soluble soil phosphate content (compartment), g-PO_4 /m^2 , in the ith soil layer
$X_{13,3,4}$ = root phosphate content (compartment), g-PO_4 /m^2 , in the ith soil layer
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{204} = uptake coefficient

where,

$$I_{204} = 0.5 \qquad \text{(App. B.,I, No. 137)}$$
$$= 1 \qquad (C_3 \leq 50)$$

I_{204} = uptake factor
C_3 = mean life-span, years, of the plant type simulated

The resulting phosphate concentrations in are limited in this model to be less than or equal to 0.5% of the root biomass expressed on a dry weight basis after contributions from all soil layers into the roots have been combined.

Ammonium and nitrate ion uptakes by plant roots are initially estimated from estimates of the densities of these ions in soil:

$$I_{205i} = 0.002 / (1 + 84 / I_{206i}) + 0.004 / (1 + 4.8 / I_{206i}) \qquad \text{(App. B.,I, No. 138)}$$

$$I_{207i} = 0.0002 / (1 + 84 / I_{208i}) + 0.00004 / (1 + 4.8 / I_{208i}) \qquad \text{(App. B.,I, No. 139)}$$

I_{205i} = unadjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{207i} = unadajusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, from the ith soil layer

where,

$$I_{206i} = 1.4e06\ I_{209i} \qquad \text{(App. B.,I, No. 140)}$$

$$I_{208i} = 1.4e06\ I_{210i} \qquad \text{(App. B.,I, No. 141)}$$

I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, for the ith soil layer
I_{209i} = nitrate concentration, g-N/ g-soil, in the ith soil layer
I_{210i} = ammonium concentration, g-N/ g-soil, in the ith soil layer

A reverse flow of nitrate-N and ammonium-N back to the soil is modeled to prevent excess root concentrations of this nutrient. Also, the effects of soil temperature, evapotranspiration ratio, and the effect of soil matric potential become part of the formulation:

$$I_{211i} = (I_{205i} - I_{212i})\ I_{199i}\ I_{202i}\ I_{213i} \qquad \text{(App. B.,I, No. 142)}$$

$$I_{214i} = (I_{207i} - I_{215i})\ I_{199i}\ I_{202i}\ I_{213i} \qquad \text{(App. B.,I, No. 143)}$$

I_{211i} = adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{214i} = adjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{205i} = unadjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{207i} = unadajusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{199i} = effect of soil temperature on uptake of nitrate and ammonium ions for ith soil layer
I_{202i} = transpiration ratio effect on uptake of nitrate and ammonium ions for ith soil layer
I_{212i} = nitrate ion back-flow rate, g-N /g-root / m^2 / day, to the ith soil layer
I_{215i} = ammonium ion back-flow rate, g-N /g-root / day, to the ith soil layer
I_{213i} = effect of matric potential (Ψ) on ammonium and nitrate uptake from the ith soil layer

where,

$$I_{212i} = I_{205i} \exp\{ [X_{133,3}\ X_{4,1} / (X_{3,1}\ X_{5,1}) - 0.006]\ I_{216i} \} - 1 \qquad \text{(App. B.,I, No. 144)}$$

$$I_{215i} = I_{207i} \exp\{ [X_{13,3,3} X_{4,1} / (X_{3,1} X_{5,1}) - 0.006] I_{217i} \} - 1 \qquad \text{(App. B.,I, No. 145)}$$

I_{212i} = nitrate ion back-flow rate, g-N /g-root / m^2 / day, to the ith soil layer
I_{215i} = ammonium ion back-flow rate, g-N /g-root / day, to the ith soil layer
I_{205i} = un-adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{207i} = un-adajusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
$X_{13,3,3}$ = live root nitrate and ammonium content , g-N /m^2 (see App. A., No. 51)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{216i} = nitrate ion soil density back-flow factor for the ith soil layer
I_{217i} = ammonium ion soil density back-flow factor for the ith soil lay

where,

$$I_{216i} = 35 \exp(-0.15 I_{206i}) + 40 \qquad \text{(App. B.,I, No. 146)}$$

$$I_{217i} = 35 \exp(-0.15 I_{208i}) + 40 \qquad \text{(App. B.,I, No. 147)}$$

I_{216i} = nitrate ion soil density back-flow factor for the ith soil layer
I_{217i} = ammonium ion soil density back-flow factor for the ith soil layer
I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, for the ith soil layer

and,

$$\begin{aligned} I_{213i} &= 0.4 - 0.0133 I_{218i} && \text{(App. B.,I, No. 148)} \\ &= 1 - 0.0533 I_{218i} && I_{218i} \le 15 \\ &= 0 && I_{218i} > 30 \end{aligned}$$

I_{213i} = effect of matric potential (Ψ) on ammonium and nitrate ion uptake from the ith soil layer
I_{218i} = soil matric potential, Bars, for the ith soil layer

where,

$$\begin{aligned} I_{218i} &= (0.6 - I_{219i}) 0.3 /(0.6 - I_{220i}) && I_{219i} > I_{220i} && \text{(App. B.,I, No. 149)} \\ &= 0.3 + (I_{220i} - I_{219i}) 2.7 /(I_{220i} - 0.16) && I_{219i} \le I_{220i} \\ &= 3 + (0.16 - I_{219i}) 12 /(0.16 - I_{221i}) && I_{219i} < 0.16 \\ &= 15 + (I_{221i} - I_{219i}) 2000 && I_{219i} \le I_{221i} \\ &= 0 && I_{218i} \le 0 \\ &= 15 && I_{218i} \ge 15 \end{aligned}$$

I_{218i} = soil matric potential, Bars, for the ith soil layer
I_{219i} = available moisture volume, ml, of the ith soil layer
I_{220i} = available moisture volume, ml, of the ith soil layer at field capacity
I_{221i} = available moisture volume, ml, of the ith soil layer at wilt point

where,

$$I_{219i} = 1.4 X_{1,i} / I_{37i} \qquad \text{(App. B.,I, No. 150)}$$

$I_{220i} = 1.4\ I_{58i} / I_{37i}$ (App. B.,I, No. 151)

$I_{221i} = 1.4\ I_{22i} / I_{37i}$ (App. B.,I, No. 152)

I_{219i} = available moisture volume, ml, of the ith soil layer
I_{220i} = available moisture volume, ml, of the ith soil layer at field capacity
I_{221i} = available moisture volume, ml, of the ith soil layer at wilt point
$X_{1,i}$ = ith soil layer moisture content, mm
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of the jth plant type in a given region (input)
I_{22i} = soil moisture content, mm, for the ith soil layer of the jth plant at wilt point (input)
I_{37i} = thickness of ith soil layer, mm

The bulk density of the soil is assumed to be 1.4 g /cm^3 for the available moisture volume estimates.

Before incrementing the ammonium and nitrate nitrogen compartment of the model ($X_{13,3,,3}$) with the corresponding adjusted uptake rates (I_{211} and I_{214}, respectively), the nitrogen concentration in the plant root is adjusted due to plant phenological development[35] which is expressed as a function of the fraction of the growth period expended during the year:

$$I_{222} = I_{222p}\ X_{4,1} / X_{5,1} + 0.024 - 0.12\ I_{223} \quad \text{(App. B.,I, No. 153)}$$
$$= I_{222p} \qquad I_{223} > 0.2$$
$$= 0.06 \qquad I_{122} > 0.06$$
$$= 0.006 \qquad I_{122} \leq 0.006$$

I_{222} = root nitrogen concentration, g-N/gdwt, from NH_4^+, NO_3^- uptake
I_{222p} = previous root nitrogen concentration, g-N/ gdwt, from NH_4^+, NO_3^- uptake
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
I_{223} = fraction of growth period expended

where,

$$I_{223} = (I_5 - I_{135}) / (I_{136} - I_{135}) \quad \text{(App. B.,I, No. 154)}$$
$$= 1 \qquad I_5 > I_{136};\ I_5 < I_{135}$$

I_5 = julian day number (1 - 365)
I_{135} = beginning of frost-free period for the region, julian day number
I_{136} = end of frost-free period for the region, julian day number

The transport of nitrate and ammonium ions from the soil compartment to the root compartment of a plant can be estimated as:

$$F\ (X_{12,i,2} + X_{12,i,3}\ ,\ X_{13,3,3}\ ) = (I_{211i} + I_{214i}\ )\ I_{204}\ X_{3,1}\ X_{5,1}\ /X_{4,1} \quad \text{(App. A., No. 51)}$$

$F\ (X_{12,i,2} + X_{12,i,3}\ ,\ X_{13,3,3}\ )$ = nitrogen transport from the ith soil layer to plant roots, g-N / m^2 /day, from NO_3^- and NH_4^+ uptake
I_{211i} = adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
$_{214i}$ = adjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer

$X_{12,i,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , of the ith soil layer (see App. A., No. 55)

$X_{12,i,3}$ = soil ammonium ion content (compartment), g-NH_4 /m^2
$X_{13,3,3}$ = root nitrogen content (compartment), g-N /m^2 , of the ith soil layer (see App. A., No. 51)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{204} = uptake coefficient

Hence, the total transport of ammonium and nitrate ions from the soil compartment to the root compartment of a plant is the sum contributions from all soil layers. The resulting ammonium and nitrate nitrogen concentrations in the plant roots (I_{222}) are not allowed to exceed 0.06 g-N/gdwt

Besides the transport of nitrate and ammonium ion from the soil compartment to roots of plants, other processes are modeled that affect the total content of these ions in the soil. These processes involve the denitrification of nitrate to nitrogen gas (sink), the nitrification of ammonium ions to nitrate, and the addition of both ammonium and nitrate ions to the top layer of the soil from dry deposition and from precipitation events (source). The transport of these ions from the atmosphere in the form of dry deposition and precipitation events is estimated as:

$$F(P_{ni}, X_{12,1,2}) = 0.5\,(0.00055 + 0.0006\,P_p) / (1400\,I_{37i}) \qquad \text{(App. A., No. 52)}$$

$$F(P_{nh}, X_{12,1,3}) = 0.5\,(0.00055 + 0.0006\,P_p) / (1400\,I_{37i}) \qquad \text{(App. A., No. 53)}$$

$F(P_{ni}, X_{12,1,2})$ = transport of nitrate ion from atmosphere to 1st soil layer, g-NO_3 /g-soil /day
$F(P_{nh}, X_{12,1,3})$ = transport of ammonium ion from atmosphere to 1st soil layer, g-NH_4 /g-soil /day
P_{ni} = atmospheric source term for nitrate ion, g-NO_3 /m^2
P_{nh} = atmospheric source term for ammonium ion, g-NH_4 /m^2
P_p = precipitation , mm /day
$X_{12,1,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the 1st layer of soil
$X_{12,1,3}$ = soil ammonium ion content (compartment), g-NH_4 /m^2 , in the 1st layer of soil
I_{37i} = thickness of 1st (i=1) soil layer, mm

It should be noted that the transport rate is in concentration units, and that correct transport units are the same as the source terms, which can be obtained by omitting the denomenator of the above relationships. The nutrients deposited on the surface are then transported into deeper soil horizons by precipitation and eluviational process to be described later in this document. Finally, the above transport estimate in concentration units assumes a bulk density of 1.4 g/cm^3 .

The production of nitrate ion from the nitrification of ammonium ion in the soil is estimated assumes both a forward and a back reaction as a first estimate:

$$I_{224i} = 0.025 + 0.437 \exp(-0.00575\, I_{210i}\, 1.4e06) \qquad \text{(App. B.,I, No. 155)}$$

$$I_{226i} = 0.0631 / (1.4e06\, I_{209i})^{0.4} \qquad \text{(App. B.,I, No. 156)}$$

so that the net conversion of ammonium ion to nitrate ion which is also affected by the soil temperature and the soil matric potential is estimated as:

$$I_{225i} = (I_{224i}\, I_{210i} - I_{226i}\, I_{209i})\, I_{227i}\, I_{228i} \qquad \text{(App. B.,I, No. 157)}$$
$$= 0 \qquad I_{225i} < 0$$

I_{224i} = conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil /day for the ith soil layer
I_{226i} = conversion of nitrate ion to ammonium ion, g-NH_4/ gdwt-soil /day for the ith soil layer
I_{209i} = nitrate concentration, g-N/ g-soil, in the ith soil layer
I_{210i} = ammonium concentration, g-N/ g-soil, in the ith soil layer

I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer
I_{227i} = effect of temperature on soil nitrificaton in the ith soil layer
I_{228i} = effect of soil matric potential on soil nitrification in the ith soil layer

where,

$$I_{227i} = 0.20 + 0.043\, I_{172i} \qquad \text{(App. B.,I, No. 158)}$$
$$= 0.0114\, I_{172i} \qquad I_{172i} \leq 7$$
$$= 1 \qquad I_{172i} > 28$$

I_{227i} = effect of temperature on soil nitrificaton in the ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

and,

$$I_{228i} = \exp(-0.693\, I_{218i} / 8) \qquad \text{(App. B.,I, No. 159)}$$

I_{228i} = effect of soil matric potential on soil nitrification in the ith soil layer
I_{218i} = soil matric potential, Bars, for the ith soil layer

Denitrification of nitrate ion to nitrogen gas is related to the net nitrification of ammonium ion to nitrate, as affected by the effects of matric potential and soil temperature:

$$I_{229i} = 0.0033\, I_{225i}\, I_{230i}\, I_{231i} \qquad \text{(App, B.,I, No. 160)}$$
$$= 0 \qquad I_{229i} < 0$$

I_{229i} = denitrification rate, g-NO_3 / gdwt-soil /day, for the ith soil layer
I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer
I_{230i} = effect of soil temperature on denitrification for the ith soil layer
I_{231i} = effect of soil matric potential on denitrification for the ith soil layer

where,

$$I_{230i} = -0.31 + 0.03266\, I_{172i} \qquad \text{(App. B.,I, No. 161)}$$
$$= 0.002\, I_{172i} \qquad I_{172i} \leq 10$$
$$= 1 \qquad I_{172i} > 40$$

I_{230i} = effect of soil temperature on denitrification for the ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

and,

$$I_{231i} = 1.25\,(1 - I_{218i}) \qquad \text{(App. B.,I, No. 162)}$$
$$= 0 \qquad I_{218i} > 1$$
$$= 1 \qquad I_{218i} \leq 0.2$$

I_{218i} = soil matric potential, Bars, for the ith soil layer
I_{231i} = effect of soil matric potential on denitrification for the ith soil layer

The flow or transport of nitrogen out of the soil through denitrification is estimated as:

$$F(X_{12,i,2}, P_{ng}) = \sum_{i=1}^{n} I_{229i} \qquad \text{(App. A., No. 54)}$$

$F(X_{12,i,2}, P_{ng})$ = denitrification rate of nitrate to nitrogen gas, g-N/ m^2 /day
I_{229i} = denitrification rate, g-NO_3 / gdwt-soil /day, for the ith soil layer
P_{ng} = nitrogen gas production from nitrate ion, g-N/m^2 /day
$X_{12,i,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the ith soil layer

and the flow or conversion of ammonium to nitrate ion through nitrification is estimated as:

$$F(X_{12,i,3}, X_{12,i,2}) = \sum_{i=1}^{n} I_{225i} \qquad \text{(App. A., No. 55)}$$

$X_{12,i,3}$ = soil ammonium ion content (compartment), g-NH_4 /m^2 , in the ith soil layer
$X_{12,i,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the ith soil layer
I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer

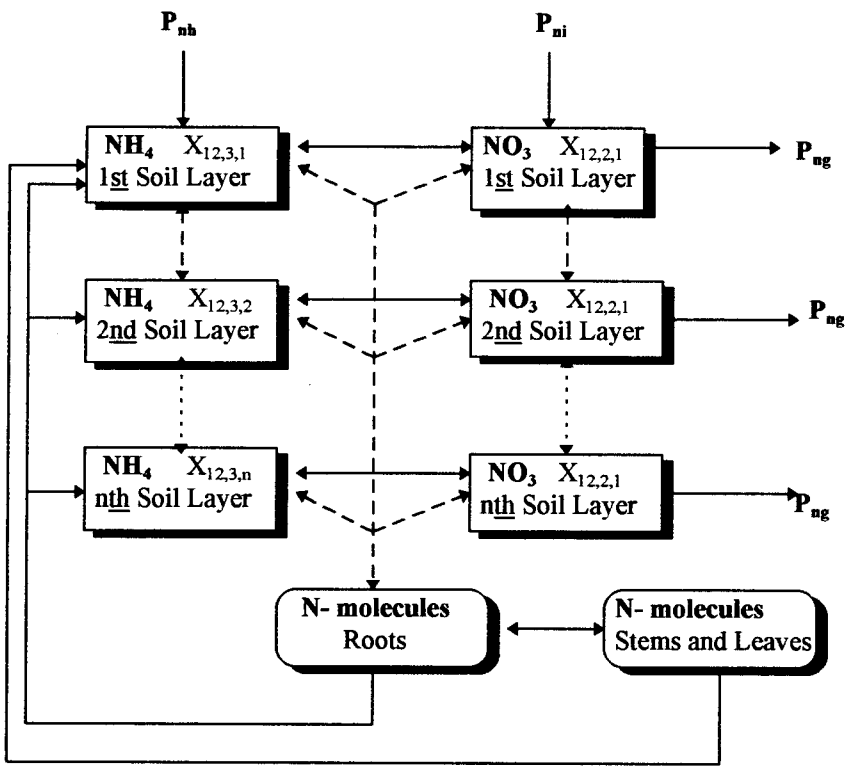
The strategies for ammonium - nitrate and phosphate transport and exchange process in plant uptake utilized in this model are presented in Fig. 11 and Fig. 12, respectively, although , the process involving recycling through plant die-off has yet to be described in a later section of this report.

6.2. Animals

The modeling of biomass flow through animal systems[35,36] is separate from the model employed to simulate biomass flow to humans[2], although humans can be simulated using the former modeling strategy as well. A situation where one might want to simulate a human population using the animal model is where the latter lives in a hunter-gather environment, but it has not been attempted at the time of this writing. The modeling of animal populations in BIOTRAN.2 are basically the modeling of mammalian and invertebrate populations (insects) where the latter is required as part of the food supply for the former in some cases. Much more effort has gone into the modeling of mammalian populations than invertebrates, hence the quality of modeling of the former is much greater and more versatile than the invertebrate model. In the previous documentation of the model BIOTRAN[1] , the only model available was for an herbivore mammal (ruminant) which was then used to provide a food supply (beef, milk) to humans, although the human model HUMTRN[2] was yet to be developed for incorporation into the larger model. It was possible to simulate the growth of beef and dairy cows, elk, and deer at most [32]. The current mammalian model is now capable of simulating the growth of herbivores (ruminant and non-ruminant), omnivores, and carnivores, although more specific metabolic input information is required than before. However, the model has retained the supplementary feed and grazing allocation strategies developed in the earlier model versions, and it has been extended to include animal tissues as consumption items as well. Finally, the population dynamics portion of the model has been extended and improved to handle a variety of population structures not previously addressed. The discussion to follow will include the modeling of food preference and feeding strategies, food consumption, metabolic processes, population structure, animal growth processes , and the population dynamics currently used by the model.

6.2.1. Food Preference and Feeding Strategies

Food preference and feeding strategies are modeled to depend on user specified input for the animal population being simulated. The number of different feeding strategies specified (I_{232}) for a given animal type indicates the number of different plant food combinations which will be used in a simulation for given year sub-intervals. The number of year sub-intervals (I_{233}) into which the total simulation interval is divided to allow changes in food sources is also specified by input, as is the the number of daily (I_{234}) sub-intervals within a calendrical year for all years included with a given year sub-interval;

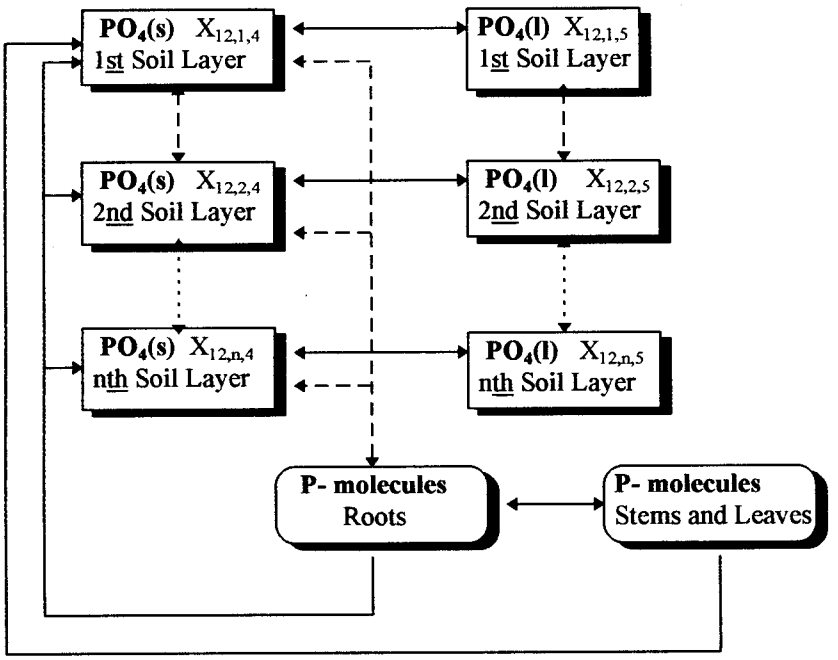


Legend

$X_{12,i,3}$ = ith soil layer NH_4 content, g / m^2
$X_{12,i,2}$ = ith soil layer NO_3 content, g / m^2
P_{nh} = atmospheric NH_4 increment to top soil layer, g-NH_4 / m^2 /day
P_{ni} = atmospheric NO_3 increment to top soil layer, g-NO_3 / m^2 /day
P_{ng} = nitrogen gas releases from denitrification of NO_3 in soil, g-N_2 / m2 /day

Fig. 11

Ammonium-nitrate Transport and Exchange Strategies Employed in BIOTRAN.2



Legend

$X_{12,i,4}$ = ith soil layer soluble $PO_4(s)$ content, g / m^2
$X_{12,i,5}$ = ith soil layer labile (insoluble) $PO_4(l)$ content, g / m^2

Fig. 12

Phosphate Transport and Exchange Strategies Employed in BIOTRAN.2

The latter marks changes in food source(s) during all years included within a given year sub-interval. The year sub-intervals (I_{235}), and day sub-intervals (I_{236}) within each year sub-interval are user specified to reflect seasonal changes in food resources. Having established day sub-intervals for specific year sub-intervals as specified above, then the next task is to specify the number (I_{237}) of plants which are available for consumption, the identification of the plants themselves (I_{238} , see C_8 in App. C. for selection possibilites), the fraction of the animal's diet contributed by each identified plant (I_{239} ,I_{239b}), and a food preference/selection ranking scheme for the same (I_{240} ,I_{240b}). An illustration of the feeding strategy is presented in Fig. 13. using an example where two year sub-intervals having different day sub-intervals are compared.

The sum of the diet fractions of plants composing the diet of an animal must be equal to one with one exception: when an animal is also a predator with a specified diet fraction(s) (I_{241}) of animal(s). When the latter condition occurs, then the sum of all plant and animal food contribution fractions to the animal's diet must be equal to one. The food preference strategy ($1 \geq I_{240b} \leq 3$) is specific for grazed plants where all plant parts are consumed, whereas the food selection strategy ($I_{240b} > 3$) specifies certain portions of plants which are consumed as supplementary feed:

I_{240b}	**Description**
1	first preference for grazed food
2	second preference for grazed food
3	third preference for grazed food
4	total harvested above ground biomass including seeds (supplementary feed)
5	harvested grains or fruits (supplementary feed)
6	total harvested above ground biomass excluding seeds (supplementary feed)
7	currently same as 5

The food preference strategy operates independently or in conjunction with selection strategies for supplementary feeds. The model uses the food preference strategies to estimate the fraction of the diet which is composed of a certain range plant:

$$
\begin{aligned}
I_{242\,i} &= 0.7\,(1 - I_{243})\,(I_{244b})_j && I_{240b} = 1, i=1 \\
&= 0.2\,(1 - I_{243})\,(I_{244b})_j && I_{240b} = 2, i=2 \\
&= 0.1\,(1 - I_{243})\,(I_{244b})_j && I_{240b} = 3, i=3 \\
&= I_{241k}\,(I_{244b})_j && i > 3
\end{aligned}
\qquad \text{(App. B.,I, No. 163)}
$$

where,

$$
I_{243} = \sum_{j=1}^{I_{273}} \sum_{k=1}^{I_{274k}} I_{241k}\,(I_{244b})_j \qquad \text{(App. B.,I, No. 164)}
$$

I_{273} = number of regions simulated (input)

I_{274k} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannabalism is being modeled) in a given region

I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)

I_{241k} = initial diet fraction of the kth animal prey type in the food of a specific animal predator type (input)

I_{242i} = initial food fraction contribution from the ith preference group ($i \leq 3$) of plants of a given region

I_{243} = total fraction of food content derived from animal food from all regions combined

I_{244j}, $(I_{244b})_j$ = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

If the plant food preference value for a given plant type (I_{240b}) is greater than three, then the un-adjusted diet fractions for specific plant foods default to the input diet fraction (I_{239}, I_{239b}) described previously, however, for lower input values of this parameter, the latter is used only to determine the total un-adjusted amount of plant food in the diet apart from that designated as animal food. Finally, for strict carnivores, the un-adjusted fractions for animal sources of food are the only sources of food specified as input (I_{241}).

The un-adjusted plant and animal diet fractions are then applied to the food needs of the animal population in question , and the diet fractions are adjusted accordingly if food availability is limiting in any specific food category being used to sustain that population. The amount of food required to sustain a specific mammalian population is estimated as:

$$I_{245k} = \sum_{m=1}^{I_{248k}} F(P_f, X_{14,m,k})\, I_{246m,k}\, I_{247k} \qquad \text{(App. B.,I, No. 165)}$$

I_{245k} = food requirement, kg /day, by the kth mammalian population

$F(P_f, X_{14,m,k})$ = transport of food, kg/day, to an animal of the mth age group of the kth mammalian population

$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population (see App. B.,V, No. 9)

$I_{246m,k}$ = fraction of the population in the mth age group of the kth mammalian population (see App. B.,I, No. 187)

I_{247k} = total number of animals in the kth mammalian population (initial input)

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

where,

$$F_e(P_f, X_{14,m,k}) = 70\, X_{16,m,k}\, I_{249k}\, I_{250k}\, I_{251m,k}\, I_{252k} / (4500\, I_{253k}) \qquad \text{(App. A., No. 56)}$$

$$\begin{aligned} F(P_f, X_{14,m,k}) &= 70\, X_{15,m,k}\, I_{249k}\, I_{250k}\, I_{251m,k}\, I_{252k} / (4500\, I_{253k}) && \text{(App. A., No. 57)} \\ &= 1.25\, F_e(P_f, X_{14e,m,k}) && F_{57} > 1.25\, F_{56} \\ &= F_e(P_f, X_{14e,m,k}) && F_{57} < F_{56} \\ &= F(P_f, X_{14,m,k})\, I_{254k} / I_{245k} && I_{254k} > I_{245k} \end{aligned}$$

$F(P_f, X_{14,m,k})$ = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population

$F_e(P_f, X_{14,m,k})$ = expected transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population

P_f = transport of food biomass, kgdwt/day, from plant and/or other animal compartments

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population (see App. B.,V, No. 9)

$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population (see App. B.,V, No. 8)

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population (see App. B.,V, No. 6)

$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population (see App. B.,V, No. 7)

I_{245k} = food requirement, kg /day, by the kth mammalian population

I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population
I_{250k} = activity factor for the kth mammalian population (input)
$I_{251m,k}$ = age effect factor on food intake for the mth age group of the kth mammalian population
I_{252k} = food digestibility effect factor for the kth mammalian population
I_{253k} = digestibility fraction for the kth mammalian population (see App. B.,I, No. 182)
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

Whereas the detailed analysis of the food intake relationships just presented will be analyzed in the next sub-section of this report, the current discussion will consider the food availability-food requirement to complete this subsection.

The food requirement (F_{57}) is satisfied by as many as three sources of food: plants on range, supplementary feed derived from harvested plants, and from animal foods as shown in Fig. 14. The preset un-adjusted diet fractions for these food components have been described previously. The next step in the simulation is to determine the availability of these food sources to satisfy dietary needs at the specified diet levels. If the food requirement is not satisfied by using un-adjusted diet fractions (one or more sources of food are not present in sufficient amount(s)), the model sets up an iterative process of as many as ten cycles in an attempt to satisfy dietary needs ($I_{254} = I_{245}$). The latter process results in a modification of the preset un-adjusted diet fractions in all iterations above the first. The amount of food contributed by range plants and harvested plants is estimated as :

$$I_{255k,j} = I_{245k} (I_{244b})_j \qquad \text{(App. B.,I, No. 166)}$$

$$I_{256k} = I_{245k} (1 - (I_{244b})_j) \qquad \text{(App. B.,I, No. 167)}$$

I_{245k} = food requirement, kg /day, by the kth mammalian population
$I_{255k,j}$ = range plants in food intake, kgdwt, for the kth mammalian population from the jth region
I_{256k} = harvested plant/animal in food intake, kgdwt, for the kth mammalian population from
I_{244j}, $(I_{244b})_j$ = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

The total amount of food present in each of the three preference categories for range plants which is used to determine the amount available to a given animal population (P_{fr}) is estimated as:

$$I_{257i,m} = [X_{5,1} + X_{6,1} + X_{6,2} + (X_{4,1} - X_{5,1}) X_{4,1} / X_{5,1}] I_{258i} + I_{259i} (X_{5,2} + X_{4,2}) \qquad \text{(App. B.,I, No. 168)}$$

$$I_{260m} = \sum_{i=1}^{n} I_{257i,m} \qquad \text{(App. B.,I, No. 169)}$$

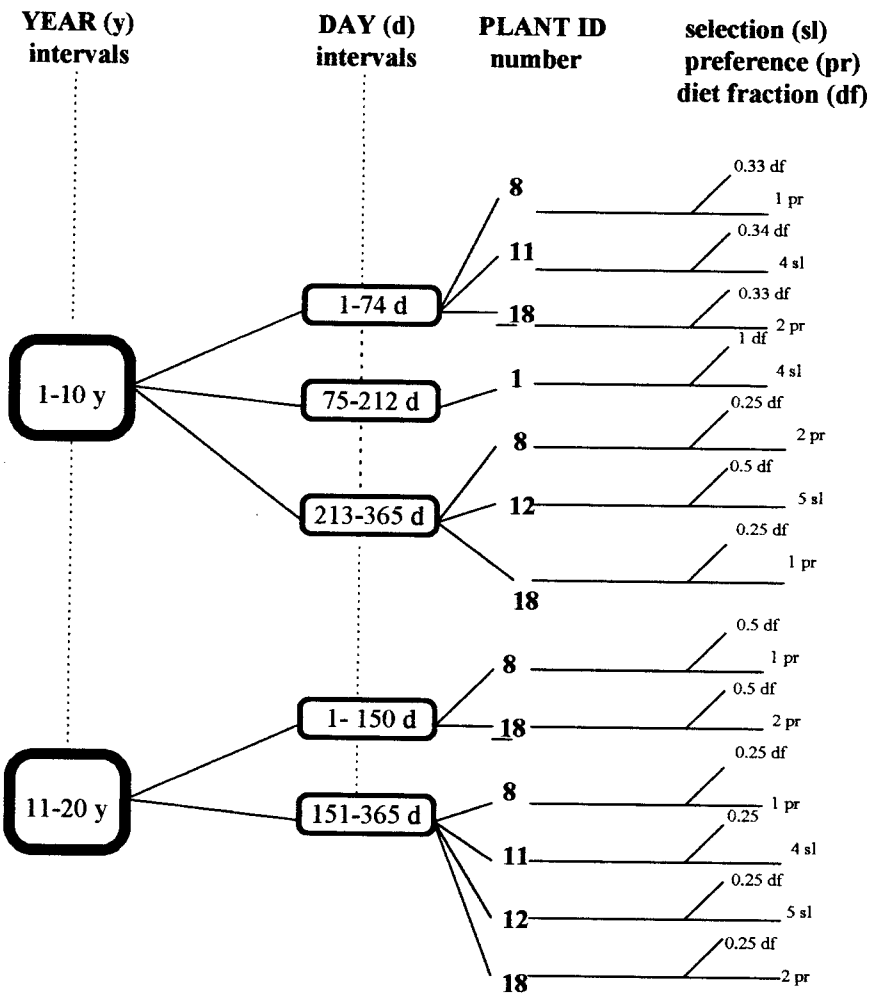
$$I_{258i} = 3 I_{261i} / (3 I_{261i} + I_{262i}) \qquad \text{(App. B.,I, No. 170)}$$

$$I_{259i} = 1 - I_{258i} \qquad \text{(App. B.,I, No. 171)}$$

$$I_{261i} = [X_{5,1} + X_{6,1} + X_{6,2} + (X_{4,1} - X_{5,1}) X_{4,1} / X_{5,1}] \qquad \text{(App. B.,I, No. 172)}$$

$$I_{262i} = X_{5,2} + X_{4,2} X_{5,1} /X_{4,1} \qquad \text{(App. B.,I, No. 173)}$$

$I_{257i,m}$ = total biomass density, gdwt/m^2, for the ith plant in the mth region
I_{258i} = live plant fraction (including seeds) ingested from the ith plant type
I_{259i} = dry plant fraction (excluding seeds) ingested from the ith plant type
I_{260m} = total biomass density, gdwt/m^2, in the mth preference category for range plants
I_{261i} = live plant biomass (including seeds), gdwt/m^2, for the ith plant



legend

plant ID number	plant ID
1	corn or sorghum
8	warm season perennial grass
11	alfalfa
12	winter wheat
18	cool season perennial grass

Notes: example illustrates a 20 year simulation of a domestic beef or dairy cow which is partly or totally on range. The food preferences (pr = 1,2,3) apply to grazing only. Selection numbers (sl = 4,5) apply exclusively to supplementary feed (harvested) where 4= total harvested above ground biomass, and 5= harvested grains. The diet factors (df) are fraction of the diet in the food for supplementary feeds only. For range plants (pr = 1,2,3), the diet factors are used to estimate the total fraction of food which is composed of plants obtained from grazing. The latter is then used to estimate food consumption from such plants based on as many as three preference choices. See the text for a more complete explanation of the selection criteria.

Fig. 13

Example of Food Selection Criteria for an Herbivore in BIOTRAN.2

I_{262i} = dry plant biomass (excluding seeds), gdwt/m^2, for the ith plant
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2
n = number of animals in simulation

This model of ingestion assumes that a grazing animal prefers live (green) biomass including seeds by a three-to-one ratio over dry plant material. Similarly, harvested crops are initialized at specified harvest dates by region for each animal population:

$$\begin{aligned} I_{263i,m} &= X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2} && m=4;\ I_{266i} = I_5 \\ &= X_{6,1} + X_{6,2} && m=5;\ I_{266i} = I_5 \\ &= X_{4,1} + X_{4,2} + X_{5,2} && m=6;\ I_{266i} = I_5 \\ &= X_{6,1} + X_{6,2} && m=7;\ I_{266i} = I_5 \\ &= 0 && I_{263i,m} \leq 20 \end{aligned}$$

(App. B.,I, No. 174)

$$I_{264\,i,m} = 4047\,(I_{263i,m} - 20)\,[\,I_{263i,m} / (X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2})]\,I_{265i}$$

(App. B.,I, No. 175)

$I_{263i,m}$ = harvested biomass, gdwt/m^2, from the ith plant in the mth selection category for a given region
$I_{264i,m}$ = harvested biomass, gdwt, from the ith plant in the mth selection category for a given region
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
I_{266i} = harvesting date(s), julian day, for the ith plant in a given region (input)
I_5 = julian day number (1 - 365)
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

Finally, the amount of food ingested daily from animal prey by a given mammalian population is estimated as:

$$I_{260j} = \sum_{m=1}^{I_{248k}} 500\, X_{14,m,k}\, I_{246m,k}\, I_{247k} / I_{267k} \qquad i > 3;\ I_{248k} > 0$$

(App. B.,I, No. 176)

$$= \sum_{m=1}^{I_{248k}} 1000\, X_{14,m,k}\, I_{246m,k}\, I_{247k} / I_{267k} \qquad i > 3;\ I_{248k} = 0$$

I_{260j} = total biomass density, gfwt/m^2, in the jth selection category of animals
I_{267k} = area grazed, m^2, by the kth mammalian population in a given region
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth animal population (see App. B.,V, No. 9)
I_{246nmk} = fraction of the population in the mth age group of the kth mammalian population; set to 1 for invertebrate populations (App. B.,I, No. 187)
I_{247k} = total number of animals in the kth mammalian population ; invertebrate populations set to 1 (input)

I_{248k} = lifespan, years, of the an animal in the kth mammalian population; set to 0 for invertebrate populations (not used) (input)

where,

$$I_{267k} = \sum_{i=1}^{I_{69}} I_{268i} \qquad 0 \le I_{240b} \le 3 \qquad \text{(App. B.,I, No. 177)}$$

and,

$$I_{268i} = 4047\ I_{265i} \qquad \text{(App. B.,I, No. 177b)}$$

I_{268i} = coverage area, m^2, of the ith plant type for a given region
I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)
I_{69} = number of plants simulated in the a given region (input)
I_{267k} = area grazed, m^2, by the kth animal population in a given region
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)

The above relationship assumes that 50 percent of the weight of a given prey is utilized by its respective predator unless the prey is an invertebrate; is assumed that 100 percent of the weight of latter is utilized by the predator.

The amount of food ingested daily from a given region including both range plants and prey to a given animal consumer population is estimated on a daily and basis as:

$$I_{269j,m,k} = I_{255k}\ I_{270k,j,m}\ I_{271k,j,m} \qquad j > 0 \qquad \text{(App.B.,I, No. 178)}$$

$I_{269j,m,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population
I_{255k} = range plant/animal in food intake, kgdwt /day, for the kth mammalian population from all regions combined
$I_{270k,j,m}$ = fraction of food available from the jth range plant/animal category for the mth region combined for the kth mammalian population
$I_{271k,j,m}$ = fraction of range plant/animal food contribution from the mth region of the jth range plant category for the kth mammalian population

where,

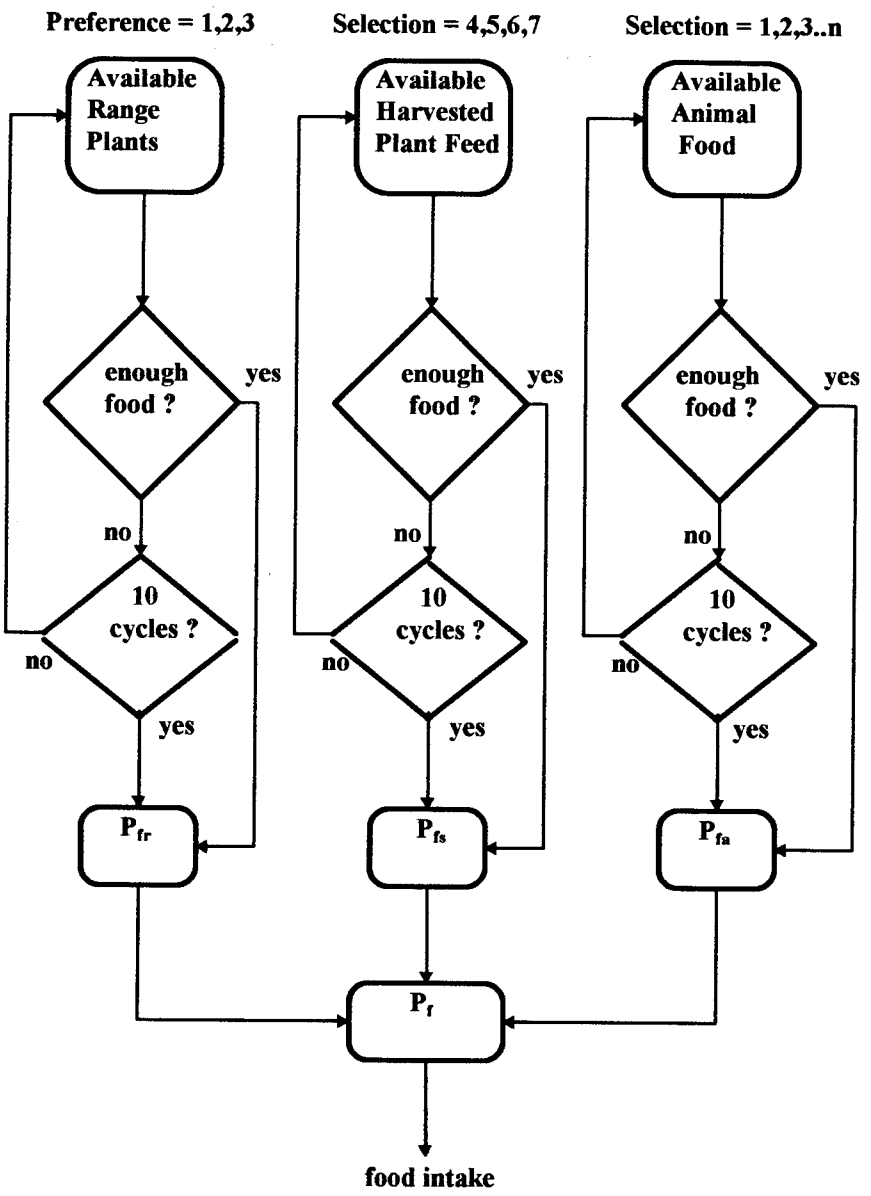
$$I_{270k,j,m} = I_{242j,m}\,(\,I_{260j,m} - 2\,)\,I_{267k,j} / 1000\,) / (I_{255k}\ I_{242j,m}\,) \qquad 0 < j \le 3 \qquad \text{(App.B.,I, No. 179)}$$

$$= I_{242j,m}\,(\,I_{260j,m} - 0.0001\,)\,I_{267k,j} / 1000\,) / (I_{255k}\ I_{242j,m}\,) \qquad j > 3$$

$$= I_{242j,m}\,(\,I_{260j,m} - 0.02\,)\,I_{267k,j} / 1000\,) / (I_{255k}\ I_{242j,m}\,) \qquad j > 3;\ I_{248k} = 0$$

$$= I_{242j,m} \qquad I_{270k,j,m} \ge 1$$

$I_{270k,j,m}$ = fraction of food available from the jth range plant category for the mth region combined for the kth mammalian population
I_{267k} = area grazed, m^2, by the kth mammalian population in a given region



legend

P_{fr} = plant fo intake from range
P_{fs} = food in e from harvested food
P_{fa} = animal food intake from range
P_f = total food intake, all sources
n = number of animals in simulation

Fig. 14

Food Preference/ Selection Strategy From Available Food Sources

I_{255k} = range plant/animal in food intake, kgdwt /day, for the kth mammalian population from all regions combined
$I_{242j,m}$ = food fraction contribution from the jth preference group of plants by the mth region
I_{260i} = total biomass density, gfwt/m^2, in the jth selection category of animals
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

and,

$$I_{271k,j,m} = \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \Big/ \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \qquad 0 < I_{239bk,j,m} \leq 3 \qquad \text{(App. B.,I, No. 180)}$$

$$= \sum_{i=1}^{I_{274k}} I_{241k,i,m} \Big/ \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{274k}} I_{241k,i,m} \qquad 0 < I_{241k,i,m}$$

$I_{271k,j,m}$ = fraction of range plant/animal food contribution from the mth region of the jth range plant/animal category for the kth animal population
I_{273} = number of regions simulated (input)
$I_{274k,m}$ = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannabalism is being modeled) in the mth region
I_{69m} = number of plants simulated in the mth region (input)
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith plant in the mth region for the kth animal population
$I_{241k,i,m}$ = initial diet fraction of the ith animal prey animal type in the food of the kth predator animal for the kth predator population type from the mth region (input)

The contribution of supplementary plant foods to the diet of an animal is not generally used with wildlife animal populatons, but is intended to simulate domestic beef and dairy cattle, however, it is not restricted to the latter. The daily requirement for a given animal population is estimated as:

$$I_{272k} = \sum_{m=1}^{I_{273}} I_{256k,m} \left[\sum_{i=1}^{I_{69m}} I_{239bk,i,m} \Big/ \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \right] \qquad 0 < I_{239bk,j,m} > 3 \qquad \text{(App. B.,I, No. 181a)}$$

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population
$I_{256k,m}$ = harvested plants in food intake, kgdwt, for the kth animal population from the mth region
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith plant in the mth region for the kth animal population
I_{273} = number of regions simulated (input)
I_{69m} = number of plants simulated in the mth region (input)

The total amount of food available for consumption by a given animal population is thus estimated as:

$$I_{254k} = I_{272k} + \sum_{j=1} I_{269j,m,k} \qquad \text{(App. B.,I, No. 181b)}$$

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population
$I_{269j,m,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population

The above relationships show that animal consumption of foods can be estimated on a multi-regional basis, and that the fractional amounts of each food item can be varied depending on the availability of the different food sources as shown in Fig. 14. Hence, the amount of food required by an animal during the year may vary depending on the nutritive value of the ingested food. As will be shown later in this report, a corresponding variation in contaminant intake is associated with both the quality and quantity of food ingested by a given animal type.

Finally, the digestible fraction of the food intake is estimated as:

$$I_{253k} = 0.75\, I_{275k} + 0.45\,(1 - I_{275k}) \qquad \text{(App., B.,I, No. 182)}$$

where the range plant weighted component of digestibility is estimated as:

$$I_{275kr} = \sum_{m=1}^{I_{273}} \sum_{j=1}^{3} I_{269j,m,k}\, I_{276jjm} / I_{254k} \qquad \text{(App.,B.,I, No. 183a)}$$

and for animal tissues (assuming 100% disgestibility):

$$I_{275ka} = \sum_{m=1}^{I_{273}} \sum_{j=4}^{7} I_{269j,m,k} / I_{254k} \qquad \text{(App.,B.,I, No. 183b)}$$

where,

$$I_{276j,m} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{258i,m}\, I_{257i,m} / I_{260j,m} \qquad 0 \le j \le 3 \qquad \text{(App.,B.,I, No. 184)}$$

$I_{276j,m}$ = weighted green or live fraction in consumed range plants
I_{275k} = digestible fraction from all sources of food in the diet of the the kth mammalian population (see App. B.,I, No. 183d)
I_{275kr} = digestible fraction from range plant food consumption for the kth mammalian population
I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population
I_{69} = total number of plants in a given region (input)
I_{253k} = digestibility fraction for the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
$I_{257i,m}$ = total biomass density, gdwt/m^2, for the ith plant in the mth region
I_{258i} = live plant fraction (including seeds) ingested from the ith plant type from the mth region
I_{260j} = total biomass density, gdwt/m^2, in the jth preference category for range plants of the mth region
$I_{269j,m,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population

I_{273} = number of regions simulated (input)

Similarly, the weighted green or live fraction in consumed supplementary feeds are estimated as:

$$I_{275ks} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} (X_{4,1} + X_{6,1} + X_{6,2}) \, I_{239bk} / (X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2})_{i,m} / I_{276k} \qquad I_{240b} = 4 \qquad \text{(App.,B.,I, No. 183c)}$$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bk,m} / I_{276k} \qquad I_{240b} = 5$$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} (X_{4,1} \; I_{239bk} / (X_{4,1} + X_{5,2} + X_{4,2})_{i,m} / I_{276k} \qquad I_{240b} = 6$$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bki,m} / I_{276k} \qquad I_{240b} = 7$$

where,

$$I_{276k} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bki,m} \qquad \text{(App. B.,I, No. 185)}$$

I_{275ks} = digestible fraction from supplementary plant food consumption for the kth mammalian population
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith plant in the mth region for the kth animal population
I_{276k} = sum of the selection categories for supplementary feeds selected for consumption by the kth animal population

The digestibility of supplementary feeds is assumed to equal 70% in all cases. The weighted digestibilty of all range plant and animal and supplementary feeds is thus estimated as:

$$I_{275k} = [(I_{254k} - I_{272k}) (I_{275kr} + I_{275ka}) + I_{272k} \, I_{275ks}] / (I_{254k} + I_{272k}) \qquad \text{(App. B.,I, No. 183d)}$$

I_{275k} = digestible fraction from all sources of food in the diet of the the kth mammalian population
I_{275kr} = digestible fraction from range plant food consumption for the kth mammalian population
I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population
I_{275ks} = digestible fraction from supplementary plant food consumption for the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

The resultant digestibility of the ingested food is thus estimated from a complex mixture of the different food sources outlined in this sub-section, and can be modeled on a multi-regional basis to satisfy occupancy factors for specific animals with large home ranges such as mule deer or coyotes.

6.2.2. Food Consumption, Metabolic Processes and Population Structure

Food consumption, metabolic processes and population structure strategies were incorporated into the biomass flows F_{56} and F_{57} mentioned in the last sub-section where it was shown that food intake was indirectly proportional to body weight via the metabolic biomass:

$$X_{15,m,k} = 5.623e\text{-}03\ (1000\ X_{14,m,k})^{0.75} \qquad I_{248k} > 0 \qquad \text{(App. B.,V, No. 6)}$$

$$X_{16,m,k} = 5.623e\text{-}03\ (1000\ X_{14e,m,k})^{0.75} \qquad I_{248k} > 0 \qquad \text{(App. B.,V, No. 7)}$$

where,

$$X_{14e,m,k} = I_{277k}\ [\ 1 - \exp(-9\ I_{278m,k} / \{365\ I_{248k}\})] \qquad I_{248k} > 0 \qquad \text{(App. B.,V, No. 8)}$$

$$= I_{277k}\ \{\ 1 - \exp[\ -9\ (I_{278m,k} - 2.19 I_{248k}) / (365\ I_{248k})\ ]\} \qquad I_{248k} > 0;\ m=1$$

$$X_{14,m,k} = X_{14em,k} \qquad I_{29}=1;\ I_5=1 \qquad \text{(App. B.,V, No. 9)}$$

and,

$$I_{278m,k} = 365\ (I_{29} - 1) + I_5 \qquad I_{248k} > 0 \qquad \text{(App. B.,I, No. 186)}$$

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population
I_5 = julian day number (1 - 365)
I_{29} = julian year number of specific year of simulation
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)
I_{277k} = maximum body weight, kgfwt, for an animal in the kth mammalian population (input)
$I_{278m,k}$ = age, days, of the mth age group of the kth mammalian population

The population of specific mammalian animal types is modeled either through user input or through default selected population structures available in the model. The fraction of the animals in a given age group of a population which has been used in the previous sub-section to estimate food intake requirements for that age group, and for the entire population is estimated as:

$$I_{246m,k} = (I_{279am,k}\ I_{280ak} + I_{279bm,k}\ I_{280bk}) / I_{247k} \qquad \text{(App. B.,I, No. 187)}$$

$I_{246m,k}$ = fraction of the population in the mth age group of the kth mammalian population
I_{247k} = total number of animals in the kth mammalian population (initial input)
$I_{279am,k}$ = fraction of male population in the the mth age group of the the kth mammalian population (initial input or default)
$I_{279bm,k}$ = fraction of female population in the the mth age group of the the kth mammalian population (initial input or default)

I_{280ak} = total male population of the kth mammalian population

I_{280bk} = total female population of the kth mammalian population

where,

$$I_{280ak} = I_{247k} \; (1 - I_{281k}) \qquad \text{(App. B,I, No. 188)}$$

$$I_{280bk} = I_{247k} \; I_{281k} \qquad \text{(App. B.,I, No. 189)}$$

I_{280ak} = total male population of the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{281k} = initial female fraction of population (input)
I_{247k} = total number of animals in the kth mammalian population (initial input)

The fraction of animals in any age group of a specific animal population are either entered as input into the model, or the population structure by age is estimated internally depending on the user's choice of the structure in question. Hence, there are four choices which the user has in characterizing a particular animal population structure:

- *user specified* population fraction for each age group
- estimated population fractions from a *parabolic* population structure for each age group
- estimated population fractions from a *hyperbolic* population structure for each age group
- estimated population fractions from an *exponential* population struture for each age group

The estimated initial population fractions are estimated by the following relationships for females assuming that they apply to the male fractions as well ($I_{279am,k} = I_{279bm,k}$):

$$I_{279m,k} = (I_{282m,k} - I_{282m-1,k}) \qquad \text{(App. B.,I, No. 190)}$$

$$\begin{aligned} I_{282m,k} &= [(I_{283k} - I_{284m,k}) / I_{283k}]^{1.5} && I_{285k}=2 \qquad \text{(App. B.,I, No. 191)} \\ &= \exp[-(10 \, ln2 / I_{283k}) (I_{283k} - I_{284m.k})] && I_{285k}=3 \\ &= [\, ln \, I_{284m,k} + I_{284m,k} / I_{283k} + 1/ I_{283k}]/ [ln \, I_{283k} - 1 + 1/I_{283k}] && I_{285k}=4 \end{aligned}$$

where,

$$I_{283m} = 365 \, I_{286k} \qquad I_{248k} > 0 \qquad \text{(App. B.,I, No. 192)}$$

$$I_{284m,k} = \sum_{i=1}^{m} 365_{i,k} \qquad I_{248k} > 0 \qquad \text{(App. B.,I, No. 193)}$$

$I_{279m,k}$ = fraction of population in the the mth age group of the the kth mammalian population (male and female fractions assumed to be equal to each other)
$I_{282m,k}$ = population fraction parameter (upper integral) for the mth age group of the kth mammalian population
$I_{282m-1,k}$ = population fraction parameter (lower integral) for the mth age group of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
$I_{284m,k}$ = maximum age,years, of the mth age group of the kth mammalian population
I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
I_{286k} = oldest age group, years, for the kth mammalian population (input)
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

Having established the both the individual animal food needs for any age group (App. A., No. 57), and the total food needs of the population (App. B.,I, No. 181b.) on the basis of an estimated or entered population structure, it is necessary to discuss the metabolism of ingested food by the former before

proceeding to a discussion of population dynamics. The ingested food undergoes metabolism to estimate both catabolic and anabolic processes which involve the loss of food and body masses through respiration, gas losses, and/or losses due to undigested food which is passed through the intestines of the animal. The net result is either a gain (growth or fat deposition), or mass losses from the organism. Respiratory losses from an animal unit of a given age group is estimated as:

$$F(X_{15,m,k}, P_r) = 70\, X_{15,m,k}\, I_{249k}\, I_{250k} / 4500 \qquad \text{(App. A, No. 58)}$$

$F(X_{15,m,k}, P_r)$ = metabolic heat production losses, kgfwt/day
$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population
I_{250k} = activity factor for the kth mammalian population (input)
P_r = respiration component, kg/day

Gas losses from ingested food are estimated as:

$$F(F_{57}, P_{gs}) = I_{287k}\, F_{57}\, I_{253k} \qquad \text{(App. A., No. 59)}$$

Mass losses from urine excretion are estimated as:

$$F(F_{57}, P_{ur}) = I_{288k}\, F_{57}\, I_{253k} \qquad \text{(App. A., No. 60)}$$

Mass losses from fecal excretion are estimated as:

$$F(F_{57}, P_{fc}) = F_{57}\,(1 - I_{253k}) \qquad \text{(App. A., No. 61)}$$

Hence, the unadjusted anabolic or mass increment for a given animal unit is estimated as:

$$I_{289m,k} = F_{57} - F_{58} - F_{59} - F_{60} - F_{61} \qquad \text{(App. B.,I, No. 194)}$$

$F(F_{57}, P_{gs})$ = gas losses kg/day from ingested food
$F(F_{57}, P_{ur})$ = mass losses kg/day from ingested food via urine
$F(F_{57}, P_{fc})$ = mass losses kg/day from ingested food via urine
F_{57} = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population
I_{287k} = digestive gas mass loss fraction for the kth mammalian population (input)
I_{288k} = digestive urine mass loss fraction (input)
$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population.
I_{253k} = digestibility fraction for the kth mammalian population
P_{gs} = digestive mass losses from gas production, kg/day
P_{ur} = digestive mass losses from urine excretion, kg/day
P_{fc} = digestive mass losses from fecal excretion, kg/day

If the growth or mass increment (I_{289}) is positive, then the lean body mass growth requirement for a given animal unit on a given day is estimated from the expected lean body mass under normal growth conditions:

$$I_{290m,k} = (X_{16m,k} - X_{15m,k}) / 2.5 \qquad \text{(App. B.,I, No. 195)}$$
$$= I_{289m,k} \qquad I_{289m,k} \le I_{290um,k}$$

$I_{290m,k}$ = metabolic growth requirement of lean body mass, kg/day, for the mth age group of the the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

If the metabolic growth requirement (I_{290}) is greater than zero, then the unadjusted protein fraction of the growth increment is estimated as:

$$I_{291m,k} = 0.6944\ I_{290m,k} \qquad I_{292k} \leq I_{291um,k}$$
$$= I_{292k}$$
$$= 2.272\ I_{294m,k} \qquad I_{294m,k} > I_{293m,k}$$

(App. B.,I, No. 196)

$$I_{293m,k} = I_{290m,k} - I_{291m,k} \qquad I_{293m,k} < I_{294m,k}$$
$$= I_{294m,k}$$

(App. B.,I, No. 197)

where,

$$I_{294m,k} = 0.44\ I_{291um,k}$$

(App. B.,I, No. 198)

$I_{291m,k}$ = protein fraction the growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{290m,k}$ = metabolic growth requirement of lean body mass, kg/day, for the mth age group of the the kth mammalian population

$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{294m,k}$ = protein portion of ingested food that is not deamminated for the mth age group of the kth mammalian population, kg/day

I_{292k} = digested protein from the ingested food, kg/day, for the kth mammalian population

where,

$$I_{292k} = 0.01\ I_{289m,k}\ I_{294k}$$

(App. B.,I, No. 199)

I_{292k} = digested protein from the ingested food, kg/day, for the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

The nitrogen fraction in ingested food (I_{295}) is estimated from three food fractions consisting of range plants, range animals serving as food, and from supplementary feeds:

$$I_{295kr} = \sum_{m=1}^{I_{273}} \sum_{j=1}^{3} I_{269j,m}\ I_{296j,m,k} / I_{254k}$$

(App.,B.,I, No. 200a)

and for animal tissues using muscle nitrogen content:

$$I_{295ka} = \sum_{m=1}^{I_{273}} \sum_{j=4}^{7} \sum_{i=1}^{I_{288k}} I_{269j,m,k}\ I_{246i,k}\ X_{17,2,3} / (0.45\ X_{14i,k}\ I_{254k})$$

(App.,B.,I, No. 200b)

where,

$$I_{296j,m} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} 0.01\, I_{258i,m}\, I_{297i,m} / I_{260j,m} \qquad 0 \le j \le 3 \qquad \text{(App.,B.,I, No. 201)}$$

and,

$$I_{297i,m} = I_{258i}\, I_{298i,m} + I_{259i}\, I_{298i,m} / 2 \qquad 0 \le j \le 3 \qquad \text{(App.,B.,I, No. 202)}$$

and finally,

$$I_{298i,m} = 100\, [\, X_{13,5,1,2} + X_{13,6,1,2} + X_{13,4,1,2}\, (X_{4,1} - X_{5,1})(X_{4,1} / X_{5,1}{}^{2})] / [(X_{4,1} - X_{5,1})(X_{4,1} / X_{5,1}) + X_{5,1} + X_{6,1}\,]$$

$$0 \le j \le 3 \qquad \text{(App.,B.,I, No. 203)}$$

$I_{296j,m}$ = weighted nitrogen fraction in consumed range plants of the jth food category in the mth region
$I_{297i,m}$ = percent nitrogen in consumed food from the ith plant of the mth region
$I_{298i,m}$ = percent nitrogen in live biomass from the ith plant of the mth region
I_{294k} = weighted nitrogen fraction from all sources of food in the diet of the the kth mammalian population
I_{295kr} = weighted nitrogen fraction from range plant food consumption for the kth mammalian population
I_{295ka} = weighted nitrogen fraction from range animal tissue consumption for the kth mammalian population
I_{288k} = oldest age group, years, for the kth mammalian population (input)
$X_{17,2,3}$ = nitrogen content of muscle tissue,grams
$X_{13,5,1,2}$ = nitrogen content in live leaves, g/m^2
$X_{13,6,1,2}$ = nitrogen content in green seeds, g/m^2
$X_{13,4,1,2}$ = nitrogen content in live etible stems, g/m^2
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2
$X_{6,1}$ = live seed biomass density, g/m^2
I_{69} = total number of plants in a given region (input)
I_{253k} = digestibility fraction for the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
$I_{257i,m}$ = total biomass density, $gdwt/m^2$, for the ith plant in the mth region
I_{258i} = live plant fraction (including seeds) ingested from the ith plant type from the mth region
I_{259i} = dry plant fraction (excluding seeds) ingested from the ith plant type

I_{260j} = total biomass density, $gdwt/m^2$, in the jth preference category for range plants of the mth region
$I_{246i,k}$ = fraction of the population in the ith age group of the kth mammalian population
$I_{269j,m,k}$= food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population
I_{273} = number of regions simulated (input)

Similarly, the weighted nitrogen fraction in consumed supplementary feeds is estimated as:

$$I_{295ks} = \sum^{I_{273}} \sum^{I_{69}} I_{298i,m}\, I_{239bk} / I_{276k} \qquad \text{(App.,B.,I, No. 200c)}$$

m=1 i=1

where,

$$I_{298i,m} = [(X_{4,2}\, X_{5,1} / X_{4,1})\,(X_{13,4,2.2} + X_{13r,4,2,2} + X_{13s,4,2,2}) + X_{13,5,2,2} + X_{13r,5,2,2} + X_{13s,5,2,2} + X_{13,5,1,2} + X_{13r,5,1,2} + X_{13s,5,1,2} + (X_{4,1} - X_{5,1})\,(X_{5,1} / X_{4,1})\,(X_{13,4,1,2} + X_{13r,4,1,2} + X_{13s,4,1,2}) / X_{4,1} + X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}] \,/\, [(X_{4,2}\, X_{5,1} / X_{4,1}) + X_{5,2} + X_{5,2} + (X_{4,1} - X_{5,1})\,(X_{5,1} / X_{4,1}) + X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}] \qquad I_{240b} = 4 \qquad \text{(App., B.,I, No. 204)}$$

$$= (X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}) / X_{6,1} \qquad I_{240b} = 5$$

$$= [(X_{4,2}\, X_{5,1} / X_{4,1})\,(X_{13,4,2,2} + X_{13r,4,2,2} + X_{13s,4,2,2}) + X_{13,5,2,2} + X_{13r,5,2,2} + X_{13s,5,2,2} + X_{13,5,1,2} + X_{13r,5,1,2} + X_{13s,5,1,2} + (X_{4,1} - X_{6,1})\,(X_{5,1} / X_{4,1})\,(X_{13,4,1,2} + X_{13r,4,1,2} + X_{13s,4,1,2}) / X_{4,1}] \,/\, [(X_{4,2}\, X_{5,1} / X_{4,1}) + X_{5,2} + X_{5,2} + (X_{4,1} - X_{5,1})\,(X_{5,1} / X_{4,1})] \qquad I_{240b} = 6$$

$$= (X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}) / X_{6,1} \qquad I_{240b} = 7$$

I_{295ks} = fraction of nitrogen fraction from supplementary plant food consumption for the k<u>th</u> mammalian population
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by i<u>th</u> plant in the m<u>th</u> region for the k<u>th</u> animal population
I_{276k} = sum of the selection categories for supplementary feeds selected for consumption by the kth animal population
$X_{13,5,1,2}$ = nitrogen content in live leaves, g/m^2
$X_{13,6,1,2}$ = nitrogen content in green seeds, g/m^2
$X_{13,4,1,2}$ = nitrogen content in live etible stems, g/m^2
$X_{13r,5,1,2}$ = nitrogen content on live leaves from resuspension events, g/m^2
$X_{13s,5,1.,2}$ = nitrogen content on live leaves from saltation-creep events, g/m^2
$X_{13r,6,1,2}$ = nitrogen content on green seeds from resuspension events, g/m^2
$X_{13s6,1,2}$ = nitrogen content on green seeds from saltation-creep events, g/m^2
$X_{13r,4,1,2}$ = nitrogen content on live etible stems from resuspension events, g/m^2
$X_{13s,4,1,2}$ = nitrogen content on live etible stems from saltation-creep events, g/m^2
$X_{13,5,2,2}$ = nitrogen content in dry leaves, g/m^2
$X_{13r,2,2}$ = nitrogen content on dry leaves from resuspension events, g/m^2
$X_{13s,5,2,2}$ = nitrogen content on dry leaves from saltation-creep, g/m^2
$X_{13,6,2,2}$ = nitrogen content in dry seeds, g/m^2
$X_{13r,6,2,2}$ = nitrogen content on dry seeds from resuspension events, g/m^2
$X_{13s,6,2,2}$ = nitrogen content on dry seeds from saltation-creep events, g/m^2
$X_{13,4,2,2}$ = nitrogen content in dry etible stems, g/m^2
$X_{13r,4,2,2}$ = nitrogen content on dry etible stems from resuspension events, g/m^2
$X_{13s,4,2,2}$ = nitrogen content on dry etible stems from saltation-creep, g/m^2
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2

The weighted nitrogen fraction of all range plant and animal and supplementary feeds is thus estimated as:

$$I_{295k} = [(I_{254k} - I_{272k})(I_{295kr} + I_{295ka}) + I_{272k}\, I_{295ks}] / (I_{254k} + I_{272k}) \qquad \text{(App. B.,I, No. 200d)}$$

I_{295k} = nitrogen fraction from all sources of food in the diet of the the kth mammalian population

I_{295kr} = nitrogen fraction from range plant food consumption for the kth mammalian population

I_{295ka} = nitrogen fraction from range animal tissue consumption for the kth mammalian population

I_{295ks} = nitrogen fraction from supplementary plant food consumption for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

The metabolic growth increment adjusted for energy utilization costs is estimated by the following :

$$I_{299m,k} = I_{289m,k} - I_{291m,k} - I_{293m,k} \qquad \text{(App. B.,I, No. 205)}$$

$I_{299m,k}$ = metabolic growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

$I_{291m,k}$ = protein fraction the growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

The net change in the animal metabolic biomass is estimated as:

$$F(P_{mt}, X_{15m,k}) = 2.5\,(I_{291m,k} + I_{293m,k}) \qquad I_{287m,k} > 0 \qquad \text{(App. A., No. 62)}$$
$$= -2.13\, I_{289m,k} \qquad I_{289m,k} < 0$$

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

$I_{291m,k}$ = protein fraction growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

P_{mt} = metabolic source/sink, kg/day

The conversion of animal metabolic mass after addition of the above metabolic increment to amimal weight is estimated by the following relationship:

$$X_{14,m,k} = 1.023e\text{-}4\,(1000\, X_{15,m,k})^{1.33} \qquad \text{(App. B.,V, No. 10)}$$

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

An illustration of the growth and metabolism processes described abo ıre presented in Fig. 15. for clarification purposes.

6.2.3. Population Dynamics

The population dynamics currently developed in this model reflects an equibrium between natalities and mortalities in the absence of predation or starvation processes operating in the model for both animals and humans. The effects of contaminants on the dynamics of a population are currently in the planning stages. The user of the model has the choice of initially entering both natality and mortality rates independent of predation and starvation effects for a given population which will then operate throughout the simulation. The discussion to follow will discuss the default natality and mortality developed in the model as well as the effects of predation and starvation which contribute to the mortality. The previous sub-section has already defined the fraction of the population composed of a specific age group for a given mammalian population (see App. B.,I, No. 190-193). The total number of individuals in any age group of a mammalian population is thus:

$$I_{300am,k} = I_{280ak}\, I_{279am,k} = I_{280ak}\,(I_{301k} - I_{302k}) \qquad m=1 \qquad \text{(App. B.,I, No. 206a)}$$

$$I_{300bm,k} = I_{280bk}\, I_{279bm,k} = I_{280bk}\,(I_{301k} - I_{302k}) \qquad m=1 \qquad \text{(App. B.,I, No. 206b)}$$

where,

$$I_{301k} = [\,(I_{283k} - I_{303k}) / I_{283k}]^{1.5} \qquad I_{285k}=2 \qquad \text{(App. B.,I, No. 207)}$$
$$= \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - I_{303k})\,] \qquad I_{285k}=3$$
$$= [\,ln\, I_{303k} + I_{303k} / I_{283k} + 1/ I_{283k}\,]/\,[ln\, I_{283k} - 1 + 1/I_{283k}\,] \qquad I_{285k}=4$$

and,

$$I_{302m,k} = [\,(I_{283k} - 365) / I_{283k}]^{1.5} \qquad I_{285k}=2 \qquad \text{(App. B.,I, No. 208)}$$
$$= \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - 365)\,] \qquad I_{285k}=3$$
$$= [\,ln\, 365 + 365 / I_{283k} + 1/ I_{283k}\,]/\,[ln\, I_{283k} - 1 + 1/I_{283k}\,] \qquad I_{285k}=4$$

$I_{300am,k}$ = number of males in the m<u>th</u> age group of the k<u>th</u> mammalian population
$I_{300bm,k}$ = number of females in the m<u>th</u> age group of the k<u>th</u> mammalian population
I_{280ak} = total male population of the k<u>th</u> mammalian population
I_{280bk} = total female population of the k<u>th</u> mammalian population
I_{283k} = maximum age, days, of oldest age group in the k<u>th</u> mammalian population
I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
$I_{279am,k}$ = fraction of male population in the the m<u>th</u> age group of the the k<u>th</u> mammalian population (initial input or default)
$I_{279bm,k}$ = fraction of female population in the the m<u>th</u> age group of the the k<u>th</u> mammalian population (initial input or default)
I_{301k} = upper age correction factor for 1 year-olds
I_{302k} = lower age correction factor for 1 year-olds

The corrected age of a one year-old animal (I_{303k}) is estmated as:

$$I_{303k} = I_{284k} - I_{304k} - I_{305k} \qquad \text{(App. B.,I, No. 209)}$$

where,

$$I_{304k,1} = I_{306k,1} + 14 + I_{307k} \qquad \text{(App. B.,I, No. 210)}$$

but if more than one litter is produced per year:

$$I_{306k,2} = I_{306k,1} + 14 + I_{307k} + 9.125\, I_{248k} \qquad I_{308k} = 2 \qquad \text{(App. B.,I, No. 211)}$$
$$= 0 \qquad I_{308k} = 1$$

and,

$$I_{304k,2} = I_{306k,2} + 14 + I_{307k} \qquad I_{308k} = 2 \qquad \text{(App. B.,I, No. 211)}$$

where,

$$I_{307k} = 0.15\,(365\, I_{248k})^{0.8} \qquad \text{(App. B.,I, No. 212)}$$

and also,

$$I_{305k} = (14 + 0.025(365\, I_{248k}) + I_{307k})\,/\,2 \qquad I_{308k} = 2 \qquad \text{(App. B.,I, No. 213)}$$
$$= 0 \qquad I_{308k} = 1$$

I_{303k} = corrected age of a one year-old, days, for the kth mammalian population
$I_{284m,k}$ = maximum age,years, of the mth age group of the kth mammalian population
$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)
$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)
I_{305k} = age correction, days, for newborns of the kth mammalian population
$I_{306k,1}$ = initiation of gestation period, first litter, for the kth mammalain population (1= day 92, 2=day 270) (input)
$I_{306k,2}$ = initiation of gestation period ,second litter, for the kth mammalian population
I_{307k} = estimated gestation period, days, for the kth mammalian population
I_{308k} = number of gestation periods per year (input)
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

Thus, the age of the newborns is synchronized so that they can be treated independently during the year for contaminant uptake growth, and mortality purposes. At present the model is limited to two gestation periods in a given year, and is limiting only for mammalian populations where very short gestation periods relative to one year are being simulated.

6.2.3.1 Natural Mortality

The natural mortality rate due to factors other than predation or starvation is modeled after the selected population structure if a default structure has been selected by the user assuming that the same mortality rate applies to males as estimated for females below:

$$I_{309m,k} = (0.5/\,I_{283k})\,(1 - I_{278m,k}\,/I_{283k})^{-0.5/\,I310k} \qquad I_{285k} = 2;m>1 \qquad \text{(App. B.,I, No. 214)}$$

$$= \exp[\,-(10\,\ln 2\,/\,I_{283k})\,(I_{283k} - I_{278m,k})\,]\,/I_{310k} \qquad I_{285k} = 3;m>1$$

$$= I_{278m,k}^{\;-2/\,I310k} \qquad I_{285k} = 4;m>1$$

$I_{309m,k}$ = natural mortality rate, day^{-1}, for the mth age group of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
$I_{278m,k}$ = age, days, of the mth age group of the kth mammalian population
I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population

where,

$$I_{310k} = I_{280bk} / I_{311k}$$ (App. B.,I, No. 215)

I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{311k} = original number of females in the kth mammalian population at the start of the simulation (input)

The mortality rate of one year-olds is divided into as many as three groupings: carried over one year-olds from the previous year, those born after the first gestation period, and those born after the second gestation period in a given year if included in the simulation:

$$I_{309m,k} = (0.5/ I_{283k}) (1- I_{312k1} /I_{283k})^{-0.5/ I310k}$$
$$= (0.5/ I_{283k}) (1- I_{312k2} /I_{283k})^{-0.5/ I310k}$$
$$= (0.5/ I_{283k}) (1- I_{312k3} /I_{283k})^{-0.5/ I310k}$$
I_{285k}= 2;m=1 (App. B.,I, No. 214a)

$$I_{309m,k} = \exp[-(10\ ln2 / I_{283k}) (I_{283k} - I_{312k1})] /I_{310k}$$
$$= \exp[-(10\ ln2 / I_{283k}) (I_{283k} - I_{312k2})] /I_{310k}$$
$$= \exp[-(10\ ln2 / I_{283k}) (I_{283k} - I_{312k3})] /I_{310k}$$
I_{285k}= 3;m=1 (App. B., No. 214b)

$$I_{309m,k} = I_{312k1}^{-2/ I310k}$$
$$= I_{312k2}^{-2/ I310k}$$
$$= I_{312k3}^{-2/ I310k}$$
I_{285k}= 4;m=1 (App. B., No. 214c)

$I_{309m,k}$ = natural mortality rate, day^{-1}, for the 1st age group of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population
I_{312k1} = age, days, of newborns carried over from the previous year for the kth mammalian population
I_{312k2} = age, days, of newborns from the first gestation period during a given year for the kth mammalian population
I_{312k3} = age, days, of newborns from the second gestation period during a given year for the kth mammalian population

where,

$$I_{312k1} = I_{278m,k}$$
$$= 1$$
(App. B.,I, No. 216a) $I_{278m,k} < 1.0$

$$I_{312k2} = \sum_{i=1}^{n} 1_i$$
$$= 1$$
(App. B.,I, No. 216b) $I_{304k,1} = I_{313k1}$

$$I_{312k3} = \sum_{i=1}^{n} 1_i$$
$$= 1$$
(App. B.,I, No. 216c) $I_{304k,2} = I_{313k2}$

where,

$$I_{313k1} = \sum^{n} 1_i$$ (App. B.,I, No. 217a)

$$= I_5 \qquad I_{312k1} > I_{304k,1} + 9.125\, I_{248k}$$

$$I_{313k2} = \sum_{i=1}^{n} 1_i \qquad \text{(App. B.,I, No. 217b)}$$

$$= I_5 \qquad I_{312k2} > I_{304k,2} + 9.125\, I_{248k}$$

I_{312k1} = age, days, of newborns carried over from the previous year for the kth mammalian population

I_{312k2} = age, days, of newborns from the first gestation period during a given year for the kth mammalian population

I_{312k3} = age, days, of newborns from the second gestation period during a given year for the kth mammalian population

I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population

I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter) for the kth mammalian population

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter) for the kth mammalian population

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

I_5 = julian day number (1 - 365)

6.2.3.2. Starvation

The mortality rate due to starvation effects is based on the ratio of the simulated weight of the animal in question to the expected weight:

$$I_{314m,k} = 0.1\, I_{315m,k} \exp(-I_{316m,k}) \qquad \text{(App. B.,I, No. 218)}$$
$$= 0 \qquad X_{14} \geq 0.7\, X_{14e}$$

where,

$$I_{315m,k} = (X_{14e} - X_{14}) / (0.250663\, X_{14e}) \qquad \text{(App. B.,I, No. 219)}$$

and,

$$I_{316m,k} = [(X_{14e} - X_{14}) / (0.1\, X_{14e})] / 8 \qquad \text{(App. B.,I, No. 220)}$$

$X_{314m,k}$ = mortalities due to starvation, day^{-1}, for the mth age group of the kth mammalian population

$X_{315m,k}$ = starvation factor, day^{-1}

$X_{316m,k}$ = starvation coefficient

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

This estimate of mortality due to negative deviations from the expected weight of a mammal in a given age group is set to reach about 80% day^{-1} mortality when the simulated weight is about 70% of the expected weight under conditions of adequate food and environmental conditions, but is not estimated for simulated weights above 70% of the expected weights just described.

6.2.3.3. Prey Mortality

The estimate of prey mortalities for animals is based on the total biomass lost from each age group in a given population under the assumption that all age groups of both sexes are equally vulnerable:

$$I_{317m,k,l} = 1000\, F_{57}\, (I_{269j,i,k} / I_{254k})\, (I_{300am,k} + I_{300bm,k}) / 1000\, I_{318k} \qquad j > 3; I_{241k} > 0 \qquad \text{(App. B.,I, No. 221)}$$

where,

$$I_{318k} = 0.33 \qquad I_{248k} > 0 \qquad \text{(App. B.,I, No. 222)}$$
$$= 1 \qquad I_{248k} = 0$$

$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region
$I_{269j,i,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the ith region combined for the kth mammalian population
$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population
$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population
F_{57} = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
I_{318k} = weight fraction of prey utilized for food
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

The food consumption by each age group of the mammalian predator is then used to reduce the population of all age groups of the prey on a proportional basis:

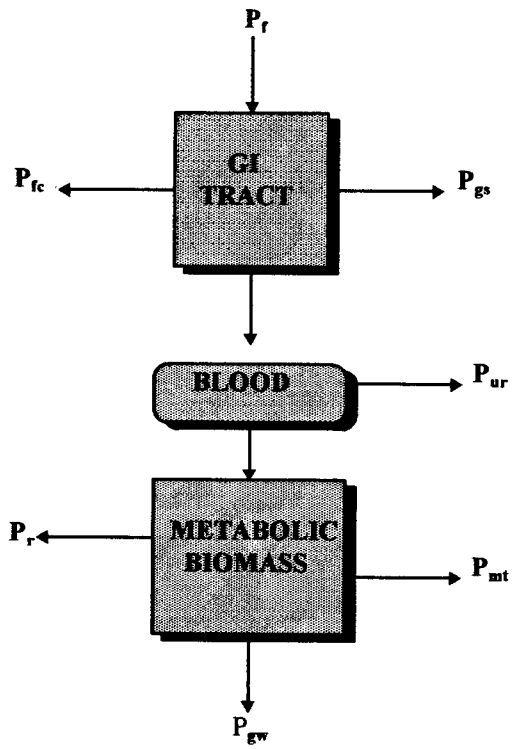
$$I_{319mp,kp} = I_{317m,k,l} / [\sum_{kp=1}^{I274} \sum_{mp=1}^{I248} (I_{300amp,kp} + I_{300bmp,kp})\, X_{14mp,kp}]\, I_{246mp,kp} \qquad I_{241k,l} > 0 \qquad \text{(App. B.,I, No. 223)}$$

$I_{319mp,kp}$ = mortality rate, day-1, of the mpth age group of the kpth prey animal due to the mth age group of kth mammalian predator population
$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region
$X_{14,mp,kp}$ = animal biomass compartment, kgfwt, for an animal in the mpth age group of the kpth animal prey population
$I_{300amp,kp}$ = number of males in the mpth age group of the kpth prey mammalian population
$I_{300bmp,kp}$ = number of females in the mpth age group of the kpth prey mammalian population
$I_{246mp,kp}$ = fraction of the population in the mpth age group of the kpth mammalian prey population comsumed by the kth mammalian predator population
I_{274k} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannabalism is being modeled) in a given region
I_{248kp} = lifespan, years (largest whole number), of the an animal in the kpth mammalian prey population (input)
I_{241kp} = initial diet fraction of the kpth animal prey type in the food of a specific animal predator type (input)

where,

$$I_{246mp,kp} = (I_{246mp,kp}^{2} / I_{i246mp,kp}) \qquad (I_{246mp,kp} / I_{i246mp,kp}) < 1 \qquad \text{(App. B.,I, No. 224)}$$
$$= I_{246mp,kp} \qquad (I_{246mp,kp} / I_{i246mp,kp}) \geq 1$$

$I_{246mp,kp}$ = fraction of the population in the mpth age group of the kpth mammalian prey population available for consumption by the kth mammalian predator population



Legend (see text)
P_f = transport of food biomass, kgdwt/day, from plant and/or other animal compartments
P_{gs} = digestive mass losses from gas production, kg/day
P_{ur} = digestive mass losses from urine excretion, kg/day
P_{fc} = digestive mass losses from fecal excretion, kg/day
P_r = respiration component, kg/day
P_{mt} = metabolic source/sink, kg/day
P_{gw} = net growth or mass loss, kg/day

Fig. 15

Mammalian Metabolic and Growth Strategies Used in BIOTRAN.2

$I_{i246mp,kp}$ = initial fraction of the population in the mpth age group of the kpth mammalian prey population available for consumption by the kth mammalian predator population (initial conditions derived from input)

Thus, the amount of prey biomass available to a given predator is limited by the divergence of the prey population below the expected on the assumption that increased scarcity leads to alternative food choices for the given predator population. Finally, the total mortality rate of the prey animal group in question is obtained on a daily basis by summing the mortalities over all of the predator age groups.

6.2.3.4. Invertebrate Prey Mortality

The mortality of invertebrate animals is obtained by adjustment of the biomass densitites of these organisms due to predation (there is only 1 age group modeled):

$$I_{319kp} = (I_{317m,k,l} / X_{14kp})\, I_{246kp} \qquad I_{241k,l} > 0 \qquad \text{(App. B.,I, No. 225)}$$

where,

$$I_{246kp} = 1 \qquad I_{320kp} \geq 1 \qquad \text{(App. B.,I, No. 226)}$$
$$I_{246kp} = I_{320kp} \qquad I_{320kp} < 1$$

I_{319kp} = mortality rate, day-1, of the kpth prey invertebrate animal due to the mth age group of kth mammalian predator population

$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth invertebrate prey by the mth age group of the kth mammalian population used as food from a given region

X_{14kp} = animal biomass compartment, kgdwt, for an invertebrate animal of the kpth animal prey population

I_{246kp} = fraction of the population of the kpth invertebrate prey population available for consumption by the kth mammalian predator population

I_{320kp} = combined biomass density, gdwt /m^2, of all of the kpth immature and mature invertebrate animal

6.2.3.5. Natality

The birth or natality rates of mammalian populations is based on the availability and differential natality rates functions obtained from data on humans. The model assumes that females with ages below 25% and above 75% of the life-span of the respective populations are not contributors to population regeneration of a given mammalian population. The females that are within the regeneration group are further divided into eight reproductive groupings have different natality rates:

$$\begin{aligned}
I_{321k,l,n} &= 0.003\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=1;n=1 \qquad \text{(App. B.,I, No. 227)}\\
&= 0.003\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=1;n=2\\
&= 0.138\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=2;n=1\\
&= 0.138\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=2;n=2\\
&= 0.296\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=3;n=1\\
&= 0.296\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=3;n=2\\
&= 0.295\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=4;n=1\\
&= 0.295\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=4;n=2\\
&= 0.207\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=5;n=1\\
&= 0.207\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=5;n=2\\
&= 0.050\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=6;n=1\\
&= 0.050\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=6;n=2\\
&= 0.010\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=7;n=1
\end{aligned}$$

$$= 0.010\,(1 - I_{282m,k})\, I_{280bk} \,/\, (I_{280bk}\, I_{322k,l}) \qquad l=7;n=2$$
$$= 0.001\,(1 - I_{282m,k})\, I_{280ak} \,/\, (I_{280bk}\, I_{322k,l}) \qquad l=8;n=1$$
$$= 0.001\,(1 - I_{282m,k})\, I_{280bk} \,/\, (I_{280bk}\, I_{322k,l}) \qquad l=8;n=2$$

where,

$$I_{322k,l} = (I_{279bmr,k} + I_{279bmr-1,k}) / 2 \qquad I_{285k}=1 \text{ (App. B.,I, No. 228)}$$

$$= [(I_{283k} - I_{323k,l}) / I_{283k}]^{1.5} - \{[I_{283k} - (I_{323k,l} + I_{324k})] / I_{283k}\}^{1.5} \qquad I_{285k}=2$$

$$= \exp[-(10\, ln2 / I_{283k})(I_{283k} - I_{323k,l})] - \exp\{-(10\, ln2 / I_{283k})[I_{283k} - (I_{323k,l} + I_{324k})]\} \qquad I_{285k}=3$$

$$= [\, ln\, I_{323k,l} - I_{323k} / I_{283k} - ln(I_{323k,l} + I_{324k})] + (I_{323k,l} + I_{324k})/I_{283k} \qquad I_{285k}=4$$

$I_{322k,l}$ = fraction of the reproductive females in the lth reproductive age grouping for the kth mammalian population

$I_{321k,l,n}$ = reproductive rate, indivivdual-1, for the lth female reproductive grouping for males (n=1), and females (n=2)

I_{280ak} = total male population of the kth mammalian population

I_{280bk} = total female population of the kth mammalian population

I_{283k} = maximum age, days, of oldest age group in the kth mammalian population

I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)

$I_{279bmr,k}$ = fraction of female population in the the mrth reproductive age group of the the kth mammalian population (initial input or default)

$I_{282m,k}$ = population fraction parameter (upper integral) for the mth age group of the kth mammalian population

$I_{282m-1,k}$ = population fraction parameter (lower integral) for the mth age group of the kth mammalian population

$I_{323k,l}$ = age, days, of a reproductive female in the lth grouping (l= 1-8) , initial age set at 15% of the life-span, days

I_{324k} = age increment, days, expressed as 1/8 th of the age at 55% of the life span

In the determination of the reproductive age grouping for any reproductive female, the summation increment(I_{324k}) is added successively to the initial age value at 15% of the life-span ($I_{323k,l}$). whenever the female age exceeds the initial age plus an age increment, the reproductive grouping index (*l*) is incremented to the next higher value, and an additional age increment is added to the initial age...etc., until all eight groupings have been estimated. The intent of grouping females for reproductive purposes is modeled to account for differential birth rates at various stages of female development. The grouping selected for this model is based on human reproductive rates as a function of age under the assumption that this type of information can be scaled to other mammals.

The total number of animals of either sex born after a given gestation period is thus estimated as:

$$I_{325k,n} = \sum_{l=1}^{8} I_{321k,l,n}\, I_{322k,l}\, I_{311k}\, I_{325i,k}\, I_{304k,l} \qquad I_{304k,1} = I_{313k,1} \qquad \text{(App. B.,I, No. 229)}$$

$$= \sum_{l=1}^{8} I_{321k,l,n}\, I_{322k,l}\, I_{311k}\, I_{325i,k}\, I_{304k,2} \qquad I_{304k,2} = I_{313k,2}$$

$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population

$I_{321k,l,n}$ = reproductive rate, indivivdual-1, for the lth female reproductive grouping for males (n=1), and females (n=2)

$I_{322k,l}$ = fraction of the reproductive females in the lth reproductive age grouping for the kth mammalian population

I_{311k} = original number of females in the kth mammalian population at the start of the simulation (input)

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

$I_{325i,k}$= fraction of new-born in the ith gestation period of a given year for the kth mammalian population (equals 1 if only one gestation period specified, input)

The unchanging number of births on a gestation by gestation period sequence is included to provide resiliency to population changes due to predation and/or starvation mortality losses by allowing population numbers in each age category to regenerate under these conditions. The model is currently poised to include the effects of these events in addition to the effects of contaminants by either modifying the established differential birth rates (I_{321}), or the number of reproductive females in any or all age groupings.

The total number of individuals in a given age group of a mammalian population on a day by day basis is thus estimated as:

$$
\begin{aligned}
I_{300am,k} &= I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} && m>1 \\
&= I_{300am-1,k} && m>1;\ I_5 = 365 \\
&= I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n} && m=1;\ I_{304k,1} = I_{313k,1} \\
&= I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n} && m=1;\ I_{304k,2} = I_{313k,2}
\end{aligned}
\qquad \text{(App. B,I, No. 230a)}
$$

$$
\begin{aligned}
I_{300bm,k} &= I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} && m>1 \\
&= I_{300bm-1,k} && m>1;\ I_5 = 365 \\
&= I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n} && m=1;\ I_{304k,1} = I_{313k,1} \\
&= I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n} && m=1;\ I_{304k,2} = I_{313k,2}
\end{aligned}
\qquad \text{(App. B,I, No. 230b)}
$$

$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population

$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population

$I_{309m,k}$ = natural mortality rate, day^{-1}, for the mth age group of the kth mammalian population of either sex

$I_{314m,k}$ = mortalities due to starvation, day^{-1}, for the mth age group of the kth mammalian population

$I_{319m,k}$ = mortality rate, day^{-1}, of the mth age group of the kth animal type due to predation given predator

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population

I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population

I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population

I_5 = julian day number (1 - 365)

As shown, on Julian day 365, all age groups are transferred to the next higher age group, whereas the new 1 year old age group is estimated from up to 3 separate age sub-groups mentioned previously consisting

of carry-overs, new-borns from the first litter, and new-borns from the second litter if one has been specified. The new 1 year-old age group on Julian day 365 is estimated as:

$I_{300am,k} = (I_{326ak,1} + I_{326ak,2} + I_{326ak,3})$ $\quad m=1;\ I_5=365$ $\quad$ (App. B.,I, No. 231a)

$I_{300bm,k} = (I_{326bk,1} + I_{326bk,2} + I_{326bk,3})$ $\quad m=1;\ I_5=365$ $\quad$ (App. B.,I, No. 231b)

where,

$I_{326ak,n} = I_{325k,1}$ $\quad I_{327k}=1; I_{304k,1} = I_{313k,1}; I_{328k}=0; n=1$ $\quad$ (App. B.,I, No.232a)

$= I_{326ak,n-1} + I_{325k,1}$ $\quad I_{327k}=2; I_{304k,2} = I_{313k,2}; I_{328k}=0; n=2$

$= I_{326ak,n} + I_{326ak,n+1}$ $\quad I_{304k,1} = I_{313k,n}; I_{328k} > 1; n \leq 2$

$I_{326bk.n} = I_{325k,2}$ $\quad I_{327k}=1; I_{304k,1} = I_{313k,1}; I_{328k}=0; n=1$ $\quad$ (App. B.,I, No.232b)

$= I_{326bk,n-1} + I_{325k,2}$ $\quad I_{327k}=2; I_{304k,2} = I_{313k,2}; I_{328k}>0; n=2$

$= I_{326bk,n} + I_{326bk,n+1}$ $\quad I_{304k,2} = I_{313k,n}; I_{328k} > 1; n \leq 2$

$I_{326ak,n+1} = I_{325k,1}$ $\quad I_{304k,1} = I_{313k,1}; I_{328k} > 1; n \leq 2$ $\quad$ (App. B.,I, No.233a)

$I_{326bk,n+1} = I_{325k,2}$ $\quad I_{304k,1} = I_{313k,1}; I_{328k} > 1; n \leq 2$ $\quad$ (App. B.,I, No.233b)

$I_{326ak,n}$ = number of male animals in the nth sub-group of the first age group for the kth mammalian population

$I_{326bk,n}$ = number of female animals in the nth sub-group of the first age group for the kth mammalian population

I_{327k} = index number (1,2) designating the first or second gestation period if simulated for the kth mammalian population

$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population

I_{328k} = index number (0,>0) designating default beef and dairy cow population dynamics apart from other mammalian populations for the kth mammalian population

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population

I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population

The above strategy separate tracking of the three one-year-old sub-groups until they are combined at the end of each calendrical year into the first sub-group in preparation for the next year time interval. Since beef and dairy cows may mbe treated as artificially maintained populations, the model allows for exact replacement of losses during any given year, however, the user has the option of specifying a natural population of these organisms if desired. A diagram of the current and proposed population dynamics processes are shown in the Fig. 16 for a mammalian population.

7.0. CONTAMINANT FLOW

7.1. plants

The transport of contaminants by plants to animal receptors is simulated to occur by both internal and external processes taking into account internal root uptake ($\mathbf{P_{sr}}$), deposition,resuspension, and wash-off of resuspendable particulates ($\mathbf{P_{dp}}$,$\mathbf{P_{rs}}$) from global or local sources; and deposition and wash-off of saltation-creep and rainsplash events of erodable soil particulates ($\mathbf{P_{sc}}$). An illustration of these processes presented in Fig. 17. shows that in addition to soil surface deposition, resuspension, and wash-off events, the model simulates these processes on both standing and lying plant surfaces as well. The flow of contaminants in BIOTRAN.2 has been firmly coupled to both moisture and biomass flows as discussed in

an earlier section because so little is known of the behavior of many contaminants in ecosystems. Consequently, it was assumed advantageous to vector contaminant flows with flows which could be more easily modeled and understood provided the coupling was reasonable and accurate. In addition to modeling of concomitant flow of contaminants with moisture and biomass, other processes involving the cycling of contaminants between the soil surface, plant surfaces, and the atmosphere have been incorporated into the model. The flow of contaminants into the modeled to occur via atmospheric deposition of contaminants in particulate form, or from vapors [44,45], however, contaminant air concentrations from chronic and/or acute release events are provided by user input which may have been estimated from the use of available plume dispersion codes such as CAP-88 [46] or other available estimation methods external to BIOTRAN.2. For chronic input:

$F(P_{pe}, X_{18r,m}) = I_{329m}$ $\quad O_6 > 0;\ O_{7m} > 1;\ O_{8n} > 1;\ I_{331m} \le I_5 \le I_{332m}$ $\quad I_{333m} \le I_{29} \le I_{334m}$ (App. A., No. 63a)

$F(P_{ge}, X_{18g,m}) = I_{330m}$ $\quad O_6 > 0;\ O_{7m} > 1;\ O_{9m} > 1;\ I_{331m} \le I_5 \le I_{332m}$ $\quad I_{333m} \le I_5 \le I_{334m}$ (App. A., No. 64a)

O_6 = chronic/acute access gate (0,1) (input)
O_{7m} = contaminant gate (0,1) for the mth contaminant (input)
O_{8m} = chronic access gate(0,1) for the mth particulate air contaminant (input)
O_{9m} = chronic access gate(0,1) for the mth vaporous air contaminant (input)
I_{329m} = air concentration, m^{-3}, for the mth particulate contaminant for a given region
I_{330m} = air concentration, m^{-3}, for the mth vaporous contaminant for a given region
I_{331m} = julian day number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region
I_{332m} = julian day number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region
I_{333m} = year number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region
I_{334m} = year number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region
$F(P_{pe}, X_{18r,m})$ = transport of external resuspenable atmospheric contamination of the mth contaminant to a given to a given region, m^{-3}, as particulates
$F(P_{ge}, X_{18g,m})$ = transport of external atmospheric contamination of the mth contaminant to a given to a given region, m^{-3}, as a vapor
P_{pe} = external source of contamination for a given contaminant, m^{-3}, as particulates
P_{ge} = external source of contamination for a given contaminant, m^{-3}, as a vapor
$X_{18r,m}$ = regional particulate concentration, m^{-3}, for the mth contaminant
$X_{18g,m}$ = regional vapor concentration, m^{-3}, for the mth contaminant
I_5 = julian day number (1 - 365)
I_{29} = julian year number of specific year of simulation

where,

$$
\begin{aligned}
I_{329m} &= I_{335m} \exp(0.693/I_{336m})(I_5 - I_{331m}) && I_{340}=1; I_{336m} > 0 && \text{(App. B.,I, No. 234)}\\
&= I_{335m} + I_{336m}(I_5 - I_{331m}) && && I_{340}=2\\
&= I_{337m}(1 - \exp(0.693/I_{336m})(I_5 - I_{331m}) && && I_{340}=3\\
&= 0.5(I_{337m} - I_{335m})\{\sin[2.0\,\pi\,(I_{338} - I_{339})/(I_{332m} - I_{331m})] + 1\} + I_{335} && && I_{340}=4\\
&= I_{335} && && I_{340}=5
\end{aligned}
$$

$$I_{330m} = I_{329m} \qquad \text{(App. B., I, No. 235)}$$

I_{329m} = air concentration, m^{-3}, for the mth particulate contaminant for a given region

I_{330m} = air concentration, m^{-3}, for the mth vaporous contaminant for a given region

I_{335m} = initial air concentration , m^{-3}, for the mth contaminant in either particulate or vapor form (input); **note:** when $I_{340}>3$, this variable equals the minimum air concentration instead of the initial air concentration

I_{336m} = decay (-) or growth(+) half-time,days, or linear coefficient , day^{-1},of the chronic function for the mth contaminant in a given region (input)

I_{331m} = julian day number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region (input)

I_{332m} = julian day number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region (input)

I_{337m} = maximum air concentration , m^{-3}, for the mth contaminant in either particulate or vapor form (input)

I_{338} = current Julian day

I_{339} = lag(-) or lead(+) constant, degrees, for determining Julian day at which maximum air concentration (amplitude) will occur in the sine wave function (input)

I_{340} = chronic function selector (1=exponential, 2=linear, 3=ingrowth, 4= sine-wave, 5= constant (input)

In addition to the choice of air concentration function, a second option allows either enhancement or dampening of the selected function:

$$I_{329m} = I_{329m} \exp[(0.693 / I_{341m}) (I_{29} - I_{333m})] \qquad I_{342} = 1 \qquad \text{(App. B.,I, No. 236)}$$

$$= I_{329m} I_{341m}(I_{29} - I_{333m}) \qquad I_{342} = 2$$

$$= I_{329m} (1 - \exp[(0.693/I_{341m}) (I_{29} - I_{333m})] \qquad I_{342} = 3$$

$$= I_{329m} I_{341m} \qquad I_{342} = 4$$

$$I_{330m} = I_{329m} \qquad \text{(App. B.,I, No. 237)}$$

I_{333m} = year number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region

I_{341m} = decay (-) or growth(+) half-time,years, or linear coefficient , $year^{-1}$,of the chronic function damper/ enhancer for the mth contaminant in a given region (input)

I_{342} = chronic function damper/enhancer selector (1=exponential, 2=linear, 3=ingrowth, 4= constant) (input)

An acute event(s) can be created by the user using input specification of year(s) and julian day(s) over which this event(s) will occur :

$$F (P_{pe}, X_{18r,m}) = I_{329m} \qquad O_6 > 0;\ O_{7m} > 1;\ I_{343m} \le I_5 \le I_{343m} + I_{344m} \qquad I_{345m} = I_{29};\ I_{346m,n} \ge 0 \qquad \text{(App. A., No. 63b)}$$

$$F (P_{ge}, X_{18g,m}) = I_{330m} \qquad O_6 > 0;\ O_{7m} > 1;\ I_{343m} \le I_5 \le I_{343m} + I_{344m} \qquad I_{345m} = I_{29};\ I_{346m,n} \ge 0 \qquad \text{(App. A., No. 64b)}$$

O_6 = chronic/acute access gate (0,1) (input)

O_{7m} = contaminant gate (0,1) for the mth contaminant (input)

I_{329m} = air concentration, m^{-3}, for the mth particulate contaminant for a given region

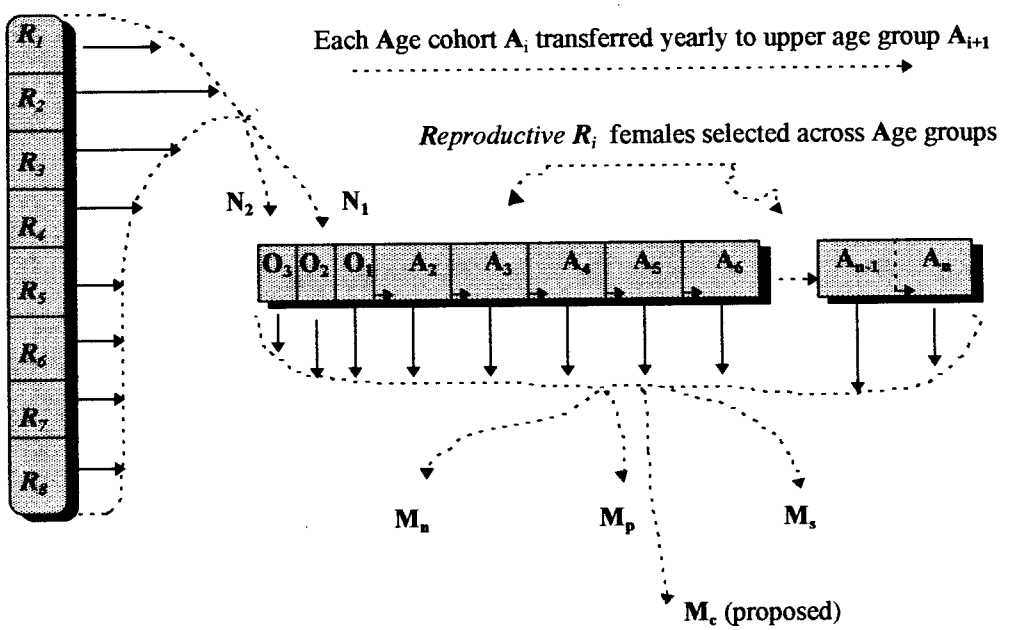
I_{330m} = air concentration, m^{-3}, for the mth vaporous contaminant for a given region

$I_{346m,n}$ (O_{10}) = number of acute events in simulation for the mth contaminant in the nth region (input)

I_{343m} = julian day number for the initiation of an acute event for entering daily air concentration(s) of the mth contaminant in a given region

I_{344m} = duration of an acute event,days, for entering daily air concentration(s) of the mth contaminant in a given region

I_{345m} = year number(s) for initiating an acute event by the user

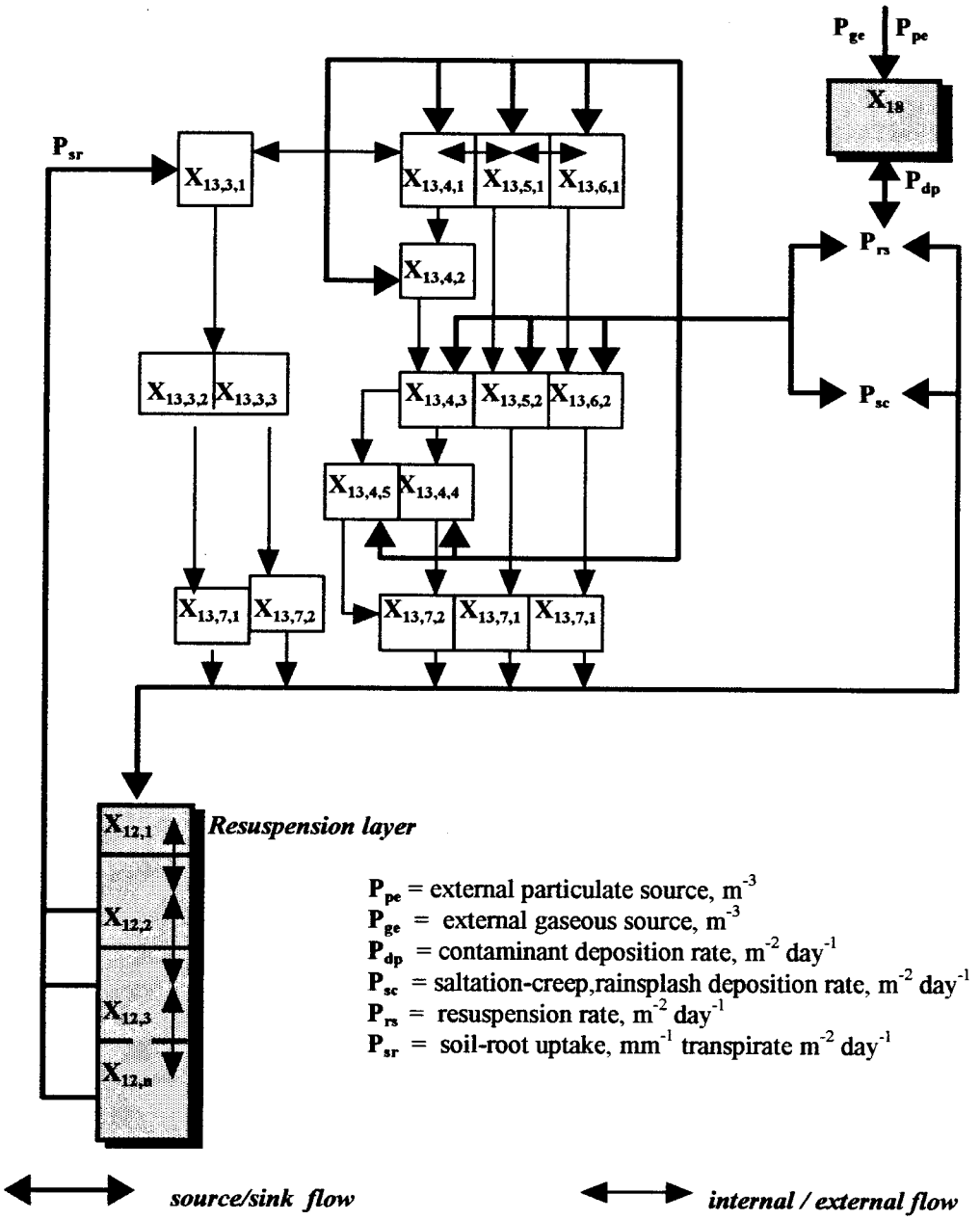


Legend

R_i = number of females in reproductive age groups selected from reproductive females in population (i= 1-8)
O_i = number of males/females in sub-groups among 1 year-olds, 1= carryover, 2= 1st litter, 3= 2nd litter
A_i = number of males/females in the ith age group (i= 2-n; n= life-span)
N_i = number of live births (natality) in the ith litter , 1= first litter, 2=second litter
M_n = natural mortality rate, day^{-1} , from causes other than predation,starvation, or contaminant effects
M_p = natural mortality rate, day^{-1} , from predation
M_s = natural mortality rate, day^{-1} , from starvation
M_c = mortality rate, day^{-1} , from effects of contaminants (proposed)

Fig. 16

Mammalian Population Dynamics Processes Used In BIOTRAN.2



Legend

contaminant concentration, m^{-2}, in/on biomass compartments

$X_{13,3,1}$ = live roots
$X_{13,3,2}$ = dry root twigs
$X_{13,3,3}$ = dry root bole-branches
$X_{13,4,1}$ =live above standing
$X_{13,4,2}$ = dry standing
$X_{13,4,3}$ = dry lying
$X_{13,4,4}$ = dry lying branch-twigs
$X_{13,4,5}$ = dry lying boles
$X_{13,5,1}$ = live leaves
$X_{13,5,2}$ = dry leaves
$X_{13,6,1}$ = seeds
$X_{13,6,2}$ = dry seeds
$X_{13,7,1}$ = root twig , leaf, and seed humus
$X_{13,7,2}$ = branch-twig ,root bole-stem ,and bole humus
$X_{12,n}$ = nth soil layer, contaminant concentration, $gdwt^{-1}$
X_{18} = regional air concentration, m^{-3}

Fig. 17

Air, Soil, and Plant Interaction Strategies In BIOTRAN.2 For Contaminant Transport

$F(P_{pe}, X_{18r,m})$ = transport of external atmospheric contamination of the mth contaminant to a given to a given region, m^{-3}, as particulates
$F(P_{ge}, X_{18g,m})$ = transport of external atmospheric contamination of the mth contaminant to a given region, m^{-3}, as a vapor
P_{pe} = external source of resuspendable contamination for a given contaminant, m^{-3}, as particulates
P_{ge} = external source of contamination for a given contaminant, m^{-3}, as a vapor
$X_{18r,m}$ = regional particulate concentration, m^{-3}, for the mth contaminant
$X_{18g,m}$ = regional vapor concentration, m^{-3}, for the mth contaminant
I_5 = julian day number (1 - 365)
I_{29} = julian year number of specific year of simulation

Thus for each year when an acute event is specified, the user specifies the duration and the initial julian date for this event. More information on the use of the acute event scenario option is presented in Appendix G. of this document. If both a chronic and an acute event overlap, then both are are summed to obtain the air concentration of the contaminant obtained from both sources.

7.1.1. Resuspension: Air, Soils and Plants

Interaction of the external sources of resuspendable particulate air contaminants ($\mathbf{P_{pe}}$) with local sources of contaminant present on the ground and plant surfaces further modifies the overall air concentration of these contaminants on a regional basis. The deposition of contaminants present as resuspendable particles is estimated as:

$$F(X_{18r}, X_{r12,1}) = 0.67\, I_{347}\, X_{18r}\, (1 - I_{348}) / (I_{37j}\, 1400) \qquad \text{(App. A., No. 65)}$$

$$F(X_{18r}, X_{r13,5,1}) = 0.67\, I_{347}\, X_{18r}\, I_{19} \qquad \text{(App. A., No. 66a)}$$

$$F(X_{18r}, X_{r13,5,2}) = 0.9\, I_{347}\, X_{18r}\, I_{348} \qquad \text{(App. A., No. 66b)}$$

$$F(X_{18r}, X_{r13,6,1}) = 0.67\, I_{347}\, X_{18r}\, I_{348} \qquad \text{(App. A., No. 67)}$$

$$F(X_{18r}, X_{r13,4,1}) = 0.67\, I_{347}\, X_{18r}\, I_{349} \qquad \text{(App. A., No. 68)}$$

$$F(X_{18r}, X_{r13,6,2} + X_{r13,4,2}) = 0.67\, I_{347}\, X_{18r}\, I_{350}\, (X_{4,2} + X_{6,2})/(X_{4,1} - X_{5,1}) \qquad \text{(App. A., No. 69)}$$

$$F(X_{18r}, X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}) = 0.1\, I_{347}\, X_{18r}\, I_{348} \qquad \text{(App. A., No. 70)}$$

$F(X_{18r}, X_{r12,1})$ = transport of resuspendable contaminant, gdwt-1 soil day-1, from the atmosphere to the soil surface for a specific contaminant in a given region
$F(X_{18r}, X_{r13,5,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to live leaf surfaces
$F(X_{18r}, X_{r13,5,2})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dry leaf surfaces
$F(X_{18r}, X_{r13,6,2} + X_{r13,4,2})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dead standing vegetation and dry seeds
$F(X_{18r}, X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dead lying boles and branches
I_{347} = U.S. mean dry deposition velocity, m/ day, for a given region
X_{18r} = regional resuspendable particulate concentration, m^{-3}, for a given contaminant
$X_{r12,1}$ = top soil layer, resuspendable contaminant concentration, $gdwt^{-1}$
$X_{r13,5,1}$ = resuspendable contaminant concentration, m^{-2}, on live leaves
$X_{r13,5,2}$ = resuspendable contaminant concentration, m^{-2}, on dry leaves
$X_{r13,4,1}$ = resuspendable contaminant concentration, m^{-2}, on live standing vegetation

$X_{r13,4,2}$ = resuspendable contaminant concentration, m^{-2}, on dry standing vegetation
$X_{r13,4,3}$ = resuspendable contaminant concentration, m^{-2}, on dry lying vegetation
$X_{r13,4,4}$ = resuspendable contaminant concentration, m^{-2}, on dry lying branches and twigs
$X_{r13,4,5}$ = resuspendable contaminant concentration, m^{-2}, on dry lying boles (main stem)
$X_{r13,6,1}$ = resuspendable contaminant concentration, m^{-2}, on seeds
$X_{r13,6,2}$ = resuspendable contaminant concentration, m^{-2}, on dry seeds
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone
I_{350} = total surface area, m^2/m^2, of combined bole bole and branches of a given plant
I_{37j} = thickness, mm, of the first soil layer (j=1) (input)
I_{19} = leaf area index, m^2/m^2, (LAI)
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

where,

$$I_{347} = 0.0021 \qquad F_1 \leq 2 \qquad \text{(App. B.,I, No. 238)}$$
$$= 0.7\,[\,1 - \exp(-0.6518\,F_1)\,] \qquad F_1 > 2$$

and,

$$I_{348} = \left[\sum_{i=1}^{I_{69}} I_{43}\, I_{265i}\right] / \sum_{i=1}^{I_{69}} I_{265i} \qquad O_1 > 0;\ O_2 > 0 \qquad \text{(App. B.,I, No. 239)}$$
$$= I_{43} \qquad O_1 = 0;\ O_2 = 0$$

I_{347} = U.S. mean dry deposition velocity, m/ day, for a given region
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
F_1 = transport of moisture from rain events to soil, mm/day
I_{43} = plant fractional crown cover, m^2/m^2
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
O_1 = niche competition option (1,0) (input)
O_2 = succession option (1,0) (input)

The total surface area combined boles of a given plant based on an assumed surface area of a cone for these structures:

$$I_{349} = 0.5\,\pi\, I_{163}\, I_{162}\,[\,(0.5\, I_{163}^{\,2}) + I_{351}^{\,2}\,]^{0.5} \qquad \text{(App. B.,I, No. 240)}$$

where,

$$I_{350} = \pi\, I_{352} + I_{350} \qquad \text{(App.. B.,I, No. 241)}$$

$$I_{351} = 0.001\, X_{4,1} \qquad \text{(App. B.,I, No. 242)}$$
$$= 0.3048\, I_{167} \qquad L_2 > 1; I_{166} > 0; I_{167} > 0; I_{165} > 0$$

I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone
I_{350} = total surface area, m^2/m^2, of combined bole bole and branches of a given plant
I_{351} = estimated plant height, meters
I_{352} = projected branch cover, m^2/m^2, on ground surface assuming branches are parallel to the ground

I_{162} = estimated plant areal density, number / m2 for a given plant type at a given time
I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{167} = mean height, ft, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
$X_{4,1}$ = live stem biomass density, g /m^2
I_{166} = mean bole diameter, inches, for a stand of a given plant
L_2 = vegetation index (1,2,3) set to 1 if tree structure simulation not accessed; set to 2 if tree structure simulation is accessed yearly; equal 3 if accessed daily

The loss of resuspendable particulates from the surfaces of live and dead biomass is modeled under the assumption of a removal half-time of 14 days which in includes the effect of rainfall in the wash-off process, and a variable resuspension rate. The losses from resuspension alone are estimated as:

$$F(X_{r13,5,1} + X_{r13,6,1}, X_{18r}) = 0.2\, I_{353}\, (X_{r13,5,1} + X_{r13,6,1})$$ (App. A., No. 71a)

$$F(X_{r13,5,1} + X_{r13,6,1}, X_{r12,1}) = 0.693\, (X_{r13,5,1} + X_{r13,6,1})/14$$ (App. A., No. 71b)

$$F(X_{r13,4,1}, X_{18r}) = 0.02\, I_{353}\, X_{r13,4,1}$$ (App. A., No. 72a)

$$F(X_{r13,4,1}, X_{r12,1}) = 0.693\, X_{r13,4,1} / 42$$ (App. A., No. 72b)

$$F(X_{r13,5,2} + X_{r13,4,2} + X_{r13,6,2}, X_{r18}) = 0.2\, I_{353}\, (X_{r13,5,2} + X_{r13,6,2} + 0.1\, X_{r13,4,2})$$ (App. A., No. 73a)

$$F(X_{r13,5,2} + X_{r13,4,2} + X_{r13,6,2}, X_{r12,1}) = 0.693\, (X_{r13,5,2} + X_{r13,6,2} + 0.1\, X_{r13,4,2})/ 14$$ (App. A., No. 73b)

$$F(X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}, X_{18r}) = 0.02\, I_{353}\, X_{r13,4,3}$$ (App. A., No. 74a)
$$F(X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}, X_{r12,1}) = 0.693\, X_{r13,4,3} / 28$$ (App. A., No. 74b)

where,

$$I_{353} = 2.5e\text{-}08 \exp(-4.6\, I_{348})$$ (App. B.,I, No. 243)
$$= 0 \qquad D_2 > 2 \text{ or } F_2 > 1$$

and where any:

$$X_{r13,j,k} = \left[\sum_{i=1}^{I_{69}} X_{r13,j,k}\, I_{265i} \right] / \sum_{i=1}^{I_{69}} I_{265i} \qquad O_1 > 0;\ O_2 > 0$$ (App. B.,V, No. 11)
$$= X_{13,j,k} \qquad O_1 = 0;\ O_2 = 0$$

$F(X_{r13,5,1}, X_{18r})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from live leaf surfaces to the atmosphere
$F(X_{r13,5,1} + X_{r13,6,1}, X_{r12,1})$ = transport of resuspenable contaminant, m^{-2} day^{-1}, from live leaf and seed surfaces to the soil surface
$F(X_{r13,4,1}, X_{18r})$ = transport of resuspenable contaminant, m^{-2} day^{-1}, from live bole and branch surfaces to the atmosphere
$F(X_{r13,4,1}, X_{r12,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from live bole and branch surfaces to the soil surface
$F(X_{r13,5,2} + X_{r13,4,2} + X_{r13,6,2}, X_{18r})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from dry bole,branches leaves, and seeds surfaces to the atmosphere

$F(X_{r13,5,2} + X_{r13,4,2} + X_{r13,6,2}, X_{r12,1})$ = transport of resuspenable contaminant, m^{-2} day^{-1}, from dry bole,branches leaves, and seeds surfaces to the soil surface

$F(X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}, X_{18r})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from dry lying bole and branch surfaces to the atmosphere

$F(X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}, X_{r12,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from dry lying bole and branch surfaces to the soil surface

O_1 = niche competition option (1,0) (input)

O_2 = succession option (1,0) (input)

D_2 = mean daily temperature, $^\circ$ C

F_2 = transport of moisture from snowfall events to soil, mm/day

I_{265i} = coverage area, acres, of the ith plant type for a given region (input)

I_{353} = resuspension fraction from plant surface to the atmosphere

I_{348} = areally weighted mean plant crown cover, fraction, of a given region

I_{69} = total number of plants in a given region (input)

$X_{12,1}$ = top soil layer contaminant concentration, $gdwt^{-1}$

$X_{r13,j,k}$ = contaminant resuspendable concentration, m-2, on the surface of the jth plant part (roots,stems,leaves,seeds), and kth state (live, dry, dry-lying) for a given plant type in a given region

$X_{r13,5,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of live leaves for a given plant type

$X_{r13,5,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry leaves for all plants in agiven region

$X_{r13,4,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of live standing vegetationfor a given plant type

$X_{r13,4,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region

$X_{r3,4,3}$ = weighted average contaminant resuspendable concentration, on the surface of dry lying vegetation for all plants in a given region

$X_{r13,4,4}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying branches and twigs for all plants in a given region

$X_{r3,4,5}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying boles (main stem) for all plants in a given region

$X_{r3,6,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of seeds for a given plant type

$X_{r13,6,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry seeds for all plants in a given region

7.1.2. Saltation-creep and Rainsplash: Air, Soils and Plants

Analogous relationships exist for estimating the transport of contaminants to plant surfaces from saltation-creep and rainsplash energy events, however, direct deposition of airborne non-resuspendable particles on the soil surface is not modeled because of its highly local nature. The deposition processes are modeled as:

$$F(X_{18s}, X_{s13,5,1}) = 0.4\ X_{18s}\ I_{348}\ X_{5,1} / (X_{5,1} + X_{6,1})$$ (App. A., No. 75a)

$$F(X_{18s}, X_{s13,5,2}) = 0.4\ X_{18s}\ I_{348}\ (1 - I_{354})\ X_{5,2} / (X_{5,2} + X_{4,4} + X_{4,5})$$ (App. A., No. 75b)

$$F(X_{18s}, X_{s13,6,1}) = 0.4\ X_{18s}\ I_{348}\ X_{6,1} / (X_{6,1} + X_{5,1})$$ (App. A., No. 76)

$$F_s(X_{18s}, X_{s13,4,1}) = 0.4\ X_{18s}\ I_{349}\ [1 - X_{4,2} / (X_{4,2} + X_{4,6} + X_{4,7})]\ X_{4,7} / (X_{4,7} + X_{4,6})$$ (App. A., No. 77a)

$$F_b(X_{18s} ,X_{s13,4,1}) = 0.4\, X_{18s}\, I_{352} \{1 - [X_{4,1} /(X_{4,1} + X_{4,6} +X_{4,7})]\}\, X_{4,6} / (X_{4,7} +X_{4,6})$$ (App. A., No. 77b)

$$F(X_{18s} ,X_{s13,6,2} +X_{s13,4,2}) = 0.4\, X_{18s}\, [I_{349}\, X_{4,2} / (X_{4,2} + X_{4,6} +X_{4,7})][I_{352}\, X_{4,1} /(X_{4,1} + X_{4,6} +X_{4,7})]\, I_{355}$$ (App. A., No. 78)

$$F(X_{18s} , X_{s3,4,3} \rightarrow X_{s13,4,4} + X_{s13,4,5}) = 0.4\, X_{18s}\, I_{348} (1- I_{354})\, X_{4,4} / (X_{5,2} + X_{4,4} +X_{4,5})$$ (App. A., No. 79)

$F(X_{18s} ,X_{s3,5,1})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to live leaf surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,5,2})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to dry leaf surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,6,1})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to seed surfaces from saltation-creep and rainsplash events

$F_s(X_{18s} ,X_{s13,4,1})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to bole surfaces from saltation-creep and rainsplash events

$F_b(X_{18s} ,X_{s13,4,1})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to branch surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,6,2} +X_{s13,4,2})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to dead standing vegetation and dry seeds from saltation-creep and rainsplash events

$F(X_{18s} , X_{s13,4,3} \rightarrow X_{s13,4,4} + X_{s13,4,5})$ = transport of contaminant, $m^{-2}\, day^{-1}$, from the atmosphere to dead lying vegetation from saltation-creep and rainsplash events

X_{18s} = regional particulate concentration, m^{-3}, for a given contaminant from saltation-creep and rainsplash events

$X_{4,6}$ = branch-twig component of live above ground biomass, g / m^2

$X_{4,7}$ = bole component of live above ground biomass, g / m^2

$X_{s12,1}$ = top soil layer, saltation-creep-rainsplash contaminant concentration, $gdwt^{-1}$

$X_{s13,5,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live leaves

$X_{s13,5,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry leaves

$X_{s13,4,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live standing vegetation

$X_{s13,4,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry standing vegetation

$X_{s13,4,3}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying vegetation

$X_{s13,4,4}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying branches and twigs

$X_{s13,4,5}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying boles (main stem)

$X_{s13,6,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on seeds

$X_{s13,6,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry seeds

I_{348} = areally weighted mean plant crown cover, fraction, of a given region

I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone

I_{352} = projected branch cover, m^2 / m^2, on ground surface assuming branches are parallel to the ground

$X_{5,1}$ = live leaf compartment, g/ m^2

$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2

$X_{4,2}$ = dry stem biomass density, g / m^2

$X_{6,2}$ = dry seed biomass density, g / m^2

where,

$$I_{b354} = - (76.45/1.21)(I_{164}^{-1.21} - 7^{-1.21}) / (I_{164} - 7)$$ (App. B.,I, No. 244a)

$$I_{354} = - (76.45/1.21)(I_{164}^{-1.21} - I_{356}^{-1.21}) / (I_{164} - I_{356})$$ (App. B.,I, No. 244b)

and,

$$I_{356} = 0.33\, I_{164} \qquad L_2 > 1;\ L_4 = 1 \qquad \text{(App. B.,I, No. 245)}$$
$$= 0.66\, I_{164} \qquad L_2 > 1;\ L_4 = 0$$
$$= 7 \qquad L_2 = 1;\ I_{356} < 7$$

I_{b354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of boles (stems) for various plants
I_{354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of branches for various plants
I_{356} = mean lower height of plant branches, cm, based on predicted plant height for estimating saltation-creep and rainsplash contaminant transport on plants
I_{164} = estimated height of a given plant type, cm, at a given time
L_2 = forest structure access gate (1,2,3) (input)
L_4 = evergreen tree and shrub gate (1,0) (input)

The ratio of the integral average contamination on boles and branches of a given plant to the maximum estimates on these surfaces from saltation-creep, and rainsplash events is modeled as:

$$I_{355} = (X_{sa13,4,6} + X_{sa13,4,7}) / (X_{s13,4,6} + X_{s13,4,7}) \qquad \text{(App. B.,I, No. 246)}$$

where,

$$X_{sa13,4,6} = I_{b354}\, X_{s13,4,6} \qquad \text{(App. B.,V, No. 12)}$$
$$X_{sa13,4,7} = I_{354}\, X_{s13,4,7} \qquad \text{(App. B.,V, No. 13)}$$

I_{355} = the ratio of the integral average contamination on boles and branches of a given plant to the maximum estimates on these surfaces assuming height independence from saltation-creep-rainsplash events
I_{354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of branches for various plants
$X_{sa13,4,6}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant
$X_{sa13,4,7}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant
$X_{s13,4,6}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant
$X_{s13,4,7}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant

The integral average saltation-creep-rainsplash are included in the model to facilitate comparsions between model output and field sampling studies where vegetation samples are obtained from varying elevations above the ground surface [5,47,48], and to account for the rapid decline of surficial contamination from this source as the height of the vegetation above the ground surface is increased [49].

The loss of saltation-creep-rainsplash contaminantion from plant surfaces is also modeled analogous to losses of resuspendable contamination:

$$F\,(X_{s13,5,1} + X_{s13,6,1}\,,\ X_{18s}) = 0.2\, I_{357}\ (X_{s13,5,1} + X_{s13,6,1}) \qquad \text{(App. A., No. 80a)}$$

$$F\,(X_{s13,5,1} + X_{s13,6,1}\,,\ X_{12,1}) = 0.693\ (X_{r13,5,1} + X_{r13,6,1})/\,8 \qquad \text{(App. A., No. 80b)}$$

$$F\,(X_{s13,4,1}\,,\ X_{18s}) = 0.02\, I_{357}\, X_{s13,4,1} \qquad \text{(App. A., No. 81a)}$$

$$F\,(X_{s13,4,1}\,,\ X_{12,1}) = 0.693\, X_{s13,4,1}\,/\,24 \qquad \text{(App. A., No. 81b)}$$

$$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{18s}) = 0.2\, I_{357}\,(X_{sal3,5,2} + X_{sal3,6,2} + 0.1\, X_{sal3,4,2}) \quad \text{(App. A., No. 82a)}$$

$$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{12,1}) = 0.693\,(X_{sal3,5,2} + X_{sal3,6,2} + 0.1\, X_{sal3,4,2})/16 \quad \text{(App. A., No. 82b)}$$

$$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{18s}) = 0.2\, I_{357}\, X_{sal3,4,3} \quad \text{(App. A., No. 83a)}$$

$$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{12,1}) = 0.693\, X_{sal3,4,3} / 16 \quad \text{(App. A., No. 83b)}$$

where,

$$I_{357} = 2.5e\text{-}04 \exp(-3.2\, I_{348}) \quad \text{(App. B.,I, No. 247)}$$
$$= 10\, I_{357} \qquad F_1 > 5;\ D_2 \geq 0$$
$$= 0.0 \qquad F_2 > 0$$

and where any:

$$X_{sal3,j,k} = \left[\sum_{i=1}^{I_{69}} X_{s13,j,k}\, I_{265i} \right] / \sum_{i=1}^{I_{69}} I_{265i} \qquad O_1 > 0;\ O_2 > 0 \quad \text{(App. B.,V, No. 14)}$$
$$= X_{13s,j,k} \qquad O_1 = 0;\ O_2 = 0$$

$F(X_{s13,5,1}, X_{18s})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from live leaf surfaces to the atmosphere from saltation-creep-rainsplash events

$F(X_{s13,5,1} + X_{s13,6,1}, X_{12,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from live leaf and seed surfaces to the soil surface from saltation-creep-rainsplash events

$F(X_{s13,4,1}, X_{18s})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from live bole and branch surfaces to the atmosphere from saltation-creep-rainsplash events

$F(X_{s13,4,1}, X_{12,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from live bole and branch surfaces to the soil surface from saltation-creep-rainsplash events

$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{18s})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from dry bole,branches leaves, and seeds surfaces to the atmosphere from saltation-creep-rainsplash events

$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{12,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from dry bole,branches leaves, and seeds surfaces to the soil surface from saltation-creep-rainsplash events

$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{18s})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from dry lying bole and branch surfaces to the atmosphere from saltation-creep-rainsplash events

$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{12,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from dry lying bole and branch surfaces to the soil surface from saltation-creep-rainsplash events

O_1 = niche competition option (1,0) (input)
O_2 = succession option (1,0) (input)
F_1 = transport of moisture from rain events to soil, mm/day
D_2 = mean daily temperature, °C
F_2 = transport of moisture from snowfall events to soil, mm/day
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
I_{357} = saltation creep-rainsplash fraction from plant and soil surfaces to the atmosphere
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
I_{69} = total number of plants in a given region (input)
$X_{12,1}$ = top soil layer contaminant concentration, $gdwt^{-1}$

$X_{s13,j,k}$ = contaminant concentration, m-2, on the surface of the jth plant part (roots,stems,leaves,seeds), and kth state (live, dry, dry-lying) for a given plant type in a given region from saltation-creep-rainsplash events

$X_{s13,5,1}$ = contaminant concentration, m^{-2}, on the surface of live leaves for a given plant type from saltation-creep-rainsplash events

$X_{sa13,5,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry leaves for all plants in agiven region from saltation-creep-rainsplash events

$X_{s13,4,1}$ = contaminant concentration, m^{-2}, on the surface of live standing vegetationfor a given plant type from saltation-creep-rainsplash events

$X_{sa13,4,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region from saltation-creep-rainsplash events

$X_{sa3,4,3}$ = weighted average contaminant concentration, on the surface of dry lying vegetation for all plants in a given region from saltation-creep-rainsplash events

$X_{sa13,4,4}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry lying branches and twigs for all plants in a given region from saltation-creep-rainsplash events

$X_{sa3,4,5}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying boles (main stem) for all plants in a given region

$X_{s3,6,1}$ = contaminant concentration, m^{-2}, on the surface of seeds for a given plant type from saltation-creep-rainsplash events

$X_{sa13,6,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry seeds for all plants in a given region from saltation-creep-rainsplash events

7.1.3. Root Uptake

The internal transport and translocation of contaminants is initiated by root uptake from each soil horizon that the root has penetrated, and is dependent on the amount tranpiration from each horizon or layer simulated:

$$F(P_{sr}, X_{13,3,1}) = I_{369i} \qquad \text{(App. A., No. 87)}$$

where,

$$\begin{aligned} I_{369i} &= 0.375\,(1 - 1/X_{5,1})\, I_{370i}\, F_{5ai}\, I_{19}\, X_{12,i} && C_3 > 50 \qquad \text{(App. B. ,I, No. 253)} \\ &= 1.25\,(1 - 1/X_{5,1})\, I_{370i}\, F_{5ai}\, I_{19}\, X_{12,i} && C_3 \leq 50 \\ &= 0 && X_{3,1} = 1.0 \text{ or } X_{4,1} = 1.0 \end{aligned}$$

I_{369i} = transport of contaminant, $m^{-2}\ day^{-1}$, from the ith soil layer to the root compartment of a given plant

I_{370i} = partition coefficient , gdwt-plant^{-1} / gdwt-soil^{-1} , between plant and soil components for a given contaminant in the ith soil layer (input)

F_{5ai} = transpiration rate, mm $m^{-2}\ day^{-1}$, from the ith soil layer for a given plant type

I_{19} = leaf area index, m^2/m^2, (LAI)

P_{sr} = soil-to root source term for a given contaminant

$X_{12,i}$ = contaminant concentration, gdwt-soil^{-1}, of the ith soil layer

$X_{5,1}$ = live leaf compartment, g/ m^2

$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2

$X_{3,1}$ = live root compartment, g/ m^2

C_3 = mean life-span, years, of the plant type simulated

$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type

The partition coefficients for organic compounds are estimated from the octanol/ water ratio (kow) [45] :

$$I_{370o} = \exp\,[\,2.30259\,(\,1.588 - 0.578\, I_{371o})] \qquad \text{(App. B.,I, No. 254)}$$

I_{370o} = partition coefficient , gdwt-plant^{-1} / gdwt-soil^{-1} , between plant and soil components for a given organic contaminant in the ith soil layer (input)

I_{371o} = $\log_{10}$ of octanol/ water ratio (kow) of a given organic contaminant (input)

An illustration of the transport strategies for contaminant transport from the soil to the internal portions of roots and to the atmosphere for volatile contaminants is shown in **Fig. 18** along with translocation processes to be described below.

7.1.4. Translocation

The translocation of contaminants within a plant structure is based on mass flow of photosynthate from leaf structures counter-current to the flow of moisture from the roots, and an assumed exchange of contaminants between these flow systems [53,54,55,56]. The following relationships describe the translocation to and from roots, stems, leaves and seeds based on biomass density of the different plant parts and on photosynthate production:

$$F(P_{rs}, X_{13,3,1}) = I_{372}\, X_{13,4,1}\, I_{373b}$$ (App. A., No. 88)

$$F(P_{rs} + P_{sw} + P_{ss}, X_{13,4,1}) = I_{372}\, [X_{13,3,1}\, I_{374} + (X_{13,5,1} + X_{13,6,1})I_{375}]$$ (App. A., No. 89)

$$= I_{372}\, [X_{13,3,1}\, I_{374}/2 + (X_{13,5,1} + X_{13,6,1})I_{375}] \quad (NO_3^- \text{ or soluble } PO_4^{-3})$$

$$F(P_{sw}, X_{13,5,1}) = I_{372}\, X_{13,4,1}\, I_{376b}$$ (App. A., No. 90)

$$F(P_{ss}, X_{13,6,1}) = I_{372}\, X_{3,4,1}$$ (App. A., No. 91)

where,

$$I_{372} = F_{23} / X_{5,1}$$ (App. B.,I, No. 255)

$= 0$ $\quad X_{4,1} = 1.0$ or $I_{372} < 0$

$= 0.99$ $\quad I_{372} > 0.99$

$$I_{373a} = X_{3,1} / (X_{3,1} + X_{4,1} + X_{5,1} + X_{6,1})$$ (App. B.,I, No. 256a)

$= 0.99$ $\quad I_{373a} > 0.99$

$$I_{373b} = I_{373a}$$ (App. B.,I, No. 256b)

$= I_{373a} / (I_{373a} + I_{376a} + I_{377a}) \quad (I_{373a} + I_{376a} + I_{377a}) > 1$

$$I_{374} = 0.5\,(X_{4,1} + X_{5,1} + X_{6,1}) / (X_{4,1} + X_{5,1} + X_{6,1} + X_{3,1})$$ (App. B.,I, No. 257)

$= 0.99$ $\quad I_{374} > 0.99$

$$I_{375} = X_{4,1} / (X_{4,1} + X_{5,1} + X_{6,1})$$ (App. B.,I, No. 258)

$= 0.5$ $\quad I_{375} > 0.5$

$$I_{376a} = 0.5\, X_{5,1} / (X_{4,1} + X_{5,1} + X_{6,1})$$ (App. B.,I, No. 259a)

$= 0.99$ $\quad I_{376a} > 0.99$

$$I_{376b} = I_{376a}$$ (App. B.,I, No. 259b)

$= I_{376a} / (I_{373a} + I_{376a} + I_{377a}) \quad (I_{373a} + I_{376a} + I_{377a}) > 1$

$$I_{377a} = 0.5\, X_{6,1} / (X_{6,1} + X_{5,1} + X_{6,1})$$ (App. B.,I, No. 260a)

$= 0.99$ $\quad I_{377a} > 0.99$

$$I_{377b} = I_{377a}$$ (App. B.,I, No. 260b)

$= I_{377a} / (I_{373a} + I_{376a} + I_{377a})$ $\quad (I_{373a} + I_{376a} + I_{377a}) > 1$

$F(P_{rs}, X_{13,3,1})$ = transport of contaminant, m^{-2} day^{-1}, from stem compartment of a given plant type to the root compartment

$F(P_{rs} + P_{sw} + P_{ss}, X_{13,4,1})$ = transport of contaminant, m^{-2} day^{-1}, from the root, leaf and seed compartments to the stem compartment of a given plant type

$F(P_{sw}, X_{13,5,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the leaf compartment of a given plant type

$F(P_{ss}, X_{13,6,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the seed or fruit compartment of a given plant type

I_{372} = ratio of net daily photosynthate production, to standing live leaf biomass, day^{-1}

I_{373a} = initial biomass density factor: stems to total biomass

I_{373b} = adjusted biomass density factor: stems to total biomass

I_{374} = above ground biomass density factor: stems,leaves, and seeds to total biomass

I_{375} = biomass density factor: stems to above ground biomass

I_{376a} = initial biomass density factor: leaves to above ground biomass

I_{376b} = adjusted biomass density factor: leaves to above ground biomass

I_{377a} = initial biomass density factor: seeds to above ground biomass

I_{377b} = adjusted biomass density factor: seeds to above ground biomass

$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type

$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type

$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type

$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type

F_{23} = net photosynthesis, $g/m^2/day$, for a given plant type

$X_{5,1}$ = live leaf compartment, g/m^2

$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2

$X_{3,1}$ = live root compartment, g/m^2

$X_{6,1}$ = live seed biomass density, g/m^2

P_{rs} = root-stem contaminant exchange source

P_{sw} = stem-leaf contaminant exchange source

P_{ss} = stem-seed contaminant exchange source

7.2. Soils

7.2.1. Eluviation

The loss of contaminant from the surface layer and consequent transport to the second layer of soil is modeled as an eluviation process assuming that it eluviates with a rate equal to that of humus (0.2mm/year) which was estimated from humus profile data [35,50].

$F_e(X_{12,1}, X_{12,2}) = I_{358} I_{37i} X_{12,1} / I_{37j}$ (App. A., No. 84)

where,

$I_{358} = 0.000025 / I_{37i}$ (App. B.,I, No. 248)

$= I_{358}/10$ $\quad I_{359} = 19$

$= I_{358}/10$ $\quad I_{359} = 55$

$F_e(X_{12,1}, X_{12,2})$ = eluviation of a given contaminant, day^{-1}, from the soil surface into the second soil layer

I_{358} = eluviation rate, mm^{-1} day^{-1}, of contaminant from soil surface to the second layer

I_{359} = atomic number of elements or for other compounds identification numbers which are greater than 100 (input)

$X_{12,1}$ = contaminant concentration, gdwt-soil^{-1}, of the surface layer of soil (20 mm thick)
$X_{12,2}$ = contaminant concentration, gdwt-soil^{-1}, of the second soil layer
I_{37i} = thickness of the 1st (i=1)soil layer, mm (input)
I_{37j} = thickness of the 2nd (j=2) soil layer, mm (input)

Cesium and potasssium eluviation rates are modeled as one-tenth the eluviation of other substances because they are trapped in the lattice structure of montmorillonite clays [51].

7.2.2. Soil Moisture

The transport of contaminants within the soil column is modeled through the use of published soil/ water distribution coefficients [52] and soil flow parameters described earlier:

$$I_{360} = 1000\, F_{7a}\, X_{12,i} / I_{362} \qquad \text{(App. B.,I, No. 249a)}$$

$$I_{361} = 1000\, F_{7}\, X_{12,i} / I_{362} \qquad \text{(App. B.,I, No. 249b)}$$

$F_w (X_{12,i} , X_{12,i-1})$ = transport of contaminant from the ith soil layer to the (i-1)th layer (top layer excluded), day^{-1} gdwt-soil^{-1}, by soil moisture flow
$F_w (X_{12,i} , X_{12,i+1})$ = transport of contaminant from the ith soil layer to the (i+1)th layer (bottom layer excluded), day^{-1} gdwt-soil^{-1} from soil moisture flow
I_{37j+1} = thickness, of the (i+1)th soil layer, mm (input)
I_{37j-1} = thickness, of the (i-1)th soil layer, mm (input)
I_{360} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i-1)th layer (top layer excluded) by soil moisture flow
I_{361} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i+1)th layer (bottom layer excluded) by soil moisture flow
I_{362} = distribution coefficient , ml /gdwt-soil, for a given contaminant (input)
F_{7a} = transport of moisture, mm/day, from the ith soil layer to the (i-1)th (layer (top layer excluded)
F_7 = transport of moisture, mm/day, from the ith soil layer to the (i+1)th (layer (bottom layer excluded)
$X_{12,i}$ = contaminant concentration, gdwt-soil^{-1}, of the ith soil layer
$X_{12,i+1}$ = contaminant concentration, gdwt-soil^{-1}, of the (i+1)th soil layer
$X_{12,i-1}$ = contaminant concentration, gdwt-soil^{-1}, of the (i-1)th soil layer

7.2.3. Volatile Contaminants

The loss of volatile and semi-volatile contaminants from any given soil layer and consequent air concentration increment from the gas vapor from contaminant attached to soil particles is simulated assuming a two meter mixing height , a mean wind velocity of 2.25 m/s, a ten degree Celsius mean soil profile temperature, a soil porosity of 0.35, a soil density of 2.65 g/cm3, a 2% organic matter content in each soil layer modeled, and a one day exposure (86,400 seconds)[45]. The total daily loss of contaminant to the atmosphere is estimated from a given soil layer is estimated as:

$$F (X_{12,i} , X_{18g}) = 1400\, I_{37i}\, I_{363}\, X^2_{12,i} / \sum_{i=1}^{I_{20}} 1400\, I_{37i}\, X_{12,i} \qquad \text{(App. A., No. 86)}$$

where,

$$I_{363} = \sum_{j=1}^{I_{69}} \sum_{i=1}^{I_{20}} 86400\, I_{364}\, X_{12,i}\, I_{37i,j}\, I_{265j} / (I_{56j} \sum_{j=1}^{I_{69}} I_{265j}) \qquad \text{(App. B.,I, No. 250)}$$

$F(X_{12,i}, X_{18g})$ = transport of volatile contaminant to the atmosphere, ug day-1, from the ith soil layer of a given region
$I_{37j,j}$ = thickness, of the ith soil layer, mm, of the jth plant type (input)
I_{363} = mean loss of contaminant, ug day^{-1} m^{-2}, from a given region averaged over all plant types and soil horizons
$X_{12,i}$ = volatile contaminant concentration, ug gdwt-$soil^{-1}$, of the ith soil layer
I_{20} = number of soil layers specified for the ith plant (input or estimated)
I_{69} = total number of plants in a given region (input)
I_{265j} = coverage area, acres, of the jth plant type for a given region (input)
I_{56j} = total depth, mm, of soil profile for the rooting zone of the jth plant type (input)
I_{364} = volatile contaminant emission rate, ug m^{-2} sec^{-1}, assuming a soil concentration of one ug /gdwt

The volatile contaminant emission rate is estimated as:

$$I_{364} = 0.203\, I_{365}\, I_{366} / [\, 1.919e{+}06\, I_{366} /(10 + I_{367} / I_{365})]^{1/2} \qquad \text{(App. B.,I, No. 251)}$$

I_{364} = volatile contaminant emission rate, ug m^{-2} sec^{-1}, averaged over one day assuming a soil concentration of one ug /gdwt
I_{365} = Henry's Law constant, atm-m^3 / mole (input)
I_{366} = molecular diffusivity, cm^2 / sec (input)
I_{367} = distribution coefficient of contaminant on organic carbon, ml/g, (input)

When the molecular diffusivity of a given contaminant is unknown, it may be estimated by using the known diffusivity of benzene:

$$I_{366x} = I_{366bn}\,(I_{368bn} / I_{368x})^{1/2} \qquad \text{(App. B.,I, No. 252)}$$

I_{366x} = estimated molecular diffusivity, cm^2 / sec, of a specific organic compound using the molecular diffusivity of benzene
I_{366bn} = molecular diffusivity of benzene (0.08195 cm^2 /sec) (input)
I_{368x} = molecular weight, grams, of specific organic compound (input)
I_{368bn} = molecular weight, grams, of benzene (78g) (input)

The resulting air concentration from the volatilization of a contaminant is then estimated by assuming a two meter mixing layer above the ground surface [45].

7.3. Mammals

7.3.1. Pharmacokinetic Transport

The flow of contaminants through mammals uses a physiologically based pharmacokinetics [57] approach. A diagram of this strategy is presented in Fig. 19. which is applicable to invertebrates and humans as well. The uptake, translocation, and elimination of contaminants from the body of an organism is simulated by utilizing the blood plasma perfusion rates of separate organs and tissues [57,58], coupled with a set of partition coefficients between these organs and tissues and the blood plasma itself. The partition coefficient for any particular organ or tissue is defined as the ratio of the contaminant concentration in the organ or tissue of interest to the concentration of the contaminant in the blood plasma under equilibrium conditions. The latter is assumed to alternate between arterial transport from the heart lumen to the perfused organs , and return to that lumen via the venous return. During the perfusion of any given organ or tissue, the contaminant concentration present in blood plasma will either decrease (organ or tissue uptake), increase (organ or tissue release), or remain constant (equilibrium conditions), as the plasma traverses the organ to the venous side of the circulation. In cases where volatile contaminants are

involved, the lung lumen also serves as an exchange site, and will either concentrate, release, or not affect blood plasma concentrations, depending on the partition coefficient between alveolar air and blood plasma, and on the relative concentrations of contaminants in the these media. The kidneys are of particular importance in that they represent the main source of excretion for most contaminants in the body. Endogenous excretion by the colon is not modeled directly in BIOTRAN.2, but is included as part of the fecal excretion route along with any molecular transformations and detoxification of contaminants in the liver. Molecular transformation of contaminants to other toxic forms by the liver is not addressed in the present version of BIOTRAN.2, however, the model is poised for such development should the need arise. Hence, the fraction of contaminant which enters the blood circulation is lumped to include all endogenous excretion by the colon and liver in the manner described above. Also, the hepatic-portal shunt betrween the GI tract and the liver is modeled in BIOTRAN.2 to represent the initial transport of absorbed contaminants from the GI to this organ before entering the general circulation. Finally, the circulation and exchange of contminant between mother and fetus is simulated in BIOTRAN.2 as well as contaminant transport during the weaning period in mammals.

The blood plasma compartment is modeled as the central compartment of a mammillary system [59] with other organs and tissues as peripheral compartments as shown in Fig. 19. Of necessity, the time units are set in hours, rather than days as they are for all other processes in BIOTRAN.2 with one exception. This modification was included to more closely simulate concentrations changes of contaminants with rapid equilibration rates. For very rapid equilibration rates, the simulations are only approximate, but for the intended purpose of BIOTRAN.2 in simulating body burdens on a daily basis, the approximate burdens are assumed to be acceptable. The analysis of the pharmacokinetics equations should make the rationale for this assumption clearer. This analysis will proceed by analyzing the simpler organ and tissue equilbrations initially saving the more complex equilibrations for later treatment. The exchange of a given contaminant in the muscle with blood plasma is estimated as:

$$F(fM_a, X_{17,2,n}) = X_{fl7,2}\ [X_{17,1,n} / X_{vl7,1} - X_{17,2,n} / (X_{vl7,2}\, X_{rl7,2,n})]\ I_{378} \qquad \text{(App. A., No. 92)}$$

where,

$$I_{378} = X_{vl7,1} / \sum_{i=1}^{6} X_{fl7,i} \qquad \text{(App. B.,I, No. 261)}$$

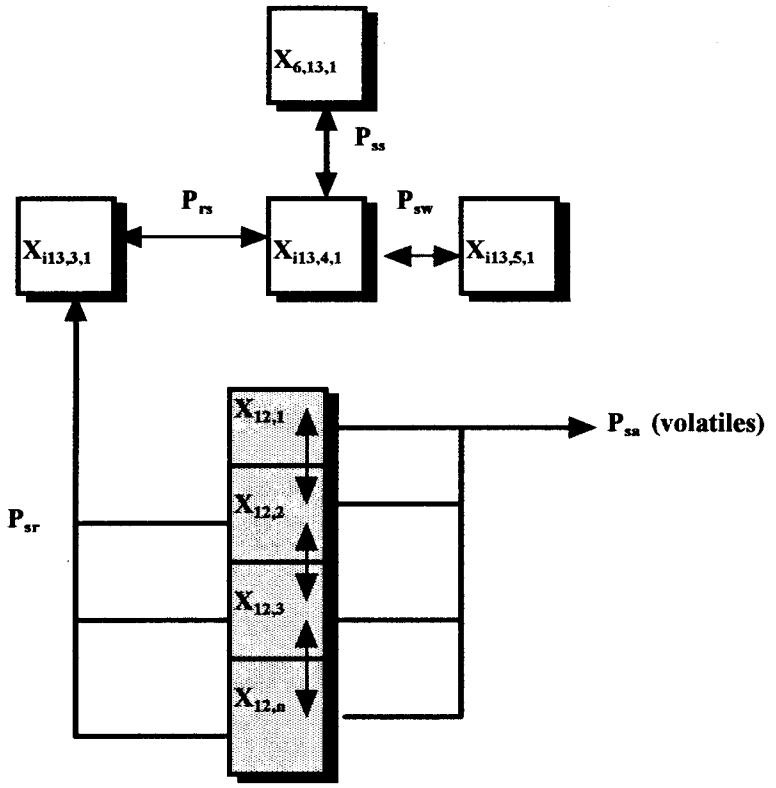
$F(fM_a, X_{17,2,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and muscle for the nth contaminant
fM_a = transport of contaminant from blood plasma to muscle tissue, hr^{-1}
$X_{17,2,n}$ = contaminant content of muscle tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,2}$ = perfusion rate, liters hr^{-1}, of blood plasma into muscle tissue
$X_{fl7,i}$ = perfusion rate, liters hr^{-1}, of the ith organ or tissue
$X_{vl7,1}$ = volume of blood plasma, liters
$X_{vl7,2}$ = perfusion volume of muscle tissue, liters
$X_{rl7,2,n}$ = partition coefficient for the nth contaminant in muscle tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

With exception of the normalizing constant employed for short equilibration periods , the relationship is the standard pharmacokinetic equation commonly used [57,] for estimating this type material exchange between blood plasma and a given organ or tissue. The volume of blood plasma for a given animal and age group is estimated as:

$$X_{vl7,1} = 4.29e\text{-}05\ (1000\ X_{14,m,k})^{0.992} \qquad \text{(App. B.,V, No. 15)}$$

and the perfusion volume of the muscle tissue for a given animal and age group is estimated as,

$$X_{vl7,2} = 4.63e\text{-}04\ (1000\ X_{14,m,k})^{1.009} \qquad \text{(App. B.,V, No. 16)}$$



P_{sr} = soil-root uptake, mm^{-1} transpirate m^{-2} day^{-1}
P_{rs} = root-stem exchange , m^{-2} day^{-1}
P_{sw} = stem-leaf exchange , m^{-2} day^{-1}
P_{ss} = stem-seed exchange , m^{-2} day^{-1}
P_{sw} = stem-leaf exchange , m^{-2} day^{-1}
P_{sa} = soil-atmosphere transport , m^{-2} day^{-1}

Legend

$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type
$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type
$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type
$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type

Fig. 18

Internal Contaminant Concentration In Biomass Compartments

Similarly, the perfusion rate of plasma into muscle tissue for a given animal and age group is estimated as:

$$X_{fl7,2} = 1.18e\text{-}01\ (1000\ X_{14,m,k})^{0.81} \qquad \text{(App. B.,V, No. 17)}$$

$X_{vl7,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{vl7,2}$ = perfusion volume of muscle tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,2}$ = perfusion rate, liters hr^{-1}, of blood plasma into muscle tissue for a given animal and age group

When not available, some partition coefficients can be estimated from available transport coefficient or associated biological half-life databases for radionuclides, metals in humans[52], The following relationship is used in BIOTRAN.2 to make the estimates using a 70 kg Standard Man [60] as a point of reference:

$$X_{rl7,n} = 0.109 / X_{tl7,i,n} \qquad \text{(App. B.,V, No. 18)}$$

$X_{rl7,i,n}$ = partition coefficient of the ith organ or tissue for the nth contaminant in muscle of a given animal type
$X_{tl7,i,n}$ = transport coefficient, day^{-1}, of the ith organ or tissue for the nth contaminant in Standard Man (input)

The assumption is made for modeling purposes in BIOTRAN.2 that partition coefficients should not vary by more than a factor of two for most contaminants across all mammalian types. The exchange of a given contaminant between adipose tissue and blood plasma is estimated as:

$$F\ (fF_a, X_{17,3,n}) = X_{fl7,3}\ [X_{17,1,n} / X_{vl7,1} - X_{17,3,n} / (X_{vl7,3}\ X_{rl7,3,n})]\ I_{378} \qquad \text{(App. A., No. 93)}$$

$F\ (fF_a, X_{17,3,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and adipose tissue for the nth contaminant
fF_a = transport of contaminant from blood plasma to muscle tissue, hr^{-1} $X_{17,3,n}$ = contaminant content of adipose tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,3}$ = perfusion rate, liters hr^{-1}, of blood plasma into adipose tissue
$X_{vl7,1}$ = volume of blood plasma, liters
$X_{vl7,3}$ = perfusion volume of adipose tissue, liters
$X_{rl7,3,n}$ = partition coefficient for the nth contaminant in adipose tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

The perfusion volume of the adipose tissue for a given animal and age group is estimated as,

$$X_{vl7,3} = 1.1e\text{-}04\ (1000\ X_{14,m,k}) \qquad \text{(App. B.,V, No. 19)}$$

Similarly, the perfusion rate of plasma into adipose tissue for a given animal and age group is estimated as:

$$X_{fl7,3} = 0.75\ X_{fl7,2} \qquad \text{(App. B.,V, No. 20)}$$

$X_{vl7,3}$ = perfusion volume of adipose tissue, liters, for a given animal and age group

$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{fl7,3}$ = perfusion rate, liters hr^{-1}, of blood plasma into adipose tissue for a given animal and age group

The exchange of a given contaminant between residual tissue and blood plasma is estimated as:

$$F(fR_a, X_{17,4,n}) = X_{fl7,4}\ [X_{17,1,n} / X_{vl7,1} - X_{17,4,n} / (X_{vl7,4}\ X_{rl7,4,n})]\ I_{378} \qquad \text{(App. A., No. 94a)}$$

and the transport of contaminant to milk from nursing females is estimated as:

$$F(fM_{ka}, X_{wnl7,1,n}) = X_{fl7,4}\ [X_{17,1,n} / (X_{vl7,1}\ X_{fl7,1})] \qquad \text{(App. A., No. 94b)}$$

$F(fR_a, X_{17,3,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and residual tissue for the nth contaminant

$F(fM_{ka}, X_{wnl7,1,n})$ = nth transport of contaminant, hr^{-1}, from residual tissue to milk

$X_{wnl7,1,n}$ = nth contaminant transport, mass or activity units day^{-1}, from mother's milk to a weaning newborn animal of a given animal population

fR_a = transport of contaminant from blood plasma to residual tissue, hr^{-1}

$X_{17,4,n}$ = contaminant content of residual tissue, mass or activity units, for the nth contaminant

$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant

$X_{fl7,4}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue

$X_{fl7,1}$ = total blood plasma flow, liters hr^{-1}

$X_{vl7,1}$ = volume of blood plasma, liters

$X_{vl7,4}$ = perfusion volume of residual tissue, liters

$X_{rl7,4,n}$ = partition coefficient for the nth contaminant in residual tissue (input)

I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

The perfusion volume of the residual tissue for a given animal and age group is estimated as,

$$X_{vl7,4} = X_{vl7,5} + X_{vl7,6}\ X_{vl7,7} + X_{vl7,8} \qquad \text{(App. B.,V, No. 21)}$$

where,

$$X_{vl7,5} = 2.0e\text{-}06\ (1000\ X_{14,m,k})^{1.043} \qquad \text{(App. B.,V, No. 22)}$$

$$X_{vl7,6} = 3.16e\text{-}06\ (1000\ X_{14,m,k})^{1.104} \qquad \text{(App. B.,V, No. 23)}$$

$$X_{vl7,7} = 8.65e\text{-}05\ (1000\ X_{14,m,k})^{0.909} \qquad \text{(App. B.,V, No. 24)}$$

$$X_{vl7,8} = 4.52e\text{-}06\ (1000\ X_{14,m,k})^{0.901} \qquad \text{(App. B.,V, No. 25)}$$

Similarly, the perfusion rate of plasma into residual tissue for a given animal and age group is estimated as:

$$X_{fl7,4} = X_{fl7,5} + 0.1\ X_{fl7,6} + 0.1\ X_{fl7,7} + X_{fl7,8} \qquad \text{(App. B.,V, No. 26)}$$

where,

$$X_{fl7,5} = 8.31e\text{-}03\ (1000\ X_{14,m,k})^{0.965} \qquad \text{(App. B.,V, No. 27)}$$

$$X_{fl7,6} = 0.31\ (1000\ X_{14,m,k})^{0.82} \qquad \text{(App. B.,V, No. 28)}$$

$$X_{fl7,7} = 0.147\ (1000\ X_{14,m,k})^{0.787} \qquad \text{(App. B.,V, No. 29)}$$

fP_a = arterial plasma flow

fP_v = venous plasma flow

fR_a = arterial flow to residual tissue volumes

fR_v = venous flow from residual tissue volumes

fLa = arterial flow to liver tissue volume

fL_v = venous flow from liver tissue volume

fG_v = hepatic-portal shunt flow from gut lumen to liver

fV_a = volatile-gas and souble exchange/transport between alveoli and arterial blood, hr^{-1}

fG_p = particulate transport from lung to gut lumen

fM_a = arterial flow to muscle tissue volume

fM_v = venous flow from muscle tissue volume

fK_a = arterial flow to kidney tissue volume

fK_v = venous flow from kidney tissue volume

fB_a = arterial flow to bone tissue volume

fB_v = venous flow from bone tissue volume

fF_a = arterial flow to adipose tissue volume

fF_v = venous flow from adipose tissue volume

note: all flows in liters hr^{-1}
all volumes in liters

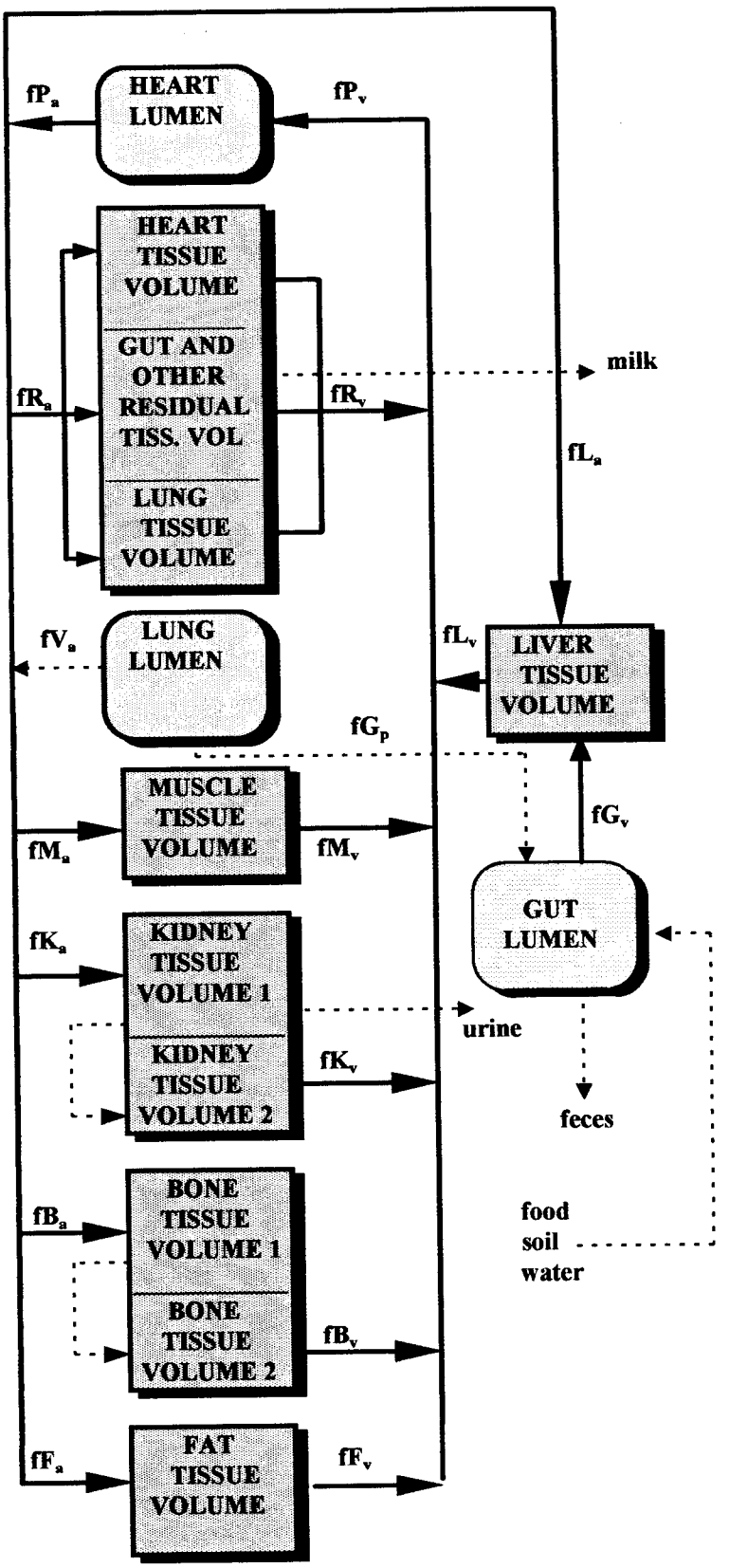


Fig. 19
Pharmacokinetics Transport Strategy for Contaminants used in BIOTRAN.2

$$X_{fl7,8} = 2.58e\text{-}03\ (1000\ X_{14,m,k})^{1.028}$$ (App. B.,V, No. 30)

$X_{v17,4}$ = perfusion volume of residual tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,4}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue for a given animal and age group
$X_{v17,5}$ = perfusion volume of heart tissue, liters, for a given animal and age group
$X_{v17,6}$ = perfusion volume of lung tissue, liters, for a given animal and age group
$X_{v17,7}$ = perfusion volume of gut tissue, liters, for a given animal and age group
$X_{v17,8}$ = perfusion volume of spleen tissue, liters, for a given animal and age group
$X_{fl7,5}$ = perfusion rate, liters hr^{-1}, of blood plasma into heart tissue for a given animal and age group
$X_{fl7,6}$ = perfusion rate, liters hr^{-1}, of blood plasma into lung tissue for a given animal and age group
$X_{fl7,7}$ = perfusion rate, liters hr^{-1}, of blood plasma into gut tissue for a given animal and age group
$X_{fl7,8}$ = perfusion rate, liters hr^{-1}, of blood plasma into spleen tissue for a given animal and age group

The residual tissue is shown to be represented in BIOTRAN.2 by the heart, lungs, gut including stomach and intestines, and the spleen. other minor organs and tissues are assumed to reach contaminant concentrations equal to these tissues. Furthermore, the major perfusion flows to residual tissues are estimated to equal the entire flow of plasma to the heart and spleen, and one-tenth the flow to the lungs and gut tissues. The remaining plasma flow to the lungs becomes part of the total plasma flow, and the remaining plasma flow is assumed to enter the hepatic-portal circulation, and part of the plasma flow to the liver to be described below.

The exchange of a given contaminant in bone tissue with blood plasma may be associated with both bone marrow, and trabecular bone compartments in BIOTRAN.2, however, the user may choose to consider this type of tissue homogeneously for some contaminants, The perfused plasma into both components is assumed to be in series rather than in parallel because it is assumed that a contaminant will be distributed homogeneously throughout the plasma in that tissue. It is also assumed for modeling purposes, that the perfusion volume of marrow and trabecular bone compartments are equal. This dual exchange is estimated as:

$$F\ (fB_a, X_{17,9a,n}) = 2.0\ X_{fl7,9}\ [X_{17,1,n} / X_{v17,1} - X_{17,9an} / (X_{v17,9}\ X_{rl7,9a,n})]\ I_{378}$$ (App. A., No. 95a)

$$F\ (fB_a, X_{17,9b,n}) = 2.0\ X_{fl7,9}\ [X_{17,1,n} / X_{v17,1} - X_{17,9bn} / (X_{v17,9}\ X_{rl7,9b,n})]\ I_{378}$$ (App. A., No. 95b)

where,

$F\ (fB_a, X_{17,9a,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and bone marrow for the nth contaminant
$F\ (fB_a, X_{17,9b,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and trabecular bone for the nth contaminant
fB_a = transport of contaminant from blood plasma to bone tissue, hr^{-1}
$X_{17,9a,n}$ = contaminant content of bone marrow tissue, mass or activity units, for the nth contaminant
$X_{17,9b,n}$ = contaminant content of trabecular bone tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,9}$ = perfusion rate, liters hr^{-1}, of blood plasma into both marrow and trabecular tissue
$X_{fl7,i}$ = perfusion rate, liters hr^{-1}, of the ith organ or tissue

$X_{v17,1}$ = volume of blood plasma, liters
$X_{v17,9a}$ = perfusion volume of bone marrow tissue, liters
$X_{v17,9b}$ = perfusion volume of trabecular bone tissue, liters
$X_{r17,9a,n}$ = partition coefficient for the nth contaminant in bone marrow tissue (input)
$X_{r17,9b,n}$ = partition coefficient for the nth contaminant in trabecular bone tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

The perfusion volume of both bone marrow and trabecular bone tissue are assumed tobe equal, and are estimated for a given animal and age group as:

$$X_{v17,9} = 3.3e\text{-}05\ (1000\ X_{14,m,k})^{0.936} \qquad \text{(App. B., V, No. 31)}$$

Similarly, the perfusion rate of plasma into both bone marrow and trabecular bonetissue for a given animal and age group is estimated as:

$$X_{f17,9} = 2.36e\text{-}01\ (1000\ X_{14,m,k})^{0.84} \qquad \text{(App. B.,V, No. 32)}$$

$X_{v17,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{v17,9}$ = perfusion volume of bone tissues, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{f17,9}$ = perfusion rate, liters hr^{-1}, of blood plasma into both bone marrow and trabecular bone tissue for a given animal and age group

The exchange of a given contaminant in kidney tissue with blood plasma is transport may associated with both medullary, and cortical bone compartments in BIOTRAN.2, however, the user may choose to consider this type of tissue homogeneously for some contaminants, The perfused plasma into both components is assumed to be series because it is assumed that a contaminant will be distributed homogeneously throughout the plasma in that tissue as was noted for bone tisssues. It is also assumed for modeling purposes, that the perfusion volumes of medullary and cortical kidney compartments are equal. It is also assumed that the total perfusion rate into bone tissue is twice the flow into the bone marrow. These exchanges are estimated in the following manner:

$$F\ (fK_a, X_{17,10a,n}) = X_{f17,10}\ [X_{17,1,n} / X_{v17,1} - X_{17,10an} / (X_{v17,10}\ X_{r17,10a,n})]\ I_{378} \qquad \text{(App. A., No. 96a)}$$

$$F\ (fK_a, X_{17,10b,n}) = X_{f17,10}\ [X_{17,1,n} / X_{v17,1} - X_{17,10bn} / (X_{v17,10}\ X_{r17,10b,n})]\ I_{378} \qquad \text{(App. A., No. 96b)}$$

where,

$F\ (fK_a, X_{17,10a,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and cortical tissue of the kidney for the nth contaminant
$F\ (fK_a, X_{17,10b,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and medullary tissue of the kidney for the nth contaminant
fK_a = transport of contaminant from blood plasma to kidney tissue, hr^{-1}
$X_{17,10a,n}$ = contaminant content of kidney cortical tissue, mass or activity units, for the nth contaminant
$X_{17,10b,n}$ = contaminant content of kidney medullary tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{f17,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into both cortical and medullary tissue
$X_{v17,1}$ = volume of blood plasma, liters

$X_{v17,10a}$ = perfusion volume of kidney cortical tissue, liters
$X_{v17,10b}$ = perfusion volume of kidney medullary tissue, liters
$X_{r17,10a,n}$ = partition coefficient for the nth contaminant in kidney cortical tissue (input)
$X_{r17,10b,n}$ = partition coefficient for the nth contaminant in kidney medullary tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

The perfusion volume of both kidney cortical and medullary tissues are assumed equal, and are estimated for a given animal and age group as:

$$X_{v17,10} = 2.18e\text{-}05\ (1000\ X_{14,m,k})^{0.843} \qquad \text{(App. B., V, No. 33)}$$

Similarly, the perfusion rate of plasma into both bone marrow and trabecular bonetissue for a given animal and age group is estimated as:

$$X_{fl7,10} = 1.22e\text{-}01\ (1000\ X_{14,m,k})^{0.802} \qquad \text{(App. B.,V, No. 34)}$$

$X_{v17,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{v17,10}$ = perfusion volume of kidney tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into kidney tissue for a given animal and age group

The production of urine by the kidneys removes contaminant from the blood plasma which is modeled as follows:

$$F\ (X_{17,1,n}\ ,\ X_{ul7,11,n}) = X_{fl7,10}\ X_{17,1,n} / X_{fl7,1} \qquad \text{(App. A., No. 97)}$$

where,

$$X_{fl7,1} = 8.65e\text{-}01(1000\ X_{14,m,k})^{0.778} \qquad \text{(App. B.,V, No. 35)}$$

$F\ (X_{17,1,n}\ ,\ X_{ul7,11,n})$ = transport of contaminant from blood plasma, hr^{-1}, to urine
$X_{fl7,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into kidney tissue for a given animal and age group
$X_{fl7,1}$ = total plasma flow, liters hr^{-1}
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{17,1,n}$ = contaminant content, mass or activity units, in the blood plasma
$X_{ul7,11,n}$ = contaminant content, mass or activity units, in urine

The exchange of a given contaminant between liver tissue and blood plasma is estimated as:

$$F\ (fL_a + fG_v\ ,\ X_{17,12,n}) = X_{fl7,12}\ [X_{17,1,n} / X_{v17,1} - (X_{17,12,n} + I_{379}) / (X_{v17,12}\ X_{r17,12,n})]\ I_{378} \qquad \text{(App. A., No. 98)}$$

$F\ (fL_a + fG_v,\ X_{17,12,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and liver tissue including gut transport for the nth contaminant

fL_a = transport of contaminant from blood plasma to liver tissue, hr^{-1}

fG_v = transport of contaminant from blood plasma to liver tissue from the gut via the hepatic portal vein, hr^{-1}

$X_{17,12,n}$ = contaminant content of liver tissue, mass or activity units, for the nth contaminant

$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant

$X_{fl7,12}$ = perfusion rate, liters hr^{-1}, of blood plasma into liver tissue

$X_{v17,1}$ = volume of blood plasma, liters

$X_{v17,12}$ = perfusion volume of liver tissue, liters

$X_{rl7,12,n}$ = partition coefficient for the nth contaminant in liver tissue (input)

I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

The perfusion volume of liver tissue for a given animal and age group is estimated as,

$$X_{v17,12} = 8.59e\text{-}05\ (1000\ X_{14,m,k})^{0.885} \qquad \text{(App. B.,V, No. 36)}$$

Similarly, the perfusion rate of plasma into liver tissue for a given animal and age group including hepatic portal addition is estimated as:

$$X_{fl7,12} = 0.136\ (1000\ X_{14,m,k})^{0.792} + 0.9\ X_{fl7,7} \qquad \text{(App. B.,V, No. 37)}$$

$X_{v17,12}$ = perfusion volume of liver tissue, liters, for a given animal and age group

$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{fl7,12}$ = perfusion rate, liters hr^{-1}, of blood plasma into liver tissue for a given animal and age group including hepatic-portal vein contribution from the gut

$X_{fl7,7}$ = perfusion rate, liters hr^{-1}, of blood plasma into gut tissue for a given animal and age group

The transport of contaminants from the gut to the liver is estimated as:

$$I_{379} = I_{380m,k} + I_{381m,k} + I_{382m,k} + I_{383m,k} + I_{384m,k} \qquad \text{(App. B.,I, No. 262)}$$

where the contribution of contamination from the lung lumen surface is estimated as:

$$I_{380m,k} = I_{385}\ (\ I_{386}\ X_{1el7,6,n} + I_{387}\ X_{2el7,6,n}\ ) \qquad \text{(App. B.,I, No. 263)}$$

and,

$$I_{380b} = I_{388}\ X_{3el7,6,n} \qquad \text{(App. B.,I, No. 263b)}$$

I_{380} = transport of contamination, day^{-1}, from the surface of the lung lumen to the liver via the hepatic portal-vein from the gut

I_{380b} = transport of contamination ,day^{-1}, from the pulmonary surface of the lung directly into arterial blood

I_{379} = total transport of contaminant, day^{-1}, from the gut to the liver via the hepatic-vein

$I_{381m,k}$ = transport of contaminant from the N-P, and upper T-B regions of the respiratory tract to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population

$I_{382m,k}$ = regionally weighted contaminant transport from ingested food to the mth age group of the kth mammalian predator population

$I_{383m,k}$ = amount of soil consumed, gdwt day^{-1}, by an an animal from grazing for the mth age group of the kth mammalian population

$I_{384m,k}$ = amount of contaminant, day-1, transported to the hepatic-portal vein from drinking water via the gut for the mth age group of the kth mammalian population

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)

I_{386} = fraction of a given contaminant transferred, day^{-1}, from the bronchus region of the lung lumen surface to the gut via particulates (input)

I_{387} = fraction of a given contaminant transferred, day^{-1}, from the bronchiole region of the lung lumen surface to the gut via particulates (input)

I_{388} = fraction of a given contaminant transferred, day^{-1}, from the pulmonary region of the lung lumen surface to the blood (input)

$X_{1e17,6,n}$ = contaminant content on the surface of the bronchus region of the lung lumen as particulates

$X_{2e17,6,n}$ = contaminant content on the surface of the bronchiole region of the lung lumen as particulates

$X_{3e17,6,n}$ = contaminant content on the surface of the pulmonary region of the lung lumen as particulates

7.3.2. Inhaled Resuspendable Particulates

The contribution of inhaled particulate contaminant from the naso-phyrangeal region of the respiratory tract to the liver via the hepatic portal vein is estimated from a given region is estimated as:

$$I_{381m,k} = 0.38\, I_{385}\, X_{b17,6,m,k}\, X_{18r,m}\, I_{244j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \,/\, \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

(App. B.,I, No. 264)

where,

$$X_{b17,6,m,k} = 1.4\, X_{16,m,k}$$

(App. B.,V, No. 38)

and,

$$I_{389k,j} = \sum_{i=1}^{I_{69}} I_{239bi,j,k} \qquad 0 < I_{240b} \leq 3$$

(App. B.,I, No. 265)

$$I_{390k,j} = \sum_{i=1}^{I_{69}} I_{239bi,j,k} \qquad 0 < I_{240b} > 3$$

(App. B.,I, No. 266)

$$I_{391k,j} = \sum_{k=1}^{I_{274}} I_{241k,j} \qquad I_{241k,j} > 0$$

(App. B.,I, No. 267)

I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region

$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region

$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region

$I_{381m,k}$ = transport of contaminant from the N-P, and upper T-B regions of the respiratory tract to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)
X_{18r} = regional particulate concentration, m^{-3}, for a given contaminant
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$X_{b17,6,m,k}$ = inhalation rate, $m^3\ day^{-1}$, for the mth age group of the kth animal type

$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population
$I_{239bi,j,k}$ = the fraction of the animal's diet contributed by ith plant in the jth region for the kth animal population (input)
I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)
I_{274} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannabalism is being modeled) in a given region (input)
$I_{241,k,j}$ = initial diet fraction of the kth animal prey type in the food of a specific animal predator type from the jth region (input)

An illustration of particulate and gaseous transport of contaminants to the GI tract and directly to the blood is presented in Fig. 20. As described above ,an estimated 38 percent of resuspendable particulates inhaled by a given animal is transported directly to the gut on a daily basis, and the amount actually transferring to the blood plasma from the gut is dependent on the gut-to-blood fractional transport rate of a given contaminant; the remaining amount is lost through feces production.

The transport of resuspendable particulates in ambient air to the bronchus, bronchiole, and pulmonary regions of the lung is modeled as a regionally weighted estimate:

$$F(X_{18r}, X_{1e17,6}) = I_{392}\, X_{18r}\, X_{b17,6,m,k}\, I_{244j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \Big/ \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

(App. A., No. 100)

$$F(X_{18r}, X_{1e17,6}) = I_{393}\, X_{18r}\, X_{b17,6,m,k}\, I_{244j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \Big/ \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

(App. A., No. 101)

$$F(X_{18r}, X_{1e17,6}) = I_{394}\, X_{18r}\, X_{b17,6,m,k}\, I_{244j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \Big/ \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

(App. A., No. 102)

$X_{b17,6,m,k}$ = inhalation rate, $m^3\ day^{-1}$, for the mth age group of the kth animal type
$X_{18r,m}$ = regional particulate concentration, m^{-3}, for the mth contaminant
$X_{1e17,6,n}$ = nth contaminant content on the surface of the bronchus region of the lung lumen as particulates
$X_{2e17,6,n}$ = nth contaminant content on the surface of the bronchiole region of the lung lumen as particulates
$X_{3e17,6,n}$ = nth contaminant content on the surface of the pulmonary region of the lung lumen as particulates
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region

$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region

I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

I_{273} = number of regions simulated (input)

I_{392} = deposition fraction of resuspendable air particulates in the bronchus region of the lung, day^{-1} (input, default = 0.1)

I_{393} = deposition fraction of resuspendable air particulates in the bronchiole region of the lung, day^{-1} (input, default=0.1)

I_{394} = deposition fraction of resuspendable air particulates in the pulmonary region of the lung, day^{-1} (input, default=0.05)

7.3.3. Inhaled Volatiles

The exchange of a volatile or semi-volatile contaminant between aveolar air and the atmosphere depends on the relative concentrations of the specific contaminant in both media, and on the partition coefficient between blood plasma in the aveolar tissues and ambient air. The regionally weighted exchange is estimated as:

$$F(fV_{a,g}, X_{3e17,6}) = \{[X_{b17,6,m,k} X_{18g} + X_{fl7,1} X_{17,1} / (5.5e\text{-}05\,(1000\, X_{14,m,k})^{0.99})] / (X_{fl7,1} + 1000\, X_{b17,6,m,k} / X_{r17,1,n})\}\, 5.5e\text{-}05\,(1000\, X_{14,m,k})^{0.99}\, I_{244j}\,(I_{389k,j} + I_{390k,j} + I_{391k,j}) / \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

(App. A., No. 99)

$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region

$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region

$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region

I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

I_{273} = number of regions simulated (input)

$X_{b17,6,m,k}$ = inhalation rate, $m^3\ day^{-1}$, for the mth age group of the kth animal type

X_{18g} = regional vapor concentration, m^{-3}, for a given contaminant

$X_{fl7,1}$ = total plasma flow, liters hr^{-1}

$X_{17,1}$ = contaminant content of blood plasma, mass or activity units

$X_{r17,1,n}$ = partition coefficient for the nth contaminant between plasma and ambient air in aveolar tissue (input)

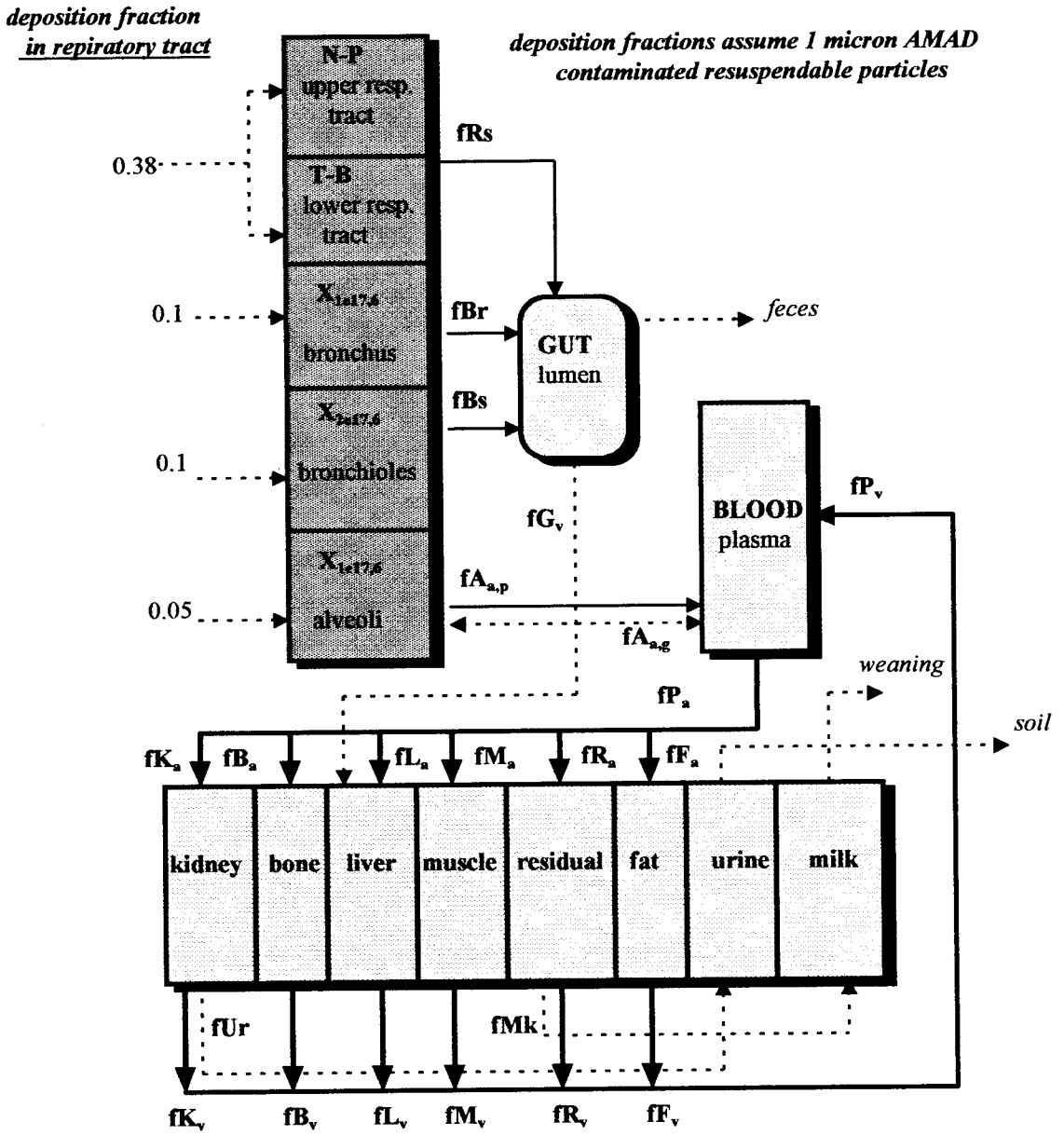
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

Thus, in cases where a contaminant is a volatile or semi-volatile substance, both particulate transport and vapor exchange transport are active in the model simultaneously, however, the exchange of vapor between contaminated particles on the lung surface and ambient air or blood plasma in the alveoli is not modeled in BIOTRAN.2. Finally, the net transport from this exchange can result in either positive or negative gain from the blood plasma depending on the relative concentrations in ambient air and blood plasma in relation to the plasma-ambient air partition coefficient for a given contaminant.

7.3.4. Soil Ingestion

The regionally weighted transport of a given contaminant from ingested food is estimated as:

$$I_{382m,k} = 1000\, F_{57}\, [(1 - I_{275ka}) (\sum_{j=1}^{I_{273}} X_{r13t,j}\, I_{244j}\, I_{389k,j} + \sum_{j=1}^{I_{273}} \sum_{i=1}^{I_{69}} X_{s13t,i,j}\, I_{390k,j}\, I_{239bk,i,j})$$



Legend:

fP_a, fP_v, fK_a, fK_v, fB_a, fB_v, fL_a, fL_v, fM_a, fM_v, fR_a, fR_v, fF_a, fF_v, flow to and from designated organs of blood plasma (fP_a, fP_v), liters hr^{-1}

$fA_{a,p}$, $fA_{a,g}$ = fraction of contaminant, day^{-1}, transported/exchanged into arterial plasma from the pulmonary region of the lung surface from particulates and vapors

$X_{1e17,6}$, $X_{2e17,6}$, $X_{3e17,6}$ = surface contamination on bronchus, bronchiole, and pulmonary regions of the lung surface, mass or activity units

fRs, fBr, fBs = fraction of contaminant transferred from the naso-phyrangeal, bronchus, and bronchiole regions of the lung surface, day^{-1}

fG_v = hepatic-portal vein shunt flow of plasma from the gut lumen to the liver, hr^{-1}

fUr, fMk = fraction of contaminant transported to urine and milk, hr^{-1}

Fig. 20

Transport of Contaminants From the Lung Surface to Organs and Tissues In BIOTRAN.2

$$+ I_{275ka} \sum_{j=1}^{I_{273}} \sum_{l=1}^{I_{274l}} \sum_{m=1}^{I_{248l}} X_{p17,j,m,l}\, I_{395j,m,l} \;] \qquad \text{(App. B.,I, No. 268)}$$

$I_{382m,k}$ = regionally weighted contaminant transport from ingested food to the mth age group of the kth mammalian predator population

I_{248l} = lifespan, years, of the an animal in the lth mammalian prey population (input)

$X_{r13t,j}$ = contaminant concentration, gdwt-1, in and on consumed food from range from the jth region

$X_{s13t,j}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed supplementary feed from the jth region

$X_{p17,j,m,l}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed food for the mth age group from the lth prey of the jth region

F_{57} = transport of food, kgdwt/day, to an animal of a given age group for a given mammalian population

I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population

I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from the jth region

$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from the jth region

$I_{239bk,i,j}$ = the fraction of the animal's diet contributed by ith plant in the jth region for the kth animal population

I_{69} = total number of plants in a given region (input)

I_{273} = number of regions simulated (input)

I_{274l} = number of animals simulated: to be used as a potential food sources for the lth animal population except for the predator population itself in a given region

$I_{395j,m,l}$ = population reduction factor for the mth age group of prey the lth prey population from the jth region

The contaminant concentration in and on consumed range and supplementary feed plants are estimated as:

$$X_{ra13t} = \sum_{j=1}^{I_{273}} \sum_{n=1}^{3} I_{269j,j,k}\, X_{c13t,n,j} / I_{254} \qquad \text{(App. B.,V, No. 39)}$$

$$X_{sp13t,i,j} = I_{398i,j} / I_{399i,j} \qquad \text{(App. B.,V, No. 40)}$$

where,

$$X_{c13t} = \sum_{j=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{258i,j}\; I_{396i,j} + I_{259i,j}\, I_{397i,j} \qquad \text{App. B.,V, No. 41)}$$

and,

$$I_{396i,j} = [X_{13,5,1}+X_{r13,5,1}+X_{sa13,5,1}+X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1} + (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1})\, X_{5,1} / (X_{4,1}+ X_{5,1})] / [X_{5,1} + X_{6,1} + X_{4,1}\, X_{5,1} /(X_{4,1} + X_{5,1})] \qquad \text{(App. B.,I, No. 269)}$$

$$I_{397i,j} = [X_{13,5,2}+X_{r13,5,2}+X_{sa13,5,2} + (X_{13,4,2}+X_{r13,4,2}+X_{s13,4,2}\, X_{5,1} /(X_{4,1} + X_{5,1})] / [\, X_{5,2} + X_{4,2}\, X_{5,1} / (\, X_{5,1} +X_{4,1})] \qquad \text{(App. B.,I, No. 270)}$$

$$I_{398i,j} = (X_{5,1} / X_{5,1}+X_{4,1}) (X_{13,4,2}+X_{r13,4,2}+X_{s13,4,2}) + X_{13,5,2}+X_{r13,5,2}+X_{sa13,5,2} + X_{13,5,1}+ X_{r13,5,1} + X_{sa13,5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})^2] (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1}) \qquad I_{240b}= 4$$

(App. B.,I, No. 271)

$$= X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1} \qquad I_{240b}= 5$$

$$= [X_{13,5,1}+X_{r13,5,1}+X_{sa13,5,1}+ (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1}) X_{5,1} (X_{4,1}+ X_{5,1})] / [X_{5,1} + X_{6,1} + X_{4,1} X_{5,1} /(X_{4,1} + X_{5,1})] \qquad I_{240b}= 6$$

$$= X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1} \qquad I_{240b}= 7$$

and,

$$I_{399i,j} = [X_{4,2} X_{5,1} / (X_{5,1} + X_{4,1})] + X_{5,2} + X_{5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})] +X_{6,1} +X_{6,2}$$

$I_{240b}= 4$ (App. B.,I, No. 272)

$$= X_{6,1} \qquad I_{240b}=5$$

$$= [X_{4,2} X_{5,1} / (X_{5,1} + X_{4,1})] + X_{5,2} + X_{5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})] \qquad I_{240b}=6$$

$$= X_{6,2} \qquad I_{240b}=6$$

X_{ral3t} = contaminant concentration, $gdwt^{-1}$, in and on consumed food from range from all regions combined

$X_{spl3t,i,j}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed supplementary feed from the jth region

$X_{cl3t,}$ = concentration, $gdwt^{-1}$, of the range feed from all food categories for all plant types combined for a given mammalian population

$I_{258i,j}$ = live plant fraction (including seeds) ingested from the ith plant type from the jth region

I_{273} = number of regions simulated (input)

$I_{259i,j}$ = dry plant fraction (excluding seeds) ingested from the ith plant type from the jth region

$I_{396i,j}$ = concentration, $gdwt^{-1}$, of the live plant fraction of consumed range plants for the ith plant of the jth region

I_{69} = total number of plants in a given region (input)

$I_{397i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed range plants for the ith plant of the jth region

$I_{398i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed supplementary feed for the ith plant of the jth region

$I_{399i,j}$ = total biomass density, gdwt m^{-2}, for supplementary feed produced by the ith plant of the jth region

$I_{269j,j,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)

$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type

$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type

$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type

$X_{13,4,2}$ = internal contaminant concentration, m^{-2}, in dry stem of a given plant type

$X_{13,5,2}$ = internal contaminant concentration, m^{-2}, in dry leaves of a given plant type

$X_{13,6,2}$ = internal contaminant concentration, m^{-2}, in dry seeds of a given plant type

$X_{r13,5,1}$ = resuspendable contaminant concentration, m^{-2}, on live leaves

$X_{r13,5,2}$ = resuspendable contaminant concentration, m^{-2}, on dry leaves
$X_{r13,4,1}$ = resuspendable contaminant concentration, m^{-2}, on live standing vegetation
$X_{r13,4,2}$ = resuspendable contaminant concentration, m^{-2}, on dry standing vegetation
$X_{r13,6,1}$ = resuspendable contaminant concentration, m^{-2}, on seeds
$X_{r13,6,2}$ = resuspendable contaminant concentration, m^{-2}, on dry seeds
$X_{s13,5,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live leaves
$X_{s13,5,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry leaves
$X_{s13,4,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live standing vegetation
$X_{sa13,5,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry leaves for all plants in agiven region from saltation-creep-rainsplash events
$X_{sa13,4,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region from saltation-creep-rainsplash events
$X_{sa13,6,2}$ = weighted average contaminant concentration, m^{-2}, on the surface
$X_{s13,4,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry standing vegetation
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

The contaminant concentration in an on consumed prey from a given region is estimated as:

$$X_{p17,j,m,l} = X_{17,1j,m,l} + X_{17,2,j,m,l} + X_{17,3,j,m,l} + X_{17,4,j,m,l} + X_{17,10,j,m,l} + X_{17,12,j,m,l} + X_{1e17,6j,m,l} + X_{2e17,6j,m,l} + X_{3e17,6,j,m,l} + X_{17,9j,m,l} + X_{17,13,j,m,l} + X_{17,14j,m,l} \quad \text{(App. B.,V, No. 42)}$$

$$= X_{17,1j,m,l} + X_{17,2,j,m,l} + X_{17,3,j,m,l} + X_{17,4,j,m,l} + X_{17,10,j,m,l} + X_{17,12,j,m,l} + X_{1e17,6j,m,l} + X_{2e17,6j,m,l} + X_{3e17,6,j,m,l} + 0.1X_{17,9j,m,l} + 0.1X_{17,14j,m,l} \qquad X_{14m,l} > 2$$

where,

$$I_{395j,m,l} = I_{246m,l}\ I_{317m,k,l} / [\ (I_{300m,1,k} + I_{300m,2,k}\ )\ X_{14,m,l}\ ] \quad \text{(App. B.,I, No. 273)}$$

$I_{395j,m,l}$ = population reduction factor for the mth age group of prey the lth prey population from the jth region
$X_{p17,j,m,l}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed food of the mth age group from the lth prey of the jth region
$X_{14,m,l}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the lth mammalian population

$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population
$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population
$I_{246m,l}$ = fraction of the population in the mth age group of the lth animal population
$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region
$X_{1e17,6,j,m,l}$ = contaminant content on the surface of the bronchus region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population
$X_{2e17,6,j,m,l}$ = contaminant content on the surface of the bronchiole region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population
$X_{3e17,6,j,m,l}$ = contaminant content on the surface of the pulmonary region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population

$X_{17,2,j,m,l}$ = contaminant content of muscle tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,1,j,m,l}$ = contaminant content of blood plasma, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,3,j,m,l}$ = contaminant content of adipose tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,4,j,m,l}$ = contaminant content of residual tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,10,j,m,l}$ = contaminant content of kidney tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,9,i,m,l}$ = contaminant content of bone tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,12,n}$ = contaminant content of liver tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,13,j,m,l}$ = contaminant content of gut ingesta, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

The contaminant content of gut ingesta is estimated as:

$$X_{17,13,j,m,l} = I_{379} / I_{385} \qquad \text{(App. B.,V, No. 43)}$$

and the contaminant content of the pelt is estimated to be one-third the content of the top soil layer on a gdwt basis [61] :

$$X_{17,14j,m,l} = 50\, X_{14,m,l} \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})\, I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\, I_{265i,j} / \sum_{i=1}^{I_{69}} I_{265i,j}\,] \qquad \text{(App. B.,V, No. 44)}$$

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,13,j,m,l}$ = contaminant content of gut ingesta, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)

I_{379} = total transport of contaminant, day^{-1}, from the gut to the liver via the hepatic-vein

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{14,m,l}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the lth mammalian population

$X_{12,i,1,j}$ = top soil layer contaminant concentration, $gdwt^{-1}$, for the ith plant of the jth region

$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)

$I_{265i,j}$ = coverage area, acres, of the ith plant type for the jth region (input)

I_{69} = total number of plants in a given region (input)

I_{273} = number of regions simulated (input)

The amount of soil ingested by grazing animals may range anywhere from two to almost 30 percent of the food intake on a gdwt basis [62] . The modeling of this process in BIOTRAN.2 is made dependent on the amount of crown cover present on range [63] :

$$I_{383m,k} = 3\, X_{14,m,k}\, I_{385} \sum_{j=1}^{I_{273}} I_{389k,j}\, I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\;\; I_{265i,j} \,/\, \sum_{i=1}^{I_{69}} I_{265i,j}\;] \qquad \text{(App. B.,I, No. 274)}$$

$$= (3\, X_{14,m,k} \,/\, I_{400})\, I_{385} \sum_{j=1}^{I_{273}} I_{389k,j}\, I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\;\; I_{265i,j} \,/\, \sum_{i=1}^{I_{69}} I_{265i,j}\;] \qquad I_{400} > 20$$

where,

$$I_{400} = \sum_{j=1}^{I_{273}} I_{389k,j}\, I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\;\; I_{265i,j} \,/\, \sum_{i=1}^{I_{69}} I_{265i,j}\;] \qquad \text{(App. B.,I, No. 275)}$$

$I_{383m,k}$ = amount of soil consumed, gdwt day^{-1}, by an an animal from grazing
I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)
$X_{12,i,1,j}$ = top soil layer contaminant concentration, gdwt^{-1} , for the ith plant of the jth region
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from the jth region
$I_{265i,j}$ = coverage area, acres, of the ith plant type for the jth region (input)
$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
I_{400} = sum of all available feed on range, gdwt m^{-2}, from all regions combined based on food utilization factors in each region

7.3.5. Water Ingestion

The amount of contaminant ingested from drinking water is based on metabolic needs as well as the concentration of contaminant in that media:

$$I_{384m,k} = I_{385}\; I_{401m,k} \sum_{j=1}^{I_{273}} I_{402k,j}\, I_{244k,j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \qquad \text{(App. B.,I, No. 276)}$$

$I_{384m,k}$ = amount of contaminant, day-1, transported to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population
I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)
$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region
$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region
$I_{401m,k}$ = net water requirement, liters day^{-1}, for the mth age group of the kth mammalian population
$I_{402k,j}$ = contaminant concentration, liter-1, in the drinking water of the kth mammal population in the jth region (input)

where,

$$I_{401m,k} = 0.04\, X_{14,m,k} - 0.522\, I_{403m,k} - 0.3\, I_{295kr}\, F_{57} \qquad \text{(App. B.,I, No. 277)}$$

$I_{401m,k}$ = net water requirement, liters day^{-1}, for the mth age group of the kth mammalian population

$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

F_{57} = transport of food, kgdwt/day, to an animal of a given age group for a given mammalian population

I_{295kr} = weighted nitrogen fraction from range plant food consumption for the kth mammalian population

and,

$$I_{403m,k} = 70\, X_{15m,k}\, I_{249k}\, I_{250k} \,/\, 4500 \qquad \text{(App. B.,I, No. 278)}$$

$I_{403m,k}$ = metabolic heat production losses expressed in kgdwt day^{-1}

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population

I_{250k} = activity factor for the kth mammalian population (input)

8.0. BIOMASS and CONTAMINANT FLOW

8.1. Fetal Mammals

Biomass flow in weaning mammals is modeled identically to the flow of biomass in other age groups with the exception that milk is used as a food source during this period in the life-cycle. However, the flow of biomass in the mammalian fetus is based on the following relationship [35] :

$$X_{ft14k} = 0.05256\, I_{277k} \{ 1 - \exp[-9 (\sum_{m=1}^{I_{307k}} m \,/\, I_{307k})] \} \qquad \text{(App. B.,V, No. 45)}$$

X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population

I_{277k} = maximum body weight, kgfwt, for an animal in the kth mammalian population (input)

I_{307k} = estimated gestation period, days, for the kth mammalian population

This relationship produces a fetal weight of about five percent of the maximum body weight of the specific animal at the end of the gestation period. The organ and tissue weights, perfusion rates, plasma volumes are estimated as for other age groups already described, however, the pharmacokinetic treatment of contaminant flow has some varaitions. The contaminant concentration in fetal blood plasma is assumed to be in equilibrium with the concentration in parent blood plasma :

$$X_{ft17,1k} = 0.2\, X_{ft14k}\, X_{ftc17,1,k} \qquad \text{(App. B.,V, No. 46)}$$

where,

$$X_{ftc17,1,k} = (\sum_{m=2}^{I_{248k}} X_{fl7,4,m,k}\, X_{17,1,m,k} \,/\, [X_{v17,1,m,k}\, X_{fl7,1,m,k} \{ \sum_{i=1}^{8} I_{322k,i} \,/\, \sum_{l=1}^{8} I_{322k,l} \}]) + (X_{fl7,4,1,k}\, X_{17,1,1,k} \,/$$

$$[X_{v17,1,1,k}\, X_{fl7,1,1,k} \{ \sum_{i=1}^{8} I_{322k,i} \,/\, \sum_{l=1}^{8} I_{322k,l} \} \{ \sum_{j=1}^{3} I_{326bkj} \,/\, \sum_{n=1}^{3} I_{326bkn} \}])$$

$$0.15\, I_{248k} \le m \le 0.75\, I_{248k} \,;\; 0.15\, I_{248k} \le I_{312bkn} \le 0.75\, I_{248k} \qquad \text{(App. B.,V, No. 47)}$$

$X_{ft17,1,k}$ = contaminant content, mass or activity units, in fetus blood plasma of the kth mammalian population for a given contaminant

$X_{17,1,m,k}$ = contaminant content of blood plasma, mass or activity units, for the mth age group of reproductive females for a given contaminant

$X_{ftc17,1,k}$ = weighted contaminant concentration, liter-1, in the fetus blood plasma of the kth mammalian population for a given contaminant

X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population

$X_{fl7,4,m,k}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue, of the mth age group of reproductive females of the kth mammalian population

$X_{vl7,1,m,k}$ = volume of blood plasma, liters, of the mth age group of reproductive females of the kth mammalian population

$X_{fl7,1,m,k}$ = total plasma flow, liters hr^{-1}, of the mth age group of reproductive females of the kth mammalian population

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

$I_{322k,l}$ = fraction of the reproductive females in the *l*th reproductive age grouping for the kth mammalian population

$I_{312k,n}$ = age, days, of newborns from the nth gestation period during a given year for the kth mammalian population

$I_{326bk,n}$ = number of female animals in the nth sub-group of the first age group for the kth mammalian population

This relationship takes into account the average blood concentration of all reproductive females in the population to estimate a weighted average blood transport to the fetus. In cases where the reproductive age is reached before the first year of age such as with small mammals, then the same methodology must be applied to estimate the number in three age groups defined earlier. The transport of contaminant from the mother to the fetus assumes that the plasma compartment of the latter is about twenty percent of the mass of the fetus as a whole as shown above.

8.2. Weaning Mammals

The transport of contaminant to mammals in the weaning stage is estimated similarly to transport during the fetal stages assuming milk consumption at ten percent of body weight, and placental transport rate of contaminant is equal to rate of transport from the blood to the mammary gland:

$$X_{wnl7,1k} = 0.1\, X_{14m,k}\, X_{wncl7,1k} \qquad m=1;\ I_{304k,n} \le I_{313k,n} \le I_{304k,n} + 0.025\, I_{248k} \qquad \text{(App. B.,V, No. 48)}$$

where,

$$X_{wncl7,1,k} = X_{ftcl7,1,k} \qquad \text{(App. B.,V, No. 49)}$$

X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population

$X_{wnl7,1k}$ = contaminant transport, mass or activity units day^{-1}, from mother's milk to a weaning newborn animal of the kth mammalian population

$X_{ftcl7,1,k}$ = weighted contaminant concentration, liter-1, in the fetus blood plasma of the kth mammalian population for a given contaminant

$X_{wncl7,1,k}$ = weighted contaminant concentration, liter-1, in weaning milk of the kth mammalian population for a given contaminant

$I_{304k,n}$ = time delay, days, for newborns from one litter to the next for the nth litter

$I_{313k,n}$ = fetal age, days, of the nth gestation period during a given year for the kth mammalian population

9.0. REFERENCES

1. Gallegos, A.F.. Garcia, B.J., and C.M. Sutton, 1980. Documentation of transuranium (TRU) biological transport model (BIOTRAN). Informal Report. Los Alamos Scientific Laboratory , pp. 200. (LA-8213-MS).

2. Gallegos, A.F., and W.J. Wenzel, 1984. HUMTRN: Documentation and verification for an ICRP-based age and sex specific human simulation model for radionuclide dose assessment. Los Alamos National Laboratory , pp. 83. (LA-9994-MS).

3. Wenzel, W.J., and A.F. Gallegos, 1985. EFFECTS : Documentation and verification for a BEIR III cancer risk model based on age, sex, and population dynamics for BIOTRAN. Los Alamos National Laboratory, pp. 100. (LA-10371-MS).

4. Gallegos, A.F., and W.J. Wenzel, 1990. Runoff and sediment yield model for predicting nuclide transport in watersheds using BIOTRAN. Los Alamos National Laboratory, pp. 118. (LA-11851-MS).

5. Wenzel, W.J., Fox, T.S., Gallegos, A.F., Tierney, G., and J.C. Rodgers, 1987. Cs-137, Pu-239/240, total uranium, and scandium in trees and shrubs growing in transuranic waste at Area B, Los Alamos National Laboratory, pp. 62. (LA-11126-MS).

6. Wenzel, W.J., and A.F. Gallegos, 1982. Supplementary documentation for an Environmental Impact Statement regarding the Pantex Plant: long-term radiological risk assessment for postulated accidents, Los Alamos National Laboratory, pp. 75. (LA-9445-PNTX-O).

7. Wenzel, W.J., Wallwork-Barber, K.M., Horton, J.M., Rea, K.H., Hollis, L.C., Gladney, E.S., Mayfield, D.L., Gallegos, A.F., Rodgers, J.C., Thomas, R.G., and G. Trujillo, 1982. Supplementary documentation for an Environmental Impact Statement regarding the Pantex Plant: agricultural food chain radiological assessment, Los Alamos National Laboratory, pp. 50. (LA-9445-PNTX-M).

8. Wenzel, W.J., Wallwork-Barber, K.M., Rodgers, J.C., and A.F. Gallegos, 1982. The Texas panhandle soil-crop-beef food chain for uranium: a dynamic model validated by experimental data, Proceedings of the Fourth DOE Environmental Protection Information Meeting, CONF-821215, Report 6D, pp. 311-321, Denver, Colorado.

9. Wenzel, W.J., Gallegos, A.F., and W.R. Hansen. 1982. Dynamic modeling of agricultural pathways in three terrestrial ecosystems for plutonium and uranium, Health Physics Society Meeting, Las Vegas, Nevada, July.

10. U.S. Department of Energy (DOE), 1983. Final Environmental Impact Statement, Pantex Plant Site, Amarillo, Texas, (DOE/EIS-0098).

11. Gallegos, A.F., and W.J. Wenzel, 1992. BIOTRAN users manual, Los Alamos National Laboratory, pp.277. (unpublished manuscript)

12. Gallegos, A.F., and W.J. Wenzel, 1989. HUMTRN and EFFECTS: age and sex specific dosimetric and physiological human population dynamics models for dose assessment, Health Physics, Special Supplement, Modeling for Scaling to Man, Vol. 57, Supplement 1. pp. 443-453.

13. Semiannual Report. 1975. Transuranic solid waste management research programs, Los Alamos Scientific Laboratory, July - December, 1974. (LA-6100-PR).

14. Semiannual Report. 1976. Transuranic solid waste management research programs, Los Alamos Scientific Laboratory, January - June, 1975. (LA-6264-PR.).

15. Semiannual Report. 1976. Transuranic soiled waste management research programs, Los Alamos Scientific Laboratory, July - December, 1975. (LA-6481-PR).

16. Wheeler, M.L., Smith, W.J., and A.F. Gallegos. 1977. A preliminary evaluation of the potential for plutonium release from burial grounds at Los Alamos Scientific Laboratories, Los Alamos Scientific Laboratory. (LA-6694-MS).

17. Gallegos, A.F., Wheeler, M.L., Whitty, W.J., and W.J. Smith. 1975. Environmental evaluation of solid radioactive waste disposal areas. Bulletin, New Mexico Academy of Science 15 (2) (abstract).

18. Gallegos, A.F., and L.J. Johnson. 1976. A simulation study of Mineral recycling in Ponderosa Pine forests, Proceedings Supplement, J. Arizona Academy of Sciences (11) (abstract)

19. Smith, W.J., Gallegos, A.F., and L.J. Johnson. 1977. LASL models for environmental transport of radionuclides. Modeling and Simulation, Instrument Society of America (8).

20. Gallegos, A.F., Smith, W.J, and L.J. Johnson. 1979. LASL models for environmental transport of radionuclides in forests. Modeling and Simulation, Instrument Society of America (9).

21. Knisel, W.G., Ed.. 1980. CREAMS: A field-scale model for chemicals, runoff, and erosion from agricultural Management systems., U.S. Dept. of Agric., Conservation Research Report (26).

22. Hillel, D., and C.H.M. Von Bavel. 1976. Simulation of profile water storage as related to soil hydraulic studies. Soil Science Journal (40:6), pp. 807-815.

23 Running, S.W., Waring, R.H., and R.A. Rydell. 1975. Physiological control of water flux in conifers. Oecologia (Berl.) (18). pp. 1-16.

24 Foxx, T.S., Tierney, G.D., and J.M. Williams. 1984. Rooting depths of plants relative to biological and environmental factors, Los Alamos National Laboratory. (LA-10254-MS).

25 Branson, F.A. et. al., 1981. Rangeland hydrology. 2nd ed., Society for Range Management, Denver, Colorado, Kendall/Hunt Publishing Co., Dubuque, Iowa.

26 Hann, C.T., et. al., Eds., 1982. Hydrologic modeling of small watersheds. American Society of Agricultural Engineers, St. Joseph, Michigan.

27 Lane, L.J., et. al.. 1985. New estimating procedures for surface runoff, sediment yield and contaminant transport in Los Alamos County, New Mexico. Los Alamos National Laboratory, Los Alamos, New Mexico. (LA-10335-MS).

28 Baltz, E.H., Abrahams, J.H., and W.D. Purtyman. 1963. Preliminary report on the geology and hydrology of Mortandad Canyon near Los Alamos, New Mexico with reference to the disposal of low-level radioactive waste. USGS in cooperation with USAEC and LASL. Open File Report, Los Alamos, New Mexico.

29 Wight, J.R., et al. 1982. SPUR - simulation of production and utilization of rangelands: a rangeland model for management and research. USDA, Agricultural Research Service, Miscellaneous Pub. 1431.

30 Nyhan, J.W., and L.J. Lane. 1986. Erosion control technology: a user's guide to the use of the universal soil loss equation at waste burial facilities. Los Alamos National Laboratory, Los Alamos, New Mexico. (LA-10262-M).

31 Edmonds, R.L., Ed. 1974. An initial synthesis of results in the Coniferous Forest Biome, 1970-1973, Coniferous Forest Biome, Ecosystems Analysis Studies, US/IBP, Bulletin No. 5.

32 Gilbert, B.J., 1974. Range IV Grassland Simulation Model, regional analysis of grassland environmental systems, Federal Grant No. ESR-72-033396-A02, Colorado State University, Ft. Collins, CO, Report No. 17-18.

33 Precht, H., Christophersen, J., Hensen, H., and W. Larcher. 1973. Temperature and life. Springer-Verlag , New York.

34 McGinnies, W.G., and J.F. Arnold. 1939. Relative water requirement of Arizona range plants. Arizona Agricultural Experiment Station. Bulletin 80, 167-246.

35 Cole, G.W., et al.. 1976. ELM, Version 2.0. Colorado State University Range Sciences Department. Science Series 20. Colorado State University, Fort Collins, Colorado.

36 Sauer, R.H. 1975. Simulation model for grassland primary producer phenology and biomass dynamics. IBP Grasslands Biome, Ecosystems Analysis Studies, Preprint No. 152.

37 Helseth, F. 1968. The effect of oxygen Tension and Transpiration on uptake of calcium in intact plants. Ph. D. Dissertation, University of Georgia, Athens Georgia.

38 Botkin, D. B., Janak, J.F., and J.R. Wallace. 1972. Some ecological consequences of a computer model of forest growth. Journal of Ecology, 60. pp 849-872.

39 Horn, H.S. 1975. Markovian properties of forest succession. In: M.L. Cody and J.M. Diamond (eds), Ecology and evolution of communities. Harvard Universe Press. Cambridge, Mass.

40 Anderson, M.C. 1966. Ecological groupings of plants. Nature (Lond) 212. pp 54-56.

41 Gallegos, A.F., Garcia, B.J., and C.M. Sutton. 1978. Documentation of transuranic biological transport model BIOTRAN. Los Alamos National Laboratory Report . (unpublished manuscript). Los Alamos, NM.

42 Larson, F.R., 1975. Simulating growth and management of Ponderosa Pine stands, Ph.D Dissertation. Colorado State University, Ft. Collins, CO.

43 Garcia, B.J., 1977. Selected characteristics of pinyon pine. Masters Thesis. New Mexico Highlands University, Las Vegas, NM

4 Slade, D.H., 1968. Meteorology and atomic energy, USAEC report TID-24190

45 EPA (U.S. Environmental Protection Agency), 1988. Superfund Exposure Assessment Manual, Environmental Protection Agency, Office of Remedial Response, EPA 1540/1-88/001.

46 Moore, R.E.,et al., 1979. AIRDOS-EPA: A computerized methodology for estimating environmental concentrations and doses to man from airborne releases of radionuclides. Oakridge National Laboratory, Oakridge, Tennessee (NTIS), DOE report, ORNL 5332.

47 Dreicer, M., Hakonson, T.E., White, G.C., and F.W. Whicker. 1984. Rainsplash as a mechanism for soil contamination of plant surfaces. Health Physics (46), 177-187.

48 Smith, W.J., Whicker, F.W., and H.R. Meyer. 1982. Review and categorization of siltation, suspension, and resuspension models. Nuclear Safety (23), 685-699.

49 Gallegos, A.F.. 1978. Preliminary model of plutonium transport by wind at Trinity Site. In: Selected environmental plutonium research reports. USDOE, NVO-192, 681-695.

50 Paul, E. A., Campbell, C.A., Rennie, D.A., and K.J. McCallum. 1964. Investigations of the dynamics of soil humus utilizing carbon dating techniques. 8th International Congress Soil Science, Bucharest Comm. II. (3), 201-208.

51 Straub, C.P.., 1963. Low-level radioactive wastes. TID, U.S. Atomic Energy Commission, U.S. Government Printing Office, Washington, DC.

52. Baes, C.F., III, Sharp, R.D., Sjoreen, A.L., And R.W. Shor. 1984. A review and analysis of parameters for assessing transport of environmentally released radionuclides through agriculture. Oakridge National Laboratory, Oakridge, Tennessee. ORNL 5786.

53 Kramer, P.J.. 1969. Plant and soil water relationships: A modern synthesis. McGraw Hill Book Conpany, New York, NY. 482pp.

54 Sutcliffe, J.F., and D.A. Baker. 1974. Plants and mineral salts studies. In: Studies in Biology No. 48. Edward Arnold, Publisher. 60pp.

55 Crafts, A.S.. 1961. Translocation in plants. Biology Studies. Holt, Rhinehart and Winston, New York, NY. 182pp.

56 Wallace, A.. 1963. Solute uptake by intact plants. Edward Brothers Inc., Ann Arbor, Michigan. 178pp.

57 Mordenti, J.. 1986. Man versus beast: Pharmacokinetic Scaling in Mammals. Journal of Pharmaceutical Sciences (75), No. 11. 1028-1040.

58 Adolph, E.F.. 1949. Quantitative relations in the physiological constitution of mammals. Science (109), 579-585.

59 Sheppard, C.W. 1962. Basic principles of the tracer method. John Wiley & Sons, Inc., New York. 282pp.

60 International Commission on Radiological Health (ICRP). 1975. Report of the Task Force Group on Reference Man, ICRP-23. Pergamon Press, New York, NY.

61 Miera, F.R., Hansen, W.C., Gladney, E.S., and P. Jose. 1980. Mobility of elevated levels of uranium in the environment. In: Natural Radiation Environment III, Vol. 1., DOE Symposium Series 51. NTIS, Springfield, Virginia. 681-699.

62 Beyer, W.N., Conner, E.E., and S. Gerould. 1994. Estimates of soil ingestion by wildlife. Journal of Wildlife Management (58), 375-382.

63 Martin, W.E., Bloom, S.G., and R.J. Yarde. 1975. Plutonium transport and dose estimation model, the dynamics of plutonium in desert environments. NAEG Plutonium Study Modeling Group, Nevada Applied Ecology Group. USAEC Progress Report, NVO-153

10.0. APPENDIX A

FLOWS

I. Individual Flows

1. $F(P_p, X_{1,1}) = F(P_p, X_2) = f(D_1, I_1, S_1, I_6, S_2, S_3, I_4)$

$F(P_p, X_{1,1})$ = transport of moisture from rain events to soil, mm/day
$F(P_p, X_2)$ = transport of moisture from melting snow to soil, mm/day
D_1 = mean annual precipitation, inches
I_1 = January through December monthly precipitation fractions
S_1 = normalized random number fraction obtained from a normal distribution ($m = 0, \sigma = 1$) to assign precipitation amount to a given precipitation event for a given month (see App. B, IV, no.1)
I_6 = coefficient of variation for precipitation per event of reference site
S_2 = product of random number drawn from an even distribution (range 0 to 1) with the number of days in a given month to make a precipitation assignment (see App. B, IV, no.2)
S_3 = random number drawn from a normal distribution ($m = 0, \sigma = 1$) to estimate precipitation increment/decrement for a given rainfall event
I_4 = mean number of rainfall events in a given month totaling more than 0.1 inches

2. $F(P_p, X_{1,1}) = D_1 I_1 S_1 (1 + I_6 S_3)$ $\qquad I_5 = S_2$

$F(P_p, X_{1,1}) = 0$ $\qquad I_5 \neq S_2$

I_5 = Julian day number

3. $F(X_2, X_{1,1}) = C_o D_2 + C_1$

$F(X_2, X_{1,1}) = X_2$ $\qquad C_o D_2 + C_1 > X_2$

$F(X_2, X_{1,1}) = 0$ $\qquad X_2 = 0$

$F(X_2, X_{1,1})$ = transport of moisture from snow to top layer of soil, mm/day
C_o = snowmelt regression coefficient, 11.28 mm $^\circ C^{-1}$
C_1 = snowmelt intercept, 1.1128 mm

4. $F(X_{1,1}, P_e) = I_{17a} \exp(-0.4 I_{19})$

$F(X_{,1,1}, P_e) = (X_{1,1} - 1) \exp(-0.4 I_{19})$ $\qquad I_{17a} > (X_{1,1} - 1)$

$F(X_{1,1}, P_e) = 0$ $\qquad (X_{1,1} - 1) \leq 0$

$F(X_{1,1}, P_e)$ = transport of moisture from soil surface (evaporation), mm/ m^2 / day
I_{17a} = I_{17} (PET) adjusted for transpiration losses and low soil moisture in top layer
I_{19} = leaf area index, m^2/m^2, (LAI)
I_1 = soil moisture in surface layer, mm

5.0 $F(X_5, P_t) = \sum_{i=2} [(X_{1,i} - I_{22}) / (I_{23} - I_{22})] (X_{1,i} - I_{22}) I_{21} I_{19}$

$F(X_5, P_t)$ = total transpiration, mm/m^2/ day, for a given plant type
I_{20} = number of soil layers specified for a given plant type (input)
X1,i = soil moisture content, mm, for the ith soil layer for a given plant type
I_{22} = soil moisture content, mm, for the ith soil layer at wilt point (input)
I_{23} = soil moisture content, mm, for the ith soil layer at holding capacity (input)
I_{19} = leaf area index, see above
I_{21} = root penetration fraction (0 - 1) of a given plant type into the ith soil layer

6.0 $F(X_{1,i}, X_3{:}X_4{:}X_5) = F(X_5, P_t)$

$F(X_{1,i}, X_3{:}X_4{:}X_5)$ = flow of moisture from the ith soil to the leaf surface via the plant roots and stems

7.0 $F(X_{1,i}, X_{1,i+1}) = [\, \overset{i}{I}_{30} + (\overset{i}{I}_{30} / 2)(\overset{i+1}{I}_{31} - \overset{i-1}{I}_{31})\,]\, I_{37}$ $\qquad i \neq 1$

$F(X_{1,i}, X_{1,i+1}) = [\, \overset{i}{I}_{30} + \overset{i}{I}_{30}\overset{i+1}{I}_{31} / 2\,]\, I_{37}$ $\qquad i = 1$

$F(X_{1,i}, X_{1,i+1})$ = transport of moisture from the ith soil layer to next successive layer

$\overset{i}{I}_{30}$ = weighted soil moisture conductivity (κ) , mm / day/ mm -soil, based on rock, sand, silt, and clay composition of the ith soil layer obtained by regression analysis of information in Hillel [22] as a function of soil volumetric wetness (θ), p = 0.01, r > 0.9.

$\overset{i}{I}_{37}$ = thickness of ith soil layer, mm

$\overset{i+1;\, i-1}{I}_{31}$ = weighted soil matrix potential (Ψ) , mm / mm-soil based on rock, sand, silt, and clay composition of the upper and lower layer adjacent to the ith soil layer obtained by regression of information in Hillel[22] as a function soil volumetric wetness (θ), p = 0.01, r > 0.9.

8.0 $F(X_{1,1}, Pr) = (P_p - 0.2I_{52})^2 / (P_p + 0.8I_{52})$ $\qquad P_p > 0.2I_{52}$
$= 0.0$ $\qquad P_p \leq 0.2I_{52}$

$F(X_{1,1}, P_r)$ = daily moisture runoff, mm, from the surface of soil
P_p = daily precipitation event, mm
I_{52} = weighted soil moisture parameter, mm

9.0 $F(X_{1,1}, P_i) = P_p$ $\qquad P_p \leq 0.2I_{52}$
$= P_p - P_r$ $\qquad P_p > 0.2I_{52}$

10. $F(X_{1,i}, P_l) = 2\, I_{59j}\, I_{41}\, I_{60}$ $\qquad I_{57} > 0;\ X_{1,i} > I_{58}$
$= I_{66i}$ $\qquad F(X_{1,i}, P_l) > I_{66i}$
$= 0$ $\qquad I_{57} = 0;\ X_{1,i} \leq I_{58}$

$F(X_{1,i}, P_l)$ = potential sub-surface return flow from the ith soil layer with a jth plant type, ft^3 / day
2 = conversion factor: 1 inch/hr = 2 feet/day
I_{59j} = crossectional area, ft^2, for return flow into a channel from the jth plant type
I_{41} = soil moisture conductivity, inches / hr, for region (input)
I_{60} = fractional slope of regional land surface (input)

I_{66i} = total water available in ith soil layer with a jth plant type above field capacity, ft^3

11. $\Sigma F_a (X_{1,i}, P_l) = \sum_{j=1}^{I_{69}} \sum_{i=2}^{I_{37}} F(X_{1,i}, P_l)$

$\Sigma F_a (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, ft^3 /day, into a receiving channel

I_{69} = total number of plants in a given region (input)

I_{37} = total number of soil layers for the jth plant in a given region (input)

$F(X_{1,i}, P_l)$ = sub-surface return flow from the ith soil layer with a jth plant type, ft^3 / day , (see App. A. No. 10)

12. $\Sigma F_b (X_{1,i}, P_l) = 12\, \Sigma F_a (X_{1,i}, P_l) / (43562\, I_{68})$

$\Sigma F_b (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, inches/ day, into a receiving channel

$\Sigma F_a (X_{1,i}, P_l)$ = total lateral or subsurface return flow from a given region, ft^3 /day, into a receiving channel

I_{68} = area, acres, of the given region (input)

13. $F(X_8, P_c) = \{ I_{70} + I_{71} [\Sigma_{j \neq Pc} (P_{c,j}) + P_m + P_a] + I_{72} (P_r + P_l) / I_{63} \}\, 43654$

$F(X_8, P_c)$ = daily flow or transmission rate, ft^3 , from a given channel or channel segment

X_8 = amount of moisture, ft^3, in a given channel or channel segment

P_c = flow of moisture, acre-ft, from a given channel or channel segment

I_{70} = regression intercept, acre-ft, for a given channel or channel segment

I_{71} = regression slope factor for a given channel or channel segment

$P_{c,j}$ = flow of moisture, acre-ft, from jth connecting upslope channel or channel segment

P_m = flow of moisture from icemelt, acre-ft, from a given channel or channel segment

P_a = flow of moisture from underlying aquifer, acre-ft, from below a given channel or channel segment (value = 0.0 if aquifer not present under channel or channel segment)

I_{72} = lateral flow parameter for transmission, mi , for the given channel or channel segment

P_r = return flow, acre-inch, entering a given channel or channel segment

P_l = lateral flow, acre-inch, entering a given channel or channel segment (value = 0.0 if aquifer present under channel or channel segment)

I_{63} = length of a given channel or channel segment , miles (input)

14. $F(P_t, X_9) = 304.8\, [(P_r + P_l + \Sigma_{j \neq P_c} (P_{c,j}) + P_a) / I_{80}]$ $\quad D_2 < 0°\,C$

$= 0$ $\quad D_2 \geq 0°\,C$

$F(P_t, X_9)$ = ice thickness, mm, deposited on given channel or channel segment on a given day

P_t = total amount of moisture added from all sources to a given channel or channel segment, ft^3, on a given day

X_9 = a given channel or channel segment moisture content, ft^3, stored as ice

P_r = return flow, ft^3, entering a given channel or channel segment

P_l = lateral flow, ft^3, entering a given channel or channel segment (value = 0.0 if aquifer present)

$P_{c,j}$ = flow of moisture, ft^3, from jth connecting upslope channel or channel segment

P_c = flow of moisture, ft^3, from a given channel or channel segment

P_a = flow of moisture from underlying aquifer, ft^3, from below a given channel or channel segment (value = 0.0 if aquifer not present under channel or channel segment)
I_{80} = surface area of a given channel or channel segment, ft^2

15. $F(X_9, X_8) = (1.11280 + 11.28\, D_4)\, I_{80} / 304.8$ — $D_4 > 0°\,C$
$= X_9$ — $F(X_9, X_8) > X_9$
$= 0$ — $D_4 \leq 0°\,C$
$= 0$ — $X_{9,i} = 0$

$F(X_9, X_8)$ = flow of moisture, ft^3, from channel icemelt to a given channel or channel segment on a given day
X_8 = amount of moisture, ft^3, in a given channel or channel segment
X_9 = a given channel or channel segment moisture content, ft^3, stored as ice
I_{80} = surface area of a given channel or channel segment, ft^2
D_4 = mean daily drainage temperature, ° C, of a given channel or channel segment

16. $F(X_{10i-1}, X_{10i}) = 2\, I_{105}\, I_{41}\, I_{60}$

$F(x_{10i-1}, X_{10i})$ = flow of moisture, ft^3 /day, from (i-1)th up slope aquifer compartment to ith down slope aquifer compartment
x_{10i-1} = aquifer moisture content, ft^3, of (i-1)th up slope compartment
X_{10i} = aquifer moisture content , ft^3, of ith down slope compartment
I_{41} = soil moisture infiltration rate, inches/ hour (input)
I_{60} = fractional slope of regional land surface (input)
I_{105} = cross sectional area, ft^2, of the distal vertical cross section of the aquifer

17. $F(X_8, P_c)_{pk} = (I_{107} / I_{96})\,(F_8 / 25.4)\,(I_{68} / 640)^{(1 - I_{97})}$ (for a region)
$= I_{107}\, F_{13} / I_{75}$ (for a channel)

$F(X_8, P_c)_{pk}$ = peak discharge rate, ft^3/s, for a given region or channel as a result of a runoff event
I_{96} = duration equation coefficient for a given channel or region (input)
F_8 = runoff event , mm /m^2 /day, from a given region
F_{13} = channel runoff, ft^3 / day
I_{68} = area, acres, of the region containing plant(s) where runoff has occurred (input)
I_{75} = estimated duration of a runoff event in a given channel or region, hours
I_{97} = duration equation exponent for a given channel or region (input)
I_{107} = regional or channel peak flow parameter (input)

18. $F(X_{1,1}, P_s) = 95\,(F_8\, F_{17} / 25.4)^{0.56}\, I_{108}\, I_{109}\, I_{110}\, I_{111}\, I_{61}$

$F(X_{1,1}, P_s)$ = weight of regional surface soil , tons /acre /day, eroded from a given moisture runoff event
F_8 = runoff event , mm /m^2 /day, from a given region
F_{17} = peak discharge rate, ft^3/s, for a given region as a result of a runoff event
I_{61} = ratio of crown cover fraction of a given plant type to the total cover of all plant types in a given region
I_{108} = estimate of (K) in the USLE equation (input)
I_{109} = estimate of (C) in the USLE equation (input)
I_{110} = estimate of (P) in the USLE equation (input)
I_{111} = estimate of (L) in the USLE equation (input)

19. $F(X_{1,1}, P_{s,i}) = 2000\, F_{18}\, I_{68}\, I_{112i}\, I_{129} / I_{113}$ (for regions)
$= 3600\, I_{75}\, I_{74}\, I_{112i}\, I_{129}$ (for channels)

$F(X_{1,1}, P_{s,i})$ = loss of particles of the ith diameter class, lbs /day, from a region or channel bed
F_{18} = weight of regional surface soil , tons /acre /day, eroded from a given moisture runoff event
I_{68} = area, acres, of the region containing a given number of plant type(s) (input)
I_{74} = width of a given channel or channel segment, feet (input)
I_{75} = estimated duration of a runoff event for a given channel or channel segment, hours
I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed
I_{113} = sum of all particle transport capacities, lbs /s-ft, for all diameter classes from a given region or channel bed normalized to the median particle diameter for a given soil or sediment
I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

20. $F(X_1, P_s) = \sum_{i=1}^{7} F_{19i}$

$F(X_1, P_s)$ = total soil or sediment loss , lbs /day, over all diameter classifications from a region or channel bed
F_{19i} = loss of particles of the ith diameter class, lbs /day, from a region or channel bed

21. $F(P_b, X_{5,1}) = X_{5,1} L_3 I_{133} I_{134}$

$F(P_b, X_{5,1})$ = potential gross photosynthesis, g/m^2/day, of a given plant type
$X_{5,1}$ = plant leaf compartment, g / m^2
L_3 = photosynthetic gate (0,1)
I_{133} = photosynthetic efficiency, g-photosynthate /ly -day-g-leaf
I_{134} = effect of temperature on photosynthesis

22. $F(P_b, X_{5,1})_a = F_{21}$

$= F_{21} I_{145} / I_{141}$ $\qquad I_{141} > I_{145}$

$F(P_b, X_{5,1})_a$ = adjusted potential gross photosynthesis, g/m^2/day
F_{21} = potential gross photosynthesis, g/m^2/day, of a given plant type
I_{145} = adjusted amount of moisture available for photosynthesis, ml /m^2 /day
I_{141} = moisture requirement, ml /m^2 /day, to realize potential gross photosynthesis

23. $F(P_b, X_{5,1})_n = F_{22} C_9 / C_{10}$

$F(P_b, X_{5,1})_n$ = net photosynthesis, g/m^2/day, for a given plant type
F_{22} = adjusted potential gross photosynthesis, g/m^2/day, for a given plant type
C_9 = respiratory maintenance fraction (database)
C_{10} = photosynthetic efficiency factor (database)

24. $F(X_{5,1} : X_{4,1} : X_{3,1}) = (1 - C_{11})(1 - I_{149}) F_{23}$

$F(X_{5,1} : X_{4,1} : X_{3,1})$ = photosynthate transport from leaves-stems to root, g /m^2 /day
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live stem compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
C_{11} = photosynthate transport fraction to stems (database)
I_{149} = photosynthate transport fraction to fruiting body

F_{23} = net photosynthesis, g/m^2/day, for a given plant type

25. $F(X_{5,1} : X_{4,1}) = C_{11}(1 - I_{149}) F_{23}$

$F(X_{5,1} : X_{4,1})$ = photosynthate transport from leaves to stems, g/m^2/ day
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live stem compartment, g/ m^2
I_{149} = photosynthate transport fraction to fruiting body
F_{23} = net photosynthesis, g/m^2/day, for a given plant type
C_{11} = photosynthate transport fraction to stems

26. $F(X_{5,1} : X_{4,1} : X_{6,1}) = F_{23} I_{149}$

$F(X_{5,1} : X_{4,1} : X_{6,1})$ = photosynthate transport to developing fruits and seeds, g/m^2 /day
F_{23} = net photosynthesis, g/m^2/day, for a given plant type
I_{149} = photosynthate transport fraction to fruiting body
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live stem compartment, g/ m^2

27. $F(X_{5,1}, X_{5,2}) = 0.005 X_{5,1}$

$= 0.693 X_{5,1} / 365$	$L_4 = 1; L_5 = 0$
$= [2 / (365 C_3)] C_{14} X_{5,1}$	$L_4 = 0; L_5 = 0$
$= X_{5,1}$	$L_4 = 0; L_5 = 0; I_5 = I_{136}$
$= \{1.1 - [1 / (C_6 - 5)] D_2\} C_{14} X_{5,1}$	$L_4 = 0; L_5 = 1; D_2 < C_6 - 5$
$= [C_{15} (D_2 - C_5) + 0.1] C_{14} X_{5,1}$	$L_4 = 0; L_5 = 1; D_2 > C_5$
$= X_{5,1}$	$L_4 = 0; L_5 = 1; I_5 = I_{136}; C_3 = 1$
$= 0.1 X_{5,1}$	$L_6 = 1; I_{139} \neq I_{154}; C_3 \leq 2; I_{139} = I_{137}$
$= 0.0$	$L_6 = 1; I_{139} \neq I_{154}; C_3 \leq 2;$ $I_{139} \neq I_{137}; I_{137} < I_{138}; I_{137} \geq I_{139}$
$= 0.0$	$F_{27} \leq 0$

28. $F(X_{4,1}, X_{4,2}) = 0.005 X_{4,1} (1 - X_{5,1} / X_{4,1})$

$= (X_{4,1} - X_{5,1}) [2 / (365 C_3)] C_{14}$	$L_4 = 1; L_5 = 0$
$= [2 / (365 C_3)] C_{14} (1 - X_{5,1} / X_{4,1}) X_{4,1}$	$L_4 = 0; L_5 = 0$
$= F_{27} (X_{4,1} - X_{5,1}) / X_{4,1}$	$L_4 = 0; L_5 = 1; D_2 < C_6 - 5$
$= F_{27} (X_{4,1} - X_{5,1}) / X_{4,1}$	$L_4 = 0; L_5 = 1; D_2 > C_5$
$= X_{4,1} - X_{5,1}$	$L_4 = 0; L_5 = 1; I_5 = I_{136}; C_3 = 1$
$= 0.1 (1 - X_{5,1} / X_{4,1}) X_{4,1}$	$L_6 = 1; I_{139} \neq I_{154}; C_3 \leq 2;$ $I_{139} = I_{137}$
$= 0.0$	$L_6 = 1; I_{139} \neq I_{154}; C_3 \leq 2;$ $I_{139} \neq I_{137}; I_{137} < I_{138}; I_{137} \geq I_{139}$
$= 0.0$	$F_{28} \leq 0$

29. $F(X_{6,1}, X_{6,2}) = 0.005 X_{6,1}$

$= X_{6,1}$	$L_4 = 0; L_5 = 0; I_5 = I_{136}$
$= X_{6,1}$	$L_4 = 0; L_5 = 1; I_5 = I_{136}; C_3 = 1$
$= 0$	$F_{29} \leq 0$

30. $F(X_{3,1}, X_{3,2}) = 0.005 X_{3,1} (X_{5,1} / X_{4,1})$

$= 8[2 / (365 C_3)] X_{3,1} (X_{5,1} / X_{4,1})$	$L_4 = 1; L_5 = 0$
$= 15 [2 / (365 C_3)] X_{3,1} (X_{5,1} / X_{4,1})$	$L_4 = 0; L_5 = 0$
$= X_{3,1} (X_{5,1} / X_{4,1})$	$L_4 = 0; L_5 = 1; I_5 = I_{136}; C_3 = 1$
$= X_{3,1} \quad (X_{5,1} / X_{4,1})$	$L_6 = 1; I_{139} \neq I_{154}; C_3 \leq 2;$ $I_{139} = I_{137}$

$= 0.0$ $\qquad F_{30} \leq 0$

31. $F(X_{3,1}, X_{3,3}) = 0.005\, X_{3,1}\,(1 - X_{5,1} / X_{4,1})$

$= 8[2 / (365\, C_3)]\, X_{3,1}\,(1 - X_{5,1} / X_{4,1})$ $\qquad L_4 = 1;\ L_5 = 0$

$= 15\,[2 / (365\, C_3)]\, X_{3,1}\,(1 - X_{5,1} / X_{4,1})$ $\qquad L_4 = 0;\ L_5 = 0$

$= X_{3,1}\,(1 - X_{5,1} / X_{4,1})$ $\qquad L_4 = 0;\ L_5 = 1;\ I_5 = I_{136};\ C_3 = 1$

$= X_{3,1}\,(1 - X_{5,1} / X_{4,1})$ $\qquad L_6 = 1;\ I_{139} \neq I_{154};\ C_3 \leq 2;\ I_{139} = I_{137}$

$= 0.0$ $\qquad F_{30} \leq 0$

$F(X_{5,1}, X_{5,2})$ = conversion of live leaf biomass to dry leaf biomass, g /m^2 /day
$F(X_{4,1}, X_{4,2})$ = conversion of live stem biomass to dry stem biomass, g /m^2 /day
$F(X_{6,1}, X_{6,2})$ = conversion of live seed biomass to dry seed biomass, g /m^2 /day
$F(X_{3,1}, X_{3,2})$ = conversion of live root biomass density to readily decomposable dry root biomass g /m^2 /day
$F(X_{3,1}, X_{3,3})$ = conversion of live root biomass density resistant to decomposition dry root biomass, g /m^2 /day
$X_{3,1}$ = live root biomass density, g /m^2 /day
$X_{3,2}$ = dry root biomass, g /m^2 /day, which is readily decomposed
$X_{3,3}$ = dry root biomass, g /m^2 /day, which is not readily decomposed
$X_{4,1}$ = live stem biomass density, g /m^2 /day
$X_{4,2}$ = dry stem biomass density, g /m^2 /day
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2 /day
$X_{6,2}$ = dry seed biomass density, g /m^2 /day
L_4 = evergreen tree and shrub gate (1,0)
L_5 = woody or deciduous plant gate (0,1)
L_6 = irrigation gate (1,0) (input)
C_3 = mean life-span, years, of the plant type simulated
C_5 = optimum temperature for photosynthesis for a given plant type, ° C (database)
C_6 = difference between optimum and lowest temperature tolerance for photosynthesis of a given plant type, ° C (database)
C_{14} = live-to-dry stem and leaf conversion factor (database)
C_{15} = live-to-dry leaf conversion factor for herbaceous plants (database)
D_2 = mean daily temperature, ° C
I_5 = Julian day number (1 - 365)
I_{136} = end of frost-free period for the region, Julian day number
I_{137} = day counter during the growing season, day number (may exceed 365 for some crops)
I_{138} = planting date , Julian day number (input)
I_{139} = last harvest date, Julian day number (may exceed 365 for some crops) (input)
F_{27} = conversion of live leaf biomass to dry leaf biomass, g /m^2 /day
I_{154} = last irrigation date, Julian day number (input)

32. $F(X_{5,2}, X_{7,1}) = I_{155}\, I_{156}\, X_{5,2}$

33. $F(X_{6,2}, X_{7,1}) = I_{155}\, I_{156}\, X_{6,2}$

$F(X_{5,2}, X_{7,1})$ = conversion of dry leaf litter to humus, g /m^2 /day
$F(X_{6,2}, X_{7,1})$ = conversion of dry seed litter to humus, g /m^2 /day
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{7,1}$ = leaf, seed, and root-twig humus combined biomass density, g /m^2

$X_{6,2}$ = dry seed biomass density, g /m^2 /day
I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{156} = effect of soil moisture on litter decomposition

34. $F(X_{4,2},X_{4,4}) = 0.693\ X_{4,2}\ [100\ X_{4,4})/\ (365\ C_3)]\ X_{4,6}\ /\ (X_{4,6} + X_{4,7})$

35. $F(X_{4,2},X_{4,5}) = 0.693\ X_{4,2}\ [10\ X_{4,5})/\ (365\ C_3)]\ X_{4,7}\ /\ (X_{4,6} + X_{4,7})$

36. $F(X_{4,2},X_{4,3}) = [0.693\ X_{4,2}\ /\ (365\ C_3)]\ I_{159}$

$F(X_{4,2},X_{4,4})$ = conversion of dry standing stem biomass to lying branch and twig litter, g /m^2 /day
$F(X_{4,2},X_{4,5})$ = conversion of dry standing biomass to lying bole litter, g /m^2 /day
$F(X_{4,2},X_{4,3})$ = total conversion of dry standing stem to lying stem biomass, g /m^2 /day
$X_{4,2}$ = standing dry stem biomass density, g /m^2
$X_{4,3}$ = dry stem biomass lying on ground surface, g /m^2
$X_{4,4}$ = branch and twig component of lying stem biomass, g /m^2
$X_{4,5}$ = bole component of lying stem biomass, g /m^2
$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2
$X_{4,7}$ = bole component of live above ground biomass, g /m^2
C_3 = mean life-span, years, of the plant type simulated
I_{159} = weighting factor for estimating litter formation from stems (boles) and branches

37. $F(X_{4,4},X_{7,2}) = I_{155}\, I_{156}\, X_{4,4}\ /\ 10$

38. $F(X_{4,5},X_{7,2}) = I_{155}\, I_{156}\, X_{4,5}\ /\ 10$

$F(X_{4,4},X_{7,2})$ = conversion of branch-twig biomass density to humus, g /m^2 /day
$F(X_{4,5},X_{7,2})$ = conversion of bole biomass density to humus, g /m^2 /day
I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{156} = effect of soil moisture on litter decomposition
$X_{4,4}$ = branch and twig component of standing stem biomass, g /m^2
$X_{4,5}$ = bole component of standing stem biomass, g /m^2
$X_{7,2}$ = branch, bole, twig, and root bole-branch humus combined biomass density for a given plant, g /m^2

39. $F(X_{3,2},X_{7,1}) = I_{169i}\, I_{170i}\, I_{171i}\, X_{3,2}$

40. $F(X_{3,3},X_{7,2}) = I_{169i}\, I_{170i}\, I_{171i}\, X_{3,3}\ /10$

$F(X_{3,3},X_{7,1})$ = conversion of readily decomposable root fraction to humus, g /m^2 / day
$F(X_{3,4},X_{7,2})$ = conversion of resistant to decomposition root fraction to humus, g /m^2 /day
I_{169i} = effect of soil temperature of the ith soil on root litter decomposition
I_{170i} = effect of soil moisture of the ith soil layer on root litter decomposition
I_{171i} = root penetration fraction into the ith soil layer
$X_{3,2}$ = dry root biomass, g /m^2 /day, which is readily decomposed
$X_{3,3}$ = dry root biomass, g /m^2 /day, which is not readily decomposed
$X_{7,1}$ = leaf, seed, and root-twig humus combined biomass density, g /m^2
$X_{7,2}$ = branch, bole, twig, and root bole-branch humus combined biomass density for a given plant, g /m^2

41. $F(X_{7,1i},P_b) = I_{179}\, X_{7,1}$

42. $F(X_{7,2i},P_b) = 0.05\ I_{179}\, X_{7,2}$

$F(X_{7i,1}, P_b)$ = humus recycling losses, g /m^2 /day, for readily mineralized humus
$F(X_{7i,2}, P_b)$ = humus recycling losses, g / m_2 /day, for resistant humus
$X_{7,1i}$ = readily mineralized portion of humus, g /m^2, in the ith soil layer
$X_{7,2i}$ = resistant portion of humus, g /m^2 , in the ith soil layer
I_{179} = mineralization factor for readily mineralized humus
P_b = biomass source/sink

43. $F(X_{7,1i}, X_{7,1i+1}) = (0.00055 / I_{37i}) X_{7,1i}$
44. $F(X_{7,2i}, X_{7,2i+1}) = (0.00055 / I_{37i}) X_{7,2i}$

$F(X_{7,1i}, X_{7,1i+1})$ = transport of readily decomposable humus from the ith soil layer to the (i=1)th soil layer, g /m^2 /day
$F(X_{7,2i}, X_{7,2i+1})$ = transport of resistant humus from the ith soil layer to the (i+1)th soil layer, g /m^2 /day
$X_{7,1i}$ = readily mineralized portion of humus, g /m^2, in the ith soil layer
$X_{7,2i}$ = resistant portion of humus, g /m^2 , in the ith soil layer
I_{37j} = thickness of the jth soil layer, mm

45. $F(X_{5w}, P_t) = [\sum_{j=1}^{n} \sum_{i=2}^{I_{20j}} F_{5j,i} X_{5,1j}] / \sum_{j=1}^{n} X_{5,1j}$ $\qquad I_{179,1} = I_{179,2} = \ldots I_{179,n}$

$F(X_{5w}, Pt)$ = weighted total transpiration, mm /m^2 /day, for the niche matched plant group
n = number of plants in a niche matching group of plants
$F_{5j,i}$ = total transpiration, mm/m^2/ day, for the ith soil layer of the jth plant (see App. A., No. 5)
I_{20j} = number of soil layers specified for the jth plant (input or estimated)
$X_{5,1j}$ = live leaf biomass density, g /m^2, of the jth plant in the niche matching group
$I_{179,j}$ = niche competition match number for the jth plant in the matched plant group (input)

46. $F(P_{n,v}, X_{11i}) = P_{n,v} (I_{180i-1} - I_{180i}) / I_{188}$ $\qquad i > 1$
$= 0.0$ $\qquad X_{11i} \leq 0.01\, I_{183i}$
47. $F(X_{11i}, P_{m,v}) = X_{11i} I_{187}$ $\qquad P_{n,v} < 0.0$

$F(P_{n,v}, X_{11i})$ = transport of annual net wood production to the ith diameter class, ft^3 /yr
$F(X_{11i}, P_{m,v})$ = net annual mortality loss from the ith diameter class, ft^3 /year
$P_{n,y}$ = net wood increment, ft^3 / yr
$P_{m,v}$ = net wood losses, ft^3 / yr
X_{11i} = total volume of trees in the ith diameter class, ft^3
I_{180i} = intra-class competition factor for the ith diameter class
I_{187} = mortality factor for wood volume losses
I_{188} = sum of all intra-class competition factors

48. $F(X_{11i}, P_{rji}) = X_{11i} - 126 \exp(-0.394i) I_{183i}$
$= 0.0$ $\qquad X_{11i} < 126 \exp(-0.394i) I_{183i}$
$= X_{11i}$ $\qquad i \geq 13$

P_{rji} = wood removed, ft^3, from the ith diameter class using Reverse-J cutting option
X_{11i} = total volume of trees in the ith diameter class, ft^3
I_{183i} = volume of a unit tree in the ith diameter class, ft^3

49. $F(11_t, P_{rj}) = \sum_{i=1}^{n} P_{rji}$

P_{rji} = wood removed, ft^3, from the ith diameter class using Reverse-J cutting option
P_{rj} = wood removed, ft3, from all diameter class combined using Reverse-J cutting option
11_t = total volume of trees from all diameter classes combined, ft^3

50. $F(X_{12,5}, X_{13,4}) = I_{198i} I_{204} X_{3,1} X_{5,1} / X_{4,1}$

$F(X_{12,5}, X_{13,4})$ = phosphate transport from soil to plant roots, g-PO_4 / m^2 /day
I_{198i} = adjusted plant root uptake g-PO_4 / g -root/ m^2 /day of inorganic phosphate for ith soil layer
$X_{12,5}$ = labile soil phosphate content (compartment), g-PO_4 /m^2 (see App. A., No. XXX)
$X_{13,4}$ = root phosphate content (compartment), g-PO_4 /m^2 (see App. A., No. XXX)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{204} = uptake coefficient

51. $F(X_{12,2} + X_{12,3}, X_{13,3}) = (I_{211i} + I_{214i}) I_{204} X_{3,1} X_{5,1} / X_{4,1}$

$F(X_{12,2} + X_{12,3}, X_{13,3})$ = nitrogen transport from soil to plant roots, g-N / m^2 /day, from NO_3^- and NH_4^+ uptake
I_{211i} = adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
$_{214i}$ = adjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
$X_{12,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 (see App. A., No. XXX)
$X_{12,3}$ = soil ammonium ion content (compartment), g-NH_4 /m^2 (see App. A., No. XXX)
$X_{13,3}$ = root nitrogen content (compartment), g-N /m^2 (see App. A., No. XXX)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{204} = uptake coefficient

52. $F(P_{ni}, X_{12,2,1}) = 0.5\,(0.00055 + 0.0006\,P_p) / (1400\,I_{37})$

53. $F(P_{nh}, X_{12,3,1}) = 0.5\,(0.00055 + 0.0006\,P_p) / (1400\,I_{37})$

$F(P_{ni}, X_{12,2,1})$ = transport of nitrate ion from atmosphere to 1st soil layer, g-NO_3 /g-soil /day
$F(P_{nh}, X_{12,3,1})$ = transport of ammonium ion from atmosphere to 1st soil layer, g-NH_4 /g-soil /day
P_{ni} = atmospheric source term for nitrate ion, g-NO_3 /m^2
P_{nh} = atmospheric source term for ammonium ion, g-NH_4 /m^2
P_p = precipitation , mm /day
$X_{12,2,1}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the 1st layer of soil
$X_{12,3,1}$ = soil ammonium ion content (compartment), g-NH_4 /m^2 , in the 1st layer of soil
I_{37} = thickness of 1st soil layer, mm (set at 20mm in the model)

54. $F(X_{12,2}, P_{ng}) = \sum_{i=1}^{n} I_{229i}$

$F(X_{12,2,i}, P_{ng})$ = denitrification rate of nitrate to nitrogen gas, g-N/ m^2 /day
I_{229i} = denitrification rate, g-NO_3 / gdwt-soil /day, for the ith soil layer
P_{ng} = nitrogen gas production from nitrate ion, g-N/m^2 /day

$X_{12,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the soil

55. $F(X_{12,3}, X_{12,2}) = \sum_{i=1}^{n} I_{225i}$

$X_{12,3}$ = soil ammonium ion content (compartment), g-NH_4 /m^2 , in the soil
$X_{12,2}$ = soil nitrate ion content (compartment), g-NO_3 /m^2 , in the soil
I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer

56. $F_e(P_f, X_{14,m,k}) = 70\, X_{16,m,k}\, I_{249k}\, I_{250k}\, I_{251m,k}\, I_{252k} / (4500\, I_{253k})$

57. $F(P_f, X_{14,m,k}) = 70\, X_{15,m,k}\, I_{249k}\, I_{250k}\, I_{251m,k}\, I_{252k} / (4500\, I_{253k})$
$= 1.25\, F_e(P_f, X_{14,m,k})$ $\quad F_{57} > 1.25\, F_{56}$
$= F_e(P_f, X_{14,m,k})$ $\quad F_{57} < F_{56}$
$= F(P_f, X_{14,m,k})\, I_{254k} / I_{245k}$ $\quad I_{254k} > I_{245k}$

$F(P_f, X_{14,m,k})$ = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population
$F_e(P_f, X_{14,m,k})$ = expected transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population
P_f = transport of food biomass, kgdwt/day, from plant and/or other animal compartments
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population
I_{245k} = food requirement, kg /day, by the kth mammalian population
I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population
I_{250k} = activity factor for the kth mammalian population (input)
$I_{251m,k}$ = age effect factor on food intake for the mth age group of the kth mammalian population
I_{252k} = food digestibility effect factor for the kth mammalian population
I_{253k} = digestibility fraction based for the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

58. $F(X_{15,m,k}, P_r) = 70\, X_{15,m,k}\, I_{249k}\, I_{250k} / 4500$

$F(X_{15,m,k}, P_r)$ = metabolic heat production losses, kgfwt/day
$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population
I_{250k} = activity factor for the kth mammalian population (input)
P_r = respiration component, kgfwt/day

59. $F(F_{57}, P_{gs}) = I_{287k}\, F_{57}\, I_{253k}$

60. $F(F_{57}, P_{ur}) = I_{288k}\, F_{57}\, I_{253k}$

61. $F(F_{57}, P_{fc}) = F_{57}(1 - I_{253k})$

62. $F(P_{mt}, X_{15m,k}) = 2.5\,(I_{291m,k} + I_{293m,k})$ $\quad I_{287m,k} > 0$

$= -2.13\ I_{289m,k}$ $\qquad I_{289m,k} < 0$

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

$I_{291m,k}$ = protein fraction the growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

P_{mt} = metabolic source/sink, kg/day

$F(F_{57}, P_{gs})$ = gas losses kg/day from ingested food

$F(F_{57}, P_{ur})$ = mass losses kg/day from ingested food via urine

$F(F_{57}, P_{fc})$ = mass losses kg/day from ingested food via urine

F_{57} = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population

I_{287k} = digestive gas mass loss fraction for the kth mammalian population (input)

I_{288k} = digestive urine mass loss fraction (input)

I_{253k} = digestibility fraction for the kth mammalian population

P_{gs} = digestive mass losses from gas production, kg/day

P_{ur} = digestive mass losses from urine excretion, kg/day

P_{fc} = digestive mass losses from fecal excretion, kg/day

63a. $F(P_{pe}, X_{18p,m}) = I_{329m}$ $\qquad O_6 > 0;\ O_{7m} > 1;\ O_{8n} > 1;\ I_{331m} \le I_5 \le I_{332m}$
$I_{333m} \le I_{29} \le I_{334m}$

63b. $F(P_{pe}, X_{18p,m}) = I_{329m}$ $\qquad O_6 > 0;\ O_{7m} > 1;\ I_{343m} \le I_5 \le I_{343m} + I_{344m}$
$I_{345m} = I_{29};\ I_{346m,n} \ge 0$

64a. $F(P_{ge}, X18_{g,m}) = I_{330m}$ $\qquad O_6 > 0;\ O_{7m} > 1;\ O_{9m} > 1;\ I_{331m} \le I_5 \le I_{332m}$
$I_{333m} \le I_5 \le I_{334m}$

64b. $F(P_{ge}, X_{18g,m}) = I_{330m}$ $\qquad O_6 > 0;\ O_{7m} > 1;\ I_{343m} \le I_5 \le I_{343m} + I_{344m}$
$I_{345m} = I_{29};\ I_{346m,n} \ge 0$

O_6 = chronic/acute access gate (0,1) (input)

O_{7m} = contaminant gate (0,1) for the mth contaminant (input)

O_{8m} = chronic access gate(0,1) for the mth particulate air contaminant (input)

O_{9m} = chronic access gate(0,1) for the mth vaporous air contaminant (input)

I_{329m} = air concentration, m^{-3}, for the mth particulate contaminant for a given region

I_{330m} = air concentration, m^{-3}, for the mth vaporous contaminant for a given region

I_{331m} = Julian day number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region

I_{332m} = Julian day number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region

I_{333m} = year number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region

I_{334m} = year number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region

$F(P_{pe}, X_{18p,m})$ = transport of external resuspendable atmospheric contamination of the mth contaminant to a given to a given region, m^{-3}, as particulates

$F(P_{ge}, X_{18p,m})$ = transport of external atmospheric contamination of the mth contaminant to a given region, m^{-3}, as a vapor

P_{pe} = external source of contamination for a given contaminant, m^{-3}, as particulates

P_{ge} = external source of contamination for a given contaminant, m^{-3}, as a vapor
$X_{18p,m}$ = regional particulate concentration, m^{-3}, for the mth contaminant
$X_{18g,m}$ = regional vapor concentration, m^{-3}, for the mth contaminant
I_5 = Julian day number (1 - 365)
I_{29} = Julian year number of specific year of simulation

65. $F(X_{18r}, X_{r12,1}) = 0.67\, I_{347}\, X_{18r}\, (1 - I_{348}) / (I_{37j}\, 1400)$

66a. $F(X_{18r}, X_{r13,5,1}) = 0.67\, I_{347}\, X_{18r}\, I_{19}$

66b. $F(X_{18r}, X_{r13,5,2}) = 0.9\, I_{347}\, X_{18r}\, I_{348}$

67. $F(X_{18r}, X_{r13,6,1}) = 0.67\, I_{347}\, X_{18r}\, I_{348}$

68. $F(X_{18r}, X_{r13,4,1}) = 0.67\, I_{347}\, X_{18r}\, I_{349}$

69. $F(X_{18r}, X_{r13,6,2} + X_{r13,4,2}) = 0.67\, I_{347}\, X_{18r}\, I_{350}\, (X_{4,2} + X_{6,2})/(X_{4,1} - X_{5,1})$

70. $F(X_{18r}, X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5}) = 0.1\, I_{347}\, X_{18r}\, I_{348}$

$F(X_{18r}, X_{r12,1})$ = transport of resuspendable contaminant, gdwt-1 soil day-1, from the atmosphere to the soil surface for a specific contaminant in a given region
$F(X_{18r}, X_{r13,5,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to live leaf surfaces
$F(X_{18r}, X_{r13,5,2})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dry leaf surfaces
$F(X_{18r}, X_{r13,6,2} + X_{r13,4,2})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dead standing vegetation and dry seeds
$F(X_{18r}, X_{r13,4,3} \rightarrow X_{r13,4,4} + X_{r13,4,5})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from the atmosphere to dead lying boles and branches
I_{347} = U.S. mean dry deposition velocity, m/ day, for a given region
X_{18r} = regional resuspendable particulate concentration, m^{-3}, for a given contaminant
$X_{r12,1}$ = top soil layer, resuspendable contaminant concentration, $gdwt^{-1}$
$X_{r13,5,1}$ = resuspendable contaminant concentration, m^{-2}, on live leaves
$X_{r13,5,2}$ = resuspendable contaminant concentration, m^{-2}, on dry leaves
$X_{r13,4,1}$ = resuspendable contaminant concentration, m^{-2}, on live standing vegetation
$X_{r13,4,2}$ = resuspendable contaminant concentration, m^{-2}, on dry standing vegetation
$X_{r13,4,3}$ = resuspendable contaminant concentration, m^{-2}, on dry lying vegetation
$X_{r13,4,4}$ = resuspendable contaminant concentration, m^{-2}, on dry lying branches and twigs
$X_{r13,4,5}$ = resuspendable contaminant concentration, m^{-2}, on dry lying boles (main stem)
$X_{r13,6,1}$ = resuspendable contaminant concentration, m^{-2}, on seeds
$X_{r13,6,2}$ = resuspendable contaminant concentration, m^{-2}, on dry seeds
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone
I_{350} = total surface area, m^2/m^2, of combined bole and branches of a given plant
I_{37j} = thickness, mm, of the first soil layer (j=1) (input)
I_{19} = leaf area index, m^2/m^2, (LAI)
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

71a. $F(X_{r13,5,1} + X_{r13,6,1}, X_{18r}) = 0.2\, I_{353}\, (X_{r13,5,1} + X_{r13,6,1})$

71b. $F(X_{r13,5,1} + X_{r13,6,1}, X_{r12,1}) = 0.693 (X_{r13,5,1} + X_{r13,6,1})/14$

72a. $F(X_{r13,4,1}, X_{18r}) = 0.02\, I_{353}\, X_{r13,4,1}$

72b. $F(X_{r13,4,1}, X_{r12,1}) = 0.693\, X_{r13,4,1} / 42$

73a. $F(X_{ra13,5,2} + X_{ra13,4,2} + X_{ra13,6,2}, X_{r18}) = 0.2\, I_{353} (X_{ra13,5,2} + X_{ra13,6,2} + 0.1\, X_{ra13,4,2})$

73b. $F(X_{ra13,5,2} + X_{ra13,4,2} + X_{ra13,6,2}, X_{r12,1}) = 0.693 (X_{ra13,5,2} + X_{ra13,6,2} + 0.1\, X_{ra13,4,2})/ 14$

74a. $F(X_{ra13,4,3} \rightarrow X_{ra13,4,4} + X_{ra13,4,5}, X_{18r}) = 0.02\, I_{353}\, X_{ra13,4,3}$

74b. $F(X_{ra13,4,3} \rightarrow X_{ra13,4,4} + X_{ra13,4,5}, X_{r12,1}) = 0.693\, X_{ra13,4,3} / 28$

$F(X_{r13,5,1}, X_{18r})$ = transport of resuspendable contaminant, $m^{-2}\ day^{-1}$, from live leaf surfaces to the atmosphere

$F(X_{r13,5,1} + X_{r13,6,1}, X_{r12,1})$ = transport of resuspendable contaminant, $m^{-2}\ day^{-1}$, from live leaf and seed surfaces to the soil surface

$F(X_{r13,4,1}, X_{18r})$ = transport of resuspendable contaminant, $m^{-2}\ day^{-1}$, from live bole and branch surfaces to the atmosphere

$F(X_{r13,4,1}, X_{r12,1})$ = transport of resuspendable contaminant, $m^{-2}\ day^{-1}$, from live bole and branch surfaces to the soil surface

$F(X_{ra13,5,2} + X_{ra13,4,2} + X_{ra13,6,2}, X_{18r})$ = transport of resuspendable contaminant, $m^{-2}\ day^{-1}$ from dry bole, branches leaves, and seeds surfaces to the atmosphere

$F(X_{ra13,5,2} + X_{ra13,4,2} + X_{ra13,6,2}, X_{r12,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1} from dry bole, branches leaves, and seeds surfaces to the soil surface

$F(X_{ra13,4,3} \rightarrow X_{ra13,4,4} + X_{ra13,4,5}, X_{18r})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from dry lying bole and branch surfaces to the atmosphere

$F(X_{ra13,4,3} \rightarrow X_{ra13,4,4} + X_{ra13,4,5}, X_{r12,1})$ = transport of resuspendable contaminant, m^{-2} day^{-1}, from dry lying bole and branch surfaces to the soil surface

O_1 = niche competition option (1,0) (input)

O_2 = succession option (1,0) (input)

D_2 = mean daily temperature, $^\circ C$

F_2 = transport of moisture from snowfall events to soil, mm/day

I_{353} = resuspension fraction from plant surface to the atmosphere

$X_{12,1}$ = top soil layer contaminant concentration, $gdwt^{-1}$

$X_{r13,j,k}$ = contaminant resuspendable concentration, m-2, on the surface of the jth plant part (roots, stems, leaves, seeds), and kth state (live, dry, dry-lying) for a given plant type in a given region

$X_{r13,5,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of live leaves for a given plant type

$X_{ra13,5,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry leaves for all plants in a given region

$X_{r13,4,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of live standing vegetation for a given plant type

$X_{ra13,4,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region

$X_{ra3,4,3}$ = weighted average contaminant resuspendable concentration, on the surface of dry lying vegetation for all plants in a given region

$X_{ra13,4,4}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying branches and twigs for all plants in a given region

$X_{ra3,4,5}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying boles (main stem) for all plants in a given region

$X_{r3,6,1}$ = contaminant resuspendable concentration, m^{-2}, on the surface of seeds for a given plant type

$X_{ra13,6,2}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry seeds for all plants in a given region

75a. $F(X_{18s} ,X_{s13,5,1}) = 0.4\ X_{18s}\ I_{348}\ X_{5,1} / (X_{5,1} +X_{6,1})$

75b. $F(X_{18s} ,X_{s13,5,2}) = 0.4\ X_{18s}\ I_{348} (1- I_{354})\ X_{5,2} / (X_{5,2} + X_{4,4} +X_{4,5})$

76. $F(X_{18s} ,X_{s13,6,1}) = 0.4\ X_{18s}\ I_{348}\ X_{6,1} / (X_{6,1} +X_{5,1})$

77a. $F_s(X_{18s} ,X_{s13,4,1}) = 0.4\ X_{18s}\ I_{349}\ [1 - X_{4,2} / (X_{4,2} + X_{4,6} +X_{4,7})]\ X_{4,7} / (X_{4,7} +X_{4,6})$

77b. $F_b(X_{18s} ,X_{s13,4,1}) = 0.4\ X_{18s}\ I_{352}\ \{1 - [X_{4,1} /(X_{4,1} + X_{4,6} +X_{4,7})]\}\ X_{4,6} / (X_{4,7} +X_{4,6})$

78. $F(X_{18s} ,X_{s13,6,2} +X_{s13,4,2}) = 0.4\ X_{18s}\ [I_{349}\ X_{4,2} / (X_{4,2} + X_{4,6} +X_{4,7})][I_{352}\ X_{4,1} /(X_{4,1} + X_{4,6} +X_{4,7})]\ I_{355}$

79. $F(X_{18s} , X_{s3,4,3} \rightarrow X_{s13,4,4} + X_{s13,4,5}) = 0.4\ X_{18s}\ I_{348} (1- I_{354})\ X_{4,4} / (X_{5,2} + X_{4,4} +X_{4,5})$

$F(X_{18s} ,X_{s3,5,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to live leaf surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,5,2})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to dry leaf surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,6,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to seed surfaces from saltation-creep and rainsplash events

$F_s(X_{18s} ,X_{s13,4,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to bole surfaces from saltation-creep and rainsplash events

$F_b(X_{18s} ,X_{s13,4,1})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to branch surfaces from saltation-creep and rainsplash events

$F(X_{18s} ,X_{s13,6,2} +X_{s13,4,2})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to dead standing vegetation and dry seeds from saltation-creep and rainsplash events

$F(X_{18s} , X_{s13,4,3} \rightarrow X_{s13,4,4} + X_{s13,4,5})$ = transport of contaminant, $m^{-2}\ day^{-1}$, from the atmosphere to dead lying vegetation and dry seeds from saltation-creep and rainsplash events

X_{18s} = regional particulate concentration, m^{-3}, for a given contaminant from saltation-creep and rainsplash events

$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2

$X_{4,7}$ = bole component of live above ground biomass, g /m^2

$X_{s12,1}$ = top soil layer, saltation-creep-rainsplash contaminant concentration, $gdwt^{-1}$

$X_{s13,5,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live leaves

$X_{s13,5,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry leaves

$X_{s13,4,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live standing vegetation

$X_{s13,4,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry standing vegetation

$X_{s13,4,3}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying vegetation

$X_{s13,4,4}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying branches and twigs

$X_{s13,4,5}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry lying boles (main stem)

$X_{s13,6,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on seeds

$X_{s13,6,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry seeds
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone
I_{352} = projected branch cover, m^2/m^2, on ground surface assuming branches are parallel to the ground
I_{355} = the ratio of the integral average contamination on boles and branches of a given plant to the maximum estimates on these surfaces assuming height independence from saltation-creep-rainsplash events
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2
$X_{4,2}$ = dry stem biomass density, g/m^2
$X_{6,2}$ = dry seed biomass density, g/m^2

80a. $F(X_{s13,5,1} + X_{s13,6,1}, X_{18s}) = 0.2\, I_{357}\, (X_{s13,5,1} + X_{s13,6,1})$

80b. $F(X_{s13,5,1} + X_{s13,6,1}, X_{12,1}) = 0.693\, (X_{r13,5,1} + X_{r13,6,1})/8$

81a. $F(X_{s13,4,1}, X_{18s}) = 0.02\, I_{357}\, X_{s13,4,1}$

81b. $F(X_{s13,4,1}, X_{12,1}) = 0.693\, X_{s13,4,1}/24$

82a. $F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{18s}) = 0.2\, I_{357}\, (X_{sal3,5,2} + X_{sal3,6,2} + 0.1\, X_{sal3,4,2})$

82b. $F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{12,1}) = 0.693\, (X_{sal3,5,2} + X_{sal3,6,2} + 0.1\, X_{sal3,4,2})/16$

83a. $F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{18s}) = 0.2\, I_{357}\, X_{sal3,4,3}$

83b. $F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{12,1}) = 0.693\, X_{sal3,4,3}/16$

$F(X_{s13,5,1}, X_{18s})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from live leaf surfaces to the atmosphere from saltation-creep-rainsplash events
$F(X_{s13,5,1} + X_{s13,6,1}, X_{12,1})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from live leaf and seed surfaces to the soil surface from saltation-creep-rainsplash events
$F(X_{s13,4,1}, X_{18s})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from live bole and branch surfaces to the atmosphere from saltation-creep-rainsplash events
$F(X_{s13,4,1}, X_{12,1})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from live bole and branch surfaces to the soil surface from saltation-creep-rainsplash events
$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{18s})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from dry bole, branches leaves, and seeds surfaces to the atmosphere from saltation-creep-rainsplash events
$F(X_{sal3,5,2} + X_{sal3,4,2} + X_{sal3,6,2}, X_{12,1})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from dry bole, branches leaves, and seeds surfaces to the soil surface from saltation-creep-rainsplash events
$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{18s})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from dry lying bole and branch surfaces to the atmosphere from saltation-creep-rainsplash events
$F(X_{sal3,4,3} \rightarrow X_{sal3,4,4} + X_{sal3,4,5}, X_{12,1})$ = transport of contaminant, $m^{-2}\,day^{-1}$, from dry lying bole and branch surfaces to the soil surface from saltation-creep-rainsplash events

I_{357} = saltation creep-rainsplash fraction from plant and soil surfaces to the atmosphere

$X_{12,1}$ = top soil layer contaminant concentration, gdwt^{-1}

$X_{s13,5,1}$ = contaminant concentration, m^{-2}, on the surface of live leaves for a given plant type from saltation-creep-rainsplash events

$X_{sa13,5,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry leaves for all plants in a given region from saltation-creep-rainsplash events

$X_{s13,4,1}$ = contaminant concentration, m^{-2}, on the surface of live standing vegetation for a given plant type from saltation-creep-rainsplash events

$X_{sa13,4,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region from saltation-creep-rainsplash events

$X_{sa3,4,3}$ = weighted average contaminant concentration, on the surface of dry lying vegetation for all plants in a given region from saltation-creep-rainsplash events

$X_{sa13,4,4}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry lying branches and twigs for all plants in a given region from saltation-creep-rainsplash events

$X_{sa3,4,5}$ = weighted average contaminant resuspendable concentration, m^{-2}, on the surface of dry lying boles (main stem) for all plants in a given region

$X_{s3,6,1}$ = contaminant concentration, m^{-2}, on the surface of seeds for a given plant type from saltation-creep-rainsplash events

$X_{sa13,6,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry seeds for all plants in a given region from saltation-creep-rainsplash events

84. $F_e (X_{12,1}, X_{12,2}) = I_{358} I_{37i} X_{12,1} / I_{37j}$

$F_e (X_{12,1}, X_{12,2})$ = eluviation of a given contaminant, day^{-1}, from the soil surface into the second soil layer

I_{358} = eluviation rate, mm^{-1} day^{-1}, of contaminant from soil surface to the second layer

$X_{12,1}$ = contaminant concentration, gdwt-soil^{-1}, of the surface layer of soil (20 mm thick)

$X_{12,2}$ = contaminant concentration, gdwt-soil^{-1}, of the second soil layer

I_{37i} = thickness of the 1st (i=1)soil layer, mm (input)

I_{37j} = thickness of the 2nd (j=2) soil layer, mm (input)

85a. $F_w (X_{12,i}, X_{12,i-1}) = I_{360} / (1400\ I_{37i-1})$

85b. $F_w (X_{12,i}, X_{12,i+1}) = I_{361} / (1400\ I_{37i+1})$

$F_w (X_{12,i}, X_{12,i-1})$ = transport of contaminant from the ith soil layer to the (i-1)th layer (top layer excluded), day^{-1} gdwt-soil^{-1}, by soil moisture flow

$F_w (X_{12,i}, X_{12,i+1})$ = transport of contaminant from the ith soil layer to the (i+1)th layer (bottom layer excluded), day^{-1} gdwt-soil^{-1} from soil moisture flow

I_{37j+1} = thickness, of the (i+1)th soil layer, mm (input)

I_{37j-1} = thickness, of the (i-1)th soil layer, mm (input)

I_{360} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i-1)th layer (top layer excluded) by soil moisture flow

I_{361} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i+1)th layer (bottom layer excluded) by soil moisture flow

$X_{12,i}$ = contaminant concentration, gdwt-soil^{-1}, of the ith soil layer

$X_{12,i+1}$ = contaminant concentration, gdwt-soil^{-1}, of the (i+1)th soil layer

$X_{12,i-1}$ = contaminant concentration, gdwt-soil^{-1}, of the (i-1)th soil layer

86. $F(X_{12,i}, X_{18g}) = 1400\, I_{37i}\, I_{363}\, X^2_{12,i} \,/ \sum_{i=1}^{I_{20}} 1400\, I_{37i}\, X_{12,i}$

$F(X_{12,i}, X_{18g})$ = transport of volatile contaminant to the atmosphere, ug day-1, from the ith soil layer of a given region
$I_{37j,j}$ = thickness, of the ith soil layer, mm, of the jth plant type (input)
I_{363} = mean loss of contaminant, ug day^{-1} m^{-2}, from a given region averaged over all plant types and soil horizons
$X_{12,i}$ = volatile contaminant concentration, ug gdwt-$soil^{-1}$, of the ith soil layer
I_{20} = number of soil layers specified for the ith plant (input or estimated)

87. $F(P_{sr}, X_{13,3,1}) = I_{369\,i}$

I_{369i} = transport of contaminant, m^{-2} day^{-1}, from the ith soil layer to the root compartment of a given plant
P_{sr} = soil-to root source term for a given contaminant
$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type

88. $F(P_{rs}, X_{13,3,1}) = I_{372}\, X_{13,4,1}\, I_{373}$

89. $F(P_{rs} + P_{sw} + P_{ss}, X_{13,4,1}) = I_{372}\, [X_{13,3,1}\, I_{374} + (X_{13,5,1} + X_{13,6,1}) I_{375}]$
$= I_{372}\, [X_{13,3,1}\, I_{374}/2 + (X_{13,5,1} + X_{13,6,1}) I_{375}]$ (NO_3^- or soluble PO_4^{-3})

90. $F(P_{sw}, X_{13,5,1}) = I_{372}\, X_{13,4,1}\, I_{376}$

91. $F(P_{ss}, X_{13,6,1}) = I_{372}\, X_{3,4,1}$

$F(P_{rs}, X_{13,3,1})$ = transport of contaminant, m^{-2} day^{-1}, from stem compartment of a given plant type to the root compartment
$F(P_{rs} + P_{sw} + P_{ss}, X_{13,4,1})$ = transport of contaminant, m^{-2} day^{-1}, from the root, leaf and seed compartments to the stem compartment of a given plant type
$F(P_{sw}, X_{13,5,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the leaf compartment of a given plant type
$F(P_{ss}, X_{13,6,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the seed or fruit compartment of a given plant type
I_{372} = ratio of net daily photosynthate production, to standing live leaf biomass, day^{-1}
I_{373b} = adjusted biomass density factor: stems to total biomass
I_{374} = above ground biomass density factor: stems, leaves, and seeds to total biomass
I_{375} = biomass density factor: stems to above ground biomass
I_{376b} = adjusted biomass density factor: leaves to above ground biomass
$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type
$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type
$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type
$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type
P_{rs} = root-stem contaminant exchange source
P_{sw} = stem-leaf contaminant exchange source
P_{ss} = stem-seed contaminant exchange source

92. $F(fM_a, X_{17,2,n}) = X_{f17,2}\, [X_{17,1,n} / X_{v17,1} - X_{17,2,n} / (X_{v17,2}\, X_{r17,2,n})]\, I_{378}$

$F(fM_a, X_{17,2,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and muscle for the nth contaminant

fM_a = transport of contaminant from blood plasma to muscle tissue, hr^{-1}
$X_{17,2,n}$ = contaminant content of muscle tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,2}$ = perfusion rate, liters hr^{-1}, of blood plasma into muscle tissue
$X_{fl7,i}$ = perfusion rate, liters hr^{-1} , of the ith organ or tissue
$X_{vl7,1}$ = volume of blood plasma, liters
$X_{vl7,2}$ = perfusion volume of muscle tissue, liters
$X_{rl7,2,n}$ = partition coefficient for the nth contaminant in muscle tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

93. $F(fF_a, X_{17,3,n}) = X_{fl7,3}\ [X_{17,1,n} / X_{vl7,1} - X_{17,3,n} / (X_{vl7,3}\ X_{rl7,3,n})]\ I_{378}$

$F(fF_a, X_{17,3,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and adipose tissue for the nth contaminant
fF_a = transport of contaminant from blood plasma to muscle tissue, hr^{-1}
$X_{17,3,n}$ = contaminant content of adipose tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,3}$ = perfusion rate, liters hr^{-1}, of blood plasma into adipose tissue
$X_{vl7,1}$ = volume of blood plasma, liters
$X_{vl7,3}$ = perfusion volume of adipose tissue, liters
$X_{rl7,3,n}$ = partition coefficient for the nth contaminant in adipose tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

94a. $F(fR_a, X_{17,4,n}) = X_{fl7,4}\ [X_{17,1,n} / X_{vl7,1} - X_{17,4,n} / (X_{vl7,4}\ X_{rl7,4,n})]\ I_{378}$

94b. $F(fM_{ka}, X_{wnl7,1,n}) = X_{fl7,4}\ [X_{17,1,n} / (X_{vl7,1}\ X_{fl7,1})]$

$F(fR_a, X_{17,3,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and residual tissue for the nth contaminant
$F(fM_{ka}, X_{wnl7,1,n})$ = nth transport of contaminant, hr^{-1}, from residual tissue to milk
$X_{wnl7,1,n}$ = nth contaminant transport, mass or activity units day^{-1}, from mother's milk to a weaning newborn animal of a given animal population
fR_a = transport of contaminant from blood plasma to residual tissue, hr^{-1}
$X_{17,4,n}$ = contaminant content of residual tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,4}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue
$X_{fl7,1}$ = total blood plasma flow, liters hr^{-1}
$X_{vl7,1}$ = volume of blood plasma, liters
$X_{vl7,4}$ = perfusion volume of residual tissue, liters
$X_{rl7,4,n}$ = partition coefficient for the nth contaminant in residual tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

95a. $F(fB_a, X_{17,9a,n}) = 2.0\, X_{fl7,9}\, [X_{17,1,n} / X_{v17,1} - X_{17,9an} / (X_{v17,9}\, X_{rl7,9a,n})]\, I_{378}$

95b. $F(fB_a, X_{17,9b,n}) = 2.0\, X_{fl7,9}\, [X_{17,1,n} / X_{v17,1} - X_{17,9bn} / (X_{v17,9}\, X_{rl7,9b,n})]\, I_{378}$

$F(fB_a, X_{17,9a,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and bone marrow for the nth contaminant

$F(fB_a, X_{17,9b,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and trabecular bone for the nth contaminant

fB_a = transport of contaminant from blood plasma to bone tissue, hr^{-1}

$X_{17,9a,n}$ = contaminant content of bone marrow tissue, mass or activity units, for the nth contaminant

$X_{17,9b,n}$ = contaminant content of trabecular bone tissue, mass or activity units, for the nth contaminant

$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant

$X_{fl7,9}$ = perfusion rate, liters hr^{-1}, of blood plasma into both marrow and trabecular tissue

$X_{fl7,i}$ = perfusion rate, liters hr^{-1}, of the ith organ or tissue

$X_{v17,1}$ = volume of blood plasma, liters

$X_{v17,9a}$ = perfusion volume of bone marrow tissue, liters

$X_{v17,9b}$ = perfusion volume of trabecular bone tissue, liters

$X_{rl7,9a,n}$ = partition coefficient for the nth contaminant in bone marrow tissue (input)

$X_{rl7,9b,n}$ = partition coefficient for the nth contaminant in trabecular bone tissue (input)

I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

96a. $F(fK_a, X_{17,10a,n}) = X_{fl7,10}\, [X_{17,1,n} / X_{v17,1} - X_{17,10an} / (X_{v17,10}\, X_{rl7,10a,n})]\, I_{378}$

96b. $F(fK_a, X_{17,10b,n}) = X_{fl7,10}\, [X_{17,1,n} / X_{v17,1} - X_{17,10bn} / (X_{v17,10}\, X_{rl7,10b,n})]\, I_{378}$

$F(fK_a, X_{17,10a,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and cortical tissue of the kidney for the nth contaminant

$F(fK_a, X_{17,10b,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and medullary tissue of the kidney for the nth contaminant

fK_a = transport of contaminant from blood plasma to kidney tissue, hr^{-1}

$X_{17,10a,n}$ = contaminant content of kidney cortical tissue, mass or activity units, for the nth contaminant

$X_{17,10b,n}$ = contaminant content of kidney medullary tissue, mass or activity units, for the nth contaminant

$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant

$X_{fl7,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into both cortical and medullary tissue

$X_{v17,1}$ = volume of blood plasma, liters

$X_{v17,10a}$ = perfusion volume of kidney cortical tissue, liters

$X_{v17,10b}$ = perfusion volume of kidney medullary tissue, liters

$X_{rl7,10a,n}$ = partition coefficient for the nth contaminant in kidney cortical tissue (input)

$X_{rl7,10b,n}$ = partition coefficient for the nth contaminant in kidney medullary tissue (input)

I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

97. $F(X_{17,1,n}, X_{u17,11,n}) = X_{f17,10}\ X_{17,1,n} / X_{f17,1}$

$F(X_{17,1,n}, X_{u17,11,n})$ = transport of contaminant from blood plasma, hr^{-1}, to urine
$X_{f17,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into kidney tissue for a given animal and age group
$X_{f17,1}$ = total plasma flow, liters hr^{-1}
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{17,1,n}$ = contaminant content, mass or activity units, in the blood plasma
$X_{u17,11,n}$ = contaminant content, mass or activity units, in urine

98. $F(fL_a + fG_v, X_{17,12,n}) = X_{f17,12}\ [X_{17,1,n} / X_{v17,1} - (X_{17,12,n} + I_{379}) / (X_{v17,12}\ X_{r17,12,n})]\ I_{378}$

$F(fL_a + fG_v, X_{17,12,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and liver tissue including gut transport for the nth contaminant
fL_a = transport of contaminant from blood plasma to liver tissue, hr^{-1}
fG_v = transport of contaminant from blood plasma to liver tissue from the gut via the hepatic portal vein, hr^{-1}
$X_{17,12,n}$ = contaminant content of liver tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{f17,12}$ = perfusion rate, liters hr^{-1}, of blood plasma into liver tissue
$X_{v17,1}$ = volume of blood plasma, liters
$X_{v17,12}$ = perfusion volume of liver tissue, liters
$X_{r17,12,n}$ = partition coefficient for the nth contaminant in liver tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

99. $F(fV_{a,g}, X_{3e17,6}) = \{[X_{b17,6,m,k}\ X_{18g} + X_{f17,1}\ X_{17,1} / (5.5e\text{-}05\ (1000\ X_{14,m,k})^{0.99})] / (X_{f17,1} + 1000\ X_{b17,6,m,k} / X_{r17,1,n})\}\ 5.5e\text{-}05\ (1000\ X_{14,m,k})^{0.99}\ I_{244j}\ (I_{389k,j} + I_{390k,j} + I_{391k,j}) / \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$

$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region
$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region
I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)
I_{273} = number of regions simulated (input)
$X_{b17,6,m,k}$ = inhalation rate, $m^3\ day^{-1}$, for the mth age group of the kth animal type
X_{18g} = regional vapor concentration, m^{-3}, for a given contaminant
$X_{f17,1}$ = total plasma flow, liters hr^{-1}
$X_{17,1}$ = contaminant content of blood plasma, mass or activity units
$X_{r17,1,n}$ = partition coefficient for the nth contaminant between plasma and ambient air in alveolar tissue (input)
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

100. $F(X_{18r}, X_{1e17,6}) = I_{392} X_{18r} X_{b17,6,m,k} I_{244j} (I_{389k,j} + I_{390k,j} + I_{391k,j}) / \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$

101. $F(X_{18r}, X_{1e17,6}) = I_{393} X_{18r} X_{b17,6,m,k} I_{244j} (I_{389k,j} + I_{390k,j} + I_{391k,j}) / \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$

102. $F(X_{18r}, X_{1e17,6}) = I_{394} X_{18r} X_{b17,6,m,k} I_{244j} (I_{389k,j} + I_{390k,j} + I_{391k,j}) / \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$

$X_{b17,6,m,k}$ = inhalation rate, m^3 day^{-1}, for the mth age group of the kth animal type
$X_{18r,m}$ = regional particulate concentration, m^{-3}, for the mth contaminant
$X_{1e17,6,n}$ = nth contaminant content on the surface of the bronchus region of the lung lumen as particulates
$X_{2e17,6,n}$ = nth contaminant content on the surface of the bronchiole region of the lung lumen as particulates
$X_{3e17,6,n}$ = nth contaminant content on the surface of the pulmonary region of the lung lumen as particulates
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region
$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region
I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)
I_{273} = number of regions simulated (input)
I_{392} = deposition fraction of resuspendable air particulates in the bronchus region of the lung, day^{-1} (input, default = 0.1)
I_{393} = deposition fraction of resuspendable air particulates in the bronchiole region of the lung, day^{-1} (input, default=0.1)
I_{394} = deposition fraction of resuspendable air particulates in the pulmonary region of the lung, day^{-1} (input, default=0.05)

II. Net Flows Constructed From Listed Individual Flows

1.	$\Delta X_{1,1} = F_2 + F_3 - F_4 - F_8 - F_9$	(moisture flow, top soil layer, mm day^{-1})
2.	$\Delta X_{1,i+1} = F_7$	(moisture flow, lower soil layer, day^{-1})
3.	$\Delta X_{1,i} = - F_7 - F_{10} - F_{11}$ (or F_{12})	(moisture flow, lower soil layer, day^{-1})
4.	$\Delta X_{1,1} = - F_{18}$	(soil erosion, tons $acre^{-1}$ day^{-1})
5.	$\Delta X_{1,1i} = - F_{19}$	(soil erosion, lbs day^{-1}, by particle size)
6.	$\Delta X_{1,1t} = - F_{20}$	(soil erosion, lbs day^{-1}, total, all sizes)
7.	$\Delta X_2 = F_1 - F_3$	(moisture flow, snow, top soil layer, day^{-1})
8.	$\Delta X_{3,1} = F_{24} - F_{30} - F_{31}$	(biomass flow, g m^{-2} day^{-1})
9.	$\Delta X_{3,2} = F_{30} - F_{39}$	(biomass flow, g m^{-2} day^{-1})
10.	$\Delta X_{3,3} = F_{31} - F_{40}$	(biomass flow, g m^{-2} day^{-1})
11.	$\Delta X_{4,1} = F_{25} - F_{24} - F_{26} - F_{28}$	(biomass flow, g m^{-2} day^{-1})
12.	$\Delta X_{4,2} = F_{28} - F_{34} - F_{35} - F_{36}$	(biomass flow, g m^{-2} day^{-1})
13.	$\Delta X_{4,3} = F_{36} - F_{37} - F_{38}$	(biomass flow, g m^{-2} day^{-1})
14.	$\Delta X_{4,4} = F_{34} - F_{37}$	(biomass flow, g m^{-2} day^{-1})
15.	$\Delta X_{4,5} = F_{35} - F_{38}$	(biomass flow, g m^{-2} day^{-1})
16.	$\Delta X_{5,1} = F_{23} - F_{24} - F_{25} - F_{26} - F_{27}$	(biomass flow, g m^{-2} day^{-1})

17.	$\Delta X_{5,1} = - F_5$	(moisture flow, mm day^{-1})
18.	$\Delta X_{5,1} = - F_{45}$	(moisture flow, mm day^{-1}, niche option)
19.	$\Delta X_{5,2} = F_{27} - F_{32}$	(biomass flow, g m^{-2} day^{-1})
20.	$\Delta X_{6,1} = F_{26} - F_{29}$	(biomass flow, g m^{-2} day^{-1})
21.	$\Delta X_{6,2} = F_{29} - F_{33}$	(biomass flow, g m^{-2} day^{-1})
22.	$\Delta X_{7,1} = F_{32} + F_{33} + F_{39} + F_{40} - F_{41} - F_{43}$	(biomass flow, g m^{-2} day^{-1})
23.	$\Delta X_{7,2} = F_{37} + F_{38} - F_{42} - F_{44}$	(biomass flow, g m^{-2} day^{-1})
24.	$\Delta X_8 = F_{15} - F_{13}$	(moisture flow, ft^3 day^{-1}, channel icemelt)
25.	$\Delta X_8 = - F_{17}$	(moisture flow, ft^3 sec^{-1}, runoff)
26.	$\Delta X_9 = F_{14} - F_{15}$	(moisture flow, ft^3 day^{-1}, channel ice)
27.	$\Delta X_{10i} = F_{16}$	(moisture flow, ft^3 day^{-1}, aquifer segment)
28.	$\Delta X_{11i} = F_{46} - F_{47} - F_{48}$	(biomass flow, ft^3 yr^{-1}, wood, by dia. class)
29.	$\Delta X_{11t} = - F_{49}$	(biomass flow, ft^3, total wood removal)
30.	$\Delta X_{r12,1} = F_{65} + F_{71b} + F_{72b} + F_{73b} + F_{74b}$	(contaminant flow, m^{-2} day^{-1}, resuspension, top soil layer)
31.	$\Delta X_{s12,1} = F_{80b} + F_{81b} + F_{82b} + F_{83b}$	(contaminant flow, m^{-2} day^{-1}, saltation-creep, top soil layer)
32.	$\Delta X_{12,1} = \Delta X_{r12,1} + \Delta X_{s12,1} - F_{84}$	(contaminant flow, m^{-2} day^{-1}, all processes, top soil layer)
33.	$\Delta X_{12,2} = F_{84}$	(contaminant flow, m^{-2} day^{-1}, all processes, top soil layer)
34.	$\Delta X_{12,i} = - F_{85a} - F_{86}$	(contaminant flow, m^{-2} day^{-1}, all processes, upper soil layer)
35.	$\Delta X_{12,i+1} = F_{85a}$	(contaminant flow, m^{-2} day^{-1}, all processes, lower soil layer)
36.	$\Delta X_{12,2,1} = F_{52}$	(nitrate flow, g g^{-1} soil day^{-1}, top layer)
37.	$\Delta X_{12,2,i} = F_{55} - F_{51} - F_{54}$	(nitrate flow, g m^{-2} day^{-1}, all soil layers)
38.	$\Delta X_{12,3,1} = F_{53}$	(ammonia flow, g g^{-1} soil day^{-1}, top layer)
39.	$\Delta X_{12,3,i} = - F_{51} - F_{55}$	(ammonia flow, g m^{-2} day^{-1}, all soil layers)
40.	$\Delta X_{12,5,i} = - F_{50}$	(phosphate flow, g m^{-2} day^{-1}, all soil layers)
41.	$\Delta X_{13,3,1,4} = F_{50}$	(phosphate flow, g m^{-2} day^{-1}, internal)
42.	$\Delta X_{13,3,1,3} = F_{51}$	(ammonia flow, g m^{-2} day^{-1}, internal)
43.	$\Delta X_{13,3,1} = F_{87} + F_{88}$	(contaminant flow, m^{-2} day^{-1}, internal)
44.	$\Delta X_{r13,4,1} = F_{68} - F_{72a}$	(ammonia flow, g m^{-2} day^{-1}, all soil layers)
45.	$\Delta X_{13,4,1} = F_{89}$	(contaminant flow, m^{-2} day^{-1}, internal)
46.	$\Delta X_{r13,4,2} + \Delta X_{r13,6,2} = F_{69}$	(contaminant flow, m^{-2} day^{-1}, resuspension)
47.	$\Delta X_{ra13,4,2} + \Delta X_{ra13,5,2} + \Delta X_{ra13,6,2} = - F_{73a} - F_{73b}$	(contaminant flow, m^{-2} day^{-1}, resuspension, weighted average)
48.	$\Delta X_{r13,4,3} + \Delta X_{13r,4,4} + \Delta X_{r13,4,5} = F_{70}$	(contaminant flow, m^{-2} day^{-1}, resuspension)
49.	$\Delta X_{ra13,4,3} + \Delta X_{13ra,4,4} + \Delta X_{ra13,4,5} = - F_{74a} - F_{74b}$	(contaminant flow, m^{-2} day^{-1}, resuspension, weighted average)
50.	$\Delta X_{r13,5,1} = F_{66a} - F_{71a} - F_{71b}$	(contaminant flow, m^{-2} day^{-1}, resuspension)
51.	$\Delta X_{r13,5,2} = F_{66b}$	(contaminant flow, m^{-2} day^{-1}, resuspension)
52.	$\Delta X_{s13,4,1} = F_{77a} + F_{77b} - F_{81a} - F_{81b}$	(contaminant flow, m^{-2} day^{-1}, saltation-creep)
53.	$\Delta X_{s13,4,2} + \Delta X_{s13,6,2} = F_{78}$	(contaminant flow, m^{-2} day^{-1}, saltation-creep)
54.	$\Delta X_{sa13,4,2} + \Delta X_{sa13,5,2} + \Delta X_{sa13,6,2} = - F_{82a} - F_{82b}$	(contaminant flow, m^{-2} day^{-1}, saltation-creep, integral average)
55.	$\Delta X_{s13,4,4} + \Delta X_{13,4,5} = F_{79}$	(contaminant flow, m^{-2} day^{-1}, saltation-creep)
56.	$\Delta X_{13,5,1} = F_{90}$	(contaminant flow, m^{-2} day^{-1}, internal)

57. $X_{s13,5,1} + \Delta X_{s13,6,1} = F_{76a} + F_{75a} - F_{80a} - F_{80b}$ (contaminant flow, m^{-2} day^{-1}, saltation-creep)

58. $\Delta X_{s13,5,2} = F_{75b}$ (contaminant flow, m^{-2} day^{-1}, saltation-creep)

59. $\Delta X_{13,6,1} = F_{91}$ (contaminant flow, m^{-2} day^{-1}, internal)

60. $\Delta X_{14,m,k} = F_{56e}$ (biomass flow, kgdwt day^{-1}, expected)

61. $\Delta X_{14,m,k} = F_{57} - F_{59} - F_{60} - F_{61}$ (biomass flow, kgdwt day^{-1}, net)

62. $\Delta X_{15,m,k} = F_{62} - F_{58}$ (biomass flow, kgfwt day-1, metabolic

63. $\Delta X_{17,1,n} = - F_{97} - F_{94a,b} - F_{92} - F_{93} - F_{94} - F_{95a,b} - F_{96a,b} - F_{98}$ (contaminant transport, hr^{-1})

64. $\Delta X_{wn17,1} = F_{94b}$ contaminant transport, hr^{-1}, weaning)

65. $\Delta X_{17,2,n} = F_{92}$ (contaminant transport, hr^{-1})

66. $\Delta X_{17,3,n} = F_{93}$ (contaminant transport, hr^{-1})

67. $\Delta X_{17,4,n} = F_{94a}$ (contaminant transport, hr^{-1})

68. $\Delta X_{1e17,6,n} = F_{100}$ (contaminant transport, hr^{-1}, particulate)

69. $\Delta X_{2e17,6,n} = F_{101}$ (contaminant transport, hr^{-1}, particulate)

70. $\Delta X_{3e17,6,n} = F_{99} + F_{102}$ (contaminant transport, hr^{-1}, particulate

71. $\Delta X_{17,9a,n} = F_{95a}$ (contaminant transport, hr^{-1})

72. $\Delta X_{17,9b,n} = F_{95b}$ (contaminant transport, hr^{-1})

73. $\Delta X_{17,10a,n} = F_{96a}$ (contaminant transport, hr^{-1})

74. $\Delta X_{17,10b,n} = F_{96b}$ (contaminant transport, hr^{-1})

75. $\Delta X_{u17,11,n} = F_{97}$ (contaminant transport, hr^{-1}, urine)

76. $\Delta X_{17,12,n} = F_{98}$ (contaminant transport, hr^{-1})

77. $\Delta X_{18p,m} = F_{63}$ (contaminant transport, m^{-3} day^{-1}, particulate)

78. $\Delta X_{18g,m} = F_{64}$ (contaminant transport, m^{-3} day^{-1}, volatile)

79. $\Delta X_{18r} = - F_{65} - F_{66a,b} - F_{67} - F_{68} - F_{69} - F_{70} - F_{100} - F_{101} - F_{102} + F_{72a} + F_{73a} + F_{74a}$ (contaminant transport, m^{-2} day^{-1}, resuspendable)

80. $\Delta X_{18g} = F_{86}$ (contaminant transport, ug day^{-1}, volatile)

81. $\Delta X_{18s} = - F_{75a,b} - F_{76} - F_{77a,b} - F_{78} - F_{79} - F_{80a} - F_{81a} + F_{82a} + F_{83a} + F_{83a}$ (contaminant transport, m^{-2} day^{-1}, saltation-creep)

11.0. APPENDIX B

EQUATIONS INVOLVING SECONDARY INTERMEDIATE VARIABLES

I. EQUATIONS INVOLVING INTERMEDIATE VARIABLES

1. $I_{17} = (1.28 I_3 I_{15}) / (I_3 + I_{16})$

   I_{17} = potential evapotranspiration (PET) from soil surface, mm/day
   I_3 = slope of psychrometric saturation line, mbars/ ° C
   I_{15} = net daily solar radiation absorbed by land surface at mean air temperature, Langelys
   I_{16} = psychrometric constant, mbars/ ° C

2. $I_3 = [5304 / (D_2 + 273)^2] \exp[21.255 - 5304 / (D_2 + 273)]$

   I_3 = slope of psychrometric saturation line, mbars / ° C
   D_2 = mean daily temperature, ° C
   I_3 = see above

3. $I_{15} = (1 - C_2) D_3$

   I_{15} = net solar radiation, ly, absorbed by the regional land surface
   C_2 = albedo of site (set in code to 0.2)
   D_3 = solar radiation at site, Langleys/ day

4. $I_{16} = 0.0006595 I_{18}$

   I_{16} = psychrometric constant, mbars / ° C
   I_{18} = site barometric pressure, mbars

5. $I_{18} = 1013 - 0.1152 (0.3048 I_{12}) + 0.00000544 (0.3048 I_{12})^2$

   I_{12} = site elevation, feet, amsl

6. I_{19} = total leaf area index (TLAI), m^2/m^2

   $I_{19} = 0.013 X_{5,1}$

   $X_{5,1}$ = plant leaf biomass density, gdwt/ m^2

7. $I_{21} = 2 I_{27} / I_{28}$

   I_{21} = root penetration fraction for moisture withdrawal from a given soil layer
   I_{27} = vertical root penetration into soil from the surface, mm
   I_{28} = depth to the bottom of a given soil layer, mm

8. $I_{27} = \sum_{j=1}^{I_{29}} \sum_{i=1}^{365} C_4$

   I_{27} = vertical extent of root penetration into a given soil, mm
   I_{29} = number of years simulated, years (input)

C_4 = mean root growth rate in the vertical direction, mm/day

9. $I_{30} = 1.56e06\ I_{32}^{5.27}$ rock
$= 1.56e05\ I_{32}^{5.27}$ sand
$= 6.99e04\ I_{32}^{6.4}$ silt
$= 1.04e04\ I_{32}^{9.2}$ clay

I_{30} = soil moisture hydraulic conductivity (κ), mm /day
I_{32} = the soil volumetric wetness (θ), mm/mm, for a given soil layer.

10. $I_{31} = 50\ I_{32}^{0.5}$ rock
$= 50\ I_{32}^{1.5}$ sand
$= 40\ I_{32}^{2.6}$ silt
$= 41.4\ I_{32}^{4.1}$ clay

I_{31} = soil moisture matric potential (ψ), mm / mm
I_{32} = soil volumetric wetness (θ), mm /mm

11. $\underline{I}_{30} = I_{33}\ 156e06\ I_{32}^{5.27} + I_{34}\ 1.56e05\ I_{32}^{5.27} + I_{35}\ 6.99e04\ I_{32}^{6.4} + I_{36}\ 1.04e04\ I_{32}^{9.2}$

$\underline{I}_{30}$ = weighted mean soil hydraulic conductivity ($\underline{\kappa}$) , mm /day
I_{32} = soil volumetric wetness (θ)
I_{33} = rock fraction in soil (input)
I_{34} = sand fraction in soil (input)
I_{35} = silt fraction in soil (input)
I_{36} = clay fraction in soil (input)

12. $\underline{I}_{31} = I_{33}\ 50\ I_{32}^{0.5} + I_{34}\ 50\ I_{32}^{1.5} + I_{35}\ 40 I_{32}^{2.6} + I_{36}\ 41.4\ I_{32}^{4.1}$

$\underline{I}_{31}$ = weighted mean soil matric potential (ψ), mm/ mm
I_{32} = soil volumetric wetness (θ)
I_{33} = rock fraction in soil (input)
I_{34} = sand fraction in soil (input)
I_{35} = silt fraction in soil (input)
I_{36} = clay fraction in soil (input)

13. $I_{38} = I_{39} - I_{40}I_{41} - I_{42}C_3$

I_{38} = the normal soil moisture condition curve number
I_{39} = plant crown cover factor
I_{40} = plant crown cover-conductivity interaction factor, hours/ inch
I_{41} = soil moisture infiltration rate, inches/ hour (input)
I_{42} = plant crown cover -vegetation type interaction factor, $year^{-1}$
C_3 = mean life-span of plant type, years

14. $I_{39} = 96 - 16I_{43}$

15. $I_{40} = 9 + 5.9I_{43}$

16. $I_{42} = 0.8 + 0.04I_{43}$

I_{43} = plant fractional crown cover, m^2/m^2

17. $I_{43} = [\,1 - \exp(-0.00154\, I_{44})]$ $\qquad L_2 \le 1$

$$= \sum_{i=1}^{I_{45}} 0.0929\pi\,(I_{46} I_{47i} / 2)^2\,(I_{48i} / 4047) \qquad L_2 > 1$$

I_{43} = plant fractional crown cover, m^2/m^2
I_{44} = 1 growing season cumulative net productivity, gdwt / m^2/ day $\qquad C_3 = 1$
= 10 year running average net cumulative productivity, gdwt / m^2/ day $\qquad C_3 > 1$
I_{45} = current number of tree diameter classifications simulated on a given year/day
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (inches)
I_{47i} = mean of ith bole diameter class, inches
I_{48i} = number of trees in ith bole diameter class

18. $I_{49} = -16.91 + 1.348\, I_{38} - 0.01379\, I_{38}^2 + 0.0001177\, I_{38}^3$

I_{49} = the dry antecedent soil moisture condition curve number
I_{38} = the normal soil moisture condition curve number

19. $I_{49} = [\,100\, I_{49}\,(1 - I_{50}) + 9900\, I_{50}\,] / 100$

I_{49} = corrected dry antecedent soil moisture condition curve number
I_{50} = fraction of land surface covered by impervious material or bedrock

20. $I_{51} = 25.4\,[\,(1000/ I_{49}) - 10]$

I_{51} = maximum soil moisture retention parameter, mm
I_{49} = dry antecedent soil moisture curve number

21. $$I_{52} = \sum_{i=2}^{I_{20}} (I_{51} I_{53i}\,[\,(I_{23i} - X_{1,i}) / I_{23i}\,]$$

I_{20} = total number of soil layers in the rooting zone of a given plant type (input)
I_{51} = maximum soil moisture retention parameter, mm
I_{52} = soil moisture retention parameter over all soil layers, mm
I_{53i} = ith soil layer weighting factor
I_{23i} = ith soil layer holding capacity, mm
$X_{1,i}$ = ith soil layer moisture content, mm

22. $I_{53i} = I_{54} \exp(-4.16\, I_{55i} / I_{56})$

23. $$I_{54} = \sum_{i=2}^{I_{37i}} [\,\exp(-4.16 I_{55i} / I_{56})]^{-1}$$

I_{53i} = ith soil layer weighting factor
I_{54} = a parameter which normalizes the sum of all soil weighting factors to 1.0
I_{55i} = ratio of the depth to bottom of the ith soil layer to the bottom of the soil profile
I_{56} = total depth, mm, of soil profile for the rooting zone of a given plant type (input)

24. $I_{59j} = 5280\ I_{61j}\ I_{62j}\ I_{63}$

I_{59j} = crossectional area, ft^2, for return flow into a channel from the jth plant type
I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{62j} = total depth, feet, of the soil horizon for the jth plant type in a given region
I_{63} = total length, miles, of a given region along a receiving channel (input)

25. $I_{61j} = (I_{43j} + 0.01 / I_{64}) / I_{65}$

I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{43j} = crown cover fraction for jth plant type in a given region
I_{64} = number of plant types in a region excluding those where the area specified is not equal to the regional area such as for irrigated or non-irrigated plants under cultivation
I_{65} = total cover fraction for all plant types considered (I_{64}) in a given region

26. $I_{66i} = (X_{1,i} - I_{58i})\ I_{67j} / 304.8$

I_{66i} = total water available in ith soil layer with a jth plant type above field capacity, ft^3
$X_{1,i}$ = total soil moisture, mm, in the ith soil layer of the jth plant type in a given region
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of the jth plant type in a given region (input)
I_{67j} = estimated crown cover area, ft^2, of the jth plant in a given region

27. $I_{67j} = 43562\ I_{61j}\ I_{68}$

I_{61j} = ratio of crown cover fraction of the jth plant type to the total cover of all plant types in a given region
I_{68} = area, acres, of the region containing the jth plant type (input)

28. $I_{71i} = \exp (- I_{73i}\ I_{63i}\ I_{74i})$

I_{71i} = regression slope factor for the ith channel or channel segment
I_{63i} = length of ith channel or channel segment , miles (input)
I_{74i} = width of ith channel or channel segment, feet (input)

29. $I_{73i} = -1.09\ ln (1 - I_{79i})$

I_{73i} = decay factor for transmission losses, $(mi\text{-}ft)^{-1}$ for the ith channel or channel segment
I_{79i} = decay factor coefficient, in/ acre-ft, for the ith channel or channel segment

30. $I_{79i} = 0.00545\ I_{41i}\ I_{75i} / (I_{76i}\ I_{68i} / 12)$

I_{79i} = decay factor coefficient, in/ acre-ft, for the ith channel or channel segment
I_{41i} = channel hydraulic conductivity, inches/ hour, for the ith channel or channel segment

I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)
I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours
I_{76i} = potential mean runoff volume into the ith channel or channel segment, inches

31. $I_{75i} = I_{84i} \; (I_{68i} / 640)^{I85i}$

I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours
I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)
I_{84i} = duration equation coefficient for ith channel or channel segment (input)
I_{85i} = duration equation exponent for ith channel or channel segment (input)

32. $I_{76i} = 0.05 \; (I_{68i} / 640)^{-2}$

I_{76i} = potential mean runoff volume into the ith channel or channel segment, inches
I_{68i} = area, acres, of the region containing the ith channel or channel segment (input)

33. $I_{70i} = [I_{77i} / (1 - I_{78i})] (1 - I_{71i})$

I_{70i} = regression intercept, acre-ft, for the ith channel or channel segment
I_{71i} = regression slope factor for the ith channel or channel segment
I_{77i} = unit channel intercept for transmission loss, acre-ft, for the ith channel or channel segment
I_{78i} = unit channel regression slope for transmission loss for the ith channel or channel segment

34. $I_{77i} = 0.00465 \; I_{41i} \; I_{75i}$

35. $I_{78i} = \exp (- I_{71i})$

I_{77i} = unit channel intercept for transmission loss, acre-ft, for the ith channel or channel segment
I_{78i} = unit channel regression slope for transmission loss for the ith channel or channel segment
I_{41i} = channel hydraulic conductivity, inches/ hour, for the ith channel or channel segment
I_{75i} = estimated duration of a runoff event in the ith channel or channel segment, hours
I_{71i} = regression slope factor for the ith channel or channel segment

36. $I_{72i} = (1 - I_{71i}) / (I_{73i} \; I_{74i})$

I_{72i} = lateral flow parameter for transmission, mi , for the ith channel or channel segment
I_{71i} = regression slope factor for the ith channel or channel segment
I_{73i} = decay factor for transmission losses, $(\text{mi-ft})^{-1}$ for the ith channel or channel segment
I_{74i} = width of ith channel or channel segment, feet (input)

37. $I_{80i} = 2640 \; (I_{81i} + I_{82i}) \; I_{63i}$

I_{80i} = surface area of channel or channel segment, ft^2
I_{63i} = length of ith channel or channel segment , miles (input)
I_{81i} = width of aquifer, ft, at proximal end of ith channel or channel segment
I_{82i} = width of aquifer, ft, at distal end of ith channel or channel segment

38. $I_{102} = I_{84} I_{85}^{3} + I_{86} I_{85}^{2} + I_{87} I_{85}$

I_{102} = maximum aquifer volume, ft^3
I_{84} = cubic coefficient
I_{85} = linear distance between proximal and distal ends of aquifer, ft
I_{86} = quadratic coefficient, ft
I_{87} = linear coefficient, ft^2

39. $I_{85} = (I_{88}^{2} + I_{89}^{2})^{1/2}$

I_{85} = linear distance between proximal and distal ends of aquifer, ft
I_{88} = vertical distance, ft, to the bottom of the aquifer at the proximal end relative to the bottom of the aquifer at the distal end of a channel or channel segment
I_{89} = horizontal distance, ft, between proximal and distal ends of a channel or channel segment

40. $I_{89} = [(5280 I_{63})^{2} - I_{90}^{2}]^{1/2}$

I_{89} = horizontal distance, ft, between proximal and distal ends of a channel or channel segment
I_{63} = length of the given channel or channel segment , miles (input)
I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment

41. $I_{90} = 5280 I_{63} I_{60}$

I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment
I_{63} = length of the given channel or channel segment , miles (input)
I_{60} = fractional slope of a given channel or channel segment (input)

42. $I_{88} = I_{91} - I_{92}$

I_{88} = vertical distance, ft, to the bottom of the aquifer at the proximal end relative to the bottom of the aquifer at the distal end of a channel or channel segment
I_{91} = vertical distance, ft, between bottom of aquifer at distal end relative to the surface elevation of the aquifer at the proximal end
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)

43. $I_{91} = I_{93} + I_{90}$

I_{91} = vertical distance, ft, between bottom of aquifer at distal end relative to the surface elevation of the aquifer at the proximal end
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{90} = difference in elevation, ft, between proximal and distal ends of aquifer that is below a given channel or channel segment

44. $I_{84} = (I_{93} - I_{92}) [(I_{82} - I_{81}) + (I_{95} - I_{94})] / (6 I_{85}^{2})$

I_{84} = cubic coefficient

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)

45. $I_{86} = \{ I_{93} [I_{81} + I_{94} - 2 (I_{95} + I_{82})] + I_{92} (I_{82} + I_{95}) \} / (4 I_{85})$

I_{86} = quadratic coefficient, ft
I_{81} = width of aquifer, ft, at proximal end below a given channel or channel segment (input)
I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)
I_{85} = linear distance between proximal and distal ends of aquifer, ft
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)
I_{94} = width, ft, of the bottom of the aquifer at the proximal end below a given channel or channel segment (input)
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

46. $I_{87} = I_{93} (I_{82} + I_{95}) / 2$

I_{87} = linear coefficient, ft^2
I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

47. $I_{98} = I_{85} (I_{100} I_{99}{}^2 + I_{101} I_{99})$

I_{98} = aquifer volume, ft3, at a given height, ft, above the bottom
I_{85} = linear distance between proximal and distal ends of aquifer, ft
I_{99} = estimated aquifer height, ft, for a given aquifer volume estimated using Newton's iterative approximation method from a known volume (I_{98}) , ft^3
I_{100} = quadratic coefficient
I_{101} = linear coefficient, ft

48. $I_{100} = (I_{82} - I_{95}) / (4 I_{93}) + (I_{81} - I_{94}) / (4 I_{92})$

49. $I_{101} = (I_{95} + I_{94}) / 2$

I_{100} = quadratic coefficient
I_{101} = linear coefficient, ft
I_{81} = width of aquifer, ft, at proximal end of a given channel or channel segment (input)
I_{82} = width of aquifer, ft, at distal end below a given channel or channel segment (input)
I_{92} = vertical height of aquifer at proximal end, ft, below a given channel or channel segment (input)

I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)
I_{94} = width, ft, of the bottom of the aquifer at the proximal end below a given channel or channel segment (input)
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)

50. $I_{99n} = I_{99n} - I_{103} / I_{104}$

I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of the above equation is close to 0.0
I_{103} = difference between the estimated aquifer volume using I_{99n} as the estimated height of the aquifer at the distal end and the known volume (I_{98})
I_{104} = first derivative estimate of volume equation (I_{98}) using aquifer estimated height at the distal end (I_{99n})

51. $I_{103} = I_{98n} - I_{98}$

52. $I_{104} = 2\, I_{85} \,(I_{100}\, I_{99n} + I_{101})$

I_{103} = difference between the estimated aquifer volume using I_{99n} as the estimated height of the aquifer at the distal end and the known volume (I_{98})
I_{104} = first derivative estimate of volume equation (I_{98}) using aquifer estimated height at the distal end (I_{99n})
I_{98n} = aquifer volume estimate, ft^3, for the nth iteration at which convergence of equation (50) above is close to 0.0 using (I_{99n}) as the estimated height
I_{98} = aquifer volume, ft^3, at an unknown height, ft, above the bottom
I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0
I_{85} = linear distance between proximal and distal ends of aquifer, ft

53. $I_{105} = I_{99n} \,(I_{95} + I_{106}) / 2$

I_{105} = cross sectional area, ft^2, of the distal vertical cross section of the aquifer
I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)
I_{106} = width of distal end of aquifer, ft, at the estimated height (I_{99n}) above the bottom of the aquifer

54. $I_{106} = I_{95} + (I_{82} - I_{95}) / (I_{93}\, I_{99n})$

I_{106} = width of distal end of aquifer, ft, at the estimated height (I_{99n}) above the bottom of the aquifer
I_{95} = width, ft, of the bottom of the aquifer at the distal end below a given channel of channel segment (input)
I_{99n} = aquifer height estimate, ft, at the distal end of the aquifer for the nth iteration at which convergence of equation (50) above is close to 0.0
I_{82} = width of aquifer, ft, at distal end of the given channel or channel segment (input)
I_{93} = vertical height of aquifer at distal end, ft, below a given channel or channel segment (input)

55. $I_{112i} = 2\, I_{114i}\, I_{115}\, I_{116}^{\,2}$ (i = 1, suspended load)

$= I_{114i} I_{115} I_{117i} (I_{115} - I_{118i})$ (i > 1, bedload)

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed
I_{114i} = mass fraction of ith particle diameter classification for soil or sediment from a given region or channel bed (initial input)
I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed
I_{116} = The mean velocity of surface runoff, ft /s, for a given region or channel
I_{117i} = soil or sediment transport coefficient, ft^3 /lb-s, for the ith diameter class for soil or sediment from a given region or channel bed
I_{118i} = critical shear stress, lbs /ft^2, for the ith diameter class for soil or sediment from a given region or channel bed

56. $I_{117i} = 40 / I_{128i}^{1.5}$

I_{117i} = soil or sediment transport coefficient, ft^3 /lb-s, for the ith diameter class for soil or sediment from a given region or channel bed
I_{128i} = particle diameter, mm, for the ith diameter class of soil or channel sediment for a given region or channel bed (input)

57. $I_{118i} = 0.0022 + 0.01 I_{128i}$ $\quad I_{128i} \leq 1$ mm
$= 0.0078 + 0.02 I_{128i}$ $\quad I_{128i} > 1$ mm

I_{118i} = critical shear stress, lbs /ft^2, for the ith diameter class for soil or sediment from a given region or channel bed
I_{128i} = diameter, mm, of ith diameter class for a given regional or channel surface (input)

58. $I_{115} = 62.43 I_{119} I_{60}$

I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed
I_{60} = fractional slope of regional land surface or channel (input)
I_{119} = hydraulic radius, ft, for resistance of particles on the surface of regions or channels

59. $I_{119} = I_{120} (I_{121} / I_{122})^{1.5}$

I_{119} = hydraulic radius, ft, for resistance of particles on the surface of regions or channels
I_{120} = hydraulic radius for bed resistance, ft, for a given region or channel
I_{121} = grain or particle resistance coefficient for particles on the surface of regions or channels
I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels

60. $I_{121} = 0.132 I_{127}^{0.166}$

I_{121} = grain or particle resistance coefficient for particles on the surface of regions or channels
I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

61. $I_{122} = [I_{124}^{0.66} + (2 I_{125} / I_{74}) (I_{124}^{0.66} - 1.32 I_{124}^{1.5})]^{0.66}$

$= 0.63\ I_{124}$ $\qquad I_{122} < 0.63\ I_{124}$

$= I_{126}$ $\qquad I_{122} > I_{126}$

62. $I_{126} = I_{124}\ [\ (\ I_{74} + 4\ I_{125}\ )\ /(\ 4\ I_{125})]^{0.66}$

I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels
I_{126} = hydraulic roughness of the area of the channel or regional surface perpendicular to the direction of flow
I_{74} = width of a given channel or channel segment, feet (input)
I_{124} = roughness or Manning coefficient for a given region or channel (input)
I_{125} = depth of surface runoff flow, ft, from a given region or channel

63. $I_{125} = (F_8 / 25.4\ )\ /12$ (for regions)
$= F_{13} / (43200\ I_{75})$ (for channels)

I_{125} = depth of surface runoff flow, ft, from a given region or channel
I_{75} = estimated duration of a runoff event for a given channel or channel segment, hours
F_8 = runoff event , mm /m^2 /day, from a given region
F_{13} = channel runoff, ft^3 / day

64. $I_{120} = I_{123}\ (I_{122} / I_{124}\ )^{1.5}$

I_{120} = hydraulic radius, ft, for bed resistance for a given region or channel
I_{122} = hydraulic roughness of the fluid bed for on the surface of regions or channels
I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels
I_{124} = roughness or Manning coefficient for a given region or channel (input)

65. $I_{123} = I_{74} I_{125} / (I_{74} + 2\ I_{125}\ )$

I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels
I_{125} = depth of surface runoff flow, ft, from a given region or channel
I_{74} = width of a given channel or channel segment, feet (input)

66. $I_{116} = (\ 1.49 / I_{124}\ )\ I_{60}^{0.5}\ I_{123}^{0.66}$

I_{116} = The mean velocity of surface runoff, ft /s, for a given region or channel
I_{124} = roughness or Manning coefficient for a given region or channel (input)
I_{60} = fractional slope of regional land surface (input)
I_{123} = hydraulic radius, ft, of surface moisture flow for regions or channels

67. $I_{129} = I_{130} / (\ \sum_{i=2}^{7} I_{112i}\ )$

I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed
I_{130} = transport capacity , lbs /ft-s, for the median particle diameter, mm (I_{127})

68.

$$I_{113} = I_{112,1} + I_{129} \sum_{i=2}^{7} I_{112i}$$

I_{113} = sum of all particle transport capacities, lbs /s-ft, for all diameter classes from a given region or channel bed normalized to the median particle diameter for a given soil or sediment

$I_{112,1}$ = transport capacity , lbs /s-ft, of the suspended load particles ($\leq$ 0.063 mm)

I_{129} = normalization factor to align transport capacities with the transport capacity of the median particle diameter of the soil or sediment media under consideration

I_{112i} = particle transport capacity, lbs /s-ft, for the ith particle diameter classification from a given region or channel bed

69. $I_{130} = I_{131}\, I_{115}\, (I_{115} - I_{132})$

I_{130} = transport capacity , lbs /ft-s, for the median particle diameter, mm (I_{127})

I_{115} = effective shear stress for soil or sediment transport, lbs / ft^2, for a given region or channel bed

I_{131} = particle transport coefficient, ft^3 / lb-s, for the median particle diameter, mm (I_{127})

I_{132} = critical shear stress, lbs /ft^2, for the median for the median particle diameter, mm (I_{127})

70. $I_{131} = 40 / I_{127}^{\ 1.5}$

I_{131} = particle transport coefficient, ft^3 / lb-s, for the median particle diameter, mm (I_{127})

I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

71. $I_{132} = 0.0022 + 0.01\, I_{127}$ $\qquad I_{127} < 1.0$ mm

$\quad = -0.0078 + 0.02\, I_{127}$ $\qquad I_{127} \geq 1.0$ mm

I_{132} = critical shear stress, lbs /ft^2, for the median for the median particle diameter, mm (I_{127})

I_{127} = median particle diameter, mm, of soil or channel sediment for a given region or channel bed (input)

72. $I_{135} = 210. - I_{140} / 2$

73. $I_{136} = 210 + I_{140} / 2$

I_{135} = beginning of frost-free period for the region, Julian day number

I_{136} = end of frost-free period for the region, Julian day number

I_{140} = frost-free period, days (input)

74. $I_{133} = 0.001\, D_3$

$\quad = 0.45$ $\qquad I_{133} > 0.45$

I_{133} = photosynthetic efficiency, g-photosynthate /ly -day-g-leaf

D_3 = total solar radiation intensity at site, Langleys / day

75. $I_{134} = 1 - (D_2 - C_5)^2 / C_6^{\ 2}$

$\quad = 1 - (D_2 - C_5)^2 / C_7^{\ 2}$ $\qquad D_2 > C_5$

$\quad = 0$ $\qquad I_{134} \leq 0$

I_{134} = effect of temperature on photosynthesis
D_2 = mean daily temperature, °C
C_5 = optimum temperature for photosynthesis for a given plant type, °C (database)
C_6 = difference between optimum and lowest temperature tolerance for photosynthesis of a given plant type, °C (database)
C_7 = difference between highest temperature tolerance and the optimum temperature for photosynthesis for a given plant type (database)

76. $I_{141} = C_8 F_{21} / I_{142}$

I_{141} = moisture requirement, ml /m^2 /day, to realize potential gross photosynthesis
F_{21} = potential gross photosynthesis, g/m^2/day, of a given plant type
C_8 = water utilization requirement, ml-moisture /g-photosynthate production (database)
I_{142} = effect of root soluble nitrate and phosphate concentrations on photosynthesis

77. $I_{142} = 1.32 \{ 1 - \exp[(- 0.693 / 0.014) I_{143}]\}\{ 1 - \exp[(- 0.693 / 0.014) I_{144}]\}$

I_{142} = effect of soluble root nitrate and phosphate concentrations on photosynthesis
I_{143} = soluble nitrate concentration, g-NO_3 / gdwt-root in metabolically active tissues
I_{144} = soluble phosphate concentration, g-PO4 / gdwt-root in metabolically active tissues

78. $I_{145} = 1000 F_5 I_{146} I_{147} I_{148}$

I_{145} = adjusted amount of moisture available for photosynthesis, ml /m^2 /day
F_5 = total transpiration, mm/m^2/ day, for the a given plant type
I_{146} = adjustment factor for leaf areas indices (LAI) below 1 m^2 /m^2
I_{147} = niche competition factor (> 0, ≤ 1)
I_{148} = unadjusted succession competition factor (> 0, ≤ 1)

79. $I_{146} = 1$ $\quad I_{19} \geq 1$
$= 1/I_{19}$ $\quad I_{19} < 1$

I_{146} = adjustment factor leaf areas indices below 1 m^2 /m^2
I_{19} = leaf area index, m^2/m^2, (LAI)

80. $I_{150} = I_{149}$
$= I_{149} / 2$ $\quad C_3 > 2$
$= 2 I_{149}$ $\quad L_4 = 1$

I_{150} = adjusted photosynthate fraction to fruiting body of a specific plant type
I_{149} = photosynthate transport fraction to fruiting body of a specific plant type
C_3 = mean life-span, years, of the plant type simulated
L_4 = evergreen tree and shrub gate (0,1)

81. $I_{149} = 0.0$ $\quad L_3 = 0$; or $I_{149} \leq 0.0$ (all plant types)
$I_{137} > I_{139}$ (irrigated plants)
(fruit trees, deciduous and evergreen)
(shrubs less than 5 years old; and)
(other deciduous and evergreen trees)
(less than 10 years old)
$= 0.95$ $\quad I_{149} \geq 0.95$ (all plant types)

$$= \sum_{i=I_{135}+1}^{239} I_{149} + 0.004 \qquad \text{(warm season plants)}$$

$$= \sum_{i=240}^{270} I_{149} - 0.004 \qquad \text{(warm season plants)}$$

$$= \sum_{i=I_{135}-30}^{179} I_{149} + 0.004 \qquad \text{(cool season plants)}$$

$$= \sum_{i=180}^{209} I_{149} - 0.004 \qquad \text{(cool season plants)}$$

$$= \sum_{i=I_{138}+1}^{I139} I_{149} + 0.004 \qquad \text{(irrigated plants)}$$

I_{149} = photosynthate transport fraction to fruiting body of a specific plant type
L_3 = photosynthetic gate (0,1)
I_{137} = day counter during the growing season, day number (may exceed 365 for some crops)
I_{139} = harvest date, day number (may exceed 365 for some crops) (input)
I_{135} = beginning of frost-free period for the region, Julian day number
I_{138} = planting date , Julian day number (input)

82. $I_{44} = I_{151} \qquad C_3 = 1$
$= I_{152} / I_{153} \qquad C_3 > 1$

I_{44} = 10 year running average net cumulative productivity, gdwt / m^2/ year (cumulative daily productivity for annual plants)
F_{23i} = net photosynthesis, g/m^2/day, for a given plant type on the ith day
I_{135} = beginning of frost-free period for the region, Julian day number
I_{136} = end of frost-free period for the region, Julian day number
C_3 = mean life-span, years, of the plant type simulated
I_{151} = cumulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
I_{152} = cumulative net productivity for 1 year, g /m2/ year: the sum of current and previous year totals (previous year is running average between 1 - 10 years)
I_{153} = number of years of net productivity used in estimating (I_{153}) above (1 - 10 years)

83.

$$I_{151} = \sum_{i=1}^{365} F_{23i}$$

I_{151} = cumulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
F_{23i} = net photosynthesis, g/m^2/day, for a given plant type on the ith day

84.

$$I_{152} = \sum_{i=1}^{n} I_{151} \qquad 1 \le n \le 10$$

85.

$$I_{153} = \sum^{n} i \qquad 1 \le n \le 10$$

i=1

I_{152} = cumulative net productivity for 1 year, g /m2/ year: the sum of current and previous year totals (previous year is running average between 1 - 10 years)
I_{151} = cumulative daily net productivity, g /m^2 /day, (from 1 - 365 days inclusive)
I_{153} = number of years of net productivity used in estimating (I_{153}) above (1 - 10 years)

86. $I_{155} = 0.004 + 0.0032\, I_{157}$

I_{155} = effect of soil surface temperature on litter (dry biomass) decomposition
I_{157} = estimated temperature of soil surface, ° C

87. $I_{157} = D_2 - 4\,(X_{5,1} - 300)\,/\,600 \qquad X_{5,1} > 300$
$\quad = D_2 + 15\, I_{158}\,(1 - X_{5,1}\,/\,300) \qquad X_{5,1} \le 300$

I_{157} = estimated temperature of soil surface. ° C
D_2 = mean daily temperature, ° C
$X_{5,1}$ = live leaf biomass density, g /m^2
I_{158} = actual to potential evaporation ratio

88. $I_{158} = 1 - (F_5\, I_{19} + F_4\,/\,I_{17}) \qquad I_{19} \le 1$
$\quad = 1 - (F_5 + F_4\,/I_{17}) \qquad I_{19} > 1$

I_{158} = actual to potential evaporation ratio
I_{17} = potential evapotranspiration (PET) from soil surface, mm/day
F_5 = total transpiration, mm/m^2/ day, for the jth plant
F_4 = transport of moisture from soil surface (evaporation), mm/ m^2 / day
I_{19} = leaf area index, m^2/m^2, (LAI)

89. $I_{156} = [1\,/\,(I_{58} - I_{22})]\,(X_{1,1} - I_{22})$

I_{156} = effect of soil moisture on litter decomposition
I_{58} = total soil moisture present at field capacity, mm, for the 1st soil layer of the jth plant type in a given region (input)
I_{22} = soil moisture content, mm, for the 1st soil layer of the jth plant at wilt point (input)
$X_{1,1}$ = soil moisture in surface layer, mm

90. $I_{159} = (100\, X_{4,6} + 10\, X_{4,7})\,/\,(X_{4,6} + X_{,4,7})$

I_{159} = weighting factor for estimating litter formation from stems (boles) and branches
$X_{4,6}$ = branch-twig component of live above ground biomass, g /m^2
$X_{4,7}$ = bole component of live above ground biomass, g /m^2

91. $I_{160} = \pi\, I_{162}\, I_{163}^{\,2}\, I_{164}\,/\,3$

I_{162} = estimated plant areal density, number / m2 for a given plant type at a given time
I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{164} = estimated height of a given plant type, cm, at a given time

92. $I_{162} = I_{165}\,/\,4047 \qquad L_2 > 1$
$\quad = 1 \qquad L_2 \le 1$

93. $I_{163} = 2.54\, I_{166} \qquad L_2 > 1$

$= 0.003\ X_{4,1}$ $\qquad L_2 \leq 1$
$= 0.1$ $\qquad I_{163} \leq 0.1$

94. $I_{164} = 308.4\ I_{167}$ $\qquad L_2 > 1$
$= 0.1\ X_{4,1}$ $\qquad L_2 \leq 1$
$= 7$ $\qquad I_{164} < 7$

95. $I_{161} = I_{168}$ $\qquad L_2 > 1$
$= 0.2$ $\qquad L_2 \leq 1$

I_{161} = estimated density of dry plant tissue, g / m^3, for a given plant
I_{162} = estimated plant areal density, number / m2 for a given plant type at a given time
I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{164} = estimated height of a given plant type, cm, at a given time
L_2 = forest structure access gate (1,2,3)
I_{165} = tree density, trees /acre, for a given plant type (input, or see App. B,I, No. xx)
$X_{4,1}$ = live stem biomass density, g /m^2 /d
I_{166} = mean bole diameter, inches, for a stand of a given plant (see App. B.,I, No. xx)
I_{167} = mean height, cm, for a stand of a given plant (see App. B., I, No. xx)
I_{168} = wood density, g /cm3, of a given plant type (input)

96. $I_{169i} = 0.004 + 0.0032\ I_{172i}$

I_{169i} = effect of soil temperature of the ith soil on root litter decomposition
I_{172i} = estimated temperature of ith soil layer, °C

97. $I_{172i} = 12.78 + (D_2 - 12.78) \exp[(0.693 / 396)(I_{173i} - 0.5\ I_{37i})]$

I_{172i} = estimated temperature of ith soil layer, °C
D_2 = mean daily temperature, °C
I_{37i} = thickness of ith soil layer, mm
I_{173i} = depth to the bottom of the ith soil layer, mm

98. $I_{173i} = \sum_{j=1}^{i} I_{37j}$

I_{173i} = depth to the bottom of the ith soil layer, mm
I_{37j} = thickness of jth soil layer, mm

99. $I_{170i} = 1 + [1 / (I_{58i} - I_{22i})](X_{1,i} - I_{22i})$
$= 1$ $\qquad X_{1,i} < I_{22i}$

I_{170i} = effect of soil moisture of the ith soil layer on root litter decomposition
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of a given plant type (input)
I_{22i} = soil moisture content, mm, for the ith soil layer of a given plant at wilt point (input)
$X_{1,i}$ = soil moisture in the ith layer, mm

100. $I_{171i} = I_{21i}\ I_{174i}$
$= 1$ $\qquad C_3 = 1$

I_{171i} = root penetration fraction into the ith soil layer

I_{21i} = root penetration fraction (0 - 1) of a given plant into the ith soil layer
I_{174i} = fraction of root biomass in ith soil layer of a given plant (input or estimated)
C_3 = mean life-span, years, of the plant type simulated

101. $I_{174i} = (I_{175i} - I_{175i-1}) / I_{176}$

I_{174i} = fraction of root biomass in ith soil layer of a given plant (estimated)
I_{175i} = cumulative root fraction up to the ith soil layer
I_{175i-1} = cumulative root fraction up to the (i-1)th soil layer
I_{176} = maximum soil penetration factor for a given soil depth (I_{56})

102. $I_{175i} = [1 - \exp(- 0.004\ I_{177i})] / 0.632$

I_{175i} = cumulative root fraction up to the ith soil layer
I_{177i} = depth to bottom of ith soil layer, mm

103. $I_{177i} = \sum_{j=1}^{i} I_{37j}$

I_{177i} = depth to bottom of ith soil layer, mm
I_{37j} = thickness of the jth soil layer, mm

104. $I_{176} = [1 - \exp(-0.004\ I_{56})] /0.632$

I_{176} = maximum soil penetration factor for a given soil depth (I_{56})
I_{56} = total depth, mm, of soil profile for the rooting zone of a given plant type (input)

105. $I_{178} = 0.1\ C_{16} \exp(C_{17}\ I_{172i})$

I_{178} = mineralization factor for readily mineralized humus
I_{172i} = estimated temperature of ith soil layer, °C
C_{16} = mineralization coefficient
C_{17} = mineralization exponential constant, $°C^{-1}$

106. $I_{147} = F_{23} / X_{5,1}$ — $O_1 = 1;\ I_{179} > 0$
$= 1.0$ — $X_{5,1} = 1$
$= 0.1$ — $I_{147} \leq 0.1$

I_{147} = moisture or niche competition factor, day^{-1}
F_{23} = net photosynthesis, $g/m^2/day$, for a given plant type
$X_{5,1}$ = live leaf compartment, g/m^2
I_{179} = niche competition match number (input)
O_1 = niche competition option (1,0) (input)

107. $I_{148} = 1$ — $O_5 = 1$
$= 1$ — $O_2 = 0;\ O_{2a} = 0$
$= 1$ — $I_5 = 1, I_{29} = 1, C_3 < 30;\ O_2 = 1; O_{2a} = 0$
$= 0$ — $I_5 = 1, I_{29} = 1, C_3 \geq 30;\ O_2 = 1; O_{2a} = 0$

$= \ln (X_{4,1\,m} + X_{3,1} + 0.7183)\ I_{43\,m}^{2}\ \ I_{167m}^{0.6}$ — $I_5 = 1, I_{29} > 1;\ O_2 = 1$ or $O_{2a} = 1$

$I_{43m} = 0.3163$ — $I_{43m} < 0.3163$

$I_{167m} = I_{167}$ $\quad L_2 > 1$

$= X_{4,1m} / 360$ $\quad L_2 = 1$

$= 0.022$ $\quad I_{167m} < 0.022$

I_{148} = succession competition factor (> 0, ≤ 1)
$X_{4,1\ m}$ = maximum live above ground biomass (stem) compartment, g/ m^2 for a given year
$X_{3,1}$ = live root compartment, g/ m^2 on Julian day 365
$I_{43\ m}$ = maximum plant fractional crown cover, m^2/m^2, for a given year
I_{43} = plant fractional crown cover, m^2/m^2
I_{29} = Julian year number of specific year of simulation
I_{167m} = mean or maximum height, ft, for a stand of a given plant
I_{167} = mean height, ft, for a stand of a given plant (see App. B., I, No. xx)
O_5 = irrigation access option (1,0) (input)
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)
L_2 = forest structure access gate (1,2,3) (input)

108. $I_{148in} = I_{148j} / \sum_{j=1} I_{148j}$

$= 0.97$ $\quad I_{148in} > 0.97$

$= 0.01$ $\quad I_{148in} < 0.01$

$= 1$ $\quad X_{4,1} = 1$

I_{148jn} = normalized succession factor for the jth plant type
I_{148j} = unadjusted succession competition factor (> 0, ≤ 1) for the jth plant type
n = number of plants undergoing succession

109. $I_{148ja} = 1$ $\quad I_{29}=1;\ O_2=1;O_{2a}=0$

$I_{148jp} = 0$ $\quad 30 \le C_3 < 70;\ I_{29}= 1O_2=1;O_{2a}=0$

$I_{148js} = 0$ $\quad X_{4,1}=1;\ 70 \le C_3 < 150;\ I_{148ja} > I_{148jp} / 3\ O_2=1;O_{2a}=0$

$I_{148j}t = 0$ $\quad X_{4,1}=1; 150 \le C_3;\ I_{148jp} > I_{148js} / 3O_2=1;O_{2a}=0$

I_{148ja} = composite normalized succession factor for the jth annual plant type
I_{148jp} = composite normalized succession factor for the jth perennial plant type
I_{148js} = composite normalized succession factor for the jth shrub plant type
I_{148jt} = composite normalized succession factor for the jth tree plant type
C_3 = mean life-span, years, of the plant type simulated
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
I_{29} = Julian year number of specific year of simulation
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)

110. $I_{148jt} = 0.7$ $\quad I_5=1;I_{29}=1;O_2=0;\ O_{2a}=1;O_5=0$

$= 0.7$ $\quad I_5=1;\ I_{148jt} < 0.7$

$I_{148js} = 0.2$ $\quad I_5=1;I_{29}=1;O_2=0;\ O_{2a}=1;\ C_3 \le 70;O_5=0$

$= 0.2$ $\quad I_5=1;\ I_{148js} < 0.2$

$I_{148jp} = 0.1$ $\quad I_5=1;I_{29}=1;O_2=0;\ O_{2a}=1;\ C_3 \le 30;O_5=0$

$= 0.1$ $\quad I_5=1;\ I_{148jp} < 0.1$

$I_{148ja} = 0.1$ $\quad I_5=1;I_{29}=1;O_2=0;\ O_{2a}=1;\ C_3 \le 30;O_5=0$

$= 0.1$ $\quad I_5=1;\ I_{148ja} > 0.1$

O_5 = irrigation access option (1,0) (input)
O_2 = succession option (1,0) (input)
O_{2a} = advanced succession option (0,1) (input)
I_5 = Julian day number (1 - 365)
I_{29} = Julian year number of specific year of simulation
I_{148ja} = composite normalized succession factor for the jth annual plant type
I_{148jp} = composite normalized succession factor for the jth perennial plant type
I_{148js} = composite normalized succession factor for the jth shrub plant type
I_{148jt} = composite normalized succession factor for the jth tree plant type

111. $I_{180i} = 0.2212\, I_{46}^2\, I_{48}$ $\quad I_{47i} = 1$
$= I_{46}\, I_{47i}^2\, I_{48i}\, I_{181}$ $\quad I_{47i} > 1$

I_{180i} = diameter class competition factor for the ith diameter class
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (input)
I_{47i} = mean of ith bole diameter class, inches
I_{48i} = number of trees in ith bole diameter class
I_{181} = diameter class ingrowth factor for the ith diameter class

112. $I_{47i} = (2i - 1.0)\, I_{182} / 2$

113. $I_{181} = 1 - \exp[-0.5(2i - 1)]$ $\quad i > 1$

I_{47i} = mean of ith bole diameter class, inches
I_{182} = diameter range, inches, within a given diameter class (input)
i = diameter class, 1,2,3,...n
I_{181} = diameter class ingrowth factor for the ith diameter class

114. $I_{48i} = X_{11i} / I_{183i}$

I_{48i} = number of trees in ith bole diameter class
X_{11i} = total volume of trees in the ith diameter class, ft^3
I_{183i} = volume of a unit tree in the ith diameter class, ft^3

115. $I_{183i} = [\,(0.035 I_{184} / I_{168})\,(2.54\, I_{182})^2\,] / 2$ $\quad i = 1$

$= [0.035\, I_{184} / I_{168}\,(\{2.54\, I_{182}(i - 1)\} + \{2.54\, i\, I_{182}\}^2)] / 2$ $\quad i > 1$

I_{183i} = volume of a unit tree in the ith diameter class, ft^3
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{182} = diameter range, inches, within a given diameter class (input)
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)

116. $I_{185} = 0.142\, X_{4,1} / I_{168}$

I_{185} = total volume of tree stand ,ft^3 /acre
I_{168} = wood density, g /cm3, of a given plant type (input)
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2

117. $I_{187} = I_{185m} / I_{185\ m-1}$ $\qquad P_{n,v} < 0.0$

I_{187} = mortality factor for wood volume losses
I_{185m}, I_{m-1} = total volume of tree stand ,ft^3 /acre, for current (mth) and previous year(m-1th)

118. $I_{188} = \sum_{j=1}^{k} I_{180j}$

I_{188} = sum of all intra-class competition factors
I_{180j} = intra-class competition factor for the jth diameter class
k = total number of diameter classes (value varies during a simulation)

119. $I_{189i} = (307\ I_{184} / I_{168})\ (I_{47i} / 24)^{0.9}$

I_{189i} = mean tree height, ft, in the ith diameter class
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{47i} = mean of ith bole diameter class, inches

120. $I_{167} = \exp (I_{190} / I_{165})$

121. $I_{165} = \sum_{i=2}^{n} I_{48i}$

I_{167} = mean geometric height, ft, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
I_{190} = logarithmic sum of tree heights

122. $I_{190} = \sum_{i=2}^{n} I_{48i}\ ln\ (I_{189i})$

I_{190} = logarithmic sum of tree heights
I_{48i} = number of trees in ith bole diameter class
I_{189i} = mean tree height, ft, in the ith diameter class

123. $I_{166} = \exp(I_{191} / I_{165})$

I_{166} = geometric mean bole diameter, inches, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
I_{191} = logarithmic sum of tree diameters

124. $I_{191} = \sum_{i=2}^{n} I_{48i}\ ln\ (I_{47i})$

I_{191} = logarithmic sum of tree diameters
I_{48i} = number of trees in ith bole diameter class
I_{47i} = mean of ith bole diameter class, inches

125. $I_{192} = (307\ I_{184} / I_{168})\ (I_{166} / 24)^{0.9}$

I_{192} = mean tree height, ft, using the geometric mean bole diameter, inches
I_{184} = stem diameter (cm) to mass (kg) regression coefficient (input)
I_{168} = wood density, g /cm3, of a given plant type (input)
I_{166} = geometric mean bole diameter, inches, for a stand of a given plant

126. $I_{43} = (0.0929\pi / 4047) \sum_{i=2}^{n} I_{48i} (I_{47i}\ I_{46} / 2)^2$

I_{43} = plant fractional crown cover, m^2/m^2
I_{48i} = number of trees in ith bole diameter class
I_{47i} = mean of ith bole diameter class, inches
I_{46} = bole diameter to crown diameter regression coefficient, feet/ inch for all diameter classes (input)

127. $I_{193i} = 1 / \{15 + 185\ [\ 1 - \exp(-0.693\ I_{36i} / 3)\ ]\ \}$

I_{193i} = phosphate exchange fraction from labile to soluble form in the ith soil layer
I_{36i} = clay fraction in ith soil layer (input)

128. $I_{194i} = I_{193}\ I_{195i}\ I_{32i}$

I_{194i} = soluble phosphate concentration, g-P /gdwt-soil, in the ith soil layer
I_{195i} = labile phosphate concentration, g-P / gdwt-soil in the ith soil layer
I_{193} = phosphate exchange fraction from labile to soluble form in the ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer.

129. $I_{196i} = \{\ [1.635e\text{-}04/(\ 1 + 1.43 / I_{197i}\ )]\}+\{\ [3.27e\text{-}05/(\ 1 + 0.084 / I_{197i})]\}$

I_{196i} = unadjusted plant root uptake rate, g-P / g-root / m^2/day of inorganic phosphate in the ith soil layer
I_{197i} = ratio of labile phosphate in 1 m^3 of soil to the volumetric moisture ratio

130. $I_{197i} = 1.4e06\ I_{195i} / I_{32}$
$= 0$ $\qquad I_{32} = 0$
$= 0.00001$ $\qquad I_{32} > 0,\ I_{197i} \le 0.00001$

I_{197i} = ratio of labile phosphate in 1 m^3 of soil to volumetric moisture ratio in the ith soil layer
I_{195i} = labile phosphate concentration, g / gdwt-soil in the ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer

131. $I_{198i} = I_{196i}\ I_{199}\ (1 - I_{200})\ I_{201}\ I_{202}\ I_{203}$
$= 0$ $\qquad X_{3,1} \le 1.0$
$= 0$ $\qquad I_{198i} < 0$

I_{198i} = adjusted plant root uptake g-P/ g -root/ m^2 /day of inorganic phosphate for ith soil layer
I_{196i} = unadjusted plant root uptake rate, g-P / g-root / m^2/day of inorganic phosphate in the ith soil layer
I_{199i} = effect of soil temperature on uptake of phosphate for ith soil layer

I_{200i} = leakage fraction of phosphate from roots for ith soil layer
I_{201i} = effect of soil moisture on uptake of phosphate for ith soil layer
I_{202i} = evapotranspiration ratio effect on uptake of phosphate for ith soil layer
I_{203i} = effect of nitrate concentration in roots on uptake of phosphate for ith soil layer

132. $I_{199i} = 0.19 + 0.045\ I_{172i}$
$= 0.41 + 0.0175\ I_{172i}$ $\quad I_{172i} \leq 10$
$= 0.25\ I_{172i}$ $\quad I_{172i} \leq 2$
$= 1$ $\quad I_{172i} > 22$

I_{199i} = effect of soil temperature on uptake of phosphate for ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

133. $I_{200i} = \exp\{ [X_{13,i,4}\ X_{4,1} / (X_{5,1}\ X_{3,1}) - 0.001]\ 173.29\} - 1$
$= 0$ $\quad X_{13,4}\ X_{4,1} / (X_{5,1}\ X_{3,1}) < 0.001$

I_{200i} = phosphate leakage fraction from roots for ith soil layer
$X_{13,i,4}$ = phosphate content of root compartment, g-P/ m^2, of the ith soil layer
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2

134. $I_{201i} = 10\ (I_{32} - 0.073)$

$= 1$ $\quad I_{32} > 0.173$
$= 0$ $\quad I_{32} < 0.073$

I_{201i} = effect of soil moisture on uptake of phosphate for ith soil layer
I_{32i} = the soil volumetric wetness (θ), mm/mm, of the ith soil layer

135. $I_{202i} = F_{6i} / F_5$
$= 0$ $\quad F_{6a} = 0$

I_{202i} = transpiration ratio effect on uptake of phosphate for ith soil layer
F_5 = flow of moisture from all soil layers combined, mm/m^2/day, to the leaf surface via roots and stems (see App. A., No. 5)
F_{6i} = flow of moisture from the ith soil layer to the leaf surface via the plant roots and stems , mm/m^2/day (see App. A., No. 6)

136. $I_{203i} = 1.21\ \{ 1 - \exp[-0.693\ X_{13,i,3} / (0.014\ X_{3,1})]\}$
$= 0.001$ $\quad I_{203i} < 0.001$

I_{203i} = effect of nitrate-N concentration in roots on uptake of phosphate for ith soil layer
$X_{3,1}$ = live root compartment, g/ m^2
$X_{13,i,3}$ = live root nitrate and ammonium content , g-N /m^2 of the ith soil layer

137. $I_{204} = 0.5$
$= 1$ $\quad (C_3 \leq 50)$

I_{204} = uptake factor
C_3 = mean life-span, years, of the plant type simulated

138. $I_{205i} = 0.002 / (1 + 84 / I_{206i}) + 0.004 / (1 + 4.8 / I_{206i})$

139. $I_{207i} = 0.0002 / (1 + 84 / I_{208i}) + 0.00004 / (1 + 4.8 / I_{208i})$

I_{205i} = unadjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{207i} = unadjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, from the ith soil layer

140. $I_{206i} = 1.4e06\ I_{209i}$

141. $I_{208i} = 1.4e06\ I_{210i}$

I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, for the ith soil layer
I_{209i} = nitrate concentration, g-N/ g-soil, in the ith soil layer
I_{210i} = ammonium concentration, g-N/ g-soil, in the ith soil layer

142. $I_{211i} = (I_{205i} - I_{212i})\ I_{199i}\ I_{202i}\ I_{213i}$

143. $I_{214i} = (I_{207i} - I_{215i})\ I_{199i}\ I_{202i}\ I_{213i}$

I_{211i} = adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{214i} = adjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{205i} = unadjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{207i} = unadjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
I_{199i} = effect of soil temperature on uptake of nitrate and ammonium ions for ith soil layer
I_{202i} = transpiration ratio effect on uptake of nitrate and ammonium ions for ith soil layer
I_{212i} = nitrate ion back-flow rate, g-N /g-root / m^2 / day, to the ith soil layer
I_{215i} = ammonium ion back-flow rate, g-N /g-root / day, to the ith soil layer
I_{213i} = effect of matric potential (Ψ) on ammonium and nitrate uptake from the ith soil layer

144. $I_{212i} = I_{205i} \exp\{ [X_{13,i,3}\ X_{4,1} / (X_{3,1}\ X_{5,1}) - 0.006]\ I_{216i} \} - 1$

145. $I_{215i} = I_{207i} \exp\{ [X_{13,i,3}\ X_{4,1} / (X_{3,1}\ X_{5,1}) - 0.006]\ I_{217i} \} - 1$

I_{212i} = nitrate ion back-flow rate, g-N /g-root / m^2 / day, to the ith soil layer
I_{215i} = ammonium ion back-flow rate, g-N /g-root / day, to the ith soil layer
I_{205i} = un-adjusted nitrate ion uptake rate, g-N /g-root / m^2 / day, from the ith soil layer
I_{207i} = un-adjusted ammonium ion uptake rate, g-N /g-root / day, from the ith soil layer
$X_{13,i,3}$ = live root nitrate and ammonium content , g-N /m^2, of the ith soil layer
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
I_{216i} = nitrate ion soil density back-flow factor for the ith soil layer
I_{217i} = ammonium ion soil density back-flow factor for the ith soil lay

146. $I_{216i} = 35 \exp(-0.15\ I_{206i}) + 40$

147. $I_{217i} = 35 \exp(-0.15\ I_{208i}) + 40$

I_{216i} = nitrate ion soil density back-flow factor for the ith soil layer
I_{217i} = ammonium ion soil density back-flow factor for the ith soil layer
I_{206i} = estimated density of nitrate ion in the soil , g-N / m^3, for the ith soil layer
I_{208i} = estimated density of ammonium ion in the soil, g-N / m^3, for the ith soil layer

148. $I_{213i} = 0.4 - 0.0133\ I_{218i}$
$= 1 - 0.0533\ I_{218i}$ $\quad I_{218i} \le 15$
$= 0$ $\quad I_{218i} > 30$

I_{213i} = effect of matric potential (Ψ) on ammonium and nitrate ion uptake from the ith soil layer
I_{218i} = soil matric potential, Bars, for the ith soil layer

149. $I_{218i} = (0.6 - I_{219i})\ 0.3 / (0.6 - I_{220i})$ $\quad I_{219i} > I_{220i}$
$= 0.3 + (I_{220i} - I_{219i})\ 2.7 / (I_{220i} - 0.16)$ $\quad I_{219i} \le I_{220i}$
$= 3 + (0.16 - I_{219i})\ 12 / (0.16 - I_{221i})$ $\quad I_{219i} < 0.16$
$= 15 + (I_{221i} - I_{219i})\ 2000$ $\quad I_{219i} \le I_{221i}$
$= 0$ $\quad I_{218i} \le 0$
$= 15$ $\quad I_{218i} \ge 15$

I_{218i} = soil matric potential, Bars, for the ith soil layer
I_{219i} = available moisture volume, ml, of the ith soil layer
I_{220i} = available moisture volume, ml, of the ith soil layer at field capacity
I_{221i} = available moisture volume, ml, of the ith soil layer at wilt point

150. $I_{219i} = 1.4\ X_{1,i} / I_{37i}$

151. $I_{220i} = 1.4\ I_{58i} / I_{37i}$

152. $I_{221i} = 1.4\ I_{22i} / I_{37i}$

I_{219i} = available moisture volume, ml, of the ith soil layer
I_{220i} = available moisture volume, ml, of the ith soil layer at field capacity
I_{221i} = available moisture volume, ml, of the ith soil layer at wilt point
$X_{1,i}$ = ith soil layer moisture content, mm
I_{58i} = total soil moisture present at field capacity, mm, for the ith soil layer of the jth plant type in a given region (input)
I_{22i} = soil moisture content, mm, for the ith soil layer of the jth plant at wilt point (input)
I_{37i} = thickness of ith soil layer, mm

153. $I_{222} = I_{222p}\ X_{4,1} / X_{5,1} + 0.024 - 0.12\ I_{223}$
$= I_{222p}$ $\quad I_{223} > 0.2$
$= 0.06$ $\quad I_{122} > 0.06$
$= 0.006$ $\quad I_{122} \le 0.006$

I_{222} = root nitrogen concentration, g-N/gdwt, from NH_4^+, NO_3^- uptake
I_{222p} = previous root nitrogen concentration, g-N/ gdwt, from NH_4^+, NO_3^- uptake
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{5,1}$ = live leaf compartment, g/ m^2
I_{223} = fraction of growth period expended

154. $I_{223} = (I_5 - I_{135}) / (I_{136} - I_{135})$
$= 1$ $\quad I_5 > I_{136};\ I_5 < I_{135}$

I_5 = Julian day number (1 - 365)
I_{135} = beginning of frost-free period for the region, Julian day number
I_{136} = end of frost-free period for the region, Julian day number

155. $I_{224i} = 0.025 + 0.437 \exp(-0.00575\ I_{210i}\ 1.4e06)$

156. $I_{226i} = 0.0631 / (1.4e06\ I_{209i})^{0.4}$

157. $I_{225i} = (I_{224i}\ I_{210i} - I_{226i}\ I_{209i})\ I_{227i}\ I_{228i}$

$= 0 \qquad I_{225i} < 0$

I_{224i} = conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil /day for the ith soil layer
I_{226i} = conversion of nitrate ion to ammonium ion, g-NH_4/ gdwt-soil /day for the ith soil layer
I_{209i} = nitrate concentration, g-N/ g-soil, in the ith soil layer
I_{210i} = ammonium concentration, g-N/ g-soil, in the ith soil layer
I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer
I_{227i} = effect of temperature on soil nitrificaton in the ith soil layer
I_{228i} = effect of soil matric potential on soil nitrification in the ith soil layer

158. $I_{227i} = 0.20 + 0.043\ I_{172i}$

$= 0.0114\ I_{172i} \qquad I_{172i} \le 7$
$= 1 \qquad I_{172i} > 28$

I_{227i} = effect of temperature on soil nitrificaton in the ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

159. $I_{228i} = \exp(-0.693\ I_{218i} / 8)$

I_{228i} = effect of soil matric potential on soil nitrification in the ith soil layer
I_{218i} = soil matric potential, Bars, for the ith soil layer

160. $I_{229i} = 0.0033\ I_{225i}\ I_{230i}\ I_{231i}$
$= 0 \qquad I_{229i} < 0$

I_{229i} = denitrification rate, g-NO_3 / gdwt-soil /day, for the ith soil layer
I_{225i} = net conversion of ammonium ion to nitrate ion, g-NO_3/ gdwt-soil/day for the ith soil layer
I_{230i} = effect of soil temperature on denitrification for the ith soil layer
I_{231i} = effect of soil matric potential on denitrification for the ith soil layer

161. $I_{230i} = -0.31 + 0.03266\ I_{172i}$
$= 0.002\ I_{172i} \qquad I_{172i} \le 10$
$= 1 \qquad I_{172i} > 40$

I_{230i} = effect of soil temperature on denitrification for the ith soil layer
I_{172i} = estimated temperature of ith soil layer, °C

162. $I_{231i} = 1.25\,(1 - I_{218i})$

$= 0$ $\qquad I_{218i} > 1$

$= 1$ $\qquad I_{218i} \leq 0.2$

I_{218i} = soil matric potential, Bars, for the ith soil layer

I_{231i} = effect of soil matric potential on denitrification for the ith soil layer

163. $I_{242\,i} = 0.7\,(1 - I_{243})\,(I_{244b})_j$ $\qquad I_{240b} = 1, i=1$

$= 0.2\,(1 - I_{243})\,(I_{244b})_j$ $\qquad I_{240b} = 2, i=2$

$= 0.1\,(1 - I_{243})\,(I_{244b})_j$ $\qquad I_{240b} = 3, i=3$

$= I_{241k}\,(I_{244b})_j$ $\qquad i > 3$

164. $$I_{243} = \sum_{j=1}^{I_{274}} \sum_{k=1}^{I_{275k}} I_{241k}\,(I_{244b})_j$$

I_{274} = number of regions simulated (input)

I_{275k} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannibalism is being modeled) in a given region

I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)

I_{241k} = initial diet fraction of the kth animal prey type in the food of a specific animal predator type

I_{242i} = initial food fraction contribution from the ith preference group ($i \leq 3$) of plants of a given region

I_{243} = total fraction of food content derived from animal food from all regions combined

I_{244j}, $(I_{244b})_j$ = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

165. $$I_{245k} = \sum_{m=1}^{I_{248k}} F\,(P_f, X_{14,m,k})\, I_{246m,k}\, I_{247k}$$

I_{245k} = food requirement, kg /day, by the kth mammalian population

$F\,(P_f, X_{14,m,k})$ = transport of food, kg/day, to an animal of the mth age group of the kth mammalian population

$I_{246m,k}$ = fraction of the population in the mth age group of the kth mammalian population (initial input or default)

I_{247k} = total number of animals in the kth mammalian population (initial input)

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

166. $I_{255k,j} = I_{245k}\,(I_{244b})_j$

167. $I_{256k} = I_{245k}\,(1 - (I_{244b})_j)$

I_{245k} = food requirement, kg /day, by the kth mammalian population

$I_{255k,j}$ = range plants in food intake, kgdwt, for the kth mammalian population from the jth region

I_{256k} = harvested plant/animal in food intake, kgdwt, for the kth mammalian population from

I_{244j}, $(I_{244b})_j$ = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

168. $I_{257i,m} = [X_{5,1} + X_{6,1} + X_{6,2} + (X_{4,1} - X_{5,1}) X_{4,1} / X_{5,1}] I_{258i} + I_{259i} (X_{5,2} + X_{4,2})$

169. $I_{260m} = \sum_{i=1}^{n} I_{257i,m}$

170. $I_{258i} = 3 I_{261i} / (3 I_{261i} + I_{262i})$

171. $I_{259i} = 1 - I_{258i}$

172. $I_{261i} = [X_{5,1} + X_{6,1} + X_{6,2} + (X_{4,1} - X_{5,1}) X_{4,1} / X_{5,1}]$

173. $I_{262i} = X_{5,2} + X_{4,2} X_{5,1} / X_{4,1}$

$I_{257i,m}$ = total biomass density, gdwt/m^2, for the ith plant in the mth preference category
I_{258i} = live plant fraction (including seeds) ingested from the ith plant type
I_{259i} = dry plant fraction (excluding seeds) ingested from the ith plant type
I_{260m} = total biomass density, gdwt/m^2, in the mth preference category for range plants
I_{261i} = live plant biomass (including seeds), gdwt/m^2, for the ith plant
I_{262i} = dry plant biomass (excluding seeds), gdwt/m^2, for the ith plant
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

174. $I_{263i,m} = X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2}$ m=4; $I_{266i} = I_5$
$= X_{6,1} + X_{6,2}$ m=5; $I_{266i} = I_5$
$= X_{4,1} + X_{4,2} + X_{5,2}$ m=6; $I_{266i} = I_5$
$= X_{6,1} + X_{6,2}$ m=7; $I_{266i} = I_5$
$= 0$ $I_{263i,m} \leq 20$

175. $I_{264\,i,m} = 4047 (I_{263i,m} - 20) [I_{263i,m} / (X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2})] I_{265i}$

$I_{263i,m}$ = harvested biomass, gdwt/m^2, from the ith plant in the mth selection category for a given region
$I_{264i,m}$ = harvested biomass, gdwt, from the ith plant in the mth selection category for a given region
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
I_{266i} = harvesting date(s), Julian day, for the ith plant in a given region (input)
I_5 = Julian day number (1 - 365)
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

I_{248k}

176. $I_{260i} = \sum_{m=1} 500\, X_{14,m,k}\, I_{246m,k}\, I_{247k} / I_{267k}$ $\quad i > 3;\ I_{248k} > 0$

$= \sum_{m=1}^{I_{248k}} 1000\, X_{14,m,k}\, I_{246m,k}\, I_{247k} / I_{267k}$ $\quad i > 3;\ I_{248k} = 0$

I_{260i} = total biomass density, gfwt/m^2, in the ith selection category of animals
I_{267k} = area grazed, m^2, by the kth mammalian population in a given region
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth animal population
I_{246nmk} = fraction of the population in the mth age group of the kth mammalian population; set to 1 for invertebrate populations (initial input or default)
I_{247k} = total number of animals in the kth mammalian population ; invertebrate populations set to 1 (input)
I_{248k} = lifespan, years, of the an animal in the kth mammalian population; set to 0 for invertebrate populations (not used) (input)

177.
$I_{267k} = \sum_{i=1}^{n} 4047\, I_{268i}$ $\quad 0 \le I_{240b} \le 3$

I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)
I_{267k} = area grazed, m^2, by the kth animal population in a given region
I_{268i} = area, acres, of the ith plant type in a given region that is grazed by kth mammalian population in that region (input)

178. $I_{269kj} = \sum_{m=1}^{nr} I_{255k}\, I_{270k,j,m}\, I_{271k,j,m}$ $\quad 0 < j \le 3$

$I_{269k,j}$ = food contribution, kgdwt /day, from the jth range plant/animal category from all regions combined for the kth mammalian population
I_{255k} = range plant/animal in food intake, kgdwt /day, for the kth mammalian population from all regions combined
$I_{270k,j,m}$ = fraction of food available from the jth range plant/animal category for the mth region combined for the kth mammalian population
$I_{271k,j,m}$ = fraction of range plant/animal food contribution from the mth region of the jth range plant category for the kth mammalian population
nr = number of regions simulated

179. $I_{270k,j,m} = I_{242j,m}\,(\,I_{257j,m} - 2\,)\, I_{267k,j} / 1000\,) / (I_{255k}\, I_{242j,m}\,)$ $\quad 0 < j \le 3$

$= I_{242j,m}\,(\,I_{257j,m} - 0.0001\,)\, I_{267k,j} / 1000\,) / (I_{255k}\, I_{242j,m}\,)$ $\quad j > 3$

$= I_{242j,m}\,(\,I_{257j,m} - 0.02\,)\, I_{267k,j} / 1000\,) / (I_{255k}\, I_{242j,m}\,)$ $\quad j > 3;\ I_{248k} = 0$

$= I_{242j,m}$ $\quad I_{270k,j,m} \ge 1$

$I_{270k,j,m}$ = fraction of food available from the jth range plant category for the mth region combined for the kth mammalian population
I_{267k} = area grazed, m^2, by the kth mammalian population in a given region

I_{255k} = range plant/animal in food intake, kgdwt /day, for the kth mammalian population from all regions combined

$I_{242j,m}$ = food fraction contribution from the jth preference group of plants by the mth region

$I_{257i,j}$ = total biomass density, gdwt/m^2, for the ith plant in the jth preference category

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

180. $$I_{271k,j,m} = \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \Big/ \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \qquad 0 < I_{239bk,j,m} \leq 3$$

$$= \sum_{i=1}^{I_{274k}} I_{241k,i,m} \Big/ \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{274k}} I_{241k,i,m} \qquad 0 < I_{241k,i,m}$$

$I_{271k,j,m}$ = fraction of range plant/animal food contribution from the mth region of the jth range plant/animal category for the kth animal population

I_{273} = number of regions simulated (input)

$I_{274k,m}$ = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannibalism is being modeled) in the mth region

I_{69m} = number of plants simulated in the mth region (input)

$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith range plant in the mth region for the kth animal population

$I_{241k,i,m}$ = initial diet fraction of the ith animal prey type in the food of the kth predator animal for the kth predator population type

181a. $$I_{272k} = \sum_{m=1}^{I_{274}} I_{256k,m} \left[\sum_{i=1}^{I_{69m}} I_{239bk,i,m} \Big/ \sum_{m=1}^{I_{274}} \sum_{i=1}^{I_{69m}} I_{239bk,i,m} \right] \qquad 0 < I_{239bk,j,m} > 3$$

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

$I_{256k,m}$ = harvested plants in food intake, kgdwt, for the kth animal population from the mth region

$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith range plant in the mth region for the kth animal population

181b. $$I_{254k} = I_{272k} + \sum_{j=1}^{7} I_{269k,j}$$

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

$I_{269k,j}$ = food contribution, kgdwt /day, from the jth range plant/animal category from all regions combined for the kth mammalian population

182. $$I_{253k} = 0.75\, I_{275k} + 0.45\, (1 - I_{275k})$$

183a. $$I_{275kr} = \sum_{m=1}^{I_{273}} \sum_{j=1}^{3} I_{269j,m}\, I_{276jjm} / I_{254k}$$

183b. $$I_{275ka} = \sum_{m=1}^{I_{273}} \sum_{j=4}^{7} I_{269j,m} / I_{254k}$$

183c. $$I_{275ks} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} (X_{4,1} + X_{6,1} + X_{6,2})\, I_{239bk} / (X_{4,1} + X_{6,1} + X_{6,2} + X_{5,2} + X_{4,2})_{i,m} / I_{276k}$$

$I_{240b} = 4$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bk,m} / I_{276k}$$ $I_{240b} = 5$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} (X_{4,1}\ I_{239bk} / (X_{4,1} + X_{5,2} + X_{4,2})_{i,m} / I_{276k}$$ $I_{240b} = 6$

$$= \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bki,m} / I_{276k}$$ $I_{240b} = 7$

183d. $$I_{275k} = [\,(I_{254k} - I_{272k})\,(I_{275kr} + I_{275ka}) + I_{272k}\, I_{275ks}\,] / (I_{254k} + I_{272k})$$

I_{275k} = digestible fraction from all sources of food in the diet of the kth mammalian population

I_{275kr} = digestible fraction from range plant food consumption for the kth mammalian population

I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population

I_{275ks} = digestible fraction from supplementary plant food consumption for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

184. $$I_{276j,m} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{258i,m}\, I_{257i,m} / I_{260j,m} \qquad 0 \le j \le 3$$

$I_{276j,m}$ = weighted green or live fraction in consumed range plants

I_{275k} = digestible fraction from all sources of food in the diet of the kth mammalian population

I_{275kr} = digestible fraction from range plant food consumption for the kth mammalian population

I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population

I_{69} = total number of plants in a given region (input)

I_{253k} = digestibility fraction for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

$I_{257i,m}$ = total biomass density, gdwt/m^2, for the ith plant in the mth region

I_{258i} = live plant fraction (including seeds) ingested from the ith plant type from the mth region

I_{260j} = total biomass density, gdwt/m^2, in the jth preference category for range plants of the mth region

$I_{269k,j}$ = food contribution, kgdwt /day, from the jth range plant/animal category from all regions combined for the kth mammalian population
I_{273} = number of regions simulated (input)

185. $$I_{276k} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{239bki,m}$$

I_{275ks} = digestible fraction from supplementary plant food consumption for the kth mammalian population
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith plant in the mth region for the kth animal population
I_{276k} = sum of the selection categories for supplementary feeds selected for consumption by the kth animal population

186. $$I_{278m,k} = 365\,(I_{29} - 1) + I_5 \qquad I_{248k} > 0$$

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population
I_5 = Julian day number (1 - 365)

187. $$I_{246m,k} = (I_{279am,k}\, I_{280k} + I_{279bm,k}\, I_{280k}) / I_{247k}$$

$I_{246m,k}$ = fraction of the population in the mth age group of the kth mammalian population
I_{247k} = total number of animals in the kth mammalian population (initial input)
$I_{279am,k}$ = fraction of male population in the mth age group of the kth mammalian population (initial input or default)
$I_{279bm,k}$ = fraction of female population in the mth age group of the kth mammalian population (initial input or default)
I_{280ak} = total male population of the kth mammalian population
I_{280bk} = total female population of the kth mammalian population

188. $$I_{280ak} = I_{247k}\,(1 - I_{281k})$$

189. $$I_{280bk} = I_{247k}\, I_{281k}$$

I_{280ak} = total male population of the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{281k} = initial female fraction of population (input)
I_{247k} = total number of animals in the kth mammalian population (initial input)

190. $$I_{279m,k} = (I_{282m,k} - I_{282m-1,k})$$

191. $$I_{282m,k} = [\,(I_{283k} - I_{284m,k}) / I_{283k}]^{1.5} \qquad I_{285k}=2$$ (App. B.,I, No. 191)

$= \exp[-(10\, ln2 / I_{283k}) (I_{283k} - I_{284m,k})]$ $\quad I_{285k}=3$

$= [ln\, I_{284m,k} + I_{284m,k} / I_{283k} + 1/ I_{283k}] / [ln\, I_{283k} - 1 + 1/I_{283k}]$ $\quad I_{285k}=4$

192. $I_{283m} = 365\, I_{286k}$ $\quad I_{248k} > 0$

193. $I_{284m,k} = \sum_{i=1}^{m} 365_{i,k}$ $\quad I_{248k} > 0$

$I_{279m,k}$ = fraction of population in the mth age group of the kth mammalian population (male and female fractions assumed to be equal to each other)

$I_{282m,k}$ = population fraction parameter (upper integral) for the mth age group of the kth mammalian population

$I_{282m-1,k}$ = population fraction parameter (lower integral) for the mth age group of the kth mammalian population

I_{283k} = maximum age, days, of oldest age group in the kth mammalian population

$I_{284m,k}$ = maximum age, years, of the mth age group of the kth mammalian population

I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola (input)

I_{286k} = oldest age group, years, for the kth mammalian population (input)

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

194. $I_{289m,k} = F_{57} - F_{58} - F_{59} - F_{60} - F_{61}$

$F(F_{57}, P_{gs})$ = gas losses kg/day from ingested food

$F(F_{57}, P_{ur})$ = mass losses kg/day from ingested food via urine

$F(F_{57}, P_{fc})$ = mass losses kg/day from ingested food via urine

F_{57} = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population

I_{287k} = digestive gas mass loss fraction for the kth mammalian population (input)

I_{288k} = digestive urine mass loss fraction (input)

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population.

I_{253k} = digestibility fraction for the kth mammalian population

P_{gs} = digestive mass losses from gas production, kg/day

P_{ur} = digestive mass losses from urine excretion, kg/day

P_{fc} = digestive mass losses from fecal excretion, kg/day

195. $I_{290m,k} = (X_{16m,k} - X_{15m,k}) / 2.5$

$= I_{289m,k}$ $\quad I_{289m,k} \le I_{290um,k}$

$I_{290m,k}$ = metabolic growth requirement of lean body mass, kg/day, for the mth age group of the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

196. $I_{291m,k} = 0.6944\, I_{290m,k}$

$= I_{292k}$ $\quad I_{292k} \le I_{291um,k}$

$= 2.272\, I_{294m,k}$ $\quad I_{294m,k} > I_{293m,k}$

197. $I_{293m,k} = I_{290m,k} - I_{291m,k}$

$= I_{294m,k}$ $\quad I_{293m,k} < I_{294m,k}$

198. $I_{294m,k} = 0.44\, I_{291um,k}$

$I_{291m,k}$ = protein fraction the growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{290m,k}$ = metabolic growth requirement of lean body mass, kg/day, for the mth age group of the kth mammalian population

$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

$I_{294m,k}$ = protein portion of ingested food that is not deaminated for the mth age group of the kth mammalian population, kg/day

I_{292k} = digested protein from the ingested food, kg/day, for the kth mammalian population

199. $I_{292k} = 0.01\ I_{289m,k}\ I_{294k}$

I_{292k} = digested protein from the ingested food, kg/day, for the kth mammalian population

$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population

200a. $I_{295kr} = \sum_{m=1}^{I_{273}} \sum_{j=1}^{3} I_{269j,m}\ I_{296j,m}\ /\ I_{254k}$

200b. $I_{295ka} = \sum_{m=1}^{I_{273}} \sum_{j=4}^{7} \sum_{i=1}^{I_{288k}} I_{269j,m}\ I_{246i,k}\ X_{17,1,3}\ /\ (\ 0.45\ X_{14i,k}\ I_{254k}\ )$

200c. $I_{295ks} = \sum_{m=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{298i,m}\ I_{239bk}\ /\ I_{276k}$

200d. $I_{295k} = [\ (\ I_{254k} - I_{272k}\ )\ (\ I_{295kr} + I_{295ka}\ ) + I_{272k}\ I_{295ks}\ ]\ /\ (\ I_{254k} + I_{272k}\ )$

I_{295k} = nitrogen fraction from all sources of food in the diet of the kth mammalian population

I_{295kr} = nitrogen fraction from range plant food consumption for the kth mammalian population

I_{295ka} = nitrogen fraction from range animal tissue consumption for the kth mammalian population

I_{295ks} = nitrogen fraction from supplementary plant food consumption for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{272k} = food contribution, kgdwt /day, from all supplementary feed categories from all regions combined for the kth mammalian population

201. $I_{296j,m} = \sum_{m=1}^{I_{273}} \sum_{I=1}^{I_{69}} 0.01\ I_{258i,m}\ I_{297i,m}\ /\ I_{260j,m}$ $\qquad 0 \le j \le 3$

202. $I_{297i,m} = I_{258i}\ I_{298i,m} + I_{259i}\ I_{298i,m}\ /\ 2$ $\qquad 0 \le j \le 3)$

203. $I_{298i,m} = 100\ [\ X_{13,5,1,2} + X_{13,6,1,2} + X_{13,4,1,2}\ (X_{4,1} - X_{5,1})(X_{4,1}\ /X_{5,1}{}^{2}\ )]\ /\ [(X_{4,1} - X_{5,1})(X_{4,1}\ /X_{5,1}\ ) + X_{5,1} + X_{6,1}\ ]$

$I_{296j,m}$ = weighted nitrogen fraction in consumed range plants of the jth food category in the mth region

$I_{297i,m}$ = percent nitrogen in consumed food from the ith plant of the mth region
$I_{298i,m}$ = percent nitrogen in live biomass from the ith plant of the mth region
I_{294k} = weighted nitrogen fraction from all sources of food in the diet of the kth mammalian population
I_{295kr} = weighted nitrogen fraction from range plant food consumption for the kth mammalian population
I_{295ka} = weighted nitrogen fraction from range animal tissue consumption for the kth mammalian population
I_{288k} = oldest age group, years, for the kth mammalian population (input)
$X_{17,1,3}$ = nitrogen content of muscle tissue, grams
$X_{13,5,1,2}$ = nitrogen content in live leaves, g/m^2
$X_{13,6,1,2}$ = nitrogen content in green seeds, g/m^2
$X_{13,4,1,2}$ = nitrogen content in live edible stems, g/m^2
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2
$X_{6,1}$ = live seed biomass density, g/m^2
I_{69} = total number of plants in a given region (input)
I_{253k} = digestibility fraction for the kth mammalian population
I_{254k} = food available, kgdwt, on a given day for the kth mammalian population
$I_{257i,m}$ = total biomass density, $gdwt/m^2$, for the ith plant in the mth region
I_{258i} = live plant fraction (including seeds) ingested from the ith plant type from the mth region
I_{259i} = dry plant fraction (excluding seeds) ingested from the ith plant type
I_{260j} = total biomass density, $gdwt/m^2$, in the jth preference category for range plants of the mth region
$I_{246i,k}$ = fraction of the population in the ith age group of the kth mammalian population
$I_{269j,m,k}$= food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population
I_{273} = number of regions simulated (input)

204. $$I_{298i,m} = [(X_{4,2}\, X_{5,1} /X_{4,1})\, (X_{13,4,2.2} + X_{13r,4,2,2} + X_{13s,4,2,2}) + X_{13,5,2,2} + X_{13r,5,2,2} + X_{13s,5,2,2} + X_{13,5,1,2} + X_{13r,5,1,2} + X_{13s,5,1,2} + (X_{4,1} - X_{5,1})\, (X_{5,1} /X_{4,1})\, (X_{13,4,1,2} + X_{13r,4,1,2} + X_{13s,4,1,2})/ X_{4,1} + X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}] / [(X_{4,2}\, X_{5,1} /X_{4,1}) + X_{5,2} + X_{5,2} + (X_{4,1} - X_{5,1})\, (X_{5,1} /X_{4,1}) + X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}] \qquad I_{240b} = 4$$

$$= (X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}) / X_{6,1} \qquad I_{240b} = 5$$

$$= [(X_{4,2}\, X_{5,1} /X_{4,1})\, (X_{13,4,2,2} + X_{13r,4,2,2} + X_{13s,4,2,2}) + X_{13,5,2,2} + X_{13r,5,2,2} + X_{13s,5,2,2} + X_{13,5,1,2} + X_{13r,5,1,2} + X_{13s,5,1,2} + (X_{4,1} - X_{6,1})\, (X_{5,1} /X_{4,1})\, (X_{13,4,1,2} + X_{13r,4,1,2} + X_{13s,4,1,2})/ X_{4,1}] / [(X_{4,2}\, X_{5,1} /X_{4,1}) + X_{5,2} + X_{5,2} + (X_{4,1} - X_{5,1})\, (X_{5,1} /X_{4,1})] \qquad I_{240b} = 6$$

$$= (X_{13,6,1,2} + X_{13r,6,1,2} + X_{13s,6,1,2}) / X_{6,1} \qquad I_{240b} = 7$$

I_{295ks} = fraction of nitrogen fraction from supplementary plant food consumption for the kth mammalian population
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$I_{239bk,i,m}$ = the fraction of the animal's diet contributed by ith plant in the mth region for the kth animal population
I_{276k} = sum of the selection categories for supplementary feeds selected for consumption by the kth animal population
$X_{13,5,1,2}$ = nitrogen content in live leaves, g/m^2
$X_{13,6,1,2}$ = nitrogen content in green seeds, g/m^2
$X_{13,4,1,2}$ = nitrogen content in live edible stems, g/m^2
$X_{13r,5,1,2}$ = nitrogen content on live leaves from resuspension events, g/m^2

$X_{13s,5,1.,2}$ = nitrogen content on live leaves from saltation-creep events, g/m^2
$X_{13r,6,1,2}$ = nitrogen content on green seeds from resuspension events, g/m^2
$X_{13s6,1,2}$ = nitrogen content on green seeds from saltation-creep events, g/m^2
$X_{13r,4,1,2}$ = nitrogen content on live edible stems from resuspension events, g/m^2
$X_{13s,4,1,2}$ = nitrogen content on live edible stems from saltation-creep events, g/m^2
$X_{13,5,2,2}$ = nitrogen content in dry leaves, g/m^2
$X_{13r,2,2}$ = nitrogen content on dry leaves from resuspension events, g/m^2
$X_{13s,5,2,2}$ = nitrogen content on dry leaves from saltation-creep, g/m^2
$X_{13,6,2,2}$ = nitrogen content in dry seeds, g/m^2
$X_{13r,6,2,2}$ = nitrogen content on dry seeds from resuspension events, g/m^2
$X_{13s,6,2,2}$ = nitrogen content on dry seeds from saltation-creep events, g/m^2
$X_{13,4,2,2}$ = nitrogen content in dry edible stems, g/m^2
$X_{13r,4,2,2}$ = nitrogen content on dry edible stems from resuspension events, g/m^2
$X_{13s,4,2,2}$ = nitrogen content on dry edible stems from saltation-creep, g/m^2
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2

205. $I_{299m,k} = I_{289m,k} - I_{291m,k} - I_{293m,k}$

$I_{299m,k}$ = metabolic growth increment, kg/day, for the mth age group of the kth mammalian population
$I_{289m,k}$ = growth or mass increment, kg/day, for the mth age group of the kth mammalian population
$I_{291m,k}$ = protein fraction the growth increment, kg/day, for the mth age group of the kth mammalian population
$I_{293m,k}$ = non-nitrogen portion of growth increment, kg/day, for the mth age group of the kth mammalian population

206a. $I_{300am,k} = I_{280ak}\, I_{279am,k}$

$= I_{280ak}\,(I_{301k} - I_{302k})$ m=1

206b. $I_{300bm,k} = I_{280bk}\, I_{279bm,k}$
$= I_{280bk}\,(I_{301k} - I_{302k})$ m=1

207. $I_{301k} = [\,(I_{283k} - I_{303k}) / I_{283k}]^{1.5}$ $I_{285k}=2$
$= \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - I_{303k})\,]$ $I_{285k}=3$
$= [\,ln\, I_{303k} + I_{303k} / I_{283k} + 1/ I_{283k}\,]/ [ln\, I_{283k} - 1 + 1/I_{283k}\,]$ $I_{285k}=4$

208. $I_{302m,k} = [\,(I_{283k} - 365) / I_{283k}]^{1.5}$ $I_{285k}=2$
$= \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - 365)\,]$ $I_{285k}=3$
$= [\,ln\, 365 + 365 / I_{283k} + 1/ I_{283k}\,]/ [ln\, I_{283k} - 1 + 1/I_{283k}\,]$ $I_{285k}=4$

$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population
$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population
I_{280ak} = total male population of the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population

I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
$I_{279am,k}$ = fraction of male population in the mth age group of the kth mammalian population (initial input or default)
$I_{279bm,k}$ = fraction of female population in the mth age group of the kth mammalian population (initial input or default)
I_{301k} = upper age correction factor for 1 year-olds
I_{302k} = lower age correction factor for 1 year-olds

209. $I_{303k} = I_{284k} - I_{304k} - I_{305k}$

210a. $I_{304k,1} = I_{306k,1} + 14 + I_{307k}$

210b. $I_{304k,2} = I_{306k,2} + 14 + I_{307k}$ $\qquad I_{308k}=2$

211. $I_{306k,2} = I_{306k,1} + 14 + I_{307k} + 9.125\, I_{248k}$ $\qquad I_{308k} = 2$
$= 0$ $\qquad I_{308k} = 1$

212. $I_{307k} = 0.15\,(365\, I_{248k})^{0.8}$

213. $I_{305k} = (14 + 0.025(365\, I_{248k}) + I_{307k}) / 2$ $\qquad I_{308k} = 2$
$= 0$ $\qquad I_{308k} = 1$

I_{303k} = corrected age of a one year-old, days, for the kth mammalian population
$I_{284m,k}$ = maximum age, years, of the mth age group of the kth mammalian population
$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)
$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)
I_{305k} = age correction, days, for newborns of the kth mammalian population
$I_{306k,1}$ = initiation of gestation period, first litter, for the kth mammalian population (1= day 92, 2=day 270) (input)
$I_{306k,2}$ = initiation of gestation period ,second litter, for the kth mammalian population
I_{307k} = estimated gestation period, days, for the kth mammalian population
I_{308k} = number of gestation periods per year (input)
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

214. $I_{309m,k} = (0.5/ I_{283k})\,(1 - I_{278m,k} / I_{283k})^{-0.5/ I310k}$ $\qquad I_{285k}= 2;m>1$

$= \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - I_{278m,k})\,]\,/I_{310k}$ $\qquad I_{285k}= 3;m>1$

$= I_{278m,k}^{-2/ I310k}$ $\qquad I_{285k}= 4;m>1$

$I_{309m,k}$ = natural mortality rate, day^{-1}, for the mth age group of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
$I_{278m,k}$ = age, days, of the mth age group of the kth mammalian population
I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population

214a. $I_{309m,k} = (0.5/ I_{283k})\,(1 - I_{312k1} / I_{283k})^{-0.5/ I310k}$ $\qquad I_{285k}= 2;m=1$
$= (0.5/ I_{283k})\,(1 - I_{312k2} / I_{283k})^{-0.5/ I310k}$
$= (0.5/ I_{283k})\,(1 - I_{312k3} / I_{283k})^{-0.5/ I310k}$

214b. $I_{309m,k} = \exp[\,-(10\, ln2 / I_{283k})\,(I_{283k} - I_{312k1})\,]\,/I_{310k}$ $\qquad I_{285k}= 3;m=1$

$= \exp[-(10\ ln2 / I_{283k}) (I_{283k} - I_{312k2})] / I_{310k}$
$= \exp[-(10\ ln2 / I_{283k}) (I_{283k} - I_{312k3})] / I_{310k}$

214c. $I_{309m,k} = I_{312k1}^{-2/I310k}$ $\qquad I_{285k} = 4; m=1$
$= I_{312k2}^{-2/I310k}$
$= I_{312k3}^{-2/I310k}$

$I_{309m,k}$ = natural mortality rate, day^{-1}, for the 1st age group of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population
I_{312k1} = age, days, of newborns carried over from the previous year for the kth mammalian population
I_{312k2} = age, days, of newborns from the first gestation period during a given year for the kth mammalian population
I_{312k3} = age, days, of newborns from the second gestation period during a given year for the kth mammalian population

215. $I_{310k} = I_{280bk} / I_{311k}$

I_{310k} = the ratio of the current number of females to the original number at the start of the simulation for the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{311k} = original number of females in the kth mammalian population at the start of the simulation (input)

216a. $I_{312k1} = I_{278m,k}$
$= 1$ $\qquad I_{278m,k} < 1.0$

216b. $I_{312k2} = \sum_{i=1}^{n} 1_i$
$= 1$ $\qquad I_{304k,1} = I_{313k1}$

216c. $I_{312k3} = \sum_{i=1}^{n} 1_i$
$= 1$ $\qquad I_{304k,2} = I_{313k2}$

217a. $I_{313k1} = \sum_{i=1}^{n} 1_i$
$= I_5$ $\qquad I_{312k1} > I_{304k,1} + 9.125\ I_{248k}$

217b. $I_{313k2} = \sum_{i=1}^{n} 1_i$
$= I_5$ $\qquad I_{312k2} > I_{304k,2} + 9.125\ I_{248k}$

I_{312k1} = age, days, of newborns carried over from the previous year for the kth mammalian population
I_{312k2} = age, days, of newborns from the first gestation period during a given year for the kth mammalian population

I_{312k3} = age, days, of newborns from the second gestation period during a given year for the kth mammalian population

I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population

I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter) for the kth mammalian population

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter) for the kth mammalian population

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

I_5 = Julian day number (1 - 365)

218. $I_{314m,k} = 0.1\ I_{315m,k} \exp(-I_{316m,k})$

$= 0 \qquad X_{14} \geq 0.7\ X_{14e}$

219. $I_{315m,k} = (X_{14e} - X_{14}) / (0.250663\ X_{14e})$

220. $I_{316m,k} = [(X_{14e} - X_{14}) / (0.1\ X_{14e})] / 8$

$X_{314m,k}$ = mortalities due to starvation, day^{-1}, for the mth age group of the kth mammalian population

$X_{315m,k}$ = starvation factor, day^{-1}

$X_{316m,k}$ = starvation coefficient

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

221. $I_{317m,k,l} = 1000\ F_{57}\ (I_{269j,i,k} / I_{254k})\ (I_{300am,k} + I_{300bm,k}) / 1000\ I_{318k}) \qquad j > 3; I_{241k} > 0$

222. $I_{318k} = 0.33 \qquad I_{248k} > 0$

$= 1 \qquad I_{248k} = 0$

$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region

$I_{269j,i,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the ith region combined for the kth mammalian population

$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population

$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population

F_{57} = transport of food, kgdwt/day, to an animal of the mth age group of the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{318k} = weight fraction of prey utilized for food

I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

223. $I_{319mp,kp} = I_{317m,k,l} / [\sum_{kp=1}^{I274} \sum_{mp=1}^{I248} (I_{300amp,kp} + I_{300bmp,kp})\ X_{14mp,kp}]\ I_{246mp,kp} \qquad I_{241k,l} > 0$

$I_{319mp,kp}$ = mortality rate, day-1, of the mpth age group of the kpth prey animal due to the mth age group of kth mammalian predator population

$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region

$X_{14,mp,kp}$ = animal biomass compartment, kgfwt, for an animal in the mpth age group of the kpth animal prey population

$I_{300amp,kp}$ = number of males in the mpth age group of the kpth prey mammalian population

$I_{300bmp,kp}$ = number of females in the mpth age group of the kpth prey mammalian population

$I_{246mp,kp}$ = fraction of the population in the mpth age group of the kpth mammalian prey population comsumed by the kth mammalian predator population

I_{274k} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannibalism is being modeled) in a given region

I_{248kp} = lifespan, years (largest whole number), of the an animal in the kpth mammalian prey population (input)

I_{241kp} = initial diet fraction of the kpth animal prey type in the food of a specific animal predator type (input)

224. $I_{246mp,kp} = (I_{246mp,kp}^{2} / I_{i246mp,kp})$ $\quad (I_{246mp,kp} / I_{i246mp,kp}) < 1$

$= I_{246mp,kp}$ $\quad (I_{246mp,kp} / I_{i246mp,kp}) \geq 1$

$I_{246mp,kp}$ = fraction of the population in the mpth age group of the kpth mammalian prey population available for consumption by the kth mammalian predator population

$I_{i246mp,kp}$ = initial fraction of the population in the mpth age group of the kpth mammalian prey population available for consumption by the kth mammalian predator population (initial conditions derived from input)

225. $I_{319kp} = (I_{317m,k,l} / X_{14kp}) I_{246kp}$ $\quad I_{241k,l} > 0$

$I_{246kp} = 1$ $\quad I_{320kp} \geq 1$

226. $I_{246kp} = I_{320kp}$ $\quad I_{320kp} < 1$

I_{319kp} = mortality rate, day-1, of the kpth prey invertebrate animal due to the mth age group of kth mammalian predator population

$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth invertebrate prey by the mth age group of the kth mammalian population used as food from a given region

X_{14kp} = animal biomass compartment, kgdwt, for an invertebrate animal of the kpth animal prey population

I_{246kp} = fraction of the population of the kpth invertebrate prey population available for consumption by the kth mammalian predator population

I_{320kp} = combined biomass density, gdwt /m^2, of all of the kpth immature and mature invertebrate animal

227.
$$
\begin{aligned}
I_{321k,l,n} &= 0.003\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=1;n=1\\
&= 0.003\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=1;n=2\\
&= 0.138\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=2;n=1\\
&= 0.138\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=2;n=2\\
&= 0.296\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=3;n=1\\
&= 0.296\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=3;n=2\\
&= 0.295\,(1-I_{282m,k})\, I_{280ak} / (I_{280bk}\, I_{322k,l}) && l=4;n=1\\
&= 0.295\,(1-I_{282m,k})\, I_{280bk} / (I_{280bk}\, I_{322k,l}) && l=4;n=2
\end{aligned}
$$

$= 0.207\ (1 - I_{282m,k})\ I_{280ak} / (I_{280bk}\ I_{322k,l})$ l=5;n=1
$= 0.207\ (1 - I_{282m,k})\ I_{280bk} / (I_{280bk}\ I_{322k,l})$ l=5;n=2
$= 0.050\ (1 - I_{282m,k})\ I_{280ak} / (I_{280bk}\ I_{322k,l})$ l=6;n=1
$= 0.050\ (1 - I_{282m,k})\ I_{280bk} / (I_{280bk}\ I_{322k,l})$ l=6;n=2
$= 0.010\ (1 - I_{282m,k})\ I_{280ak} / (I_{280bk}\ I_{322k,l})$ l=7;n=1
$= 0.010\ (1 - I_{282m,k})\ I_{280bk} / (I_{280bk}\ I_{322k,l})$ l=7;n=2
$= 0.001\ (1 - I_{282m,k})\ I_{280ak} / (I_{280bk}\ I_{322k,l})$ l=8;n=1
$= 0.001\ (1 - I_{282m,k})\ I_{280bk} / (I_{280bk}\ I_{322k,l})$ l=8;n=2

228. $I_{322k,l} = (I_{279bmr,k} + I_{279bmr-1,k}) / 2$ $I_{285k}=1$

$= [(I_{283k} - I_{323k,l}) / I_{283k}]^{1.5} - \{[I_{283k} - (I_{323k,l} + I_{324k})] / I_{283k}\}^{1.5}$ $I_{285k}=2$

$= \exp[-(10\ ln2 / I_{283k})(I_{283k} - I_{323k,l})] - \exp\{-(10\ ln2 / I_{283k})[I_{283k} - (I_{323k,l} + I_{324k})]\}$ $I_{285k}=3$

$= [ln\ I_{323k,l} - I_{323k} / I_{283k} - ln(I_{323k,l} + I_{324k})] + (I_{323k,l} + I_{324k})/I_{283k}$ $I_{285k}=4$

$I_{322k,l}$ = fraction of the reproductive females in the lth reproductive age grouping for the kth mammalian population
$I_{321k,l,n}$ = reproductive rate, individual-1, for the lth female reproductive grouping for males (n=1), and females (n=2)
I_{280ak} = total male population of the kth mammalian population
I_{280bk} = total female population of the kth mammalian population
I_{283k} = maximum age, days, of oldest age group in the kth mammalian population
I_{285k} = population structure type (1=user input, 2=parabola, 3=exponential, 4=hyperbola) (input)
$I_{279bmr,k}$ = fraction of female population in the mrth reproductive age group of the the kth mammalian population (initial input or default)
$I_{282m,k}$ = population fraction parameter (upper integral) for the mth age group of the kth mammalian population

$I_{282m-1,k}$ = population fraction parameter (lower integral) for the mth age group of the kth mammalian population
$I_{323k,l}$ = age, days, of a reproductive female in the lth grouping (l= 1-8) , initial age set at 15% of the life-span, days
I_{324k} = age increment, days, expressed as 1/8 th of the age at 55% of the life span

229. $I_{325k,n} = \sum_{l=1}^{8} I_{321k,l,n}\ I_{322k,l}\ I_{311k}\ I_{325i,k}\ I_{304k,1}$ $I_{304k,1} = I_{313k,1}$

$= \sum_{i=1}^{8} I_{321k,l,n}\ I_{322k,l}\ I_{311k}\ I_{325i,k}\ I_{304k,2}$ $I_{304k,2} = I_{313k,2}$

$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population
$I_{321k,l,n}$ = reproductive rate, individual-1, for the lth female reproductive grouping for males (n=1), and females (n=2)
$I_{322k,l}$ = fraction of the reproductive females in the lth reproductive age grouping for the kth mammalian population
I_{311k} = original number of females in the kth mammalian population at the start of the simulation (input)
$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)
$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

$I_{325i,k}$= fraction of new-born in the ith gestation period of a given year for the kth mammalian population (equals 1 if only one gestation period specified, input)

230a. $I_{300am,k} = I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k}$ m>1
$= I_{300am-1,k}$ m>1; I_5 =365
$= I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n}$ m=1; $I_{304k,1} = I_{313k,1}$
$= I_{300am,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n}$ m=1; $I_{304k,2} = I_{313k,2}$

230b. $I_{300bm,k} = I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k}$ m>1
$= I_{300bm-1,k}$ m>1; I_5 =365
$= I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n}$ m=1; $I_{304k,1} = I_{313k,1}$
$= I_{300bm,k} - I_{309m,k} - I_{314m,k} - I_{319m,k} + I_{325k,n}$ m=1; $I_{304k,2} = I_{313k,2}$

$I_{300am,k}$ = number of males in the mth age group of the kth mammalian population
$I_{300bm,k}$ = number of females in the mth age group of the kth mammalian population
$I_{309m,k}$ = natural mortality rate, day^{-1}, for the mth age group of the kth mammalian population of either sex
$I_{314m,k}$ = mortalities due to starvation, day^{-1}, for the mth age group of the kth mammalian population
$I_{319m,k}$ = mortality rate, day^{-1}, of the mth age group of the kth animal type due to predation from a given predator
$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)
$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population
I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population
I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population
I_5 = Julian day number (1 - 365)

231a. $I_{300am,k}$= $(I_{326ak,1} + I_{326ak,2} + I_{326ak,3})$ m=1; I_5=365
231b. $I_{300bm,k}$= $(I_{326bk,1} + I_{326bk,2} + I_{326bk,3})$ m=1; I_5=365

232a. $I_{326ak,n} = I_{325k,1}$ I_{327k} =1;$I_{304k,1} = I_{313k,1}$;I_{328k}= 0;n=1
$= I_{326ak,n-1} + I_{325k,1}$ I_{327k} =2;$I_{304k,2} = I_{313k,2}$;I_{328k}= 0;n=2
$= I_{326ak,n} + I_{326ak,n+1}$ $I_{304k,1} = I_{313k,n}$;$I_{328k} > 1$;$n \leq 2$

232b. $I_{326bk,n} = I_{325k,2}$ I_{327k} =1;$I_{304k,1} = I_{313k,1}$;I_{328k}= 0;n=1
$= I_{326bk,n-1} + I_{325k,2}$ I_{327k} =2;$I_{304k,2} = I_{313k,2}$;I_{328k}> 0;n=2
$= I_{326bk,n} + I_{326bk,n+1}$ $I_{304k,2} = I_{313k,n}$;$I_{328k} > 1$;$n \leq 2$

233a. $I_{326ak,n+1} = I_{325k,1}$ $I_{304k,1} = I_{313k,1}$;$I_{328k} > 1$;$n \leq 2$
233b. $I_{326bk,n+1} = I_{325k,2}$ $I_{304k,1} = I_{313k,1}$;$I_{328k} > 1$;$n \leq 2$

$I_{326ak,n}$ = number of male animals in the nth sub-group of the first age group for the kth mammalian population
$I_{326ak,n}$ = number of female animals in the nth sub-group of the first age group for the kth mammalian population
I_{327k} = index number (1,2) designating the first or second gestation period if simulated for the kth mammalian population
$I_{325k,n}$ = total number newborns of either sex (n=1; males, n=2; females) for the kth mammalian population

234. $I_{329m} = I_{335m} \exp(0.693/I_{336m})(I_5 - I_{331m})$ $\quad I_{340}=1; I_{336m} > 0$

$= I_{335m} + I_{336m}(I_5 - I_{331m})$ $\quad I_{340}=2$

$= I_{337m}(1 - \exp(0.693/I_{336m})(I_5 - I_{331m})$ $\quad I_{340}=3$

$= 0.5(I_{337m} - I_{335m})\{\sin[2.0\pi(I_{338} - I_{339})/(I_{332m} - I_{331m})] + 1\} + I_{335}$ $\quad I_{340}=4$

$= I_{335}$ $\quad I_{340}=5$

235. $I_{330m} = I_{329m}$

I_{329m} = air concentration, m^{-3}, for the mth particulate contaminant for a given region

I_{330m} = air concentration, m^{-3}, for the mth vaporous contaminant for a given region

I_{335m} = initial air concentration , m^{-3}, for the mth contaminant in either particulate or vapor form (input); **note:** when $I_{340} > 3$, this variable equals the minimum air concentration instead of the initial air concentration

I_{336m} = decay (-) or growth(+) half-time, days, or linear coefficient , day^{-1},of the chronic function for the mth contaminant in a given region (input)

I_{331m} = Julian day number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region (input)

I_{332m} = Julian day number for end of chronic function for estimating daily air concentration of the mth contaminant in a given region (input)

I_{337m} = maximum air concentration , m^{-3}, for the mth contaminant in either particulate or vapor form (input)

I_{338} = current Julian day

I_{339} = lag(-) or lead(+) constant, degrees, for determining Julian day at which maximum air concentration (amplitude) will occur in the sine wave function (input)

I_{340} = chronic function selector (1=exponential, 2=linear, 3=ingrowth, 4= sine-wave, 5= constant (input)

236. $I_{329m} = I_{329m} \exp[(0.693/I_{341m})(I_{29} - I_{333m})]$ $\quad I_{342}=1$

$= I_{329m} I_{341m}(I_{29} - I_{333m})$ $\quad I_{342}=2$

$= I_{329m}(1 - \exp[(0.693/I_{341m})(I_{29} - I_{333m})]$ $\quad I_{342}=3$

$= I_{329m} I_{341m}$ $\quad I_{342}=4$

237. $I_{330m} = I_{329m}$

I_{333m} = year number for start of chronic function for estimating daily air concentration of the mth contaminant in a given region

I_{341m} = decay (-) or growth(+) half-time, years, or linear coefficient , $year^{-1}$,of the chronic function damper/ enhancer for the mth contaminant in a given region (input)

I_{342} = chronic function damper/enhancer selector (1=exponential, 2=linear, 3=ingrowth, 4= constant) (input)

I_{328k} = index number (0,>0) designating default beef and dairy cow population dynamics apart from other mammalian populations for the kth mammalian population

$I_{304k,1}$ = time delay, days, for newborns from one litter to the next (1st litter)

$I_{304k,2}$ = time delay, days, for newborns from one litter to the next (2nd litter)

I_{313k1} = fetal age, days, of the first gestation period during a given year for the kth mammalian population

I_{313k2} = fetal age, days, of the second gestation period during a given year for the kth mammalian population

238. $I_{347} = 0.0021$ $\quad F_1 \leq 2$

$= 0.7 [1 - \exp(-0.6518 F_1)]$ $\quad F_1 > 2$

and,

239. $I_{348} = [\sum_{i=1}^{I_{69}} I_{43} I_{265i}] / \sum_{i=1}^{I_{69}} I_{265i}$ $\quad O_1 > 0; O_2 > 0$

$= I_{43}$ $\quad O_1 = 0; O_2 = 0$

I_{347} = U.S. mean dry deposition velocity, m/ day, for a given region
I_{348} = areally weighted mean plant crown cover, fraction, of a given region
F_1 = transport of moisture from rain events to soil, mm/day
I_{43} = plant fractional crown cover, m^2/ m^2
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
O_1 = niche competition option (1,0) (input)
O_2 = succession option (1,0) (input)

240. $I_{349} = 0.5 \pi I_{163} I_{162} [(0.5 I_{163}^2) + I_{351}^2]^{0.5}$

241. $I_{350} = \pi I_{352} + I_{350}$

242. $I_{351} = 0.001 X_{4,1}$

$= 0.3048 I_{167}$ $\quad L_2 > 1; I_{166} > 0; I_{167} > 0; I_{165} > 0$

I_{349} = total surface area, m^2/m^2, of combined boles of a given plant based on the surface area of a cone
I_{350} = total surface area, m^2/m^2, of combined bole and branches of a given plant
I_{351} = estimated plant height, meters
I_{352} = projected branch cover, m^2 /m^2, on ground surface assuming branches are parallel to the ground
I_{162} = estimated plant areal density, number / m2 for a given plant type at a given time
I_{163} = estimated effective stem diameter, cm, of a given plant type at a given time
I_{167} = mean height, ft, for a stand of a given plant
I_{165} = tree density, trees /acre, for a given plant type
$X_{4,1}$ = live stem biomass density, g $/m^2$
I_{166} = mean bole diameter, inches, for a stand of a given plant
L_2 = vegetation index (1,2,3) set to 1 if tree structure simulation not accessed; set to 2 if tree structure simulation is accessed yearly; equal 3 if accessed daily

243. $I_{353} = 2.5e\text{-}08 \exp (- 4.6 I_{348})$

$= 0$ $\quad D_2 > 2$ or $F_2 > 1$

D_2 = mean daily temperature, ° C
F_2 = transport of moisture from snowfall events to soil, mm/day
I_{353} = resuspension fraction from plant surface to the atmosphere
I_{348} = areally weighted mean plant crown cover, fraction, of a given region

244. $I_{b354} = - (76.45/1.21)(I_{164}^{-1.21} - 7^{-1.21}) / (I_{164} - 7)$

244b. $I_{354} = - (76.45/1.21)(I_{164}^{-1.21} - I_{356}^{-1.21}) / (I_{164} - I_{356})$

245. $I_{356} = 0.33 I_{164}$ $\quad L_2 > 1; L_4 = 1$

$= 0.66\ I_{164}$ $\qquad L_2 > 1;\ L_4 = 0$

$= 7$ $\qquad L_2 = 1;\ I_{356} < 7$

I_{b354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of boles (stems) for various plants

I_{354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of branches for various plants

I_{356} = mean lower height of plant branches, cm, based on predicted plant height for estimating saltation-creep and rainsplash contaminant transport on plants

I_{164} = estimated height of a given plant type, cm, at a given time

L_2 = forest structure access gate (1,2,3) (input)

L_4 = evergreen tree and shrub gate (1,0) (input)

246. $I_{355} = (X_{sal3,4,6} + X_{sal3,4,7}) / (X_{s13,4,6} + X_{s13,4,7})$

I_{355} = the ratio of the integral average contamination on boles and branches of a given plant to the maximum estimates on these surfaces assuming height independence from saltation-creep-rainsplash events

I_{354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of branches for various plants

$X_{sal3,4,6}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant

$X_{sal3,4,7}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant

$X_{s13,4,6}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant

$X_{s13,4,7}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant

247. $I_{357} = 2.5e\text{-}04 \exp (- 3.2\ I_{348})$

$= 10\ I_{357}$ $\qquad F_1 > 5;\ D_2 \geq 0$

$= 0.0$ $\qquad F_2 > 0$

F_1 = transport of moisture from rain events to soil, mm/day

D_2 = mean daily temperature, °C

F_2 = transport of moisture from snowfall events to soil, mm/day

I_{357} = saltation creep-rainsplash fraction from plant and soil surfaces to the atmosphere

I_{348} = areally weighted mean plant crown cover, fraction, of a given region

248. $I_{358} = 0.000025 / I_{37i}$

$= I_{358}/10$ $\qquad I_{359} = 19$

$= I_{358}/10$ $\qquad I_{359} = 55$

I_{358} = eluviation rate, $mm^{-1}\ day^{-1}$, of contaminant from soil surface to the second layer

I_{37i} = thickness of the 1st (i=1)soil layer, mm (input)

249a. $I_{360} = 1000\ F_{7a}\ X_{12,i} / I_{362}$

249b. $I_{361} = 1000\ F_7\ X_{12,i} / I_{362}$

$F_w (X_{12,i} , X_{12,i-1})$ = transport of contaminant from the ith soil layer to the (i-1)th layer (top layer excluded), day^{-1} gdwt-$soil^{-1}$, by soil moisture flow

F_w ($X_{12,i}$, $X_{12,i+1}$) = transport of contaminant from the ith soil layer to the (i+1)th layer (bottom layer excluded), day^{-1} gdwt-$soil^{-1}$ from soil moisture flow

I_{37j+1} = thickness, of the (i+1)th soil layer, mm (input)

I_{37j-1} = thickness, of the (i-1)th soil layer, mm (input)

I_{360} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i-1)th layer (top layer excluded) by soil moisture flow

I_{361} = amount of contaminant transported, day^{-1}, from the ith soil layer to the (i+1)th layer (bottom layer excluded) by soil moisture flow

I_{362} = distribution coefficient , ml /gdwt-soil, for a given contaminant (input)

F_{7a} = transport of moisture, mm/day, from the ith soil layer to the (i-1)th (layer (top layer excluded)

F_7 = transport of moisture, mm/day, from the ith soil layer to the (i+1)th (layer (bottom layer excluded)

$X_{12,i}$ = contaminant concentration, gdwt-$soil^{-1}$, of the ith soil layer

$X_{12,i+1}$ = contaminant concentration, gdwt-$soil^{-1}$, of the (i+1)th soil layer

$X_{12,i-1}$ = contaminant concentration, gdwt-$soil^{-1}$, of the (i-1)th soil layer

250. $$I_{363} = \sum_{j=1}^{I_{69}} \sum_{i=1}^{I_{20}} 86400\, I_{364}\, X_{12,i}\, I_{37i,j}\, I_{265j} / (I_{56j} \sum_{j=1}^{I_{69}} I_{265j})$$

F ($X_{12,i}$, X_{18g}) = transport of volatile contaminant to the atmosphere, ug day-1, from the ith soil layer of a given region

$I_{37j,j}$ = thickness, of the ith soil layer, mm, of the jth plant type (input)

I_{363} = mean loss of contaminant, ug day^{-1} m^{-2}, from a given region averaged over all plant types and soil horizons

$X_{12,i}$ = volatile contaminant concentration, ug gdwt-$soil^{-1}$, of the ith soil layer

I_{20} = number of soil layers specified for the ith plant (input or estimated)

I_{69} = total number of plants in a given region (input)

I_{265j} = coverage area, acres, of the jth plant type for a given region (input)

I_{56j} = total depth, mm, of soil profile for the rooting zone of the jth plant type (input)

I_{364} = volatile contaminant emission rate, ug m^{-2} sec^{-1}, assuming a soil concentration of one ug /gdwt

251. $$I_{364} = 0.203\, I_{365}\, I_{366} / [\, 1.919e{+}06\, I_{366} /(10 + I_{367} / I_{365})]^{1/2}$$

I_{364} = volatile contaminant emission rate, ug m^{-2} sec^{-1}, averaged over one day assuming a soil concentration of one ug /gdwt

I_{365} = Henry's Law constant, atm-m^3 / mole (input)

I_{366} = molecular diffusivity, cm^2 / sec (input)

I_{367} = distribution coefficient of contaminant on organic carbon, ml/g, (input)

252. $$I_{366x} = I_{366bn} (I_{368bn} / I_{368x})^{1/2}$$

I_{366x} = estimated molecular diffusivity, cm^2 / sec, of a specific organic compound using the molecular diffusivity of benzene

I_{366bn} = molecular diffusivity of benzene (0.08195 cm^2 /sec) (input)

I_{368x} = molecular weight, grams, of specific organic compound (input)

I_{368bn} = molecular weight, grams, of benzene (78g) (input)

253. $$I_{369i} = 0.375 (1 - 1/ X_{5,1}) I_{370i} F_{5ai} I_{19} X_{12,i} \qquad C_3 > 50$$
$$= 1.25 (1 - 1/ X_{5,1}) I_{370i} F_{5ai} I_{19} X_{12,i} \qquad C_3 \leq 50$$
$$= 0 \qquad X_{3,1} = 1.0 \text{ or } X_{4,1} = 1.0$$

I_{369i} = transport of contaminant, m^{-2} day^{-1}, from the ith soil layer to the root compartment of a given plant
I_{370i} = partition coefficient , gdwt-plant^{-1} / gdwt-soil^{-1} , between plant and soil components for a given contaminant in the ith soil layer (input)
F_{5ai} = transpiration rate, mm m^{-2} day^{-1} , from the ith soil layer for a given plant type
I_{19} = leaf area index, m^2/m^2, (LAI)
P_{sr} = soil-to root source term for a given contaminant
$X_{12,i}$ = contaminant concentration, gdwt-soil^{-1}, of the ith soil layer
$X_{5,1}$ = live leaf compartment, g/ m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/ m^2
$X_{3,1}$ = live root compartment, g/ m^2
C_3 = mean life-span, years, of the plant type simulated

254. $I_{370o} = \exp [2.30259 (1.588 - 0.578 I_{371o})]$

I_{370o} = partition coefficient , gdwt-plant^{-1} / gdwt-soil^{-1} , between plant and soil components for a given organic contaminant in the ith soil layer (input)
I_{371o} = $\log_{10}$ of octanol/ water ratio (kow) of a given organic contaminant (input)

255. $I_{372} = F_{23} / X_{5,1}$
$= 0$ $\quad X_{4,1} = 1.0$ or $I_{372} < 0$
$= 0.99$ $\quad I_{372} > 0.99$

256a. $I_{373a} = X_{3,1} / (X_{3,1} + X_{4,1} + X_{5,1} + X_{6,1})$
$= 0.99$ $\quad I_{373a} > 0.99$

256b. $I_{373b} = I_{373a}$
$= I_{373a} / (I_{373a} + I_{376a} + I_{377a})$ $\quad (I_{373a} + I_{376a} + I_{377a}) > 1$

257. $I_{374} = 0.5 (X_{4,1} + X_{5,1} + X_{6,1}) / (X_{4,1} + X_{5,1} + X_{6,1} + X_{3,1})$
$= 0.99$ $\quad I_{374} > 0.99$

258. $I_{375} = X_{4,1} / (X_{4,1} + X_{5,1} + X_{6,1})$
$= 0.5$ $\quad I_{375} > 0.5$

259a. $I_{376a} = 0.5 X_{5,1} / (X_{4,1} + X_{5,1} + X_{6,1})$
$= 0.99$ $\quad I_{376a} > 0.99$

259b. $I_{376b} = I_{376a}$
$= I_{376a} / (I_{373a} + I_{376a} + I_{377a})$ $\quad (I_{373a} + I_{376a} + I_{377a}) > 1$

260a. $I_{377a} = 0.5 X_{6,1} / (X_{6,1} + X_{5,1} + X_{6,1})$
$= 0.99$ $\quad I_{377a} > 0.99$

260b. $I_{377b} = I_{377a}$
$= I_{377a} / (I_{373a} + I_{376a} + I_{377a})$ $\quad (I_{373a} + I_{376a} + I_{377a}) > 1$

$F (P_{rs}, X_{13,3,1})$ = transport of contaminant, m^{-2} day^{-1}, from stem compartment of a given plant type to the root compartment
$F (P_{rs} + P_{sw} + P_{ss} , X_{13,4,1})$ = transport of contaminant, m^{-2} day^{-1}, from the root, leaf and seed compartments to the stem compartment of a given plant type
$F (P_{sw}, X_{13,5,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the leaf compartment of a given plant type

$F(P_{ss}, X_{13,6,1})$ = transport of contaminant, m^{-2} day^{-1}, from the stem compartment to the seed or fruit compartment of a given plant type
I_{372} = ratio of net daily photosynthate production, to standing live leaf biomass, day^{-1}
I_{373a} = initial biomass density factor: stems to total biomass
I_{373b} = adjusted biomass density factor: stems to total biomass
I_{374} = above ground biomass density factor: stems, leaves, and seeds to total biomass
I_{375} = biomass density factor: stems to above ground biomass
I_{376a} = initial biomass density factor: leaves to above ground biomass
I_{376b} = adjusted biomass density factor: leaves to above ground biomass
I_{377a} = initial biomass density factor: seeds to above ground biomass
I_{377b} = adjusted biomass density factor: seeds to above ground biomass
$X_{13,3,1}$ = internal contaminant concentration, m^{-2}, in live roots of a given plant type
$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type
$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type
$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type
F_{23} = net photosynthesis, $g/m^2/day$, for a given plant type
$X_{5,1}$ = live leaf compartment, g/m^2
$X_{4,1}$ = live above ground biomass (stem) compartment, g/m^2
$X_{3,1}$ = live root compartment, g/m^2
$X_{6,1}$ = live seed biomass density, g/m^2
P_{rs} = root-stem contaminant exchange source
P_{sw} = stem-leaf contaminant exchange source
P_{ss} = stem-seed contaminant exchange source

261. $I_{378} = X_{v17,1} / \sum_{i=1}^{6} X_{fl7,i}$

$F(fM_a, X_{17,2,n})$ = exchange of contaminant, hr^{-1}, between blood plasma and muscle for the nth contaminant
fM_a = transport of contaminant from blood plasma to muscle tissue, hr^{-1}
$X_{17,2,n}$ = contaminant content of muscle tissue, mass or activity units, for the nth contaminant
$X_{17,1,n}$ = contaminant content of blood plasma, mass or activity units, for the nth contaminant
$X_{fl7,2}$ = perfusion rate, liters hr^{-1}, of blood plasma into muscle tissue
$X_{fl7,i}$ = perfusion rate, liters hr^{-1}, of the ith organ or tissue
$X_{v17,1}$ = volume of blood plasma, liters
$X_{v17,2}$ = perfusion volume of muscle tissue, liters
$X_{r17,2,n}$ = partition coefficient for the nth contaminant in muscle tissue (input)
I_{378} = normalizing constant, hr^{-1}, which is used to estimate fractional flow to any given organ or tissue during a given time interval over short equilibration periods for a given contaminant.

263. $I_{380m,k} = I_{385} (I_{386} X_{1e17,6,n} + I_{387} X_{2e17,6,n})$

263b. $I_{380b} = I_{388} X_{3e17,6,n}$

I_{380} = transport of contamination, day^{-1}, from the surface of the lung lumen to the liver via the hepatic portal-vein from the gut
I_{380b} = transport of contamination ,day^{-1}, from the pulmonary surface of the lung directly into arterial blood

I_{379} = total transport of contaminant, day^{-1}, from the gut to the liver via the hepatic-vein

$I_{381m,k}$ = transport of contaminant from the N-P, and upper T-B regions of the respiratory tract to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population

$I_{382m,k}$ = regionally weighted contaminant transport from ingested food to the mth age group of the kth mammalian predator population

$I_{383m,k}$ = amount of soil consumed, gdwt day^{-1}, by an animal from grazing for the mth age group of the kth mammalian population

$I_{384m,k}$ = amount of contaminant, day-1, transported to the hepatic-portal vein from drinking water via the gut for the mth age group of the kth mammalian population

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)

I_{386} = fraction of a given contaminant transferred, day^{-1}, from the bronchus region of the lung lumen surface to the gut via particulates (input)

I_{387} = fraction of a given contaminant transferred, day^{-1}, from the bronchiole region of the lung lumen surface to the gut via particulates (input)

I_{388} = fraction of a given contaminant transferred, day^{-1}, from the pulmonary region of the lung lumen surface to the blood (input)

$X_{1e17,6,n}$ = contaminant content on the surface of the bronchus region of the lung lumen as particulates

$X_{2e17,6,n}$ = contaminant content on the surface of the bronchiole region of the lung lumen as particulates

$X_{3e17,6,n}$ = contaminant content on the surface of the pulmonary region of the lung lumen as particulates

264. $$I_{381m,k} = 0.38\, I_{385}\, X_{b17,6,m,k}\, X_{18r,m}\, I_{244j}\, (I_{389k,j} + I_{390k,j} + I_{391k,j}) \,/\, \sum^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

265. $$I_{389k,j} = \sum_{i=1}^{I_{69}} I_{239bi,j,k} \qquad 0 < I_{240b} \leq 3$$

266. $$I_{390k,j} = \sum_{i=1}^{I_{69}} I_{239bi,j,k} \qquad 0 < I_{240b} > 3$$

$$I_{391k,j} = \sum_{k=1}^{I_{274}} I_{241k,j} \qquad I_{241k,j} > 0$$

I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)

$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region

$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region

$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region

$I_{381m,k}$ = transport of contaminant from the N-P, and upper T-B regions of the respiratory tract to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)

X_{18r} = regional particulate concentration, m^{-3}, for a given contaminant

I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
$X_{b17,6,m,k}$ = inhalation rate, m^3 day^{-1}, for the mth age group of the kth animal type
$I_{239bi,j,k}$ = the fraction of the animal's diet contributed by ith plant in the jth region for the kth animal population (input)
I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)
I_{274} = number of animals simulated: to be used as a potential food sources for the kth animal population except for the kth population itself (unless cannibalism is being modeled) in a given region (input)
$I_{241,k,j}$ = initial diet fraction of the kth animal prey type in the food of a specific animal predator type from the jth region (input)

268. $$I_{382m,k} = 1000\, F_{57}\, [(1 - I_{275ka})\, (\sum X_{r13t,j}\, I_{244j}\, I_{389k,j} + \sum_{j=1}^{I_{273}} \sum_{i=1}^{I_{273}\, I_{69}} X_{s13t,i,j}\, I_{390k,j}\, I_{239bk,i,j}\,) + I_{275ka} \sum_{j=1}^{I_{273}} \sum_{l=1}^{I_{274l}} \sum_{m=1}^{I_{248l}} X_{p17,j,m,l}\, I_{395j,m,l}\,]$$

$I_{382m,k}$ = regionally weighted contaminant transport from ingested food to the mth age group of the kth mammalian predator population
I_{248l} = lifespan, years, of the an animal in the lth mammalian prey population (input)
$X_{r13t,j}$ = contaminant concentration, gdwt-1, in and on consumed food from range from the jth region
$X_{s13t,j}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed supplementary feed from the jth region
$X_{p17,j,m,l}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed food for the mth age group from the lth prey of the jth region
F_{57} = transport of food, kgdwt/day, to an animal of a given age group for a given mammalian population
I_{275ka} = digestible fraction from range animal tissue consumption for the kth mammalian population
I_{244j} = fraction of region utilized as a source of food for a specific animal type in a particular year and day time sub-interval for the jth region (input)
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from the jth region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from the jth region
$I_{239bk,i,j}$ = the fraction of the animal's diet contributed by ith plant in the jth region for the kth animal population
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)
I_{274l} = number of animals simulated: to be used as a potential food sources for the lth animal population except for the predator population itself in a given region
$I_{395j,m,l}$ = population reduction factor for the mth age group of prey the lth prey population from the jth region

269. $$I_{396i,j} = [X_{13,5,1}+X_{r13,5,1}+X_{sa13,5,1}+X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1} + (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1})\, X_{5,1} / (X_{4,1}+ X_{5,1})] / [X_{5,1} + X_{6,1} + X_{4,1}\, X_{5,1} /(X_{4,1} + X_{5,1})]$$

270. $$I_{397i,j} = [X_{13,5,2}+X_{r13,5,2}+X_{sa13,5,2} + (X_{13,4,2}+X_{r13,4,2}+X_{s13,4,2}\, X_{5,1} /(X_{4,1} + X_{5,1})] / [\, X_{5,2} + X_{4,2}\, X_{5,1} / (\, X_{5,1} +X_{4,1})]$$

271. $I_{398i,j} = (X_{5,1} / X_{5,1}+X_{4,1}) (X_{13,4,2}+X_{r13,4,2}+X_{s13,4,2}) + X_{13,5,2}+X_{r13,5,2}+X_{sa13,5,2} + X_{13,5,1}+ X_{r13,5,1} + X_{sa13,5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})^2] (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1})$ $\quad I_{240b}= 4$

$= X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1}$ $\quad I_{240b}= 5$

$= [X_{13,5,1}+X_{r13,5,1}+X_{sa13,5,1}+ (X_{13,4,1}+X_{r13,4,1}+X_{s13,4,1}) X_{5,1} (X_{4,1}+ X_{5,1})] / [X_{5,1} + X_{6,1} + X_{4,1} X_{5,1} /(X_{4,1} + X_{5,1})]$ $\quad I_{240b}= 6$

$= X_{13,6,1}+X_{r13,6,1}+X_{sa13,6,1}$ $\quad I_{240b}= 7$

272. $I_{399i,j} = [X_{4,2} X_{5,1} / (X_{5,1} + X_{4,1})] + X_{5,2} + X_{5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})] +X_{6,1} +X_{6,2}$ $\quad I_{240b}= 4$

$= X_{6,1}$ $\quad I_{240b}=5$

$= [X_{4,2} X_{5,1} / (X_{5,1} + X_{4,1})] + X_{5,2} + X_{5,1} + [X_{4,1} X_{5,1} / (X_{5,1} + X_{4,1})]$ $\quad I_{240b}=6$

$= X_{6,2}$ $\quad I_{240b}=6$

X_{ral3t} = contaminant concentration, gdwt-1, in and on consumed food from range from all regions combined

$X_{sp13t,i,j}$ = contaminant concentration, gdwt-1, in and on consumed supplementary feed from the jth region

$X_{c13t,}$ = concentration, $gdwt^{-1}$, of the range feed from all food categories for all plant types combined for a given mammalian population

$I_{258i,j}$ = live plant fraction (including seeds) ingested from the ith plant type from the jth region

I_{273} = number of regions simulated (input)

$I_{259i,j}$ = dry plant fraction (excluding seeds) ingested from the ith plant type from the jth region

$I_{396i,j}$ = concentration, $gdwt^{-1}$, of the live plant fraction of consumed range plants for the ith plant of the jth region

I_{69} = total number of plants in a given region (input)

$I_{397i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed range plants for the ith plant of the jth region

$I_{398i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed supplementary feed for the ith plant of the jth region

$I_{399i,j}$ = total biomass density, gdwt m^{-2}, for supplementary feed produced by the ith plant of the jth region

$I_{269j,j,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

I_{240}, I_{240b} = food preference/selection scheme for a given plant type to be utilized by a specific animal type in a particular year and day time sub-interval (input)

$X_{13,4,1}$ = internal contaminant concentration, m^{-2}, in live stem of a given plant type

$X_{13,5,1}$ = internal contaminant concentration, m^{-2}, in live leaves of a given plant type

$X_{13,6,1}$ = internal contaminant concentration, m^{-2}, in green seeds of a given plant type

$X_{13,4,2}$ = internal contaminant concentration, m^{-2}, in dry stem of a given plant type

$X_{13,5,2}$ = internal contaminant concentration, m^{-2}, in dry leaves of a given plant type

$X_{13,6,2}$ = internal contaminant concentration, m^{-2}, in dry seeds of a given plant type

$X_{r13,5,1}$ = resuspendable contaminant concentration, m^{-2}, on live leaves

$X_{r13,5,2}$ = resuspendable contaminant concentration, m^{-2}, on dry leaves

$X_{r13,4,1}$ = resuspendable contaminant concentration, m^{-2}, on live standing vegetation
$X_{r13,4,2}$ = resuspendable contaminant concentration, m^{-2}, on dry standing vegetation
$X_{r13,6,1}$ = resuspendable contaminant concentration, m^{-2}, on seeds
$X_{r13,6,2}$ = resuspendable contaminant concentration, m^{-2}, on dry seeds
$X_{s13,5,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live leaves
$X_{s13,5,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry leaves
$X_{s13,4,1}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on live standing vegetation
$X_{sa13,5,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry leaves for all plants in a given region from saltation-creep-rainsplash events
$X_{sa13,4,2}$ = weighted average contaminant concentration, m^{-2}, on the surface of dry standing vegetation for all plants in a given region from saltation-creep-rainsplash events
$X_{sa13,6,2}$ = weighted average contaminant concentration, m^{-2}, on the surface
$X_{s13,4,2}$ = saltation-creep-rainsplash contaminant concentration, m^{-2}, on dry standing vegetation
$X_{4,1}$ = live stem biomass density, g /m^2
$X_{4,2}$ = dry stem biomass density, g /m^2
$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{5,2}$ = dry leaf biomass density, g /m^2
$X_{6,1}$ = live seed biomass density, g /m^2
$X_{6,2}$ = dry seed biomass density, g /m^2

273. $$I_{395j,m,l} = I_{246m,l}\ I_{317m,k,l} \ / \ [\ (I_{300m,1,k} + I_{300m,2,k}\ )\ X_{14,m,l}\ ]$$

$I_{395j,m,l}$ = population reduction factor for the mth age group of prey the lth prey population from the jth region
$X_{14,m,l}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the lth mammalian population
$I_{246m,l}$ = fraction of the population in the mth age group of the lth animal population
$I_{317m,k,l}$ = food consumption, kgfwt/day, from predation of the lth prey by the mth age group of the kth mammalian population used as food from a given region
$I_{300m,1,k}$ = number of males in the mth age group of the kth mammalian population
$I_{300m,2,k}$ = number of females in the mth age group of the kth mammalian population

274. $$I_{383m,k} = 3\ X_{14,m,k}\ I_{385} \sum_{j=1}^{I_{273}} I_{389k,j}\ I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\ \ I_{265i,j} \ / \sum_{i=1}^{I_{69}} I_{265i,j}\ ]$$

$$= (3\ X_{14,m,k} \ / \ I_{400}\ )\ I_{385} \sum_{j=1}^{I_{273}} I_{389k,j}\ I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\ \ I_{265i,j} \ / \sum_{i=1}^{I_{69}} I_{265i,j}\ ] \qquad I_{400} > 20$$

275. $$I_{400} = \sum_{j=1}^{I_{273}} I_{389k,j}\ I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\ \ I_{265i,j} \ / \sum_{i=1}^{I_{69}} I_{265i,j}\ ]$$

$I_{383m,k}$ = amount of soil consumed, gdwt day^{-1}, by an animal from grazing

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)
$X_{12,i,1,j}$ = top soil layer contaminant concentration, $gdwt^{-1}$, for the ith plant of the jth region
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from the jth region

$I_{265i,j}$ = coverage area, acres, of the ith plant type for the jth region (input)
$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
I_{400} = sum of all available feed on range, gdwt m^{-2}, from all regions combined based on food utilization factors in each region

276. $$I_{384m,k} = I_{385}\ I_{401m,k} \sum_{j=1}^{I_{273}} I_{402k,j}\ I_{244k,j}\ (I_{389k,j} + I_{390k,j} + I_{391k,j})$$

$I_{384m,k}$ = amount of contaminant, day-1, transported to the hepatic-portal vein via the gut for the mth age group of the kth mammalian population
I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)
$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)
$I_{389k,j}$ = fraction of the kth animal's diet obtained from range plants from a given region
$I_{390k,j}$ = fraction of the kth animal's diet obtained from supplementary feeds from a given region
$I_{391k,j}$ = fraction of the kth animal's diet obtained from animal consumption from a given region
$I_{401m,k}$ = net water requirement, liters day^{-1}, for the mth age group of the kth mammalian population
$I_{402k,j}$ = contaminant concentration, liter-1, in the drinking water of the kth mammal population in the jth region (input)

277. $$I_{401m,k} = 0.04\ X_{14,m,k} - 0.522\ I_{403m,k} - 0.3\ I_{295kr}\ F_{57}$$

$I_{401m,k}$ = net water requirement, liters day^{-1}, for the mth age group of the kth mammalian population
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
F_{57} = transport of food, kgdwt/day, to an animal of a given age group for a given mammalian population
I_{295kr} = weighted nitrogen fraction from range plant food consumption for the kth mammalian population

278. $$I_{403m,k} = 70\ X_{15m,k}\ I_{249k}\ I_{250k}\ /\ 4500$$

$I_{403m,k}$ = metabolic heat production losses expressed in kgdwt day^{-1}
$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
I_{249k} = heat increment factor as a function of environmental temperature for the kth mammalian population
I_{250k} = activity factor for the kth mammalian population (input)

II. EQUATIONS INVOLVING DRIVING VARIABLES

1. D_1 = mean annual precipitation, inches (input)

2. $D_2 = [(I_7 - I_8)/2]\ [\sin(2\pi\{I_7 - I_8\}/365) + 1] + I_8$

   D_2 = mean daily temperature, °F
   I_7 = maximum mean daily temperature on a yearly basis, °F (input)
   I_8 = minimum mean daily temperature on a yearly basis, °F (input)
   I_9 = lag period to maximum mean daily temperature on a yearly basis, days (input)

3. $D_2 = D_2 + I_{10}(I_{11} - I_{12}) + I_{13} + S_3 I_{14}$

   I_{10} = environmental lapse rate, °F per foot from reference site elevation (input)
   I_{11} = reference site elevation, feet amsl (input)
   I_{12} = elevation of test site, feet amsl (input)
   I_{13} = correction for latitude difference between reference and test site, °F (input)
   I_{14} = standard deviation from mean monthly temperature for all months combined, °F (input)
   S_3 = random number drawn from a normal distribution ($m = 0$, $\sigma = 1$)

4. $D_2 = (5/9)(D_2 - 32)$

   D_2 = mean daily temperature, °C

5. $D_3 = [(I_{24} - I_{25})/2]\ \mathrm{Sin}(2\pi\{I_5 - I_{26}\}/365) + 1] + I_{25}$

   D_3 = total solar radiation intensity at site, Langleys / day
   I_5 = Julian day number
   I_{24} = maximum yearly solar radiation intensity at site, Langleys / day (input)
   I_{25} = minimum yearly solar radiation intensity at site, Langelys / day (input)
   I_{26} = lag period to maximum insolation, days

6. $D_4 = \sum_{j=1} D_{3j} / I_{83}$

D_4 = mean daily channel drainage temperature, ° C for a given channel or channel segment

D_{3j} = mean daily regional temperature of the jth region contributing surface runoff to a given channel or channel segment

I_{83} = number of regions supplying a given channel or channel segment with surface runoff (input)

III. EQUATIONS INVOLVING LOGIC AND BOUNDARY VARIABLES

1. $F(P_p, X_{1,1})$ $\quad D_2 > 2°\,C$

   $F(P_p, X_2)$ $\quad D_2 \leq 2°\,C$

$F(P_p, X_{1,1})$ = transport of moisture from precipitation event to soil surface

$F(P_p, X_2)$ = transport of moisture from snow to soil surface

D_2 = mean daily temperature, ° C

IV. EQUATIONS INVOLVING STOCHASTIC VARIABLES

1.

$$S_1 = S_1 / \sum_{j=1}^{I_4} S_j$$

S_1 = normalized random number fraction derived from a normal distribution used to derive the fraction of the assigned mean monthly precipitation to selected day of the month

S_j = random number drawn from a normal distribution ($m = 0, \sigma = 1$) inclusive of S_1

I_4 = mean number of precipitation events in a given month exceeding 0.1 inches

2.

$$S_2 = | S_j I_o |$$

S_2 = Julian day number assigned a precipitation event for a given month based on a random draw from an even distribution

S_j = random number selected from an even distribution (range between 0 and 1)

I_o = number of days in a given month

3.

S_3 = random number selected from a random number distribution ($m = 0, \sigma = 1$)

V. EQUATIONS INVOLVING COMPARTMENTS OR STATE VARIABLES

1. $X_{5,1} = 928\,(X_{4,1} / 1000)\,\{72 + \exp[-3.11\,(X_{4,1} / 1000)]\}$

   $= C_{12}\,X_{4,1}$ $\quad L_5 = 0$

   $= C_{13}\,I_{44}$ $\quad L_5 = 0;\ X_{5,1} > C_{13}\,I_{44}$

$X_{5,1}$ = live leaf biomass density, g /m^2
$X_{4,1}$ = live above ground biomass (stem), g /m^2
L_5 = woody or deciduous plant gate (0,1)
C_{12} = initial fraction of above ground biomass density assigned to leaf biomass density
C_{13} = mean fraction of above ground biomass density assigned to leaf biomass density as a function of a 10 year running average net productivity estimate
I_{44} = 10 year running average net productivity, g /m^2/ year, for a given plant type

2. $X_{4,7} = I_{160}\, I_{161}$
$= 1$ $\qquad X_{4,7} \leq 1$

3. $X_{4,6} = (X_{4,1} - X_{5,1}) - X_{4,7}$
$= 1$ $\qquad X_{4,4} \leq 1$

$X_{4,7}$ = bole component of standing live stem biomass, g /m^2
$X_{4,6}$ = branch-twig component of standing live stem biomass
I_{160} = estimated biomass volume density, cm^3 / m^2 for a given plant at a given time
I_{161} = estimated density of dry plant tissue, g / m^3, for a given plant (input or estimated)
$X_{4,1}$ = live stem biomass density, g /m^2 /d
$X_{5,1}$ = live leaf biomass density, g /m^2

4. $X_{11i} = I_{185} \exp(- 0.0037\, I_{185})$ $\qquad I_{147i} = 1$
$= (I_{186}\, I_{183i})^3 / I_{185}^{\ 2}$ $\qquad I_{147i}=1,\ I_{186} > 1$

X_{11i} = total volume of trees in the ith diameter class, ft^3, i=1
I_{183i} = volume of a unit tree in the ith diameter class, ft^3
I_{185} = total volume of tree stand ,ft^3 /acre
I_{186} = number of trees in even-aged stand, otherwise set to 0.0 (input)

5.
$$X_{11t} = \sum_{i=1}^{n} X_{11i}$$

11_t = total volume of trees from all diameter classes combined, ft^3
X_{11i} = total volume of trees in the ith diameter class, ft^3

6. $X_{15,m,k} = 5.623e\text{-}03\ (1000\ X_{14,m,k})^{0.75}$ $\qquad I_{248k} > 0$

7. $X_{16,m,k} = 5.623e\text{-}03\ (1000\ X_{14e,m,k})^{0.75}$ $\qquad I_{248k} > 0$

8. $X_{14e,m,k} = I_{277k}\ [1 - \exp (-9\ I_{278m,k} / I_{248k})]$ $\qquad I_{248k} > 0$
$= I_{277k}\ \{ 1 - \exp [-9 (I_{278m,k} - 0.006\ I_{248k}) / I_{248k}]\}$ $\qquad I_{248k} > 0;\ m=1$

9. $X_{14,m,k} = X_{14em,k}$ $\qquad I_{29}=1;\ I_5=1$

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{14e,m,k}$ = expected animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population
$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population
I_5 = Julian day number (1 - 365)
I_{29} = Julian year number of specific year of simulation
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)
I_{277k} = maximum body weight, kgfwt, for an animal in the kth mammalian population (input)
$I_{278m,k}$ = age, days, of the mth age group of the kth mammalian population

10. $X_{14,m,k} = 1.023e\text{-}4\ (1000\ X_{15,m,k})^{1.33}$

$X_{14,m,k}$ = simulated animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

11. $X_{15,m,k} = X_{15,m,k} + F_{57} - F_{58} - F_{59} - F_{60} - F_{61}$

$X_{15,m,k}$ = animal metabolic biomass compartment, kgfwt, for an animal in mth age group of the kth mammalian population

12. $X_{sa13,4,6} = I_{b354}\ X_{s13,4,6}$

13. $X_{sa13,4,7} = I_{354}\ X_{s13,4,7}$

I_{354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of branches for various plants
I_{b354} = mean integral average deposition , fraction, for saltation-creep and rainsplash events based on estimated heights of boles (stems) for various plants
$X_{sa13,4,6}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant
$X_{sa13,4,7}$ = integral average saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant
$X_{s13,4,6}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live bole components of a given plant
$X_{s13,4,7}$ = maximum saltation-creep-rainsplash contaminant concentration, m^{-2}, for live branch and twig components of a given plant

14. $X_{sa13,j,k} = \left[\sum_{i=1}^{I_{69}} X_{s13,j,k}\ I_{265i} \right] / \sum_{i=1}^{I_{69}} I_{265i}$ $\qquad O_1 > 0;\ O_2 > 0$

$= X_{13s,j,k}$ $\qquad O_1 = 0;\ O_2 = 0$

O_1 = niche competition option (1,0) (input)
O_2 = succession option (1,0) (input)
I_{265i} = coverage area, acres, of the ith plant type for a given region (input)
I_{69} = total number of plants in a given region (input)
$X_{12,1}$ = top soil layer contaminant concentration, $gdwt^{-1}$
$X_{s13,j,k}$ = contaminant concentration, m^{-2}, on the surface of the jth plant part (roots, stems, leaves, seeds), and kth state (live, dry, dry-lying) for a given plant type in a given region from saltation-creep-rainsplash events

$X_{sal3,j,k}$ = weighted average contaminant concentration, m^{-2}, on the surface of the jth plant part (roots, stems, leaves, seeds), and kth state (live, dry, dry-lying) for a given plant type in a given region from saltation-creep-rainsplash events

15. $X_{v17,1} = 4.29e\text{-}05\ (1000\ X_{14,m,k})^{0.992}$

16. $X_{v17,2} = 4.63e\text{-}04\ (1000\ X_{14,m,k})^{1.009}$

17. $X_{fl7,2} = 1.18e\text{-}01\ (1000\ X_{14,m,k})^{0.81}$

$X_{v17,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{v17,2}$ = perfusion volume of muscle tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,2}$ = perfusion rate, liters hr^{-1}, of blood plasma into muscle tissue for a given animal and age group

18. $X_{rl7,n} = 0.109 / X_{tl7,i,n}$

$X_{rl7,i,n}$ = partition coefficient of the ith organ or tissue for the nth contaminant in muscle of a given animal type
$X_{tl7,i,n}$ = transport coefficient, day^{-1}, of the ith organ or tissue for the nth contaminant in Standard Man (input)

19. $X_{v17,3} = 1.1e\text{-}04\ (1000\ X_{14,m,k})$

20. $X_{fl7,3} = 0.75\ X_{fl7,2}$

$X_{v17,3}$ = perfusion volume of adipose tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,3}$ = perfusion rate, liters hr^{-1}, of blood plasma into adipose tissue for a given animal and age group

21. $X_{v17,4} = X_{v17,5} + X_{v17,6}\ X_{v17,7} + X_{v17,8}$

22. $X_{v17,5} = 2.0e\text{-}06\ (1000\ X_{14,m,k})^{1.043}$

23. $X_{v17,6} = 3.16e\text{-}06\ (1000\ X_{14,m,k})^{1.104}$

24. $X_{v17,7} = 8.65e\text{-}05\ (1000\ X_{14,m,k})^{0.909}$

25. $X_{v17,8} = 4.52e\text{-}06\ (1000\ X_{14,m,k})^{0.901}$

26. $X_{fl7,4} = X_{fl7,5} + 0.1\ X_{fl7,6} + 0.1\ X_{fl7,7} + X_{fl7,8}$

27. $X_{fl7,5} = 8.31e\text{-}03\ (1000\ X_{14,m,k})^{0.965}$

28. $X_{fl7,6} = 0.31\ (1000\ X_{14,m,k})^{0.82}$

29. $X_{fl7,7} = 0.147\ (1000\ X_{14,m,k})^{0.787}$

30. $X_{fl7,8} = 2.58e\text{-}03\ (1000\ X_{14,m,k})^{1.028}$

$X_{v17,4}$ = perfusion volume of residual tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,4}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue for a given animal and age group
$X_{v17,5}$ = perfusion volume of heart tissue, liters, for a given animal and age group
$X_{v17,6}$ = perfusion volume of lung tissue, liters, for a given animal and age group
$X_{v17,7}$ = perfusion volume of gut tissue, liters, for a given animal and age group
$X_{v17,8}$ = perfusion volume of spleen tissue, liters, for a given animal and age group
$X_{fl7,5}$ = perfusion rate, liters hr^{-1}, of blood plasma into heart tissue for a given animal and age group
$X_{fl7,6}$ = perfusion rate, liters hr^{-1}, of blood plasma into lung tissue for a given animal and age group
$X_{fl7,7}$ = perfusion rate, liters hr^{-1}, of blood plasma into gut tissue for a given animal and age group
$X_{fl7,8}$ = perfusion rate, liters hr^{-1}, of blood plasma into spleen tissue for a given animal and age group

31. $X_{v17,9} = 3.3e\text{-}05\ (1000\ X_{14,m,k})^{0.936}$

32. $X_{fl7,9} = 2.36e\text{-}01\ (1000\ X_{14,m,k})^{0.84}$

$X_{v17,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{v17,9}$ = perfusion volume of bone tissues, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,9}$ = perfusion rate, liters hr^{-1}, of blood plasma into both bone marrow and trabecular bone tissue for a given animal and age group

33. $X_{v17,10} = 2.18e\text{-}05\ (1000\ X_{14,m,k})^{0.843}$

34. $X_{fl7,10} = 1.22e\text{-}01\ (1000\ X_{14,m,k})^{0.802}$

$X_{v17,1}$ = volume of blood plasma, liters, for a given animal and age group
$X_{v17,10}$ = perfusion volume of kidney tissue, liters, for a given animal and age group
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population
$X_{fl7,10}$ = perfusion rate, liters hr^{-1}, of blood plasma into kidney tissue for a given animal and age group

35. $X_{fl7,1} = 8.65e\text{-}01(1000\ X_{14,m,k})^{0.778}$

$X_{fl7,1}$ = total plasma flow, liters hr^{-1}
$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

36. $X_{v17,12} = 8.59e\text{-}05\ (1000\ X_{14,m,k})^{0.885}$

37. $X_{fl7,12} = 0.136\ (1000\ X_{14,m,k})^{0.792} + 0.9\ X_{fl7,7}$

$X_{v17,12}$ = perfusion volume of liver tissue, liters, for a given animal and age group

$X_{14,m,k}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the kth mammalian population

$X_{f17,12}$ = perfusion rate, liters hr^{-1}, of blood plasma into liver tissue for a given animal and age group including hepatic-portal vein contribution from the gut

$X_{f17,7}$ = perfusion rate, liters hr^{-1}, of blood plasma into gut tissue for a given animal and age group

38. $X_{b17,6,m,k} = 1.4\, X_{16,m,k}$

$X_{b17,6,m,k}$ = inhalation rate, $m^3\ day^{-1}$, for the mth age group of the kth animal type

$X_{16,m,k}$ = expected animal metabolic biomass compartment, kgfwt, of an animal in the mth age group of the kth mammalian population

39. $$X_{ral3t} = \sum_{j=1}^{I_{273}} \sum_{n=1}^{3} I_{269j,j,k}\, X_{cl3t,n,j} / I_{254}$$

40. $$X_{spl3t,i,j} = I_{398i,j} / I_{399i,j}$$

41. $$X_{cl3t} = \sum_{j=1}^{I_{273}} \sum_{i=1}^{I_{69}} I_{258i,j}\, I_{396i,j} + I_{259i,j}\, I_{397i,j}$$

X_{ral3t} = contaminant concentration, $gdwt^{-1}$, in and on consumed food from range from all regions combined

$X_{spl3t,i,j}$ = contaminant concentration, $gdwt^{-1}$, in and on consumed supplementary feed from the jth region

$X_{cl3t,}$ = concentration, $gdwt^{-1}$, of the range feed from all food categories for all plant types combined for a given mammalian population

$I_{258i,j}$ = live plant fraction (including seeds) ingested from the ith plant type from the jth region

I_{273} = number of regions simulated (input)

$I_{259i,j}$ = dry plant fraction (excluding seeds) ingested from the ith plant type from the jth region

$I_{396i,j}$ = concentration, $gdwt^{-1}$, of the live plant fraction of consumed range plants for the ith plant of the jth region

I_{69} = total number of plants in a given region (input)

$I_{397i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed range plants for the ith plant of the jth region

$I_{398i,j}$ = concentration, $gdwt^{-1}$, of the dry plant fraction of consumed supplementary feed for the ith plant of the jth region

$I_{399i,j}$ = total biomass density, gdwt m^{-2}, for supplementary feed produced by the ith plant of the jth region

$I_{269j,j,k}$ = food contribution, kgdwt /day, from the jth range plant/animal category from the mth region combined for the kth mammalian population

I_{254k} = food available, kgdwt, on a given day for the kth mammalian population

42. $$X_{p17,j,m,l} = X_{17,1j,m,l} + X_{17,2,j,m,l} + X_{17,3,j,m,l} + X_{17,4,j,m,l} + X_{17,10,j,m,l} + X_{17,12,j,m,l} + X_{1e17,6j,m,l} + X_{2e17,6j,m,l} + X_{3e17,6,j,m,l} + X_{17,9j,m,l} + X_{17,13,j,m,l} + X_{17,14j,m,l}$$

$$= X_{17,1j,m,l} + X_{17,2,j,m,l} + X_{17,3,j,m,l} + X_{17,4,j,m,l} + X_{17,10,j,m,l} + X_{17,12,j,m,l} + X_{1e17,6j,m,l} + X_{2e17,6j,m,l} + X_{3e17,6,j,m,l} + 0.1X_{17,9j,m,l} + 0.1X_{17,14j,m,l} \qquad X_{14m,l} > 2$$

$X_{p17,j,m,l}$ = contaminant concentration, gdwt^{-1}, in and on consumed food of the mth age group from the lth prey of the jth region

$X_{14,m,l}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the lth mammalian population

$X_{1e17,6,j,m,l}$ = contaminant content on the surface of the bronchus region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population

$X_{2e17,6,j,m,l}$ = contaminant content on the surface of the bronchiole region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population

$X_{3e17,6,j,m,l}$ = contaminant content on the surface of the pulmonary region of the lung lumen as particulates from the jth region for the mth age group of the lth prey population

$X_{17,2,j,m,l}$ = contaminant content of muscle tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,1,j,m,l}$ = contaminant content of blood plasma, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,3,j,m,l}$ = contaminant content of adipose tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,4,j,m,l}$ = contaminant content of residual tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,10,j,m,l}$ = contaminant content of kidney tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,9,i,m,l}$ = contaminant content of bone tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,12,n}$ = contaminant content of liver tissue, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,13,j,m,l}$ = contaminant content of gut ingesta, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

43. $X_{17,13,j,m,l} = I_{379} / I_{385}$

44. $$X_{17,14j,m,l} = 50\, X_{14,m,l} \sum_{j=1}^{I_{273}} (I_{389k,j} + I_{390k,j} + I_{391k,j})\, I_{244k,j} \sum_{i=1}^{I_{69}} [X_{12,i,1,j}\, I_{265i,j} / \sum_{i=1}^{I_{69}} I_{265i,j}]$$

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{17,13,j,m,l}$ = contaminant content of gut ingesta, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

I_{385} = fraction of a given contaminant transferred, day^{-1}, from the gut to the hepatic-portal vein (input)

I_{379} = total transport of contaminant, day^{-1}, from the gut to the liver via the hepatic-vein

$X_{17,14j,m,l}$ = contaminant content of animal pelt, mass or activity units, for the mth age group of the lth prey population of the jth region contaminant

$X_{14,m,l}$ = animal biomass compartment, kgfwt, for an animal in the mth age group of the lth mammalian population

$X_{12,i,1,j}$ = top soil layer contaminant concentration, gdwt^{-1}, for the ith plant of the jth region

$I_{244k,j}$ = fraction of region utilized as a source of food for a kth animal type in a particular year and day time sub-interval for the jth region (input)

$I_{265i,j}$ = coverage area, acres, of the ith plant type for the jth region (input)
I_{69} = total number of plants in a given region (input)
I_{273} = number of regions simulated (input)

45. $X_{ft14k} = 0.05256\, I_{277k} \{ 1 - \exp[-9 (\sum_{m=1}^{I_{307k}} m / I_{307k})] \}$

X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population
I_{277k} = maximum body weight, kgfwt, for an animal in the kth mammalian population (input)
I_{307k} = estimated gestation period, days, for the kth mammalian population

46. $X_{ft17,1k} = 0.2\, X_{ft14k}\, X_{ftc17,1,k}$

47. $X_{ftc17,1,k} = (\sum_{m=2}^{I_{248k}} X_{fl7,4,m,k} X_{17,1,m,k} / [X_{v17,1,m,k} X_{fl7,1,m,k} \{\sum_{i=1}^{8} I_{322k,i} / \sum_{l=1}^{8} I_{322k,l} \}]) + (X_{fl7,4,1,k} X_{17,1,1,k} /$

$[X_{v17,1,1,k} X_{fl7,1,1,k} \{\sum_{i=1}^{8} I_{322k,i} / \sum_{l=1}^{8} I_{322k,l} \}\{ \sum_{j=1}^{3} I_{326bkj} / \sum_{n=1}^{3} I_{326bkn} \}])$

$0.15\, I_{248k} \le m \le 0.75\, I_{248k}$; $0.15\, I_{248k} \le I_{312bkn} \le 0.75\, I_{248k}$

$X_{ft17,1,k}$ = contaminant content, mass or activity units, in fetus blood plasma of the kth mammalian population for a given contaminant
$X_{17,1,m,k}$ = contaminant content of blood plasma, mass or activity units, for the mth age group of reproductive females for a given contaminant
$X_{ftc17,1,k}$ = weighted contaminant concentration, liter-1, in the fetus blood plasma of the kth mammalian population for a given contaminant
X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population
$X_{fl7,4,m,k}$ = perfusion rate, liters hr^{-1}, of blood plasma into residual tissue, of the mth age group of reproductive females of the kth mammalian population

$X_{v17,1,m,k}$ = volume of blood plasma, liters, of the mth age group of reproductive females of the kth mammalian population
$X_{fl7,1,m,k}$ = total plasma flow, liters hr^{-1} , of the mth age group of reproductive females of the kth mammalian population
I_{248k} = lifespan, years, of the an animal in the kth mammalian population (input)

$I_{322k,l}$ = fraction of the reproductive females in the *l*th reproductive age grouping for the kth mammalian population
$I_{312k,n}$ = age, days, of newborns from the nth gestation period during a given year for the kth mammalian population
$I_{326bk,n}$ = number of female animals in the nth sub-group of the first age group for the kth mammalian population

48. $X_{wn17,1k} = 0.1\, X_{14m,k}\, X_{wnc17,1k}$ $\qquad m=1;\ I_{304k,n} \le I_{313k,n} \le I_{304k,n} + 0.025\, I_{248k}$

49. $X_{wnc17,1,k} = X_{ftc17,1,k}$

X_{ft14k} = body mass, kgfwt, of the fetus of the kth mammalian population
$X_{wn17,1k}$ = contaminant transport, mass or activity units day^{-1}, from mother's milk to a weaning newborn animal of the kth mammalian population

$X_{ftc17,1,k}$ = weighted contaminant concentration, liter-1, in the fetus blood plasma of the kth mammalian population for a given contaminant

$X_{wnc17,1,k}$ = weighted contaminant concentration, liter-1, in weaning milk of the kth mammalian population for a given contaminant

$I_{304k,n}$ = time delay, days, for newborns from one litter to the next for the nth litter

$I_{313k,n}$ = fetal age, days, of the nth gestation period during a given year for the kth mammalian population

12.0. APPENDIX C

VARIABLES AND THEIR MNEMONIC EQUIVALENTS

I. Intermediate Variable listing (I_i)

I_i	Array Size	Code Symbol	Main Code Location	
1	12	X	7	(input)
2	n	SNM	7	
3	1	DELTT	7	
4	12, n	TRE	7	(input)
5	1	AIII	all	
6	n	PRCV	7	(input)
7	n	T4	7	(input)
8	n	T3	7	(input)
9	n	L2	7	(input)
10	n	ALPSR	7	(input)
11	n	RELV	7	(input)
12	n	G1	7	(input)
13	n	L3	7	(input)
14	n	TMXV	7	(input)
15	1	HO	7	
16	1	APSYC	7	
17	n	E2	7	
18	1	PMBAR	7	
19	i,n	TLAI	13	
20	i,n	RCODE	16	(input)
21	i,ks,n	ZFF	16	
22	i,ks,n	W1	16	(input)
23	i,ks,n	H	16	
24	n	T2	7	(input)
25	n	T1	7	(input)
26	n	L1	7	(input)
27	i,ks,n	RPNTP	16	
28	1	DEPSUM	1	
29	1	AI1	1	
30	i,ks,n	CZ	16	
31	i,ks,n	YM	16	
32	i,ks,n	WFR	16	
33	i,ks,n	ROCK	16	(input)
34	i,ks,n	SAND	16	(input)
35	i,ks,n	SILT	16	(input)
36	i,ks,n	CLAY	16	(input)
37	i,ks,n	SLICE	16	(input)
38	1	CN2	14	
39	1	ACN2	14	
40	1	BCN2	14	
41	n	AKOND	14	
42	1	BCN3	14	
43	i,n	CCVV	13	

43m	i,n	CCMAX	13	
44	i,n	WURAT ($L_2 \leq 1$)	13	
44	i,n	TPGNW ($L_2 > 1$)	13	
45	1	NN	14	
46	i,n	C6	10	
47	1	AN	10	
48	i,kt,n	DTBR	10	
49	1	CN1	14	
50	n	BDRK	16	(input)
51	1	SMX	14	
52	i,n	SIR	14	
53	1	WIS	14	
54	1	AWIS	14	
55	1	SLCM	14	
56	i,n	SDPTH	16	(input)
57	1	CHNNU	16,11	(input)
58	i,ks,n	F1	16	(input)
59	1	ARRX	11	
60	n + chnnu	SCHAN	11	(input)
61	i,n	SCVR	11	
62	1	DPX	11	
63	n	XCHAN	11	(input)
64	n	APLX	8,11	
65	n	TCVR	8,11	
66	1	WATVX	11	
67	1	ARX	11	
68	n	RAREA	11	
69	n	APL	1	(input)
70	1	ATRN	11	
71	1	BTRN	11	
72	1	FTRN	11	
73	1	AKDCF	11	
74	n	WCHAN	11	(input)
75	1	DUHR	11	
76	1	VMEAN	11	
77	1	AUCI	11	
78	1	BCTR	11	
79	1	ALNC	11	
80	1	CHAREX	11	
81	n	TCH	11	(input)
82	n	TBCH	11	(input)
83	n	REGENT	11	(input)
84	1	C11	4	
85	1	HPX	4	
86	1	C22	4	
87	1	C33	4	
88	1	H0H	4	
89	1	HXH	4	
90	1	DCH	4	
91	1	RCH	4	
92	n + chnnu	HTP	4	(input)
93	n + chnnu	HBT	4	(input)
94	n + chnnu	BCH	4	(input)
95	n + chnnu	BBCH	4	(input)

96		n + chnnu	C1HR	11	(input)
97		n + chnnu	C2HR	11	(input)
98		n + chnnu	VAQF	5	
99		n + chnnu	HTX	5	
100		1	C111	5	
101		1	C222	5	
102		n +chnnu	VAQT	5	
103		1	FXX	5	
104		1	FXD	5	
105		n + chnnu	ATX	5	
106		n + chnnu	WTX	5	
107		n + chnnu	C5HR	8	(input)
108		n	AKSOIL	8	(input)
109		n	CSOIL	8	(input)
110		n	PSOIL	8	(input)
111		n	ALSOIL		8
	(input)				
112		7, n + chnnu	GSBD	8	
113		n + chnnu	GSBDTT	8	
114		7, n + chnnu	DSF	8	(input)
115		1	SSTRES		8
116		n + chnnu	VCHAN	8	
117		1	BSDI	8	
118		1	SSTRI	8	
119		1	RHYRG	8	
120		1	RHYRB	8	
121		1	RCHANG	8	
122		1	RCHANB	8	
123		1	RHYRC	8	
124		n + chnnu	RCHAN	8	(input)
125		n + chnnu	DCHAN	8	
126		1	RCHANB	8	
127		1, n + chnnu	DSZ	8	(input)
128		7, n + chnnu	DSZ	8	(input)
129		1	ALPHI	8	
130		1	GSBD50	8	
131		1	BSD50	8	
132		1	SSTR50	8	
133		1	SA	13	
134		i,n	E5	13	
135		i,n	SFFP	13	
136		i,n	EFFP	13	
137		i,n	ATTT	13	
138		i,16 (1)	DIRA	13	
139		i,n	CUTMAX	13	
140		i,n	FFP	13	(input)
141		1	WATRQ	13	
142		1	EFONUT	13	
143		1	SNCN	13	
144		1	SPCN	13	
145		i,n	T5P	13	
146		1	EAMN	13	
147		i,n	CFAC	13	
148		i,n	SFAC	13	

149	i,n	ETRANS	13
150	1	ETRN	13
151	i,n	WURAT	13
152	i,n	WYAV	13
153	i,n	SYAV	13
154	i,n	DIRMAX	13 (input)
155	1	H2	13
156	1	H3	13
157	1	TSRF	13
158	1	EVRAT	13
159	1	BBBS	13
160	1	VCM3	15
161	1	C9DEN	15
162	1	TRM2	15
163	1	BMNM	15
164	1	HTAVV	15
165	i,n	TREES	10 (input or est.)
166	i,n	BMN	10
167	i,n	HTAV	10,15
167m	1	HTAVX	13
168	i,n	C9	10,15 (input)
169	1	H2L	15
170	1	H3L	15
171	1	ROTFR	15
172	1	TEMPL	15
173	i,ks,n	ZFF	15,16
174	i,ks,n	ZF	15,16 (input/est.)
175	1	ZFLL	16
176	i,n	SOILF	16
177	1	SDPSM	15
178	1	ACL	15
179	i,n	RSS	13 (input)
180	kt,n	XX	10
181	1	GRWFAC	10
182	i,n	XJ1	10 (input)
183	kt,n	Z	10
184	i,n	V8	10 (input)
185	1;i,n	V4; BBP(prev. yr.)	10
186	i,n	TNTR	10 (input)
187	1	DGRWTH	10
188	i,n	C8	10
189	kt,n	HGT	10
190	1	HAAV	10
191	1	BAAV	10
192	1	BATR	10
193	1	PEXCH	15
194	i,ks,4,n	Z1	15
195	i,ks,4,n	Z1	15
196	1	UPPL	15
197	1	CONPHL	15
198	1	UPPL(2)	15
199	1	TMPEF	15
200	1	PLEAK	15
201	1	SMPEF	15

202	1	WNEFF	15	
203	1	EFNIT	15	
204	1	UFP	15	
205	1	UPNI	15	
206	1	DENNO3	15	
207	1	UPNH	15	
208	1	DENNH4	15	
209	i,ks,2,n	Z1	15	
210	i,ks,3,n	Z1	15	
211	1	UPNI(2)	15	
212	1	BPNI	15	
213	1	WTEFF	15	
214	1	UPNH(2)	15	
215	1	BPNH	15	
216	1	FNI2	15	
217	1	FNH2	15	
218	i,ks,n	PHI	16,15	
219	1	THEL	16	
220	1	THEF	16	
221	1	THEW	16	
222	1	SNCN	15	
223	1	GROPER	15	
224	1	SFRP	15	
225	1	SPBF	15	
226	1	SBRP	15	
227	1	TMPEF(2)	15	
228	1	WTEFF(2)	15	
229	1	DSBF	15	
230	1	DTMPEF	15	
231	1	DWTEFF	15	
232	ko	RZOON	3	(input)
233	ko	EATIN	3	(input)
234	ko	FODIN	3,17	(input)
235	ko,32	YRFED	3,17	(input)
236	ko,32	ONFOD	3.17	(input)
237	1	AISCH	3	(input)
238	ko,8,32,n	PLTUSD	3,17	(input)
239	ko,8,32,n	PLTF	3,17	(input)
239b	ko,i,n	PUTFRC	3,17	
240	ko,8,32,n	PLTRNK	3,17	(input)
240b	ko,i,n	PUTZD	3,17	
241	ko,7,n	AUTZD	3	(input)
242	7,n	QFRAC	17	
243	1	CNSM	17	
244	ko,lz	RFRZ	3,17	(input)
245	ko	OTOTA	3,17	
246	1	XINS	3,17	
247	ko	ZOO	3,17	(input)
248	ko	ALSP	3,17	(input)
249	1	HEAT	3	
250	ko	ACTI	3	(input)
251	1	CDMX	3	
252	1	CFX	3	
253	ko	DIGE	3,17	

254	ko	OTOTB	3,17	
255	ko	OTOT	17	
256	ko	OTOTS	17	
257	i,n	Q10	3,17	
258	i,n	G	17	
259	i,n	D	17	
260	7,n	QSUM	3,17	
261	1	P2	17	
262	1	P3	17	
263	1	B1SD	17	
264	ko,i,n	HRVST	17	
265	i,n	ACRES	3,13,17	(input)
266	i, (i x n)	CUTA	3,13,17	(input)
267	ko,n	AGZ	3,17	
268	ko,n	AGR	3,17	
269	7,n	UP	3,17	
270	1	AFRAC	17	
271	1	FADJ	17	
272	ko	OTOTBS	17	
273	1	APX	1	(input)
274	1	NAN	3	(input)
275	ko	GRNA	17	
276	ko	GTDR	17	
277	ko	RR	3	(input)
278	kr,ko	AGEM	3	
279	kr,ko,2	A	3,1	(input/est.)
280	2,ko	POP	3,17	
281	ko	SXRT	3	(input)
282	1	SFC	3	
283	1	AGEO	3	
284	1	AGEMI	3	
285	ko	POPT	3	
286	ko	AGEA	3,17	(input)
287	ko	GASLOS	3	(input)
288	ko	URNLOS	3	(input)
289	1	W5	3	
290	1	AMD(1)	3	
291	1	AMPR	3	
292	1	PRTD	3	
293	1	AMCH	3	
294	1	AMCHR	3	
295	ko	ANTRO	3,17	
296	ko,n	TNIT	17	
297	i,n	CNIT	17	
298	i,n	SNIT	17	
299	1	AMD(2)	3	
300	kr,2,ko	TPOP	3	
301	1	SFR	3	
302	1	SFRF	3	
303	1	AGEMRI	3	
304	ko,2	CAGM	3	
305	ko	AGCOR	3	
306	ko,2	GST	3	(input)
307	ko	GESTP	3	(input)

308	ko	GSTN	3	(input)
309	kr,ko	TQX	3	
310	ko	POP2R	3	
311	2,ko	POPPI	3	
312	ko,3	AFFSP	3	
313	ko,2	AIIIR	3	
314	kr,ko	TQS	3	
315	1	STRV1	3	
316	1	STRV2	3	
317	1	QC	3	
318	1	DRYF	3	
319	kr,ko	TQP	3	
320	1	AINSM	3,19	
321	8,ko,2	BRRT	3	
322	8,ko	SFD	3	
323	1	SREP	3	
324	1	TREP	3	
325	2,ko	BIRTHS	3	
326	ko,3,2	OFFSP	3	
327	1	IRP	3	
328	ko	JAGE	3	
329	m,n	AIRCA	2	
330	m,n	AIRCG	2	
331	m,n	CHRSD	2	
332	m,n	CHRED	2	
333	m,n	CHRSY	2	
334	m,n	CHREY	2	
335	m,n	ARIN	2	
336	m,n	AEXP	2	
337	m,n	ARFL	2	
338	n	DNN	2,7	
339	m,n	ARLG	2	
340	m,n	EQTYP	2	
341	m,n	AEXY	2	
342	m,n	DAMPE	2	
343	m,n	DAIC	2	
344	16,m,n	DURC	2	
345	m,n	YRCNT	2	
346	m,n	CYR	2	
347	1	DPVEL	15	
348	n	CVTT	15	
349	1	CVTRE	15	
350	1	CVTREB	15	
351	1	HTAVM	15	
352	1	CM2PR	15	
353	i,m,n	RSUSP	15	
354	1	A3AVI, BB3AVI	15	
354b	1	B3AVI	15	
355	1	BRAT	15	
356	1	HT3	15	
357	i,m,n	SALCRP	15	
358	m,n	SLOSS	15	
359	m	ATNO	15	
360	1	WATU	15	

361	l	WATD	15	
362	m	AKD	15	
363	m,n	AIRLOS	15	
364	m	VFE	15	
365	m	HENRC	15	
366	m	DIFMOL	15	
367	m	AKOC	15	
368	m	AWT	15	
369	m,n	RGAIN	15	
370	i,ks,m,n	ZU	15	
371	m,2	Y1	15	
372	l	ADD	15	
373	l	CZ3O2	15	
374	l	CZ2O	15	
375	l	WZ3O	15	
376	l	CZ3O	15	
377	l	CZ3O1	15	
378	l	DBO	3,12	
379	l	DBI	3,12	
380	m	AL4	3	
380b	m	AL5	3	
381	m	CAR	3,12	
382	m	C99	3,12	
383	m	C66	3,12	
384	m	C56	3,12	
385	ko,m	GTBL	3,12	(input)
386	ko,m	PLTG1	3,12	(input)
387	ko,m	PLTG2	3,12	(input)
388	ko,m	PLTBL	3,12	(input)
389	n	SPTFRR	3,17	
390	n	SPTFRF	3,17	
391	n	SPTFRA	3,17	
392	ko	PLCM1	3,12	(input)
393	ko	PLCM2	3,12	(input)
394	ko	PLCM3	3,12	(input)
395	l	WADJ	3	
396	l	SACTG	17	
397	l	SACTD	17	
398	l	ZZA	17	
399	l	B1SD	17	
400	l	TSUMB	3	
401	l	WATUPT	3	
402	ko,m,n	WATCNR	3	(input)
403	l	RSPM	3	

II. Driving Variables Listing (D_i)

D_i	Array Size	Code Symbol	Main Code Location
1	n	R	7 (input)
2	n	T	all
3	1	SOLRAD	7,13
4	n	TCHAN	11

III. Flows Listing (F_i) for Flows Identified as Specific Variables

F_i	Array Size	Code Symbol	Main Code Location
1	n	P	7
2	n	F7	7
3	n	SNM	7
4	i,n	EVAP	16
5	i,n	EA (total)	16
5a	i,ks,n	EAVAL (soil layer)	16
5t	1	EASUM	16
6	i,n	EA (total)	16
6a	i,ks,n	EAVAL (soil layer)	16
7	i,ks,n	WATAKD (downflow)	16
7a	i,ks,n	WATAKU (upflow)	16
8	i,n	RNOFDS	14
9	i,n	WSRP	16
10	1	WATVA	11
11	n	RFDCF	11
12	n	RFDLX	11
13	n	VTRND	11
14	n	AICE	11
15	1	AMELT	11
16	n + chnnu	VLOSS	11
17	n + chnnu	CFSPD	11
18	n + chnnu	TONFLW	11
19	7, n + chnnu	GSBD	11
20	n	DSMT	11
21	i,n	B5	13
22	i,n	$B5_{adj}$	13
23	i,n	$B5_{net}$	13
24	i,n	B7	13
25	i,n	$B5_{st}$	13
26	i,n	$B5_{sd}$	13
27	i,n	WD3	13
28	i,n	D3	13
29	i,n	SD3	13
30	i,n	RD3	13
45	i,n	$B5_{nic}$	13

48	1	RVJ	9
49	1	VTOT	9
56	1	FDINE	3
57	kr,ko	FDIN	3
58	1	RSPM	3
59	1	GASLS	3
60	1	URNLS	3
61	1	FECLS	3
85a	1	WATD	15
85b	1	WATU	15
87	m,n	RGAIN	15
92	1	FLXMS	3,12
93	1	FLXFT	3,12
94a	1	FLXRS	3,12
94b	1	DBMK	3,12
95a	1	FLXBN1	3,12
95b	1	FLXBN2	3,12
96a	1	FLXKD1	3,12
96b	1	FLXKD2	3,12
97	1	DBU	3,12
98	1	FLXLV	3,12

IV. Logic and Boundary Variables Listing (Li)

L_i	Array Size	Code Symbol	Main Code Location
1	1	$2 \leq T > 2$	7
2	i,n	RGC (1,2,3) (input)	13
3	1	BAP (0,1)	13
4	i,n	ASTP (0,1), for plants 1 - 22: 0,0,0,0,1,0,0,0,1,0,0,0,0,1,1,0, 1,0,0,1,0,0	13
5	i,n	SCTP (0,1), for plants 1 - 22: 1,1,1,0,0,0,1,1,0,0,1,1,1,0,0,0, 0,1,1,0,0,1	13

V. Stochastic Variables Listing (S_i)

S_i	Array Size	Code Symbol	Main Code Location
1	n	G5	7
2	24,n	NX	7
3	n	G4	7

VI. User Option Variables Listing (O_i)

O_i	Array Size	Code Symbol	Main Code Location
1	n	AICHE(1,0) (input)	13
2	n	RSCDE(1,0) (input)	13
2a	n	ADSUC(1,0) (input)	13
5	i,n	RIRRC(0,1,2) (input)	13
6	n	AIRX (0,1) (input)	1
7	m	ARCZ (0,1) (input)	2
8	m,n	CHRAC(0,v) (input)	2
9	m,n	CHRAG(0,v) (input)	2
10	m,n	CYR(0,v) (input)	2

VII. State And Associated Variables Listing (X_i / $X_{i,j}$)

X_i / $X_{i,j}$	Array Size	Code Symbol	Main Code Location
1	i, ks,n	WA	16
2	n	F7	7
3,1	i,n	R1	13
3,2	i,n	RD1	13
3,3	i,n	RD2	13
4,1	i,n	B1	13
4,1m	i,n	B1SD	13,15
4,2	i,n	D1	13
4,3	i,n	D1GT	13
4,4	i,n	D1GBB	13
4,5	i,n	D1GB	13
5,1	i,n	WND	13
5,2	i,n	WD1	13
6,1	i,n	SD	13
6,2	i,n	SD1	13
7,1	i,ks,n	RH1	13,15
7,2	i,ks,n	RH2	13,15
8	n+chnnu	VTRND	11
9	n+chnnu	AICE	11
10	n+chnnu	VAQF	5
11	i,kt,m	VTMBR	10
12	i,ks,m,n	Z1	16
13,3,1	i,m,n	Z2	15
13,4,1	i,m,n	Z3	15
$_r$13,4,1	i,m,n	B2 + BB2	15
$_s$13,4,1	i,m,n	B3 + BB3	15
$_{sa}$13,4,6	i,m,n	B3AV	15
$_{sa}$13,4,7	i,m,n	BB3AV	16
13,4,2	i,m,n	Z4	15
$_r$13,4,2	i,m,n	Z42	15
$_{sa}$13,4,2	i,m,n	Z43	15
13,4,3-5	i,m,n	Z4G	15
$_r$13,4,3-5	i,m,n	Z4G2	15
$_{sa}$13,4,3-5	i,m,n	Z4G3	15
13,5,1	i,m,n	WZ3	15
$_r$13,5,1	i,m,n	A2	15
$_s$13,5,1	i,m,n	A3	15
$_{sa}$13,5,1	i,m,n	A3AV	15
$_r$13,5,2	i,m,n	WZ42	15
$_{sa}$13,5,2	i,m,n	WZ43	15
13,6,1-2	i,m,n	ZSD	15
$_{sa}$13,6,1-2	i,m,n	AFSAV	15
$_r$13,6,1or 2	i,m,n	AFR	15
$_s$13,6,1or 2	i,m,n	AFS	15
13,7,1	i,m,n	Z5	15
13,7,2	i,m,n	Z5R	15

$_{ra}$13t	ko,i,m,n	SPAC	3,17
$_{su}$13t	ko,i,m,n	SPAS	3,17
14	kr,ko	WT	3,17
14e	kr,ko	WT1	3
$_{ft}$14	ko	WT	3,17
15	kr,ko	AMW	3,17
16	kr,ko	AMX	3,17
17,1	kr,ko,m	V	3,17
$_{v}$17,1	1	VOLPLS	3,17
$_{f}$17,1	1	FLOPLS	3,17
$_{wn}$17,1	kr,ko,m	VMK	3,17
17,2	kr,ko,m	VMS	3,17
$_{f}$17,2	1	FLOMUS	3,17
17,3	kr,ko,m	VFAT	3,17
$_{f}$17,3	1	FLOFAT	3,17
17,4	kr,ko,m	VR	3,17
$_{f}$17,4	1	FLORES	3,17
$_{v}$17,5	1	VOLHRT	3,17
$_{f}$17,5	1	FLOHRT	3,17
$_{v}$17,6	1	VOLLUN	3,17
$_{1e}$17,6	kr,ko,m	Q1	3,17
$_{2e}$17,6	kr,ko,m	Q2	3,17
$_{3e}$17,6	kr,ko,m	Q3	3,17
$_{f}$17,6	1	FLOLUN	3,17
$_{v}$17,7	1	VOLGUT	3,17
f17,7	1	FLOGUT	3,17
$_{v}$17,8	1	VOLSPL	3,17
$_{f}$17,8	1	FLOSPL	3,17
17,9a	kr,ko,m	VBN1	3,17
17,9b	kr,ko,m	VBN2	3,17
$_{f}$17,9	1	FLOMAR	3,17
17,10$_{a}$	kr,ko,m	VKD1	3,17
17,10$_{b}$	kr,ko,m	VKD2	3,17
$_{f}$17,10	1	FLOKID	3,17
$_{u}$17,11	kr,ko,m	VURIN	3,17
17,12	kr,ko,m	VLV	3,17
$_{f}$17,12	1	FLOLIV	3,17
17,13	kr,ko,m	VGUT	3,17
17,14	kr,ko,m	VPELT	3,17
18$_{r}$	m,n	AIRCA	2
18$_{g}$	m,n	AIRCG	2
18$_{r}$	i,m,n	A1	15
18$_{g}$	i,m,n	A1G	15
18$_{s}$	i,m,n	AS	15

VIII. Constants Listing (C_i)

C_i			Value/Units/Id
0	snowmelt intercept		1.1128 mm
1	snowmelt regression coefficient		11.28 mm/ ° C
2	albedo		0.2
3	mean life-span of plant type		years, HF
4	mean vertical growth rate of roots	$C_3 \leq 1$	12.7 mm / day, RPNTP
4		$1 < C_3 \leq 50$	7.62
4		$50 < C_3 \leq 150$	2.54
4		$C_3 > 150$	1.0
5	optimum temperature for photosynthesis	for plant types 1 - 22: (° C), TOPT 30,25,25,21,25,21,30,30,21,21,21,18,18,14,18,15, 12,18,18,15,15,18	
6	difference between optimum and lowest temperature tolerance for photosynthesis	for plant types 1- 22: (° C), TOFL 20,15,15,11,25,11,20,20,21,11,11,10,10,14,18,10, 12,10,10,15,10,11	
7	difference between the highest tolerance temperature and the optimum temperature for photosynthesis	for plant types 1 - 22: (° C), TOFH 9,9,9,13,9,13,9,9,13,13,13,16,16,20,16,19,22,16,16, 19,19,16	
8	water utilization efficiencies for a given plant type	for plant types 1 - 22: , WUTP 200,200,200,150,300,150,200,200,300,150,200,200, 200,300,300,150,300,200,200,300,150,100 (ml-water/ g-photosynthate)	
9	photosynthetic efficiency factor	for plants 1 -22: , PSEP 0.5,1,1,1,1,1,0.5,0.5,1,1,1,1,1,1,1,1,0.5,0.5,1,1,1,1	
10	maintenance respiration fraction	for plants 1 - 22: , BCP .6,.6,.6,.4,.4,.4,.6,.6,.4,.4,.6,.6,.6,.4,.4,.4,.4,.6,.6,.4, .4,.6	
11	photosynthate transport fraction to stems	for plants 1 -22: , STRANS .75,.75.,.3,.85,.85,.75,.5,.3,.75,.85,.5,.5,.3,.85,.85.75, .85,.3,.5,.75,.85,.80	
12	initial leaf biomass density factor	for plants 1 -22: , FLEAF 0,0,0,.25,.125,.25,0,0,.25,.25,0,0,0,.125,.125,.25, .125,0,0,.25,.25,0	
13	mature leaf biomass density factor	for plants 1 -22: , ULEF 0,0,0,.4,.5,.4,0,0,.5,.4,0,0,0,.5,.5,.4,.5,0,0,.5,.4,0.	
14	live-to-dry leaf/stem conversion factor	for plants 1 -22: , RESPF .05,.05,.05,5,5,5,.1,.05,5.,5,.05,.05,.05,5,5,5 5,.05,.10,9.,5.,.1	
15	live-to-dry deciduous leaf conv. factor	for plants 1 - 22: , RSPF2	

		.05,.05,.1,.1,.1,.1,.1,.1,.1,.1,.05,.05,.1,.1,.1, .1,.1,.1,.1,.1,.1,.1
16	humus mineralization constant	for plants 1 - 22:, RFP .0013,.0013,.0013,.0013,.0013,.0013, .0013,.0013,.0013,.0013,.0013 .0054,.0054,.0054,.0054,.0054,.0054, .0054,.0054,.0054,.0054,.0054
17	humus mineralization exp. constant	for plants 1 - 22:, RXPP .161,.161,.161,.161,.161,.161,.161, .161,.161,.161,.161 .12,.12,.12.,.12,.12,.12,.12,.12,.12,.12,.12
18	plant identification	for plants 1 - 22:, PLTID (1 -10 = warm season; 11 -22 = cool season) 1= corn or sorghum, 2= annual vegetables and grains, 3= pasture grasses, 4= fruit trees, 5= Pinyon-Juniper, 6= deciduous shrubs, 7= annual grasses and herbs, 8= perennial grasses, 9= evergreen shrubs, 10= deciduous trees, 11= alfalfa, 12= annual vegetables and grains, 13= pasture grasses, 14= Douglas Fir, 15= Ponderosa Pine, 16= deciduous shrubs, 17= Spruce-Fir, 18= perennial grasses, 19= annual grasses and herbs, 20= evergreen shrubs, 21= deciduous shrubs, 22 = cheatgrass (B. tectorum) **note 1**: 23- 44 used to simulate plant more than once; 45-.., third...etc.. **note 2** : -1 to -22 used for user plant design; cannot use corresponding positive number in simulation -5, 5, etc..

13.0. APPENDIX D

GLOSSARY OF VARIABLES AND PARAMETERS THAT CAN BE SELECTED FOR OUTPUT IN BIOPLT SUBROUTINE FROM BIOTRAN.2 OPERATIONS

This compilation is not an exhaustive list of variables present in **BIOTRAN.2** subroutines, but only those which are in COMMON statements throughout the code that are present in the variable subroutine **BIOPLT**. However, most of the arrayed variables are present. There are a considerable number of non-arrayed variables not presented in this listing; most of them are temporary variables that are represented by arrayed variables presented below. Some of the parameters listed below are constant input parameters that do not change during the simulation, but some are variables entered as initial conditions that do change with simulation time. Because this listing is used for purposes other than obtaining plotting information, it has been presented in alphabetical order, however, the user may choose to regroup this listing to fit a specific need. Finally, the variables listed are totally available only when all subroutines are exercised, and care should be taken by the user to select variable outputs that are simulated by the subroutines accessed. For example, any variables pertaining to tree stand information would be available only if the **FORCUT**(9) and/or the **FORMAN**(10) subroutine was accessed in the simulation (see below) Also, contaminant concentrations are given as per M2, per gdwt..etc. to allow a number of units such as pCi, ug, gdwt,.. etc., depending on the contaminant, to be used for the mass or radiation unit.

The array elements: **I1,I2,I3**.., are integers which are set by user input specifications. The symbols or integers within the parentheses of the variables to be presented below correspond to the parameter statement limits set within the code, but whose size can be reset if necessary; however, the symbol(s) used to define the parameter of interest must remain the same. Below is given the symbol(s) which must be entered as plotting information (the integer number enclosed in parenthesis is the current parameter setting size(s) used in **BIOTRAN.2**):

i = number of plants(7)
ks = number of soil horizons(10)
kt = number of tree diameter classifications(20)
kw = number of water layers in a lake(10)
ka = number of cumulative exposure years for plants(200)
kr = number of age classifications in years for animals and humans(86), see (ke) below
m = number of contaminants that can be simulated(10)
n = number of regions that can be simulated(8)
nq = number of plant parts/tissues used in dosimetry(36)
ko = number of animals plus male and female humans(10)
ke = maximum number of age classifications for animals(25)

When any of the symbols listed above(or for integers in the array as shown below) are used as input, they must be entered in lower case with single quotes..ex. **'n' 'ko' '200'**....etc. Integers utilized in the same manner as above will be described with specific variables shown below that contain integers. Likewise, when variable identifiers are entered as input, they must be in upper case and surrounded by single quotes..ex. **'B1' 'TPOP' 'WTBN'** ..etc. More information of this type is presented in the input instructions of **BIOTRAN.2** in subroutine **BIOPLT.**

There are several variable attributes assigned to each variable. The first attribute relates to the type of variable, and the second to the subroutines in the code where the variable is derived and/or processed. The former are defined by letter designation, whereas subroutines locators are identified by an integer number as shown below:

Variable types: C=constant, D=dynamic, I=input, T=temporary

Subroutines: **1=MAIN, 2=AIRAC, 3=ANIMAL, 4=AQUAT, 5=AQUIFER, 6=BIOPLT, 7=CLIMAT, 8=EROSON, 9=FORCUT, 10=FORMAN, 11=GEOFLX, 12=HUMTRN, 13=PLTGRO, 14=RUNOFF, 15=UPTKE, 16=WATFLX, 17=ZOOGRO, 18=ARRAYS, 19=INVERT, 20=EFFECTS, 21=TRUPLT, 22=TPLOTR**

For each variable shown below is a specific variable attribute described above, and a brief description together with appropriate units

ATTRIBUTE/VARIABLE **DESCRIPTION**

D,3,12
A(kr,ko,n)- population fraction, I1=age, I2=organism, I3=region

D,15
A1(i,m,n)- air concentration from resuspension., per M3, I1=plant, I2=contaminant, I3=region

D,15
A1AV(i,m,n)- mean annual air concentration from resuspension, per M3, I1=plant, I2=contaminant, I3=region

D,15
A1HUM(m,n)- air concentration due to resuspended dead organic matter, per M3, I1=contaminant, I2=region

D,12
A1TH(m,n)- air concentration from resuspension inhaled by humans, per M3, I1=contaminant, I2=region

D,15
A1TT(m,n)- areally weighted mean concentration from resuspension from all plants in a given region, per M3, I1=contaminant, I2=region

D,15
A1TG(m,n)- areally weighted mean gaseous concentration from all plants, in a given region, per M3, I1=contaminant, I2=region

D,15
A1Z(m,n)- contaminant resuspension concentration from bare soil, per M3, I1=contaminant, I2=region

D,15
A1ZS(m,n)- contaminant saltation-creep-rainsplash concentration from bare soil, per M3, I1=contaminant, I2=region

D,15
A2(i,m,n)- contaminant concentration on leaf surface from resuspension events, per M2, I1=plant, I2=contaminant, I3=region

D,15
A22(m,n)- total contaminant concentration on leaf surface from all resuspendable sources, per M2, I1=contaminant, I2=region

D,15

A23(m,n)- total contaminant concentration on leaf surface from all saltation-creep-rainsplash sources, per M2, I1=contaminant, I2=region

D,15
A2HUM(m,n)-air concentration due to saltation-creep-rainsplash of dead organic matter, per M3, I1=contaminant, I2=region

D,15
A2TT(m,n)-areally weighted mean external concentration from resuspension on live standing biomass for all plants in a given region, per M2, I1=contaminant, I2=region

D,15
A2X(i,m,n)- same as variable (a2) except that sources other than re-entrainment of resuspendable surface deposits included, per M2, I1=plant, I2=contaminant, I3=region

D,15
A3(i,m,n)-contaminant concentration on leaf surface, from saltation-creep-rainsplash events, per M2, I1=plant,I2=contaminant, I3=region

D,15
A3AC(i,m,n)- integral average (with height) contaminant concentration in and on leaves, per gdwt, I1=plant, I2=contaminant, I3=region

D,15
A3AV(i,m,n)- integral average (with height, minimum of 7 cm) contaminant concentration on leaf surface, from saltation-creep-rainsplash events, per M2, I1=plant, I2=contaminant, I3=region

D,15
A3C(i,m,n)- total contaminant concentration in and on leaves, per gdwt, I1=plant, I2=contaminant, I3=region

D,15
A3TT(m,n)-areally weighted mean external concentration fromsaltation-creep-rainsplash events on live standing biomass for all plants in a given region, per M2, I1=contaminant, I2=region

C,I,3,12
AA2(ko)- fraction of occupied area utilized by animal/human, I1=animal/human

D,1
AAA(i,m,n)- contaminant concentration in air as resuspendable particulate matter, per M3, I1=plant, I2=contaminant, I3=region

D,10
AATB(i,kt,n)- mean age of tree, years, I1=plant, I2=diameter, I3=region

D,20
ABSCAN(2,ko)- total number of cancers predicted from absolute risk during the simulation period model, I1=sex,I2=animal/human

C,I,20
ABSR(ko)- selector for relative or absolute risk models, I1=animal/human

D,4

ABSRAD(kw,n)- daily insolation available for photosynthesis for aquatic plants, ly/day, I1=water layer, I2=region

D,19
ABSW(120,ko)- soil moisture diapause development gate, days, I1= diapause egg age, I2=invertebrate

D,13
AC- root respiration factor for each plant simulated, fraction

C,I,13
ACRES(i,n)- specified area for vegetation, acres, I1=plant, I2=region

D,10
ACTAV(i,m,n)- tree stand mean contaminant concentration, per gdwt, I1=plant, I2=contaminant, I3=region

C,I,3,12
ACTI(ko)- activity level for animal/human, number, I1=animal/human

D,10
ACTS(m,n)- internal contaminant concentration in trees, per M^2/year, I1=contaminant, I2=region

D,10
ACTSM(m,n)- tree stand summation of contaminant activity used to determine variable (actav), per gdwt, I1=contaminant, I2=region

C,I,9
ACUT(i,2,n)- first diameter interval scheduled for total or partial removal, inches.., I1=plant, I2=index (1=lower,2=upper), I3=region

D,16
ADDWAT- amount of water added to soil, mm/day, from irrigation

D,19
ADIA(120,ko)- invertebrate diapause development gate, days, for cold treatment, I1=egg age(days), I2=invertebrate

D,19
ADLT(2,ko)- biomass density of adult invertebrate, gdwt/M^2, I1=sex(male=1,female=2), I2=invertebrate

C,I,13
ADSUC(n)- sucession gate (0,1) for advanced succession simulation,I1=region

D,15
AFR(i,m,n)- total resuspendable contaminant on fruit surface, per M2, I1=plant, I2=contaminant, I3=region

D,15
AFRTT(m,n)- mean resuspendable contaminant concentration, per M2, on fruit(seed) surface, I1=contaminant, I2=region

D,15

AFS(i,m,n)- total saltation-creep-rainsplash contaminant on fruit surface, per M2, I1=plant, I2=contaminant, I3=region

D,15
AFSAC(i,m,n)- integral average saltation-creep-rainsplash contaminant, per gdwt, in and on fruit(seed), I1=plant, I2=contaminant, I3=region

D,15
AFSAV(i,m,n)- integral average saltation-creep-rainsplash contaminant on fruit surface, per M2, I1=plant, I2=contaminant, I3=region

D,15
AFSC(i,m,n)- total contaminant concentration, per gdwt, in and on fruits(seeds), I1=plant, I2=contaminant,I3=region

D,15
AFSTT(m,n)- mean saltation-creep-rainsplash contaminant concentration, per M2, on fruit(seed surface, I1=contaminant, I2=region

D,10
AGAVE- geometric mean age of tree stand

D,10
AGE(i,n)- tree stand age, years, I1=plant, I2=region

C,I,3,12
AGEA(ko)- maximum age of organism, years, I1=animal/human

C,3
AGR(ko,n)- area, M2, of first food preference by an animal, I1=animal, I2=region

D,10
AGTB(i,kt,n)- age of stand when tree diameter class is reached, years, I1=plant, I2=diameter class, I3=region

C,3,12
AGZ(ko)- total area, acres, utilized by organism, I1=animal/human

D,19
AHA(ko)- egg hatching index number, I1=invertebrate

C,I,13
AHRV(i,n)- number of harvesting events, I1=plant, I2=region

D,1-17,19
AI1- year number of simulation

D,11
AICE(30)- ice depth, mm, in channel, I1=channel

C,I,13
AICHE(n)- niche competition gate(0,1), I1=region

D,3,12

AII- animal age group, years

D,1-17,19
AIII- julian day number

C,I,4
AIOPT(5,n)- optimum light insolation, ly/day, for photosynthesis in specific phyotplankton, I1=phytoplankton species, I2=region

D,2
AIRCA(m,n)- resuspendable contaminant concentration in air, per M^3, I1=contaminant, I2=region

D,2
AIRCG(m,n)- gaseous contaminant concentration in air, per M^3, I1=contaminant, I2=region
C,I,2
AIRX(n)- regional gate(0,1) for AIRAC subroutine output

C,4
AKC(5,n)- minimum phytoplankton cell biomass, ug, I1=phytoplankton species, I2=region

C,I,11
AKOND(30)- hydraulic conductivity, inches/hr, I1=region/channel

C,4
AKQNT(5,n)- minimum nitrate content, ug, for specific phytoplankton, I1=phyotplankton species, I2=region

C,4
AKQPH(5,n)- minimum phosphate content, ug, for specific phytoplankton, I1=phytoplankton species, I2=region

C,4
AKQSI(5,n)- minimum silicate content, ug, for specific phytoplankton, I1=phytoplankton species, I2=region

C,4
AKSNT(5,n)- Michaelis-Menten half-saturation concentration constant for nitrate, ug/l, I1=phytoplankton species, I2=region

C,I,11
AKSOIL(n)- estimate of K in MUSLE soil erosion equation,
I1=region

C,4
AKSPH(5,n)- Michaelis-Menten half-saturation concentration constant for phosphate, ug/l, I1=phytoplankton species, I2=region

C,4
AKSSI(5,n)- Michaelis-Menten half-saturation concentration constant for silicate, ug/l, I1=phytoplankton species, I2=region

D,T,3
AL4(m)- contaminant transport, per day, from N-P, and T-B of lung to blood via the gut, I1=contaminant

D,T,3
AL5(m)- contaminant transport, per day, from P of lung to blood, I1=contaminant

C,I,7
ALATD(n)- regional latitude, deg., I1=region

C,I,7
ALPSR(n)- regional lapse rate, deg. F./ft, I1=region

C,I,11
ALSOIL(n)- estimate of L in MUSLE soil erosion equation, I1=region

C,I,3,12
ALSP(ko)- mean life-span of organism, years, I1=animal

C,I,20
ALTNT(6,8,ko)- latent period for expression of cancer, years, for up to 8 age groups, I1= cancer type (1-6),I2=cancer age group(1-8), I3=animal/human

C,I,20
ALTYR(6,8,ko)- upper age, years, for which a new latent period starts, I1=cancer type(1-6), I2=cancer age group(1-8), I3=animal/human

D,3,12
AMW(ke,ko)- simulated metabolic weight, kg, I1=age(years), I2=animal

C,3
AMX(ke,ko)- expected metabolic weight, kg, I1=age(years), I2=animal

C,7
ANDAY(n)- number of days in each month of the year, I1=region

D,4
ANITR(kw,n)- nitrate nitrogen concentration, ug/l, for a specific water layer, I1=water layer, I2=region

D,7
ANTAV(n)- mean nightly temperature, deg. C., for a given day, I1=region

D,T,3
ANTRO(ko)- nitrogen fraction in consumed food mix, I1=animal

D,T,3
ANTROS(ko)- nitrogen fraction in supplementary feed, I1=animal

C,I,1
APL(n)- total number of plants simulated, I1=region

C,11
APLX(n)- number of non-irrigated plants simulated, I1=region

C,I,5
AQUGAT- aquifer gate(0,1)

C,I,13

ARIG(i,n)- number of irrigation applications, I1=plant, I2=region

D,15
AS(i,m,n)- saltation-creep-rainsplash contaminant concentration in air, per M^3, I1=plant, I2=contaminant, I3=region

D,15
ASAV(i,m,n)- contaminant mean annual air concentration from saltation-creep-rainsplash events, per M3, I1=plant, I2=contaminant, I3=region

C,13
ASTP(i,n)- evergreen/deciduous gate(0,1), I1=plant, I2=region

C,I,16
ATNO(m)- atomic number or contaminant identification, I1=contaminant

D,13
ATTT(i,n)- counter, days, for perennial plants, I1=plant, I2=region

D,5
ATX(30)- crossectional area, ft2, of distal end adjusted for estimated height of aquifer(htx), I1=channel

I,D,3
AUTZD(ko,7,n)- fraction of daily consumption due to prey, I1=predator, I2=prey, I3=region

C,I,13
AWATA(i,36,n)- water applications, inches, I1=plant, I2=irrigation day, I3=region

C,I,16
AWT(m)- molecular/atomic weight, grams, I1=contaminant

D,13
B1(i,n)-above ground biomass, gdwt/M2, I1=plant, I2=region

D,15
B2(i,m,n)- total resuspendable contaminant on bole(stem) surface, per M2, I1=plant, I2=contaminant, I3=region

D,15
B3(i,m,n)- total saltation-creep-rainsplash contaminant on bole(stem), per M2, I1=plant, I2=contaminant, I3=region

D,15
B3AC(i,m,n)- integral average(with height), contaminant concentration in plant bole(stem), per gdwt, from both internal and external sources, I1=plant, I2=contaminant, I3=region

D,15
B3AV(i,m,n)- integral average(with height) of saltation-creep-rainplash contamination on the surface of live boles(stems), per M2, I1=plant, I2=contaminant, I3=region

D,15
B3C(i,m,n)- total contaminant concentration, per gdwt, for plant bole(stem) from all sources, I1=plant, I2=contaminant, I3=region

D,13
B5(i,n)- photosynthate production, gdwt/M2/day, I1=plant, I2=region

D,13
B7(i,n)- photsynthate flow to roots, gdwt/M2/day, I1=plant, I2=region

D,10
BAR(kt,n)- tree basal area, M2, I1=diameter class, I2=region

D,10
BARSQ(kt,n)- logarithmic bole diameter sum of squares, I1=diameter, I2=region

D,10
BATR- stand mean tree height, ft, using geometric mean

D,15
BB2(i,m,n)- total resuspendable contaminant on live branch surfaces, per M2, I1=plant, I2=contaminant, I3=region

D,15
BB3(i,m,n)- total saltation-creep-rainsplash contaminant on live branch surfaces, per M2, I1=palnt, I2=contaminant, I3=region

D,15
BB3AC(i,m,n)- integral average contaminant on live branches, per gdwt, from all sources, I1=plant, I2=contaminant, I3=region

D,15
BB3AV(i,m,n)- integral average contaminant on live branches, per M^2, I1=plant, I2=contaminant, I3=region

D,15
BB3C(i,m,n)- total contaminant concentration, per gdwt, on live branches from all sources, I1=plant, I2=contaminant, I3=region

C,I,5
BBCH(30)- width, ft, of bottom of distal end of channel aquifer,
I1=channel

D,10
BBP(i,n)- tree stand volume, ft3, I1=plant, I2=region

C,I,5
BCH(30)- width, ft, of bottom of proximal end of channel aquifer, I1=channel

C,13
BCP(i,n)- photosynthate fraction, per day, I1=plant, I2=region

C,I,9
BCUT(i,2,n)- second diameter interval scheduled for total or partial removal, inches.., I1=plant, I2=index (1=lower,2=upper), I3=region

D,19
BDDT(120,ko)- accumulator, deg. C.-day, above minimum developmental temperature(dvtmp), I1=egg age(days), I2=invertebrate

C,I,16
BDRK(n)- impervious fraction covering ground surface, I1=region

D,3,12
BIRTHS(ko,2)- number of offspring added to a population, I1=animal/human, I2= male or female(1,2)
C,I,3,12
BLTBN1(ko,m)- transport fraction, per day, from blood to bone1, I1=animal/human, I2=contaminant

C,I,3,12
BLTBN2(ko,m)- transport fraction, per day, from blood to bone2, I1=animal/human, I2=contaminant

C,I,19
BLTGT(ko,m)- transport fraction, per day, from blood to gut, I1=invertebrate, I2=contaminant

C,I,3,12
BLTKD1(ko,m)- transport fraction, per day, from blood to kidney1, I1=animal/human, I2=contaminant

C,I,3,12
BLTKD2(ko,m)- transport fraction, per day, from blood to kidney2, I1=animal/human, I2=contaminant

C,I,3,12
BLTLU(ko,m)- transport fraction, per day, from blood to lung, I1=animal/human, I2=contaminant

C,I,3,12
BLTLV(ko,m)- transport fraction, per day, from blood to liver, I1=animal/human, I2=contaminant

C,I,3,12
BLTMK(ko,m)- transport fraction, per day, from blood to milk, I1=animal/human, I2=contaminant

C,I,3,12
BLTMS(ko,m)- transport fraction, per day, from blood to muscle, I1=animal/human, I2=contaminant

C,I,3,12
BLTRS(ko,m)- transport fraction, per day, from blood to residual organs and tissues, I1=animal/human, I2=contaminant

C,I,3,12
BLTUR(ko,m)- transport fraction, per day, from blood to urine, I1=animal/human, I2=contaminant

D,10
BMN(i,n)- logarithmic mean diameter of stand, inches, I1=plant, I2=region

C,I,3,12
BNTBL1(ko,m)- transport fraction, per day, from bone1 to blood, I1=animal/human I2=contaminant

C,I,3,12
BNTBL2(ko,m)- transport fraction, per day, from bone2 to blood, I1=animal/human, I2=contaminant

C,D,3,12

BRRT(8,ko,2)- birth rate, per year/female, for male and female offspring, I1=female age class(8 total), I2=animal/human, I3=sex(1=male, 2=female)

C,3,12
BRTY(8,ko)- reproductive female classification based on age, I1=female age class(8 total), I2=animal/human

D,9
BSD- tree stand bole standard deviation, inches

D,13
BSMAX(i,n)- maximum simulated biomass for year, gdwt/M^2, I1=plant, I2=region

C,I,11
C1HR(30)- region/channel flow duration parameter1, I1=channel/region

D,T,3
C2- total mass, gdwt/day, of plant food consumed by animal

C,I,11
C2HR(30)- region/channel flow duration parameter2, I1=region/channel

D,T,12
C3(m)- total contaminant ingested, per day, by human of a given age, I1=contaminant

D,3
C31(ko,m,n)- total contaminant ingested, per day, by animal of a given age, I1=animal, I2=contaminant, I3=region

D,3
C31PLT(kr,ko,m)- nuclide intake from plants, per day, by animal from food ingestion, I1=age,I2=animal, I3=contaminant

D,3
C31TT(ko,m)- total nuclide intake, per day, by animal from food ingestion, I1=animal, I2=contaminant

D,T,3
C4- area grazed, M2/day, by one of a given age animal for range food

D,T,3
C5- soil consumption gdwt/day, by a grazing animal of a given age, and for particular range conditions

D,T,3,12
C56(m)- total contaminant intake, per day, from ingested water by an animal or human of a specific age group, I1=contaminant

C,I,11
C5HR(30)- region/channel peak flow parameter, I1=region/channel

C,I,10
C6(i,n)- bole(inches) to crown diameter(feet) regression coefficient, I1=plant, I2=region

D,T,3

C66(m)- total contaminant transfered to blood, per day, from soil ingestion by animal of a given age, I1=contaminant

D,10
C7- total tree stand volume,ft3

D,10
C8(i,n)-tree stand competition factor, I1=plant, I2=region

C,I,10
C9(i,n)- specific gravity of dry wood, I1=plant, I2=region

D,T,3,12
C99(m)- total contaminant transferred from gut to blood, per day, for an animal or human of a given age, I1=contaminant

D,4
CALATM- heat tranfer, deg. C., from ambient air to top 1 cm of top water layer

D,4
CALRA- net energy available,gram-calories/M2/day, after adjustment for scatter losses and water evaporation

D,4
CALRI- energy reaching water surface, gram-calories/M2, less that backscattered from water layers
D,4
CALSUB- heat transfer, deg. C., from sediment surface to bottom water layer(assumes 1 cm depth into substrate)

D,T,3,12
CAR(m)- total contaminant enter gut, per day, from inhalation deposition for an animal or human of a given age, I1=contaminant

C,3,12
CBBNZ(100)- bone burden, ug, I1=nuclide

C,3,12
CBKDZ(100)- kidney burden, ug, I1=nuclide

C,3,12
CBLVZ(100)- liver burden, ug, I1=nuclide

C,3,12
CBWBZ(100)- whole body burden, ug, I1=nuclide

C,I,9
CCUT(i,2,n)- third diameter interval scheduled for total or partial removal, inches.., I1=plant, I2=index (1=lower,2=upper), I3=region

D,13
CCVV(i,n)- crown cover, fraction, I1=plant, I2=region

C,16

CDRP(ks,n)- reciprocal of total potential available moisture, per mm, for plant transpiration, I1=soil layer, I2=region

D,4
CELCT(kw,n)- total phytoplankton biomass, ug, per specified volume(liters), I1=water layer, I2=region

C,4
CELKI(2,n)- Michaelis-Menten half-saturation constant for filtration, ml/day, I1=zooplankton type, I2=region

D,4
CELLS(5,kw,n)- number of phytoplankton cells, per 250 l, I1= phytoplankton species, I2=water layer, I3=region

D,13
CFAC(i,n)- niche competition factor, fraction, I1=plant, I2=region

D,14
CFSPD(30)- runoff peak flow, cfs, I1=region/channel

D,12
CH20B(kr,2)- weight of carbohydrate, kg, in human, I1=age(years), I2=sex(1=male, 2=female)

C,I,11
CHLAT(30,20)- specific laterals(regions) adjacent to channel I1, I1=channel, I2=region adjacent to channel i1

C,I,11
CHNAQ(30)- aquifer gate (0,i1) for presence of aquifer below channel, I1=region

C,I,11
CHNE(30,20)- specfic channels entering channel i1, I1=channel, I2=channels entering channel i1

C,I,11
CHNNU- number of channels in watershed

D,4
CHO(5,kw,n)- phytoplankton cell biomass, ug, I1=phyotplankton species, I2=wate layer, I3=region

C,I,16
CLAY(i,ks,n)- clay fraction in soil, I1=plant, I2=soil layer, I3=region

C,I,7
CLM(n)- region output selector gate(0,1) for climate, I1=region

C,4
CMAX(5,n)- maximum phytoplankton cell biomass, ug,
I1=phytoplankton species, I2=region

C,I,20
CMORTA(6,8,2,ko)- specific cancer rate, per 100000, for use with the linear model, I1=cancer type(1-6), I2=cancer age group(1-8), I3=sex(1=male,2=female), I4=animal/human

C,I,20

CMORTB(6,8,2,ko)- same as (cmortb) except for use with dose linear-dose squared model: a*d+b*d**2

C,I,20
CMORTY(6,8,2,ko)- upper age, years, for which a new cancer rate starts, I1=cancer type(1-6), I2=cancer age group(1-8), I3=sex(1=male,2=female), I4=animal/human (for cmorta and cmortb)

C,I,20
CNATR(6,8,2,ko)- natural cancer rate(fatal), per 100000, I1=cancer type(1-6), I2=cancer age group(1-18), I3=sex(1=male, 2=female), I3=animal/human

C,I,20
CNATY(6,8,2,ko)- upper age, years, for which a new cancer rate starts, I1=cancer type(1-6), I2=cancer age group(1-18), I3=sex(1=male,2=female), I4=animal/human

D,3,12
CNBLO(kr,ko,m)- blood contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNBOD(kr,ko,m)- whole body contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNBON(kr,ko,m)- bone contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNFAT(kr,ko,m)- fat contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNFEC(kr,ko,m)- contaminant concentration in feces, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNGUT(kr,ko,m)- contaminant concentration in gut, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,17
CNIT(i,n)- percent nitrogen in consumed food from range, I1=plant, I2=region

D,3,12
CNKID(kr,ko,m)- kidney contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNLIV(kr,ko,m)- liver contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNLUG(kr,ko,m)- lung clearance contribution to ingestion, per kgfwt of body weight, I1=age, I2=animal/human, I3=contaminant

D,3,12

CNLUN(kr,ko,m)- lung surface contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNMLK(kr,ko,m)- contaminant clearance from milk, per kgfwt of body weight, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNMUS(kr,ko,m)- muscle contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3
CNPEL(kr,ko,m)- concentration of contaminant on pelt, per kgfwt of body weight, I1=age, I2=animal, I3=contaminant

D,3,12
CNPLN(kr,ko,m)- contaminant contribution from plant food, per kgfwt of body weight, I1=age, I2=animal/human, I3=contaminant

D,3,12
CNRES(kr,ko,m)- residual soft tissue contaminant concentration, per kgfwt, I1=age, I2=animal/human, I3=contaminant

D,3
CNSOL(kr,ko,m)- contaminant ingestion from soil, per kgfwt of body weight, I1=age, I2=animal, I3=contaminant

D,3,12
CNURN(kr,ko,m)- mean urine contaminant concentration, per liter, I1=age, I2=animal, I3=contaminant

D,3,12
CNURNC(kr,ko,m)- urine contaminant clearance, per kgfwt of body weight, I1=age, I2=animal, I3=contaminant

D,19
COLDAY(ko)- degree-day accumulator for cold treatment of diapause eggs, I1=invertebrate

D,10
COV(kt,n)- tree crown cover, M^2, I1=diameter class, I2=region

D,10
COVAV- tree stand fractional crown cover

D,10
COVSQ(kt,n)- tree crown cover sum of squares, I1=diameter class, I2=region

C,I,11
CSOIL(n)- estimate of C in the MUSLE soil erosion equation, I1=region
D,2
CT(m,n)- resuspendable contaminant air concentration, per M3, obtained from chronic function generator, I1=contaminant, I2=region

D,2

CTG(m,n)- gaseous contaminant air concentration, per M3, obtained from chronic function generator, I1=contaminant, I2=region

C,I,13
CUTA(i,8,n)- plant harvest dates, julian day, I1=plant, I2=harvest, I3=region

C,I,13
CUTMAX(i,n)- last harvest date, julian day, I1=plant, I2=region

D,10
CVMN- mean tree stand crown cover, M2

D,10
CVSD- crown cover standard deviation, M2

D,10
CVTR- tree stand basal area, ft2

D,15
CVTT(n)- mean crown cover, fraction, for all plants in region, I1=region

D,13
CXFAC(i,n)- initial value of (cfac)

D,16
CZ(i,ks,n)- fractional hydraulic conductivity, mm/mm/day, I1=plant, I2=soil layer, I3=region

D,3
D(i,n)- fraction of dry biomass in food for consumption from range, I1=plant, I2=region

D,13
D1(i,n)- dead standing above ground biomass, gdwt/M^2, I1=plant, I2=region

D,13
D1GB(i,n)- dead standing bole(stem) biomass, gdwt/M^2, I1=plant, I2=region

D,13
D1GBB(i,n)- dead standing branch biomass, gdwt/M^2, I1=plant,I2=region

D,13
D1GT(i,n)- dead standing branch and bole biomass, gdwt/M^2, I1=plant,I2=region

D,13
D3(i,n)- dead standing above ground biomass production, gdwt/M2/day, I1=plant, I2=region

C,I,2
DAIC(16,m,n)- days of contaminant release for acute events, I1=year, I2=contaminant, I3=region

C,I,9
DATCT(i,10,n)- year numbers for tree stand management(fire), I1=plant, I2=management, I3=region

D,7
DAYFRC- fraction of 24 hour day that has sunlight for site for a given Julian day

D,T,12
DBURD(m)- total contaminant, per day, ingested from soil intake for an animal of a certain age

D,8
DCHAN(30)- depth of runoff channel, feet, I1=channel

D,7
DECLIN- daily declination angle, deg., for site

D,19
DEGDAY(ko)- invertebrate deg-day accumulator, deg. C.-day, for egg hatching to occur, I1=invertebrate

D,19
DEGG(120,ko)- biomass density, gdwt/M2, of diapause eggs, I1=age (days), I2=invertebrate

D,19
DEGGS- biomass density, gdwt/M2, of all diapause egg age cohorts

D,4
DENDIF(kw,n)- density difference, g/cm3, between adjacent water layers(top-bottom), I1=water layer, I2=region

D,4
DENSE(kw,n)- density of water layer, g/cm3, I1=water layer, I2=region

C,I,4
DEPTHI(n)- thickness of water layers, meters, I1=region

C,I,19
DIADAY(ko)- julian day at which 100% production of diapause eggs occurs, I1=invertebrate

D,T,3
DIGE(ko)- digestibility of food mix, fraction, I1=animal

C,I,13
DIRA(i,36,n)- irrigation schedules, I1=plant, I2=irrigation date, I3=region

C,I,13
DIRMAX(i,n)- last irrigation date, julian day, I1=plant, I2=region

D,T,15
DIRT- total soil, gdwt/M2, present on leaf surfaces due resuspendable and saltation-creep-rainsplash transport

C,I,16
DISINT(m)- fractional decay rate, per day, I1=contaminant

C,I,4
DIT(n)- number of water layers simulated in AQUAT, I1=region

C,4
DIVD(5,n)-maximum division rate, per day, of phytoplankton, I1=phytoplankton species, I2=region

D,7
DNN(n)- day accumulator in a given year, days, I1=region

D,T,15
DOSE(36)- dose, rads, to different plant parts, I1=plant part (may be mean(1-6) or integral average(7-11) dose)

C,7
DRMM(n)- average annual precipitation for a given month, mm,
I1=region

D,T,12
DRNKK- drinking water intake, l/day, by humans of different age groups

C,I,11
DSF(7,30)- region/channel sediment fraction, I1= sediment diameter class, I2=region/channel

D,11
DSM(7,30)- eroded sediment, lbs/day, I1=soil diameter class, I2=region/channel

D,11
DSMA(7,30)- amount of contaminant in soil(assume 20 mm thickness and 1.5 bulk density), I1=soil diameter class, I2=region/channel(assume 200 mm thickness)

D,11
DSMS(7,30)- weight, lbs, of soil(assume 20 mm thickness and bulk density of 1.5) I1=soil diameter class, I2=region/channel(assume 200 mm thickness)

D,11
DSMT(30)- total sediment, lbs/day eroded, I1=region/channel

C,I,16
DSRT(m,n)- day of initial soil contaminantion, I1=contaminant, I2=region

C,I,11
DSZ(8,30)- region/channel sediment diameter, mm, I1=sediment diameter class, I2=region/channel (I2=8 used for median diameter size)

D,9
DTBR(i,kt,n) - number of trees, I1=plant, I2=diameter class, I3=region

D,11
DUHR(30)- runoff duration, hours, I1=region/channel

D,4
DV(5,kw,n)- phytoplankton cell division rate, per day, I1=phytoplankton species, water layer, I3=region

C,I,19
DVTMP(ko)- invertebrate minimum developmental temperature, deg. C.,I1=invertebrate

D,4
DVZ(2,kw,n)- division rate, per hour, for zooplankton, I1=zooplankton type, I2=water layer, I3=region

I,D,11

DZAS(7,30)- contaminant concentration, per gdwt, for soil, I1=soil diameter class, I2=region

D,7
E2(n)- potential evapotranspiration, mm/M2/day, I1=region

D,13
E5(i,n)- effect of temperature on photosynthesis, fraction, I1=plant, I2=region
D,16
EA(i,n)- cummulative evapotranspiration from all soil layers, mm/M2/day, I1=plant, I2=region

D,4
EAT(5,kw,2,n)- phytoplankton biomass consumption rate, ug/day, by zooplankton, I1=phytoplankton species, I2=water layer, I3=zooplankton type, I3=region

C,I,3,12
EATIN(ko)- twice the number of yearly feeding(type) intervals, I1=animal/human

C,I,3,12
EATREC(ko,32)- number of feeding sub-intervals for each year' feeding interval, I1=animal/human, I2=feeding interval

D,16
EAVAL(i,ks,n)- plant transpiration rate, mm/M2/day, I1=plant, I2=soil layer, I3=region

C,7
EFFP(n)- end of frost-free period, julian day, I1=region

C,I,15
EGEOP(ko)- plant tissue exposure geometry(0.5 or 1.0), I1=radionuclide

D,19
EGG(120,ko)- biomass density, gdwt/M2, of invertebrate regular eggs,I1=age(days), I2=invertebrate

C,I,19
EGGMRT(ko)- invertebrate egg mortality rate, fraction/day, I1=invertebrate

D,19
EGGS- total biomass density, gdwt/M2, of all regular egg age cohorts

D,19
EGLAY- fraction of female adult invertebrate converted to egg biomass, per day

C,I,8
ENTCHN(30)- number of channels entering channel i1, I1=channel

C,I,4
EPLD(n)- depth to epilimnion, M, I1=region

D,13
ETRANS(i,n)- fraction of photosynthate transferred to fruits, I1=plant, I2=region

D,16
EVAP(i,n)- evaporation rate, mm/M2/day, I1=plant, I2=region

C,I,3,12
EVB(ko,m)- bone tissue radiation deposition, Mev-Rem/dis, I1=animal/human, I2=radionuclide

C,I,3,12
EVE(ko,m)- endosteal tissue radiation deposition, Mev-Rem/dis, I1=animal/human, I2=radionuclide

C,I,3,12
EVS(ko,m)- soft tissue radiation deposition, Mev-Rem/dis, I1=animal/human, I2=radionuclide

C,I,15
EVSP(m)- plant soft tissue dose factor, MeV-Rem/dis-Rad, I1=radionuclide

D,T,4
EVTEMP- ambient air temperature based on 24 hour day, deg. C.

C,I,15
EXCP(m)- exposure constant for plants, I1=radionuclide

D,4
EXIP(5,n)- phytoplankton cell specific light extinction coefficient, per M, I1=phytoplanton species, I2=region

C,I,4
EXTC(n)- light extinction coefficient of water layers excluding the presence of phytoplankton, per M, I1=region

D,4
EXTD- total extinction coefficient for water, per M, for water and phytoplankton combined
C,I,16
F1(i,ks,n)- soil field capacity, mm, in soil layer, I1=plant, I2=soil layer, I3=region

D,7
F7(n)- snow accumulation, mm, I1=region

D,3,12
FATB(kr,ko)- total body fat in organism, kg, I1=age(years), I2=animal/human

D,3
FDIN(ke,ko)- food intake, kgdwt/day, by organism, I1=age(yrs), I2=animal

C,I,19
FDTMP(ko)- minimum invertebrate feeding temperature, deg. C., I1=invertebrate
C,I,7
FFP(n)- frost-free period of site, days, I1=region

C,13
FLEAF(i,n)- initial fraction of above ground biomass assigned to leaves, I1=plant, I2=region

D,4
FLXDN(kw,n)- heat transport vertically downward from water layer, deg. C., I1=water layer, I2=region

D,4
FLXUP(kw,n)- heat transport vertically upward from water layer, deg. C., I1=water layer, I2=region

C,I,3,12
FMK(m)- fractional transport, per day, from blood to milk, I1=radionuclide

C,I,3,12
FODIN(ko)- twice the number of daily feeding intervals, I1=animal/human

C,I,10
FRM(n)- regional output gate(0,1) for FORMAN subroutine, I1=region

C,I,9
FRT(n)- regional output gate(0,1) for FORCUT subroutine, I1=region

C,I,3,12
FSUPF(ko)- fraction of non-meat food (feeds, grains..)obtained from study area, I1=animal/human

C,I,3,12
FTTBL(ko,m)- fractional transport, per day, from fat tissue to blood, I1=animal/human, I2=contaminant
D,3
G(i,n)- fraction of live(green or animal prey) biomass in food for consumption from range, I1=plant, I2=region

C,I,7
G1(n)- elevation, ft, a.m.s.l. of simulated site, I1=region

D,7
G4(n)- random numbers drawn from a normal distribution, I1=region

D,7
G5(n)- random numbers drawn from an even distribution, I1=region

D,10
GA(kt,n)- age difference between youngest and oldest trees in a given diameter class, years, I1=diameter class, I2=region

C,I,3
GASLOS(ko)- fractional gas losses from ingested food, I1=animal

D,12
GBURD(m)- N-P and T-B lung deposition, per day, in humans of different ages, I1=contaminant

D,12
GBURI(kr,2,m)- cummulative deposition of (gburd) in humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,13
GM2B(i,n)- live biomass of branches, gdwt/M2, I1=plant, I2=region

D,13
GM2R(i,n)- live biomass of root bole (stem), gdwt/M2, I1=plant, I2=region

D,13
GM2RB(i,n)- live biomass of root branches, gdwt/M2, I1=plant, I2=region

D,13
GM2T(i,n)- live biomass of bole(stem), gdwt/M2, I1=plant, I2=region

D,12
GMEAT- meat intake, gfwt/day, by humans of various age groups

D,12
GMILK- milk intake, ml/day, by humans of various age groups

D,T,3
GRNA(ko)- fraction of food mix that is live(green and/or animal prey), I1=animal

D,T,3
GRNAS(ko)- live(green and/or animal prey) fraction in supplementary feed, I1=animal

D,8
GSBD(7,30)- particle transport capacity, lbs/sec/ft, I1= particle diameter class(mm), I2=region/channel
D,8
GSBDTT(30)- sum of (gsbd), I1=region/channel

C,I,3,12
GTBL(ko,m)- fractional transport coefficient, per day, from gut to blood, I1=animal/human, I2=contaminant

D,17
GTDR(ko,n)- live (green) biomass fraction in forage for animal consumption, I1=animal/human, I2=region

C,I,16
H(i,ks,n)- soil holding capacity, mm, in soil layer, I1=plant, I2=soil layer, I3=region

D,T,13
H2- effect on soil surface temperature on litter decomposition

D,T,13
H3- effect of soil moisture content on litter decomposition

D,20
HASUM(2,ko)- cummulative fraction of different age groups in a population, I1=sex(1=male,2=female), I2=animal/human

D,19
HATCE(120,ko)- hatching deg-day change gate, deg. C.-day, difference from developmental temperature for regular eggs,I1=egg age(days), I2=invertebrate

D,19
HATCH(120,ko)- degree-day, deg. C.-day, gate above minimum temperature required for egg hatching, I1=egg age(days), I2=invertebrate

D,19
HATCHS- sum of all hatching egg gate age cohorts(hatch)

D,19

HATMP(120,ko)- same as (hattmp) below except it is applied to regular eggs, I1=egg age(days), I2=invertebrate

D,19
HATTMP(120,ko)- hatching temperature accumulator(1,2..n) above a minimum mean maximum daily temperature for diapause eggs, I1=diapause egg age(days), I2=invertebrate

D,16
HBAR(ks,n)- volumetric water content, mm, I1=soil layer, I2=region

C,I,5
HBT(30)- height of distal end of channel aquifer, ft, I1=channel

D,19
HEMLOS(ko,m,n)- transport of contaminant, per day, from hemolymph to the environment via the gut from nymphs, young adults and adult invertebrates combined, I1=invertebrate, I2=contaminant, I3=region

C,I,13
HF(i,n)- mean plant lifespan, years, I1=plant, I2=region

C,I,16
HFLIF(m)- contaminant half-life(persistence), I1=contaminant

D,10
HGT(kt,n)- mean tree height, ft, I1=diameter class, I2=region

D,10
HGTSQ(kt,n)- logarithmic height sum of squares, I1=diameter class,I2=region

D,T,12
HL4(m)- blood contaminant uptake, per day, from N-P and T-B portions of respiratory tract via the gut for humans, I1=contaminant

D,T,12
HL5(m)- blood contaminant uptake, per day, from P region of lung directly into lung for humans, I1=contaminant

D,T,17
HM8- total area, M2/day, grazed by a given animal population

D,17
HRVST(kr,2,m)- food harvest storage, kgdwt, for animal or human consumption, I1=animal/human, I2=plant, I3=region

D,10
HTAV(i,n)- logarithmic mean tree height, feet, I1=plant, I2=region

C,I,5
HTP(30)- height of proximal end of channel aquifer, ft, I1=channel

D,10
HTSD- tree height standard deviation, feet

D,5
HTX(30)- estimated aquifer height, ft, I1=channel

D,3,12
HUMUS(ko,m,n)- contaminant concentration, per M2, in animal/human humus, I1=animal/human, I2=contaminant, I3=region

D,13
ICA(i,n)- harvest counter, days, I1=plant, I2=region

D,19
IEGD(ko)- invertebrate diapause egg age index, days, I1=invertebrate

D,19
IEGG(ko)- invertebrate regular egg age index, days, I1=invertebrate

D,19
IEGH(ko)- invertebrate nymph stage age index, days, I1=invertebrate

C,I,1
INZ- contaminant index number

C,I,1
IPL- plant type index number

C,I,1
IPX- region index number

D,13
IRG(i,n)-irrigation counter, days, I1=plant, I2=region

C,1
IU- screen display gate(0,1) for input data

C,4
KDI- number of specified water layers

C,I,3,12
KDTBL1(ko,m)- fractional transport, per day, from kidneys to blood(1), I1=animal/human, I2=contaminant

C,I ,3,12
KDTBL2(ko,m)- fractional transport, per day, from kidneys to blood, I1=animal/human, I2=contaminant

C,I,7
L1(n)- lag period, days, to maximum insolation, I1=region

C,I,7
L2(n)- lag period, days, to maximum temperature, I1=region

C,I,7
L3(n)- temperature correction, deg. F., for mean daily temperature differences between reference and simulated site, I1=region

C,I,3,12
LVTBL(ko,m)- fractional transport, per day, from liver to blood, I1=animal/human, I2=contaminant

D,7
MMX(n)- day accumulator for a given month, days, I1=region

C,I,3,12
MSTBL(ko,m)- fractional transport, per day, I1=animal/human, I2=contaminant

C,I,3
NAN- number of animals simulated

C,7
NDAY(12)- number of days in each month, I1=month number(1-12)

C,12
NHM- total number of organisms simulated including humans

C,I,15
NPPTS- number of plant parts simulated for radiation dosimetry:
1=seeds, 2=leaves, 3=stems, 4=boles, 5=root stems, 6=root boles, 7=seeds, 8=leaves, 9=stems, 10=boles, 11=seeds(internal only), 7-8 are integral average doses

C,I,1
NU- gate(0,1) which determines whether input is from file input or directly from user

D,7
NX(24,n)- day number assigner for rain events within a given month, I1=rain event number, I2=region

D,19
ONDV(30,ko)- old nymph developmental index, I1=nymph age(days), I2=invertebrate

C,I,3,12
ONFOD(ko,32)- daily feeding intervals(2 julian days/interval), I1=animal/human, I2=daily feeding interval

D,19
ONYPH(30,ko)- biomass of old nymphs, g/M2, I1=nymph age(days), I2=invertebrate, I2=invertebrate

D,19
ONYPHS- total biomass, g/M2, of all age cohorts of young nymphs

D,20
ORGBRD(6,2,ko)- organ contaminant burden for different animals and humans of various ages, I1=cancer type (1=kidneys,2=bone,3=liver,4=lungs,5=not used,6=endosteal bone tissue), I2=sex(1=male,2=female), I3=animal/human

D,20
ORGDOS(6,2,ko)- cummulative organ doses, rems, for (orgbrd) contaminant burdens above (radionuclides only)

D,20
ORGRAD(6,2,ko)- cancer type selection gate(0,1) for choice of cancer types in the siumulation, I1=cancer type (1-6), I2=sex(1=male, 2=female), I3=animal/human

D,17
OTOT(ko)- total mass of food, kgdwt/day, consumed by animal population from range, I1=animal

D,17
OTOTA(ko)- total mass of food required, kgdwt/day, by animal population, I1=animal

D,17
OTOTB(ko)- total mass of food avialable for consumption, kgdwt/day,by animal population, I1=animal

D,17
OTOTBS(ko)- food available for consumption, kgdwt/day, from supplementary feed, I1=animal

D,17
OTOTS(ko)- total mass of food, kgdwt/day, consumed by animal population from non-range sources, I1=animal

D,17
OTTSM(ko)- food selection iteration counter(10 maximum), I1=animal

D,7
P(n)- daily precipitation event, mm, I1=region

D,16
P1(i,n)- cummulative moisture addition to top layer of soil, mm, from all sources excluding ponded water, I1=plant, I2=region

C,I,3,12
P5(ko)- fraction of meat consumed from region, I1=animal/human

D,17
PAV(ko)- mean precipitation, mm, for utilized area(may include more than 1 region with different utilization factors(rfz)), I1=animal/human

D,T,3,12
PBURD(m)- total contaminant deposition in lungs, per day, for animals and humans of different ages, I1=contaminant

D,12
PBURI(kr,2,m)- cummulative contaminant deposition(pburd) in lungs for humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,19
PDIAPE(120,ko)- biomass density of invertebrate post-diapause eggs, gdwt/M2, I1=age(days), I2=invertebrate

D,19
PDIAPS- total biomass density, gdwt/M2, of all invertebrate diapause egg age cohorts

C,I,13
PGR(n)- regional output selector gate(0,1) for PLTGRO, I1=region

C,I,16
PH(i,ks,n)- soil pH, I1=plant, I2=soil layer, I3=region

D,16
PHI(i,ks,n)- soil matrix potential, bars, I1=plant, I2=soil layer, I3=region

D,16
PHITT(3)- mean matrix potential, bars, for second soil layer, I1= 1-3

D,4
PHOSP(kw,n)- phosphate phosphorus concentration, ug/l, in a specific water layer, I1=water layer, I2=region

D,7
PHOTPR- photoperiod, hours, for site

D,4
PHYTO(5,kw,n)- total phytoplankton biomass, ug/250 l(or other specified volume), I1=phytoplankton species, I2=water layer, I3=region

C,I,3,12
PLCM1(ko)- total particulate deposition fraction for N-P region of lung, I1=animal/human

C,I,3,12
PLCM2(ko)- total particulate deposition fraction for T-B region of lung

C,I,3,12
PLCM3(ko)- total particulate deposition fraction for P region of lung, I1=animal/human

C,I,3,12
PLTBL(ko,m)- transport fraction, per day, from P region of lung to blood, I1=animal/human, I2=contaminant

C,I,3,12
PLTBLG(ko,m)- transport fraction, per day, from P region of lung to blood from gaseous contaminant, I1=animal/human, I2=contaminant

C,I,20
PLTEA(8,ko)- expression period, years, for cancer induction, I1=expression period(1-8), I2=animal/human (sex differences attributed to humans only)

C,I,20
PLTEAY(8,ko)- upper age limit, year, where a new plateau period must be specified specified, I1=expression period(1-8), I2=animal/human(sex differences attributed to humans only

I,D,3,12
PLTF(ko,8,32,n)- fraction of each plant type in diet, I1=animal/human, I2=plant type, I3=feed schedule, I4=region

C,I,3,12
PLTG1(ko,m)- transport fraction, per day, from N-P region of lung to gut, I1=animal/human, I2=contaminant

C,I,3,12
PLTG2(ko,m)- transport fraction, per day, from T-B region of lung to gut, I1=animal/human, I2=contaminant

I,D,13
PLTID(i,n)- plant identification number, I1=plant, I2=region

I,D,3,12
PLTIND(ko,32,n)- number of changed feeding schedules, I1=animal/human, I2=feeding schedule,I3=region

I,D,3,12
PLTRNK(ko,8,32,n)- range preference(1,2,3), and/or feed types(4,5,6,7), I1=animal/human, I2=number of plant type used, I3=feed schedule, I4=region

I,D,3,12
PLTUSD(ko,8,32,n)- plant types used as food, I1=animal/human, I2=number plant types used, feed schedule, I4=region

D,7
PM(12,n)- mean monthly precipitation, mm, for climograph generation, I1=month, I2=region

D,7
PMS(12,n)- cummulative monthly precipitation, mm, for site, I1=month, I2=region

D,7
PN(n)- cummulative daily precipitation for a given month, mm, I1=region

I,D,3,12
POP(ko,2)- total number in population, I1=animal/human,
I2=sex(1=male,2=female)

C,I,20
POPFR(ko)- population multiplication constant applied to animal and human populations to account for cancer production in the entire population, I1=animal/human

D,20
POPSUM(2,ko)- total population size, I1=sex(1=male,2=female), I2=animal/human

D,11
PRCH(n)- mean precipitation, mm/day, of watershed-receiving region, I1=region

C,I,7
PRCV(n)- precipitation coefficient of variation, I1=region

D,7
PREC(n)- water increment, mm/day, to top of first soil layer from all sources including ponding, I1=region

C,12
PROT1(kr,2)- expected human body protein weight, kg, I1=age(years), I2=sex(1=male,2=female)

D,12
PROTB(kr,2)- simulated human body protein weight, kg, I1= age(years), I2=sex(1=male,2=female)

C,I,13
PSEP(i,n)- C3/C4 plant gate(0,1), I1=plant, I2=region

C,I,11
PSOIL(n)- estimate of P in MUSLE soil erosion equation, I1=region

C,D,3,12
PUTFRC(ko,i,n)- diet fraction of plant in food mix, I1=animal/human, I2=plant, I3=region

C,D,3,12
PUTZD(ko,i,n)- food preference/type index, 1,2,.., for consumption purposes, I1=animal/human, I2=plant, I3=region

D,7
PX(12,n)- running mean yearly precipitation for site at end of simulation, mm, I1=month, I2=region

D,7
Q(40,n)- day/month multiplier for rain events, I1= number of rain events for month, I2=region

D,3,12
Q1(kr,ko,m)- total contaminant in N-P region of respiratory tract, I1= age(years), I2=animal/human, I3=contaminant

D,17
Q10(i,n)-total live and dead biomass, gdwt/M2, available for consumption from range, I1=plant, I2=region

D,3,12
Q2(kr,ko,m)- total contaminant in T-B region of lung, I1=age(years), I2=animal/human, I3=contaminant

D,3,12
Q3(kr,ko,m)- total contaminant in P region of lung, I1=age(years),I2=animal/human, I3=contaminant

D,7
Q8(n)- random number array for determination of precipitation events and temperature variations, I1=region

D,17
QFRAC(i,n)- fraction in diet of plants consumed, I1=plant preference, I2=region

C,4
QMNT(5,n)- maximum nitrate content, ug, for specific phytoplankton, I1=phytoplankton species, I2=region

C,4
QMPH(5,n)- maximum phosphate content, ug, for specific phytoplankton, I1=phytoplankton species, I2=region

C,4
QMSI(5,n)- maximum silicate content, ug, for specific phytoplankton, I1=phytoplankton species, I2=region

D,4
QNITRO(5,kw,n)- phytoplankton body content of nitrate, ug, I1=phytoplankton species, I2=water layer, I3=region

D,4
QPHOSP(5,kw,n)- phytoplankton body content of phosphate, ug, I1=phytoplankton species, I2=water layer, I3=region

D,8
QPKD(30)- runoff peak flow, in/hr, I1=region/channel

D,4
QSILCN(5,kw,n)- phytoplankton body content of silicate, ug, I1=phytoplankton species, I2=water layer, I3=region

D,17
QSUM(ko,n)- total biomass available as food, gdwt/M2, I1=plant animal/human, I2=region

C,I,7
R(n)- mean annual precipitation, inches, I1=region

D,13
R1(i,n)- root biomass, gdwt/M2, I1=plant, I2=region

D,13
R1MAX(i,n)- maximum observed root biomass for year, gdwt/M2, I1=plant, I2=region

C,I,9
RACUT(i,n)- fraction of timber remaining after cutting/fire event for first cutting interval, I1=plant, I2=region

D,7
RADA(n)- angle of latitude in radians, I1=region

D,T,7
RADANG- daily zenith angle, deg., for site

D,13
RANK(i,n)- plant gate (1,0) for consideration in succession, I1=plant, I2=region

C,11
RAREA(n)- regional area, acres, I1=region

D,15
RB3(i,m,n)- contamination on surface of live root bole(stem) derived from soil, per M2, I1=plant, I2=contaminant, I3=region

D,15
RB3C(i,m,n)- contamination on surface of live root bole(stem) bark derived from soil, per gdwt, I1=plant, I2=contaminant, I3=region

D,15
RBB3(i,ks,n)- contamination on surface of live root branches derived from soil, per M2, I1=plant, I2=soil layer, I3=region

D,15
RBB3C(i,m,n)- contamination on surface of live root branch bark derived from soil, per gdwt, I1=plant, I2=contaminant, I3=region

D,9
RBCUT(i,n)- fraction of timber remaining after cutting/fire event,for second diameter interval, I1=plant, I2=region

D,3,12
RBN(kr,ko,m)- dose to bone, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

D,3,12
RKD(kr,ko,m)- dose to kidney, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

D,3,12
RLU(kr,ko,m)- dose to lung, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

D,3,12
RLV(kr,ko,m)- dose to liver, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

D,3,12
RMS(kr,ko,m)- dose to muscle, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

D,3,12
RNDO(kr,ko,m)- dose to bone endoplastic reticulum, rems, I1=age(yrs), I2=animal/human, I3=radionuclide

C,I,9
RCCUT(i,n)- fraction of timber remaining after cutting/fire event
for third cutting interval, I1=plant, I2=region

C,I,11
RCHAN(30)- manning coefficient, dimensionless, I1=region/channel

C,I,16
RCODE(i,n)- number of soil layers, I1=plant, I2=region

D,13
RD1(i,n)- easily decomposed dead standing root biomass, gdwt/M2, I1=plant, I2=region

D,13
RD2(i,n)- resistant to decomposition dead standing root biomass, gdwt/M2, I1=plant, I2=region

D,13
RD3(i,n)- dead standing root biomass production, gdwt/M2/day, I1=plant, I2=region

D,11
RDCF(n)- daily runoff, cf, I1=region

D,T,4
REFRAC- angle of refraction from sunlight at water surface, deg.

C,I,11
REGENT(n)- number of regions contributing sediment to region i1, I1=region

C,I,11

REGX(n,10)- specific regions contributing sediment to region i1, I1=region, I2=region contributing sediment to region i1

D,20
RELCAN(2,ko)- total number of cancers during simulation period predicted from relative risk model, I1=sex(1=male, 2=female), I2=animal/human

C,I,7
RELV(n)- elevation, ft, a.m.s.l. of reference site, I1=region

D,4
RESP(5,n)- phytoplankton metabolic rate, ug-CH2O/day, I1=phytoplankton species, I2=region

C,13
RESPF(i,n)- plant live-to-dead biomass mortality rate 1, per day, I1=plant, I2=region

D,11
RFDCF(n)- subsurface return flow, cf/day, I1=region

D,11
RFDLX(n)- subsurface return flow, inches/day, I1=region

D,4
RFLCT- fraction of solar radiation reflected from surface of water

C,13
RFP(i,n)- root respiration coefficient 2, per day, I1=plant, I2=region

C,I,17
RFRAC(ko)- overall range plant utilization fraction, I1=animal

C,I,3,12
RFRZ(ko,n)- fraction of region utilized, I1=animal/human, I2=region

D,11
RFYCF(n)- subsurface return flow, cf/yr, I1=region

D,11
RFYLX(n)- subsurface return flow, inches/year, I1=region

D,15
RGAIN(m,n)- contaminant uptake rate, per M2/day, I1=contaminant, I2=region

C,I,13
RGC(i,n)- plant gate(1,2,3) for diameter class simulation, I1=plant, I2=region

C,I,13
RGCPLT(i,n)- plant output gate(0,1), I1=plant, I2=region

D,13
RH1(i,ks,n)- rapidly decomposing plant humus, gdwt/M2/day, I1=plant, I2=soil layer, I3=region

D,13
RH2(i,ks,n)- slowly decomposing plant humus, gdwt/M2/day, I1=plant, I2=soil layer, I3=region

D,9
RINGT(i,n)- mean tree ring width, inches, I1=plant, I2=region

C,I,13
RIRRC(i,n)- plant irrigation gate(0,1), I1=plant, I2=region

C,I,11
RNCHN(30)- number of lateral flows (regions) adjacent to channel i1, I1=channel

D,8
RNDLX(30)- daily runoff, inches, I1=region/channel

C,I
RNFRC(ko,32,n)- regional utilization fraction, I1=animal/human, I2=daily feeding schedule, I3=region

D,16
RNODX(i,n)- same as (rnofds), I1=plant, I2=region

D,16
RNOFDG(i,n)- moisture runoff, mm/M2/day, from bottom soil layer, I1=plant, I2=region

D,16
RNOFDM(i,m,n)- contaminant lost from bottom soil layer, per M2/day, I1=plant, I2=contaminant, I3=region

D,16
RNOFDS(i,n)- moisture runoff from soil surface, mm/M2/day, I1=plant, I2=region

D,16
RNOFYG(i,n)- moisture runoff, mm/M2/day, from bottom soil layer, I1=plant, I2=region

D,16
RNOFYM(i,m,n)- contaminant loss from bottom soil layer, per M2/year, I1=plant, I2=contaminant, I3=region

D,16
RNOFYS(i,n)- moisture runoff from soil surface, mm/M2/year, I1=plant, I2=region

D,11
RNYLX(n)- yearly runoff, inches, I1=region

C,I,16
ROCK(i,ks,n)- rock fraction in soil, I1=plant, I2=soil layer, I3=region

D,13
RPNTP(i,n)- root penetration into soil horizon, mm, I1=plant, I2=region

D,15
RPT(ka,i,nq,m,n)- cummulative dose to plant, rads, I1=simulation year(ai1), I2=plant, I3=number of plant parts, I4=radionuclide, I5=region

C,I,3,12
RR(ko)- animal/human gate(0,1,-1,mature weight,-mature weight), I1=animal/human

C,I,13
RSCDE(n)- succession gate (0,1) for succession simulation, I1=region

C,13
RSPF2(i,n)- live-to-dead plant mortality rate, per day, I1=plant,I2=region

D,3,12
RSRT(kr,ko)- animal/human respiration rate, M3/day, I1=age, I2=animal/human

C,I,13
RSS(i,n)- niche competition selector, I1=plant, I2=region

C,I,3,12
RSTBL(ko,m)- fractional transport, per day, from residual tissues and organs to blood, I1=animal/human, I2=contaminant

D,15
RSUSP(i,m,n)- estimated resuspension factor, I1=plant, I2=contaminant, I3=region

D,13
RTRANS(i,n)- fraction of photosynthate going to roots, I1=plant, I2=region

C,13
RXPP(i,n)- root respiration coefficient 1, per day, I1=plant, I2=region

D,11
RYCF(n)- yearly runoff, cf, I1=region

C,I,3,12
RZOON(ko)- number of different feeding schedules, I1=animal/plant

C,13
SA- photsynthetic efficiency for C4 plants, gdwt/ly/day

D,17
SACT(ko,m,n)- contaminant concentration, per gdwt, in and on consumed forage, I1=plant preference, I2=contaminant, I3=region

D,17
SAIR(ko,m,n)- contaminant concentration, per gdwt, on consumed forage (dead and live leaves and fruits), I1=plant preference, I2=contaminant, I3=region

D,15
SALCRP(i,m,n)- estimated saltation-creep fraction, I1=plant, I2=contaminant,I3=region

C,I,16
SAND(i,ks,n)- sand fraction in soil, I1=plant, I2=soil layer, I3=region

C,11
SAREA(n)- total area of watershed-receiving area, I1=region

D,10

SATB(i,kt,m,n)- contaminant concentration, per gdwt, in tree, I1=plant, I2=diameter class, I3=contaminant, I4=region

D,9
SATRS(m,n)- removed contaminant, per M2, from both internal and external sources as a result of timber management (fire), I1=contaminant, I2=region

C,I,11
SCHAN(30)- slope, fraction, of locale, I1=region/channel

C,13
SCTP(i,n)- woody/herbaceous plant gate(0,1), I1=plant, I2=region

D,8
SCVR(i,n)- fraction of regional crown cover, I1=plant, I2=region

D,13
SD(i,n)- fresh fruit biomass, gdwt/M2, I1=plant, I2=region

D,13
SD1(i,n)-dry fruit biomass, gdwt/M2, I1=plant, I2=region

D,13
SD3(i,n)- dry standing biomass of fruit production, gdwt/M2/day, I1=plant, I2=region

I,D,8
SDMNCT(30)- contaminant concentration, per gdwt, in surface soil or channel alluvium, I1=region/channel

I,D,8
SDMNSA(30)- amount of contaminant in lateral area(region) assuming 20 mm soil depth and 1.5 bulk density, I1=region/channel(assume 200 mm thickness)

D,13
SDMX(i,n)- maximum simulated fruit biomass for year, gdwt/M2, I1=plant, I2=region

C,I,16
SDPTH(i,n)- soil depth, mm, for a given vegetation, I1=plant, I2=region

C,I,16
SEDGAT- simulation year when watershed operation becomes active

C,8
SEDMNS(30)- weight, lbs, of soil in lateral area(region) assuming 20 mm thickness and 1.5 bulk density, I1=region/channel(assume 200 mm thickness)

D,13
SFAC(i,n)- succession factor,fraction, I1=plant, I2=region

C,7
SFFP(n)- start of frost-free period, julian day, I1=region

D,4

SG(5,kw,n)- death increment function, 1,2,..., for phytoplankton, I1=phytoplankton species, I2=wate layer, I3=region

D,4
SILCN(kw,n)- silicate silicon concentration, ug/l, for a specific soil layer, I1=water layer, I2=region

C,I,16
SILT(i,ks,n)- silt fraction in soil, I1=plant, I2=soil layer, I3=region

C,I,4
SINK(5,n)- sink rate, M/day, for phytoplankton, I1=phytoplankton species, I2=region

D,14
SIR(i,n)- runoff soil moisture water storage parameter, mm, I1=plant, I2=region

C,I,16
SLICE(i,ks,n)- thickness, mm, of soil layer, I1=plant, I2=soil layer, I3=region

C,5
SLOPX(30)- slope of bottom of channel, fraction, I1=channel

C,15
SLOSS(m,n)- eluviation rate, mm/year, of contaminant in or on humus, I1=contaminant, I2=region

D,T,12
SLUNG(m)- contaminant content in human lung, I1=contaminant

D,4
SNFRC(n)- fraction incident light not backscattered from water layer, I1=region

D,17
SNIT(i,n)- percent nitrogen in live biomass from range for animal consumption, I1=plant, I2=region

D,7
SNM(n)- total snow accumulation, mm, I1=region

C,16
SOILF(i,n)- soil depth conversion fraction for default soil layer determination, I1=plant, I2=region

D,T,7
SOLRAD- solar radiation, ly/day, on ground surface

D,20
SORGB(6,2,ko)- total contaminant in population, person-content, I1=cancer type(1-6), I2=sex(1=male,2=female), I3=animal/human

D,20
SORGD(6,2,ko)- total dose, person-rems, to population, I1=cancer type(1-6), I2=sex(1=male,2=female), I3=animal/human

D,17
SOTOT(ko)- total food available from all sources, kgdwt/day, I1=animal

D,17
SPAA(ko,i,m,n)- contaminant concentration, per gdwt, on surface of plant or plant parts(fruits, seeds..), designated for animal or human consumption, I1=animal/human, I2=plant, I3=contaminant, I4=region

D,17
SPAC(ko,m,n)- contaminant in consumed feed mix, per gdwt, I1=animal/human, I2=contaminant, I3=region

D,17
SPAR(ko,m,n)- surficial contaminate concentration on consumed feed mix, per gdwt, I1=animal, I2=contaminant, I3=region

D,17
SPAS(ko,i,m,n)- contaminant concentration, per gdwt, in plant or plant or plant parts(fruits, seeds...), I1=animal/human, I2=plant, I3=contaminant, I4=region

D,20
SPNCAN(2,ko)- total number of spontaneous cancers exclusive those caused by radiation, I1=sex(1=male,2=female), I2=animal/human

C,I,20
SRUN(ko)- year of initial radiation exposure, I1=animal/human

D,15
STEM12(i,n)- temperature, deg. C., of second soil layer, I1=plant, I2=region

D,T,15
STMTT- mean soil temperature, deg. C., for second soil layer

D,13
STRANS(i,n)- fraction of photosynthate going to above non-fruit vegetation, per day, I1=plant, I2=region

D,17
SUMPUT(ko)- diet fraction accumulator(cannot exceed 1), I1=animal

D,7
SUNMN(n)- minimum solar inclination angle, deg., I1=region

D,7
SUNMX(n)- maximum solar inclination angle, deg., I1=region

C,15
SUSFC(m,n)- potential resuspension factor, I1=contaminant, I2=region

D,13
SYAV(i,n)- year counter for estimating yearly weighted net productivities, year, I1=plant, I2=region

D,7
T(n)- mean daily temperature, deg. C., I1=region

C,I,7
T1(n)- minimum insolation, ly/day, for year, I1=region

C,I,7
T2(n)- maximum insolation, ly/day, for year, I1=region

C,I,7
T3(n)- minimum average daily temperature for year, deg. C., I1=region

C,I,7
T4(n)- maximum average daily temperature for year, deg. C., I1=region

D,13
T5P(i,n)- transpiration rate, ml/M2/day, I1=plant, I2=region

D,7
TAV(n)- mean annual temperature for site, I1=region

C,3,12
TBBNZ(100)- biological half-life in bone , days, of contaminant in standard man, I1=nuclide

C,I,5
TBCH(30)- width, ft, of top of distal end of channel aquifer, I1=channel

C,3,12
TBKDZ(100)- biologiccal half-life in kidneys, days, of contaminant in standard man, I1=nuclide

C3,12
TBLVZ(100)- biological half-life in liver, days, of contaminant in standard man, I1=nuclide

C,I,10
TBRM(i,n)- number of timber managements(including a fire(s)), I1=plant, I2=region

C,3,12
TBWBZ(100)- biological half-life in whole body, days, of contaminant in standard man, I1=nuclide

C,I,5
TCH(n)- width, feet, of top of proximal end of channel aquifer, I1=region

D,11
TCHAN(30)- mean ambient temperature, deg. C., in region/channel area, I1=region/channel

D,20
TCN(200,kr,2,ko)- total number of projected cancers due to absolute risk model, I1=simulation year(1-200), I2=age(years), I3=sex(1=male,2=female), I4=animal/human

D,20
TCNAT(6,2,ko)- total spontaneous cancer produced by other thanradiation exposure, I1=age(years), I2=sex(1=male,2=female), I3=animal/human

D,20
TCREL(200,kr,2,ko)- total number of projected cancers due to relative risk model, I1=simulation year(1-200),I2=age(years), I3=sex(1=male, 2=female), I4=animal/human

C,I,9
TCUT(i,10,n)- type of management(1-3) specified, I1=plant, I2=management, I3=region

D,11
TCVR(n)- total crown cover, fractional, of all non-irrigated plants, I1=region

D,4
TF(2,kw,n)- effect of water temperature on water filtration rate of zooplankton, fraction, I1=zooplankton type, I2=water layer, I3=region

D,7
THR(24,n)- night temperatures, deg. C., for a given day, I1=hour, I2=region

D,19
THZ(24)- hourly temperatures, deg. C., used in INVERT, I1=hour(24)

D,13
TIM(i,n)- julian day counter for harvesting events which go beyond 365, I1=plant, I2=region

D,13
TLAI(i,n)- leaf area index, M2/M2, I1=plant, I2=region

C,4
TLLV(5,n)- lower temperature tolerance for photosynthesis, deg. C., for phytoplankton, I1=phytoplankton species, I2=region

D,7
TM(12,n)- mean monthly average daily temperature, deg. C., for climograph generation, I1=month, I2=region

D,7
TMN(n)- minimum daily temperature of site for a given day, deg. C., I1=region

D,4
TMPDAY(30,ko)- tempearture-day, deg. C.-day, index increment for young adult development, I1=young adult age(days), I2=invertebrate

D,7
TMS(12,n)- cummulative monthly temperature, deg. C.-month, I1=month, I2=region

D,7
TMX(n)- maximum daily temperature of site for a given day, deg. C., I1=region

C,I,7
TMXV(n)- mean monthly temperature standard deviation, deg. C., I1=region

D,7
TN(n)- cummulative temperature, deg-day, for a given month, I1=region

D,8
TNDLX(n)- sediment yield, tons/acre/day, I1=region

D,8
TNDTN(n)- regional sediment yield, tons/day, I1=region

D,17
TNIT(ko,n)- total nitrogen fraction in consumed forage, I1=animal/human, I2=region

C,I,10
TNTR(i,n)- total number of trees in an even-aged stand, I1=plant, I2=region

D,11
TNYLX(n)- sediment yield, tons/acre/day, I1=region

D,11
TNYTN(n)- regional sediment yield, tons/year, I1=region

C,I,13
TOFH(i,n)- maximum temperature tolerance, deg. C., I1=plant, I2=region

C,I,13
TOFL(i,n)- plant lower temperature tolerance, deg. C., I1=plant, I2=region

C,I,4
TOPM(5,n)- optimum temperature for photosynthesis, deg. C., I1=phytoplankton species, I2=region

C,I,13
TOPT(i,n)- optimum temperature for photosynthesis, deg. C., I1=plant, I2=region

D,10
TOTAC(m,n)- total internal live above ground contaminant concentration, per M2, for first diameter class, I1=contaminant, I2=region

D,10
TOTSA(m,n)- total stand internal live above ground contaminant concentration, per M2, I1=contaminant, I2=region

D,17
TOTSF(ko,i,n)- total non-range feed...grains, fruits.., kg/day, in diet for an animal or human of a given age, I1=animal/human, I2=plant, I3=region

D,13
TGGNW(i,n)- running average(10 year intervals) net annual productivity, gdwt/M2/year, I1=plant, I2=region

D,3,12
TPOP(kr,2,ko)- number of organisms in a population age group, I1=age(yrs), I2=male/female(1,2), I3=animal/human

D,19
TPSM(ko)- temperature gate(0,2,..n) which allows feeding to begin, I1=invertebrate

C,3,12
TQX(kr,2,ko)- fractional mortality rate for organisms in a population age group, I1=age(yrs), I2=male/female(1,2), I3=animal/human

C,I ,7
TRE(12,n)- total number of rain events in each month, I1= month number (1-12), I2=region

D,10
TREES(i,n)- number of trees in a stand, I1=plant, I2=region

D,13
TSRF- surface temperature of soil, deg. C. for a given region

D,17
TTZ(ko)- mean average daily temperature, deg. C., for utilized area(may include more than one region with different uilization fractions(rfz)), I1=animal/human

C,I,4
TULV(5,n)- upper temperature tolerance for photosynthesis, deg. C., for phytoplankton, I1=phytoplankton species, I2=region

D,7
TX(12,n)- running mean annual average daily temperature of site, deg. C., on last year of simulation, I1=month, I2=region

D,T,12
UGRN- grain intake, gdwt/day, by humans of a given age group

C,I,13
ULEF(i,n)- final fraction of above ground biomass assigned to
leaves, I1=plant, I2=region

D,T,12
ULV- leafy vegetable intake, gdwt/day, by humans of a given age group

D,4
UMNT(5,n)- phytoplankton nitrate incorporation rate, ug/hr, I1=phytoplankton species, I2=region

D,4
UMPH(5,n)- phytoplankton phosphate incorporation rate, ug/hr, I1=phytoplankton species, I2=region

D,4
UMSI(5,n)- phytoplankton silicate incorporation rate, ug/hr, I1=phytoplankton species, I2=region

D,17
UP(i,n)- food available for consumption, kgdwt/day, I1=plant, I2=region

C,I,15
UPK(n)- regional output gate(0,1) for UPTKE subroutine, I1=region

D,3
URNLOS(ko)- fractional loss from ingested food to urine, I1=animal

D,T,12
USV- seed(fruit) intake, gdwt/day, by humans of a given age group

D,3,12
V(kr,ko,m)- total contaminant in blood, I1=age(years), I2=animal/human, I3=contaminant

D,10
V4- volume, ft3/acre, of tree stand estimated from live standing biomass

C,I,10

V8(i,n) -stem diameter(cm) to mass(kg) regression coefficient for tree stand simulation, I1=plant, I2=region

D,10
VA(kt,m,n)- contaminant concentration in tree, per M2, I1=diameter class, I2=contaminant, I3=region

D,5
VAQF(30)- channel aquifer water volume, ft3, I1=channel

D,5
VAQT(30)- channel aquifer volume, ft3, assuming a porosity of 0.25, I1=channel

D,3,12
VBN1(kr,ko,m)- total contaminant in bone(1), I1=age(years), I2=animal/human, I3=contaminant

D3,12
VBN2(kr,ko,m)- total contaminant in bone(2), I1=age(years), I2=animal/human, I3=contaminant

D,8
VCHAN(30)- mean velocity of surface flow, ft/s, from surfacerunoff, I1=region/channel

D,9
VCUT(kt,n)- volume of timber, ft3, removed from stand, I1=diameter class, I2=region

D,3,12
VFAT(kr,ko,m)- total contaminant in fat tissue, I1=age(years), I2=animal/human, I3=contaminant

D,12
VFEC(kr,2,m)- total contaminant loss to feces by humans, per day, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,3,12
VGUT(kr,ko,m)- total contaminant ingested, per day, I1=age, I2=animal/human, I3=contamiant

D,3,12
VIIIN(m)- same as (V) above, I1=contaminant

D,12
VINTK(kr,2,m)- total contaminant intake, per day, by humans from all sources including respiratory tract deposition, foods, and water, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,3,12
VKD1(kr,ko,m)- total contaminant in kidney(1), I1=age(years), I2=animal/human, I3=contaminant

D,3,12
VKD2(kr,ko,m)- total contaminant in kidney(2), I1=age(years), I2=animal/human, I3=contaminant

D,11
VLAT(30)- lateral surface flow into channel, cfd, I1=channel

D,12
VLOS(kr,2,m)- cumulative urine, fecal, and milk losses of contaminant, per day, in humans I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,5
VLOSS(30)- runoff losses from channel aquifer, cfd, I1=channel

D,3,12
VLUG(kr,ko,m)- ingested contaminant from lung clearance, per day

D,3,12
VLV(kr,ko,m)- total contaminant in liver, I1=age(years), I2=animal/human, I3=contaminant

C,3,12
VMK(kr,ko,m)- fractional transport, per day, of contaminant fromblood to milk, I1=age(years), I2=animal/human, I3=contaminant

D,3,12
VMS(kr,ko,m)- total contaminant in muscle, I1=age(years), I2=animal/human, I3=contaminant

D,3
VPELT(kr,ko,m)- total contaminant on pelt, I1=age, I2=animal, I3=contaminant

D,3,12
VPLN(kr,ko,m)- contaminant ingestion from plant foods, per day, I1=age, I2=animal/human, I3=contaminant

D,3
VSOL(kr,ko,m)- contaminant ingestion from soil, per day

D,3,12
VR(kr,ko,m)- total contaminant in residual tissues and organs, I1=age(years), I2=animal/human, I3=contaminant

D,11
VRFL(30)- sub-surface return flow into channel aquifer, cfd, I1=channel

D,12
VRT(kr,2,m)- total body content of contaminant including bone, kidneys, muscle, liver, lungs, and blood in humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,12
VRT1(kr,2,m)- same as (vrt) excluding lung and blood contents

D,10
VTMBR(i,kt,n)- timber volume, ft3, I1=plant, I2=diameter class, I3=region

D,11
VTRND(30)- runoff channel transmission loss, cfd, I1=channel

C,I,11
VTRNDP(30)- perrenial flow gate(0, cfs) for any channel, I1=channel

D,12
VUPTK(kr,2,m)- total contaminant uptake, per day, by humans from all sources entering blood, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,3,12

VURIN(kr,ko,m)- fractional transport, per day, to urine from kidneys, I1=age(years), I2=animal/human, I3=contaminant

D,10
VVTM(i,kt,n)- previous year tree volume, ft3, I1=plant, I2=diameter class, I3=region

D,3,12
W(30,ko,m)- weighted average contaminant concentration in various organs, tissues, and fluids, I2=animal/human, I3=contaminant, i1(1-30):per kgfwt organ/body, unless noted: ,1=body milk clearance, 2=body urine clearance,3=lung clearance to gut, 4=body soil intake, 5=body plant intake, 6=total body, 7=pelt intake as food, 8=total intake, 9=kidneys,10=bone, 11=residual tissues, 12=not used, 13=muscle, 14=liver, 15=blood, 16=lungs, 17=fat tissues,18=daily milk, per liter, 19=urine, per liter, 20=feces, per liter, 21-30 (not used)

C,I,15
W1(i,ks,n)- soil wilt point, mm, in soil layer, I1=plant, I2=soil layer, I3=region

D,15
W1BAR(ks,n)- volumetric water content, mm, at wilt point, I1=soil layer, I2=region

D,15
WA(i,ks,n)- soil moisture content, mm, in a given soil layer, I1=plant, I2=soil layer, I3=region

D,15
WASUMR(m,n)- total removal of resuspendable contaminant, per M2/day, from all plant surfaces due washoff, I1=contaminant, I2=region

D,15
WASUMS(m,n)- total removal of saltation-creep-rainsplash contaminant, per M2/day, from all plant surfaces due to washoff

D,16
WATAKD(i,ks,n)- water loss, mm/day, from soil layer vertically downward, I1=plant, I2=soil layer, I3=region

D,16
WATAKU(i,ks,n)- water loss, mm/day, from soil layer vertically upward, I1=plant, I2=soil layer, I3=region

C,I,16
WATCN(m,n)- contaminant concentration in irrigation or drinking water, per liter, I1=contaminant, I2=region

C,12
WATCNH(m,n)- contaminant concentration, per l, in human drinking water, I1=contaminant, I2=region

C,I,12
WATCNI(m,n)- contaminant concnetration, per l, in human drinking water, I1=contaminant, I2=region

C,I,3
WATCNR(ko,m,n)- contaminant concentration, per l, in animal drinking water, I1=animal, I2=contaminant, I3=region

D,17

WATER(i,n)- percent water in range food, I1=plant, I2=region

D,3
WATFOD- amount of water intake, l/day, from food intake by animal

D,4
WATMP(kw,n)- temperature of water layer, deg. C., I1=water layer, I2=region

D,16
WATUPA(i,ks,n)- moisture flow, mm/day, from above into specific soil layer, I1=plant, I2=soil layer, I3=region

D,16
WATUPB(i,ks,n)- moisture flow flow, mm/day, from below into specific soil layer, I1=plant, I2=soil layer, I3=region

D,3
WATUPT- net water deficit, l/day, from that required by animal

D,3
WATVOL- total water consumption, l/day, ingested by animal

C,I,16
WATX(n)- regional gate for WATFLX subroutine(1,0), I1=region

D,T,3,12
WBURD(m)- total contaminant uptake, per day, from ingested drinking water by an animal or human of a given age, I1=contaminant

D,12
WBURI(kr,2,m)- cumulative contaminant uptake (wburd) from drinking water in humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,I,11
WCHAN(30)- width, feet, of channel, I1=channel

D,13
WD1(i,n)= dry leaf biomass, gdwt/M2, I1=plant, I2=region

D,13
WD3(i,n)- dead standing leaf biomass production, gdwt/M2/day, I1=plant, I2=region

D,16
WFR(i,ks,n)- volumetric soil moisture potential for water in soil, mm/mm, I1=plant, I2= soil layer, I3=region

D,3,12
WMK(ko)- total milk production, l/day, from producer females, I1=animal/human

D,3,12
WMS(i1,i2)- body-muscle masses, kgfwt, I2=animal/human, i1:
1= total body mass, 2=body muscle mass

D,13

WND(i,n)- leaf biomass, gdwt/M2, I1=plant, I2=region

D,13
WNDMAX(i,n)- maximum simulated leaf biomass for year, gdwt/M2, I1=plant, I2=region

D,16
WSRP(i,n)- ponded water, mm, above top soil layer, I1=plant, I2=region

C,3,12
WT1(kr,ko)- expected weight, kg, of animal/human, I1=age(years), I2=animal/human

D,3,12
WTBN(kr,ko)- bone weight, grams, I1=age(years), I2=animal/human

D,3,12
WT(kr,ko)-simulated weight, kg, of animal/human, I1=age(years), I2=animal/human

D,3,12
WTBL(kr,ko)- blood weight, grams, I1=age(years), I2=animal/human

C,I,16
WTF(n)- region WATFLX output selector, I1=region

D,3,12
WTKD(kr,ko)- kidney(s) weight, grams, I1=age(years), I2=animal/human

D,3,12
WTLV(kr,ko)- liver weight, grams, I1=age(years), I2=animal/human

D,3,12
WTLU(kr,ko)- lung(2) weight, grams, I1=age(years), I2=animal/human

D,3,12
WTMS(kr,ko)- muscle weight, grams, I1=age(years), I2=animal/human

D,3,12
WTRS(kr,ko)- residual tissue weight, grams, I1=age(years), I2=animal/human

D,5
WTX(30)- width of channel, feet, adjusted for aquifer height, I1=channel

D,13
WURAT(i,n)- net annual productivity, gdwt/M2/year, I1=plant, I2=region

C,I,13
WUTP(i,n)- plant water utilization efficiency, ml/gdwt photosynthate, I1=plant, I2=region

D,13
WYAV(i,n)- cummulative net productivity(10 year interval), gdwt/M2/year, I1=plant, I2=region

D,15
WZ3(i,m,n)- contaminant in live leaves, per M2, I1=plant, I2=contaminant, I3=region

D,15
WZ4(i,m,n)- contaminant in dead leaves, per M2, I1=plant, I2=contaminant, I3=region

D,15
WZ42(i,m,n)- resuspendable contaminant on surface of dead leaves, per M2, I1=plant, I2=contaminant, I3=region

D,15
WZ43(i,m,n)- saltation-creep-rainsplash contaminant on surface of dead leaves, per M2, I1=plant, I2=contaminant, I3=region

D,15
WZ4C(i,m,n)- contaminant concentration, per gdwt, in dead leaves, I1=plant, I2=contaminant, I3=region

D,15
WZ5(i,m,n)- contaminant in leaf humus, per M2, I1=plant, I2=contaminant, I3=region

D,15
WZ5C(i,m,n)- contaminant concentration, per gdwt, in leaf humus, I1=plant, I2=contaminant, I3=region

D,15
WZ5X(i,m,n)- same as (WZ5)

D,15
WZ7(i,m,n)- internal leaf contaminant concentration, per gdwt, I1=plant, I2=contaminant, I3=region

D,15
WZ7MAX(i,m,n)- maximum internal leaf contaminant concentration, per gdwt, for year, I1=plant, I2=contaminant, I3=region

D,15
WZ8(i,m,n)- concentration factor for live leaves(internal) using top soil layer as reference, I1=plant, I2=contaminant, I3=region

D,15
WZZ4C(i,m,n)- contaminant concentration(internal/external), per gwt, for lying dead leaves, boles(stems), and branches, I1=plant, I2=contaminant, I3=region

C,I,7
X(12,n)- monthly precipitation fraction of total annual, I1=month number(1-12), I2=region

D,15
X8(i,m,n)- concentration factor (internal) for live roots using top soil layer as reference, I1=plant, I2=contaminant, I3=region

C,I,11
XCHAN(30) - length, mi, of locale, I1=region/channel

D,13
XFAC(i,n)- advanced secondary succession factor, fraction, I1=plant, I2=region

C,I,10
XI1(i,n)- diameter class interval, inches.., for classification of bole diameters in stand, I1=plant, I2=region

D,10
XX(kt,n)- tree competition factor(based on crown diameter-stem diameter-no. of trees), I1=diameter class, I2=region

C,16
Y1(105,2)- default radionuclide concentration, per gdwt, in soil(I2=2), and plants(I2=1), I1=atomic number

D,19
YADLT(30,ko)- biomass density of young adult invertebrate, gdwt/M2, I1=age (days), I2=invertebrate

D,19
YADLTS- total biomass density, gdwt/M2, of all age cohorts of young adult invertebrates

C,I,16
YAZ(m,n)- soil contaminant gate(0,1), I1=contaminant, I2=region

D,16
YM(i,ks,n)- soil moisture potential, mm/mm, I1=plant, I2=soil layer, I3=region

D,19
YNDV(30,ko)- young nymph developmental index, I1=nymph age(days), I2=invertebrate

D,19
YNYPH(30,ko)- biomass of young nymphs, g/M2, I1=nymph age(days), I2=invertebrate

D,19
YNYPHS- total biomass, g/M2, of all age cohorts of young nymphs

C,I,3,12
YRFED(ko,32)- yearly feeding intervals(2 year numbers/interval), I1=animal/human, I2=yearly feeding intervals

C,I,16
YSRT(m,n)- simulation year of initial soil contaminantion, I1=contaminant, I2=region

C,I,16
YUP(m,n)- contaminant plant/soil uptake coefficient gate(0,1), I1=contaminant, I2=region

C,I,16
YZF(i,n)- root fraction gate(0,1), I1=plant ,I2=region

D,10
Z(kt,n)- volume of mean tree, ft3, I1=diameter class, I2=region

D,15
Z1(i,ks,m,n)- contaminant concentration, per gdwt, in soil, I1=plant, I2=soil layer, I3=contaminant, I4=region

D,T,15
Z1AT(i,n)- contaminant concentration, per gdwt, in bottom soil layer, I1=plant, I2=region

D,15

Z1AVE(m,n)- root weighted effective contaminant soil concentration, per gdwt, I1=contaminant, I2=region

C,I,16
Z1DL(i,ks,m,n)- contaminant concentration, per gdwt, in soil, at specified simulation time (ysrt,dsrt), I1=plant, I2=soil layer, I3=contaminant, I4=region

D,15
Z2(i,m,n)- total contaminant in live roots, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z3(i,m,n)- total contaminant in live standing vegetation, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z4(i,m,n)- total contaminant in dead standing boles and branches, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z42(i,m,n)- total resuspendable contaminant on dead standing vegetation, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z43(i,m,n)- total saltation-creep-rainsplash contaminant on dead standing vegetation, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z4C(i,m,n)- total contaminant concentration, per gdwt, in dead standing boles(stems) and branches, I1=plant, I2=contaminant, I3=region

D,15
Z4G(i,m,n)- internal contaminant concentration, per M2, in dead lying boles and branches, I1=plant, I2=contaminant, I3=region

D,15
Z4G2(i,m,n)- external resuspendable contamination, per M2, on dead lying bole and branches, I1=plant, I2=contaminant, I3=region

D,15
Z4G3(i,m,n)- external saltation-creep-rainsplash contamination, per M2, on dead lying bole and branches, I1=plant, I2=contaminant, I3=region

D,15
Z4GC(i,m,n)- total contmainant concentration, per gdwt, in dead lying boles and branches, I1=plant, I2=contaminant, I3=region

D,15
Z4R(i,ks,m,n)- contaminant in and on dead roots, per M2, I1=plant, I2=soil layer, I3=contaminant, I4=region

D,15
Z4RC(i,ks,m,n)- total contaminant concentration, per gdwt, in dead root boles and branches, I1=plant, I2=soil layer, I3=contaminant, I4=region

D,15

Z4TT(m,n)- mean resuspendable contaminant concentration of dead lying boles and branches, per M2, I1=contaminant, I2=region

D,15
Z4TTS(m,n)- mean saltation-creep-rainsplash contaminant concentration of lying vegetation, per M2, I1=contaminant, I2=region

D,15
Z4TTSW(m,n)- mean saltation-creep-rainsplash contaminant concentration, per M2, for dead leaves, I1=contaminant, I2=region

D,15
Z4TTW(m,n)- mean resuspendable contaminant concentration, per M2, for dead leaves, I1=contaminant, I2=region

D,15
Z5(i,m,n)- contaminant concentration, per M2, in humus derived from above ground boles and branches, I1=plant, I2=contaminant, I3=region

D,15
Z5C(i,m,n)- total contaminant concentration, per gdwt, in humus derived from above ground boles and branches, I1=plant, I2=contaminant, I3=region

D,15
Z5R(i,ks,m,n)- root humus concentration, per M2, I1=plant, I2=soil layer, I3=contaminant, I4=region

D,15
Z5RC(i,ks,m,n)- total contaminant concentration, per gdwt, in root humus, I1=plant, I2=soil layer, I3=contaminant, I4=region

D,15
Z5RTT(m,n)- mean root humus contaminant concentration, per M2, for first soil layer, I1=contaminant, I2=region

D ,15
Z5TT(m,n)- mean above ground humus concentration, per M2, I1=contaminant, I2=region

D,15
Z5X(i,m,n)- same as (z5)

D,15
Z6(i,m,n)- contaminant concentration in root, per gdwt, I1=plant, I2=contaminant, I3=region

D,15
Z6MAX(i,m,n)- maximum contaminant concentration in roots, per gdwt, for year, I1=plant, I2=contaminant, I3=region

D,15
Z7(i,m,n)-contaminant concentration in bole(stem) and branches, per gdwt, I1=plant, I2=contaminant, I3=region

D,15

Z77(m,n)- total internal and external contaminant concentration factor for above ground biomass (based on contaminant concentration in first soil layer), I1=contaminant, I2=region

D,15
Z77A(m,n)- total internal/external contaminant concentration, per gdwt, for above ground biomass

D,15
Z78(m,n)- surficial(resuspendable) concentration factor for live and dead leaves(based on contaminant concentration for soil layer, and using total above ground biomass), I1=contaminant, I2=region

D,15
Z79(m,n)- same as above except that saltation-creep-deposit evaluated, I1=contaminant, I2=region

D,15
Z7A(m,n)- total contaminant concentration, per gdwt, from all external and internal sources on standing above ground dead biomass(except fruits), I1=contaminant, I2=region

D,15
Z7AR(m,n)- same as above for dead root biomass in top soil layer, I1=contaminant, I2=region

D,15
Z7MAX(i,m,n)- maximum contaminant concentration, per gdwt, in bole(stem) for year, I1=plant, I2=contaminant, I3=region

D,15
Z8(i,m,n)- concentration factor(internal) for above ground boles(stems), and branches(using top soil layer as reference), I1=plant, I2=contaminant, I3=region

D,15
Z9(i,m,n)- decomposing bole(stems) and branches contaminantion accumulator, per M2, I1=plant, I2=contaminant, I3=region

D,15
Z9R(i,m,n)- decomposing root humus contamination accumulator, per M2, I1=plant, I2=contaminant, I3=region

C,16
ZF(i,ks,n)- fraction of root biomass in a given soil layer, I1=plant, I2=soil layer, I3=region

D,16
ZFF(i,ks,n)- fraction of root penetrance in a given soil layer, I1=plant, I2=soil layer, I3=region

C,I,17
ZGR(n)- regional output selector gate(0,1) for ZOOGRO subroutine, I1=region

D,T,12
ZGRN(m)- contaminant intake, per day, by humans of different ages from internal and external contamination on grain crops, I1=contaminant

D,T,12
ZGRNI(kr,2,m)- cummulative contaminant intake (zgrn) by humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,4
ZKQNT(2,n)- minimum nitrate content, ug, for specific zooplankton, I1=zooplankton type, I2=region

C,4
ZKQPH(2,n)- minimum phosphate content, ug, for specific zooplankton, I1=zooplankton type, I2=region

D,T,12
ZLV(m)- contaminant intake, per day, by humans of different ages from internal and external contaminantion on leafy vegetables, I1=contaminant

D,12
ZLVI(kr,2,m)- cummulative contaminant intake (zlv) by humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,4
ZMAX(2,n)- maximum zooplankton biomass, ug, I1=zooplankton type, I2=region

D,T,12
ZMEAT(m)- contaminant intake, per day, by humans of different ages from meat, I1=contaminant

D,12
ZMEATI(kr,2,m)- cummulative contaminant intake(zmeat) by humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

D,T,12
ZMILK(m)- contaminant intake, per day, by humans of different ages from milk, I1=contaminant

D,12
ZMILKI(kr,2,m)- cummulative contaminant intake(zmilk) by humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,4
ZMIN(2,n)- minimum zooplankton biomass, ug, I1=zooplankton type, I1=zooplankton type, I2=region

C,4
ZMNT(2,n)- maximum nitrate content, ug, for specific zooplankton, I1=zooplankton type, I2=region

C,4
ZMPH(2,n)- maximum phosphate content, ug, for specific zooplankton, I1=zooplankton type, I2=region

D,4
ZNITRO(2,kw,n)- zooplankton nitrate content, ug, I1=zooplankton type, I2=water layer, I3=region

D,4
ZOBIO(2,kw,n)- zooplankton biomass, ug, I1=zooplankton type, I2=water layer, I3=region

I,D,3,12
ZOO(ko)- number of organisms in a population, I1=animal/human

D,4
ZOOPC(2,kw,n)- number of zooplankton, per 250 l, I1=zooplankton type, I2=water layer, I3=region

D,4

ZOOPM(2,kw,n)- zooplankton population biomass, ug/250 l..or some other volume, I1=zooplankton type, I2=water layer, I3=region

D,4
ZOSUM(kw,n)- total phytoplankton consumption by zooplankton, ug/day/specified water volume(liters), I1=water layer, I2=region

D,15
ZP(m,n)- contaminant concentration, per gdwt, in top layer of soil, I1=contaminant, I2=region

D,4
ZPHOSP(2,kw,n)- zooplankton phosphate content, ug, I1=zooplankton type, I2=water layer, I3=region

D,15
ZSD(i,m,n)- total contminant in fruits/seeds, per M2, I1=plant, I2=contaminant, I3=region

D,4
ZSM(2,kw,n)- available biomass, ug, to zooplankton for consumption, I1=zooplankton type, I2=water layer, I3=region

D,T,12
ZSV(m)- contaminant intake, per day, by humans from internal and external contamination of fruits(seeds), I1=contaminant

D,12
ZSVI(kr,2,m)- cummulative contaminant intake(zsv) by humans, I1=age(years), I2=sex(1=male,2=female), I3=contaminant

C,4
ZTMN(2,n)- minimum temperature tolerance, deg C., I1=zooplankton type, I2=region

C,4
ZTX(2,n)- maximum temperature tolerance, deg. C., I1=zooplankton type, I2=region

C,16
ZU(i,ks,m,n)- plant contaminant uptake coefficient per ml/M2/day, I1=plant, I2=soil layer, I3=contaminant, I4=region

C,16
ZUP(m,n)- same as (zu)

D,10
ZZ3(i,m,n)- contaminant concentration in tree stand, per M2, I1=plant, I2=contaminant, I3=region

14.0. APPENDIX E

BIOTRAN.2 FORTRAN SOURCE CODE

14.1 MAIN (bam.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      program main
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c       BIOTRAN.2 simulation
c           model
c
c BIOTRAN.2 is used to predict the flow of contaminants through
c user specified plant, animal, and human environments using biomass,
c water, and climate as vectors. up to 15 subroutines can be used to
c to simulate a given scenario; the subroutines are:
c
c   ** airac-acute or chronic air contaminant concentrations
c
c   ** climat-estimates temperature, solar radiation, precipitation, and
c       potential evapotranspiration on a daily basis
c
c   ** aquat-freshwater lake or pond ecosystem
c
c   ** watflx-estimates soil profile moisture flux, and actual
c       evapotranspiration on a daily basis
c
c   ** geoflx- moisture runoff and sediment transport integrator
c
c   ** runoff-estimates moisture runoff and soil infiltration
c
c   ** eroson-estimates sediment transport from overland and channels
c
c   ** aquifer-estimates moisture storage and movement in shallow aquifers
c       along channels
c   ** pltgro-estimates above and below ground live and dead biomass
c       for up to 22 plant types or those created by user
c
c   ** uptake-estimates contaminant intake, transport, and uptake on
c       and in plants
c
c   ** forman-estimates forest and crop stand characteristics on
c       a yearly or daily basis, respectively
c
c   ** forcut-biomass adjuster for forest fires and selective cuttings
c
c   ** zoogro-harvests and stores crops for zoon and human foods.
c     also manages the effects of feeding on food biomass availability
c
c   ** animal-estimates animal metabolism and uptake of contaminants
```

```
c       based on daily diet compositions
c
c   ** humtrn-estimates male and female human metabolism and uptake of
c       contaminants based on age,diet,work activity level,and
c       ambient temperatures
c
c   ** effects-estimates cancer incidences in a population using cohort
c       analysis from radionuclides, or other contaminants where
c       pertinent cancer incidence data can be obtained as a function
c       specific organ burdens
c
c   ** bioplt-output selector for BIOTRAN.2 simulations(output.dat)
c
c   ** truplt-output selector for plotting designated variables(plot.dat)
c
c   ** tplotr- plotting system using INTERACTER software supported by
c       the Salford Compiler FTN77/486 (can be supported by other
c       (compilers as well)
c
c
c main program switches simulation flow based on user input
c
c
      INCLUDE 'main.inc'
      dimension aquax(n), frmnx(n)
      character *5 test
c
c Set the default random number seed in a data statement
c
      data iseed /1000/
c
      open (unit=5,file='output.dat',status='old')
      open (unit=6,file='enter.dat',status='old')
      open (unit=7,file='input.dat',status='old')
      open (unit=8,file='plot.dat',status='old')
c
   10 call scinit(' ')
      call atcol('black','yellow')
      call wnactn('cfp')
      call wnopen(0,0,50,10)
      call wnoust('                                    ')
      call wnoust('      LOS ALAMOS NATIONAL LABORATORY          ')
      call wnoust('                                    ')
      call wnoust('    ECOLOGICAL TRANSPORT SIMULATION MODEL       ')
      call wnoust('                                    ')
      call wnoust('              (BIOTRAN.2)                ')
      call wnoust('                                    ')
      call wnoust('              VERSION 1.0                ')
      call wnoust('                                    ')
      call wnouce(10,'press any key')
      call inkey(key)
      call wnclos(1)
c
      call atcol('white','red')
```

```
      call atbold('on')
      call wnopen(0,0,51,19)
      call wnoust('****************************************************')
      call wnoust('This program was prepared by the Regents of the    ')
      call wnoust('University of California at Los Alamos National    ')
      call wnoust('Laboratory (the University) under contract No.     ')
      call wnoust('W-7405-ENG-36 with the U.S. Department of Energy   ')
      call wnoust('(DOE). The University has certain rights in the    ')
      call wnoust('program pursuant to the contract and the program   ')
      call wnoust('should not be copied or distributed outside your   ')
      call wnoust('organization. All rights in the program are        ')
      call wnoust('reserved by DOE and the University. Neither the    ')
      call wnoust('U.S. Government nor the University makes any       ')
      call wnoust('warranty, expressed or implied, or assumes any     ')
      call wnoust('liability or responsibiliy for the use of this     ')
      call wnoust('software.                                          ')
      call wnoust('Contact A.F. Gallegos, EES-15 at 505-665-0862 for  ')
      call wnoust('further information or questions on the use of     ')
      call wnoust('this software.                                     ')
      call wnoust('****************************************************')
      call wnouce(19,'press any key')
      call inkey(key)
      call wnclos(1)
      call atbold('off')
c
      call atcol('black','green')
      call wnopen(0,0,51,9)
      call wnoust('****************************************************')
      call wnoust('BIOTRAN.2 is used to predict the flow of contaminants')
      call wnoust('through user specified plant, animal, and human    ')
      call wnoust('environments using biomass, moisture flux, and     ')
      call wnoust('climate as drivers. It has other potential uses    ')
      call wnoust('in agriculture,forestry and animal science. The    ')
      call wnoust('next window is a list of subroutine descriptions:  ')
      call wnoust('****************************************************')
      call wnouce(9,'press any key')
      call inkey(key)
      call wnclos(1)
      call atbold('on')
      call atcol('white','green')
      call wnopen(0,0,51,19)
      call wnoust('****************************************************')
      call wnoust('AIRAC- acute/chronic air contamination             ')
      call wnoust('CLIMAT-temp.,solar rad.,precip.,pet,...etc.        ')
      call wnoust('AQUAT- freshwater lake,zooplankton,phytoplankton.. ')
      call wnoust('WATFLX-soil moisture/flux,evapotranspiration....   ')
      call wnoust('GEOFLX-moisture runoff/sediment transport...       ')
      call wnoust('RUNOFF-moisture runoff/infiltration...             ')
      call wnoust('EROSON-overland/channel sediment transport....     ')
      call wnoust('AQUIFER-channel moisture storage/transport...      ')
      call wnoust('PLTGRO-plant live/dead biomass density.....        ')
      call wnoust('UPTAKE-contaminant uptake/transport in plants..    ')
      call wnoust('FORMAN-forest stand generator/contaminant transport')
      call wnoust('FORCUT-forest management/fire...                   ')
```

```
      call wnoust('ZOOGRO-crop storage/food management.....          ')
      call wnoust('ANIMAL-metabolism/contaminant transport/growth...  ')
      call wnoust('HUMTRN-human m/f metabolism/cont.transp./growth..  ')
      call wnoust('EFFECTS-pop.cohort analysis/radionuclide cancer..  ')
      call wnoust('BIOPLT,TRUPLT,TPLOTR-output/plotting system....    ')
      call wnoust('*****************************************************')
      call wnouce(19,'press any key')
      call inkey(key)
      call wnclos(1)
c
      call atcol('white','blue')
      call wnopen(0,0,44,10)
      call wnoust('*********************************************')
      call wnoust('BEGIN BIOTRAN.2 INPUT, YOU ARE IN PROGRAM MAIN')
      call wnoust('AFTER EVERY ENTRY PRESS RETURN or ENTER..  ')
      call wnoust('IF ENTER STATEMENT REPEATS ITSELF, THEN DATA')
      call wnoust('HAS BEEN ENTERED IMPROPERLY: REPEAT ENTRY  ')
      call wnoust('IF A LEADING QUOTE; ("), IS ENTERED, THEN  ')
      call wnoust('MODEL WILL WAIT FOR ENDING (") BEFORE      ')
      call wnoust('CONTINUING TO EXECUTE: GOOD LUCK!          ')
      call wnoust('*********************************************')
      call wnouce(10,'press any key')
      call inkey(key)
      call atbold('off')
 5001 call wnclos(1)
      call atcol('black','yellow')
      call wnopen(0,0,51,9)
      call wnoust('random number seed is set to 1000 to start sequence')
      call wnoust('  TO change random number seed:                  ')
      call wnoust('   choose "true" = no change, same sequence      ')
      call wnoust('   choose "false" = different sequence           ')
      call wnoust('   choose "end" = exit, run successful           ')
      call wnoust(' ENTER choice now, use single quotes as shown     ')
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,9)
      read(*,*,err=5001) test
 5099 call wnclos(1)
      if(test.eq.'end')call clear_screen@
      if(test.eq.'end')goto 120
      if(test.eq.'true')goto 5012
      call wnopen(0,0,51,4)
      call wnoust('CHOOSE an integer random seed...10, 1000, 200, etc.')
      call wnoust('ENTER selected random number seed now             ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5099) iseed
c
c If the run is a test run, set the logical variable TEST to 'true';
c if the run is to use randomly selected parameters, set test to
c 'false'. Setting TEST to 'true' allows the user to check the output
c because the same random number sequence will be selected each time
c through the program. Set TEST to 'end' to exit program.
```

```
c
 5012 call wnclos(1)
      call wnopen(0,0,51,12)
      call wnoust('ENTER 0 if creating input; ENTER 1 if input is    ')
      call wnoust('is to come from a free format file(input.dat)     ')
      call wnoust('obtained by renaming file(enter.dat) created      ')
      call wnoust('automatically by BIOTRAN.2 during each run.        ')
      call wnoust('A partial file with any number of COMPLETE        ')
      call wnoust('subroutine inputs may be used to save time        ')
      call wnoust('model will shift to create mode after all         ')
      call wnoust('input has been entered from partial file          ')
      call wnoust('PARTIAL subroutine input not acceptable           ')
      call wnoust('Enter 0 or 1                                      ')
      call wnouce(11,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,12)
      read(*,*,err=5012) nu
 5013 call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you want an echo of each input variable?       ')
      call wnoust('yes= 1,no= 0; ENTER now! (1 or 0)                 ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5013)iu
      call wnclos(1)
c
c enter control parameters for run; number of simulation years, number of
c contaminants, number of regions, ZOOGRO gate(0,1), ANIMAL gate(0,1),
c HUMTRN gate(0,1), and BIOPLT access(0,1)
c
      call atcol('white','red')
      call atbold('on')
      call wnopen(0,0,51,13)
      call wnoust('**************************************************** ')
      call wnoust('unless specified, all input is NON-INTEGER type    ')
      call wnoust('HOWEVER, whole numbers can be entered as integers  ')
      call wnoust('and are converted to real numbers as shown on ECHO ')
      call wnoust('SEPARATE more than 1 input/line by commas or blanks')
      call wnoust('1.0,2.0e5 4.5,2.; repetitive numbers admitted..ex. ')
      call wnoust('1.0,4*5.0,2*0.0 2.0e4,..;IF next ENTER prompt does ')
      call wnoust('not appear,THEN not enough data has been entered   ')
      call wnoust('and will remain so until complete;CONTINUE to add  ')
      call wnoust('until complete; IF more data are required than can ')
      call wnoust('go on one line, then RETURN and continue as needed ')
      call wnoust('****************************************************')
      call wnouce(13,'press any key')
      call inkey(key)
      nuu=nu
5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
```

```
      call wnoust('ENTER yes=1, no=0                          ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.gt.0.0.and.nu.eq.0)nu=1
      if(ans.gt.0.0.and.nu.eq.1)nu=0
 5002 if(nu.eq.0) call wnclos(1)
      if(nu.eq.1)goto4000
      call wnopen(1,1,80,5)
      call wnoust('ENTER number of years simulated, 1 year is minimum ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5002) y2
      if(iu.eq.1)print *,'value(s)=',y2
      if(iu.eq.1)call sleep@(5.0)
 4002 call wnclos(1)
 4000 if(nu.eq.1)read(7,*)y2
      if(nu.eq.1)goto4001
      call wnopen(0,0,51,4)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=4002) ans
      if(ans.gt.0.)goto5002
 4001 write(6,*) y2
      if(iu.eq.1.and.nu.eq.1) print *,y2
 5003 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4003
      call wnopen(1,1,80,5)
      call wnoust('ENTER number of contaminants simulated exclusive of')
      call wnoust(' nutrients NO3,NH3,PO4(s),PO4(l), and K            ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5003)anz
      if(iu.eq.1)print *,'value(s)=',anz
      if(iu.eq.1)call sleep@(5.0)
 4005 if(nu.eq.0)call wnclos(1)
 4003 if(nu.eq.1)read(7,*)anz
      if(nu.eq.1)goto4004
      call wnopen(0,0,51,4)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey (key)
      call wncuxy(1,3)
      read(*,*,err=4005) ans
      if(ans.gt.0.)goto5003
 4004 write(6,*)anz
```

```
      if(iu.eq.1.and.nu.eq.1) print *,anz
      anz=anz+5.0
 5004 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4006
      call wnopen(1,1,80,4)
      call wnoust('ENTER number of regions simulated                ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5004)apx
      if(iu.eq.1)print *,'values=',apx
      if(iu.eq.1)call sleep@(5.0)
 4008 if(nu.eq.0)call wnclos(1)
 4006 if(nu.eq.1)read(7,*)apx
      if(nu.eq.1)goto4007
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=4008) ans
      if(ans.gt.0)goto5004
 4007 write(6,*)apx
      if(iu.eq.1.and.nu.eq.1)print *,apx
 5005 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4009
      call wnopen(1,1,80,6)
      call wnoust('ENTER access gates (1.0 or 0.0) for ZOOGRO, ANIMAL,')
      call wnoust('HUMAN, and BIOPLT subroutines: 4 entries required  ')
      call wnoust('a value of 1.0 indicates access for any gate      ')
      call wnouce(4,'press any key and enter above                  ')
      call inkey(key)
      call wncuxy(1,5)
      read(*,*,err=5005)zoox,anix,humx,bpx
      if(iu.eq.1)print *,'value(s)=',zoox,anix,humx,bpx
      if(iu.eq.1)call sleep@(5.0)
 4011 if(nu.eq.0)call wnclos(1)
 4009 if(nu.eq.1)read(7,*)zoox,anix,humx,bpx
      if(nu.eq.1)goto4010
      call wnopen(0,0,51,4)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=4011) ans
      if(ans.gt.0.)goto5005
 4010 write(6,*)zoox,anix,humx,bpx
      if(iu.eq.1.and.nu.eq.1 )print *,zoox,anix,humx,bpx
      rdidx=0.0
      inz=anz
      ipx=apx
      nan=0
      nhm=0
c enter number of plants simulated, AIRAC gate(0,1),WATFLX gate(0,1),FORMAN
```

```
c gate(0,1), and AQUAT gate(0,1) for each region(LZ)
      if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4012
      call wnopen(0,0,51,4)
      call wnoust('            ********************            ')
      call wnoust('             regional inputs:             ')
      call wnoust('            ********************            ')
      call wnouce(4,'press any key')
      call inkey(key)
 5006 call wnclos(1)
      call wnopen(1,1,80,7)
      call wnoust('ENTER number of plant types in each region for    ')
      call wnoust('                                                  ')
      call wncuxy(1,2)
      print *,apx,'region(s) in simulation'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5006) (apl(lz),lz=1,ipx)
      if(iu.eq.1)print *,'value(s)=',(apl(lz),lz=1,ipx)
      if(iu.eq.1)call sleep@(5.0)
 4014 if(nu.eq.0)call wnclos(1)
 4012 if(nu.eq.1)read(7,*) (apl(lz),lz=1,ipx)
      if(nu.eq.1)goto4013
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=4014) ans
      if(ans.gt.0.)goto5006
 4013 write(6,*) (apl(lz),lz=1,ipx)
      if(iu.eq.1.and.nu.eq.1)print *, (apl(lz),lz=1,ipx)
 5007 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4015
      call wnopen(1,1,80,5)
      call wnoust('ENTER airac gate(1.0,0.0) in region for          ')
      call wnoust('                                                 ')
      call wncuxy(1,2)
      print *,apx,'region(s) in simulation'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5007) (airx(lz),lz=1,ipx)
      if(iu.eq.1)print *,'value(s)=',(airx(lz),lz=1,ipx)
      if(iu.eq.1)call sleep@(5.0)
 4017 call wnclos(1)
 4015 if(nu.eq.1)read(7,*) (airx(lz),lz=1,ipx)
      if(nu.eq.1)goto4016
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
```

```
      if(nu.eq.0)read(*,*,err=4017) ans
      if(ans.gt.0.)goto5007
 4016 write(6,*) (airx(lz),lz=1,ipx)
      if(iu.eq.1.and.nu.eq.1)print *, (airx(lz),lz=1,ipx)
 5008 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4018
      call wnopen(1,1,80,5)
      call wnoust('ENTER watflx gate(1.0,0.0) in each region for     ')
      call wnoust('                                                  ')
      call wncuxy(1,2)
      print *,apx,'region(s) in simulation'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5008)(watx(lz),lz=1,ipx)
      if(iu.eq.1)print *,'value(s)=',(watx(lz),lz=1,ipx)
      if(iu.eq.1)call sleep@(5.0)
 4020 call wnclos(1)
 4018 if(nu.eq.1)read(7,*)(watx(lz),lz=1,ipx)
      if(nu.eq.1)goto4019
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=4020) ans
      if(ans.gt.0.)goto5008
 4019 write(6,*)(watx(lz),lz=1,ipx)
      if(iu.eq.1.and.nu.eq.1)print *,(watx(lz),lz=1,ipx)
 5009 if(nu.eq.0)call wnclos(1)
      if(nu.eq.1)goto4021
      call wnopen(1,1,80,5)
      call wnoust('ENTER forman gate(1.0,0.0) in each region for')
      call wnoust('                                              ')
      call wncuxy(1,2)
      print *,apx,'region(s) in simulation'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5009) (frmnx(lz),lz=1,ipx)
      if(iu.eq.1)print *,'value(s)=',(frmnx(lz),lz=1,ipx)
      if(iu.eq.1)call sleep@(5.0)
 4023 call wnclos(1)
 4021 if(nu.eq.1)read(7,*) (frmnx(lz),lz=1,ipx)
      if(nu.eq.1)goto4022
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=4023) ans
      if(ans.gt.0.)goto5009
 4022 write(6,*) (frmnx(lz),lz=1,ipx)
      if(iu.eq.1.and.nu.eq.1)print *, (frmnx(lz),lz=1,ipx)
```

```
5010 if(nu.eq.0)call wnclos(1)
   if(nu.eq.1)goto4024
   call wnopen(1,1,80,6)
   call wnoust('ENTER aquat gate(1.0,0.0) in each region for      ')
   call wnoust('                                                  ')
   call wncuxy(1,2)
   print *,apx,'region(s) in simulation'
   call wnouce(3,'press any key and enter above')
   call inkey(key)
   call wncuxy(1,4)
   read(*,*,err=5010) (aquax(lz),lz=1,ipx)
   if(iu.eq.1)print *,'value(s)=',(aquax(lz),lz=1,ipx)
   if(iu.eq.1)call sleep@(5.0)
4026 call wnclos(1)
4024 if(nu.eq.1)read(7,*) (aquax(lz),lz=1,ipx)
   if(nu.eq.1)goto4025
   call wnopen(0,0,51,3)
   call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
   call wnouce(2,'press any key and enter above')
   call inkey(key)
   call wncuxy(1,4)
   if(nu.eq.0)read(*,*,err=4026) ans
   if(ans.gt.0.)goto5010
4025 write(6,*) (aquax(lz),lz=1,ipx)
   if(iu.eq.1.and.nu.eq.1)print *, (aquax(lz),lz=1,ipx)
5014 if(nu.eq.0)call wnclos(1)
   if(nu.eq.1)goto4027
   call wnopen(1,1,80,4)
   call wnoust('ENTER day and year modulus                        ')
   call wnouce(2,'press any key and enter above')
   call inkey(key)
   call wncuxy(1,3)
   read(*,*,err=5014) dmd,ymd
   if(iu.eq.1)print *,'value(s)=',dmd,ymd
   if(iu.eq.1)call sleep@(5.0)
4029 call wnclos(1)
4027 if(nu.eq.1)read(7,*) dmd,ymd
   if(nu.eq.1)goto4028
   call wnopen(0,0,51,4)
   call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
   call wnouce(2,'press any key and enter above')
   call inkey(key)
   call wncuxy(1,3)
   read(*,*,err=4029) ans
   if(ans.gt.0.)goto5014
4028 write(6,*) dmd,ymd
   if(iu.eq.1.and.nu.eq.1)print *, dmd,ymd
   if(nu.eq.1) print *,'********** END OF INPUT IN MAIN ***********'
   if(nu.eq.1)goto5500
   call wnclos(1)
   call clear_screen@
   call wnopen(0,0,51,2)
   call wnoust('*********** END OF INPUT IN MAIN ******************')
   call wnouce(2,'press any key')
```

```
      call inkey(key)
 5011 call wnclos(1)
      call clear_screen@
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for MAIN?          ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5011) ans
      if(ans .lt.1.0)goto5500
      rewind 6
      nu=nuu
      call clear_screen@
      goto10
 5500 continue
      nu=nuu
      call clear_screen@
c
c ** region loop ***************************************************
      do 20 lx=1,ipx
c plant number index
      ipl=apl(lx)
c conditional call to airac subroutine
      if(airx(lx).le.0.0)goto21
      call airac (iyr,lx,rdidx,bpx)
      call sleep@(2.0)
      call clear_screen@
c un-conditional call to subroutines for input and initialization
   21 call climat (lx,bpx,rdidx,y2,iseed)
      call sleep@(2.0)
      call clear_screen@
      if(watx(lx).gt.0.0)call watflx (ipn,lx,bpx,rdidx)
      call sleep@(2.0)
      call clear_screen@
      call pltgro (ipn,lx,rdidx,bpx)
      call sleep@(2.0)
      call clear_screen@
      call uptake (dn,lx,rdidx,bpx)
      call sleep@(2.0)
      call clear_screen@
c conditional call to subroutines (gated) for input and initialization
      if (frmnx(lx).gt.0.0) call forman (ipn,lx,bpx,rdidx)
      if(frmnx(lx).gt.0.0)call sleep@(2.0)
      if(frmnx(lx).gt.0.0)call clear_screen@
      if (anix.gt.0.0) call animal (ii,in,lx,bpx,rdidx)
      if(anix.gt.0.0)call sleep@(2.0)
      if(anix.gt.0.0)call clear_screen@
      if (humx.gt.0.0) call humtrn (idn,lx,ii,fedh,bpx,rdidx)
      if(humx.gt.0.0)call sleep@(2.0)
      if(humx.gt.0.0)call clear_screen@
      if (aquax(lx).gt.0.0) call aquat (in,lx,bpx,rdidx)
      if(aquax(lx).gt.0.0)call sleep@(2.0)
      if(aquax(lx).gt.0.0)call clear_screen@
```

```
c conditional call to output subroutine for input and initialization
    if (bpx.gt.0.0)call bioplt(lpn,lx,jct,y2,tyrt,tdyt,dmd,ymd,rdidx)
    if(bpx.gt.0.0)call sleep@(2.0)
    if(bpx.gt.0.0)call clear_screen@
  20 continue
c ** end region loop ***********************************************
c-----------------------------------------------------------------
c
    rdidx=1.0
c
c-----------------------------------------------------------------
c
c-----------------------------------------------------------------
c executes ij yearly cycles of simulation
    ij=int(y2)
    tyrt=0.0
    tdyt=0.0
c
c ** yearly loop ****************************************************
    do 110 i1=1,ij
    ai1=i1
    tyrt=tyrt+1
    print *,'running, year=',ai1,'of',ij,' years'
    sajj=0.0
c
c ** daily loop *****************************************************
    do 80 iii=1,365
    aiii=iii
    tdyt=tdyt+1
c
c ** regional loop **************************************************
    do 70 lx=1,ipx
    ipl=apl(lx)
c
c ** contaminant loop ***********************************************
    do 30 k=1,inz
    airca(k,lx)=0.0
    aircg(k,lx)=0.0
  30 continue
c ** end contaminant loop *******************************************
    if (airx(lx).gt.0.) call airac (iyr,lx,rdidx,bpx)
    call climat (lx,bpx,rdidx,y2,iseed)
    if (aquax(lx).gt.0.) call aquat (in,lx,bpx,rdidx)
c
c ** plant loop *****************************************************
    do 60 lpn=1,ipl
c
c ** contaminant loop ************************************************
    do 40 k=1,inz
c sets air concentration to that generated in AIRAC
    aaa(lpn,k,lx)=airca(k,lx)
    aag(lpn,k,lx)=aircg(k,lx)
  40 continue
c ** end contaminant loop *******************************************
```

```
      if (watx(lx).gt.0.) call watflx (lpn,lx,bpx,rdidx)
      call pltgro (lpn,lx,rdidx,bpx)
c calls FORMAN subroutine on a daily basis
      if (rgc(lpn,lx)-3.0) 50,51,50
   51 if (frmnx(lx).gt.0.) call forman (lpn,lx,bpx,rdidx)
   50 continue
   60 continue
c ** end plant loop ******************************************************
      if (zoox.ge.1.) call zoogro (idn,lx,bpx,humx,rdidx)
      call uptake (dn,lx,rdidx,bpx)
   70 continue
c ** end region loop *****************************************************
c
c ** regional loop ********************************************************
      jct=0
      do 100 lx=1,ipx
      ipl=apl(lx)
c
c ** plant loop **********************************************************
      do 90 lpn=1,ipl
      if(aiii.lt.365.0)goto92
c calls FORMAN subroutine on a yearly basis
      if (rgc(lpn,lx)-2.0) 92,91,92
   91 if (frmnx(lx).gt.0.) call forman (lpn,lx,bpx,rdidx)
c day and year modulus for output selection
   92 if (dmd) 61,90,61
   61 if (amod(ai1,ymd)-0.0) 90,62,90
   62 if (amod(aiii,dmd)-0.0) 90,63,90
   63 if(bpx.ge.1.0)call bioplt(lpn,lx,jct,y2,tyrt,tdyt,dmd,ymd,rdidx)
      if(lx.eq.0)goto 10
   90 continue
c ** end plant loop ******************************************************
  100 continue
c ** end region loop *****************************************************
   80 continue
c ** end daily loop ******************************************************
  110 continue
c ** end yearly loop *****************************************************
c
c  resets BIOTRAN.2 for new simulation
c
      go to 10
c
  120 stop
      end
```

14.2 AIRAC (bbm.for)

```
      subroutine airac (iyr,lx,rdidx,bpx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  airac estimates chronic and acute contaminant concentrations in
c   air from specified releases
c
      include 'main.inc'
c
      common/zb/ amac(16,48,m,n), arc(n), arcz(m), chrac(m,n),chred(m,n)
     1 , chrey(m,n), chrsd(m,n), chrsy(m,n), cyr(m,n), durc(m,n), icdy(m
     2 ,n), icyr(m,n), yrcnt(16,m,n),amag(16,48,m,n),chrag(m,n)
c
      common/pc/ anday(12), clm(n), dnn(n), drmm(n), ffp(n), g1(n),g4(n)
     1 , g5(n), mmx(n), nx(24,n), pm(12,n), pms(12,n), pn(n),
     2 px(12,n), q(40,n), q8(n), r(n), t3(n), t4(n), tm(12,n), tms(12,n)
     3 , tn(n), tre(12,n), tx(12,n), x(12,n),relv(n),l2(n),l1(n),l3(n),
     4 prcv(n),tmxv(n),t1(n),t2(n),alpsr(n),alatd(n),rada(n),sunmn(n),
     5 sunmx(n)
       dimension eqtyp(m,n),dampe(m,n),arin(m,n),aexp(m,n),arlg(m,n)
     1,arfl(m,n),aexy(m,n)
c
c excludes all but initial access
      if (rdidx-0.) 70,71,70
   71 if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5015
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-111111.)5702,5015,5702
 5555 nu=0
 5015 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('*****************************                    ')
      call wnoust(' YOU ARE IN AIRAC SUBROUTINE                     ')
      call wnoust('*****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected ')
      call wnoust('                                                 ')
      call wnoust('ENTER yes=1, no=0                                ')
      call wnoust('*************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
```

```
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 111111.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
 5001 print *,'elemental output selector (0.0,1.0): selects which'
      print *,'contaminants are to be considered in release(s)'
      if(nu.eq.1)read(7,*) (arcz(jz),jz=1,inz)
      if(nu.eq.1)goto22
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER contaminant event gates (0,1), for ',inz
      call wncuxy(1,2)
      print *,'contaminant(s)'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5001) (arcz(jz),jz=1,inz)
      if(iu.eq.1)print *,'value(s)=',(arcz(jz),jz=1,inz)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5001) ans
      if(ans.gt.0.0)goto5001
   22 write(6,*) (arcz(jz),jz=1,inz)
```

```
      if(iu.eq.1.and.nu.eq.1)print *, (arcz(jz),jz=1,inz)
c
c specifies initial and final release input for each release event
c    on daily and yearly intervals for acute and chronic releases
c
c ** regional loop ******************************************************
      do 60 l=1,ipx
c determines which region has chronic/acute events
      if (airx(l)) 61,60,61
c
c ** contaminant loop ***************************************************
   61 do 50 j=1,inz
c determines which contaminants in run are considered for event(s)
      if (arcz(j)) 51,50,51
c enters contaminant activity(per m3)for  chronic exposure function,
c enters day number for end of chronic exposure, year number for end
c of chronic exposure, day number for start of chronic exposure, and
c year number for start of chronic exposure
c
   51 if(nu.eq.1)read(7,*) chrac(j,l)
      if(nu.eq.1)goto23
      call clear_screen@
      call wnopen(0,0,51,5)
      call wncuxy(1,1)
      print *,'*********************************************'
      call wncuxy(1,2)
      print *,'for CHRONIC and/or ACUTE events for any region;'
      call wncuxy(1,3)
      print *,'enter (0.0) for CHRONIC or ACUTE if not considering'
      call wncuxy(1,4)
      print *,'*************************************************'
      call wnouce(5,'press any key')
      call inkey(key)
 5002 call wnclos(1)
      call wnopen(1,1,80,15)
      call wncuxy(1,1)
      print *,'ENTER initial Chronic particulate gate(1,0)'
      call wncuxy(1,2)
      print *,'conc. in ppm converted to gdwt/gdwt,otherwise ug/m3;'
      call wncuxy(1,3)
      print *,'or pCi/m3 if contaminant in ppm converted to pCi/g,'
      call wncuxy(1,4)
      print *,' otherwise same activity units as soil per m3 as above)'
      call wncuxy(1,5)
      print *,'for contaminant',j,' in region',l,'(0.0) cancels'
      call wnouce(6,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,7)
      read(*,*,err=5002)chrac(j,l)
      if(iu.eq.1)print *,'value(s)=',chrac(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
```

```
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5002) ans
      if(ans.gt.0)goto5002
   23 write(6,*)chrac(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,chrac(j,l)
c
 5014 if(nu.eq.1)read(7,*)chrag(j,l)
      if(nu.eq.1)goto24
      call clear_screen@
      call wnopen(1,1,80,15)
      call wncuxy(1,1)
      print *,'ENTER initial Chronic gaseous concentration gate(1,0)'
      call wncuxy(1,2)
      print *,'conc. in ppm converted to gdwt/gdwt,otherwise ug/m3;'
      call wncuxy(1,3)
      print *,'or pCi/m3 if contaminant in ppm converted to pCi/g,'
      call wncuxy(1,4)
      print *,' otherwise same activity units as soil per m3 as above)'
      call wncuxy(1,5)
      print *,'for contaminant',j,' in region',l,'(0.0 cancels)'
      call wnouce(6,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,7)
      read(*,*,err=5014)chrag(j,l)
      if(iu.eq.1)print *,'value(s)=',chrag(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5014) ans
      if(ans.gt.0.0)goto5014
   24 write(6,*)chrag(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,chrag(j,l)
      if(chrac(j,l).le.0.0.and.chrag(j,l).le.0.0)goto5007
c
 5003 if(nu.eq.1) read(7,*)chred(j,l)
      if(nu.eq.1)goto25
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER julian day number for end of chronic exposure'
      call wncuxy(1,2)
      print *,' for contaminant',j,' in region',l
      call wnouce(3,'press any and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5003) chred(j,l)
      if(iu.eq.1)print *,'value(s)=',chred(j,l)
      call sleep@(5.0)
```

```
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5003) ans
      if(ans.gt.0.0)goto5003
   25 write(6,*) chred(j,l)
      if(iu.eq.1.and.nu.eq.1)print *, chred(j,l)
 5004 if(nu.eq.1)read(7,*)chrey(j,l)
      if(nu.eq.1)goto26
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER year number for end of chronic exposure'
      call wncuxy(1,2)
      print *,'for contaminant ',j,' in region',l
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5004)chrey(j,l)
      if(iu.eq.1)print *,'value(s)=',chrey(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.0)goto5004
   26 write(6,*)chrey(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,chrey(j,l)
 5005 if(nu.eq.1)read(7,*) chrsd(j,l)
      if(nu.eq.1)goto27
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER julian day number for start of chronic exposure'
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      if(nu.eq.0)read(*,*,err=5005)chrsd(j,l)
      if(iu.eq.1)print *,'value(s)=',chrsd(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5005) ans
      if(ans.gt.0.0)goto5005
```

```
 27 write(6,*)chrsd(j,l)
    if(iu.eq.1.and.nu.eq.1)print *,chrsd(j,l)
5006 if(nu.eq.1)read(7,*)chrsy(j,l)
    if(nu.eq.1)goto28
    call clear_screen@
    call wnopen(1,1,80,10)
    call wncuxy(1,1)
    print *,'ENTER year number for start of chronic exposure'
    call wncuxy(1,2)
    print *,'for contaminant ',j,' in region',l
    call wnouce(3,'press any key and enter above')
    call inkey(key)
    call wncuxy(1,4)
    read(*,*,err=5006)chrsy(j,l)
    if(iu.eq.1)print *,'value(s)=',chrsy(j,l)
    call sleep@(5.0)
    call wnclos(1)
    call wnopen(0,0,51,3)
    call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
    call wnouce(2,'press any key and enter above')
    call inkey(key)
    call wncuxy(1,3)
    read(*,*,err=5006) ans
    if(ans.gt.0.0)goto5006
 28 write(6,*)chrsy(j,l)
    if(iu.eq.1.and.nu.eq.1)print *,chrsy(j,l)
5016 if(nu.eq.1)read(7,*) eqtyp(j,l),dampe(j,l)
    if(nu.eq.1)goto355
    call clear_screen@
    call wnopen(1,1,80,6)
    call wncuxy(1,1)
    print *,'ENTER type of chronic equation (x= julian day) used; '
    call wncuxy(1,2)
    print *,'1=exponential, 2=linear, 3=ingrowth, 4=sine, 5=constant'
    call wncuxy(1,3)
    print *,'also ENTER type of equation dampening (x= year number):'
    call wncuxy(1,4)
    print *,'1=exponential, 2=linear, 3=ingrowth, 4=constant'
    call wnouce(5,'press any key and enter above')
    call inkey(key)
    call wncuxy(1,6)
    read(*,*,err=5016) eqtyp(j,l),dampe(j,l)
    if(iu.eq.1)print *,'value(s)=',eqtyp(j,l),dampe(j,l)
    call sleep@(5.0)
    call wnclos(1)
    call wnopen(0,0,51,3)
    call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
    call wnouce(2,'press any key and enter above')
    call inkey(key)
    call wncuxy(1,3)
    read(*,*,err=5016) ans
    if(ans.gt.0.0)goto5016
355 write(6,*)eqtyp(j,l),dampe(j,l)
    if(iu.eq.1.and.nu.eq.1)print *,eqtyp(j,l),dampe(j,l)
```

```
      ieqtp=eqtyp(j,l)
      idam=dampe(j,l)
   43 call clear_screen@
      call wnopen(1,1,80,8)
      call wncuxy(1,1)
      goto (36,37,38,39,41),ieqtp
   36 if(nu.eq.1)read(7,*) arin(j,l),aexp(j,l)
      if(nu.eq.1)goto44
      print *,'ENTER initial air concentration, mass(or activity/ m3,'
      call wncuxy(1,2)
      print *,'(use mass or activity units used for soil); also ENTER'
      call wncuxy(1,3)
      print *,'decay(-)/growth(+) halftime (days) for exponential'
      call wnouce(4,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,5)
          read(*,*) arin(j,l),aexp(j,l)
      if(iu.eq.1)print *,'value(s)=',arin(j,l),aexp(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=43) ans
      if(ans.gt.0.0)goto43
   44 write(6,*) arin(j,l),aexp(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,arin(j,l),aexp(j,l)
      goto143
c
   37 if(nu.eq.1)read(7,*) arin(j,l),aexp(j,l)
      if(nu.eq.1)goto45
      print *,'ENTER initial air concentration, mass(or activity/ m3,'
      call wncuxy(1,2)
      print *,'(use mass or activity units used for soil); also ENTER'
      call wncuxy(1,3)
      print *,'proportionality constant(+ or -) for linear equation'
      call wnouce(4,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,5)
      read(*,*) arin(j,l),aexp(j,l)
      if(iu.eq.1)print *,'value(s)=',arin(j,l),aexp(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=43) ans
      if(ans.gt.0.0)goto43
   45 write(6,*) arin(j,l),aexp(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,arin(j,l),aexp(j,l)
```

```
      goto143
   38 if(nu.eq.1)read(7,*) arin(j,l),arfl(j,l),aexp(j,l)
      if(nu.eq.1)goto46
      print *,'ENTER maximum and minimum air concentration, mass'
      call wncuxy(1,2)
      print *,'(or activity/ m3, (use mass or activity units used for'
      call wncuxy(1,3)
      print *,'soil); also ENTER decay halftime (days) for exponential'
      call wncuxy(1,4)
      print *,'part of the equation'
      call wnouce(5,'press any key and reset')
      call inkey(key)
      call wncuxy(1,6)
      read(*,*) arin(j,l),arfl(j,l),aexp(j,l)
      if(iu.eq.1)print *,'value(s)=',arin(j,l),arfl(j,l),aexp(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=43) ans
      if(ans.gt.0.0)goto43
   46 write(6,*) arin(j,l),arfl(j,l),aexp(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,arin(j,l),arfl(j,l),aexp(j,l)
      goto143
   39 if(nu.eq.1)read(7,*) arfl(j,l),arin(j,l),arlg(j,l)
      if(nu.eq.1)goto47
      print *,'ENTER maximum and minimum amplitude air concentrations,'
      call wncuxy(1,2)
      print *,'mass(or activity/ m3, (use mass or activity units used'
      call wncuxy(1,3)
      print *,'for soil); also ENTER lag(-)/lead(+) constant(degrees)'
      call wncuxy(1,4)
      print *,'for determining time of maximum concentration'
      call wnouce(5,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,6)
      read(*,*) arfl(j,l),arin(j,l),arlg(j,l)
      if(iu.eq.1)print *,'value(s)=',arfl(j,l),arin(j,l),arlg(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=43) ans
      if(ans.gt.0.0)goto43
   47 write(6,*) arfl(j,l),arin(j,l),arlg(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,arfl(j,l),arin(j,l),arlg(j,l)
      goto143
   41 if(nu.eq.1)read(7,*) arin(j,l)
```

```
      if(nu.eq.1)goto48
      print *,'ENTER air concentration, mass(or activity)/m3, (use mass'
      call wncuxy(1,2)
      print *,'or activity units for soil'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*)arin(j,l)
      if(iu.eq.1)print *,'value=',arin(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=43) ans
      if(ans.gt.0.0)goto43
   48 write(6,*) arin(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,arin(j,l)
  143 call clear_screen@
      call wnopen(1,1,80,8)
      call wncuxy(1,1)
      goto (136,137,137,139),idam
  136 if(nu.eq.1)read(7,*) aexy(j,l)
      if(nu.eq.1)goto144
      print *,'ENTER decay(-)/growth(+) dampening/enchancement half-'
      call wncuxy(1,2)
      print *,'time (years) for exponential/ingrowth(- only) function'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
         read(*,*) aexy(j,l)
      if(iu.eq.1)print *,'value(s)=',aexy(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=143) ans
      if(ans.gt.0.0)goto143
  144 write(6,*) aexy(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,aexy(j,l)
      goto5007
c
  137 if(nu.eq.1)read(7,*) aexy(j,l)
      if(nu.eq.1)goto145
      print *,'ENTER proportionality constant (+ or _) for linear'
      call wncuxy(1,2)
      print *,'dampening/enhancement function'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
```

```
      call wncuxy(1,4)
      read(*,*) aexy(j,l)
      if(iu.eq.1)print *,'value(s)=',aexy(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=143) ans
      if(ans.gt.0.0)goto143
  145 write(6,*) aexy(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,aexy(j,l)
      goto5007
  139 if(nu.eq.1)read(7,*) aexy(j,l)
      if(nu.eq.1)goto148
      print *,'ENTER air concentration dampening constant (use 1.0 to'
      call wncuxy(1,2)
      print *,'maintain repeatable daily  values from year to year)'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*)aexy(j,l)
      if(iu.eq.1)print *,'value=',aexy(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=143) ans
      if(ans.gt.0.0)goto143
  148 write(6,*) aexy(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,aexy(j,l)
c
c enters total number of acute exposures,duration, days, of acute release,
c day modulus, and year modulus
c
 5007 if(nu.eq.1)read(7,*) cyr(j,l)
      if(nu.eq.1)goto29
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER number of acute events in simulation'
      call wncuxy(1,2)
      print *,'for contaminant ',j,' in region',l,'(0.0 cancels)'
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5007)cyr(j,l)
      if(iu.eq.1)print *,'value(s)=',cyr(j,l)
      call sleep@(5.0)
```

```
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5007) ans
      if(ans.gt.0.0)goto5007
   29 write(6,*)cyr(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,cyr(j,l)
      if(cyr(j,l).le.0.0)goto40
 5008 if(nu.eq.1)read(7,*)durc(j,l)
      if(nu.eq.1)goto31
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER duration, days, of acute event for contaminant ',j
      call wncuxy(1,2)
      print *,' in region',l
      call wncuxy(1,3)
      print *,' **entry should be duration of longest event in set'
      call wnouce(4,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,5)
      read(*,*,err=5008)durc(j,l)
      if(iu.eq.1)print *,'value(s)=',durc(j,l)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5008) ans
      if(ans.gt.0.0)goto5008
   31 write(6,*)durc(j,l)
      if(iu.eq.1.and.nu.eq.1)print *,durc(j,l)
      iyr=cyr(j,l)
      idur=durc(j,l)
c enters year(s) of acute release(s)
 5009 if(nu.eq.1)read(7,*) (yrcnt(k,j,l),k=1,iyr)
      if(nu.eq.1)goto32
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER year number(s) of ',iyr,' acute releases'
      call wncuxy(1,2)
      print *,'for contaminant ',j,' in region',l
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5009)(yrcnt(k,j,l),k=1,iyr)
      if(iu.eq.1)print *,'value(s)=',(yrcnt(k,j,l),k=1,iyr)
      call sleep@(5.0)
```

```
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5009) ans
      if(ans.gt.0.0)goto5009
   32 write(6,*)(yrcnt(k,j,l),k=1,iyr)
      if(iu.eq.1.and.nu.eq.1)print *,(yrcnt(k,j,l),k=1,iyr)
c enters initial day(s) of release(s)
 5010 if(nu.eq.1)read(7,*) (daic(k,j,l),k=1,iyr)
      if(nu.eq.1)goto33
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER julian day(s) of init. acute rel. for ',iyr
      call wncuxy(1,2)
      print *,' years of contaminant ',j,' in region ',l
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5010) (daic(k,j,l),k=1,iyr)
      if(iu.eq.1)print *,'value(s)=',(daic(k,j,l),k=1,iyr)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5010) ans
      if(ans.gt.0.0)goto5010
   33 write(6,*) (daic(k,j,l),k=1,iyr)
      if(iu.eq.1)print *, (daic(k,j,l),k=1,iyr)
      do 30 ll=1,iyr
c enters particulate concentrations for duration of acute event(s)
 5011 if(nu.eq.1)read(7,*) (amac(ll,k,j,l),k=1,idur)
      if(nu.eq.1)goto34
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER particulate conc(s) for ',idur,' days in'
      call wncuxy(1,2)
      print *,' year ',ll,'of contaminant ',j,' in region',l
      call wncuxy(1,3)
      print *,'conc(s) units same as for chronic releases above'
      call wnouce(4,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,5)
      read(*,*,err=5011) (amac(ll,k,j,l),k=1,idur)
      if(iu.eq.1)print *,'value(s)=',(amac(ll,k,j,l),k=1,idur)
      call sleep@(5.0)
      call wnclos(1)
```

```
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5011) ans
      if(ans.gt.0.0)goto5011
   34 write(6,*) (amac(ll,k,j,l),k=1,idur)
      if(iu.eq.1)print *, (amac(ll,k,j,l),k=1,idur)
c
c enters gaseous concentrations for duration of acute event(s)
 5012 if(nu.eq.1)read(7,*) (amag(ll,k,j,l),k=1,idur)
      if(nu.eq.1)goto35
      call clear_screen@
      call wnopen(1,1,80,10)
      call wncuxy(1,1)
      print *,'ENTER gaseous conc(s) for ',idur,' days in'
      call wncuxy(1,2)
      print *,' year ',ll,'of contaminant ',j,' in region',l
      call wncuxy(1,3)
      print *,'conc(s) units same as for chronic releases above'
      call wnouce(4,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,5)
      read(*,*,err=5012) (amag(ll,k,j,l),k=1,idur)
      if(iu.eq.1)print *,'value(s)',(amag(ll,k,j,l),k=1,idur)
      call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5012) ans
      if(ans.gt.0.0)goto5012
   35 write(6,*) (amag(ll,k,j,l),k=1,idur)
      if(iu.eq.1.and.nu.eq.1)print *, (amag(ll,k,j,l),k=1,idur)
c
   30 continue
   40 continue
c initializes year counter
      icyr(j,l)=1
   50 continue
c ** end contaminant loop ***********************************************
   60 continue
c ** end regional loop
      if(nu.eq.1)goto5500
 5013 call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('*********************                    ')
      call wnoust('END OF INPUT IN AIRAC                      ')
      call wnoust('*********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
```

```
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for AIRAC?          ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now          ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5013) ans
      if(ans)5502,5500,5502
 5502 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-111111.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      call wnclos(1)
      goto5015
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
      return
   70 continue
      l=lx
c
c ** contaminant loop ***********************************************
      do 120 j=1,inz
c contaminant selector
      if (arcz(j)) 126,120,126
c number of acute events
  126 iyr=cyr(j,l)
c contaminant air concentrations set to 0.0
      airca(j,l)=0.0
      aircg(j,l)=0.0
      if (aiii-1.0) 90,91,90
   91 icdy(j,l)=1
c acute event selector
      if (cyr(j,l)) 81,80,81
   81 jcyr=icyr(j,l)
c acute event tracking
      if (ai1-yrcnt(jcyr,j,l)) 83,82,83
   82 icyr(j,l)=icyr(j,l)+1
   83 if (icyr(j,l).gt.iyr) icyr(j,l)=iyr
   80 continue
   90 if (cyr(j,l)) 101,100,101
  101 if (ai1-yrcnt(jcyr,j,l)) 100,102,100
  102 if (aiii.lt.daic(jcyr,j,l)) go to 100
      if (aiii.ge.daic(jcyr,j,l)+durc(j,l)) go to 100
      jcdy=icdy(j,l)
c sets contaminant acute air concentrations to daily values
      airca(j,l)=amac(jcyr,jcdy,j,l)
      aircg(j,l)=amag(jcyr,jcdy,j,l)
      icdy(j,l)=icdy(j,l)+1
```

```
c selects chronic air releases
 100 if (ai1.lt.chrsy(j,l).or.ai1.gt.chrey(j,l)) go to 110
     if (aiii.lt.chrsd(j,l).or.aiii.gt.chred(j,l)) go to 110
c
c  ct may be replaced by  a chronic release function: airnf(ai1,aiii) or
c     by a constant; same for ctg
c
      syr=chrsy(j,l)
      sdy=chrsd(j,l)
      eyr=chrey(j,l)
      edy=chred(j,l)
      ain=arin(j,l)
      arf=arfl(j,l)
      axp=aexp(j,l)
      axpy=aexy(j,l)
      ieqt=eqtyp(j,l)
      idam=dampe(j,l)
      alr=arlg(j,l)
      dn=dnn(l)
      ct(j,l)=0.0
      ctg(j,l)=0.0
      if(chrac(j,l).gt.0.0)
     1ct(j,l)=airfn(ai1,aiii,syr,sdy,eyr,edy,arf,arfyr,axp,axpy,ieqt,
     2idam,ain,ainy,alr,dn)
      if(chrag(j,l).gt.0.0)
     1ctg(j,l)=airfg(ai1,aiii,syr,sdy,eyr,edy,arf,arfyr,axp,axpy,ieqt,
     2idam,ain,ainy,alr,dn)
      airca(j,l)=airca(j,l)+ct(j,l)
      aircg(j,l)=aircg(j,l)+ctg(j,l)
c
 110 continue
     jz=j
 120 continue
c ** end of contaminant loop ******************************************
      return
      end
      function airfn(ai1,aiii,syr,sdy,eyr,edy,arf,arfyr,axp,axpy,ieqt,
     1idam,ain,ainy,alr,dn)
      goto (1,2,3,4,5), ieqt
    1 airfn=ain*exp(0.693*(aiii-sdy)/axp)
      goto6
    2 airfn=ain+axp*(aiii-sdy)
      goto6
    3 if(axp.lt.0.0)airfn=arf*(1.0-exp(0.693*(aiii-sdy)/axp))
      if(axp.gt.0.0)airfn=arf*(1.0-(1.0-exp(-0.693*(aiii-sdy)/axp)))
      goto6
    4 airfn=0.5*(arf-ain)*(sin(2.0*3.1415*(dn-alr)/(edy-sdy))+1.)+ain
      goto6
    5 airfn=ain
    6 goto (7,8,9,10), idam
    7 airfn=airfn*exp((0.693/axpy)*(ai1-syr))
      goto11
    8 airfnp=airfn
      airfn=airfn+airfn*axpy*(ai1-syr)
```

```
  goto11
 9 if(axpy.lt.0.0)airfn=airfn*(1.0-exp(0.693*(ai1-syr)/axpy))
  if(axpy.gt.0.0)airfn=airfn*(1.0-(1.0-exp(-0.693*(ai1-syr)/axpy)))
  goto11
10 airfn=airfn+airfn*axpy
11 continue
  end
  function airfg(ai1,aiii,syr,sdy,eyr,edy,arf,arfyr,axp,axpy,ieqt,
 1idam,ain,ainy,alr,dn)
  goto (11,12,13,14,15), ieqt
11 airfg=ain*exp((0.693/axp)*(aiii-sdy))
  goto6
12 airfg=ain+axp*(aiii-sdy)
  goto6
13 if(axp.lt.0.0)airfg=arf*(1.0-exp(0.693*(aiii-sdy)/axp))
  if(axp.gt.0.0)airfg=arf*(1.0-(1.0-exp(-0.693*(aiii-sdy)/axp)))
  goto6
14 airfg=0.5*(arf-ain)*(sin(2.0*3.1415*(dn-alr)/(edy-sdy))+1.)+ain
  goto6
15 airfg=ain
 6 goto (17,18,19,20), idam
17 airfg=airfg*exp((0.693/axpy)*(ai1-syr))
  goto21
18 airfg=airfg*axpy*(ai1-syr)
  goto21
19 if(axpy.lt.0.0)airfg=airfg*(1.0-exp(0.693*(ai1-syr)/axpy))
  if(axpy.gt.0.0)airfg=airfg*(1.0-(1.0-exp(-0.693*ai1-syr)/axpy))
  goto21
20 airfg=airfg*axpy
21 continue
  end
```

14.3. ANIMAL

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine animal (ii,in,lx,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
      real m5, m6
c
      common/pa/ absw(120,ko), adia(120,ko), al4(m), al5(m),
     1bddt(120,ko),c66(m),degg(120,ko),egg(120,ko),
     2hatce(120,ko), hatch(120,ko),hatmp(120,ko),hattmp(120,ko),
     3ondv(30,ko), onyph(30,ko), pdiape(120,ko), thz(24), tmpday(30,ko)
     4,watcnr(ko,m,n),yadlt(30,ko),yndv(30,ko),
     5 ynyph(30,ko),hemlos(ko,m,n),bltgt(ko,m),fdtmp(ko),tpsm(ko),
     6 dvtmp(ko),adlt(2,ko),iegg(ko),iegd(ko),iegh(ko),eggs,deggs,
     7 hatchs, pdiaps,aha(ko),yadlts,ynyphs,onyphs,eglay,degday(ko),
     8 colday(ko),diaday(ko),eggmrt(ko),aii,summrt(ko,2)
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),wt1(kr,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      common/pw/ gaslos(ko),urnlos(ko),wtbn(kr,ko),
     1wtkd(kr,ko),wtlv(kr,ko),wtms(kr,ko),wtrs(kr,ko),wtlu(kr,ko)
c
      common/pj/ awt(m), cdrp(ks,n), dsrt(m,n), hbar(ks,n), ph(i,ks,n),
     1 rock(i,ks,n), sand(i,ks,n), silt(i,ks,n), soilf(i,n), tim(i,n),
     2 w1bar(ks,n), watcn(m,n), wtf(n), yaz(m,n), ysrt(m,n),
     3 yup(m,n), z1dl(i,ks,m,n)
c
      common/ha/ popt(ko),tqx0(kr,2,ko),jage(ko),aiif(ko,2),ifm(kr),
     1vurlfj(ko),vmljk(kr,ko),popp(2,ko),ikn(2,ko,8),sfpop(ko),
     2gst(ko,2),gstn(ko),gestp(ko),aiiir(ko,2),agem(kr,ko),cagm(ko,2),
     3cntk(2),brts(ko,2,2),frpr(2,ko),agcor(ko),tpppt(ko),sfd(8,ko),
     4tppcs(ko),tppps(ko),ca1(ko),clf(ko),lagf(ko),lua(ko),offsp(ko,3,2)
     5,affsp(ko,3),rat12(ko),tqp(kr,ko),tqs(kr,ko),vpelt(kr,ko,m),
     6vgut(kr,ko,m),cnmus(kr,ko,m),cnliv(kr,ko,m),cnkid(kr,ko,m),
     7cnfat(kr,ko,m),cnlun(kr,ko,m),cnbon(kr,ko,m),cnres(kr,ko,m),
     8cnblo(kr,ko,m),cnbod(kr,ko,m),cngut(kr,ko,m),cnpel(kr,ko,m),
     9vpln(kr,ko,m),vsol(kr,ko,m),vlug(kr,ko,m),cnpln(kr,ko,m),
     1cnsol(kr,ko,m),cnlug(kr,ko,m),rtpls(ko),rtmus(ko),rtkid(ko),
     2rtliv(ko),rtgut(ko),rthrt(ko),rtlun(ko),rtspl(ko),rtmar(ko),
```

```
     3rtfat(ko),rtres(ko),cbp(kr,ko,m),cbf(ko,m),cbw(ko,m),tacres(n),
     4vplt(kr,ko),c31plt(kr,ko,m),cnmlk(kr,ko,m),cnurnc(kr,ko,m),
     5cnfec(kr,ko,m),cnurn(kr,ko,m)
      dimension poppi(2,ko),pop2r(ko),afmt(ko),gsst(ko),cblc(ko),
     1cntki(ko),fracs(ko),gsfc(ko),vmkc(ko)
c
c   animal subroutine simulates animal and insect growth, feeding/food
c   schedules, and contaminant metabolism.
c
c excludes all but first access
c
     if (rdidx.gt.0.0) go to 180
c
c excludes all but one access  for all regions(lx) during first access
c
     if (lx.gt.1) return
     nuu=nu
     if(nu.eq.0)goto5050
     rewind 7
 5702 continue
     read(7,*,err=5702,end=5555)acheck
     if(acheck-222222.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','magenta')
     call wnclos(1)
     call wnopen(0,0,51,4)
     call wnoust('******************************                    ')
     call wnoust(' YOU ARE IN SUBROUTINE ANIMAL                    ')
     call wnoust('******************************                    ')
     call wnouce(4,'press any key')
     call inkey(key)
     call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
     call wnopen(0,0,51,8)
     call wnoust('*****************************************************')
     call wnoust('1=file input, 0=keyboard input: you have selected  ')
     call wnoust('                                                   ')
     call wnoust('ENTER yes=1, no=0                                  ')
     call wnoust('*****************************************************')
     call wncuxy(1,3)
     print *,'mode=',nu,'; want to reverse temporarily?'
     call wnouce(7,'press any key and enter above')
     call inkey(key)
     call wncuxy(1,8)
     read(*,*,err=5700)ans
     if(ans.le.0.0)goto5701
     nux=nu
     if(nux.eq.0)nu=1
     if(nux.eq.1)nu=0
5701 call wnclos(1)
     if(nu.eq.1)goto5706
     call wnopen(0,0,51,4)
```

```
      call wnoust('ENTER subroutine code: 222222.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
      call wnclos(1)
      call clear_screen@
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck

c
 5053 print *,'ENTER number of animal types simulated including insects'
      print *,'if simulated'
      if(nu.eq.0)read(*,*,err=5053) nan
      if(nu.eq.1)read(7,*) nan
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5053) ans
      if(ans.gt.0.and.nu.eq.0)goto5053
      write(6,*) nan
      if(iu.eq.1)print *,nan
c
 5001 print *,'ENTER mature weight, kg, for',nan,'animal types: set'
      print *,'mature weight of insect to 1.0: if user chooses to'
      print *,'enter non-default transport coefficients then enter,'
      print *,'the negative of the mature weight of that animal type'
      print *,'if humans simulated, then rr(1) must be beef cows, or'
      print *,'rr(2) must be dairy cows; one or both types required'
c
      print *,'ex..cows=550., deer=114., elk=360., coyote=16.,'
      print *,'deer mouse=.022, rabbit=1.3, kangaroo rat=.2, etc.'
c
      if(nu.eq.0)read(*,*,err=5001) (rr(l),l=1,nan)
      if(nu.eq.1)read(7,*) (rr(l),l=1,nan)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write(6,*) (rr(l),l=1,nan)
      if(iu.eq.1)print *, (rr(l),l=1,nan)
c
c
c ** region loop ******************************************************
      do 20 lz=1,ipx
      print *,'in region',lz
```

```
c ** contaminant loop ************************************************
c
      do 10 k=1,inz
c
c concentration in drinking water for each region(lx), ppm, read in
c for all animals simulated (nutrients can be excluded or included)
c
      print *,'for contaminant',k
 5002 print *,'ENTER contaminant concentration in drinking water:'
      print *,'ppm,pci..uci ,for',nan,'animal types in region',lz
      if(nu.eq.0)read(*,*,err=5002) (watcnr(l,k,lz),l=1,nan)
      if(nu.eq.1)read(7,*) (watcnr(l,k,lz),l=1,nan)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*) (watcnr(l,k,lz),l=1,nan)
      if(iu.eq.1)print *, (watcnr(l,k,lz),l=1,nan)
   10 continue
c ** end contaminant loop ****************************************
   20 continue
c ** end region loop *************************************************
c
c enter animal number of changed feed schedules
c values used to vary plant types and utilization during simulation
c
 5005 print *,'ENTER number of feed schedules for',nan,'zoon groups'
      print *,'default=1.0; values used vary plant types and diet'
      print *,'utilization fractions'
c
      if(nu.eq.0)read(*,*,err=5005) (rzoon(l),l=1,nan)
      if(nu.eq.1)read(7,*) (rzoon(l),l=1,nan)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*) (rzoon(l),l=1,nan)
      if(iu.eq.1)print *, (rzoon(l),l=1,nan)
c
c enter mean life span of animal unit, years; insects set to 0.0
c
 5006 print *,'ENTER life-span of animal, years, for',nan,'zoon groups'
      print *,'ex.. cow=14.93,coyote=4.93,rabbit=3.94,deer mouse=1.64'
      print *,'grasshopper mouse=3.28,squirrel=2.96,kangaroo rat=2.2'
      print *,'set insect life-span to 0.0'
      if(nu.eq.0)read(*,*,err=5006) (alsp(l),l=1,nan)
      if(nu.eq.1)read(7,*) (alsp(l),l=1,nan)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*) (alsp(l),l=1,nan)
      if(iu.eq.1)print *, (alsp(l),l=1,nan)
c
      insct=0
      iian=0
c ** zoon loop ******************************************************
```

```
      do 100 j=1,nan
c ** contaminant loop ************************************************
      do32k=1,inz
c initializes contaminant transport from parent to fetus and from
c milk to weaning newborn
      cbf(j,k)=0.0
      cbw(j,k)=0.0
      ian=atno(k)
      if(ian.ge.101)goto32
      if(k.gt.1.and.k.lt.7)goto32
 5055 print *,'ENTER soft tissue, bone, and endosteal bone radiation'
      print *,'deposition, MeV-Rem/dis (3 values) for zoon',j,'and'
      print *,'nuclide',k,';enter 0.0 when not applicable'
      if(nu.eq.0)read(*,*,err=5055) evs(j,k),evb(j,k),eve(j,k)
      if(nu.eq.1)read(7,*) evs(j,k),evb(j,k),eve(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5055) ans
      if(ans.gt.0.and.nu.eq.0)goto5055
      write(6,*) evs(j,k),evb(j,k),eve(j,k)
      if(iu.eq.1)print *, evs(j,k),evb(j,k),eve(j,k)
   32 continue
c************************ end contaminant loop ************************
c
c ** region loop ******************************************************
      do 40 lz=1,ipx
c
c ** contaminant loop ************************************************
      do 30 k=1,inz
c sets drinking water concentration(per liter), and halflife (years)
c of contaminants
c converts drinking water concentrations to g/l
c
      if (hflif(k)) 30,31,30
   31 watcnr(j,k,lz)=watcnr(j,k,lz)/1.0e06
   30 continue
c ** end contaminant loop ********************************************
   40 continue
c ** end of region loop ***********************************************
c input utilized in EFFECTS model
c
 5099 print *,'for accessing of the EFFECTS subroutine:'
      print *,'ENTER number of effects years simulated in this'
      print *,'run; a value of 0.0 will not allow access to'
      print *,'the EFFECTS subroutine'
      if(nu.eq.0)read(*,*,err=5099)ydos(j)
      if(nu.eq.1)read(7,*)ydos(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5099) ans
      if(ans.gt.0.and.nu.eq.0)goto5099
      write(6,*)ydos(j)
      if(iu.eq.1)print *,ydos(j)
c
 5007 print *,'for animal',j,'ENTER meat frac. consumed from area'
c
```

```
      if(nu.eq.0)read(*,*,err=5007)p5(j)
      if(nu.eq.1)read(7,*)p5(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*)p5(j)
      if(iu.eq.1)print *,p5(j)
c
 6007 print *,'for animal',j,'ENTER population size, fraction of'
      print *,'population that are females, and population structure'
      print *,'type: 1=none,2=parabolic,3=exponential,4=hyperbolic'
      print *,'a value of 1 indicates structure will be entered by user'
      print *,'or mortality factors not required; enter 3 parameters'
c
      if(nu.eq.0)read(*,*,err=6007) zoo(j),sxrt(j),popt(j)
      if(nu.eq.1)read(7,*)zoo(j),sxrt(j),popt(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6007) ans
      if(ans.gt.0.and.nu.eq.0)goto6007
      write(6,*)zoo(j),sxrt(j),popt(j)
      if(iu.eq.1)print *,zoo(j),sxrt(j),popt(j)
c
 6008 print *,'for animal',j,'ENTER frac. of area utilized'
c
      if(nu.eq.0)read(*,*,err=6008)aa2(j)
      if(nu.eq.1)read(7,*)aa2(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6008) ans
      if(ans.gt.0.and.nu.eq.0)goto6008
      write(6,*)aa2(j)
      if(iu.eq.1)print *,aa2(j)
c
 6009 print *,'for animal',j,'ENTER frac. of suppl. feed in diet'
      print *,'that is drived from the study area'
c
      if(nu.eq.0)read(*,*,err=6009)fsupf(j)
      if(nu.eq.1)read(7,*)fsupf(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6009) ans
      if(ans.gt.0.and.nu.eq.0)goto6009
      write(6,*)fsupf(j)
      if(iu.eq.1)print *,fsupf(j)
c
 6011 print *,'for animal',j,'ENTER the number of yearly feeding'
      print *,'intervals if you want to change animal diets for'
      print *,'different yearly time intervals..1-10,11-20 yrs...etc.'
c
      if(nu.eq.0)read(*,*,err=6011)eatin(j)
      if(nu.eq.1)read(7,*)eatin(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6011) ans
      if(ans.gt.0.and.nu.eq.0)goto6011
      write(6,*)eatin(j)
      if(iu.eq.1)print *,eatin(j)
```

```
c
 6012 print *,'for animal',j,'ENTER the number of daily feeding'
      print *,'intervals if you want to change animal diets for'
      print *,'different daily time intervals during a given yearly'
      print *,'time interval...1-74, 75-365 julian dates..etc..'
c
      if(nu.eq.0)read(*,*,err=6012)fodin(j)
      if(nu.eq.1)read(7,*)fodin(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6012) ans
      if(ans.gt.0.and.nu.eq.0)goto6012
      write(6,*)fodin(j)
      if(iu.eq.1)print *,fodin(j)
c
 6013 print *,'for animal',j,'ENTER oldest age group, years'
      print *,'may be set larger or equal to life-span (ALSP) as'
      print *,'complete years ,ex.. for 3.4 life-span enter 4.0'
      print *,'enter 0.0 for default to 8 yr life-span (available'
      print *,'for beef and dairy cows only);for insects enter 1.0'
      print *,'also ENTER when gestation period is initiated:'
      print *,'1=Spring, 2=Fall; also enter number of gestation'
      print *,'periods/year,and fraction of offspring produced in'
      print *,'first reproductive period (equals 1.0 if only one'
      print *,'reproductive cycle is specified)'
      print *,'4 entries required; set last 3 equal 0.0 for insects'
c
      if(nu.eq.0)read(*,*,err=6013)agea(j),gst(j,1),gstn(j),frpr(1,j)
      if(nu.eq.1)read(7,*)agea(j),gst(j,1),gstn(j),frpr(1,j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=6013) ans
      if(ans.gt.0.and.nu.eq.0)goto6013
      write(6,*)agea(j),gst(j,1),gstn(j),frpr(1,j)
      if(iu.eq.1)print *,agea(j),gst(j,1),gstn(j),frpr(1,j)
c
      if(gst(j,1).le.1.0)gst(j,1)=92.0
      if(gst(j,1).lt.92.0)gst(j,1)=270.0
      frpr(2,j)=1.0-frpr(1,j)
c
c integers values of above parameters
      iiet=eatin(j)*2.0
      iieth=iiet/2
      iifd=fodin(j)*2.0
      iifdh=iifd/2
      iage=agea(j)
c enter yearly(YRFED) and daily(ONFOD)  feeding intervals (2 values/interval)
c
 5008 print *,'ENTER',iieth,'yearly feed interval(s),2 values/interval'
      print *,'required: example: 1-10,11-20,21-50:,..for 3 yearly time'
      print *,'intervals specified by the user over a 50 year period..'
c
      if(nu.eq.0)read(*,*,err=5008) (yrfed(j,l),l=1,iiet)
      if(nu.eq.1)read(7,*) (yrfed(j,l),l=1,iiet)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
```

```
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*) (yrfed(j,l),l=1,iiet)
      if(iu.eq.1)print *, (yrfed(j,l),l=1,iiet)
 5009 print *,'ENTER',iifdh,'daily feed interval(s),2 values/interval'
      print *,'required: example: 1-74,75-230,76-365 shows three'
      print *,'different feed schedules during a given year of a'
      print *,'specified yearly time interval'
c
      if(nu.eq.0)read(*,*,err=5009) (onfod(j,l),l=1,iifd)
      if(nu.eq.1)read(7,*) (onfod(j,l),l=1,iifd)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,*) (onfod(j,l),l=1,iifd)
      if(iu.eq.1)print *, (onfod(j,l),l=1,iifd)
c enters fraction of range grazed in regions
c
c ** region loop ***********************************************************
      do 60 lz=1,ipx
      print *,'for region',lz
c rnfrc= reset value for regional utilization fraction(RFRZ) by
c daily food schedule
 5011 print *,'ENTER',iifdh,'reset frac. of range grazed by daily'
      print *,'food schedule;2 duplicate values/interval. Example:'
      print *,'1-1,.5-.5 for 2 time intervals of a given feed schedule'
c
      if(nu.eq.0)read(*,*,err=5011) (rnfrc(j,l,lz),l=1,iifd)
      if(nu.eq.1)read(7,*) (rnfrc(j,l,lz),l=1,iifd)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*) (rnfrc(j,l,lz),l=1,iifd)
      if(iu.eq.1)print *, (rnfrc(j,l,lz),l=1,iifd)
c fraction of diet for animals consumed,j refers to consumer, l refers
c to prey...etc.
 5014 print *,'for region',lz
      print *,'ENTER frac. of diet of prey for consumer',j
      print *,nan,'prey entries required; each animal simulated is'
      print *,'represented including consumer. Enter 0.0 if specific'
      print *,'animal type is not being used as prey by consumer;'
      print *,'be careful to avoid cannabilism'
      if(nu.eq.0)read(*,*,err=5014) (autzd(j,l,lz),l=1,nan)
      if(nu.eq.1)read(7,*) (autzd(j,l,lz),l=1,nan)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*) (autzd(j,l,lz),l=1,nan)
      if(iu.eq.1)print *, (autzd(j,l,lz),l=1,nan)
      if(lz.gt.1)goto5016
c
c for each year's-feeding interval ,no. of sub-intervals are entered
c
 5015 print *,'ENTER',iieth,' number of feeding sub-intervals for '
      print *,'for animal',j, '; in duplicate pairs: ex. 3 3 1 1...'
```

```
      if(nu.eq.0)read(*,*,err=5015)(eatrec(j,l),l=1,iiet)
      if(nu.eq.1)read(7,*)(eatrec(j,l),l=1,iiet)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*)(eatrec(j,l),l=1,iiet)
      if(iu.eq.1)print *,(eatrec(j,l),l=1,iiet)
c
c PLTIND=number of different feeding schedules..1,2,3,..with same or
c different plants to be utilized
 5016 kherb=rzoon(j)
      print *,'ENTER ',kherb,'number of different changed feeding'
      print *,'schedules;enter 1.0,2.0,.....,in region',lz
      if(nu.eq.0)read(*,*,err=5016)(pltind(j,l,lz),l=1,kherb)
      if(nu.eq.1)read(7,*)(pltind(j,l,lz),l=1,kherb)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*)(pltind(j,l,lz),l=1,kherb)
      if(iu.eq.1)print *,(pltind(j,l,lz),l=1,kherb)
c
c enters up to 8 different plants(PLTUSD), preferences and types of
c feed(PLTRNK), and fraction of plant in diet(PLTF) mix
c
c ** feed schedules loop ***********************************************
      do 50 l=1,kherb
      if(lz.gt.1)goto5017
 5010 print *,'ENTER number plant types used for schedule',l
      if(nu.eq.0)read(*,*,err=5010) aisch
      if(nu.eq.1)read(7,*) aisch
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*) aisch
      if(iu.eq.1)print *, aisch
 5017 isch=aisch
      print *,'for region',lz
      print *,'for each changed feed schedule',l,'and animal',j
      print *,'ENTER',isch,' plant types; up to 8 possible'
      if(nu.eq.0)read(*,*,err=5017)(pltusd(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltusd(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,*)(pltusd(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltusd(j,k,l,lz),k=1,isch)
 5018 print *,'for region',lz
      print *,'ENTER',isch,' feeding preference(1.,2.,3.) and/or feed'
      print *,'types (3.,4.,5.,6.,7.) for each plant type;'
      if(nu.eq.0)read(*,*,err=5018)(pltrnk(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltrnk(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
```

```
      write(6,*)(pltrnk(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltrnk(j,k,l,lz),k=1,isch)
 5019 print *,'for region',lz
      print *,'ENTER',isch,'fractions of each plant type in diet:'
      if(nu.eq.0)read(*,*,err=5019)(pltf(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltf(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
      write(6,*)(pltf(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltf(j,k,l,lz),k=1,isch)
c
      do51k=1,isch
      pusd=pltusd(j,k,l,lz)
      do52ll=1,ipl
      if(pusd-pltid(ll,lz)) 52,53,52
   52 continue
      goto50
   53 prnk=pltrnk(j,k,l,lz)
      pfrac=pltf(j,k,l,lz)
      putzd(j,ll,lz)=prnk
      putfrc(j,ll,lz)=pfrac
   51 continue
   50 continue
c ** end feed schedules loop ******************************************
   60 continue
c ** end region loop ****************************************************
c
c a negative value for RR signifiies user specification of transport
c coefficients
c
c*** perfusion parameters for standard man ****
c
      wtgrh=70000.
c weight of blood/hemolymph, kg
      wtblh=5.5e-05*wtgrh**0.99
c weight of bone/exoskeleton,kg
      wtbnh=5.5e-05*wtgrh
c weight of kidneys/malphigian tubules, kg
      wtkdh=2.12e-05*wtgrh**0.85
c weight of liver, kg
      wtlvh=8.2e-05*wtgrh**0.87
c weight of lung/trachae
      wtluh=1.24e-05*wtgrh**0.99
c weight of muscle, kg
      wtmsh=4.5e-04*wtgrh
c fat weight, kg
      fatbh=1.1e-04*wtgrh
c inhalation rate,m3/day
      rsrt(l,j)=1.4*amx(l,j)
c weight of residual soft tissue, kg
      wtrsh=(wtgrh/1000.)-wtblh-wtbnh-wtkdh-wtlvh-wtmsh-fatbh
c
c *** end of human organ parameters ****
```

```
c
      if (rr(j).ge.0.0) go to 86
c
 5024 print *,'ENTER total deposition fraction for n-p region of lung'
      if(nu.eq.0)read(*,*,err=5024)plcm1(j)
      if(nu.eq.1)read(7,*)plcm1(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*)plcm1(j)
      if(iu.eq.1)print *,plcm1(j)
 5025 print *,'ENTER total deposition fraction for t-b region of lung'
      if(nu.eq.0)read(*,*,err=5025)plcm2(j)
      if(nu.eq.1)read(7,*)plcm2(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5025) ans
      if(ans.gt.0.and.nu.eq.0)goto5025
      write(6,*)plcm2(j)
      if(iu.eq.1)print *,plcm2(j)
 5026 print *,'ENTER total deposition fraction for p region of lung'
      if(nu.eq.0)read(*,*,err=5026)plcm3(j)
      if(nu.eq.1)read(7,*)plcm3(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5026) ans
      if(ans.gt.0.and.nu.eq.0)goto5026
      write(6,*)plcm3(j)
      if(iu.eq.1)print *,plcm3(j)
c
c ** contaminant loop ***************************************************
   86 do 90 k=1,inz
      ian=atno(k)
c atomic number of contaminant, N,P,K,Pu,...etc
c uses default values for nutrients
      if(k.gt.1.and.k.lt.7)go to 80
 5020 print *,'for animal type',j,' and contaminant',k
      print *,'ENTER gut to blood transport coefficient; a value of'
      print *,'0.0 will default to UCRL estimate if contaminant is a'
      print *,'contaminant; if not will default to 0.5; enter value now'
c
      if(nu.eq.0)read(*,*,err=5020)gtbl(j,k)
      if(nu.eq.1)read(7,*)gtbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
      write(6,*)gtbl(j,k)
      if(iu.eq.1)print *,gtbl(j,k)
      if(ian.le.100.and.gtbl(j,k).le.0.0)gtbl(j,k)=fbwbz(ian)
      if(ian.gt.100.and.gtbl(j,k).le.0.0)gtbl(j,k)=0.5
c
 5027 print *,'ENTER blood to urine daily fractional transport; a'
      print *,'default value of 0.0 will set both nuclides and organic'
      print *,'contaminants to a 1 day biological half-lifes; enter now'
      if(nu.eq.0)read(*,*,err=5027)bltur(j,k)
      if(nu.eq.1)read(7,*)bltur(j,k)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5027) ans
      if(ans.gt.0.and.nu.eq.0)goto5027
      write(6,*)bltur(j,k)
      if(iu.eq.1)print *,bltur(j,k)
      if(bltur(j,k).le.0.0)bltur(j,k)=0.693
c
 5028 print *,'ENTER blood to milk daily fractional transport;a default'
      print *,'default value of 0.0 will set contaminants to the gut'
      print *,'absorption fraction  Enter value now'
      if(nu.eq.0)read(*,*,err=5028) bltmk(j,k)
      if(nu.eq.1)read(7,*)bltmk(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5028) ans
      if(ans.gt.0.and.nu.eq.0)goto5028
      write(6,*)bltmk(j,k)
      if(iu.eq.1)print *,bltmk(j,k)
      if(bltmk(j,k).le.0.0)bltmk(j,k)=gtbl(j,k)
c
 5023 print *,'ENTER pulmonary to blood daily fractional transport'
      print *,'from particulates; an entered value of 0.0 will default'
      print *,'to 0.1/day for all contaminants; enter value now'
      if(nu.eq.0)read(*,*,err=5023)pltbl(j,k)
      if(nu.eq.1)read(7,*)pltbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5023) ans
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*)pltbl(j,k)
      if(iu.eq.1)print *,pltbl(j,k)
      if(pltbl(j,k).le.0.0)pltbl(j,k)=0.1
c
 5032 print *,'ENTER alveoli-to-blood from gas phase partition'
      print *,'coefficient; if contaminant is non-volatile, enter'
      print *,'a value of -1.0; if user enters 0.0, then the'
      print *,'partition coefficient will default to 10.0 for volatile'
      print *,'contaminants; enter now'
c
      if(nu.eq.0)read(*,*,err=5032)pltblg(j,k)
      if(nu.eq.1)read(7,*)pltblg(j,k)
      iblg=pltblg(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5032) ans
      if(ans.gt.0.and.nu.eq.0)goto5032
      write(6,*)pltblg(j,k)
      if(iu.eq.1)print *,pltblg(j,k)
      if(iblg.eq.0)pltblg(j,k)=10.
      if(pltblg(j,k).le.-1.0)pltblg(j,k)=0.0
c
 5021 print *,'ENTER pulmonary to gut daily transport fraction from'
      print *,'particulates in bronchus region; a value of 0.0 will'
      print *,'default to a biological half-life of 1 day; enter now'
c
      if(nu.eq.0)read(*,*,err=5021)pltg1(j,k)
      if(nu.eq.1)read(7,*)pltg1(j,k)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*)pltg1(j,k)
      if(iu.eq.1)print *,pltg1(j,k)
      if(pltg1(j,k).le.0.0)pltg1(j,k)=0.693
c
 5022 print *,'ENTER pulmonary to gut daily transport fraction from'
      print *,'particulates in bronchiole region; a value of 0.0 will'
      print *,'default to a biological half-life of 10 days; enter now'
c
      if(nu.eq.0)read(*,*,err=5022)pltg2(j,k)
      if(nu.eq.1)read(7,*)pltg2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5022) ans
      if(ans.gt.0.and.nu.eq.0)goto5022
      write(6,*)pltg2(j,k)
      if(iu.eq.1)print *,pltg2(j,k)
      if(pltg2(j,k).le.0.0)pltg2(j,k)=0.0693
c
 5040 print *,'ENTER fat to blood daily transport fraction or partition'
      print *,'coefficent for contaminant',k,'; enter a positive number'
      print *,'if value is a transport coefficient, or the negative of'
      print *,'the value if you are entering a partition coefficient'
      print *,' a value of 0.0 will default to a 4 day half-life; a'
      print *,'value of -1000000.0 will default to a partition'
      print *,'coefficient of 10; enter value now'
c
      if(nu.eq.0)read(*,*,err=5040) fttbl(j,k)
      if(nu.eq.1)read(7,*)fttbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5040) ans
      if(ans.gt.0.and.nu.eq.0)goto5040
      write(6,*)fttbl(j,k)
      if(iu.eq.1)print *,fttbl(j,k)
c
      ftb=fttbl(j,k)
      if(ftb.le.-1000000.0)fttbl(j,k)=-10.0
      if(ftb.gt.0.0)fttbl(j,k)=volpls*flofat/(flopls*volfat*ftb)
      if(ftb.lt.0.0)fttbl(j,k)=-fttbl(j,k)
      if(fttbl(j,k).le.0.0)fttbl(j,k)=wtblh/(fatbh*.173)
c
c uses default values for contaminants
      if (rr(j).ge.0.0) go to 80
c
      print *,'The following entries are transport coefficients,'
      print *,'fraction/day, that are user defined. Nutrients'
      print *,'(nitrates,phosphates,potassium..2-6) excluded'
      print *,'for animal',j,'and contaminant',k
c
c enters transport coefficients
c
 5030 print *,'ENTER bone to blood(1) daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
```

```
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5030)bntbl1(j,k)
      if(nu.eq.1)read(7,*)bntbl1(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5030) ans
      if(ans.gt.0.and.nu.eq.0)goto5030
      write(6,*)bntbl1(j,k)
      if(iu.eq.1)print *,bntbl1(j,k)
      bn1=bntbl1(j,k)
      if(bn1.gt.0.0)bntbl1(j,k)=wtblh/(wtbnh*bn1)
      if(bn1.lt.0.0)bntbl1(j,k)=-bn1
c
5031 print *,'ENTER bone to blood(2) daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5031) bntbl2(j,k)
      if(nu.eq.1)read(7,*) bntbl2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5031) ans
      if(ans.gt.0.and.nu.eq.0)goto5031
      write(6,*) bntbl2(j,k)
      if(iu.eq.1)print *,bntbl2(j,k)
      bn2=bntbl2(j,k)
      if(bn2.gt.0.0)bntbl2(j,k)=wtblh/(wtbnh*bn2)
      if(bn1.lt.0.0)bntbl2(j,k)=-bn2
c
5033 print *,'ENTER liver to blood daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5033)lvtbl(j,k)
      if(nu.eq.1)read(7,*)lvtbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5033) ans
      if(ans.gt.0.and.nu.eq.0)goto5033
      write(6,*)lvtbl(j,k)
      if(iu.eq.1)print *,lvtbl(j,k)
      alv=lvtbl(j,k)
      if(alv.gt.0.0)lvtbl(j,k)=wtblh/(wtlvh*alv)
      if(alv.lt.0.0)lvtbl(j,k)=-alv
c
5036 print *,'ENTER kidney to blood(1) daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5036)kdtbl1(j,k)
      if(nu.eq.1)read(7,*)kdtbl1(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5036) ans
      if(ans.gt.0.and.nu.eq.0)goto5036
      write(6,*)kdtbl1(j,k)
      if(iu.eq.1)print *,kdtbl1(j,k)
```

```
      akd=kdtbl1(j,k)
      if(akd.gt.0.0)kdtbl1(j,k)=wtblh/(wtkdh*akd)
      if(akd.lt.0.0)kdtbl1(j,k)=-akd
c
 5037 print *,'ENTER kidney to blood(2) daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5037)kdtbl2(j,k)
      if(nu.eq.1)read(7,*)kdtbl2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5037) ans
      if(ans.gt.0.and.nu.eq.0)goto5037
      write(6,*)kdtbl2(j,k)
      if(iu.eq.1)print *,kdtbl2(j,k)
      akd=kdtbl2(j,k)
      if(akd.gt.0.0)kdtbl2(j,k)= wtblh/(wtkdh*akd)
      if(akd.lt.0.0)kdtbl2(j,k)=-akd
c
 5039 print *,'ENTER muscle to blood daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5039)mstbl(j,k)
      if(nu.eq.1)read(7,*)mstbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5039) ans
      if(ans.gt.0.and.nu.eq.0)goto5039
      write(6,*)mstbl(j,k)
      if(iu.eq.1)print *,mstbl(j,k)
      ams=mstbl(j,k)
      if(ams.gt.0.0)mstbl(j,k)=wtblh/(wtmsh*ams)
      if(ams.lt.0.0)mstbl(j,k)=-ams
c
 5041 print *,'ENTER res.tissue to blood daily transport coefficient or'
      print *,'partition coefficient; if a partition coefficient is'
      print *,'entered, then enter the negative of the value; a 0.0'
      print *,'entree will ignore this component; enter now'
      if(nu.eq.0)read(*,*,err=5041)rstbl(j,k)
      if(nu.eq.1)read(7,*)rstbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5041) ans
      if(ans.gt.0.and.nu.eq.0)goto5041
      write(6,*)rstbl(j,k)
      if(iu.eq.1)print *,rstbl(j,k)
      rsb=rstbl(j,k)
      if(rsb.gt.0.0)rstbl(j,k)=wtblh/(wtrsh*rsb)
      if(rsb.lt.0.0)rstbl(j,k)=-rsb
      ahc=1.0
      if(alsp(j).le.0.0)ahc=0.0
      go to 92
   80 continue
c initializes transport coefficient indexes for zoon groups for all
c other contaminants ; sets insect indexes to 0.0 and 10.0
```

```
      ahc=1.0
      if (alsp(j)) 83,82,83
   82 ahc=0.0
c
c estimates transport coefficients defined above for other nuclides using
c default data in code(UCRL); except for insects(estimated)
c
c **** default partition coefficients for all organic contaminants ****
   83 tbb=1.0
      tbl=1.0
      tbk=1.0
      tbw=0.6
c **** end partition coefficents ***
c
c *** estimated biological half-lives of nuclides
c
c dummy variables if UCRL data not specified
c real variables from UCRL data if specified
      fbww=1.0
      if(ian.le.100)fbww=fbwbz(ian)
      tbbb=1.0
      if(ian.le.100)tbbb=0.693/tbbnz(ian)
      tbll=1.0
      if(ian.le.100)tbll=0.693/tblvz(ian)
      tbkk=1.0
      if(ian.le.100)tbkk=0.693/tbkdz(ian)
      tbww=1.0
      if(ian.le.100)tbww=0.693/tbwbz(ian)
c
c *** end nuclide biological half-times estimates from UCRL data***
c
      if(k.gt.1.and.k.lt.7) pltbl(j,k)=0.1
      if(k.gt.1.and.k.lt.7) pltblg(j,k)=0.0
      plcm1(j)=0.25*0.4*ahc
      plcm2(j)=0.25*.4*ahc
      plcm3(j)=0.25*0.2
c
      if(k.gt.1.and.k.lt.7)gtbl(j,k)=fbwbz(ian)
      if(k.gt.1.and.k.lt.7)bltmk(j,k)=fmk(ian)
c sets contaminant biological half-time of nutrients to 1 day
      if(k.gt.1.and.k.lt.7)bltur(j,k)=0.693
      bntbl1(j,k)=wtblh/(wtbnh*tbbb)
      if(ian.gt.100)bntbl1(j,k)=tbb
      bntbl2(j,k)=wtblh/(wtbnh*tbbb)
      if(ian.gt.100)bntbl2(j,k)=tbb
      lvtbl(j,k)=wtblh/(wtlvh*tbll)
      if(ian.gt.100)lvtbl(j,k)=tbl
      kdtbl1(j,k)=wtblh/(wtkdh*tbkk)
      if(ian.gt.100)kdtbl1(j,k)=tbk
      kdtbl2(j,k)=wtblh/(wtkdh*tbkk)
      if(ian.gt.100)kdtbl2(j,k)=tbk
      mstbl(j,k)=wtblh/(wtmsh*tbww)
      if(ian.gt.100)mstbl(j,k)=tbw
      rstbl(j,k)=wtblh/(wtrsh*tbww)
```

```
      if(ian.gt.100)rstbl(j,k)=tbw
c sets biological half-time of nutrients in fat to 4 days
      if(k.gt.1.and.k.lt.7)
     1fttbl(j,k)=wtblh/(fatbh*0.1733)
   92 if(k.gt.1.and.k.lt.7)goto91
      call clear_screen@
      print *,'*******************************************************'
      print *,'entered or default partition coefficients'
      print *,'if default, partition coefficients estimated from UCRL'
      print *,'biological half-life data for nuclides; default organics'
      print *,'are best guess'
      print *,'*******************************************************'
      print *,'bone 1,2=',bntbl1(j,k),bntbl2(j,k),j
      print *,'liver=',lvtbl(j,k)
      print *,'kidney 1,2=',kdtbl1(j,k),kdtbl2(j,k)
      print *,'muscle=',mstbl(j,k)
      print *,'residual=', rstbl(j,k)
      print *,'*******************************************************'
      print *,'CAUTION, UCRL specific organ data not always reliable'
      print *,'*******************************************************'
      print *,'fat(partition coefficient)',fttbl(j,k)
      print *,'gut absorption coefficient',gtbl(j,k)
      print *,'transport fraction to urine from blood,d-1',bltur(j,k)
      print *,'transport fraction to milk, from blood,d-1',bltmk(j,k)
      print *,'*****************************************'
      pause
      call clear_screen@
c
c estimates blood to milk transport coefficient(BLTMK) from UCRL data;
c insect value not used
c
   91 if(ahc.le.0.0)bltmk(j,k)=0.0
   90 continue
c ** end contaminant loop ********************************************
c default age group spread and group weights for beef and dairy cows'
      if (iage.eq.0) go to 101
      if (popt(j).gt.1.0)goto101
      print *, iage,popt(j),j
c enters user specified age structure, fraction; not used for insects
c
      print *,'for ANIMAL',j,'the following population structure'
      print *,'parameters'
c
 5042 print *,'ENTER user specified population age structure for'
      print *,iage,'male age groups (age in years)'
      if(nu.eq.0)read(*,*,err=5042)(a(k,j,1),k=1,iage)
      if(nu.eq.1)read(7,*)(a(k,j,1),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5042) ans
      if(ans.gt.0.and.nu.eq.0)goto5042
      write(6,*)(a(k,j,1),k=1,iage)
      if(iu.eq.1)print *,(a(k,j,1),k=1,iage)
c
 5043 print *,'ENTER user specified population age structure for'
```

```
      print *,iage,'female age groups (age in years)'
      if(nu.eq.0)read(*,*,err=5043)(a(k,j,2),k=1,iage)
      if(nu.eq.1)read(7,*)(a(k,j,2),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5043) ans
      if(ans.gt.0.and.nu.eq.0)goto5043
      write(6,*)(a(k,j,1),k=1,iage)
      if(iu.eq.1)print *,(a(k,j,2),k=1,iage)
c
 5048 print *,'ENTER user specified population male mortality rate for'
      print *,iage,'age groups, units are in per yr; enter 0.0 for'
      print *,'invertebrates (they are determined in INVERT subroutine)'
c
      if(nu.eq.0)read(*,*,err=5048)(tqx0(k,1,j),k=1,iage)
      if(nu.eq.1)read(7,*)(tqx0(k,1,j),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5048) ans
      if(ans.gt.0.and.nu.eq.0)goto5048
      write(6,*)(tqx0(k,1,j),k=1,iage)
      if(iu.eq.1)print *,(tqx0(k,1,j),k=1,iage)
c
 5049 print *,'ENTER user specified population female mortality rate'
      print *,'for',iage,'age groups, units are in per yr;enter 0.0 for'
      print *,'invertebrates (they are determined in INVERT subroutine)'
c
      if(nu.eq.0)read(*,*,err=5049)(tqx0(k,2,j),k=1,iage)
      if(nu.eq.1)read(7,*)(tqx0(k,2,j),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5049) ans
      if(ans.gt.0.and.nu.eq.0)goto5049
      write(6,*)(tqx0(k,2,j),k=1,iage)
      if(iu.eq.1)print *,(tqx0(k,2,j),k=1,iage)
c
c**** initializes insect growth parameters
  101 if (alsp(j)) 5045,154,5045
c
  154 insct=insct+1
      call sleep@(2.0)
      call clear_screen@
      nux=nu
      nu=nuu
      call invert (iage,j,o,lx,insct,bpx,rdidx)
      nu=nux
      call sleep@(2.0)
      call clear_screen@
      goto100
c
 5045 print *,'ENTER  specified activity factor for animal=',j
      print *,'ex.. cows=2.,coyote=1.4,rabbit=2.,mouse=1.4,squirrel=2.'
      print *,'kangaroo rat=2., deer-elk=2., etc.'
      if(nu.eq.0)read(*,*,err=5045) acti(j)
      if(nu.eq.1)read(7,*) acti(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5045) ans
```

```
      if(ans.gt.0.and.nu.eq.0)goto5045
      write(6,*) acti(j)
      if(iu.eq.1)print *, acti(j)
c
 5046 print *,'ENTER  digestion gas loss fraction for animal=',j
      print *,'ex.. cows=.06,coyote=.01,rabbit=.02,mouse-squirrel=.02'
      print *,'kangaroo rat=.02., deer-elk=.06., etc.'
      if(nu.eq.0)read(*,*,err=5046) gaslos(j)
      if(nu.eq.1)read(7,*) gaslos(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5046) ans
      if(ans.gt.0.and.nu.eq.0)goto5046
      write(6,*) gaslos(j)
      if(iu.eq.1)print *, gaslos(j)
c
 5047 print *,'ENTER  digestion urine loss fraction for animal=',j
      print *,'ex.. cows=.06,coyote=.03,rabbit=.02,mouse-squirrel=.05'
      print *,'kangaroo rat=.05., deer-elk=.06., etc.'
      if(nu.eq.0)read(*,*,err=5047) urnlos(j)
      if(nu.eq.1)read(7,*) urnlos(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5047) ans
      if(ans.gt.0.and.nu.eq.0)goto5047
      write(6,*) urnlos(j)
      if(iu.eq.1)print *, urnlos(j)
c
      nue=nu
      nu=nuu
      if(ydos(j).gt.0.0)iian=iian+1
      if(ydos(j).le.0.0)goto100
      do1000mm=1,2
      if(ydos(j).gt.0.0)call effects(j,lx,ii,mm,rdidx)
 1000 continue
      nu=nue
  100 continue
c ** end zoon loop ******************************************************
 5013 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN ANIMAL                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for ANIMAL?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5013) ans
      if(ans) 102,5500,102
  102 rewind 6
```

```
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-222222.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c ** animal loop
      do 120 l=1,nan
c converts RR to positive value if negative previously
      if (rr(l).lt.0.0) rr(l)=-rr(l)
c integer value of oldest age group, years;default value =8.0 years
      jage(l)=agea(l)
      if(agea(l).le.0.0)agea(l)=8.0
      iage=agea(l)
c weight, kg, of mature animal
      wt1(iage,l)=rr(l)
c initial total population by sex 1=male, 2=female
      pop(1,l)=zoo(l)*(1.-sxrt(l))
      pop(2,l)=zoo(l)*sxrt(l)
      popp(1,l)=pop(1,l)
      popp(2,l)=pop(2,l)
      poppi(1,l)=pop(1,l)
      poppi(2,l)=pop(1,l)
      affsp(l,1)=0.0
      affsp(l,2)=0.0
      affsp(l,3)=0.0
      offsp(l,1,1)=0.0
      offsp(l,1,2)=0.0
      offsp(l,2,1)=0.0
      offsp(l,2,2)=0.0
      offsp(l,3,1)=0.0
      offsp(l,3,2)=0.0
      sfd(1,l)=.0001
      sfd(2,l)=.0001
      sfd(3,l)=.0001
      sfd(4,l)=.0001
      sfd(5,l)=.0001
      sfd(6,l)=.0001
      sfd(7,l)=.0001
      sfd(8,l)=.0001
c divides reproductive females into 8 groups based on fecundity
c excludes invertebrates
      if(alsp(l).le.0.0)goto129
c
      alsp(l)=alsp(l)*365.
c estimated gestation period, days estimated from mean life-span
      gestp(l)=0.15*(alsp(l))**0.8
c time delay for newborns in a given year
      cagm(l,1)=gst(l,1)+14.+gestp(l)
```

```
      gst(l,2)=0.0
      if(gstn(l).ge.2.0)gst(l,2)=cagm(l,1)+0.025*alsp(l)
      cagm(l,2)=0.0
      if(gstn(l).ge.2.0) cagm(l,2)=gst(l,2)+14.+gestp(l)
      agcor(l)=0.0
      if(gstn(l).ge.2.0) agcor(l)=(14.+0.025*alsp(l)+gestp(l))/2.0
      ipopt=popt(l)
      trep=0.55*alsp(l)/8.0
      srep=0.15*alsp(l)
      iaged=iage*365
      femgrp=0.0
      akdy=1.0
      do 626 kk=1,iaged
      akd=kk
c excludes age groups below .15, and greater than .70 of the life-span
      if (akd.lt..15*alsp(l).or.akd.gt..70*alsp(l)) go to 626
      femgrp=femgrp+1.0
      kdy=akdy
      if(akd.lt.srep+trep)goto626
      ageo=agea(l)*365.
c estimates number of reproductive females in 8 age groups
      goto (137,132,133,134), ipopt
c parabolic
  132 sfd(kdy,l)=((ageo-srep)/ageo)**1.5-((ageo-(srep+trep))/
     1ageo)**1.5
      srep=srep+trep
      goto135
c exponential
  133 plmb=10.*alog(2.0)/ageo
      srepp=ageo-srep
      trepp=ageo-(srep+trep)
      sfd(kdy,l)=exp(-plmb*srepp)-exp(-plmb*trepp)
      srep=srep+trep
      goto135
c hyperbolic
  134 sfd(kdy,l)=alog(srep)-srep/ageo-alog(srep+trep)+(srep+trep)/
     1ageo
      srep=srep+trep
      goto135
  137 isrp=srep/365.
      if(isrp.lt.1)isrp=1
      istrp=(srep+trep)/365.
      if(istrp.lt.1)istrp=1
      sfd(kdy,l)=(a(isrp,l,2)+a(istrp,l,2))/2.0
  135 akdy=akdy+1.0
      if(akdy.gt.8.0)goto136
c
  626 continue
c determines upper age for 8 age categories of reporductive females
  136 kn=1
      akn=kn
      femgpi=0.0
      do627kk=1,iaged
      if(kn.gt.8)goto627
```

```
      akd=kk
      if (akd.lt..15*alsp(l).or.akd.gt..70*alsp(l)) go to 627
      femgpi=femgpi+1.0
      femc=femgrp*akn/8.0
      if(femgpi.lt.femc)goto627
      brty(kn,l)=akd
      kn=kn+1
      akn=kn
  627 continue
c
c ** population structure parameters initialized
  129 sfp=1.0
      agemi=0.0
c ** animal age loop ***************************************************
      do 110 k=1,iage
c population dynamics not active for invertebrates
      if(alsp(l).le.0.0)goto111
      aka=k
c estimates initial age , days, of animal at start of simulation
      agemi=agemi+365.
      if(agemi.ge.alsp(l))agemi=alsp(l)
      if(k.eq.1) agemri=agemi-cagm(l,1)-agcor(l)
      if(agemri.lt.0.0)agemri=-agemri
      agemrf=365.
c estimates age stucture,fraction, for each yearly age group
      goto (128,123,124,125), ipopt
c parabolic
  123 sfc=((ageo-agemi)/ageo)**1.5
      if(k.eq.1)sfr=((ageo-agemri)/ageo)**1.5
      if(k.eq.1)sfrf=((ageo-agemrf)/ageo)**1.5
      a(k,l,2)=(sfp-sfc)
      if(k.eq.iage)a(k,l,2)=sfp
      a(k,l,1)=a(k,l,2)
      sfp=sfc
      if(k.ne.1)goto111
      sfpop(l)=(1.0-sfc)
      rat12(l)=sfrf/(sfr+sfrf)
      goto127
c exponential
  124 plmb=10.*alog(2.0)/ageo
      sfc=exp(-plmb*(ageo-agemi))
      if(k.eq.1)sfr=exp(-plmb*(ageo-agemri))
      if(k.eq.1)sfrf=exp(-plmb*(ageo-agemrf))
      a(k,l,2)=(sfp-sfc)
      if(k.eq.iage)a(k,l,2)=sfp
      a(k,l,1)=a(k,l,2)
      sfp=sfc
      if(k.ne.1)goto111
      sfpop(l)=(1.0-sfc)
      rat12(l)=sfrf/(sfr+sfrf)
      goto127
c hyperbolic
  125 chyp=alog(ageo)-1.0+1.0/ageo
      sfc=(alog(agemi)+agemi/ageo+1.0/ageo)/chyp
```

```
      if(k.eq.1)sfr=(alog(agemri)+agemri/ageo+1.0/ageo)/chyp
      if(k.eq.1)sfrf=(alog(agemrf)+agemrf/ageo+1.0/ageo)/chyp
      a(k,l,2)=(sfp-sfc)
      if(k.eq.1)a(k,l,2)=sfc
      if(k.eq.iage)a(k,l,2)=sfp
      a(k,l,1)=a(k,l,2)
      sfp=sfc
      if(k.ne.1)goto111
      sfpop(l)=(1.0-sfc)
      rat12(l)=sfrf/(sfr+sfrf)
      goto127
  128 if(k.ne.1)goto111
c assumes male and female population fraction are equal for 1 year-olds
      sfpop(l)=a(1,l,2)
      sfr=a(1,l,2)*agemri/agemrf
      if(jage(l).eq.0)sfr=a(1,l,2)
      sfrf=a(2,l,2)
      if(jage(l).eq.0)sfrf=0.0
      rat12(l)=sfr/(sfr+sfrf)
c
c differential birth rates for males and females for stable population
c given as number of births/year/female in 8 reproductive age groups
c
  127 brrt(1,l,1)=.003*sfpop(l)*pop(1,l)/(sfd(1,l)*pop(2,l)*1.0)
      brrt(1,l,2)=.003*sfpop(l)*pop(2,l)/(sfd(1,l)*pop(2,l)*1.0)
      brrt(2,l,1)=.138*sfpop(l)*pop(1,l)/(sfd(2,l)*pop(2,l)*1.0)
      brrt(2,l,2)=.138*sfpop(l)*pop(2,l)/(sfd(2,l)*pop(2,l)*1.0)
      brrt(3,l,1)=.296*sfpop(l)*pop(1,l)/(sfd(3,l)*pop(2,l)*1.0)
      brrt(3,l,2)=.296*sfpop(l)*pop(2,l)/(sfd(3,l)*pop(2,l)*1.0)
      brrt(4,l,1)=.295*sfpop(l)*pop(1,l)/(sfd(4,l)*pop(2,l)*1.0)
      brrt(4,l,2)=.295*sfpop(l)*pop(2,l)/(sfd(4,l)*pop(2,l)*1.0)
      brrt(5,l,1)=.207*sfpop(l)*pop(1,l)/(sfd(5,l)*pop(2,l)*1.0)
      brrt(5,l,2)=.207*sfpop(l)*pop(2,l)/(sfd(5,l)*pop(2,l)*1.0)
      brrt(6,l,1)=.050*sfpop(l)*pop(1,l)/(sfd(6,l)*pop(2,l)*1.0)
      brrt(6,l,2)=.050*sfpop(l)*pop(2,l)/(sfd(6,l)*pop(2,l)*1.0)
      brrt(7,l,1)=.010*sfpop(l)*pop(1,l)/(sfd(7,l)*pop(2,l)*1.0)
      brrt(7,l,2)=.010*sfpop(l)*pop(2,l)/(sfd(7,l)*pop(2,l)*1.0)
      brrt(8,l,1)=.001*sfpop(l)*pop(1,l)/(sfd(8,l)*pop(2,l)*1.0)
      brrt(8,l,2)=.001*sfpop(l)*pop(2,l)/(sfd(8,l)*pop(2,l)*1.0)
c
c estimates initial population by age group accounting for birth lag
  111 tpop(k,1,l)=a(k,l,1)*pop(1,l)
      tpop(k,2,l)=a(k,l,2)*pop(2,l)
c stores initial population numbers for all vertebrates
      tpopi(k,1,l)=tpop(k,1,l)
      tpopi(k,2,l)=tpop(k,2,l)
      if(alsp(l).le.0.0)goto112
      if(k.eq.1)tpop(k,1,l)=(sfr-sfrf)*pop(1,l)
      if(tpop(k,1,l).lt.0.0)tpop(k,1,l)=-tpop(k,1,l)
      if(k.eq.1)tpop(k,2,l)=(sfr-sfrf)*pop(2,l)
      if(tpop(k,2,l).lt.0.0)tpop(k,2,l)=-tpop(k,2,l)
  112 if(k.eq.1)offsp(l,1,1)=tpop(k,1,l)
      if(k.eq.1)offsp(l,1,2)=tpop(k,2,l)
c initial age(AGEM), and mean life span, days (ALSP), kg
```

```
      alspi=alsp(l)
c assumes life-span of invertebrates to be 1 year
      if(alsp(l).le.0.0)alspi=365.
      agem(k,l)=(aka-1.0)*365.
      if(alsp(l).le.0.0)cagm(l,1)=0.0
      if(alsp(l).le.0.0)cagm(l,2)=0.0
      if(k.eq.1)agem(k,l)=0.006*alspi
c
      wt1(k,l)=rr(l)*(1.-exp(-9.0*agem(k,l)/alspi))
      wt(k,l)=wt1(k,l)
      wtgrm=wt1(k,l)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
c assumes body weight of 1 gram for invertebrates
      if(alsp(l).le.0.0)wtgrm=1.0
c
c estimates initial metabolic weight(AMX) using mature weight of animal
      amx(k,l)=5.6234e-03*wtgrm**0.75
      amw(k,l)=amx(k,l)
c
c potential food intake, kg/day, for animal unit in age group(k)
c initial conditions assume all factors=1.0, and digestibility=0.75
      fdin(k,l)=70.0*amx(k,l)/(0.75*1000.*4.5)
  110 continue
c ** end age loop ********************************************************
  120 continue
c ** end animal loop ****************************************************
c
c ** zoon type loop ******************************************************
      do 150 j=1,nan
      if(alsp(j).gt.0.0) pop(1,j)=0.0
      if(alsp(j).gt.0.0) pop(2,j)=0.0
c integer value of oldest age group in zoon(J) population
      iage=agea(j)
c
c ** age loop ************************************************************
      do 140 l=1,iage
c excludes beef,dairy cows, and sets invertebrate fraction to one
      if(jage(j).eq.0)goto122
      if(alsp(j).gt.0.0)goto122
      a(l,j,2)=1.0
      a(l,j,1)=1.0
      tpop(l,1,j)=pop(1,j)*a(l,j,1)
      tpop(l,2,j)=pop(2,j)*a(l,j,2)
      goto119
  122 pop(1,j)=pop(1,j)+tpop(l,1,j)
      pop(2,j)=pop(2,j)+tpop(l,2,j)
c predation and starvation mortalities initialized
  119 tqp(l,j)=0.0
      tqs(l,j)=0.0
c contaminant pool sizes initialized
c
c ** contaminant loop ****************************************************
      do 130 k=1,inz
c lung mass components, second bone and kidney components set to 0.0
```

```
      q1(l,j,k)=0.0
      q2(l,j,k)=0.0
      q3(l,j,k)=0.0
      lf=iage+1
      v(l,j,k)=0.0
      vkd1(l,j,k)=0.0
      vkd2(l,j,k)=0.0
      vbn1(l,j,k)=0.0
      vbn2(l,j,k)=0.0
      vms(l,j,k)=0.0
      vr(l,j,k)=0.0
      vlv(l,j,k)=0.0
      vfat(l,j,k)=0.0
      vgut(l,j,k)=0.0
      vpelt(l,j,k)=0.0
      vpln(l,j,k)=0.0
      vsol(l,j,k)=0.0
      vlug(l,j,k)=0.0
c**** Rem cummulative doses to different animal organs initialized
      rkd(l,j,k)=0.0
      rbn(l,j,k)=0.0
      rndo(l,j,k)=0.0
      rlv(l,j,k)=0.0
      rlu(l,j,k)=0.0
      rms(l,j,k)=0.0
      rkd(lf,j,k)=0.0
      rbn(lf,j,k)=0.0
      rndo(lf,j,k)=0.0
      rlv(lf,j,k)=0.0
      rms(lf,j,k)=0.0
c
      wtgrm=wt1(l,j)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
c assumes body weight of 1 gram for invertebrates
      if(alsp(j).le.0.0)wtgrm=1.0
c weight of blood/hemolymph, kg
      wtbl(l,j)=5.5e-05*wtgrm**0.99
c weight of bone/exoskeleton,kg
      wtbn(l,j)=5.5e-05*wtgrm
c weight of kidneys/malphigian tubules, kg
      wtkd(l,j)=2.12e-05*wtgrm**0.85
c weight of liver, kg
      wtlv(l,j)=8.2e-05*wtgrm**0.87
c weight of lung/trachae
      wtlu(l,j)=1.24e-05*wtgrm**0.99
c weight of muscle, kg
      wtms(l,j)=4.5e-04*wtgrm
c fat weight, kg
      fatb(l,j)=1.1e-04*wtgrm
c inhalation rate,m3/day
      rsrt(l,j)=1.4*amx(l,j)
c weight of residual soft tissue, kg
      wtrs(l,j)=wt1(l,j)-wtbl(l,j)-wtbn(l,j)-wtkd(l,j)-
     1wtlv(l,j)-wtms(l,j)-fatb(l,j)
```

```
c
c atomic number of nuclide...N,P,K,Pu, ..etc
      ian=atno(k)
c initializes urine transport rate
      vurin(l,j,k)=0.0
c initializes metabolic waste transport to parent from fetus
      cbp(l,j,k)=0.0
c excludes organic contaminants
      if(ian.gt.100)goto130
      cbz=1.0e6
c preserves units, pCi, ug, uCi, ...etc.
      if(yaz(l,lx).ge.2)cbz=1.0
c converts to pCi
      if(hflif(l).gt.0.0)cbz=1.0
c
c initial total contaminant  content, grams, of internal/external organs
c for nutrient concentrations
      if(k.lt.2.or.k.gt.6)goto130
c
      v(l,j,k)=(cbwbz(ian)/cbz)*wtbl(l,j)*1000.
      vr(l,j,k)=(cbwbz(ian)/cbz)*wtrs(l,j)*1000.
      vbn1(l,j,k)=(cbbnz(ian)/cbz)*wtbn(l,j)*1000.
      vkd1(l,j,k)=(cbkdz(ian)/cbz)*wtkd(l,j)*1000.
      vlv(l,j,k)=(cblvz(ian)/cbz)*wtlv(l,j)*1000.
      vms(l,j,k)=(cbwbz(ian)/cbz)*wtms(l,j)*1000.
  130 continue
c ** end contaminant loop ********************************************
c
  140 continue
c ** end age loop ****************************************************
c ** animal fetal age relative to gestation period initialized
      aiif(j,1)=0.0
      aiif(j,2)=0.0
      aiiir(j,1)=0.0
      aiiir(j,2)=0.0
      pop2r(j)=1.0
  150 continue
c** end animal loop ***************************************************
      hm8=0.0
      return
c*******************************************************************
c
c animal age group
  180 aii=ii
      jjj=aiii
      iage=agea(in)
c excludes insects
      if(alsp(in).le.0.0)goto 188
      if(aiii.gt.1.0)goto194
      if(ii.gt.iage)goto194
c age of all age groups initialized
      agem(ii,in)=(aii-1.0)*365.
      if(ii.ne.1)goto194
c age of first newborn stage initialized
```

```
      agem(ii,in)=365.-cagm(in,1)-agcor(in)
      affsp(in,1)=agem(ii,in)
      if(affsp(in,1).lt.1.0)affsp(in,1)=1.0
c population of all newborn stages initialized
c
c excludes all but first age group
  194 if(ii.gt.1)goto191
c resets fetus age counter index
      lagf(in)=0
      aiiir(in,1)=aiiir(in,1)+1.0
      if(gst(in,2).ge.2.0)aiiir(in,2)=aiiir(in,2)+1.0
c
c indicates that weaning period has ended; resumes regular time
      if(aiiir(in,1).gt.cagm(in,1)+0.025*alsp(in))aiiir(in,1)=aiii
      if(aiiir(in,2).gt.cagm(in,2)+0.025*alsp(in))aiiir(in,2)=aiii
c
c gestation still active
      if(aiiir(in,1).le.aiii)goto191
      if(gstn(in).lt.2.0) agem(ii,in)=agem(ii,in)+1.0
      if(gstn(in).lt.2.0) goto 703
      if(aiiir(in,2).gt.aiii)agem(ii,in)=agem(ii,in)+1.0
      if(aiiir(in,2).gt.aiii)goto 703
c
  191 if(ii.ne.1)goto188
      if(gstn(in).ge.2.0)goto192
      iagem=affsp(in,2)
      if(iagem.ne.2)goto188
      pop2r(in)=pop(2,in)/poppi(2,in)
      goto193
  192 iagem=affsp(in,3)
      if(iagem.ne.2)goto188
      pop2r(in)=pop(2,in)/poppi(2,in)
c
c population birth rate adjusted for positive and negative excursions
c
  193 continue
c
c excludes all but newborns
  188 if(ii.le.iage)goto189
c
c excludes invertebrates
      if(alsp(in).le.0.0)return
c
c weaning constant initialized
      cntk(1)=0.0
      cntk(2)=0.0
c estimates age of newborns
      agem(ii,in)=aiiir(in,1)-cagm(in,1)
c
c weaning constant set to 1.0 after weaning period
      if(affsp(in,2).gt.cagm(in,1)+0.025*alsp(in))cntk(1)=1.0
      if(gstn(in).lt.2.0)goto703
      if(affsp(in,3).gt.cagm(in,2)+0.025*alsp(in))cntk(2)=1.0
      goto703
```

```
  189 agem(ii,in)=agem(ii,in)+1.0
      if(ii.ne.1)goto190
      affsp(in,1)=agem(ii,in)
      if(affsp(in,1).lt.1.0)affsp(in,1)=1.0
      if(affsp(in,2).ge.1.0)affsp(in,2)=affsp(in,2)+1.0
      if(affsp(in,3).ge.1.0)affsp(in,3)=affsp(in,3)+1.0
c
c beef,dairy cow, and invertebrate mortalities set to 0.0; the latter
c are estimated in INVERT subroutine with variable SUMMRT; population
c dynamics not simulated for domestic cattle
c
  190 tqx(ii,1,in)=0.0
      tqx(ii,2,in)=0.0
      if(alsp(in).le.0.0)goto181
      tqx1=0.0
      tqx2=0.0
      tqx3=0.0
c
      if(agem(ii,in).le.0.0)goto703
      if(jage(in).eq.0)goto181
c
      ipopt=popt(in)
      of1=affsp(in,1)
      of2=affsp(in,2)
      of3=affsp(in,3)
      agemh=agem(ii,in)
      ageo=agea(in)*365.
      goto (182,183,184,185), ipopt
c parabolic
  183 if(ii.gt.1)goto283
      tqx1=(0.5/ageo)*(1.-of1/ageo)**(-0.5)/pop2r(in)
      tqx2=(0.5/ageo)*(1.-of2/ageo)**(-0.5)/pop2r(in)
      tqx3=(0.5/ageo)*(1.-of3/ageo)**(-0.5)/pop2r(in)
      goto181
  283 tqx(ii,2,in)=tqx(ii-1,2,in)
      tqx(ii,1,in)=tqx(ii-1,2,in)
      if(agem(ii,in).ge.ageo-2.0)goto181
      tqx(ii,2,in)=(0.5/ageo)*(1.-agemh/ageo)**(-0.5)/pop2r(in)
      tqx(ii,1,in)=tqx(ii,2,in)
      goto 181
c exponential
  184 plmb=10.*alog(2.0)/ageo
      if(ii.gt.1)goto284
      tqx1=exp(-plmb*(ageo-of1))/pop2r(in)
      tqx2=exp(-plmb*(ageo-of2))/pop2r(in)
      tqx3=exp(-plmb*(ageo-of2))/pop2r(in)
      goto181
  284 tqx(ii,2,in)=exp(-plmb*(ageo-agemh))/pop2r(in)
      tqx(ii,1,in)=tqx(ii,2,in)
      goto181
c hyperbolic
  185 if(ii.gt.1)goto285
      tqx1=of1**(-2.0)/pop2r(in)
      tqx2=of2**(-2.0)/pop2r(in)
```

```
      tqx3=of3**(-2.0)/pop2r(in)
      goto181
  285 if(ii.gt.1)tqx(ii,2,in)=tqx(ii-1,2,in)
      if(ii.gt.1)tqx(ii,1,in)=tqx(ii-1,2,in)
      tqx(ii,2,in)=agemh**(-2.0)/pop2r(in)
      tqx(ii,1,in)=tqx(ii,2,in)
      goto181
  182 tqxp1=0.0
      tqxp2=0.0
      if(ii-1.gt.0)tqxp1=tqx0(ii-1,1,in)/365.
      if(ii-1.gt.0)tqxp2=tqx0(ii-1,2,in)/365.
      dqxp1=tqx0(ii,1,in)/365.-tqxp1
      dqxp2=tqx0(ii,2,in)/365.-tqxp2
      tqx(ii,1,in)=tqx0(ii,1,in)/365.+dqxp1*(aiii/365.)
      tqx(ii,2,in)=tqx0(ii,2,in)/365.+dqxp2*(aiii/365.)
      if(ii.gt.1)goto181
      tqx1=tqx0(ii,1,in)/365.
      tqx2=tqx0(ii,1,in)/365.
      tqx3=tqx0(ii,1,in)/365.
  181 a(ii,in,1)=tpop(ii,1,in)/pop(1,in)
      a(ii,in,2)=tpop(ii,2,in)/pop(2,in)
c
c** current weight of internal/external tissues and organs
c body weight, grams
      wtgrm=wt(ii,in)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
c
c assumes body weight of 1 gram for invertebrates
      if(alsp(in).le.0.0)wtgrm=1.0
c weight of blood/hemolymph, kg
      wtbl(ii,in)=5.5e-05*wtgrm**0.99
c weight of bone/exoskeleton,kg
      wtbn(ii,in)=5.5e-05*wtgrm
c weight of kidneys/malphigian tubules, kg
      wtkd(ii,in)=2.12e-05*wtgrm**0.85
c weight of liver, kg
      wtlv(ii,in)=8.2e-05*wtgrm**0.87
c weight of lung/trachae
      wtlu(ii,in)=1.24e-05*wtgrm**0.99
c weight of muscle, kg
      wtms(ii,in)=4.5e-04*wtgrm
c fat weight, kg
      fatb(ii,in)=1.1e-04*wtgrm
c inhalation rate,m3/day
      rsrt(ii,in)=1.4*amw(ii,in)
c weight of residual soft tissue, kg
      wtrs(ii,in)=wt(ii,in)-wtbl(ii,in)-wtbn(ii,in)-wtkd(ii,in)-
     1wtlv(ii,in)-wtms(ii,in)-fatb(ii,in)
c
c total body water, liters
      tbwl=9.077e-04*wtgrm**0.963
c
c cardiac output, liters/day
      cardrt=1.44*0.7081*wtgrm**0.79
```

```
c
c renal flow, liters/day
      renflo=1.44*0.2109*wtgrm**0.77
c
c milk volume production in mature organism
      flomlk=1.1915e-03*wtgrm**0.7
c
c** compartment volumes **
c
c plasma vol, liters
      volpls=4.29e-05*wtgrm**0.992
c muscle vol, liters
      volmus=4.63e-04*wtgrm**1.009
c kidney vol, liters
      volkid=2.18e-05*wtgrm**0.843
c liver vol, liters
      volliv=8.59e-05*wtgrm**0.885
c gut vol, liters
      volgut=8.65e-05*wtgrm**0.909
c gut lumen vol, liters
      vollum=5.78e-05*wtgrm**0.934
c heart vol, liters
      volhrt=2.0e-06*wtgrm**1.043
c lungs vol, liters
      vollun=3.16e-06*wtgrm**1.104
c spleen vol, liters
      volspl=4.52e-06*wtgrm**0.901
c marrow vol, liters
      volmar=3.3e-05*wtgrm**0.936
c fat vol, liters
      volfat=1.1e-04*wtgrm
c current residual volume
c     volres=volgut+volhrt+volspl
      volres=volgut+volhrt+volspl+vollun
c
c ** plasma flow rates **
c
c plasma flow, liters/day
      flopls=1.44*.601*wtgrm**0.778
c muscle flow, liters/day
      flomus=1.44*.0817*wtgrm**0.81
c kidney flow, liters/day
      flokid=1.44*0.0845*wtgrm**0.802
c liver flow, liters/day
      floliv=1.44*0.0944*wtgrm**0.792
c gut flow, liters/day
      flogut=1.44*0.102*wtgrm**0.787
      floliv=floliv+0.9*flogut
c heart flow, liters/day
      flohrt=1.44*5.77e-03*wtgrm**0.965
c lung flow, liters/day
      flolun=1.44*0.215*wtgrm**0.82
c spleen flow, liters/day
      flospl=1.44*1.79e-03*wtgrm**1.028
```

```
c marrow flow, liters/day
      flomar=1.44*0.0164*wtgrm**0.804
c fat flow, liters/day
      flofat=0.75*flomus
c       flores=flohrt+flospl
      flores=flohrt+flospl+0.1*flogut+0.1*flolun
c urine production, liters/day
      flourn=0.024*.0064*wtgrm**0.82
c
c excludes all animals but insects
      if (alsp(in)) 400,405,400
c
  405 call invert (ii,in,o,lx,insct,bpx,rdidx)
c
      goto 490
c*********************************************************************
c
c expected weight of animal unit based on life-span (ALSP) and
c real-time age (AGEM) for age group(II),and expected weight
  400 bfct=0.0
      if(ii.eq.1)bfct=0.006*alsp(in)
c estimates newborn effective age for weight purposes
      agemi=agem(ii,in)+bfct
      wt1(ii,in)=rr(in)*(1.-exp(-9.0*agemi/alsp(in)))
c expected metabolic weight of animal unit, kg
      wtgrm=wt1(ii,in)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
c
c ** expected metabolic weight of animal unit, kg
c
      amx(ii,in)=5.623e-03*wtgrm**0.75
c
c ** estimated metabolic weight of animal unit, kg
      wtgrm=wt(ii,in)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
c
c estimated metabolic weight of animal unit, kg
      amw(ii,in)=5.623e-03*wtgrm**0.75
c
c daily food intake per animal unit, kg
      o=0.0
c amount of water in daily food intake, liters/day
      watfod=0.0
c daily amount of water consumed ,liters/day, consumed by animal unit
      watvol=0.0
c estimated heat increment as a function of environmental temperature
      heat=2.5-.0676*t(in)
c digestibility fraction based on live(green) and dry biomass in diet
      dige(in)=(1.-grna(in))*.45+grna(in)*.75
c allows high digestibility during weaning period
      if(agem(ii,in).lt.0.025*alsp(in)) dige(in)=0.75
c estimated digestibility effect for animals on daily food intake
      cfx=.85+1.143*(dige(in)-.45)
c age(II) effect on food intake based on animal age(AGEM) and life-span of
```

```
c animal unit(ALSP)
      cdmx=(1.0+exp(-18.*agem(ii,in)/alsp(in)))
c potential food intake, kg/day, for animal unit in age group(II)
      fdin(ii,in)=70.0*amw(ii,in)*heat*acti(in)*cdmx*cfx/
     1(dige(in)*1000.*4.5)
      fdine=70.*amx(ii,in)*heat*acti(in)*cdmx*cfx/(dige(in)*1000.*4.5)
c intake adjusted for food intake (FDIN) over expected(FDINE)
      if (fdin(ii,in).gt.1.25*fdine)fdin(ii,in)=1.25*fdine
c** assumes constant age related food consumption rate
      if(fdin(ii,in).lt.fdine)fdin(ii,in)=fdine
      fdint=fdin(ii,in)
      if(ototb(in).lt.otota(in))fdint=fdint*ototb(in)/otota(in)
c metabolic heat production losses expressed in kg/day
      rspm=70*amw(ii,in)*heat*acti(in)/(4.5*1000.)
      prwt=wt(ii,in)
c estimated actual food intake, kg/day
      o=fdint
c gas losses from food, kg/day
      gasls=gaslos(in)*o*dige(in)
c urine losses from food, kg/day
      urnls=urnlos(in)*o*dige(in)
c feces losses from food, kg/day
      fecls=(1.-dige(in))*o
c water intake, l/day, based on heat losses
      watvol=rspm*1000.*4.5*.000116
c water in food based on nitrogen content, l/day
      watfod=0.3*antro(in)*o
c water required, l/day, based on total body weight(W); bounded
      watreq=0.04*wt(ii,in)
      if (ii.ge.3.0) watreq=0.10*wt(ii,in)
c anabolic mass increment,kg/d, adjusted for mass losses; bounded
      w5=o-rspm-gasls-urnls-fecls
      if (w5.lt.0.0) go to 410
c digested protein as, kg/d
      prtd=.01*antro(in)*w5
c growth requirement for lean body mass, kg/d; bounded
      amd=(amx(ii,in)-amw(ii,in))/2.50
      ampr=0.0
      amch=0.0
      if (amd.le.0.0) goto121
      if (w5.le.amd) amd=w5
c protein fraction of growth increment kg/d; bounded with protein in feed
      ampr=0.6944*amd
      if (prtd.le.ampr) ampr=prtd
c non-nitrogen portion of growth increment,kg/d
      amch=amd-ampr
c protein available for maintenance, kg/d,; rest deamminated
      amchr=0.44*ampr
c protein requirement if not available from growth increment,kg/d; bounded
      if (amchr.gt.amch) ampr=2.272*amchr
c non-nitrogen portion of growth increment bounded to available energy from
c nitrogenous portion
      if (amch.lt.amchr) amch=amchr
c metabolic growth increment adjusted for energy utilization costs, kg/d;
```

```
c bounded
    amd=w5-ampr-amch
    if (amd.le.0.0) amd=0.0
c net protein deficit or surplus estimated, kg/d; bounded
    prtd=prtd-ampr
    if (prtd.le.0.0) prtd=0.0
c metabolic weight, kg live weight, incremented
c
  121 amw(ii,in)=amw(ii,in)+(ampr+amch)*2.50
c
c total weight of animal unit incremented, kg live weight
    if(amw(ii,in).le.0.001)amw(ii,in)=0.001
    wtgrm=amw(ii,in)*1000.
    if(wtgrm.le.1.0)wtgrm=1.0
c
    wt(ii,in)=1.023e-04*wtgrm**1.33
    if(wt(ii,in).ge.wt1(ii,in))wt(ii,in)=wt1(ii,in)
    tqs(ii,in)=0.0
c starvation not allowed if weight 70% or more of expected weight
    if(wt(ii,in).ge.0.7*wt1(ii,in))goto420
c starvation not allowed if weight gain is observed
    if(wt(ii,in)-prwt.ge.0.0)goto420
c
    print *,'animal type',in,'is starving'
c
c estimates 1/10 starving fraction to more gradual population losses
    strv1=((wt1(ii,in)-wt(ii,in))/(0.250663*wt1(ii,in)))
    strv2=((wt1(ii,in)-wt(ii,in))/(0.1*wt1(ii,in)))/8.0
    tqs(ii,in)=0.1*strv1*exp(-strv2)
    go to 420
c lean body mass utilization, kg/d, estimated for food deficit; bounded to
c existing lean body mass
  410 amftl=-w5*0.484
c net metabolic mass required for maintenance, kg/d; bounded
    amprl=-w5-amftl
    if (amprl.le.0.0) amprl=0.0
c requirement adjusted for conversion losses , kg/d; bounded to existing
c metabolic biomass
    amprl=amprl*1.6197*2.50
    if (amprl.le.0.0)amprl=0.0
c metabolic weight adjusted for loss, kg/d; bounded
    amw(ii,in)=amw(ii,in)-amprl
c total weight of animal unit estimated, kg,; bounded
    wtgrm=amw(ii,in)*1000.
    if(wtgrm.le.1.0)wtgrm=1.0
c
c** insures power function applies to quantities 1 gram or larger
    wt(ii,in)=1.023e-04*wtgrm**1.333
    if(wt(ii,in).ge.wt1(ii,in))wt(ii,in)=wt1(ii,in)
    tqs(ii,in)=0.0
c
c starvation not allowed if weight 70% or more of expected weight
    if(wt(ii,in).ge.0.7*wt1(ii,in))goto420
c starvation not allowed if weight gain is observed
```

```
      if(wt(ii,in)-prwt.ge.0.0)goto420
c
      print *,'animal type',in,'is starving'
c
c estimates 1/10 starving fraction to more gradual population losses
      strv1=((wt1(ii,in)-wt(ii,in))/(0.250663*wt1(ii,in)))
      strv2=((wt1(ii,in)-wt(ii,in))/(0.1*wt1(ii,in)))/8.0
      tqs(ii,in)=0.1*strv1*exp(-strv2)
c
c water in food(WATFOD), and that from drinking(WATVOL),l/d, ; bounded
  420 if (watvol.le.0.0) watvol=0.0
      if (watfod.le.0.0) watfod=0.0
c net water deficit estimated,l/d, ; bounded
      watupt=watreq-watvol-watfod
      if (watupt.le.0.0) watupt=0.0
c consumption ratio to that required(OTOTB) from all regions for each animal
c type(IN) from animal sources
  490 cnsma=0.0
      cnto=0.0
c
c ** region loop ******************************************************
      do 440 lz=1,ipx
      ijj=0
c
c ** animal loop ******************************************************
      do 430 kj=1,nan
      if (ototb(in).le.0.0) go to 430
      if (autzd(in,kj,lz)) 430,430,431
  431 ijj=ijj+1
      cnsma=cnsma+up(3+ijj,lz)/ototb(in)
      cnto=cnto+up(3+ijj,lz)
  430 continue
c ** end animal loop ******************************************************
  440 continue
c ** end region loop ******************************************************
c grams of green(live) food consumed adjusted for non-plant sources
      gc=grna(in)*o*1000.*(1.0-cnsma)
c grams of dry food consumed adjusted for non-plant sources
      d4=(1.-grna(in))*o*1000.*(1.0-cnsma)
c grams of animal food consumed (dry weight)
      gac=o*1000.*cnsma
c total grams of plant food consumed; bounded
      c2=gc+d4
      if(c2.ge.o*1000.)c2=o*1000.
      if (c2.le.0.0) c2=0.0
c total feed consumed from all sources, kg/day, initialized
      feed=0.0
c
c total available biomass, g/m2, available for consumption initialized
      tsumb=0.0
c ** region loop ******************************************************
      do 480 lz=1,ipx
c
c ** contaminant loop ******************************************
```

```
      do 450 k=1,inz
c contaminant ingestion from plant ingestion of range crops
      if(lz.eq.1)vpln(ii,in,k)=0.0
c total contaminant ingested, per day,/ animal unit of age(II) on range
      c31(in,k,lz)=c2*spac(in,k,lz)*rfrz(in,lz)*sptfrr(lz)
      vpln(ii,in,k)=vpln(ii,in,k)+c31(in,k,lz)
  450 continue
c ** end contaminant loop ********************************************
c
c ** plant loop ******************************************************
      do 470 jj=1,ipl
c excludes range plants
      if (putzd(in,jj,lz).le.3.0) go to 470
      if(putzd(in,jj,lz).ge.7.0)goto470
c
c ** contaminant loop ************************************************
      do 460 k=1,inz
      if(ototb(in).le.0.0)goto460
c total contaminant ingested, per day/animal unit of age(ii), from
c supplementary feed
      c31(in,k,lz)=c31(in,k,lz)+c2*spas(in,jj,k,lz)*sptfrf(lz)*
     1putfrc(in,jj,lz)
      vpln(ii,in,k)=vpln(ii,in,k)+c2*spas(in,jj,k,lz)*sptfrf(lz)*
     1putfrc(in,jj,lz)/ototb(in)
  460 continue
c ** end contaminant loop ********************************************
c total supplementary feed, kg, from plant(jj)
      totsf(in,jj,lz)=c2*putfrc(in,jj,lz)/1000.
  470 continue
c ** end plant loop ******************************************************
c total feed consumed from all regions, kg/day
      feed=feed+qsum(1,lz)+qsum(2,lz)+qsum(3,lz)+qsum(4,lz)+qsum(5,lz)
     1 +qsum(6,lz)+qsum(7,lz)
  480 continue
c ** end region loop *****************************************************
      ototbt=ototb(in)
      if(ototb(in).lt.otota(in))ototbt=otota(in)
c initialize contaminant intake from ingestion by animal(IN)
      do581k=1,inz
      c31tt(in,k)=0.0
      c31plt(ii,in,k)=0.0
  581 continue
c excludes non-range food consumption
      if (rfrac(in)) 591,590,591
c square meters used/animal unit of age(II)
  591 c4=0.0
      if (feed.gt.0.0) c4=c2/(1000.*feed)
c square meters used by all animal units in age cohort(II)
      m5=c4*a(ii,in,1)*pop(1,in)+a(ii,in,2)*pop(2,in)
c total acres used/day/zoon category
      m6=m5/4047.
c cumulative acreage for all animal units/year combined, acres/yr
      hm8=hm8+m6
c animal index
```

```
      j=in
c ** region loop ***********************************************************
      do 530 lz=1,ipx
c
c ** plant loop ************************************************************
      do 520 l=1,ipl
c total acreage of all plants in a given region used as range
      if(lz.eq.1.and.l.eq.1)tacres(lz)=0.0
      tacres(lz)=tacres(lz)+acres(l,lz)
c excludes non-utilized plants
      if (putzd(j,l,lz)) 521,520,521
c plant index, excludes non-range plants as food sources
  521 kk=putzd(j,l,lz)
      go to (500,500,500,520,520,520,520), kk
c range plants only; feed sources excluded
  500 continue
c null biomass densities food requirements excluded
      if (qsum(kk,lz).le.1.0) go to 520
      if (ototbt.le.0.0) go to 520
c total food present on range, g/m2
      b1sd=b1(l,lz)+sd(l,lz)+sd1(l,lz)+d1(l,lz)+wd1(l,lz)
c total food requirements,kg, by animal units in age cohort(II)
      otbb=ototbt-cnto
c total live food utilized from plant(L) by animal units in age cohort(II),
c g/m2
      xins=a(ii,in,1)*pop(1,in)+a(ii,in,2)*pop(2,in)
      qg=gc*(q10(l,lz)/qsum(kk,lz))*(up(kk,lz)/otbb)*xins*aa2(in)/
     1agr(in,lz)
      if(qg.le.0.0)qg=0.0
c same for dry plant portions of plant(L), g/m2
      qd=d4*(q10(l,lz)/qsum(kk,lz))*(up(kk,lz)/otbb)*xins*aa2(in)/
     1agr(in,lz)
      if(qd.le.0.0)qd=0.0
      if(qd+qg.le.0.0)goto520
c ** contaminant loop ******************************************************
      do 510 k=1,inz
c internal and external concentrations, per m2, adjusted for consumption
      wz3(l,k,lz)=wz3(l,k,lz)-wz3(l,k,lz)*(qg/b1sd)*wnd(l,lz)/b1sd
      if(wz3(l,k,lz).le.0.0)wz3(l,k,lz)=0.0
      z3(l,k,lz)=z3(l,k,lz)-z3(l,k,lz)*(qg)*(b1(l,lz)-wnd(l,lz))/b1sd
      if(z3(l,k,lz).le.0.0)z3(l,k,lz)=0.0
      zsd(l,k,lz)=zsd(l,k,lz)-zsd(l,k,lz)*(qg/b1sd)*sd(l,lz)/b1sd
      if(zsd(l,k,lz).le.0.0)zsd(l,k,lz)=0.0
      z4(l,k,lz)=z4(l,k,lz)-z4(l,k,lz)*(qd/(d1(l,lz)+.1))
      if(z4(l,k,lz).le.0.0)z4(l,k,lz)=0.0
      z42(l,k,lz)=z42(l,k,lz)-z42(l,k,lz)*(qd/(d1(l,lz)+.1))
      if(z42(l,k,lz).le.0.0)z42(l,k,lz)=0.0
      z43(l,k,lz)=z43(l,k,lz)-z43(l,k,lz)*(qd/(d1(l,lz)))
      if(z43(l,k,lz).le.0.0)z43(l,k,lz)=0.0
      wz4(l,k,lz)=wz4(l,k,lz)-wz4(l,k,lz)*(qd/(wd1(l,lz)+.1))
      if(wz4(l,k,lz).le.0.0)wz4(l,k,lz)=0.0
      wz42(l,k,lz)=wz42(l,k,lz)-wz42(l,k,lz)*(qd/(wd1(l,lz)+.1))
      if(wz42(l,k,lz).le.0.0)wz42(l,k,lz)=0.0
      wz43(l,k,lz)=wz43(l,k,lz)-wz43(l,k,lz)*(qd/(wd1(l,lz)+.1))
```

```
      if(wz43(l,k,lz).le.0.0)wz43(l,k,lz)=0.0
      wz43(l,k,lz)=wz43(l,k,lz)-wz43(l,k,lz)*(qd/(wd1(l,lz)+.1))
      if(wz43(l,k,lz).le.0.0)wz43(l,k,lz)=0.0
      z4(l,k,lz)=z4(l,k,lz)-z4(l,k,lz)*(qg/b1sd)*sd1(l,lz)/b1sd
      if(z4(l,k,lz).le.0.0)z4(l,k,lz)=0.0
      z42(l,k,lz)=z42(l,k,lz)-z42(l,k,lz)*(qg/b1sd)*sd1(l,lz)/b1sd
      if(z42(l,k,lz).le.0.0)z42(l,k,lz)=0.0
      z43(l,k,lz)=z43(l,k,lz)-z43(l,k,lz)*(qg/b1sd)*sd1(l,lz)/b1sd
      if(z43(l,k,lz).le.0.0)z43(l,k,lz)=0.0
      a2(l,k,lz)=a2(l,k,lz)-a2(l,k,lz)*qg/(b1(l,lz)+d1(l,lz)+wd1(l,lz)
     1+sd(l,lz)+sd1(l,lz))
      if(a2(l,k,lz).le.0.0)a2(l,k,lz)=0.0
      wz42(l,k,lz)=wz42(l,k,lz)-wz42(l,k,lz)*qd/(b1(l,lz)+d1(l,lz)+wd1
     1 (l,lz)+sd(l,lz)+sd1(l,lz))
      if(wz42(l,k,lz).le.0.0)wz42(l,k,lz)=0.0
      wz43(l,k,lz)=wz43(l,k,lz)-wz43(l,k,lz)*qd/(b1(l,lz)+d1(l,lz)+wd1
     1 (l,lz)+sd(l,lz)+sd1(l,lz))
      if(wz43(l,k,lz).le.0.0)wz43(l,k,lz)=0.0
      a3(l,k,lz)=a3(l,k,lz)-a3(l,k,lz)*(qg+qd)/(b1(l,lz)+d1(l,lz)+wd1(l
     1 ,lz)+sd(l,lz)+sd1(l,lz))
      if(a3(l,k,lz).le.0.0)a3(l,k,lz)=0.0
      afr(l,k,lz)=afr(l,k,lz)-afr(l,k,lz)*(qg)*(sd(l,lz)+sd1(l,lz))/b1sd
      if(afr(l,k,lz).le.0.0)afr(l,k,lz)=0.0
      afs(l,k,lz)=afs(l,k,lz)-afs(l,k,lz)*(qg)*(sd(l,lz)+sd1(l,lz))/b1sd
      if(afs(l,k,lz).le.0.0)afs(l,k,lz)=0.0
c areal concentrations bounded
 510 continue
c ** end contaminant loop ***********************************************
c live(green) and dry biomass densities adjusted for animal consumption,
c g/m2
      d1p=d1(l,lz)
      d1(l,lz)=d1(l,lz)-qd*d1(l,lz)*(wnd(l,lz)/b1(l,lz))/(.1+wd1(l,lz)
     1 +d1(l,lz)*wnd(l,lz)/b1(l,lz))
      hre=otbb*(1.-dige(in))*1000.*aa2(in)/agr(in,lz)
      rh1(l,1,lz)=rh1(l,1,lz)+(q10(l,lz)/qsum(kk,lz))*(up(k,lz)/otbb)
     1*hre
      wd1p=wd1(l,lz)
      b1p=b1(l,lz)
      sdp=sd(l,lz)
      sd1p=sd1(l,lz)
      wd1p=wd1(l,lz)
      wd1(l,lz)=wd1(l,lz)-qd*wd1(l,lz)/(.1+wd1(l,lz)+d1(l,lz)*wnd(l,lz)
     1/b1(l,lz))
      b1(l,lz)=b1(l,lz)-qg*b1(l,lz)/b1sd
      sd(l,lz)=sd(l,lz)-qg*sd(l,lz)/b1sd
      sd1(l,lz)=sd1(l,lz)-qg*sd1(l,lz)/b1sd
c total supplementary feed consumed, kg, estimated
      totsf(in,l,lz)=(qd+qg)*agr(in,lz)/1000.
c biomass densities bounded
      if (b1(l,lz).lt.1.) b1(l,lz)=1.
      if (wnd(l,lz).le.1.0) wnd(l,lz)=1.0
      if (sd(l,lz).le.1.0) sd(l,lz)=1.0
      if (d1(l,lz).lt.1.) d1(l,lz)=1.
      if (wd1(l,lz).lt.1.0) wd1(l,lz)=1.0
```

```
      if (sd1(l,lz).lt.1.) sd1(l,lz)=1.
c excludes woody plants
      if (sctp(l,lz)) 522,520,522
c sums total food availability from all range plants
  522 tsumb=tsumb+(b1(l,lz)+d1(l,lz)+wd1(l,lz)+sd(l,lz)--sd1(l,lz))*
     1rfrz(in,lz)
  520 continue
c ** end plant loop ***************************************************
  530 continue
c ** end region loop **************************************************
c
c ** contaminant loop ****************************************************
c initialize contaminant intake from ingestion by animal(IN)
c ***** end contaminant loop ****************************************
c
c ** region loop ******************************************************
      do 580 lz=1,ipx
      if(cnto.le.0.0)goto580
      ijj=0
c
c ** animal loop ********************************************************
      do 570 l=1,nan
      dryf=0.0
c excludes non-utilized animals
      if (autzd(in,l,lz)) 570,570,571
  571 ijj=ijj+1
      do582kk=1,inz
      if(ijj.gt.1)goto582
      c31tt(in,kk)=0.0
      c31plt(ii,in,kk)=0.0
  582 continue
      c31a=0.0
c dry fraction of animal tissue  set
      dryf=0.33
c dry fraction of insects set to 1.0
      if (alsp(l)) 573,572,573
  572 dryf=1.0
c animal tissue consumed, kgdwt, by animal units in age cohort(II)
  573 qc=0.0
      xinp=(tpop(ii,1,in)+tpop(ii,2,in))
      qc=gac*(up(ijj+3,lz)/cnto)*xinp/(dryf*1000.)
      if(qc.le.0.0)qc=0.0
c upper age index for animal group(L)
      kage=agea(l)
      wtprey=0.0
      do562ll=1,kage
      wtprey=wtprey+(tpop(ll,1,l)+tpop(ll,2,l))*wt(ll,l)
  562 continue
c
c ** age loop ********************************************************
      do 560 ll=1,kage
      tqp(ll,l)=0.0
c animal weight adjusted for predation
      tpopc=tpop(ll,1,l)+tpop(ll,2,l)
```

```
      tpopr=tpopi(ll,1,l)+tpopi(ll,2,l)
c controls prey consumption based on normal population numbers
      tpopf=tpopc/tpopr
      if(tpopf.ge.1.0)tpopf=1.0
      xins=tpopc/zoo(l)
      if(alsp(l).gt.0.0)xins=tpopc*tpopf/zoo(l)
      if(alsp(l).le.0.0.and.ainsm.lt.1.0)xins=xins*ainsm
      tqp(ll,l)=qc*xins/wtprey
c excludes insects from population dynamic losses
      if(alsp(l).gt.0.0)goto564
      wt(ll,l)=wt(ll,l)-tqp(ll,l)*wt(ll,l)
      if (wt(ll,l).le.0.0001) wt(ll,l)=0.0001
  564 do561k=1,inz
c total contaminant in/on one prey unit
      vpl=vpelt(ll,l,k)
      if(ainsm.le.0.0.and.alsp(l).le.0.0)vpl=0.0
      vbb=vbn1(ll,l,k)+vbn2(ll,l,k)
      if(ainsm.le.0.0.and.alsp(l).le.0.0)vbb=0.0
      vgg=vgut(ll,l,k)
      if(ainsm.le.0.0.and.alsp(l).le.0.0)vgg=0.0
c adjusts for consumption of larger animals
      if(wt(ll,l).lt.2.0)goto565
      vpl=0.1*vpelt(ll,l,k)
      if(ainsm.le.0.0.and.alsp(l).le.0.0)vpl=0.0
      vbb=0.1*vbb
      if(ainsm.le.0.0.and.alsp(l).le.0.0)vbb=0.0
      vgg=0.0
  565 c31a=0.0
      c31a=c31a+vbb+vgg+vpl+vlv(ll,l,k)+vms(ll,l,k)
      c31a=c31a+vkd1(ll,l,k)+vkd2(ll,l,k)
      c31a=c31a+vr(ll,l,k)+v(ll,l,k)+vfat(ll,l,k)
      c31a=c31a+(q1(ll,l,k)+q2(ll,l,k)+q3(ll,l,k))
c
c contaminant intake of one predator unit from prey/day
      wadj=0.0
      if(qc.gt.0.0) wadj=qc*xins/(xinp*wt(ll,l))
      c31tt(in,k)=c31tt(in,k)+c31a*wadj
      c31plt(ii,in,k)=c31plt(ii,in,k)+vpl*wadj
c
c** insect trachial region adjusted for predation and mortality
      if(alsp(l).gt.0.0)goto561
      q1(ll,l,k)=q1(ll,l,k)-tqp(ll,l)*q1(ll,l,k)
      q2(ll,l,k)=q2(ll,l,k)-tqp(ll,l)*q2(ll,l,k)
      q3(ll,l,k)=q3(ll,l,k)-tqp(ll,l)*q3(ll,in,k)

      if (q1(ll,l,k).le.0.0) q1(ll,l,k)=0.0
      if (q2(ll,l,k).le.0.0) q2(ll,l,k)=0.0
      if (q3(ll,l,k).le.0.0) q3(ll,l,k)=0.0
c**hemolymph adjusted for predation
      v(ll,l,k)=v(ll,l,k)-tqp(ll,l)*v(ll,l,k)
c
      if(v(ll,l,k).le.0.0) v(ll,l,k)=0.0
c
c** organs contaminant contents corrected for predation
```

```
      vbn1(ll,l,k)=vbn1(ll,l,k)-tqp(ll,l)*vbn1(ll,l,k)
      vbn2(ll,l,k)=vbn2(ll,l,k)-tqp (ll,l)*vbn2(ll,l,k)
      vlv(ll,l,k)=vlv(ll,l,k)-tqp(ll,l)*vlv(ll,l,k)
      vms(ll,l,k)=vms(ll,l,k)-tqp(ll,l)*vms(ll,l,k)
      vkd1(ll,l,k)=vkd1(ll,l,k)-tqp(ll,l)*vkd1(ll,l,k)
      vkd2(ll,l,k)=vkd2(ll,l,k)-tqp(ll,l)*vkd2(ll,l,k)
      vr(ll,l,k)=vr(ll,l,k)-tqp(ll,l)*vr(ll,l,k)
      vfat(ll,l,k)=vfat(ll,l,k)-tqp(ll,l)*vfat(ll,l,k)
c
      if(vbn1(ll,l,k).le.0.0)vbn1(ll,l,k)=0.0
      if(vbn2(ll,l,k).le.0.0)vbn2(ll,l,k)=0.0
      if(vlv(ll,l,k).le.0.0)vlv(ll,l,k)=0.0
      if(vms(ll,l,k).le.0.0)vms(ll,l,k)=0.0
      if(vkd1(ll,l,k).le.0.0)vkd1(ll,l,k)=0.0
      if(vkd2(ll,l,k).le.0.0)vkd2(ll,l,k)=0.0
      if(vr(ll,l,k).le.0.0) vr(ll,l,k)=0.0
      if(vfat(ll,l,k).le.0.0)vfat(ll,l,k)=0.0
  561 continue
  560 continue
c ** end age loop ******************************************************
  570 continue
c**** end of animal loop*************************************************
c
c animal excreta is assigned to range plants proportionately
c
c ** plant loop ********************************************************
      do 550 kk=1,ipl
c non-utilized and non-range plants excluded
      if (putzd(in,kk,lz)) 552,550,552
  552 if (putzd(in,kk,lz).gt.3.0) go to 550
      jj=putzd(in,kk,lz)
c animal feces is converted to g/m2 dry
      hre=gac*(1.-dige(in))*1000*dryf*aa2(in)/agr(in,lz)
c insect mortalities(summrt)distributed as is feces
      if(alsp(in).le.0.0) hre=hre+summrt(in,1)+summrt(in,2)
c null consumption(OTBB); and null specific(QSUM) plant group excluded
      if (otbb.le.0.0.or.qsum(jj,lz).le.0.0) go to 550
c humus compartment incremented for feces conversion
      rh1(kk,1,lz)=rh1(kk,1,lz)+hre*(up(jj,lz)/otbb)*(q10(kk,lz)/qsum(jj
     1 ,lz))
c fraction of specific plant type in diet
      hra=(up(ijj+3,lz)/otbb)*(up(jj,lz)/otbb)*(q10(kk,lz)/qsum(jj,lz))
c invertebrate minerals are recycled to humus
      do 540 k=1,inz
      if(alsp(in).gt.0.0)hemlos(in,k,lz)=0.0
      z5(kk,k,lz)=z5(kk,k,lz)+hemlos(in,k,lz)*hra
      if(z5(kk,k,lz).le.0.0)z5(kk,k,lz)=0.0
      humus(in,k,lz)=0.0
  540 continue
c ** end contaminant loop ***********************************************
  550 continue
c ** end plant loop *****************************************************
  580 continue
c ** end region loop ****************************************************
```

```
 590 continue
c total zoon units in age cohort(II) of zoon type(IN)
    zott=1.0
    if(alsp(in).gt.0.0) zott=tpop(ii,1,in)+tpop(ii,2,in)
c
c ** contaminant loop ***************************************************
    do 610 k=1,inz
c contaminant uptake from soil transported to blood, per day initialized
    c66(k)=0.0
c total contaminant ingested from soil consumption initialized, per day
    dburd(k)=0.0
c contaminant transport from ingested food transported to blood
c initialized, per day
    c99(k)=0.0
c contaminant uptake due to water consumption, per day , initialized
    c56(k)=0.0
c contaminant uptake from lung to blood via GI , per day , initialized
    al4(k)=0.0
c contaminant uptake from lung pulmonary to blood, per day, initialized
    al5(k)=0.0
c contaminant uptake inhalation deposit entering GI tract directly,
c per day, initialized
    car(k)=0.0
    wburd(k)=0.0
c
c ** region loop ********************************************************
    do 600 lz=1,ipx
c daily contaminant transport to humus from all sources, all regions
    humus(in,k,lz)=humus(in,k,lz)+(1.-gtbl(in,k))*vgut(ii,in,k)*
   1rfrz(in,lz)*zott*(sptfrr(lz)+sptfrf(lz)+sptfra(lz))/agr(in,lz)
    if(humus(in,k,lz).le.0.0)humus(in,k,lz)=0.0
c humus incremented for contaminant losses via urine, per day, all
c regions
    humus(in,k,lz)=humus(in,k,lz)+vurin(ii,in,k)*rfrz(in,lz)*
   1zott*(sptfrr(lz)+sptfrf(lz)+sptfra(lz))/agr(in,lz)
    if(humus(in,k,lz).le.0.0)humus(in,k,lz)=0.0
c
    hemlos(in,k,lz)=0.0
c daily contaminant entering GI from food ingestion, all regions
    c31tt(in,k)=c31tt(in,k)+c31(in,k,lz)
c daily contaminant entering GI from drinking water, all regions
    wburd(k)=watcnr(in,k,lz)*watupt*rfrz(in,lz)*1000.*(sptfrr(lz)+
   1sptfrf(lz)+sptfra(lz))+wburd(k)
    if(alsp(in).le.0.0.and.ainsm.le.0.0)wburd(k)=0.0
 600 continue
c ** end region loop ****************************************************
c daily contaminant blood uptake from drinking water, all regions
    c56(k)=wburd(k)*gtbl(in,k)
 610 continue
c ** end contaminant loop ***********************************************
c pelt mass, grams, estimated to be 15% of total body mass,kg
    vplt(ii,in)=150.0*wt(ii,in)
c
c estimates daily contaminant uptake from soil, gdwt/day, all regions
```

```
c excludes null biomass density, g/m2, available for consumption
      c55=3.0*wt(ii,in)
c estimates soil consumption, gdwt/day, while grazing; bounded
      if(tsumb.gt.20.0)c55=3.0*wt(ii,in)*20.0/tsumb
      if(tsumb.le.0.0)c55=0.0
      if(alsp(in).le.0.0.and.ainsm.le.0.0)c55=0.0
      tpopc=tpop(ii,1,in)+tpop(ii,2,in)
      tpopr=tpopi(ii,1,in)+tpopi(ii,2,in)
      tpopf=tpopc/tpopr
      if(tpopf.ge.1.0)tpopf=1.0
c
c ** region loop ********************************************************
      do 640 lz=1,ipx
c
c estimates daily contaminant uptake from soil, gdwt/day, all regions
c ** plant loop *********************************************************
      do 630 jk=1,ipl
c ** contaminant loop ***************************************************
      do 661 k=1,inz
      if(jk.eq.1.and.lz.eq.1)vpelt(ii,in,k)=0.0
c estimates soil contamination of pelt to be 1/3 that of top soil layer
      vpe=vplt(ii,in)*z1(jk,1,k,lz)*rfrz(in,lz)*acres(jk,lz)/
     1(tacres(lz)*3.0)
      if(alsp(in).gt.0.0)vpe=vpe*tpopf
      if(ainsm.lt.1.0.and.alsp(in).le.0.0)vpe=vpe*ainsm
      vpelt(ii,in,k)=vpelt(ii,in,k)+vpe*(sptfrr(lz)+sptfrf(lz)
     1+sptfra(lz))
  661 continue
c ** end contaminant loop ***********************************************
c
      if(tsumb.le.0.0)goto660
c excludes non-utilized plants, and non-leafy feeds
      if (putzd(in,jk,lz)) 631,630,631
  631 if (putzd(in,jk,lz).ge.4.0) go to 630
c excludes woody plants
      if (sctp(jk,lz)) 630,630,633
c
c estimates mass of top 20 mm of soil, gdwt/m2
  633 soilms=slice(jk,1,lz)*1400.
c
c ** contaminant loop ***************************************************
      do 620 k=1,inz
c estimates daily contaminant uptake to bloodstream from soil
      c66(k)=c55*gtbl(in,k)*z1(jk,1,k,lz)*acres(jk,lz)*sptfrr(lz)+c66(k)
     1/tacres(lz)
  620 continue
c ** end contaminant loop ***********************************************
  630 continue
c ** end plant loop *****************************************************
  640 continue
c ** end region loop ****************************************************
c
c ** contaminant loop *****************************************************
      do 650 k=1,inz
```

```
c daily contaminant soil consumption estimated
      dburd(k)=c66(k)/gtbl(in,k)
 650 continue
c ** end contaminant loop ***********************************************
 660 continue
c
c ** contaminant loop ****************************************************
 703  do 740 k=1,inz
      if(agem(ii,in).le.0.0)goto740
c excludes insects and upper age groups of other animals
      if(alsp(in).le.0.0.or.ii.gt.1)goto707
c
c gestation stage still active
      if(aiiir(in,1).gt.aiii)goto 704
      if(gstn(in).lt.2.0)goto707
      if(aiiir(in,2).gt.aiii)goto704
c
c**** mean daily milk concentration initialized
 707 if(ii.eq.1) w(18,in,k)=0.0
      car(k)=0.0
      pburd(k)=0.0
      qae=0.0
      sptttt=0.0
      do682 lz=1,ipx
      sptttt=sptttt+sptfrr(lz)+sptfrf(lz)+sptfra(lz)
 682 continue
c **region loop ********************************************************
      do 680 lz=1,ipx
c contaminant disintegration rate, per year
      disintz=disint(k)
      if (lz.gt.1) disintz=0.0
c daily increment from particulates deposited in N-P region adjusted for
c losses transported to GI; bounded
      sptfrt=sptfrr(lz)+sptfrf(lz)+sptfra(lz)
      q1(ii,in,k)=q1(ii,in,k)+plcm1(in)*a1tt(k,lz)*rsrt(ii,in)*
     1rfrz(in,lz)*sptfrt/sptttt
      if(lz.eq.1)q1(ii,in,k)=q1(ii,in,k)-disintz*q1(ii,in,k)-q1(ii,in,k)
     1*pltg1(in,k)
c
c same for T-B region adjusted for losses to GI; bounded
      q2(ii,in,k)=q2(ii,in,k)+plcm2(in)*a1tt(k,lz)*rsrt(ii,in)*
     1rfrz(in,lz)*sptfrt/sptttt
      if(lz.eq.1)q2(ii,in,k)=q2(ii,in,k)-disintz*q2(ii,in,k)-q2(ii,in,k)
     1*pltg2(in,k)
c
c same for Pulmonary region adjusted for losses to blood; bounded
      q3(ii,in,k)=q3(ii,in,k)+plcm3(in)*a1tt(k,lz)*rsrt(ii,in)*
     1rfrz(in,lz)*sptfrt/sptttt
      if(lz.eq.1)q3(ii,in,k)=q3(ii,in,k)-disintz*q3(ii,in,k)-q3(ii,in,k)
     1*pltbl(in,k)
c
      if (q1(ii,in,k).le.0.0) q1(ii,in,k)=0.0
      if (q2(ii,in,k).le.0.0) q2(ii,in,k)=0.0
      if (q3(ii,in,k).le.0.0) q3(ii,in,k)=0.0
```

```
      if(ainsm.le.0.0.and.alsp(in).le.0.0)q1(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)q2(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)q3(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vkd1(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vkd2(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vbn1(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vbn2(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vlv(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)v(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vfat(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vr(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vms(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vpelt(ii,in,k)=0.0
      if(ainsm.le.0.0.and.alsp(in).le.0.0)vgut(ii,in,k)=0.0
c
c daily contaminant blood uptake from GI derived from particulates
      pgi=rsrt(ii,in)*a1tt(k,lz)*rfrz(in,lz)*0.38
      car(k)=pgi*gtbl(in,k)*sptfrt/spttt+car(k)
c daily contaminant deposition in lungs, all regions
      pburd(k)=a1tt(k,lz)*rfrz(in,lz)*rsrt(ii,in)*(plcm1(in)+
     1plcm2(in)+plcm3(in)+1.52*rfrz(in,lz))*sptfrt/spttt+pburd(k)
c
c contaminant vapor exchange between air and venous blood weighted by
c regional occupancy
      if(pltblg(in,k).le.0.0)goto680
      if(alsp(in).le.0.0)goto680
      if(wt(ii,in).le.0.0)goto680
      wtbl(ii,in)=5.5e-05*(1000.*wt(ii,in))**0.99
      qae=qae+(rsrt(ii,in)*a1tg(k,lz)+flopls*(v(ii,in,k)/wtbl(ii,in))
     1*rfrz(in,lz))*sptfrt/spttt
  680 continue
c ** end region loop ***************************************************
c
c daily contaminant transport to blood from ingested food
      c99(k)=c31tt(in,k)*gtbl(in,k)
      if(alsp(in).le.0.0.and.ainsm.le.0.0)c99(k)=0.0
c
c daily contaminant transport to blood from N-P and T-B regions via
c the GI
      al4(k)=q1(ii,in,k)*pltg1(in,k)*gtbl(in,k)+q2(ii,in,k)*pltg2(in,k)
     1 *gtbl(in,k)
      if(alsp(in).le.0.0.and.ainsm.le.0.0)al4(k)=0.0
      if(alsp(in).le.0.0.and.ainsm.le.0.0)car(k)=0.0
c
      if(pltblg(in,k).le.0.0)goto681
      if(alsp(in).le.0.0)goto681
      qbe=0.0
      if(pltblg(in,k).gt.0.0) qbe=flopls+rsrt(ii,in)*1000./pltblg(in,k)
c arterial blood concentration estimated after contaminant exchange
      if(qbe.gt.0.0) v(ii,in,k)= (qae/qbe)*wtbl(ii,in)
      if(v(ii,in,k).le.0.0)v(ii,in,k)=0.0
c soil ingestion while feeding on plants
  681 vsol(ii,in,k)=c66(k)/gtbl(in,k)
c lung clearance of particulates into gut
```

```
      vlug(ii,in,k)=(car(k)+al4(k))/gtbl(in,k)
      vgut(ii,in,k)=(car(k)+c99(k)+al4(k)+c66(k)+c56(k))/gtbl(in,k)
c daily contaminant transport to arterial blood from pulmonary region
      al5(k)=q3(ii,in,k)*pltbl(in,k)
      if(alsp(in).le.0.0.and.ainsm.le.0.0)al5(k)=0.0
      v(ii,in,k)=v(ii,in,k)+al5(k)
c arterial blood corrected for decay or persistence
      v(ii,in,k)=v(ii,in,k)-disint(k)*v(ii,in,k)
c
      if(v(ii,in,k).le.0.0) v(ii,in,k)=0.0
c total milk production liters/day , producer age groups
      agemi=agem(ii,in)
      wmk(in)=0.0
c estimates mature age fraction of 1 year-olds based on 3 sub-divisions
      fracs(in)=a(ii,in,2)
      if(ii.ne.1)goto712
      off2t=0.0
      off2i=0.0
      do713ig1=1,3
      off2t=off2t+offsp(in,ig1,2)
      if(affsp(in,ig1).ge..15*alsp(in).and.affsp(in,ig1).le..7*alsp(in))
     1off2i=off2i+offsp(in,ig1,2)
  713 continue
      fracs(in)=0.0
      if(off2t.gt.0.0) fracs(in)=a(ii,in,2)*off2i/off2t
  712 if(agemi.ge.0.15*alsp(in).and.agemi.le.0.7*alsp(in))
     1 wmk(in)=wmk(in)+flomlk*fracs(in)
c cohort animal type index
      j=in
c contaminant atomic number index
      ian=atno(k)
c
c ** region loop ************************************************************
      do 710 lz=1,ipx
c ** plant loop *************************************************************
      do 700 l=1,ipl
c excludes non-utilized and non-range plants
      if (putzd(in,l,lz)) 701,700,701
  701 if (putzd(in,l,lz).gt.3.0) go to 700
      kk=putzd(in,l,lz)
c excludes empty food categories and null intake by animal group(IN)
      if (qsum(kk,lz).le.1.0) go to 700
      if (ototbt.le.0.0) go to 700
c regional humus contaminant compartments incremented for non-absorbed
c and excreted material, per m2-day
      z5(l,k,lz)=z5(l,k,lz)+humus(j,k,lz)*(q10(l,lz)/qsum(kk,lz))*(up(kk
     1 ,lz)/ototbt)
      if(z5(l,k,lz).le.0.0)z5(l,k,lz)=0.0
  700 continue
c ** end plant loop *********************************************************
  710 continue
c ** end region loop ********************************************************
c bone(2),liver, muscle, and kidneys(2) adjusted for contaminant decay
      l=ii
```

```
c contaminants corrected for decay or persisitence
      vbn1(l,j,k)=vbn1(l,j,k)-disint(k)*vbn1(l,j,k)
      vbn2(l,j,k)=vbn2(l,j,k)-disint(k)*vbn2(l,j,k)
      vlv(l,j,k)=vlv(l,j,k)-vlv(l,j,k)*disint(k)
      vms(l,j,k)=vms(l,j,k)-disint(k)*vms(l,j,k)
      vkd1(l,j,k)=vkd1(l,j,k)-disint(k)*vkd1(l,j,k)
      vkd2(l,j,k)=vkd2(l,j,k)-disint(k)*vkd2(l,j,k)
      vr(l,j,k)=vr(l,j,k)-disint(k)*vr(l,j,k)
      vfat(l,j,k)=vfat(l,j,k)-disint(k)*vfat(l,j,k)
      v(l,j,k)=v(l,j,k)-disint(k)*v(l,j,k)
      q1(l,j,k)=q1(l,j,k)-disint(k)*q1(i,j,k)
      q2(l,j,k)=q2(l,j,k)-disint(k)*q2(i,j,k)
      q3(l,j,k)=q3(l,j,k)-disint(k)*q3(i,j,k)
c
c daily blood contaminant content updated for losses;transport to fetal
c blood assumed equivalent to blood transport to milk
      vmkc(j)=0.0
c** excludes immature or old age female mammals from milk production
      agemi=agem(l,j)
      if(l.ne.1)goto714
c      if(iage.ge.6)goto714
      off2i=0.0
      do715ig1=1,3
      if(affsp(j,ig1).ge..15*alsp(j).and.affsp(j,ig1).le..7*alsp(j))
     1off2i=off2i+offsp(j,ig1,2)
  715 continue
      if(off2i.gt.0.0.and.agemi.lt.0.16*alsp(j))agemi=0.16*alsp(j)
c
  714 if(agemi.lt.0.15*alsp(j).or.agemi.gt.0.7*alsp(j))goto716
      vmkc(j)=1.0
c ** excludes milk production during non-reproductive period
      if(aiiir(j,1).lt.gst(j,1)+14.)vmkc(j)=0.0
c ** excludes milk production after weaning period
      if(aiiir(j,1).gt.cagm(j,1)+0.025*alsp(j)) vmkc(j)=0.0
      if(gstn(j).lt.2.0)goto716
      if(aiiir(j,2).le.cagm(j,1)+0.025*alsp(j))goto716
      vmkc(j)=1.0
      if(aiiir(j,2).lt.gst(j,2)+14.)vmkc(j)=0.0
      if(aiiir(j,2).gt.cagm(j,2)+0.025*alsp(j))vmkc(j)=0.0
  716 vmk(l,j,k)=vmkc(j)*vmk(l,j,k)
c ** excludes insects
      if(alsp(in).le.0.0)vmk(l,j,k)=0.0
c
c arterial contaminant content
      viiin(k)=v(l,j,k)
      dbo=0.0
      vmk(l,j,k)=0.0
      vurin(l,j,k)=0.0
c assumes hourly major perfusions(passes)through system daily
      do719it=1,24
c arterial perfusion of organs with contaminant
c daily bone(2) contaminant content updated for gains from arterial
c blood
      dbn2=0.0
```

```
      if(bntbl2(j,k).le.0.0)goto333
      f0=2.0*flomar
      f2=viiin(k)/volpls
      f3=vbn2(l,j,k)/(volmar*bntbl2(j,k))
      dbn2=f0*(f2-f3)/24.
c same for bone(1) compartment
c
  333 dbn1=0.0
      if(bntbl1(j,k).le.0.0)goto334
      f0=2.0*flomar
      f2=viiin(k)/volpls
      f3=vbn1(l,j,k)/(volmar*bntbl1(j,k))
      dbn1=f0*(f2-f3)/24.
      dbo=f0*f2/24.
c
c same for muscle contaminant content
  334 dbms=0.0
      if(mstbl(j,k).le.0.0)goto335
      f0=flomus
      f2=viiin(k)/volpls
      f3=vms(l,j,k)/(volmus*mstbl(j,k))
      dbms=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c same for kidney(2) compartment
  335 dbk2=0.0
      if(kdtbl2(j,k).le.0.0)goto336
      f0=flokid
      f2=viiin(k)/volpls
      f3=vkd2(l,j,k)/(volkid*kdtbl2(j,k))
      dbk2=f0*(f2-f3)/24.
c
c adipose tissue content updated for gains from arterial blood
  336 dbft=0.0
      if(fttbl(j,k).le.0.0)goto337
      f0=flofat
      f2=viiin(k)/volpls
      f3=vfat(l,j,k)/(volfat*fttbl(j,k))
      dbft=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c residual soft tissue content updated for gain from arterial blood
c and losses to mammary glands and/or milk production
  337 dbrs=0.0
      dbmk=0.0
      if(rstbl(j,k).le.0.0)goto338
      f0=flores
      f2=viiin(k)/volpls
      f3=vr(l,j,k)/(volres*rstbl(j,k))
      dbrs=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltmk(j,k).le.0.0)goto338
      f2=viiin(k)*bltmk(j,k)
      dbmk=f0*f2/(flopls*24.)
```

```
c     if(vmk(l,j,k).le.0.0)dbmk=0.0
      if(dbmk.le.0.0)dbmk=0.0
c
c kidney(1) contaminant content updated for gains from arterial
c blood and losses to urine
 338 dbk1=0.0
      dbu=0.0
      if(kdtbl1(j,k).le.0.0)goto339
      f0=flokid
      f2=viiin(k)/volpls
      f3=vkd1(l,j,k)/(volkid*kdtbl1(j,k))
      dbk1=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltur(j,k).le.0.0)goto339
      f2=viiin(k)*bltur(j,k)
      dbu=f0*f2/(flopls*24.)
      if(dbu.le.0.0)dbu=0.0
      if(alsp(j).le.0.0.and.ainsm.le.0.0)dbu=0.0

c
c liver contaminant content updated for gains from arterial blood
c and gut absorption
 339 dblv=0.0
      dbi=0.0
      if(lvtbl(j,k).le.0.0)goto704
      cntki(j)=1.0
      if(l.ne.1)goto717
      if(cblc(j).ge.1.0)cntki(j)=0.0
      if(gstn(j).lt.2.0)goto717
      if(aiiir(j,2).gt.cagm(j,2))cntki(j)=(cntk(1)+cntk(2))/2.
 717 dbi=(al4(k)+car(k)+cntki(j)*(c99(k)+c66(k)+c56(k)))/24.
      if(l.eq.1)dbi=dbi+cbw(j,k)*gtbl(l,j)*0.1*wt(l,j)/24.
      f0=floliv
      f2=viiin(k)/volpls
      vlv(l,j,k)=vlv(l,j,k)+dbi
      f3=vlv(l,j,k)/(volliv*lvtbl(j,k))
      dblv=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(dbo.le.0.0)goto719
      flxbn2=dbn2*viiin(k)/dbo
      brto=0.99*vbn2(l,j,k)
      if(flxbn2.lt.0.0.and.-flxbn2.gt.brto)flxbn2=-brto
      vbn2(l,j,k)=vbn2(l,j,k)+flxbn2
      viiind=viiin(k)-flxbn2
      if(viiind.le.0.0)viiind=0.0
      if(vbn2(l,j,k).le.0.0)vbn2(l,j,k)=0.0
      flxbn1=dbn1*viiin(k)/dbo
      brto=0.99*vbn1(l,j,k)
      if(flxbn1.lt.0.0.and.-flxbn1.gt.brto)flxbn1=-brto
      if(bntbl2(j,k).le.0.0)viiind=viiin(k)
      vbn1(l,j,k)=vbn1(l,j,k)+flxbn1*viiind/viiin(k)
      if(flxbn2.lt.0.0)vbn1(l,j,k)=vbn1(l,j,k)-flxbn2
      if(flxbn2.lt.0.0)flxbn2=0.0
      if(vbn1(l,j,k).le.0.0)vbn1(l,j,k)=0.0
```

```
      flxms=dbms*viiin(k)/dbo
      brto=0.99*vms(l,j,k)
      if(flxms.lt.0.0.and.-flxms.gt.brto)flxms=-brto
      vms(l,j,k)=vms(l,j,k)+flxms
      if(vms(l,j,k).le.0.0)vms(l,j,k)=0.0
      flxkd2=dbk2*viiin(k)/dbo
      brto=0.99*vkd2(l,j,k)
      if(flxkd2.lt.0.0.and.-flxkd2.gt.brto)flxkd2=-brto
      vkd2(l,j,k)=vkd2(l,j,k)+flxkd2
      viiind=viiin(k)-flxkd2
      if(viiind.le.0.0)viiind=0.0
      if(vkd2(l,j,k).le.0.0)vkd2(l,j,k)=0.0
      flxkd1=dbk1*viiin(k)/dbo
      brto=0.99*vkd1(l,j,k)
      if(flxkd1.lt.0.0.and.-flxkd1.gt.brto)flxkd1=-brto
      if(kdtbl2(j,k).le.0.0)viiind=viiin(k)
      vkd1(l,j,k)=vkd1(l,j,k)+flxkd1*viiind/viiin(k)
      if(flxkd2.lt.0.0)vkd1(l,j,k)=vkd1(l,j,k)-flxkd2
      if(flxkd2.lt.0.0)flxkd2=0.0
      if(vkd1(l,j,k).le.0.0)vkd1(l,j,k)=0.0
      flxft=dbft*viiin(k)/dbo
      brto=0.99*vfat(l,j,k)
      if(flxft.lt.0.0.and.-flxft.gt.brto)flxft=-brto
      vfat(l,j,k)=vfat(l,j,k)+flxft
      if(vfat(l,j,k).le.0.0)vfat(l,j,k)=0.0
      flxrs=dbrs*viiin(k)/dbo
      brto=0.99*vr(l,j,k)
      if(flxrs.lt.0.0.and.-flxrs.gt.brto)flxrs=-brto
      vr(l,j,k)=vr(l,j,k)+flxrs
      if(vr(l,j,k).le.0.0)vr(l,j,k)=0.0
      vmk(l,j,k)=vmk(l,j,k)+dbmk
      if(vmkc(j).le.0.0)dbmk=0.0
      vurin(l,j,k)=vurin(l,j,k)+dbu
      flxlv=dblv*viiin(k)/dbo
      flxlvp=flxlv
      brto=0.99*vlv(l,j,k)
      if(flxlv.lt.0.0.and.-flxlv.gt.brto)flxlv=-brto
      vlv(l,j,k)=vlv(l,j,k)+flxlv
      if(vlv(l,j,k).le.0.0)vlv(l,j,k)=0.0
      v(l,j,k)=v(l,j,k)-flxbn1-flxbn2-flxkd1-flxkd2
     1-flxft-flxrs-flxlv-flxms-dbu-dbmk
      if(v(l,j,k).le.0.0)v(l,j,k)=0.0
      viiin(k)=v(l,j,k)
      dbo=0.0
  719 continue
c
c *** factors for excluding animals in early age groups as milk producers
c *** and reproducers ***************************************************
c
c ** excludes insects from milk production and reproduction
  704 if(alsp(in).le.0.0)goto702
c
      afmt(in)=0.0
      iage=agea(in)
```

```
      lf=iage+1
      l=ii
      j=in
      if(l.eq.1)gsfc(j)=0.0
      if(l.eq.1)gsst(j)=0.0
c
c** excludes non-reproductive period; allows 2 weeks for mating and
c** embryo development
      if(aiiir(j,1).lt.gst(j,1)+14.)cbf(j,k)=0.0
      if(aiiir(j,1).lt.gst(j,1)+14.)cbp(l,j,k)=0.0
      if(aiiir(j,1).lt.gst(j,1)+14.)goto702
      if(gstn(j).lt.2.0)goto801
      if(aiiir(j,2).le.cagm(j,1)+0.025*alsp(j))goto801
      if(aiiir(j,2).lt.gst(j,2)+14.)cbf(j,k)=0.0
      if(aiiir(j,2).lt.gst(j,2)+14.)cbp(l,j,k)=0.0
      if(aiiir(j,2).lt.gst(j,2)+14.)goto702
c
c ** excludes period after weaning
  801 if(aiiir(j,1).le.cagm(j,1)+0.025*alsp(j))goto802
      if(gstn(j).lt.2.0)cbw(j,k)=0.0
      if(gstn(j).lt.2.0)goto702
      if(aiiir(j,2).gt.cagm(j,2)+0.025*alsp(j))cbw(j,k)=0.0
      if(aiiir(j,2).gt.cagm(j,2)+0.025*alsp(j))goto702
c
c** excludes immature animals from milk production and reproduction
  802 agemi=agem(l,j)
      fracs(j)=1.0
      if(l.ne.1)goto721
c       if(iage.ge.6)goto721
      off2t=0.0
      off2i=0.0
      do722ig1=1,3
      off2t=off2t+offsp(j,ig1,2)
      if(affsp(j,ig1).ge..15*alsp(j).and.affsp(j,ig1).le..7*alsp(j))
     1off2i=off2i+offsp(j,ig1,2)
  722 continue
      fracs(j)=0.0
      if(off2t.gt.0.0) fracs(j)=off2i/off2t
      if(fracs(j).gt.0.0.and.agemi.lt.0.16*alsp(j))agemi=0.16*alsp(j)
c
  721 if(agemi.le.0.15*alsp(j))goto702
      gsst(j)=gsst(j)+1.0
      lua(j)=0
      if(agem(l+1,j).ge.0.7*alsp(j))lua(j)=l
      if(l.eq.iage)lua(j)=l
      if(agemi.gt.0.7*alsp(j))gsfc(j)=gsfc(j)+1.0
      if(agemi.gt.0.7*alsp(j))goto902
c
c ** estimates weighted average fraction of producer groups
      do711ia=1,iage
      ai=ia
      ifm(ia)=agemi-0.15*alsp(j)/(0.55*alsp(j)/9.0)
      if(ifm(ia).lt.1)ifm(ia)=1
      if(ifm(ia).gt.8)ifm(ia)=8
```

```
      if(ia.eq.l)ifa=ifm(ia)
      ifb=ifm(ia)
      afmt(j)=afmt(j)+sfd(ifb,j)*fracs(j)
  711 continue
c
c** initiates fetus production during Spring/Fall time frame
c
c *** weighted parental contaminant addition to fetal blood ***
c
c *** changes transport from fetal blood to GI(liver) after birth
      cblc(j)=1.0
      if(aiiir(j,1).gt.cagm(j,1)) cblc(j)=0.0
      if(aiiir(j,1).gt.cagm(j,1)) aiif(j,1)=0.0
      if(gstn(j).lt.2.0)goto803
      if(aiiir(j,1).le.cagm(j,1)+0.025*alsp(j))goto803
      cblc(j)=1.0
      if(aiiir(j,2).gt.cagm(j,2)) cblc(j)=0.0
      if(aiiir(j,2).gt.cagm(j,2)) aiif(j,2)=0.0
c
c contaminant transport to fetus blood
  803 if(cblc(j).le.0.0)v(lf,j,k)=0.0
      if(cblc(j).le.0.0)cbf(j,k)=0.0
      if(cblc(j).ge.1.0)cbw(j,k)=0.0
      if(gsst(j).gt.1.0)goto903
      v(lf,j,k)=0.0
      vbn1(lf,j,k)=0.0
      vbn2(lf,j,k)=0.0
      vkd1(lf,j,k)=0.0
      vkd2(lf,j,k)=0.0
      vms(lf,j,k)=0.0
      vlv(lf,j,k)=0.0
      vr(lf,j,k)=0.0
      vurin(lf,j,k)=0.0
      vfat(lf,j,k)=0.0
      q1(lf,j,k)=0.0
      q2(lf,j,k)=0.0
      q3(lf,j,k)=0.0
      rkd(lf,j,k)=0.0
      rbn(lf,j,k)=0.0
      rndo(lf,j,k)=0.0
      rlv(lf,j,k)=0.0
      rlu(lf,j,k)=0.0
      rms(lf,j,k)=0.0
c contaminant increment to fetal blood from mother;weighted concentration
  903 vmilk=1.1915e-03*(wt(l,j)*1000.)**0.78
      cbf(j,k)=cbf(j,k)+cblc(j)*vmk(l,j,k)*sfd(ifa,j)*fracs(j)/(afmt(j)
     1*vmilk)
c
c contaminant transport via milk during weaning,weighted concentration
      cbw(j,k)=cbw(j,k)+(1.0-cblc(j))*vmk(l,j,k)*sfd(ifa,j)*fracs(j)/
     1(afmt(j)*vmilk)
c
c weighted addition of fetal metabolic wastes to parent,weighted conc.
      cbp(l,j,k)=cblc(j)*vurin(lf,j,k)*sfd(ifa,j)*fracs(j)/afmt(j)
```

```
      v(l,j,k)=v(l,j,k)+cblc(j)*vurin(lf,j,k)*sfd(ifa,j)*fracs(j)
     1/afmt(j)
      goto702
  902 if(gsfc(j).gt.1.0)goto702
      if(cblc(j).le.0.0)goto702
      icagm1=cagm(j,1)
      iiir1=aiiir(j,1)
      iiir2=0
      icagm2=0
      igst2=gst(j,2)
      iiir2=aiiir(j,2)
      icagm2=cagm(j,2)
       if(l.ne.lua(j))goto702
c fetus index counter
      lagf(j)=lagf(j)+1
      if(lagf(j).gt.1)goto702
c fetal age, days
      if(iiir1.lt.icagm1) aiif(j,1)=aiif(j,1)+1.0
      if(iiir2.gt.icagm1.and.iiir2.lt.icagm2)
     1aiif(j,2)=aiif(j,2)+1.0
      if(aiif(j,1).gt.gestp(j))aiif(j,1)=gestp(j)
      if(aiif(j,2).gt.gestp(j))aiif(j,2)=gestp(j)
      aiifg=aiif(j,1)
      if(aiif(j,2).ge.1.0)aiifg=aiif(j,2)
c
c ** estimated fetus live weight and organ/tissue weights from beginning
c ** of growth period
      wt(lf,j)=0.05256*rr(j)*(1.0-exp(-9.*(aiifg/gestp(j))))
      if(wt(lf,j).le.0.001)wt(lf,j)=0.001
c ** estimated metabolic biomass of fetus
      wtgrm=wt(lf,j)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
      amw(lf,j)=5.623e-03*wtgrm**0.75
c
c** contaminant content of fetus organs estimated
c
c fetal total body water, liters
      tbwlf=9.077e-04*wtgrm**0.963
c
c fetal cardiac output, liters/day
      cardrf=1.44*0.7081*wtgrm**0.79
c
c** compartment volumes **
c
c fetal body weight, grams
      wtgrm=wt(lf,in)*1000.
c fetal body weight limited to a minimum of 1 gram
      if(wtgrm.lt.1.0)wtgrm=1.0
c
c fetal plasma vol, liters
      volplf=4.29e-05*wtgrm**0.992
c fetal muscle vol, liters
      volmuf=4.63e-04*wtgrm**1.009
c fetal kidney vol, liters
```

```
      volkif=2.18e-05*wtgrm**0.843
c fetal liver vol, liters
      vollif=8.59e-05*wtgrm**0.885
c fetal gut vol, liters
      volguf=8.65e-05*wtgrm**0.909
c fetal gut lumen vol, liters
      vollum=5.78e-05*wtgrm**0.934
c fetal heart vol, liters
      volhrf=2.0e-06*wtgrm**1.043
c fetal lungs vol, liters
      volluf=3.16e-06*wtgrm**1.104
c fetal spleen vol, liters
      volspf=4.52e-06*wtgrm**0.901
c fetal marrow vol, liters
      volmaf=3.3e-05*wtgrm**0.936
c fetal fat vol, liters
       volfaf=1.1e-04*wtgrm
c fetal current residual volume
c      volref=volguf+volhrf+volspf
       volref=volguf+volhrf+volspf+volluf
c
c ** fetal plasma flow rates **
c
c fetal plasma flow, liters/day
      floplf=1.44*.601*wtgrm**0.778
c fetal muscle flow, liters/day
      flomuf=1.44*.0817*wtgrm**0.81
c fetal kidney flow, liters/day
      flokif=1.44*0.0845*wtgrm**0.802
c fetal liver flow, liters/day
      flolif=1.44*0.0944*wtgrm**0.792
c fetal gut flow, liters/day
      floguf=1.44*0.102*wtgrm**0.787
      flolif=flolif+0.9*floguf
c fetal heart flow, liters/day
      flohrf=1.44*5.77e-03*wtgrm**0.965
c fetal lung flow, liters/day
      floluf=1.44*0.215*wtgrm**0.82
c fetal spleen flow, liters/day
      flospf=1.44*1.79e-03*wtgrm**1.028
c fetal marrow flow, liters/day
      flomaf=1.44*0.0164*wtgrm**0.804
c fetal fat flow
      flofaf=0.75*flomuf
c ** fetal flow to residual tissues
c      floref=flohrt+flospl
       floref=flohrf+flospf+0.1*floluf+0.1*floguf
c fetal metabolic waste production, liters/day
       flourf=0.024*.0064*wtgrm**0.82
c
c  fetal arterial blood content
       blmthr=0.2*cbf(j,k)*wt(lf,j)
       viiin(k)=v(lf,j,k)
       if(viiin(k).le.blmthr)viiin(k)=blmthr
```

```
      dbo=0.0
c
c fetal arterial perfusion of organs with contaminant
c fetal bone(2) contaminant content updated for gains from arterial
c blood
      do432it=1,24
      dbn2=0.0
      if(bntbl2(j,k).le.0.0)goto433
      f0=2.0*flomaf
      f2=viiin(k)/volplf
      f3=vbn2(lf,j,k)/(volmaf*bntbl2(j,k))
      dbn2=f0*(f2-f3)/24.
c same for bone(1) compartment
c
  433 dbn1=0.0
      if(bntbl1(j,k).le.0.0)goto434
      f0=2.0*flomaf
      f2=viiin(k)/volplf
      f3=vbn1(lf,j,k)/(volmaf*bntbl1(j,k))
      dbn1=f0*(f2-f3)/24.
      dbo=f0*f2/24.
c
c same for muscle contaminant content
  434 dbms=0.0
      if(mstbl(j,k).le.0.0)goto435
      f0=flomuf
      f2=viiin(k)/volplf
      f3=vms(lf,j,k)/(volmuf*mstbl(j,k))
      dbms=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c same for kidney(2) compartment
  435 dbk2=0.0
      if(kdtbl2(j,k).le.0.0)goto436
      f0=flokif
      f2=viiin(k)/volplf
      f3=vkd2(lf,j,k)/(volkif*kdtbl2(j,k))
      dbk2=f0*(f2-f3)/24.
c
c adipose tissue content updated for gains from arterial blood
  436 dbft=0.0
      if(fttbl(j,k).le.0.0)goto437
      f0=flofaf
      f2=viiin(k)/volplf
      f3=vfat(lf,j,k)/(volfaf*fttbl(j,k))
      dbft=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c residual soft tissue content updated for gain from arterial blood
  437 dbrs=0.0
      if(rstbl(j,k).le.0.0)goto438
      f0=floref
      f2=viiin(k)/volplf
      f3=vr(lf,j,k)/(volref*rstbl(j,k))
```

```
      dbrs=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c kidney(1) contaminant content updated for gains from arterial
c blood and losses to urine
 438 dbk1=0.0
      dbu=0.0
      if(kdtbl1(j,k).le.0.0)goto439
      f0=flokif
      f2=viiin(k)/volplf
      f3=vkd1(lf,j,k)/(volkif*kdtbl1(j,k))
      dbk1=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltur(j,k).le.0.0)goto439
      f2=viiin(k)*bltur(j,k)
      dbu=f0*f2/(24.*floplf)
      if(dbu.le.0.0)dbu=0.0
      if(ainsm.le.0.0)dbu=0.0
c
c liver contaminant content updated for gains from arterial blood
c and gut absorption
 439 dblv=0.0
      if(lvtbl(j,k).le.0.0)goto718
      f0=flolif
      f2=viiin(k)/volplf
      f3=vlv(lf,j,k)/(vollif*lvtbl(j,k))
      dblv=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
 718 if(dbo.le.0.0)goto432
      flxbn2=dbn2*viiin(k)/dbo
      brto=0.99*vbn2(lf,j,k)
      if(flxbn2.lt.0.0.and.-flxbn2.gt.brto)flxbn2=-brto
      vbn2(lf,j,k)=vbn2(lf,j,k)+flxbn2
      viiind=viiin(k)-flxbn2
      if(viiind.le.0.0)viiind=0.0
      if(vbn2(lf,j,k).le.0.0)vbn2(lf,j,k)=0.0
      flxbn1=dbn1*viiin(k)/dbo
      brto=0.99*vbn1(lf,j,k)
      if(flxbn1.lt.0.0.and.-flxbn1.gt.brto)flxbn1=-brto
      if(bntbl2(j,k).le.0.0)viiind=viiin(k)
      vbn1(lf,j,k)=vbn1(lf,j,k)+flxbn1*viiind/viiin(k)
      if(flxbn2.lt.0.0)vbn1(lf,j,k)=vbn1(lf,j,k)-flxbn2
      if(flxbn2.lt.0.0)flxbn2=0.0
      if(vbn1(lf,j,k).le.0.0)vbn1(lf,j,k)=0.0
      flxms=dbms*viiin(k)/dbo
      brto=0.99*vms(lf,j,k)
      if(flxms.lt.0.0.and.-flxms.gt.brto)flxms=-brto
      vms(lf,j,k)=vms(lf,j,k)+flxms
      if(vms(lf,j,k).le.0.0)vms(lf,j,k)=0.0
      flxkd2=dbk2*viiin(k)/dbo
      brto=0.99*vkd2(lf,j,k)
      if(flxkd2.lt.0.0.and.-flxkd2.gt.brto)flxkd2=-brto
      vkd2(lf,j,k)=vkd2(lf,j,k)+flxkd2
```

```
      viiind=viiin(k)-flxkd2
      if(viiind.le.0.0)viiind=0.0
      if(vkd2(lf,j,k).le.0.0)vkd2(lf,j,k)=0.0
      flxkd1=dbk1*viiin(k)/dbo
      brto=0.99*vkd1(lf,j,k)
      if(flxkd1.lt.0.0.and.-flxkd1.gt.brto)flxkd1=-brto
      if(kdtbl2(j,k).le.0.0)viiind=viiin(k)
      vkd1(lf,j,k)=vkd1(lf,j,k)+flxkd1*viiind/viiin(k)
      if(flxkd2.lt.0.0)vkd1(lf,j,k)=vkd1(lf,j,k)-flxkd2
      if(flxkd2.lt.0.0)flxkd2=0.0
      if(vkd1(lf,j,k).le.0.0)vkd1(lf,j,k)=0.0
      flxft=dbft*viiin(k)/dbo
      brto=0.99*vfat(lf,j,k)
      if(flxft.lt.0.0.and.-flxft.gt.brto)flxft=-brto
      vfat(lf,j,k)=vfat(lf,j,k)+flxft
      if(vfat(lf,j,k).le.0.0)vfat(lf,j,k)=0.0
      flxrs=dbrs*viiin(k)/dbo
      brto=0.99*vr(lf,j,k)
      if(flxrs.lt.0.0.and.-flxrs.gt.brto)flxrs=-brto
      vr(lf,j,k)=vr(lf,j,k)+flxrs
      if(vr(lf,j,k).le.0.0)vr(lf,j,k)=0.0
      vurin(lf,j,k)=vurin(lf,j,k)+dbu
      flxlv=dblv*viiin(k)/dbo
      flxlvp=flxlv
      brto=0.99*vlv(lf,j,k)
      if(flxlv.lt.0.0.and.-flxlv.gt.brto)flxlv=-brto
      vlv(lf,j,k)=vlv(lf,j,k)+flxlv
      if(vlv(lf,j,k).le.0.0)vlv(lf,j,k)=0.0
      blmthr= cbf(j,k)*0.2*wt(lf,j)/24.
      v(lf,j,k)=v(lf,j,k)+blmthr-flxbn1-flxbn2-flxkd1-flxkd2-flxft-flxrs
     1-flxlv-flxms-dbu
      if(v(lf,j,k).le.0.0)v(lf,j,k)=0.0
      viiin(k)=v(lf,j,k)
      dbo=0.0
  432 continue
c
c  contaminant corrected for contaminant decay or persistence
      v(lf,j,k)=v(lf,j,k)-disint(k)*v(lf,j,k)
      vbn1(lf,j,k)=vbn1(lf,j,k)-disint(k)*vbn1(lf,j,k)
      vbn2(lf,j,k)=vbn2(lf,j,k)-disint(k)*vbn2(lf,j,k)
      vlv(lf,j,k)=vlv(lf,j,k)-vlv(lf,j,k)*disint(k)
      vms(lf,j,k)=vms(lf,j,k)-disint(k)*vms(lf,j,k)
      vkd1(lf,j,k)=vkd1(lf,j,k)-disint(k)*vkd1(lf,j,k)
      vkd2(lf,j,k)=vkd2(lf,j,k)-disint(k)*vkd2(lf,j,k)
      vr(lf,j,k)=vr(lf,j,k)-disint(k)*vr(lf,j,k)
      vfat(lf,j,k)=vfat(lf,j,k)-disint(k)*vfat(lf,j,k)
c**** contaminant dose assessment for mammalian fetus****
c
c** excludes nutrients
      if(k.gt.1.and.k.lt.7)goto723
      ian=atno(k)
c
      wtgrm=wt(lf,j)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
```

```
c assumes body weight of 1 gram for invertebrates
    if(alsp(j).le.0.0)wtgrm=1.0
c
    wtbn(lf,j)=5.5e-05*wtgrm
    wtkd(lf,j)=2.12e-05*wtgrm**0.85
    wtlv(lf,j)=8.2e-05*wtgrm**0.87
    wtms(lf,j)=4.5e-04*wtgrm
    wtlu(lf,j)=1.24e-05*wtgrm**0.99
c
    if(ian.ge.101.)goto723
c fetal cummulative kidney dose, rems/gfwt
    rkd(lf,in,k)=rkd(lf,in,k)+5.12e-05*evs(j,k)*(vkd1(lf,j,k)
    1+vkd2(lf,j,k))/(wtkd(lf,j)*1000.)
c fetal cumulative bone dose, rems/gfwt
    rbn(lf,in,k)=rbn(lf,in,k)+5.12e-05*evb(j,k)*(vbn1(lf,j,k)
    1+vbn2(lf,j,k))/(wtbn(lf,j)*1000)
c fetal contaminant burden in bone(1,2)
    vndof=(vbn1(lf,j,k)+vbn2(lf,j,k))
c fetal weight of endosteal bone tissue, gfwt
    endwtf=0.025*wtbn(lf,j)*1000.
c fetal cummulative endosteal bone dose, rems/gfwt
    rndo(lf,in,k)=rndo(lf,in,k)+5.12e-05*eve(j,k)*vndof/endwtf
c fetal cummulative liver dose, rems/gfwt
    rlv(lf,j,k)=rlv(lf,j,k)+5.12e-05*evs(j,k)*vlv(lf,j,k)/
    1(wtlv(lf,j)*1000.)
c fetal cummulative muscle dose, rems/gfwt
    rms(lf,j,k)=rms(lf,j,k)+5.12e-05*evs(j,k)*vms(lf,j,k)/
    1(wtms(lf,j)*1000.)
c
c**** excludes nutrients
  702 if(k.gt.1.and.k.lt.7)goto723
      ian=atno(k)
      if(ian.ge.101)goto723
      if(alsp(j).le.0.0)goto706
c
c excludes 1 year olds if newborn not available
      if(l.ne.1)goto706
      if(aiiir(j,1).le.aiii)goto706
      if(gstn(j).lt.2.0) goto740
      if(aiiir(j,2).gt.aiii)goto740
c
c cummulative kidney dose, rems/gfwt
  706 l=ii
      j=in
      rkd(l,j,k)=rkd(l,j,k)+5.12e-05*evs(j,k)*(vkd1(l,j,k)
      1+vkd2(l,j,k))/(wtkd(l,j)*1000.)
c cumulative bone dose, rems/gfwt
      rbn(l,in,k)=rbn(l,in,k)+5.12e-05*evb(j,k)*(vbn1(l,j,k)
      1+vbn2(l,j,k))/(wtbn(l,j)*1000.)
c contaminant burden in bone(1,2)
      vndo=(vbn1(l,j,k)+vbn2(l,j,k))
c weight of endosteal bone tissue, gfwt
      endwt=0.025*wtbn(l,j)*1000.
c cummulative endosteal bone dose, rems/gfwt
```

```
      rndo(l,in,k)=rndo(l,in,k)+5.12e-05*eve(j,k)*vndo/endwt
c cummulative liver dose, rems/gfwt
      rlv(l,in,k)=rlv(l,in,k)+5.12e-05*evs(j,k)*vlv(l,j,k)/
     1(wtlv(l,j)*1000.)
c cummulative lung dose, rems/gfwt
      rlu(l,in,k)=rlu(l,in,k)+5.12e-05*evs(j,k)*(q1(l,j,k)+q2(l,j,k)+
     1q3(l,j,k))/(wtlu(l,j)*1000.)
c cummulative muscle dose, rems/gfwt
      rms(l,in,k)=rms(l,in,k)+5.12e-05*evs(j,k)*vms(l,j,k)/
     1(wtms(l,j)*1000.)
c
c excludes immature animals and invertebrates from milk production
  723 if(alsp(l).le.0.0)goto740
      agemi=agem(l,j)
      fracs(j)=1.0
      if(l.ne.1)goto724
      if(iage.ge.6)goto724
      off2t=0.0
      off2i=0.0
      do725ig1=1,3
      off2t=off2t+offsp(j,ig1,2)
      if(affsp(j,ig1).ge..15*alsp(j).and.affsp(j,ig1).le..7*alsp(j))
     1off2i=off2i+offsp(j,ig1,2)
  725 continue
      fracs(j)=0.0
      if(off2t.gt.0.0) fracs(j)=off2i/off2t
      if(fracs(j).gt.0.0.and.agemi.lt.0.16*alsp(j)) agemi=0.16*alsp(j)
  724 if(agemi.lt.0.15*alsp(j).or.agemi.gt.0.7*alsp(j))goto740
c mean daily activity per liter in milk, producer age groups
      if(afmt(j).le.0.0)goto740
      w(18,j,k)=w(18,j,k)+vmk(l,j,k)*sfd(ifa,j)*fracs(j)
     1/(afmt(j)*1.1915e-03*(wt(l,j)*1000.)**0.7)
  740 continue
c ** end contaminant loop ***********************************************
c
c excludes insects
      if(alsp(in).le.0.0)goto790
      do741ms=1,2
      mm=ms
      if(ydos(in).gt.0.0)call effects(in,lx,ii,mm,rdidx)
  741 continue
c
c assures all age categories have been processed
      if(ii.lt.iage)goto790
c excludes newborns from population dynamics
      if(ii.gt.iage)goto831
c initializes population parameters daily
      zoo(in)=0.0
      pop(1,in)=0.0
      pop(2,in)=0.0
      if(aiii.gt.1.0)goto742
c initializes daily mortalities for all age groups combined
      summrt(in,1)=0.0
      summrt(in,2)=0.0
```

```
742 ikn(1,in,1)=0
    ikn(2,in,1)=0
    ikn(1,in,2)=0
    ikn(2,in,2)=0
    ikn(1,in,3)=0
    ikn(2,in,3)=0
    ikn(1,in,4)=0
    ikn(2,in,4)=0
    ikn(1,in,5)=0
    ikn(2,in,5)=0
    ikn(1,in,6)=0
    ikn(2,in,6)=0
    ikn(1,in,7)=0
    ikn(2,in,7)=0
    ikn(1,in,8)=0
    ikn(2,in,8)=0
c ** age loop ********************************************************
    do 770 k=1,iage
c estimates population losses on a daily basis; does not allow
c mortality losses to exceed predation losses
    if(k.gt.1)goto772
    tqx1=tqx1+tqs(k,in)+tqp(k,in)
    tqx2=tqx2+tqs(k,in)+tqp(k,in)
    tqx3=tqx3+tqs(k,in)+tqp(k,in)
c
    summrt(in,1)=summrt(in,1)+tqx1*offsp(in,1,1)+tqx2*offsp(in,2,1)+
   1tqx3*offsp(in,3,1)
    summrt(in,2)=summrt(in,2)+tqx1*offsp(in,1,2)+tqx2*offsp(in,2,2)+
   1tqx3*offsp(in,3,2)
c
    offsp(in,1,1)=offsp(in,1,1)-tqx1*offsp(in,1,1)
    offsp(in,2,1)=offsp(in,2,1)-tqx2*offsp(in,2,1)
    offsp(in,3,1)=offsp(in,3,1)-tqx3*offsp(in,3,1)
    offsp(in,1,2)=offsp(in,1,2)-tqx1*offsp(in,1,2)
    offsp(in,2,2)=offsp(in,2,2)-tqx2*offsp(in,2,2)
    offsp(in,3,2)=offsp(in,3,2)-tqx3*offsp(in,3,2)
c
    tpop(k,1,in)=offsp(in,1,1)+offsp(in,2,1)+offsp(in,3,1)
    tpop(k,2,in)=offsp(in,1,2)+offsp(in,2,2)+offsp(in,3,2)
    pop(1,in)=pop(1,in)+tpop(k,1,in)
    pop(2,in)=pop(2,in)+tpop(k,2,in)
    zoo(in)=zoo(in)+tpop(k,1,in)+tpop(k,2,in)
    goto773
  772 tqx(k,1,in)=tqx(k,1,in)+tqs(k,in)+tqp(k,in)
    tqx(k,2,in)=tqx(k,2,in)+tqs(k,in)+tqp(k,in)
c
    summrt(in,1)=summrt(in,1)+tqx(k,1,in)*tpop(k,1,in)
    tpop(k,1,in)=tpop(k,1,in)-tqx(k,1,in)*tpop(k,1,in)
    summrt(in,2)=summrt(in,2)+tqx(k,2,in)*tpop(k,2,in)
    tpop(k,2,in)=tpop(k,2,in)-tqx(k,2,in)*tpop(k,2,in)
c
    pop(1,in)=pop(1,in)+tpop(k,1,in)
    pop(2,in)=pop(2,in)+tpop(k,2,in)
    zoo(in)=zoo(in)+tpop(k,1,in)+tpop(k,2,in)
```

```
c
c ** births age loop ************************************************
c
 773 if(k.ne.1)goto770
     iagem=aiiir(in,1)
     icagm=cagm(in,1)
     if(iagem.eq.icagm)irp=1
     if(iagem.eq.icagm)goto805
c
     if(gstn(in).lt.2.0)goto770
     icagm=cagm(in,2)
     iagem=aiiir(in,2)
     if(iagem.eq.icagm)irp=2
     if(iagem.eq.icagm)goto805
     goto770
c
c initializes daily births for each reproductive cycle
 805 births(1,in)=0.0
     births(2,in)=0.0
     iaged=iage*365.
     do621kk=1,iaged
     akd=kk
     if (akd.lt..15*alsp(in).or.akd.gt..70*alsp(in)) go to 621
c sets initial male-female birth rates for youngest females
     do624jj=1,2
     brate=brrt(1,in,jj)*sfd(1,in)*popp(2,in)*frpr(irp,in)
     knp=1
c ** birth rate loop ***************************************************
     do 623 kn=1,7
c natality rate changed according female age groupings specified in (BRTY)
     if(akd.le.brty(kn,in))goto623
     knp=kn+1
     brate=brrt(knp,in,jj)*sfd(knp,in)*popp(2,in)*frpr(irp,in)
 623 continue
c * end birth rate loop ************************************************
c
     if(ikn(jj,in,knp).gt.0)goto624
c new-borns of males-females estimated on a daily basis
     births(jj,in)=births(jj,in)+brate
     brts(in,jj,irp)=births(jj,in)
     ikn(jj,in,knp)=ikn(jj,in,knp)+1
 624 continue
 621 continue
     affsp(in,irp+1)=1.0
     if(jage(in).ne.0)goto625
     if(irp.eq.1) offsp(in,1,1)=births(1,in)
     if(irp.eq.1) offsp(in,1,2)=births(2,in)
     if(irp.eq.2) offsp(in,1,1)=offsp(in,1,1)+births(1,in)
     if(irp.eq.2) offsp(in,1,2)=offsp(in,1,2)+births(2,in)
     aiif(in,1)=0.0
     aiif(in,2)=0.0
     goto806
 625 offsp(in,irp,1)=offsp(in,irp,1)+offsp(in,irp+1,1)
     offsp(in,irp,2)=offsp(in,irp,2)+offsp(in,irp+1,2)
```

```
      offsp(in,irp+1,1)=births(1,in)
      offsp(in,irp+1,2)=births(2,in)
      aiif(in,1)=0.0
      aiif(in,2)=0.0
c
c allows estimation of newborn ratio to old newborns
c
c newborn ratio to other 1 year-olds estimated
c
  806 tppp1=tpop(1,1,in)
      tppp2=tpop(1,2,in)
      tppps(in)=tppp1+tppp2
      tppc1=births(1,in)
      tppc2=births(2,in)
      tppcs(in)=tppc1+tppc2
      tpppt(in)=tppps(in)+tppcs(in)
  770 continue
c
      if(aiii.lt.365.)goto777
      do771 k=1,iage
      j=k-1
c
c the following statements move animal parameters into next age group
c at the end of each Julian year
c
      if(k.eq.iage)goto771
      c12f=1.0
      if(iage-k.ne.1)goto776
      c12f=rat12(in)
      offsm=offsp(in,1,1)+offsp(in,2,1)+offsp(in,3,1)
      off1=offsp(in,1,1)/offsm
      off2=offsp(in,2,1)/offsm
      off3=offsp(in,3,1)/offsm
c
      if(jage(in).ne.0)offsp(in,1,1)=tpop(1,1,in)*(1.0-c12f)*off1
      if(jage(in).ne.0)offsp(in,1,2)=tpop(1,2,in)*(1.0-c12f)*off1
      if(jage(in).ne.0)offsp(in,2,1)=tpop(1,1,in)*(1.0-c12f)*off2
      if(jage(in).ne.0)offsp(in,2,2)=tpop(1,2,in)*(1.0-c12f)*off2
      if(jage(in).ne.0)offsp(in,3,1)=tpop(1,1,in)*(1.0-c12f)*off3
      if(jage(in).ne.0)offsp(in,3,2)=tpop(1,2,in)*(1.0-c12f)*off3
c
c updates population structure if mortality parameters are specified
c
  776 if(jage(in).ne.0) tpop(iage-j,1,in)=tpop(iage-k,1,in)*c12f
      if(jage(in).ne.0) tpop(iage-j,2,in)=tpop(iage-k,2,in)*c12f
c updates other population parameters
c
c updates total weight, kgfwt
      wt(iage-j,in)=wt(iage-k,in)
c
c ** contaminant loop ************************************************
      do 760 l=1,inz
c updates contaminant compartments
c blood
```

```
      v(iage-j,in,l)=v(iage-k,in,l)
c residual soft tissue
      vr(iage-j,in,l)=vr(iage-k,in,l)
c bone(1 & 2)
      vbn1(iage-j,in,l)=vbn1(iage-k,in,l)
      vbn2(iage-j,in,l)=vbn2(iage-k,in,l)
c liver
      vlv(iage-j,in,l)=vlv(iage-k,in,l)
c kidneys(1 & 2)
      vkd1(iage-j,in,l)=vkd1(iage-k,in,l)
      vkd2(iage-j,in,l)=vkd2(iage-k,in,l)
c muscle
      vms(iage-j,in,l)=vms(iage-k,in,l)
c fat compartment
      vfat(iage-j,in,l)=vfat(iage-k,in,l)
c lung compartment(1,2, & 3)
      q1(iage-j,in,l)=q1(iage-k,in,l)
      q2(iage-j,in,l)=q2(iage-k,in,l)
      q3(iage-j,in,l)=q3(iage-k,in,l)
c kidney(1,2), bone(1,2), endosteal tissue, liver, lung, and muscle
c cummulative doses moved into succeeding age group
      rkd(iage-j,in,l)=rkd(iage-k,in,l)
      rbn(iage-j,in,l)=rbn(iage-k,in,l)
      rndo(iage-j,in,l)=rndo(iage-k,in,l)
      rlv(iage-j,in,l)=rlv(iage-k,in,l)
      rlu(iage-j,in,l)=rlu(iage-k,in,l)
      rms(iage-j,in,l)=rms(iage-k,in,l)
c
  760 continue
c ** end contaminant loop ******************************************
  771 continue
c ** end age loop ****************************************************
  777 if (aiii.lt.365.) goto790
c
      aoffsp=affsp(in,2)
      if(gstn(in).gt.1.0)aoffsp=affsp(in,3)
      prwt=wt(1,in)
      wt(1,in)=rr(in)*(1.0-exp(-9.0*aoffsp/alsp(in)))
      wtrat=wt(1,in)/prwt
      do791l=1,inz
c blood
      v(1,in,l)=v(1,in,l)*wtrat
c residual soft tissue
      vr(1,in,l)=vr(1,in,l)*wtrat
c bone(1 & 2)
      vbn1(1,in,l)=vbn1(1,in,l)*wtrat
      vbn2(1,in,l)=vbn2(1,in,l)*wtrat
c liver
      vlv(1,in,l)=vlv(1,in,l)*wtrat
c kidneys(1 & 2)
      vkd1(1,in,l)=vkd1(1,in,l)*wtrat
      vkd2(1,in,l)=vkd2(1,in,l)*wtrat
c muscle
      vms(1,in,l)=vms(1,in,l)*wtrat
```

```
c fat
      vfat(1,in,l)=vfat(1,in,l)*wtrat
c lung compartment(1,2, & 3)
      q1(1,in,l)=q1(1,in,l)*wtrat
      q2(1,in,l)=q2(1,in,l)*wtrat
      q3(1,in,l)=q3(1,in,l)*wtrat
  791 continue
c
c ** contaminant loop ***************************************************
  790 if(ii.ne.iage)goto831
      do 780 k=1,inz
c mean daily parameter values initialized
      w(1,in,k)=0.0
      w(2,in,k)=0.0
      w(3,in,k)=0.0
      w(4,in,k)=0.0
      w(5,in,k)=0.0
      w(6,in,k)=0.0
      w(7,in,k)=0.0
      w(8,in,k)=0.0
      w(9,in,k)=0.0
      w(10,in,k)=0.0
      w(11,in,k)=0.0
      w(12,in,k)=0.0
      w(13,in,k)=0.0
      w(14,in,k)=0.0
      w(15,in,k)=0.0
      w(16,in,k)=0.0
      w(17,in,k)=0.0
      w(19,in,k)=0.0
  780 continue
c ** end contaminant loop ***********************************************
      wms(in,2)=0.0
      wtrsa=0.0
      wtbla=0.0
      wtfff=0.0
      wtkdd=0.0
      wtmuu=0.0
      wtlvv=0.0
      wtbnn=0.0
      wtluu=0.0
      wtbod=0.0
      xinf=0.0
c
c ** age loop ***********************************************************
      do 811 k=1,iage
c estimates fraction of milk producing females in female population
      do 813 l=1,inz
      if(vmk(k,in,l).gt.0.0)xinf=xinf+a(k,in,2)
  813 continue
c
c mean values of mass parameters estimated
c
c mean animal weight, kgfwt
```

```
      xins=(a(k,in,1)*pop(1,in)+a(k,in,2)*pop(2,in))/zoo(in)
      wms(in,2)=wms(in,2)+wt(k,in)*xins
      wtrsa=wtrsa+wtrs(k,in)*xins
      wtbla=wtbla+wtbl(k,in)*xins
      wtfff=wtfff+fatb(k,in)*xins
      wtkdd=wtkdd+wtkd(k,in)*xins
      wtmuu=wtmuu+wtms(k,in)*xins
      wtlvv=wtlvv+wtlv(k,in)*xins
      wtbnn=wtbnn+wtbn(k,in)*xins
      wtluu=wtluu+wtlu(k,in)*xins
      wtbod=wtbod+wt(k,in)*xins
  811 continue
c
c ** age loop ********************************************************
      do 810 k=1,iage
      if(wt(k,in).le.0.0)goto810
c ** contaminant loop ************************************************
      do 800 l=1,inz
c mean weighted and age specific concentrations in organs, and sources
c of contamination
      xins=(a(k,in,1)*pop(1,in)+a(k,in,2)*pop(2,in))/zoo(in)
      xinf2=0.0
      if(vmk(k,in,l).gt.0.0)xinf2=a(k,in,2)
c milk clearance, per kgfwt body
      if(xinf.gt.0.0) w(1,in,l)=w(1,in,l)+vmk(k,in,l)*xinf2/(xinf*wtbod)
      cnmlk(k,in,l)=vmk(k,in,l)/wt(k,in)
c urine clearance, per kgfwt body
      w(2,in,l)=w(2,in,l)+vurin(k,in,l)*xins/wtbod
      cnurnc(k,in,l)=vurin(k,in,l)/wt(k,in)
c lung clearance contribution to ingestion, per kgfwt body
      w(3,in,l)=w(3,in,l)+vlug(k,in,l)*xins/wtbod
      cnlug(k,in,l)=vlug(k,in,l)/wt(k,in)
c soil uptake contribution to ingestion, per kgfwt body
      w(4,in,l)=w(4,in,l)+vsol(k,in,l)*xins/wtbod
      cnsol(k,in,l)=vsol(k,in,l)/wt(k,in)
c plant uptake contribution to ingestion, per kgfwt body
      w(5,in,l)=w(5,in,l)+vpln(k,in,l)*xins/wtbod
      cnpln(k,in,l)=vpln(k,in,l)/wt(k,in)
c total body concentration
      bbr=vkd1(k,in,l)+vkd2(k,in,l)+vbn1(k,in,l)+vbn2(k,in,l)+vr(k,in,l)
     1+vms(k,in,l)+vlv(k,in,l)+v(k,in,l)+vfat(k,in,l)
      w(6,in,l)=w(6,in,l)+bbr*xins/wtbod
      cnbod(k,in,l)=bbr/wt(k,in)
c contaminated pelt contribution to ingestion, per kgfwt body
      w(7,in,l)=w(7,in,l)+c31plt(k,in,l)*xins/wtbod
      cnpel(k,in,l)=c31plt(k,in,l)/wt(k,in)
c total contaminant ingestion, per kgfwt body
      w(8,in,l)=w(8,in,l)+vgut(k,in,l)*xins/wtbod
      cngut(k,in,l)=vgut(k,in,l)/wt(k,in)
c mean kidney concnetration
      w(9,in,l)=w(9,in,l)+(vkd1(k,in,l)+vkd2(k,in,l))*xins/wtkdd
      cnkid(k,in,l)=(vkd1(k,in,l)+vkd2(k,in,l))/wtkd(k,in)
c mean bone contaminant concentration
      w(10,in,l)=w(10,in,l)+(vbn1(k,in,l)+vbn2(k,in,l))*xins/wtbnn
```

```
      cnbon(k,in,l)=(vbn1(k,in,l)+vbn2(k,in,l))/wtbn(k,in)
c**** mean residual soft tissue contaminant concentration
      w(11,in,l)=w(11,in,l)+vr(k,in,l)*xins/wtrsa
      cnres(k,in,l)=vr(k,in,l)/wtrs(k,in)
c mean muscle contaminant concentration
      w(13,in,l)=w(13,in,l)+vms(k,in,l)*xins/wtmuu
      cnmus(k,in,l)=vms(k,in,l)/wtms(k,in)
c mean liver contaminant concentration
      w(14,in,l)=w(14,in,l)+vlv(k,in,l)*xins/wtlvv
      cnliv(k,in,l)=vlv(k,in,l)/wtlv(k,in)
c mean blood contaminant content, age cohort(k)
      w(15,in,l)=w(15,in,l)+v(k,in,l)*xins/wtbla
      cnblo(k,in,l)=v(k,in,l)/wtbl(k,in)
c lung contaminant content
      q123=q1(k,in,l)+q2(k,in,l)+q3(k,in,l)
c mean lung contaminant content, all ages
      w(16,in,l)=w(16,in,l)+q123*xins/wtluu
      cnlun(k,in,l)=q123/wtlu(k,in)
c mean fat contaminant content
      w(17,in,l)=w(17,in,l)+vfat(k,in,l)*xins/wtfff
      cnfat(k,in,l)=vfat(k,in,l)/fatb(k,in)
c mean urine concentration, per liter
      flourn=0.024*.0064*(wt(k,in)*1000.)**0.82
      w(19,in,l)=w(19,in,l)+(vurin(k,in,l)/flourn)*xins
      cnurn(k,in,l)=vurin(k,in,l)/flourn
c mean feces concentration, per liter
      vollum=5.78e-05*(wt(k,in)*1000.)**0.934
      w(20,in,l)=w(20,in,l)+(1.0-gtbl(in,l))*vgut(k,in,l)/vollum
      cnfec(k,in,l)=(1.0-gtbl(in,l))*vgut(k,in,l)/vollum
  800 continue
c ** end contaminant ************************************************
  810 continue
c ** end age loop ****************************************************
      if (wms(in,2).le.0.0) goto831
c total herd muscle mass
      wms(in,1)=0.45*wms(in,2)*zoo(in)
c
  831 aiin=in
c
c ** establishes contaminant concentrations in newborns ****
c ** excludes insects
      if (alsp(in).le.0.0)goto860
c ** excludes other age groups
      if(ii.lt.iage+1)goto860
      iagem=aiiir(in,1)
      icagm1=cagm(in,1)
      if(iagem.ne.icagm1)goto851
      goto852
  851 if(gstn(in).lt.2.0)goto860
      icagm2=cagm(in,2)
      iagem=aiiir(in,2)
      if(iagem.ne.icagm2)goto860
c
c ** contaminant loop ***********************************************
```

```
c
 852 clf(in)=tppcs(in)/tpppt(in)
     cal(in)=tppps(in)/tpppt(in)
     lf=iage+1
c newborn and fetal contaminant parameters initialized
c
     wt(1,in)=clf(in)*wt(lf,in)+cal(in)*wt(1,in)
     wt(lf,in)=0.0
     do 850 k=1,inz
     v(1,in,k)=clf(in)*v(lf,in,k)+cal(in)*v(1,in,k)
     v(lf,in,k)=0.0
     vbn1(1,in,k)=clf(in)*vbn1(lf,in,k)+cal(in)*vbn1(1,in,k)
     vbn1(lf,in,k)=0.0
     vbn2(1,in,k)=clf(in)*vbn2(lf,in,k)+cal(in)*vbn2(1,in,k)
     vbn2(lf,in,k)=0.0
     vlv(1,in,k)=clf(in)*vlv(lf,in,k)+cal(in)*vlv(1,in,k)
     vlv(lf,in,k)=0.0
     vkd1(1,in,k)=clf(in)*vkd1(lf,in,k)+cal(in)*vkd1(1,in,k)
     vkd1(lf,in,k)=0.0
     vkd2(1,in,k)=clf(in)*vkd2(lf,in,k)+cal(in)*vkd2(1,in,k)
     vkd2(lf,in,k)=0.0
     vms(1,in,k)=clf(in)*vms(lf,in,k)+cal(in)*vms(1,in,k)
     vms(lf,in,k)=0.0
     vr(1,in,k)=clf(in)*vr(lf,in,k)+cal(in)*vr(1,in,k)
     vr(lf,in,k)=0.0
     vfat(1,in,k)=clf(in)*vfat(lf,in,k)+cal(in)*vfat(1,in,k)
     vfat(lf,in,k)=0.0
c
     q1(1,in,k)=clf(in)*q1(lf,in,k)+cal(in)*q1(1,in,k)
     q1(lf,in,k)=0.0
     q2(1,in,k)=clf(in)*q2(lf,in,k)+cal(in)*q2(1,in,k)
     q2(lf,in,k)=0.0
     q3(1,in,k)=clf(in)*q3(lf,in,k)+cal(in)*q3(1,in,k)
     q3(lf,in,k)=0.0
c
     rbn(1,in,k)=clf(in)*rbn(lf,in,k)+cal(in)*rbn(1,in,k)
     rbn(lf,in,k)=0.0
     rndo(1,in,k)=clf(in)*rndo(lf,in,k)+cal(in)*rndo(1,in,k)
     rndo(lf,in,k)=0.0
     rlv(1,in,k)=clf(in)*rlv(lf,in,k)+cal(in)*rlv(1,in,k)
     rlv(lf,in,k)=0.0
     rlu(1,in,k)=clf(in)*rlu(lf,in,k)+cal(in)*rlu(1,in,k)
     rlu(lf,in,k)=0.0
     rms(1,in,k)=clf(in)*rms(lf,in,k)+cal(in)*rms(1,in,k)
     rms(lf,in,k)=0.0
     rkd(1,in,k)=clf(in)*rkd(lf,in,k)+cal(in)*rkd(1,in,k)
     rkd(lf,in,k)=0.0

c
c
 850 continue
c ** end contaminant loop ***********************************************
c animal index
 860 ain=in
```

```
return
end
```

14.4. AQUAT

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine aquat (in,lx,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
      common/pb/ akc(5,n),
     1 celct(kw,n),
     2 cho(5,kw,n), dendif(kw,n), dense(kw,n), depthi(n), dit
     3 (n), dv(5,kw,n), dvz(2,kw,n), eat(5,kw,2,n), epld(n),
     4 extc(n), flxdn(kw,n), flxup(kw,n),
     5 qnitro(5,kw,n), qphosp(5,kw,n), qsilcn(5,kw,n),
     6 sg(5,kw,n), snfrc(n), tf(2,kw,n),
     7 zmin(2,n),
     9 znitro(2,kw,n), zosum(kw,n), zphosp(2,kw,n), zsm(2,kw,n)
c
c   aquat subroutine simulates plankton growth and contaminant
c   uptake in a freshwater lake ( uptake not yet developed)
c
c excludes all but first access to subroutine
      if (rdidx.gt.0.) go to 70
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-333333.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','blue')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE AQUAT                    ')
      call wnoust('****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
```

```
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 333333.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
 5002 print *,'ENTER zooplankton feeding gate(0.,1.);'
      print *,'determines zooplankton dependence on  '
      print *,'simulated phytoplankton'
      if(nu.eq.0)read(*,*,err=5002) skip
      if(nu.eq.1)read(7,*) skip
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*) skip
      if(iu.eq.1)print *, skip
c
c regional input
c
c fraction of incident light not backscattered from water layer
 5003 print *,'ENTER frac. of incid. light not backscattered'
      print *,'from water layers for',ipx,'regions'
      if(nu.eq.0)read(*,*,err=5003)(snfrc(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*)(snfrc(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)(snfrc(lz),lz=1,ipx)
      if(iu.eq.1)print *,(snfrc(lz),lz=1,ipx)
 5004 print *,'ENTER depth,meters, of individual water layers'
```

```
      print *,'for',ipx,'regions'
c depth of water of water layers, meters
      if(nu.eq.0)read(*,*,err=5004)(depthi(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*)(depthi(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)(depthi(lz),lz=1,ipx)
      if(iu.eq.1)print *,(depthi(lz),lz=1,ipx)
c light extinction coefficient of water excluding excluding phytoplankton
 5005 print *,'ENTER light extinc. coefficient of water excluding'
      print *,'phytoplankton for',ipx,'regions'
      if(nu.eq.0)read(*,*,err=5005)(extc(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*)(extc(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*)(extc(lz),lz=1,ipx)
      if(iu.eq.1)print *,(extc(lz),lz=1,ipx)
c number of water layers simulated
 5006 print *,'ENTER number of water layers simulated for',ipx
      print *,'regions'
      if(nu.eq.0)read(*,*,err=5006)(dit(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*)(dit(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*)(dit(lz),lz=1,ipx)
      if(iu.eq.1)print *,(dit(lz),lz=1,ipx)
c depth to epilimnion, meters
 5007 print *,'ENTER maximum depth,meters, to epilimnion for',ipx
      print *,'regions'
      if(nu.eq.0)read(*,*,err=5007)(epld(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*)(epld(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*)(epld(lz),lz=1,ipx)
      if(iu.eq.1)print *,(epld(lz),lz=1,ipx)
c
c ** region loop ****************************************************
      do 10 lz=1,ipx
c number of water layers simulated for each region
      kdi=dit(lz)
c initial water  temperature by layer, deg. C.
 5008 print *,'ENTER initial water temp.,deg C.,for',kdi
      print *,'layers for region',lz
      if(nu.eq.0)read(*,*,err=5008)(watmp(k,lx),k=1,kdi)
      if(nu.eq.1)read(7,*)(watmp(k,lx),k=1,kdi)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*)(watmp(k,lx),k=1,kdi)
      if(iu.eq.1)print *,(watmp(k,lx),k=1,kdi)
```

```
c initial phosphate phosphorus concentration, ug/liter, by layer
 5009 print *,'ENTER phosphate phosphorus concentration, ug/l'
      print *,'for',kdi,'layers in region',lz
      if(nu.eq.0)read(*,*,err=5009)(phosp(k,lz),k=1,kdi)
      if(nu.eq.1)read(7,*)(phosp(k,lz),k=1,kdi)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,*)(phosp(k,lz),k=1,kdi)
      if(iu.eq.1)print *,(phosp(k,lz),k=1,kdi)
c initial nitrate nitrogen concentration, ug/liter, by layer
 5010 print *,'ENTER nitrate nitrogen concentration,ug/l'
      print *,'for',kdi,'layers in region',lz
      if(nu.eq.0)read(*,*,err=5010)(anitr(k,lz),k=1,kdi)
      if(nu.eq.1)read(7,*)(anitr(k,lz),k=1,kdi)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*)(anitr(k,lz),k=1,kdi)
      if(iu.eq.1)print *,(anitr(k,lz),k=1,kdi)
c initial silicate silicon concentration, ug/liter, by layer
 5011 print *,'ENTER silicate silicon concentration,ug/l'
      print *,'for',kdi,'layers in region',lz
      if(nu.eq.0)read(*,*,err=5011)(silcn(k,lz),k=1,kdi)
      if(nu.eq.1)read(7,*)(silcn(k,lz),k=1,kdi)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*)(silcn(k,lz),k=1,kdi)
      if(iu.eq.1)print *,(silcn(k,lz),k=1,kdi)
   10 continue
 5013 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('*********************                    ')
      call wnoust('END OF INPUT IN AQUAT                      ')
      call wnoust('*********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for AQUAT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now        ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5013) ans
      if(ans)11,5500,11
   11 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-333333.)5501,5503,5501
 5503 backspace 6
      nu=nuu
```

```
      call clear_screen@
      goto5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c ** end region loop ***************************************************
c
c ** region loop ********************************************************
      do 30 lz=1,ipx
c
c ** plankton loop  (5 phytoplankton types; 2 zooplankton types)
      do 20 j=1,5
c specific phytoplankton cell light extinction coefficient
      exip(j,lz)=exip(j,1)*1.0e-07
c maximum phytoplankton cell biomass, ugrams
      cmax(j,lz)=cmax(j,1)*(12.02/.53)*1.0e-06
c minimum phytoplankton cell biomass, ugrams
      akc(j,lz)=0.1*cmax(j,lz)
c PO4-phosphorus, NO3-nitrogen, and Si02-silicon Michaelis-Menton half-
c saturation concentration constants, ug/liter
      aksph(j,lz)=1.0e-06*aksph(j,1)*30.97
      aksnt(j,lz)=1.0e-06*aksnt(j,1)*14.00
      akssi(j,lz)=1.0e-06*akssi(j,1)*28.08
c PO4-phosphorus, NO3-nitrogen, and SiO2-silicon  incorporation rate by
c phtoplankton, ug/hour
      umph(j,lz)=umph(j,1)*30.97*1.0e-09
      umnt(j,lz)=umnt(j,1)*14.*1.0e-09
      umsi(j,lz)=umsi(j,1)*28.08*1.0e-09
c PO4-phosphorus, NO3-nitrogen, and SiO2-silicon minimum content, ugrams, for
c each phytoplankton cell type
      akqph(j,lz)=akqph(j,1)*30.97*1.0e-09
      akqnt(j,lz)=akqnt(j,1)*14.*1.0e-09
      akqsi(j,lz)=akqsi(j,1)*28.08*1.0e-09
c PO4-phosphorus, NO3-nitrogen, and SiO2-silicon maximum content, ugrams, for
c each phytoplankton cell type
      qmph(j,lz)=qmph(j,1)*30.97*1.0e-08
      qmnt(j,lz)=qmnt(j,1)*14.*1.0e-08
      qmsi(j,lz)=qmsi(j,1)*28.08*1.0e-08
c phyoplankton respiration rate ug-CH2O/day
      resp(j,lz)=resp(j,1)*(12.01/.53)*1.0e-07
c optimum daily light insolation, langleys/day, for photosynthesis for each
c phtoplankton cell type
      aiopt(j,lz)=aiopt(j,1)*60.
c
c zooplankton initialization
c
c two classes of zooplankton simulated
      if (j.gt.2) go to 20
c minimun zooplankton NO3-nitrogen, and PO4-phosphorus content, ugrams
      zkqnt(j,lz)=zkqnt(j,1)*14.*1.0e-03
      zkqph(j,lz)=zkqph(j,1)*30.97*1.0e-03
c maximum zooplankton PO4-phosphate and NO3-nitrate content, ugrams
      zmph(j,lz)=zmph(j,1)*30.97*1.0e-02
```

```
      zmnt(j,lz)=zmnt(j,1)*14.*1.0e-02
c maximum and minimum zooplankton biomass, ug-CH2O/day
      zmax(j,lz)=zmax(j,1)*(12.02/0.53)
      zmin(j,lz)=0.1*zmax(j,1)
   20 continue
c ** end plankton loop ***********************************************
   30 continue
c ** end region loop *************************************************
c 1st, 2nd, and 3rd partial regression coefficients for determining the
c density of water, grams/cm3,  from its temperature in deg. C.
      aden=118932.
      bden=1366.75
      cden=1.90
c
c ** region loop ****************************************************
      do 60 lz=1,ipx
c
c ** water layer loop ************************************************
      do 50 k=1,kdi
c PO4-phosphorus, NO3-nitrate, and SiO2-silicon concentrations in water by
c layer, ug/liter
      phosp(k,lz)=phosp(k,lz)*30.97/94.97
      anitr(k,lz)=anitr(k,lz)*14./62.
      silcn(k,lz)=silcn(k,lz)*28.08/60.
c
c ** phytpplankton loop **********************************************
      do 40 j=1,5
c cell biomass, PO4-phosphate, NO3-nitrogen, and SiO2-silicon contents
c initialized to minimum values
      cho(j,k,lz)=akc(j,lz)
      qphosp(j,k,lz)=akqph(j,lz)
      qnitro(j,k,lz)=akqnt(j,lz)
      qsilcn(j,k,lz)=akqsi(j,lz)
c number of cells/250 liters( 1 m2 area x 0.25 m depth) or per volume
c obtained by the product: area(1 m2) X depth (meters)*1000; value
c initialized to 1.0
      cells(j,k,lz)=1.0
c death increment function for phytoplankton  initialized
      sg(j,k,lz)=0.0
c excludes all but 2 zooplankton types
      if (j.gt.2) go to 40
c initializes number of zooplankton to 1.0/250 liters...as above for CELLS
      zoopc(j,k,lz)=1.0
c biomass, PO4-phosphorus, and NO3-nitrogen content, ugrams initialized
      zobio(j,k,lz)=zmin(j,lz)
      zphosp(j,k,lz)=zkqph(j,lz)
      znitro(j,k,lz)=zkqnt(j,lz)
c division rate, per hour, for zooplankton initialized
      dvz(j,k,lz)=1.0
   40 continue
c ** end phytoplankton loop ******************************************
   50 continue
c ** end water layer loop ********************************************
   60 continue
```

```
c ** end region loop ******************************************************
      return
   70 continue
c angle of incidence (radians) based on zenith angle (RADANG, deg.)
      xi=(90.-.637*radang)/57.3
c refractive index of water
      rfind=1.33
c light reflection parameter (1)
      xsinr=(1./rfind)*sin(xi)
c angle of refraction from sunlight at water surface
      refrac=asin(xsinr)
c light reflection parameters (2,3)
      xsin1=(sin(xi-refrac))**2
      xsin2=(sin(xi+refrac))**2
c light reflection parameters (4,5)
      atnn1=(tan(xi-refrac))**2
      atnn2=(tan(xi+refrac))**2
c fraction of solar radiation refelected from water surface
      rflct=0.5*(xsin1/xsin2+atnn1/atnn2)
c ambient air temperature ( deg. C.) based on 24 hour clock; mean daily
c temperature(T), daylight fraction(DAYFRAC), and mean night temperature
c (ANTAV); bounded
      evtemp=t(lx)*dayfrc+(1.-dayfrc)*antav(lx)
      if (evtemp.le.0.0) evtemp=0.0
c energy reaching water surface, gram-calories/m2, minus that back-scattered
c from water layers; based on daily solar insolation (SOLRAD, ly/day)
      calri=snfrc(lx)*solrad*10000.
c light reflection of ice set to a constant
      if (watmp(1,lx).lt.0.0) rflct=0.4
c net energy available less that reflected from water, or used for
c evaporation of water
      calra=calri-2.*rflct*calri-(1200.+360.*evtemp)*540.
c temperature of sediment substrate set to 4.0 deg. C
      subst=4.0
c mean water temperature initialized
      watav=0.0
c same as CALRA
      calrb=calra
c same as SOLRAD
      solab=solrad
c
c ** water layer loop ***********************************************
      do 90 j=1,kdi
      aj=j
c phytoplankton extinction coefficient component initialized
      exta=0.0
c ** phytoplankton loop *********************************************
      do 80 k=1,5
c phytoplankton contribution to the light extinction coefficient estimated
c for each water layer
      exta=exta+cells(k,j,lx)*exip(k,lx)/(depthi(lx)*1000.)
   80 continue
c ** end phytoplankton loop *****************************************
c total extinction coefficient for water estimated from water and
```

```
c phytoplankton components
    extd=extc(lx)+exta
c effective linear penetration of light in water layer, meters
    dp=depthi(lx)/cos(refrac)
c daily energy absorbtion by water, gram-calories/day, adjusted for
c length greater than (DP)
    abscal=calrb-calrb*exp(-extd*(dp))
c daily insolation available for photosynthesis, ly/day
    absrad(j,lx)=solab*exp(-extd*(0.5*dp))
c energy and solar radiation not absorbed by water to depth (DP)
    calrb=calrb*exp(-extd*dp)
    solab=solab*exp(-extd*dp)
c water temperature of layer(J) adjusted for energy absorption/m2
    watmp(j,lx)=watmp(j,lx)+abscal/(depthi(lx)*1.0e06)
c energy lost from layer due to infrared radiation, gram-calories/day
c expressed as temperature change, deg. C.
    outcal=8.132e-11*(watmp(j,lx)+273.)**4
    outcal=outcal*60.*24./(depthi(lx)*100.)
c heat losses to upper and lower water layers (or to atmosphere if layer is
c at the surface expressed as temperature change, deg. C
    flxup(j,lx)=outcal*1.0
    flxdn(j,lx)=flxup(j,lx)
  90 continue
c ** end water layer loop ***********************************************
c heat transferred to bottom water layer from sediment surface, gram-calories
c /day (infrared) ; assumes 1m2 area and 1 cm depth into substrate; expressed
c as temperature change, deg. C.
    calsub=8.132e-11*(subst+273.)**4
    calsub=calsub*60.*24./(depthi(lx)*100.)
c heat transferred from ambient air to top 1cm (1m2 area) of top water layer,
c gram-calories/day; expressed as temperature change, deg. C.
    calatm=8.132e-11*(evtemp+273.)**4
    calatm=calatm*60.*24./(depthi(lx)*100.)
c
c ** water layer loop ***************************************************
    do 100 k=1,kdi
c sets initial downward heat flux to first layer as temperature change
    flxd=calatm
c sets all other layer downward fluxes to downward flux of upper layer
    if (k.gt.1) flxd=flxdn(k-1,lx)
c sets initial upward heat flux to bottom layer
    flxu=calsub
c sets all other layer upward heat fluxes of lower layers as temperature
c change
    if (k.ne.kdi) flxu=flxup(k+1,lx)
c water temperature of water layer(K) adjusted for radiative heat losses and
c gains, deg C.
    watmp(k,lx)=watmp(k,lx)-flxdn(k,lx)-flxup(k,lx)+flxu+flxd
c density of water layer estimated using regression equation, grams/cm3;
c bounded to density of ice at 0.0 deg. C.
    dense(k,lx)=1.-(watmp(k,lx)-4.)**2/(aden+bden*watmp(k,lx)-cden
   1 *watmp(k,lx)**2)
    if (watmp(k,lx).lt.0.0) dense(k,lx)=0.917
c effect of water temperature on zooplankton filtration rate fraction based
```

```
c on minimum(ZTMN) and maximum (ZTX) temperature tolerances; bounded
      tf(1,k,lx)=(ztmn(1,lx)-watmp(k,lx))/(ztmn(1,lx)-ztx(1,lx))
      tf(2,k,lx)=(ztmn(2,lx)-watmp(k,lx))/(ztmn(2,lx)-ztx(2,lx))
      if (tf(1,k,lx).le.0.0) tf(1,k,lx)=0.0
      if (tf(2,k,lx).le.0.0) tf(2,k,lx)=0.0
      if (tf(1,k,lx).ge.1.0) tf(1,k,lx)=1.0
      if (tf(2,k,lx).ge.1.0) tf(2,k,lx)=1.0
c adjusted filtration rate fractions for 2 types of zooplankton; bounded
      aing=1.-tf(1,k,lx)/3.+tf(1,k,lx)
      if (tf(1,k,lx).le.0.75) aing=tf(1,k,lx)/3.+tf(1,k,lx)
      tf(1,k,lx)=aing
      aing=1.-tf(2,k,lx)/3.+tf(2,k,lx)
      if (tf(2,k,lx).le.0.75) aing=tf(2,k,lx)/3.+tf(2,k,lx)
      tf(2,k,lx)=aing
c available biomass, ugrams, available to zooplankton types initialized
      zsm(1,k,lx)=0.0
      zsm(2,k,lx)=0.0
c total phytoplankton count/specified volume(liters) initialized for each
c water layer
      celct(k,lx)=0.0
  100 continue
c ** end water layer loop ***********************************************
c
c ** water layer loop ***************************************************
      do 110 l=2,kdi
c difference in density, g/cm3, between adjacent layers initialized
      dendif(l,lx)=0.0
c excludes lower layer densities equal or greater than adjacent upper layer
      if (dense(l,lx).ge.dense(l-1,lx)) go to 110
c estimates density difference between upper layer and lower layer, g/cm3
      dendif(l,lx)=dense(l-1,lx)-dense(l,lx)
  110 continue
c ** end water layer loop ***********************************************
c
c ** water layer loop ***************************************************
      do 130 l=2,kdi
c excludes equal density layers
      if (dendif(l,lx)) 131,130,131
c equilibrates upper and lower layer temperatures(deg.C.), density, PO4-
c phosphorus, NO3-nitrogen, and SiO2-silicon concentrations(ug/l)
  131 watmp(l,lx)=(watmp(l,lx)+watmp(l-1,lx))/2.
      watmp(l-1,lx)=watmp(l,lx)
      dense(l,lx)=(dense(l,lx)+dense(l-1,lx))/2.
      dense(l-1,lx)=dense(l,lx)
      phosp(l,lx)=(phosp(l,lx)+phosp(l-1,lx))/2.
      phosp(l-1,lx)=phosp(l,lx)
      anitr(l,lx)=(anitr(l,lx)+anitr(l-1,lx))/2.
      anitr(l-1,lx)=anitr(l,lx)
      silcn(l,lx)=(silcn(l,lx)+silcn(l-1,lx))/2.
      silcn(l-1,lx)=silcn(l,lx)
c
c ** phytoplankton loop ************************************************
      do 120 j=1,5
c upper and lower layers PO4-phosphorus, NO3-nitrogen, and SiO2-silicon
```

```
c concentrations equilibrated, ug/l
      qphosp(j,l,lx)=(qphosp(j,l,lx)+qphosp(j,l-1,lx))/2.
      qphosp(j,l-1,lx)=qphosp(j,l,lx)
      qnitro(j,l,lx)=(qnitro(j,l,lx)+qnitro(j,l-1,lx))/2.
      qnitro(j,l-1,lx)=qnitro(j,l,lx)
      qsilcn(j,l,lx)=(qsilcn(j,l,lx)+qsilcn(j,l-1,lx))/2.
      qsilcn(j,l-1,lx)=qsilcn(j,l,lx)
c upper and lower layers phytoplankton counts(per specified volume,liters)
c equilibrated
      cells(j,l,lx)=(cells(j,l,lx)+cells(j,l-1,lx))/2.
      cells(j,l-1,lx)=cells(j,l,lx)
c upper and lower layer phytoplankton cell biomass, ugrams, equilibrated
      cho(j,l,lx)=(cho(j,l,lx)+cho(j,l-1,lx))/2.
      cho(j,l-1,lx)=cho(j,l,lx)
  120 continue
c ** end phytplankton loop ********************************************
  130 continue
c ** end water layer loop *********************************************
c
c ** phytoplankton loop ************************************************
      do 210 j=1,5
c minimum phytoplankton cell biomass, ugrams
      pmax=akc(j,lx)
c total day-period PO4-phosphorus, NO3-nitrate, and SiO2-silicon
c incorporation, ug/day, based on photoperiod(PHOTPR)
      ump=umph(j,lx)*photpr
      umn=umnt(j,lx)*photpr
      ums=umsi(j,lx)*photpr
c
c ** water layer loop *************************************************
      do 210 k=1,kdi
c biomass increment, ug/day, initialized
      phtsyn=0.0
c photoperiod, hours
      ppp=photpr
c available energy (langleys) for photosynthesis in water layer(K)
      amp=(absrad(k,lx)*3.1416)/(2.0*ppp)
c integer value of photoperiod
      ipp=ppp
c
c **photoperiod loop(hours) *******************************************
      do 140 l=1,ipp
c hour number
      al=l
c incident radiation, langleys/hour in water layer(K)
      xiopt=amp*sin(3.1416*al/(2.*ppp))
c phytosynthesis biomass increment, ug/day, for phytoplankton(J), based on
c optimum insolation(AIOPT); bounded
      phtsyn=phtsyn+pmax*(xiopt/aiopt(j,lx))*exp(1.-xiopt/aiopt(j,lx))
  140 continue
c ** end photoperiod loop *********************************************
c phytoplankton respiration rate, ug/day, adjusted for total to maximum
c cell biomass(CMAX), and water temperature  related to optimum temperature
c for photosythesis(TOPM), and lower temperature tolerance(TLLV) for same;
```

```
c bounded to minimum cell biomass(AKC)
      rsp=resp(j,lx)*24.*(cho(j,k,lx)/cmax(j,lx))**0.67
      rsp=rsp*exp(2.3*(watmp(k,lx)-topm(j,lx))/(tllv(j,lx)-topm(j,lx)))
      if (cho(j,k,lx)-rsp.le.akc(j,lx)) rsp=cho(j,k,lx)-akc(j,lx)
c phytoplankton cell biomass adjusted for respiratory losses
      cho(j,k,lx)=cho(j,k,lx)-rsp
      if (phtsyn.le.0.0) phtsyn=0.0
c photosynthesis biomass increment bounded by maximum cell biomass(CMAX)
      if (cho(j,k,lx)+phtsyn.ge.cmax(j,lx)) phtsyn=cmax(j,lx)-cho(j,k,lx
     1 )
c phytoplankton(J) biomass incremented by photosynthesis increment, ugrams
      cho(j,k,lx)=cho(j,k,lx)+phtsyn
c cell division factor based on biomass; bounded
      cd=(cho(j,k,lx)-akc(j,lx))/cho(j,k,lx)
      if (cd.le.0.0) cd=0.0
c depth of layer , cm, in region(LX)
      volm=depthi(lx)*1000.
c estimated PO4-phosphorus uptake rate, ug/day, as a function of maximum
c (QMPH), actual(QPHOSP), and minimum(AKQPH) cell contents(ug); and PO4-
c phosphate concentration in water(ug/l), and the Michaelis-Menten constant
c (AKSPH); bounded to PO4 availability in water
      upp=ump*((qmph(j,lx)-qphosp(j,k,lx))/(qmph(j,lx)-akqph(j,lx)))
      upp=upp*(phosp(k,lx)/(aksph(j,lx)+phosp(k,lx)))
      if (upp*cells(j,k,lx).ge.phosp(k,lx)*volm) upp=phosp(k,lx)*volm
     1 /cells(j,k,lx)
c excludes phytoplankton that do not incorporate silica into cell walls
      if (j.ne.3) go to 150
c estimated SiO2-silicon uptake rate described and bounded as above for PO4
      us=ums*((qmsi(j,lx)-qsilcn(j,k,lx))/(qmsi(j,lx)-akqsi(j,lx)))
      us=us*(silcn(k,lx)/(akssi(j,lx)+silcn(k,lx)))
      if (us*cells(j,k,lx).ge.silcn(k,lx)*volm) us=silcn(k,lx)*volm
     1 /cells(j,k,lx)
c estimated NO3-nitrogen uptake rate described and bounded as above for PO4
  150 un=umn*((qmnt(j,lx)-qnitro(j,k,lx))/(qmnt(j,lx)-akqnt(j,lx)))
      un=un*(anitr(k,lx)/(aksnt(j,lx)+anitr(k,lx)))
      if (un*cells(j,k,lx).ge.anitr(k,lx)*volm) un=anitr(k,lx)*volm
     1 /cells(j,k,lx)
c total PO4-phosphorus,and NO3-nitrogen uptake bounded by maximum content
c QMPH, and QMNT, respectively, and not less than 0.0
      if (qphosp(j,k,lx)+upp.gt.qmph(j,lx)) upp=qmph(j,lx)-qphosp(j,k,lx
     1 )
      if (qnitro(j,k,lx)+un.gt.qmnt(j,lx)) un=qmnt(j,lx)-qnitro(j,k,lx)
      if (upp.le.0.0) upp=0.0
      if (un.le.0.0) un=0.0
c cell PO4-phosphorus and NO3-nitrogen incremented, ug
      qphosp(j,k,lx)=qphosp(j,k,lx)+upp
      qnitro(j,k,lx)=qnitro(j,k,lx)+un
c excludes cells not incorporating silica in cell walls
      if (j.ne.3) go to 160
c SiO2-silicon bounded by maximum content(QMSI), and not less than 0.0
      if (qsilcn(j,k,lx)+us.gt.qmsi(j,lx)) us=qmsi(j,lx)-qsilcn(j,k,lx)
      if (us.le.0.0) us=0.0
c cell SiO2-silicon incremented, ug
      qsilcn(j,k,lx)=qsilcn(j,k,lx)+us
```

```
c PO4-phosphorus and NO3-nitrate water concentrations adjusted for
c phytoplankton uptake, ug/l
 160 phosp(k,lx)=phosp(k,lx)-cells(j,k,lx)*upp/(1000.*depthi(lx))
     anitr(k,lx)=anitr(k,lx)-cells(j,k,lx)*un/(1000.*depthi(lx))
c excludes phytoplankton not incorporating SiO2 in cell walls
     if (j.ne.3) go to 170
c SiO2-silicon water concentration adjusted for phytoplankton uptake, ug/l
     silcn(k,lx)=silcn(k,lx)-cells(j,k,lx)*us/(1000.*depthi(lx))
c effect of cell PO4-phosphorus and NO3-nitrogen content on cell division
c rate
 170 xnd=(qphosp(j,k,lx)-akqph(j,lx))/qphosp(j,k,lx)
     xnd=xnd*(qnitro(j,k,lx)-akqnt(j,lx))/qnitro(j,k,lx)
c excludes phytoplankton not incorporating SiO2 into cell walls
     if (j.ne.3) go to 180
c additional effect of cell SiO2-silicon content on cell division
c rate; bounded
     xnd=xnd*(qsilcn(j,k,lx)-akqsi(j,lx))/qsilcn(j,k,lx)
c temperature factor for cell division set to lower tolerance temperature for
c photosynthesis, deg. C
 180 tfac=tllv(j,lx)
     if (xnd.le.0.0) xnd=0.0
c temperature factor set to upper tolerance temperature for photsynthesis for
c water temperatures above optimum
     if (watmp(k,lx).gt.topm(j,lx)) tfac=tulv(j,lx)
c effect of water temperature on cell division rate
     td=exp(-2.3*((watmp(k,lx)-topm(j,lx))/(tfac-topm(j,lx)))**2)
c cell division rate , per day, as a function of nutrients(XND),
c temperature(TD), and cell biomass(CD); bounded to maximum rate(DIVD), and
c not to be less than 0.0
     dv(j,k,lx)=divd(j,lx)*cd*td*xnd
     if (dv(j,k,lx).gt.divd(j,lx)) dv(j,k,lx)=divd(j,lx)
     if (dv(j,k,lx).le.0.0) dv(j,k,lx)=0.0
c cell division rate, per day; bounded to 1.0, and to minimum cell biomass
c (AKC)
     dvp=dv(j,k,lx)
     if (dv(j,k,lx).le.1.0) dv(j,k,lx)=1.0
     if (cho(j,k,lx)/dv(j,k,lx).lt.akc(j,lx)) dv(j,k,lx)=cho(j,k,lx)
     1 /akc(j,lx)
     if (dv(j,k,lx).le.1.0) dv(j,k,lx)=1.0
c death increment function estimated based on unadjusted division rate(DVP)
c ratio to maximum division rate(DIVID); bounded and incremented by 1.0
     if (dvp/divd(j,lx).gt..05) sg(j,k,lx)=0.0
     sg(j,k,lx)=sg(j,k,lx)+1.0
c cell death fraction estimated; bounded to cell concentration
     death=0.05*(1.-exp(-(.693/1.0)*sg(j,k,lx)))
     if (cells(j,k,lx).le.1.0) death=0.0
c PO4-phosphorus and NO3-nitrogen not utilizable by zooplankton, ug/day
     difp=cells(j,k,lx)*death*qphosp(j,k,lx)
     difn=cells(j,k,lx)*death*qnitro(j,k,lx)
c excludes cells not incorporating SiO2 into cell walls
     if (j.ne.3) go to 190
c SiO2-silicon not utilizable by zooplankton, ug/day
     difs=cells(j,k,lx)*death*qsilcn(j,k,lx)
c SiO2-silicon, PO4-phosphorus, and NO3-nitrate concentration of water in
```

```
c layer(K) incremented for lysed phytoplankton, ug/l
      silcn(k,lx)=silcn(k,lx)+difs/(1000.*depthi(lx))
  190 phosp(k,lx)=phosp(k,lx)+difp/(1000.*depthi(lx))
      anitr(k,lx)=anitr(k,lx)+difn/(1000.*depthi(lx))
c cell concentration adjusted for lysed cells, per 250 l or other specifed
      cells(j,k,lx)=cells(j,k,lx)-death*cells(j,k,lx)
c loss rate of cells that sink below epilimnion;bounded
      clost=(sink(j,lx)/2.5)*cells(j,k,lx)
      if (clost.gt.cells(j,k,lx)) clost=cells(j,k,lx)
c cell concentration  and cell minimum biomass bounded
      if (cells(j,k,lx).le.1.0) cells(j,k,lx)=1.0
      if (cho(j,k,lx).le.akc(j,lx)) cho(j,k,lx)=akc(j,lx)
c
c zooplankton metabolism
c
c ** zooplankton loop ***************************************************
      do 200 mm=1,2
c excludes zooplankton grazing at minimum cell concentrations
      if (cells(j,k,lx).le.1.0) go to 200
c zooplankton length, mm, based on zooplankton biomass(ug)
      almm=(zobio(mm,k,lx)/4.25)**(1./3.)
c phtyoplankton biomass consumption rate, ug/day, consumed by zooplankton
      eat(j,k,mm,lx)=0.0
c excludes zooplankton grazing at minimum cell concentrations
      if (cells(j,k,lx).le.1.0) go to 200
c zooplankton filtration rate, ml/day, based on zooplankton length(mm)
      filtr=24.*0.44*almm**(3.00)
c phytoplankton(J) cell concentration, per ml, in layer(K)
      celml=cells(j,k,lx)/(depthi(lx)*1.0e06)
c phytoplankton biomass consumption rate, ug/day, consumed by zooplankton
c based on effect of water temperature on filtration rate(TF), Michaelis-
c Menton half saturation constant for filtration(CELKI), estimated cell
c concentration(CELML), the filtration rate(FILTR), and biomass(ug) of the
c phytoplankton(J)
      eat(j,k,mm,lx)=tf(mm,k,lx)*(celml/(celki(mm,lx)+celml))*filtr
      eat(j,k,mm,lx)=eat(j,k,mm,lx)*cho(j,k,lx)*celml
c total biomass ,ug, required for zooplankton(MM) consumption/day
      zsm(mm,k,lx)=zsm(mm,k,lx)+eat(j,k,mm,lx)
c estimates total cell biomass per specified volume(250 l or specified)
      if (m.eq.1) celct(k,lx)=celct(k,lx)+cells(j,k,lx)*cho(j,k,lx)
  200 continue
c ** end zooplankton loop ***********************************************
  210 continue
c ** end phytoplankton loop **********************************************
c ** end water layer loop ***********************************************
c excludes zooplankton grazing
      if (skip-1.0) 251,250,251
c
c ** water layer loop ***************************************************
  251 do 220 k=1,kdi
c total phytoplankton consumption by zooplankton  per specified water volume,
c ug/day, initialized
      zosum(k,lx)=0.0
c
```

```
c zooplankton loop ******************************************************
      do 220 mm=1,2
c total phytoplankton consumption,ug/day by zooplankton per specified water
c volume(250 l,...specified)
      zosum(k,lx)=zosum(k,lx)+zoopc(mm,k,lx)*zsm(mm,k,lx)
  220 continue
c ** end water layer loop ***********************************************
c ** end zooplankton loop ************************************************
c
c ** zooplankton loop ****************************************************
      do 240 l=1,2
c
c ** water layer loop ***************************************************
      do 240 k=1,kdi
c fraction of cell biomass consumed by zooplankton in water layer(K); bounded
      celfac=0.0
      if (celct(k,lx).gt.0.0) celfac=zosum(k,lx)/celct(k,lx)
c zooplankton food requirement bounded to 0.45 depending on CELFAC
      if (zsm(l,k,lx).le.0.0) zsm(l,k,lx)=0.0
      zsmpr=zsm(l,k,lx)
      zsm(l,k,lx)=0.45*zsm(l,k,lx)
      if (celfac.gt.1.0) zsm(l,k,lx)=zsm(l,k,lx)/celfac
c
c zooplankton respiration rate, ug/day, based on zooplankton biomass(ZOBIO),
c and effect of temperature on filtration rate(TF); bounded depending on
c minimum zooplankton biomass(ZMIN)
c
      zoresp=0.008784*(zobio(l,k,lx)*.001*.53)**0.85
      zoresp=zoresp*(0.8*tf(l,k,lx)+0.4)
      zoresp=zoresp*24.*1000./.53
      if (zoresp.le.0.0) zoresp=0.0
      if (zobio(l,k,lx)-zoresp.le.zmin(l,lx)) zoresp=zobio(l,k,lx)-zmin
     1 (l,lx)
c zooplankton biomass adjusted for respiratory loss, ug/day; bounded to
c minimum(ZMIN) or maximum biomass(ZMAX)
      zobio(l,k,lx)=zobio(l,k,lx)-zoresp
      if (zobio(l,k,lx).le.zmin(l,lx)) zobio(l,k,lx)=zmin(l,lx)
      if (zobio(l,k,lx)+zsm(l,k,lx).ge.zmax(l,lx)) zsm(l,k,lx)=zmax(l,lx
     1 )-zobio(l,k,lx)
      if (zsm(l,k,lx).le.0.0) zsm(l,k,lx)=0.0
c zooplankton biomass incremented for phytoplankton intake; bounded to
c maximum (ZSM)
      zobio(l,k,lx)=zobio(l,k,lx)+zsm(l,k,lx)
      if (zobio(l,k,lx).ge.zmax(l,lx)) zobio(l,k,lx)=zmax(l,lx)
c
c ** phytoplankton loop *************************************************
      do 230 j=1,5
c phytoplankton cells consumed/day by zooplankton initialized
      eatfac=0.0
c minimum cell and zooplankton densities excluded
      if (cells(j,k,lx).le.1.0) go to 230
      if (zoopc(l,k,lx).le.1.0) go to 230
c cells consumed/day by zooplankton
      if (zsmpr.gt.0.0) eatfac=(eat(j,k,l,lx)/zsmpr)*zsm(l,k,lx)/cho(j,k
```

```
     1 ,lx)
c zooplankton PO4-phosphorus, NO3-nitrogen, and SiO2-silicon uptake, ug/d,
c nitrogen and phosphorus uptake bounded by maximum concentrations(ZMNT,
c ZMPH), and greater than or equal to 0.0
c
      upp=eatfac*qphosp(j,k,lx)
      un=eatfac*qnitro(j,k,lx)
      us=eatfac*qsilcn(j,k,lx)
      if (zphosp(l,k,lx)+upp.gt.zmph(l,lx)) upp=zmph(l,lx)-zphosp(l,k,lx
     1 )
      if (znitro(l,k,lx)+un.gt.zmnt(l,lx)) un=zmnt(l,lx)-znitro(l,k,lx)
      if (upp.le.0.0) upp=0.0
      if (un.le.0.0) un=0.0
      if (us.le.0.0) us=0.0
c PO4-phosphorus and NO3-nitrogen content, ug, of zooplankton incremented
      zphosp(l,k,lx)=zphosp(l,k,lx)+upp
      znitro(l,k,lx)=znitro(l,k,lx)+un
c un-used PO4-phosphorus, NO3-nitrate, and Si02-silicon recycled to water by
c zooplankton population; bounded
      difp=(eatfac*qphosp(j,k,lx)-upp)/(1000.*depthi(lx))
      difn=(eatfac*qnitro(j,k,lx)-un)/(1000.*depthi(lx))
      difs=(eatfac*qsilcn(j,k,lx))/(1000.*depthi(lx))
      difp=difp*zoopc(l,k,lx)
      difn=difn*zoopc(l,k,lx)
      difs=difs*zoopc(l,k,lx)
      if (difp.le.0.0) difp=0.0
      if (difn.le.0.0) difn=0.0
      if (difs.le.0.0) difs=0.0
c total phytoplankton cells/day consumed by zooplankton population(ZOOPC)
c utilized for growth
      eatfac=eatfac*zoopc(l,k,lx)
c total number of phytoplankton cells filtered (and lysed)
      totcel=eatfac/0.45
c PO4-phosphorus, NO3-nitrogen, and SiO2-silicon increments, ug/l, to
c existing water concentrations from non-utilized lysed phytoplankton
      phosp(k,lx)=phosp(k,lx)+difp+.55*totcel*qphosp(j,k,lx)/(1000.
     1 *depthi(lx))
      anitr(k,lx)=anitr(k,lx)+difn+.55*totcel*qnitro(j,k,lx)/(1000.
     1 *depthi(lx))
      silcn(k,lx)=silcn(k,lx)+difs+.55*totcel*qsilcn(j,k,lx)/(1000.
     1 *depthi(lx))
c cell concentrations(per 250 l or specified volume) adjusted for losses due
c to zooplankton grazing; bounded
      cells(j,k,lx)=cells(j,k,lx)-totcel
      if (cells(j,k,lx).le.1.0) cells(j,k,lx)=1.0
  230 continue
c ** end phytoplankton loop ******************************************
c un-utilized PO4-phosphorus, and NO3-nitrogen , ug/l, recycled to water;
c bounded due to mortality losses
      difp=.05*zoopc(l,k,lx)*zphosp(l,k,lx)/(1000.*depthi(lx))
      difn=.05*zoopc(l,k,lx)*znitro(l,k,lx)/(1000.*depthi(lx))
      if (zoopc(l,k,lx).le.1.0) difp=0.0
      if (zoopc(l,k,lx).le.1.0) difn=0.0
c PO4-phosphorus and NO3-nitrogen water concentrations incremented, ug/l,
```

```
      phosp(k,lx)=phosp(k,lx)+difp
      anitr(k,lx)=anitr(k,lx)+difn
c zooplankton population adjusted for mortality losses
      zoopc(l,k,lx)=zoopc(l,k,lx)-.05*zoopc(l,k,lx)
c zooplankton growth rate, per day; depending zooplankton biomass(ZMAX)
      dvz(l,k,lx)=1.0
      if (zobio(l,k,lx).ge.zmax(l,lx)) dvz(l,k,lx)=2.0
  240 continue
c ** end zooplankton loop ******************************************
c ** end water layer loop ******************************************
  250 continue
c
c ** phytoplankton loop ***********************************************
      do 260 j=1,5
c
c ** water layer loop *************************************************
      do 260 k=1,kdi
c phytoplankton cell density, cell biomass adjusted for growth
      cells(j,k,lx)=cells(j,k,lx)*dv(j,k,lx)
      cho(j,k,lx)=cho(j,k,lx)/dv(j,k,lx)
c phytoplankton biomass, ug/250 l, (or specified volume)
      phyto(j,k,lx)=cho(j,k,lx)*cells(j,k,lx)
c PO4-phosphorus, NO3-nitrogen, and SiO2-Silicon phytoplankton contents, ug,
c adjusted for reproduction
      qphosp(j,k,lx)=qphosp(j,k,lx)/dv(j,k,lx)
      qnitro(j,k,lx)=qnitro(j,k,lx)/dv(j,k,lx)
      qsilcn(j,k,lx)=qsilcn(j,k,lx)/dv(j,k,lx)
c excludes phytoplankton
      if (j.gt.2) go to 260
c zooplankton population incremented for reproduction; bounded
      if (zoopc(j,k,lx).le.1.0) zoopc(j,k,lx)=1.0
      zoopc(j,k,lx)=zoopc(j,k,lx)*dvz(j,k,lx)
c zooplankton biomass, ug, adjusted for reproduction(egg production_
      zobio(j,k,lx)=zobio(j,k,lx)/dvz(j,k,lx)
c zooplankton population biomass, ug/250 l (or specified volume)
      zoopm(j,k,lx)=zobio(j,k,lx)*zoopc(j,k,lx)
c NO3-nitrate and PO4-phosphorus zooplankton contents, ug, adjusted for
c reproduction
      znitro(j,k,lx)=znitro(j,k,lx)/dvz(j,k,lx)
      zphosp(j,k,lx)=zphosp(j,k,lx)/dvz(j,k,lx)
  260 continue
c ** end phytoplankton loop ******************************************
c ** end water layer loop ******************************************
      return
      end
```

14.5. AQUIFER

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine aquifer (lc)
c
c
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
c aquifer is incremented for return flow
      vaqf(lc)=vaqf(lc)+vrfl(lc)
c aquifer water surplus and losses initialized
      vqdx=0.0
      vloss(lc)=0.0
      if (vaqf(lc).le.0.0) vaqf(lc)=0.0
      if (vaqf(lc))12,11,12
   11 return
c vertical rise of channel segment using channel slope, feet;
c xchan equals length of channel, miles
   12 dch=schan(lc)*xchan(lc)*5280.
c horizontal distance across channel segemnt, feet
      hxh=sqrt((xchan(lc)*5280.)**2-dch**2)
c distance between bottom of aquifer and surface at posterior end
      rch=hbt(lc)+dch
c distance between bottom of aquifer and surface at anterior end
      h0h=rch-htp(lc)
      if (h0h.le.0.0) h0h=0.0
c length of channel at bottom of aquifer
      hpx=sqrt(h0h**2+hxh**2)
c slope of bottom of channel
      slopx(lc)=h0h/hxh
c coefficients for cubic equation of total aquifer volume
c (vertical slices)
      c33=(tbch(lc)+bbch(lc))*hbt(lc)/2.0
      c22=(hbt(lc)*(tch(lc)+bch(lc)-2.0*(bbch(lc)+tbch(lc)))
     1+htp(lc)*(tbch(lc)+bbch(lc)))/(4.0*hpx)
      c11=((tbch(lc)-tch(lc))*(hbt(lc)-htp(lc))+(bbch(lc)-bch(lc))
     1*(hbt(lc)-htp(lc)))/(6.0*hpx**2)
c coefficients for quadratic equation of temporal aquifer volume
c (horizontal slices)
      c111=(tbch(lc)-bbch(lc))/(4.0*hbt(lc))+(tch(lc)-bch(lc))/
     1(4.0*htp(lc))
      c222=(bbch(lc)+bch(lc))/2.0
c total aquifer volume, cubic feet, adjusted for porosity of .25
      vaqt(lc)=c11*hpx**3+c22*hpx**2+c33*hpx
      if (vaqf(lc)/0.25.le.vaqt(lc)) go to 60
c tests for aquifer overflow adjusted for porosity ,sets volume
c to maximum ,vaqt, cubic feet
      vqdx=4.0*vaqf(lc)-vaqt(lc)
      vqdx=vqdx/4.0
      vaqf(lc)=vaqt(lc)/4.0
```

```
c newton-cotes approximation of aquifer height from known
c moisture volume adjusted for porosity, feet
      htx(lc)=0.1
c
c ** iteration loop ******************************************************
      do 10 jv=1,10
      fxx=c111*hpx*htx(lc)**2+c222*hpx*htx(lc)
      fxx=fxx-vaqf(lc)*4.0
      fxd=2.0*c111*hpx*htx(lc)+c222*hpx
      htx(lc)=htx(lc)-fxx/fxd
   10 continue
c ** end iteration loop **************************************************
c tests for surplus water and presence of aquifer in channel
      if (vqdx.le.0.0) go to 50
      if (chnaq(lc)) 51,50,51
c estimates moisture thickness,mm, above region containing
c aquifer; region area (rarea) in acres
   51 jx=chnaq(lc)
      vmmx=vqdx*(12**3)*(25.4**3)/(rarea(jx)*4047.*1.0e06)
c apportions excess moisture thickness to soil horizon beginning
c with the bottom layer for each plant type
      ipll=apl(jx)
      vmmrs=0.0
c ** plant loop for area (jx) ********************************************
      do 30 jp=1,ipll
      iicdx=rcode(jp,jx)
c moisture thickness, mm
      vmm=vmmx
c ** soil layer loop *****************************************************
      do 20 kx=1,iicdx
      if (vmm.le.0.0) go to 20
      kk=iicdx-kx+2
c estimates moisture deficit to saturation
      vdxx=h(jp,kk,jx)-wa(jp,kk,jx)
      if (vdxx.le.0.0) go to 20
      if (vdxx.ge.vmm) vdxx=vmm
c adjusts current soil moisture
      wa(jp,kk,jx)=wa(jp,kk,jx)+vdxx
      vmm=vmm-vdxx
      if (vmm.le.0.0) vmm=0.0
c weights residual moisture thickness,mm, for crown cover of
c plant (jp)
      vmmrs=vmmrs+vmm*scvr(jp,jx)
   20 continue
c ** end soil layer loop *************************************************
   30 continue
c ** end plant loop ******************************************************
c residual surface moisture, cubic feet, on region (jx), day-1
      vqdx=vmmrs*rarea(jx)*4047.*1.0e06/((12**3)*(25.4**3))
c width of channel, feet, adjusted for aquifer height in feet
   50 wtx(lc)=bbch(lc)+((tbch(lc)-bbch(lc))/hbt(lc))*htx(lc)
c crossectional area, ft2, of saturated part of aquifer
      atx(lc)=htx(lc)*(bbch(lc)+wtx(lc))/2.
c
```

```
c loss of water through crossection of aquifer (end of segment),
c cfd, assuming 231 in/hour lateral hydraulic conductivity, and
c hydraulic gradient equal to the slope of the bottom segment
c
      vloss(lc)=2.0*atx(lc)*231.*slopx(lc)
      if (vloss(lc).ge.vaqf(lc)) vloss(lc)=vaqf(lc)
      vaqf(lc)=vaqf(lc)-vloss(lc)
      if (vaqf(lc).le.0.0) vaqf(lc)=0.0
c increments stream flow for residual flow (seepage), cfd
      vtrnd(lc)=vtrnd(lc)+vqdx
      vqdx=0.0
      return
   60 continue
c
c repeats height, area, and transmission losses from aquifer as
c above for undersaturated aquifer
c
      htx(lc)=0.1
c
c ** iteration loop ***********************************************
      do 70 jv=1,10
      fxx=c111*hpx*htx(lc)**2+c222*hpx*htx(lc)-vaqf(lc)/0.25
      fxd=2.*c111*hpx*htx(lc)+c222*hpx
      htx(lc)=htx(lc)-fxx/fxd
   70 continue
c ** end iteration loop *********************************************
      if (htx(lc).le.0.0) htx(lc)=0.0
      wtx(lc)=bbch(lc)+((tbch(lc)-bbch(lc))/hbt(lc))*htx(lc)
      atx(lc)=htx(lc)*(bbch(lc)+wtx(lc))/2.
      vloss(lc)=2.0*atx(lc)*231.*slopx(lc)
      if (vloss(lc).ge.vaqf(lc)) vloss(lc)=vaqf(lc)
c adjusts aquifer volume, cubic feet (cf)
      vaqf(lc)=vaqf(lc)-vloss(lc)
      if(vaqf(lc).le.0.0)vaqf(lc)=0.0
      return
      end
```

14.6. BIOPLT (bfm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine bioplt (lpx,lx,jct,y2,tyrt,tdyt,dmd,ymd,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
      real l1, l2, l3
      common/pa/ absw(120,ko), adia(120,ko), al4(m), al5(m),
     1bddt(120,ko),c66(m),degg(120,ko), egg(120,ko),
     2hatce(120,ko), hatch(120,ko),hatmp(120,ko),hattmp(120,ko),
     3ondv(30,ko), onyph(30,ko), pdiape(120,ko), thz(24), tmpday(30,ko)
     4,watcnr(ko,m,n),yadlt(30,ko),yndv(30,ko),
     5 ynyph(30,ko),hemlos(ko,m,n),bltgt(ko,m),fdtmp(ko),tpsm(ko),
     6 dvtmp(ko),adlt(2,ko),iegg(ko),iegd(ko),iegh(ko),eggs,deggs,
     7 hatchs, pdiaps,aha(ko),yadlts,ynyphs,onyphs,eglay,degday(ko),
     8 colday(ko),diaday(ko),eggmrt(ko),aii,summrt(ko,2)
c
      common/pc/ anday(12), clm(n), dnn(n), drmm(n), ffp(n), g1(n),g4(n)
     1 , g5(n), mmx(n), nx(24,n), pm(12,n), pms(12,n), pn(n),
     2 px(12,n), q(40,n), q8(n), r(n), t3(n), t4(n), tm(12,n), tms(12,n)
     3 , tn(n), tre(12,n), tx(12,n), x(12,n),relv(n),l2(n),l1(n),l3(n),
     4 prcv(n),tmxv(n),t1(n),t2(n),alpsr(n),alatd(n),rada(n),sunmn(n),
     5 sunmx(n)
c
      common/pd/ acut(i,2,n), bcut(i,2,n), ccut(i,2,n), datct(i,10,n),
     1 frt(n), racut(i,n), rbcut(i,n), rccut(i,n), satrs(m,n), tcut(i,10
     2 ,n), vcut(kt,n), z3wa2f(m,n)
c
      common/pe/ acts(m,n), actsm(m,n), frm(n), totac(m,n), age(i,n),v8
     1 (i,n)
c
      common/pf/ chlat(30,20), chne(30,20), entchn(30), regent(n),regx(n
     1 ,10), rfdcf(n), rfycf(n), rfylx(n), rnchn(30), rnylx(n), rycf(n),
     2 sarea(30), tnylx(n), tnytn(n), vtrndp(30)
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),wt1(kr,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      common/ph/ adsuc(n), ahrv(i,n), arig(i,n),
     1 pgr(n), rtrans(i,n),
```

```
     2 syav(i,n),
     3 xfac(i,n)
c
      common/pi/ a1hum(m,n), a1z(m,n), a1zs(m,n), a2hum(m,n), a2lp(m,n),
     1 a2tt(m,n), a2xlp(m,n), a3lp(m,n), a3tt(m,n), afrlp(m,n), afrtt(m
     2 ,n), afstt(m,n), aslp(m,n), b2lp(m,n), b3lp(m,n), bb2lp(m,n),
     3 bb3lp(m,n), cvtt(n), rb3lp(m,n), rbb3lp(m,n), rgain(m,n), sloss(m
     4 ,n), stem12(i,n), susfc(m,n), upk(n), wasumr(m,n), wasums(m,n),
     5 wz3lp(m,n), wz42lp(m,n), wz43lp(m,n), wz4lp(m,n), wz5lp(m,n),
     6 wz5x(i,m,n), wz5xlp(m,n), z1at(m,n), z1ave(m,n), z1lp1(m,n), z2lp
     7 (m,n), z3lp(m,n), z42lp(m,n), z43lp(m,n), z4g2(i,m,n), z4g2lp(m,n
     8 ), z4g3(i,m,n), z4g3lp(m,n), z4glp(m,n), z4lp(m,n), z4rlpl(m,n),
     9 z4tt(m,n), z4tts(m,n), z4ttsw(m,n), z4ttw(m,n), z5lp(m,n), z5rlpl
     1 (m,n), z5rtt(m,n), z5tt(m,n), z5x(i,m,n), z5xlp(m,n), zp(m,n),
     1 zsdlp(m,n), zup(m,n), dose(nq), evsp(m), egeop(m), excp(m),nppts
c
      common/pj/ awt(m), cdrp(ks,n), dsrt(m,n), hbar(ks,n), ph(i,ks,n),
     1 rock(i,ks,n), sand(i,ks,n), silt(i,ks,n), soilf(i,n), tim(i,n),
     2 w1bar(ks,n), watcn(m,n), wtf(n), yaz(m,n), ysrt(m,n),
     3 yup(m,n), z1dl(i,ks,m,n)
c
      common/pb/ akc(5,n),
     1 celct(kw,n),
     2 cho(5,kw,n), dendif(kw,n), dense(kw,n), depthi(n), dit
     3 (n), dv(5,kw,n), dvz(2,kw,n), eat(5,kw,2,n), epld(n),
     4 extc(n), flxdn(kw,n), flxup(kw,n),
     5 qnitro(5,kw,n), qphosp(5,kw,n), qsilcn(5,kw,n),
     6 sg(5,kw,n), snfrc(n), tf(2,kw,n),
     7 zmin(2,n),
     9 znitro(2,kw,n), zosum(kw,n), zphosp(2,kw,n), zsm(2,kw,n)
c
      common/pz/ tcnat(kr,2,ko),tcrel(200,kr,2,ko),
     1orgdos(6,2,ko),tcn(200,kr,2,ko),abscan(2,ko),
     2relcan(2,ko),sorgb(6,2,ko),spncan(2,ko),hasum(2,ko),popsum(2,ko),
     3orgbrd(6,2,ko),sorgd(6,2,ko),srun(ko),absr(ko),popfr(ko)
c
      common/pw/ gaslos(ko),urnlos(ko),wtbn(kr,ko),
     1wtkd(kr,ko),wtlv(kr,ko),wtms(kr,ko),wtrs(kr,ko),wtlu(kr,ko)
c
      common/ha/ popt(ko),tqx0(kr,2,ko),jage(ko),aiif(ko,2),ifm(kr),
     1 vurlfj(ko),vmljk(kr,ko),popp(2,ko),ikn(2,ko,8),sfpop(ko),
     2 gst(ko,2),gstn(ko),gestp(ko),aiiir(ko,2),agem(kr,ko),cagm(ko,2),
     3 cntk(2),brts(ko,2,2),frpr(2,ko),agcor(ko),tpppt(ko),sfd(8,ko),
     4tppcs(ko),tppps(ko),ca1(ko),clf(ko),lagf(ko),lua(ko),offsp(ko,3,2)
     5 ,affsp(ko,3),rat12(ko),tqp(kr,ko),tqs(kr,ko),vpelt(kr,ko,m),
     6vgut(kr,ko,m),cnmus(kr,ko,m),cnliv(kr,ko,m),cnkid(kr,ko,m),
     7cnfat(kr,ko,m),cnlun(kr,ko,m),cnbon(kr,ko,m),cnres(kr,ko,m),
     8cnblo(kr,ko,m),cnbod(kr,ko,m),cngut(kr,ko,m),cnpel(kr,ko,m),
     9vpln(kr,ko,m),vsol(kr,ko,m),vlug(kr,ko,m),cnpln(kr,ko,m),
     1cnsol(kr,ko,m),cnlug(kr,ko,m),rtpls(ko),rtmus(ko),rtkid(ko),
     2rtliv(ko),rtgut(ko),rthrt(ko),rtlun(ko),rtspl(ko),rtmar(ko),
     3rtfat(ko),rtres(ko),cbp(kr,ko,m),cbf(ko,m),cbw(ko,m),tacres(n),
     4vplt(kr,ko),c31plt(kr,ko,m),cnmlk(kr,ko,m),cnurnc(kr,ko,m),
     5cnfec(kr,ko,m),cnurn(kr,ko,m)
```

```
c
      dimension xpt(20), xptc(3,10), is(10,10), gmm(10),nv(20,10,10)
     1,aptc(10,10,10),varr(10,10),lar(10,10,10),nvv(10,10,10),
     2nvjv(20,10),alph(10,10),pltrc(10),regrc(10)

c
      character *10 lunt(10,10),lptc(10,10)
      character *10  iptc(10,10),kptc(10,10,10)
      character*1 iflag(14,10)
c excludes input from all but first access
      if (rdidx.gt.0.) go to 30
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5015
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-444444.)5702,5015,5702
 5555 nu=0
 5015 call atcol('white','green')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE BIOPLT                    ')
      call wnoust('******************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 444444.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
```

```
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
 5002 print *,'ENTER number of records desired for output..1,2,..'
      if(nu.eq.0)read(*,*,err=5002)rec
      if(nu.eq.1)read(7,*)rec
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*)rec
      if(iu.eq.1)print *,rec
      irec=rec
      do33j=1,irec
 5001 print *,'for record',j,'enter (1. or 0.) for formatted or'
      print *,'unformatted output, respectively; if graphics is'
      print *,'to be generated from the record, then enter 0.0; also'
      print *,'enter whether regional output (1. or 0.), and whether'
      print *,'plant output will be specified (1. or 0.). ENTER 3 input'
      print *,'variable now'
      if(nu.eq.0)read(*,*,err=5001) gmm(j),regrc(j),pltrc(j)
      if(nu.eq.1)read(7,*) gmm(j),regrc(j),pltrc(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write(6,*) gmm(j),regrc(j),pltrc(j)
      if(iu.eq.1)print *, gmm(j),regrc(j),pltrc(j)
   33 continue
c
c ** record loop ***************************************************
      do 1000 j=1,irec
c
 5003 print *,'ENTER specific subroutine output identifier code, and'
      print *,'daily(1.0) or yearly(0.0) output selector for record',j
      print *,'the identififer code should be even numbers(1.,2.,..etc.'
c
      if(nu.eq.0)read(*,*,err=5003) xptc(1,j),xptc(2,j)
      if(nu.eq.1)read(7,*) xptc(1,j),xptc(2,j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*) xptc(1,j),xptc(2,j)
      if(iu.eq.1)print *, xptc(1,j),xptc(2,j)
```

```
c**** excludes labels
      if(gmm(j).lt.1.0)goto5011
c
 5004 print *,'ENTER labels for 10 variables for record = ',j
      print *,'the labels must be no more than 10 columns wide. and'
      print *,'must be enclosed in single quotes: example " b1'"
      print *,'for blank labels enter an enclosed space: " '"
      print *,'10*" " is acceptable for a blank line'
c
      if(nu.eq.0)read(*,*,err=5004) (lptc(l,j),l=1,10)
 6000 format(8a10)
      if(nu.eq.1)read(7,6000) (lptc(l,j),l=1,10)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,6000) (lptc(l,j),l=1,10)
      if(iu.eq.1)print *, (lptc(l,j),l=1,10)
c enters units for output variables (10 labels, each 10 columns wide)
c
 5005 print *,'ENTER units for 10 variables for record = ',j
      print *,'the units must be no more than 10 columns wide, and'
      print *,'must be enclosed in single quotes: example " kg'"
      print *,'for blank units enter an enclosed space: " '"
      print *,'10*" " is acceptable for a blank line'
      if(nu.eq.0)read(*,*,err=5005) (lunt(l,j),l=1,10)
      if(nu.eq.1)read(7,6000) (lunt(l,j),l=1,10)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,6000) (lunt(l,j),l=1,10)
      if(iu.eq.1)print *, (lunt(l,j),l=1,10)
c
 5011 do1001k=1,10
 6001 format(a10)
c
 5008 print *,'output module, code=',xptc(1,j),';10 variables/record'
      print *,'ENTER variable identifier for variable',k,'from variable'
      print *,'library(exclude array part);and enclose in single quotes'
      print *,' example: b1(i,n) would be entered as "B1'"
c
      if(nu.eq.0)read(*,*,err=5008) iptc(k,j)
      if(nu.eq.1)read(7,6001) iptc(k,j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,6001) iptc(k,j)
      if(iu.eq.1)print *,iptc(k,j)
c
 5007 print *,'ENTER the number of array elements..2.0 for above ex;'
      print *,'if variable is not an array enter 0.0; also enter the'
      print *,'number equivalent of the first letter of the variable:'
      print *,'ex.. A=1, D=4, O=15...etc.. 2 inputs required'
c
      if(nu.eq.0)read(*,*,err=5007) varr(k,j),alph(k,j)
```

```
      if(nu.eq.1)read(7,*) varr(k,j),alph(k,j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*) varr(k,j),alph(k,j)
      if(iu.eq.1)print *,varr(k,j),alph(k,j)
      if(varr(k,j).le.0.0)goto1001
      ivar=varr(k,j)
c
 5009 print *,'ENTER array identifiers.."i","n".. for above example'
      print *,ivar,' inputs required'
      if(nu.eq.0)read(*,*,err=5009) (kptc(k,l,j),l=1,ivar)
      if(nu.eq.1)read(7,580) (kptc(k,l,j),l=1,ivar)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,580) (kptc(k,l,j),l=1,ivar)
      if(iu.eq.1)print *,(kptc(k,l,j),l=1,ivar)
c
 5010 print *,'ENTER specific output instructions for variable',k,':'
      print *,'if all values in array are to be printed, enter 0.0 for'
      print *,'the',ivar,'array values; if only part of the array is'
      print *,'then enter the highest array value for each index;'
      print *,'if only one value in an array is required, then enter'
      print *,'the negative of the value..ex. 0.,0. for all, 2.,0. for'
      print *,'first two values of "i" and all of "n" above; and'
      print *,'0.,-1. for all of "i"and only the first value of "n"'
      print *,ivar,' inputs required'
c
      if(nu.eq.0)read(*,*,err=5010) (aptc(k,l,j),l=1,ivar)
      if(nu.eq.1)read(7,*) (aptc(k,l,j),l=1,ivar)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*) (aptc(k,l,j),l=1,ivar)
      if(iu.eq.1)print *,(aptc(k,l,j),l=1,ivar)
c
 1001 continue
c ** end variable loop ********************************************
 1000 continue
c ** end record loop ************************************************
c
 5016 print *,'ENTER the year number when output is to begin'
      if(nu.eq.0)read(*,*,err=5016) styr
      if(nu.eq.1)read(7,*) styr
      if(nu.eq.0)print *,'do you wish to re-enter line? Enter y=1, no=0'
      if(nu.eq.0)read(*,*)ans
      if(ans.gt.0.0.and.nu.eq.0)goto5016
      write(6,*) styr
      if(iu.eq.1)print *,styr
c
 5012 print *,'ENTER the number of automatic plots desired for this'
      print *,'simulation; a value of 0.0 by-passes option'
c
```

```
      if(nu.eq.0)read(*,*,err=5012) aplots
      if(nu.eq.1)read(7,*) aplots
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5012) ans
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*) aplots
      if(iu.eq.1)print *, aplots
      nplots=aplots
      if(nplots.eq.0)call clear_screen@
      if(nplots.eq.0)goto 5013
      do5014jp=1,nplots
      nplts=jp
      ngrap=jp
      call clear_screen@
      nut=nu
      nu=nuu
      call truplt (krec,jp,xpt,lx,nplts,iflag,rdidx)
      nu=nut
      call clear_screen@
      nup=nu
      nu=nuu
      call tplotr (krec,jp,lx,ngrap,iflag,rdidx)
      nu=nup
      call clear_screen@
 5014 continue
c
 5013 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN BIOPLT                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for BIOPLT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5013) ans
      if(ans)1002,5500,1002
 1002 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-444444.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto5015
 5501 continue
 5500 continue
      nu=nuu
```

```
      call clear_screen@
c ** end record loop ******************************************************
c
c ** variable loop *********************************************************
      do 2000 j=1,20
c output variables initialized
 2000 xpt(j)=0.0
c ** end variable loop ****************************************************
      sajj=0.0
      do2001k=1,10
      do2001l=1,10
      is(k,l)=0.0
 2001 continue
      lpx=1
      lz=1
      return
   30 continue
      if(ai1.lt.styr)return
      jct=jct+1
c ** record loop *********************************************************
      do 540 j=1,irec
      if(jct.lt.ipx)goto540
      if(jct.lt.ipl)goto540
      do671jj=1,20
      do671kk=1,10
      do671ll=1,10
      nv(jj,kk,ll)=-1
  671 continue
      nvjx=0
      do771kk=1,20
      do771ll=1,10
      nvjv(kk,ll)=0
  771 continue
c ** variable loop *******************************************************
      do556k=1,10
      ivar=varr(k,j)
      if(ivar.eq.0)goto556
c ** indices loop ********************************************************
c** searches for numerical indicies
      do555l=1,ivar
      nvv(k,l,j)=0
      if(kptc(k,l,j).eq. '1')nvv(k,l,j)=1
      if(kptc(k,l,j).eq. '2')nvv(k,l,j)=2
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '5')nvv(k,l,j)=5
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '7')nvv(k,l,j)=7
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '8')nvv(k,l,j)=8
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '10')nvv(k,l,j)=10
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '12')nvv(k,l,j)=12
      if(nvv(k,l,j).gt.0)goto555
```

```
      if(kptc(k,l,j).eq. '16')nvv(k,l,j)=16
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '18')nvv(k,l,j)=18
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '22')nvv(k,l,j)=22
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '24')nvv(k,l,j)=24
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '30')nvv(k,l,j)=30
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '32')nvv(k,l,j)=32
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '36')nvv(k,l,j)=36
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '100')nvv(k,l,j)=100
      if(nvv(k,l,j).gt.0)goto555
      if(kptc(k,l,j).eq. '120')nvv(k,l,j)=120
  555 continue
c ** end indices loop
  556 continue
c ** end variable loop
      xpt(11)=0.0
      xpt(14)=0.0
      xpt(15)=0.0
c
c ** variable loop
c ** regional variable selected first ***************************
      do650kk=1,10
      ivar=varr(kk,j)
      if(ivar.eq.0)goto650
      do651ll=1,ivar
      if(kptc(kk,ll,j).ne.'n')goto651
      xpt(kk)=0.0
      jptc=aptc(kk,ll,j)
      if(aptc(kk,ll,j))600,601,602
  600 lar(kk,ll,j)=-jptc
      if(-jptc.ne.lx)goto651
      lz=lar(kk,ll,j)
      xpt(15)=lz
      goto651
  602 if(lz.gt.jptc)goto651
  601 lar(kk,ll,j)=lx
      lz=lar(kk,ll,j)
      xpt(15)=lz
  651 continue
  650 continue
c
c *** plant variable selected second ******************************
      do654kk=1,10
      ivar=varr(kk,j)
      if(ivar.eq.0)goto654
      do655ll=1,ivar
      if(kptc(kk,ll,j).ne.'i')goto655
      xpt(kk)=0.0
```

```
      jptc=aptc(kk,ll,j)
      if(aptc(kk,ll,j))603,604,605
  603 lar(kk,ll,j)=-jptc
      if(-jptc.ne.lpx)goto655
      xpt(14)=-jptc
      goto655
  605 if(lpx.gt.jptc)goto655
  604 lar(kk,ll,j)=lpx
      xpt(14)=lpx
  655 continue
  654 continue
c     if(regrc(j).gt.0.0.and.lx.lt.ipx)return
c     if(pltrc(j).gt.0.0.and.lpx.lt.ipl)return
      if(lx.lt.ipx)return
      if(lpx.lt.ipl)return
c
c ** remaining variables ***********************************
      kicde=rcode(lpx,lz)
      nlyr=dit(lz)
      ntoo=nan+nhm
      lage=kr
      jnz=inz
      jds=nq
  657 continue
      nvjx=nvjx+1
      do656 jx=nvjx,nvjx
      kage=agea(nvjx)
c** variable loop ****************************************
      do542k=1,10
      ivar=varr(k,j)
      if(ivar.eq.0)goto542
c ** indices loop ****************************************
      do541l=1,ivar
      jptc=aptc(k,l,j)
      if(kptc(k,l,j).ne.'ks')goto546
      xpt(k)=0.0
      if(jptc.lt.0)kicde=1
      if(jptc.gt.0)kicde=jptc
      if(nv(k,l,j).ne.0)nv(k,l,j)=kicde
      nvjv(k,l)=nvjv(k,l)+1
      if(nvjv(k,l).gt.kicde)nvjv(k,l)=1
      if(nvjv(k,l).ge.kicde)nv(k,l,j)=0
      if(aptc(k,l,j))640,641,641
  640 lar(k,l,j)=-jptc
      xpt(11)=-jptc
      goto541
  641 lar(k,l,j)=nvjv(k,l)
      xpt(11)=nvjv(k,l)
      goto541
c
  546 if(kptc(k,l,j).ne.'kt')goto547
      xpt(k)=0.0
      ktt=kt
      if(jptc.lt.0)ktt=1
```

```
      if(jptc.gt.0)ktt=jptc
      if(nv(k,l,j).ne.0)nv(k,l,j)=ktt
      nvjv(k,l)=nvjv(k,l)+1
      if(nvjv(k,l).gt.ktt)nvjv(k,l)=1
      if(nvjv(k,l).ge.ktt)nv(k,l,j)=0
      if(aptc(k,l,j))609,610,610
  609 lar(k,l,j)=-jptc
      xpt(11)=-jptc
      goto541
  610 lar(k,l,j)=nvjv(k,l)
      xpt(11)=nvjv(k,l)
      goto541
c
  547 if(kptc(k,l,j).ne.'kw')goto548
      xpt(k)=0.0
      if(jptc.lt.0)nlyr=1
      if(jptc.gt.0)nlyr=jptc
      if(nv(k,l,j).ne.0)nv(k,l,j)=nlyr
      nvjv(k,l)=nvjv(k,l)+1
      if(nvjv(k,l).ge.nlyr)nvjv(k,l)=1
      if(nvjv(k,l).ge.nlyr)nv(k,l,j)=0
      if(aptc(k,l,j))612,613,613
  612 lar(k,l,j)=-jptc
      xpt(11)=-jptc
      goto541
  613 lar(k,l,j)=nvjv(k,l)
      xpt(11)=nvjv(k,l)
      goto541
c
  548 if(kptc(k,l,j).ne.'ko')goto653
      xpt(k)=0.0
      if(jptc.lt.0)ntoo=1
      if(jptc.gt.0)ntoo=jptc
      if(nv(k,l,j).ne.0)nv(k,l,j)=ntoo
      nvjv(k,l)=nvjv(k,l)+1
      if(nvjv(k,l).gt.ntoo)nvjv(k,l)=1
      if(nvjv(k,l).ge.ntoo)nv(k,l,j)=0
      if(aptc(k,l,j))606,607,607
  606 lar(k,l,j)=-jptc
      xpt(15)=-jptc
      goto541
  607 lar(k,l,j)=nvjv(k,l)
      xpt(15)=nvjv(k,l)
      goto541
c
  653 if(kptc(k,l,j).ne.'ke')goto549
      xpt(k)=0.0
      if(jptc.lt.0)kage=1
      if(jptc.gt.0)kage=jptc
      if(nv(k,l,j).ne.0)nv(k,l,j)=kage
      nvjv(k,l)=nvjv(k,l)+1
      if(nvjv(k,l).gt.kage)nvjv(k,l)=1
      if(nvjv(k,l).ge.kage)nv(k,l,j)=0
      if(aptc(k,l,j))615,616,616
```

```
 615 lar(k,l,j)=-jptc
     xpt(14)=-jptc
     goto541
 616 lar(k,l,j)=nvjv(k,l)
     xpt(14)=nvjv(k,l)
     goto541
c
 549 if(kptc(k,l,j).ne.'kr')goto550
     xpt(k)=0.0
     if(jptc.lt.0)lage=1
     if(jptc.gt.0)lage=jptc
     if(nv(k,l,j).ne.0)nv(k,l,j)=lage
     nvjv(k,l)=nvjv(k,l)+1
     if(nvjv(k,l).gt.lage)nvjv(k,l)=1
     if(nvjv(k,l).ge.lage)nv(k,l,j)=0
     if(aptc(k,l,j))618,619,619
 618 lar(k,l,j)=-jptc
     xpt(14)=-jptc
     goto541
 619 lar(k,l,j)=nvjv(k,l)
     xpt(14)=nvjv(k,l)
     goto541
c
 550 if(kptc(k,l,j).ne.'m')goto551
     xpt(k)=0.0
     if(jptc.lt.0)jnz=1
     if(jptc.gt.0)jnz=jptc
     if(nv(k,l,j).ne.0)nv(k,l,j)=jnz
     nvjv(k,l)=nvjv(k,l)+1
     if(nvjv(k,l).gt.jnz)nvjv(k,l)=1
     if(nvjv(k,l).ge.jnz)nv(k,l,j)=0
     if(aptc(k,l,j))621,622,622
 621 lar(k,l,j)=-jptc
     xpt(11)=-jptc
     goto541
 622 lar(k,l,j)=nvjv(k,l)
     xpt(11)=nvjv(k,l)
     goto541
c
 551 if(kptc(k,l,j).ne.'nq')goto552
     xpt(k)=0.0
     if(jptc.lt.0)jds=1
     if(jptc.gt.0)jds=jptc
     if(nv(k,l,j).ne.0)nv(k,l,j)=jds
     nvjv(k,l)=nvjv(k,l)+1
     if(nvjv(k,l).gt.jds)nvjv(k,l)=1
     if(nvjv(k,l).ge.jds)nv(k,l,j)=0
     if(aptc(k,l,j))624,625,625
 624 lar(k,l,j)=-jptc
     xpt(11)=-jptc
     goto541
 625 lar(k,l,j)=nvjv(k,l)
     xpt(11)=nvjv(k,l)
     goto541
```

```
c
 552 if(nvv(k,l,j).eq.0)goto541
    xpt(k)=0.0
    if(jptc.lt.0)nvv(k,l,j)=1
    if(jptc.gt.0)nvv(k,l,j)=jptc
    if(nv(k,l,j).ne.0)nv(k,l,j)=nvv(k,l,j)
    nvjv(k,l)=nvjv(k,l)+1
    if(nvjv(k,l).gt.nvv(k,l,j))nvjv(k,l)=1
    if(nvjv(k,l).ge.nvv(k,l,j))nv(k,l,j)=0
    if(aptc(k,l,j))627,628,628
 627 lar(k,l,j)=-jptc
    xpt(11)=-jptc
    goto541
 628 lar(k,l,j)=nvjv(k,l)
    xpt(11)=nvjv(k,l)
 541 continue
c ** end indicies loop
 542 continue
c ** end variable loop
    do543k=1,10
    j1=lar(k,1,j)
    j2=lar(k,2,j)
    j3=lar(k,3,j)
    j4=lar(k,4,j)
    j5=lar(k,5,j)
    j6=lar(k,6,j)
    j7=lar(k,7,j)
    j8=lar(k,8,j)
    j9=lar(k,9,j)
    j10=lar(k,10,j)
    iap=alph(k,j)
    goto(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
    124,25,26),iap
    1 if(iptc(k,j).eq.'A')xpt(k)=a(j1,j2,j3)
    if(iptc(k,j).eq.'A1')xpt(k)=a1(j1,j2,j3)
    if(iptc(k,j).eq.'A1AV')xpt(k)=a1av(j1,j2,j3)
    if(iptc(k,j).eq.'A1HUM')xpt(k)=a1hum(j1,j2)
    if(iptc(k,j).eq.'A1TH')xpt(k)=a1th(j1,j2)
    if(iptc(k,j).eq.'A1TT')xpt(k)=a1tt(j1,j2)
    if(iptc(k,j).eq.'A1TG')xpt(k)=a1tg(j1,j2)
    if(iptc(k,j).eq.'A1Z')xpt(k)=a1z(j1,j2)
    if(iptc(k,j).eq.'A1ZS')xpt(k)=a1zs(j1,j2)
    if(iptc(k,j).eq.'A2')xpt(k)=a2(j1,j2,j3)
    if(iptc(k,j).eq.'A22')xpt(k)=a22(j1,j2)
    if(iptc(k,j).eq.'A23')xpt(k)=a23(j1,j2)
    if(iptc(k,j).eq.'A2HUM')xpt(k)=a2hum(j1,j2)
    if(iptc(k,j).eq.'A2TT')xpt(k)=a2tt(j1,j2)
    if(iptc(k,j).eq.'A2X')xpt(k)=a2x(j1,j2,j3)
    if(iptc(k,j).eq.'A3')xpt(k)=a3(j1,j2,j3)
    if(iptc(k,j).eq.'A3AC')xpt(k)=a3ac(j1,j2,j3)
    if(iptc(k,j).eq.'A3AV')xpt(k)=a3av(j1,j2,j3)
    if(iptc(k,j).eq.'A3C')xpt(k)=a3c(j1,j2,j3)
    if(iptc(k,j).eq.'A3TT')xpt(k)=a3tt(j1,j2)
    if(iptc(k,j).eq.'AA2')xpt(k)=aa2(j1)
```

```
if(iptc(k,j).eq.'AAA')xpt(k)=aaa(j1,j2,j3)
if(iptc(k,j).eq.'AATB')xpt(k)=aatb(j1,j2,j3)
if(iptc(k,j).eq.'ABSCAN')xpt(k)=abscan(j1,j2)
if(iptc(k,j).eq.'ABSR')xpt(k)=absr(j1)
if(iptc(k,j).eq.'ABSRAD')xpt(k)=absrad(j1,j2)
if(iptc(k,j).eq.'ABSW')xpt(k)=absw(j1,j2)
if(iptc(k,j).eq.'AC')xpt(k)=ac
if(iptc(k,j).eq.'ACRES')xpt(k)=acres(j1,j2)
if(iptc(k,j).eq.'ACTAV')xpt(k)=actav(j1,j2,j3)
if(iptc(k,j).eq.'ACTI')xpt(k)=acti(j1)
if(iptc(k,j).eq.'ACTS')xpt(k)=acts(j1,j2)
if(iptc(k,j).eq.'ACTSM')xpt(k)=actsm(j1,j2)
if(iptc(k,j).eq.'ACUT')xpt(k)=acut(j1,j2,j3)
if(iptc(k,j).eq.'ADDWAT')xpt(k)=addwat
if(iptc(k,j).eq.'ADIA')xpt(k)=adia(j1,j2)
if(iptc(k,j).eq.'ADLT')xpt(k)=adlt(j1,j2)
if(iptc(k,j).eq.'ADSUC')xpt(k)=adsuc(j1)
if(iptc(k,j).eq.'AFR')xpt(k)=afr(j1,j2,j3)
if(iptc(k,j).eq.'AFRTT')xpt(k)=afrtt(j1,j2)
if(iptc(k,j).eq.'AFS')xpt(k)=afs(j1,j2,j3)
if(iptc(k,j).eq.'AFSAC')xpt(k)=afsac(j1,j2,j3)
if(iptc(k,j).eq.'AFSAV')xpt(k)=afsav(j1,j2,j3)
if(iptc(k,j).eq.'AFSC')xpt(k)=afsc(j1,j2,j3)
if(iptc(k,j).eq.'AFSTT')xpt(k)=afstt(j1,j2)
if(iptc(k,j).eq.'AGAVE')xpt(k)=agave
if(iptc(k,j).eq.'AGE')xpt(k)=age(j1,j2)
if(iptc(k,j).eq.'AGEA')xpt(k)=agea(j1)
if(iptc(k,j).eq.'AGR')xpt(k)=agr(j1,j2)
if(iptc(k,j).eq.'AGTB')xpt(k)=agtb(j1,j2,j3)
if(iptc(k,j).eq.'AGZ')xpt(k)=agz(j1)
if(iptc(k,j).eq.'AHA')xpt(k)=aha(j1)
if(iptc(k,j).eq.'AHRV')xpt(k)=ahrv(j1,j2)
if(iptc(k,j).eq.'AI1')xpt(k)=ai1
if(iptc(k,j).eq.'AICE')xpt(k)=(j1)
if(iptc(k,j).eq.'AICHE')xpt(k)=aiche(j1)
if(iptc(k,j).eq.'AII')xpt(k)=aii
if(iptc(k,j).eq.'AIII')xpt(k)=aiii
if(iptc(k,j).eq.'AIOPT')xpt(k)=aiopt(j1,j2)
if(iptc(k,j).eq.'AIRCA')xpt(k)=airca(j1,j2)
if(iptc(k,j).eq.'AIRCG')xpt(k)=aircg(j1,j2)
if(iptc(k,j).eq.'AIRX')xpt(k)=airx(j1)
if(iptc(k,j).eq.'AKC')xpt(k)=akc(j1,j2)
if(iptc(k,j).eq.'AKOND')xpt(k)=akond(j1)
if(iptc(k,j).eq.'AKQNT')xpt(k)=akqnt(j1,j2)
if(iptc(k,j).eq.'AKQPH')xpt(k)=akqph(j1,j2)
if(iptc(k,j).eq.'AKQSI')xpt(k)=akqsi(j1,j2)
if(iptc(k,j).eq.'AKSNT')xpt(k)=aksnt(j1,j2)
if(iptc(k,j).eq.'AKSOIL')xpt(k)=aksoil(j1)
if(iptc(k,j).eq.'AKSPH')xpt(k)=aksph(j1,j2)
if(iptc(k,j).eq.'AKSSI')xpt(k)=akssi(j1,j2)
if(iptc(k,j).eq.'AL4')xpt(k)=al4(j1)
if(iptc(k,j).eq.'AL5')xpt(k)=al5(j1)
if(iptc(k,j).eq.'ALATD')xpt(k)=alatd(j1)
if(iptc(k,j).eq.'ALPSR')xpt(k)=alpsr(j1)
```

```
if(iptc(k,j).eq.'ALSOIL')xpt(k)=alsoil(j1)
if(iptc(k,j).eq.'ALSP')xpt(k)=alsp(j1)
if(iptc(k,j).eq.'ALTNT')xpt(k)=altnt(j1,j2,j3)
if(iptc(k,j).eq.'ALTYR')xpt(k)=altyr(j1,j2,j3)
if(iptc(k,j).eq.'AMW')xpt(k)=amw(j1,j2)
if(iptc(k,j).eq.'AMX')xpt(k)=amx(j1,j2)
if(iptc(k,j).eq.'ANDAY')xpt(k)=anday(j1)
if(iptc(k,j).eq.'ANITR')xpt(k)=anitr(j1,j2)
if(iptc(k,j).eq.'ANTAV')xpt(k)=antav(j1)
if(iptc(k,j).eq.'ANTRO')xpt(k)=antro(j1)
if(iptc(k,j).eq.'ANTROS')xpt(k)=antros(j1)
if(iptc(k,j).eq.'APL')xpt(k)=apl(j1)
if(iptc(k,j).eq.'APLX')xpt(k)=aplx(j1)
if(iptc(k,j).eq.'AQUGAT')xpt(k)=aqugat
if(iptc(k,j).eq.'ARIG')xpt(k)=arig(j1,j2)
if(iptc(k,j).eq.'AS')xpt(k)=as(j1,j2,j3)
if(iptc(k,j).eq.'ASAV')xpt(k)=asav(j1,j2,j3)
if(iptc(k,j).eq.'ASTP')xpt(k)=astp(j1,j2)
if(iptc(k,j).eq.'ATNO')xpt(k)=atno(j1)
if(iptc(k,j).eq.'ATTT')xpt(k)=attt(j1,j2)
if(iptc(k,j).eq.'ATX')xpt(k)=atx(j1)
if(iptc(k,j).eq.'AUTZD')xpt(k)=autzd(j1,j2,j3)
if(iptc(k,j).eq.'AWATA')xpt(k)=awata(j1,j2,j3)
if(iptc(k,j).eq.'AWT')xpt(k)=awt(j1)
goto543
2 if(iptc(k,j).eq.'B1')xpt(k)=b1(j1,j2)
if(iptc(k,j).eq.'B2')xpt(k)=b2(j1,j2,j3)
if(iptc(k,j).eq.'B3')xpt(k)=b3(j1,j2,j3)
if(iptc(k,j).eq.'B3AC')xpt(k)=b3ac(j1,j2,j3)
if(iptc(k,j).eq.'B3AV')xpt(k)=b3av(j1,j2,j3)
if(iptc(k,j).eq.'B3C')xpt(k)=b3c(j1,j2,j3)
if(iptc(k,j).eq.'B5')xpt(k)=b5(j1,j2)
if(iptc(k,j).eq.'B7')xpt(k)=b7(j1,j2)
if(iptc(k,j).eq.'BAR')xpt(k)=bar(j1,j2)
if(iptc(k,j).eq.'BARSQ')xpt(k)=barsq(j1,j2)
if(iptc(k,j).eq.'BATR')xpt(k)=batr
if(iptc(k,j).eq.'BB2')xpt(k)=bb2(j1,j2,j3)
if(iptc(k,j).eq.'BB3')xpt(k)=bb3(j1,j2,j3)
if(iptc(k,j).eq.'BB3AC')xpt(k)=bb3ac(j1,j2,j3)
if(iptc(k,j).eq.'BB3AV')xpt(k)=bb3av(j1,j2,j3)
if(iptc(k,j).eq.'BB3C')xpt(k)=bb3c(j1,j2,j3)
if(iptc(k,j).eq.'BBCH')xpt(k)=bbch(j1)
if(iptc(k,j).eq.'BBP')xpt(k)=bbp(j1,j2)
if(iptc(k,j).eq.'BCH')xpt(k)=bch(j1)
if(iptc(k,j).eq.'BCP')xpt(k)=bcp(j1,j2)
if(iptc(k,j).eq.'BCUT')xpt(k)=bcut(j1,j2,j3)
if(iptc(k,j).eq.'BDDT')xpt(k)=bddt(j1,j2)
if(iptc(k,j).eq.'BDRK')xpt(k)=bdrk(j1)
if(iptc(k,j).eq.'BIRTHS')xpt(k)=births(j1,j2)
if(iptc(k,j).eq.'BLTBN1')xpt(k)=bltbn1(j1,j2)
if(iptc(k,j).eq.'BLTBN2')xpt(k)=bltbn2(j1,j2)
if(iptc(k,j).eq.'BLTGT')xpt(k)=bltgt(j1,j2)
if(iptc(k,j).eq.'BLTKD1')xpt(k)=bltkd1(j1,j2)
if(iptc(k,j).eq.'BLTKD2')xpt(k)=bltkd2(j1,j2)
```

```
      if(iptc(k,j).eq.'BLTLU')xpt(k)=bltlu(j1,j2)
      if(iptc(k,j).eq.'BLTLV')xpt(k)=bltlv(j1,j2)
      if(iptc(k,j).eq.'BLTMK')xpt(k)=bltmk(j1,j2)
      if(iptc(k,j).eq.'BLTMS')xpt(k)=bltms(j1,j2)
      if(iptc(k,j).eq.'BLTRS')xpt(k)=bltrs(j1,j2)
      if(iptc(k,j).eq.'BLTUR')xpt(k)=bltur(j1,j2)
      if(iptc(k,j).eq.'BMN')xpt(k)=bmn(j1,j2)
      if(iptc(k,j).eq.'BNTBL1')xpt(k)=bntbl1(j1,j2)
      if(iptc(k,j).eq.'BNTBL2')xpt(k)=bntbl2(j1,j2)
      if(iptc(k,j).eq.'BRRT')xpt(k)=brrt(j1,j2,j3)
      if(iptc(k,j).eq.'BRTY')xpt(k)=brty(j1,j2)
      if(iptc(k,j).eq.'BSD')xpt(k)=bsd
      if(iptc(k,j).eq.'BSMAX')xpt(k)=bsmax(j1,j2)
      goto543
    3 if(iptc(k,j).eq.'C1HR')xpt(k)=c1hr(j1)
      if(iptc(k,j).eq.'C2')xpt(k)=c2
      if(iptc(k,j).eq.'C2HR')xpt(k)=c2hr(j1)
      if(iptc(k,j).eq.'C3')xpt(k)=c3(j1)
      if(iptc(k,j).eq.'C31')xpt(k)=c31(j1,j2,j3)
c *** new
      if(iptc(k,j).eq.'C31PLT')xpt(k)=c31plt(j1,j2,j3)
      if(iptc(k,j).eq.'C31TT')xpt(k)=c31tt(j1,j2)
      if(iptc(k,j).eq.'C4')xpt(k)=c4
      if(iptc(k,j).eq.'C5')xpt(k)=c5
      if(iptc(k,j).eq.'C56')xpt(k)=c56(j1)
      if(iptc(k,j).eq.'C5HR')xpt(k)=c5hr(j1)
      if(iptc(k,j).eq.'C6')xpt(k)=c6(j1,j2)
      if(iptc(k,j).eq.'C66')xpt(k)=c66(j1)
      if(iptc(k,j).eq.'C7')xpt(k)=c7
      if(iptc(k,j).eq.'C8')xpt(k)=c8(j1,j2)
      if(iptc(k,j).eq.'C9')xpt(k)=c9(j1,j2)
      if(iptc(k,j).eq.'C99')xpt(k)=c99(j1)
      if(iptc(k,j).eq.'CALATM')xpt(k)=calatm
      if(iptc(k,j).eq.'CALRA')xpt(k)=calra
      if(iptc(k,j).eq.'CALRI')xpt(k)=calri
      if(iptc(k,j).eq.'CALSUB')xpt(k)=calsub
      if(iptc(k,j).eq.'CAR')xpt(k)=car(j1)
c ** remove from glossary
c     if(iptc(k,j).eq.'CATPRT')xpt(k)=catprt(j1)
      if(iptc(k,j).eq.'CBBNZ')xpt(k)=cbbnz(j1)
      if(iptc(k,j).eq.'CBKDZ')xpt(k)=cbkdz(j1)
      if(iptc(k,j).eq.'CBLVZ')xpt(k)=cblvz(j1)
      if(iptc(k,j).eq.'CBWBZ')xpt(k)=cbwbz(j1)
      if(iptc(k,j).eq.'CCUT')xpt(k)=ccut(j1,j2,j3)
      if(iptc(k,j).eq.'CCVV')xpt(k)=ccvv(j1,j2)
      if(iptc(k,j).eq.'CDRP')xpt(k)=cdrp(j1,j2)
      if(iptc(k,j).eq.'CELCT')xpt(k)=celct(j1,j2)
      if(iptc(k,j).eq.'CELKI')xpt(k)=celki(j1,j2)
      if(iptc(k,j).eq.'CELLS')xpt(k)=cells(j1,j2,j3)
      if(iptc(k,j).eq.'CFAC')xpt(k)=cfac(j1,j2)
      if(iptc(k,j).eq.'CFSPD')xpt(k)=cfspd(j1)
      if(iptc(k,j).eq.'CH20B')xpt(k)=ch20b(j1,j2)
      if(iptc(k,j).eq.'CHLAT')xpt(k)=chlat(j1,j2)
      if(iptc(k,j).eq.'CHNAQ')xpt(k)=chnaq(j1)
```

```
      if(iptc(k,j).eq.'CHNE')xpt(k)=chne(j1,j2)
      if(iptc(k,j).eq.'CHNNU')xpt(k)=chnnu
      if(iptc(k,j).eq.'CHO')xpt(k)=cho(j1,j2,j3)
      if(iptc(k,j).eq.'CLAY')xpt(k)=clay(j1,j2,j3)
      if(iptc(k,j).eq.'CLM')xpt(k)=clm(j1)
      if(iptc(k,j).eq.'CMAX')xpt(k)=cmax(j1,j2)
      if(iptc(k,j).eq.'CMORTA')xpt(k)=cmorta(j1,j2,j3,j4)
      if(iptc(k,j).eq.'CMORTB')xpt(k)=cmortb(j1,j2,j3,j4)
      if(iptc(k,j).eq.'CMORTY')xpt(k)=cmorty(j1,j2,j3,j4)
      if(iptc(k,j).eq.'CNATR')xpt(k)=cnatr(j1,j2,j3,j4)
      if(iptc(k,j).eq.'CNATY')xpt(k)=cnaty(j1,j2,j3,j4)
c *** new
      if(iptc(k,j).eq.'CNBLO')xpt(k)=cnblo(j1,j2,j3)
      if(iptc(k,j).eq.'CNBOD')xpt(k)=cnbod(j1,j2,j3)
      if(iptc(k,j).eq.'CNBON')xpt(k)=cnbon(j1,j2,j3)
      if(iptc(k,j).eq.'CNFAT')xpt(k)=cnfat(j1,j2,j3)
      if(iptc(k,j).eq.'CNFEC')xpt(k)=cnfec(j1,j2,j3)
      if(iptc(k,j).eq.'CNGUT')xpt(k)=cngut(j1,j2,j3)
      if(iptc(k,j).eq.'CNIT')xpt(k)=cnit(j1,j2)
      if(iptc(k,j).eq.'CNKID')xpt(k)=cnkid(j1,j2,j3)
      if(iptc(k,j).eq.'CNLIV')xpt(k)=cnliv(j1,j2,j3)
      if(iptc(k,j).eq.'CNLUG')xpt(k)=cnlug(j1,j2,j3)
      if(iptc(k,j).eq.'CNLUN')xpt(k)=cnlun(j1,j2,j3)
      if(iptc(k,j).eq.'CNMLK')xpt(k)=cnmlk(j1,j2,j3)
      if(iptc(k,j).eq.'CNMUS')xpt(k)=cnmus(j1,j2,j3)
      if(iptc(k,j).eq.'CNPEL')xpt(k)=cnpel(j1,j2,j3)
      if(iptc(k,j).eq.'CNPLN')xpt(k)=cnpln(j1,j2,j3)
      if(iptc(k,j).eq.'CNRES')xpt(k)=cnres(j1,j2,j3)
      if(iptc(k,j).eq.'CNSOL')xpt(k)=cnsol(j1,j2,j3)
      if(iptc(k,j).eq.'CNURN')xpt(k)=cnurn(j1,j2,j3)
      if(iptc(k,j).eq.'CNURNC')xpt(k)=cnurnc(j1,j2,j3)
c *** fin
      if(iptc(k,j).eq.'COLDAY')xpt(k)=colday(j1)
      if(iptc(k,j).eq.'COV')xpt(k)=cov(j1,j2)
      if(iptc(k,j).eq.'COVAV')xpt(k)=covav
      if(iptc(k,j).eq.'COVSQ')xpt(k)=covsq(j1,j2)
      if(iptc(k,j).eq.'CSOIL')xpt(k)=csoil(j1)
      if(iptc(k,j).eq.'CT')xpt(k)=ct(j1,j2)
      if(iptc(k,j).eq.'CTG')xpt(k)=ctg(j1,j2)
      if(iptc(k,j).eq.'CUTA')xpt(k)=cuta(j1,j2,j3)
      if(iptc(k,j).eq.'CUTMAX')xpt(k)=cutmax(j1,j2)
      if(iptc(k,j).eq.'CVMN')xpt(k)=cvmn
      if(iptc(k,j).eq.'CVSD')xpt(k)=cvsd
      if(iptc(k,j).eq.'CVTR')xpt(k)=cvtr
      if(iptc(k,j).eq.'CVTT')xpt(k)=cvtt(j1)
      if(iptc(k,j).eq.'CXFAC')xpt(k)=cxfac(j1,j2)
      if(iptc(k,j).eq.'CZ')xpt(k)=cz(j1,j2,j3)
      goto543
    4 if(iptc(k,j).eq.'D')xpt(k)=d(j1,j2)
      if(iptc(k,j).eq.'D1')xpt(k)=d1(j1,j2)
      if(iptc(k,j).eq.'D1GB')xpt(k)=d1gb(j1,j2)
      if(iptc(k,j).eq.'D1GBB')xpt(k)=d1gbb(j1,j2)
      if(iptc(k,j).eq.'D3')xpt(k)=d3(j1,j2)
      if(iptc(k,j).eq.'DAIC')xpt(k)=daic(j1,j2,j3)
```

```
if(iptc(k,j).eq.'DATCT')xpt(k)=datct(j1,j2,j3)
if(iptc(k,j).eq.'DAYFRC')xpt(k)=dayfrc
if(iptc(k,j).eq.'DBURD')xpt(k)=dburd(j1)
if(iptc(k,j).eq.'DCHAN')xpt(k)=dchan(j1)
if(iptc(k,j).eq.'DECLIN')xpt(k)=declin
if(iptc(k,j).eq.'DEGDAY')xpt(k)=degday(j1)
if(iptc(k,j).eq.'DEGG')xpt(k)=degg(j1,j2)
if(iptc(k,j).eq.'DEGGS')xpt(k)=deggs
if(iptc(k,j).eq.'DENDIF')xpt(k)=dendif(j1,j2)
if(iptc(k,j).eq.'DENSE')xpt(k)=dense(j1,j2)
if(iptc(k,j).eq.'DEPTHI')xpt(k)=depthi(j1)
if(iptc(k,j).eq.'DIADAY')xpt(k)=diaday(j1)
if(iptc(k,j).eq.'DIGE')xpt(k)=dige(j1)
if(iptc(k,j).eq.'DIRA')xpt(k)=dira(j1,j2,j3)
if(iptc(k,j).eq.'DIRMAX')xpt(k)=dirmax(j1,j2)
if(iptc(k,j).eq.'DIRT')xpt(k)=dirt
if(iptc(k,j).eq.'DISINT')xpt(k)=disint(j1)
if(iptc(k,j).eq.'DIT')xpt(k)=dit(j1)
if(iptc(k,j).eq.'DIVD')xpt(k)=divd(j1,j2)
if(iptc(k,j).eq.'DNN')xpt(k)=dnn(j1)
if(iptc(k,j).eq.'DOSE')xpt(k)=dose(j1)
if(iptc(k,j).eq.'DRMM')xpt(k)=drmm(j1)
if(iptc(k,j).eq.'DRNKK')xpt(k)=drnkk
if(iptc(k,j).eq.'DSF')xpt(k)=dsf(j1,j2)
if(iptc(k,j).eq.'DSM')xpt(k)=dsm(j1,j2)
if(iptc(k,j).eq.'DSMA')xpt(k)=dsma(j1,j2)
if(iptc(k,j).eq.'DSMS')xpt(k)=dsms(j1,j2)
if(iptc(k,j).eq.'DSMT')xpt(k)=dsmt(j1)
if(iptc(k,j).eq.'DSRT')xpt(k)=dsrt(j1,j2)
if(iptc(k,j).eq.'DSZ')xpt(k)=dsz(j1,j2)
if(iptc(k,j).eq.'DTBR')xpt(k)=dtbr(j1,j2,j3)
if(iptc(k,j).eq.'DUHR')xpt(k)=duhr(j1)
if(iptc(k,j).eq.'DV')xpt(k)=dv(j1,j2,j3)
if(iptc(k,j).eq.'DVTMP')xpt(k)=dvtmp(j1)
if(iptc(k,j).eq.'DVZ')xpt(k)=dvz(j1,j2,j3)
if(iptc(k,j).eq.'DZAS')xpt(k)=dzas(j1,j2)
goto543
5 if(iptc(k,j).eq.'E2')xpt(k)=e2(j1)
if(iptc(k,j).eq.'E5')xpt(k)=e5(j1,j2)
if(iptc(k,j).eq.'EA')xpt(k)=ea(j1,j2)
if(iptc(k,j).eq.'EAT')xpt(k)=eat(j1,j2,j3,j4)
if(iptc(k,j).eq.'EATIN')xpt(k)=eatin(j1)
if(iptc(k,j).eq.'EATREC')xpt(k)=eatrec(j1,j2)
if(iptc(k,j).eq.'EAVAL')xpt(k)=eaval(j1,j2,j3)
if(iptc(k,j).eq.'EFFP')xpt(k)=effp(j1)
if(iptc(k,j).eq.'EGEOP')xpt(k)=egeop(j1)
if(iptc(k,j).eq.'EGG')xpt(k)=egg(j1,j2)
if(iptc(k,j).eq.'EGGMRT')xpt(k)=eggmrt(j1)
if(iptc(k,j).eq.'EGGS')xpt(k)=eggs
if(iptc(k,j).eq.'EGLAY')xpt(k)=eglay
if(iptc(k,j).eq.'ENTCHN')xpt(k)=entchn(j1)
if(iptc(k,j).eq.'EPLD')xpt(k)=epld(j1)
if(iptc(k,j).eq.'ETRANS')xpt(k)=etrans(j1,j2)
if(iptc(k,j).eq.'EVAP')xpt(k)=evap(j1,j2)
```

```
 if(iptc(k,j).eq.'EVB')xpt(k)=evb(j1,j2)
 if(iptc(k,j).eq.'EVE')xpt(k)=eve(j1,j2)
 if(iptc(k,j).eq.'EVS')xpt(k)=evs(j1,j2)
 if(iptc(k,j).eq.'EVSP')xpt(k)=evsp(j1)
 if(iptc(k,j).eq.'EVTEMP')xpt(k)=evtemp
 if(iptc(k,j).eq.'EXCP')xpt(k)=excp(j1)
 if(iptc(k,j).eq.'EXIP')xpt(k)=exip(j1,j2)
 if(iptc(k,j).eq.'EXTC')xpt(k)=extc(j1)
 if(iptc(k,j).eq.'EXTD')xpt(k)=extd
 goto543
6 if(iptc(k,j).eq.'F1')xpt(k)=f1(j1,j2,j3)
 if(iptc(k,j).eq.'F7')xpt(k)=f7(j1)
 if(iptc(k,j).eq.'FATB')xpt(k)=fatb(j1,j2)
 if(iptc(k,j).eq.'FDIN')xpt(k)=fdin(j1,j2)
 if(iptc(k,j).eq.'FDTMP')xpt(k)=fdtmp(j1)
 if(iptc(k,j).eq.'FFP')xpt(k)=ffp(j1)
 if(iptc(k,j).eq.'FLEAF')xpt(k)=fleaf(j1,j2)
 if(iptc(k,j).eq.'FLXDN')xpt(k)=flxdn(j1,j2)
 if(iptc(k,j).eq.'FLXUP')xpt(k)=flxup(j1,j2)
 if(iptc(k,j).eq.'FMK')xpt(k)=fmk(j1)
 if(iptc(k,j).eq.'FODIN')xpt(k)=fodin(j1)
 if(iptc(k,j).eq.'FRM')xpt(k)=frm(j1)
 if(iptc(k,j).eq.'FRT')xpt(k)=frt(j1)
 if(iptc(k,j).eq.'FSUPF')xpt(k)=fsupf(j1)
 goto543
7 if(iptc(k,j).eq.'G')xpt(k)=g(j1,j2)
 if(iptc(k,j).eq.'G1')xpt(k)=g1(j1)
 if(iptc(k,j).eq.'G4')xpt(k)=g4(j1)
 if(iptc(k,j).eq.'G5')xpt(k)=g5(j1)
 if(iptc(k,j).eq.'GA')xpt(k)=ga(j1,j2)
 if(iptc(k,j).eq.'GASLOS')xpt(k)=gaslos(j1)
 if(iptc(k,j).eq.'GBURD')xpt(k)=gburd(j1)
 if(iptc(k,j).eq.'GBURI')xpt(k)=gburi(j1,j2,j3)
 if(iptc(k,j).eq.'GM2B')xpt(k)=gm2b(j1,j2)
 if(iptc(k,j).eq.'GM2R')xpt(k)=gm2r(j1,j2)
 if(iptc(k,j).eq.'GM2RB')xpt(k)=gm2rb(j1,j2)
 if(iptc(k,j).eq.'GM2T')xpt(k)=gm2t(j1,j2)
 if(iptc(k,j).eq.'GMEAT')xpt(k)=gmeat
 if(iptc(k,j).eq.'GMILK')xpt(k)=gmilk
 if(iptc(k,j).eq.'GRNA')xpt(k)=grna(j1)
 if(iptc(k,j).eq.'GRNAS')xpt(k)=grnas(j1)
 if(iptc(k,j).eq.'GSBD')xpt(k)=gsbd(j1,j2)
 if(iptc(k,j).eq.'GSBDTT')xpt(k)=gsbdtt(j1)
 if(iptc(k,j).eq.'GTBL')xpt(k)=gtbl(j1,j2)
 if(iptc(k,j).eq.'GTDR')xpt(k)=gtdr(j1,j2)
 goto543
8 if(iptc(k,j).eq.'H')xpt(k)=h(j1,j2,j3)
 if(iptc(k,j).eq.'H2')xpt(k)=h2
 if(iptc(k,j).eq.'H3')xpt(k)=h3
 if(iptc(k,j).eq.'HASUM')xpt(k)=hasum(j1,j2)
 if(iptc(k,j).eq.'HATCE')xpt(k)=hatce(j1,j2)
 if(iptc(k,j).eq.'HATCH')xpt(k)=hatch(j1,j2)
 if(iptc(k,j).eq.'HATCHS')xpt(k)=hatchs
 if(iptc(k,j).eq.'HATMP')xpt(k)=hatmp(j1,j2)
```

```
      if(iptc(k,j).eq.'HATTMP')xpt(k)=hattmp(j1,j2)
      if(iptc(k,j).eq.'HBAR')xpt(k)=hbar(j1,j2)
      if(iptc(k,j).eq.'HBT')xpt(k)=hbt(j1)
      if(iptc(k,j).eq.'HEMLOS')xpt(k)=hemlos(j1,j2,j3)
      if(iptc(k,j).eq.'HF')xpt(k)=hf(j1,j2)
      if(iptc(k,j).eq.'HFLIF')xpt(k)=hflif(j1)
      if(iptc(k,j).eq.'HGT')xpt(k)=hgt(j1,j2)
      if(iptc(k,j).eq.'HGTSQ')xpt(k)=hgtsq(j1,j2)
      if(iptc(k,j).eq.'HL4')xpt(k)=hl4(j1)
      if(iptc(k,j).eq.'HL5')xpt(k)=hl5(j1)
      if(iptc(k,j).eq.'HM8')xpt(k)=hm8
      if(iptc(k,j).eq.'HRVST')xpt(k)=hrvst(j1,j2,j3)
      if(iptc(k,j).eq.'HTAV')xpt(k)=htav(j1,j2)
      if(iptc(k,j).eq.'HTP')xpt(k)=htp(j1)
      if(iptc(k,j).eq.'HTSD')xpt(k)=htsd
      if(iptc(k,j).eq.'HTX')xpt(k)=htx(j1)
      if(iptc(k,j).eq.'HUMUS')xpt(k)=humus(j1,j2,j3)
      goto543
    9 if(iptc(k,j).eq.'ICA')xpt(k)=ica(j1,j2)
      if(iptc(k,j).eq.'IEGD')xpt(k)=iegd(j1)
      if(iptc(k,j).eq.'IEGG')xpt(k)=iegg(j1)
      if(iptc(k,j).eq.'IEGH')xpt(k)=iegh(j1)
      if(iptc(k,j).eq.'INZ')xpt(k)=inz
      if(iptc(k,j).eq.'IPL')xpt(k)=ipl
      if(iptc(k,j).eq.'IPX')xpt(k)=ipx
      if(iptc(k,j).eq.'IRG')xpt(k)=irg(j1,j2)
      if(iptc(k,j).eq.'IU')xpt(k)=iu
      goto543
   10 goto543
   11 if(iptc(k,j).eq.'KDI')xpt(k)=kdi
      if(iptc(k,j).eq.'KDTBL1')xpt(k)=kdtbl1(j1,j2)
      if(iptc(k,j).eq.'KDTBL2')xpt(k)=kdtbl2(j1,j2)
      goto543
   12 if(iptc(k,j).eq.'L1')xpt(k)=l1(j1)
      if(iptc(k,j).eq.'L2')xpt(k)=l2(j1)
      if(iptc(k,j).eq.'L3')xpt(k)=l3(j1)
      if(iptc(k,j).eq.'LVTBL')xpt(k)=lvtbl(j1,j2)
      goto543
   13 if(iptc(k,j).eq.'MMX')xpt(k)=mmx(j1)
      if(iptc(k,j).eq.'MSTBL')xpt(k)=mstbl(j1,j2)
      goto543
   14 if(iptc(k,j).eq.'NAN')xpt(k)=nan
      if(iptc(k,j).eq.'NDAY')xpt(k)=nday(j1)
      if(iptc(k,j).eq.'NHM')xpt(k)=nhm
      if(iptc(k,j).eq.'NPPTS')xpt(k)=nppts
      if(iptc(k,j).eq.'NU')xpt(k)=nu
      if(iptc(k,j).eq.'NX')xpt(k)=nx(j1,j2)
      goto543
   15 if(iptc(k,j).eq.'ONDV')xpt(k)=ondv(j1,j2)
      if(iptc(k,j).eq.'ONFOD')xpt(k)=onfod(j1,j2)
      if(iptc(k,j).eq.'ONYPH')xpt(k)=onyph(j1,j2)
      if(iptc(k,j).eq.'ONYPHS')xpt(k)=onyphs
      if(iptc(k,j).eq.'ORGBRD')xpt(k)=orgbrd(j1,j2,j3)
      if(iptc(k,j).eq.'ORGDOS')xpt(k)=orgdos(j1,j2,j3)
```

```
      if(iptc(k,j).eq.'ORGRAD')xpt(k)=orgrad(j1,j2,j3)
      if(iptc(k,j).eq.'OTOT')xpt(k)=otot(j1)
      if(iptc(k,j).eq.'OTOTA')xpt(k)=otota(j1)
      if(iptc(k,j).eq.'OTOTB')xpt(k)=ototb(j1)
      if(iptc(k,j).eq.'OTOTBS')xpt(k)=ototbs(j1)
      if(iptc(k,j).eq.'OTOTS')xpt(k)=otots(j1)
      if(iptc(k,j).eq.'OTTSM')xpt(k)=ottsm(j1)
      goto543
   16 if(iptc(k,j).eq.'P')xpt(k)=p(j1)
      if(iptc(k,j).eq.'P1')xpt(k)=p1(j1,j2)
      if(iptc(k,j).eq.'P5')xpt(k)=p5(j1)
      if(iptc(k,j).eq.'PAV')xpt(k)=pav(j1)
      if(iptc(k,j).eq.'PBURD')xpt(k)=pburd(j1)
      if(iptc(k,j).eq.'PBURI')xpt(k)=pburi(j1,j2,j3)
      if(iptc(k,j).eq.'PDIAPE')xpt(k)=pdiape(j1,j2)
      if(iptc(k,j).eq.'PDIAPS')xpt(k)=pdiaps
      if(iptc(k,j).eq.'PGR')xpt(k)=pgr(j1)
      if(iptc(k,j).eq.'PH')xpt(k)=ph(j1,j2,j3)
      if(iptc(k,j).eq.'PHI')xpt(k)=phi(j1,j2,j3)
      if(iptc(k,j).eq.'PHITT')xpt(k)=phitt(j1)
      if(iptc(k,j).eq.'PHOSP')xpt(k)=phosp(j1,j2)
      if(iptc(k,j).eq.'PHOTPR')xpt(k)=photpr
      if(iptc(k,j).eq.'PHYTO')xpt(k)=phyto(j1,j2,j3)
      if(iptc(k,j).eq.'PLCM1')xpt(k)=plcm1(j1)
      if(iptc(k,j).eq.'PLCM2')xpt(k)=plcm2(j1)
      if(iptc(k,j).eq.'PLCM3')xpt(k)=plcm3(j1)
      if(iptc(k,j).eq.'PLTBL')xpt(k)=pltbl(j1,j2)
      if(iptc(k,j).eq.'PLTEA')xpt(k)=pltea(j1,j2)
      if(iptc(k,j).eq.'PLTEAY')xpt(k)=plteay(j1,j2)
      if(iptc(k,j).eq.'PLTF')xpt(k)=pltf(j1,j2,j3,j4)
      if(iptc(k,j).eq.'PLTG1')xpt(k)=pltg1(j1,j2)
      if(iptc(k,j).eq.'PLTG2')xpt(k)=pltg2(j1,j2)
      if(iptc(k,j).eq.'PLTID')xpt(k)=pltid(j1,j2)
      if(iptc(k,j).eq.'PLTIND')xpt(k)=pltind(j1,j2,j3)
      if(iptc(k,j).eq.'PLTRNK')xpt(k)=pltrnk(j1,j2,j3,j4)
      if(iptc(k,j).eq.'PLTUSD')xpt(k)=pltusd(j1,j2,j3,j4)
      if(iptc(k,j).eq.'PM')xpt(k)=pm(j1,j2)
      if(iptc(k,j).eq.'PMS')xpt(k)=pms(j1,j2)
      if(iptc(k,j).eq.'PN')xpt(k)=pn(j1)
      if(iptc(k,j).eq.'POP')xpt(k)=pop(j1,j2)
      if(iptc(k,j).eq.'POPFR')xpt(k)=popfr(j1)
      if(iptc(k,j).eq.'POPSUM')xpt(k)=popsum(j1,j2)
      if(iptc(k,j).eq.'PRCH')xpt(k)=prch(j1)
      if(iptc(k,j).eq.'PRCV')xpt(k)=prcv(j1)
      if(iptc(k,j).eq.'PREC')xpt(k)=prec(j1)
      if(iptc(k,j).eq.'PROT1')xpt(k)=prot1(j1,j2)
      if(iptc(k,j).eq.'PROTB')xpt(k)=protb(j1,j2)
      if(iptc(k,j).eq.'PSEP')xpt(k)=psep(j1,j2)
      if(iptc(k,j).eq.'PSOIL')xpt(k)=psoil(j1)
      if(iptc(k,j).eq.'PUTFRC')xpt(k)=putfrc(j1,j2,j3)
      if(iptc(k,j).eq.'PUTZD')xpt(k)=putzd(j1,j2,j3)
      if(iptc(k,j).eq.'PX')xpt(k)=px(j1,j2)
      goto543
   17 if(iptc(k,j).eq.'Q')xpt(k)=q(j1,j2)
```

```
      if(iptc(k,j).eq.'Q1')xpt(k)=q1(j1,j2,j3)
      if(iptc(k,j).eq.'Q10')xpt(k)=q10(j1,j2)
      if(iptc(k,j).eq.'Q2')xpt(k)=q2(j1,j2,j3)
      if(iptc(k,j).eq.'Q3')xpt(k)=q3(j1,j2,j3)
      if(iptc(k,j).eq.'Q8')xpt(k)=q8(j1)
      if(iptc(k,j).eq.'QFRAC')xpt(k)=qfrac(j1,j2)
      if(iptc(k,j).eq.'QMNT')xpt(k)=qmnt(j1,j2)
      if(iptc(k,j).eq.'QMPH')xpt(k)=qmph(j1,j2)
      if(iptc(k,j).eq.'QMSI')xpt(k)=qmsi(j1,j2)
      if(iptc(k,j).eq.'QNITRO')xpt(k)=qnitro(j1,j2,j3)
      if(iptc(k,j).eq.'QPHOSP')xpt(k)=qphosp(j1,j2,j3)
      if(iptc(k,j).eq.'QPKD')xpt(k)=qpkd(j1)
      if(iptc(k,j).eq.'QSILCN')xpt(k)=qsilcn(j1,j2,j3)
      if(iptc(k,j).eq.'QSUM')xpt(k)=qsum(j1,j2)
      goto543
   18 if(iptc(k,j).eq.'R')xpt(k)=r(j1)
      if(iptc(k,j).eq.'R1')xpt(k)=r1(j1,j2)
      if(iptc(k,j).eq.'R1MAX')xpt(k)=r1max(j1,j2)
      if(iptc(k,j).eq.'RACUT')xpt(k)=racut(j1,j2)
      if(iptc(k,j).eq.'RADA')xpt(k)=rada(j1)
      if(iptc(k,j).eq.'RADANG')xpt(k)=radang
      if(iptc(k,j).eq.'RANK')xpt(k)=rank(j1,j2)
      if(iptc(k,j).eq.'RAREA')xpt(k)=rarea(j1)
      if(iptc(k,j).eq.'RB3')xpt(k)=rb3(j1,j2,j3)
      if(iptc(k,j).eq.'RB3C')xpt(k)=rb3c(j1,j2,j3)
      if(iptc(k,j).eq.'RBB3')xpt(k)=rbb3(j1,j2,j3)
      if(iptc(k,j).eq.'RBB3C')xpt(k)=rbb3c(j1,j2,j3)
      if(iptc(k,j).eq.'RBCUT')xpt(k)=rbcut(j1,j2)
c
      if(iptc(k,j).eq.'RBN')xpt(k)=rbn(j1,j2,j3)
      if(iptc(k,j).eq.'RKD')xpt(k)=rkd(j1,j2,j3)
      if(iptc(k,j).eq.'RLU')xpt(k)=rlu(j1,j2,j3)
      if(iptc(k,j).eq.'RLV')xpt(k)=rlv(j1,j2,j3)
      if(iptc(k,j).eq.'RMS')xpt(k)=rms(j1,j2,j3)
      if(iptc(k,j).eq.'RNDO')xpt(k)=rndo(j1,j2,j3)
c
      if(iptc(k,j).eq.'RCCUT')xpt(k)=rccut(j1,j2)
      if(iptc(k,j).eq.'RCHAN')xpt(k)=rchan(j1)
      if(iptc(k,j).eq.'RCODE')xpt(k)=rcode(j1,j2)
      if(iptc(k,j).eq.'RD1')xpt(k)=rd1(j1,j2)
      if(iptc(k,j).eq.'RD2')xpt(k)=rd2(j1,j2)
      if(iptc(k,j).eq.'RD3')xpt(k)=rd3(j1,j2)
      if(iptc(k,j).eq.'RDCF')xpt(k)=rdcf(j1)
      if(iptc(k,j).eq.'REFRAC')xpt(k)=refrac
      if(iptc(k,j).eq.'REGENT')xpt(k)=regent(j1)
      if(iptc(k,j).eq.'REGX')xpt(k)=regx(j1,j2)
      if(iptc(k,j).eq.'RELCAN')xpt(k)=relcan(j1,j2)
      if(iptc(k,j).eq.'RELV')xpt(k)=relv(j1)
      if(iptc(k,j).eq.'RESP')xpt(k)=resp(j1,j2)
      if(iptc(k,j).eq.'RESPF')xpt(k)=respf(j1,j2)
      if(iptc(k,j).eq.'RFDCF')xpt(k)=rfdcf(j1)
      if(iptc(k,j).eq.'RFDLX')xpt(k)=rfdlx(j1)
      if(iptc(k,j).eq.'RFLCT')xpt(k)=rflct
      if(iptc(k,j).eq.'RFP')xpt(k)=rfp(j1,j2)
```

```
      if(iptc(k,j).eq.'RFRAC')xpt(k)=rfrac(j1)
      if(iptc(k,j).eq.'RFRZ')xpt(k)=rfrz(j1,j2)
      if(iptc(k,j).eq.'RFYCF')xpt(k)=rfycf(j1)
      if(iptc(k,j).eq.'RFYLX')xpt(k)=rfylx(j1)
      if(iptc(k,j).eq.'RGAIN')xpt(k)=rgain(j1,j2)
      if(iptc(k,j).eq.'RGC')xpt(k)=rgc(j1,j2)
      if(iptc(k,j).eq.'RH1')xpt(k)=rh1(j1,j2,j3)
      if(iptc(k,j).eq.'RH2')xpt(k)=rh2(j1,j2,j3)
      if(iptc(k,j).eq.'RINGT')xpt(k)=ringt(j1,j2)
      if(iptc(k,j).eq.'RIRRC')xpt(k)=rirrc(j1,j2)
      if(iptc(k,j).eq.'RNCHN')xpt(k)=rnchn(j1)
      if(iptc(k,j).eq.'RNDLX')xpt(k)=rndlx(j1)
      if(iptc(k,j).eq.'RNFRC')xpt(k)=rnfrc(j1,j2,j3)
      if(iptc(k,j).eq.'RNODX')xpt(k)=rnodx(j1,j2)
      if(iptc(k,j).eq.'RNOFDG')xpt(k)=rnofdg(j1,j2)
      if(iptc(k,j).eq.'RNOFDM')xpt(k)=rnofdm(j1,j2,j3)
      if(iptc(k,j).eq.'RNOFDS')xpt(k)=rnofds(j1,j2)
      if(iptc(k,j).eq.'RNOFYG')xpt(k)=rnofyg(j1,j2)
      if(iptc(k,j).eq.'RNOFYM')xpt(k)=rnofym(j1,j2,j3)
      if(iptc(k,j).eq.'RNOFYS')xpt(k)=rnofys(j1,j2)
      if(iptc(k,j).eq.'RNYLX')xpt(k)=rnylx(j1)
      if(iptc(k,j).eq.'ROCK')xpt(k)=rock(j1,j2,j3)
      if(iptc(k,j).eq.'RPNTP')xpt(k)=rpntp(j1,j2)
      if(iptc(k,j).eq.'RPT')j1=ai1
      if(iptc(k,j).eq.'RPT')xpt(k)=rpt(j1,j2,j3,j4,j5)
      if(iptc(k,j).eq.'RR')xpt(k)=rr(j1)
      if(iptc(k,j).eq.'RSCDE')xpt(k)=rscde(j1)
      if(iptc(k,j).eq.'RSPF2')xpt(k)=rspf2(j1,j2)
      if(iptc(k,j).eq.'RSRT')xpt(k)=rsrt(j1,j2)
      if(iptc(k,j).eq.'RSS')xpt(k)=rss(j1,j2)
      if(iptc(k,j).eq.'RSTBL')xpt(k)=rstbl(j1,j2)
      if(iptc(k,j).eq.'RSUSP')xpt(k)=rsusp(j1,j2,j3)
      if(iptc(k,j).eq.'RTRANS')xpt(k)=rtrans(j1,j2)
      if(iptc(k,j).eq.'RXPP')xpt(k)=rxpp(j1,j2)
      if(iptc(k,j).eq.'RYCF')xpt(k)=rycf(j1)
      if(iptc(k,j).eq.'RZOON')xpt(k)=rzoon(j1)
      goto543
   19 if(iptc(k,j).eq.'SA')xpt(k)=sa
      if(iptc(k,j).eq.'SACT')xpt(k)=sact(j1,j2,j3)
      if(iptc(k,j).eq.'SAIR')xpt(k)=sair(j1,j2,j3)
      if(iptc(k,j).eq.'SALCRP')xpt(k)=salcrp(j1,j2,j3)
      if(iptc(k,j).eq.'SAND')xpt(k)=sand(j1,j2,j3)
      if(iptc(k,j).eq.'SAREA')xpt(k)=sarea(j1)
      if(iptc(k,j).eq.'SATB')xpt(k)=satb(j1,j2,j3,j4)
      if(iptc(k,j).eq.'SATRS')xpt(k)=satrs(j1,j2)
      if(iptc(k,j).eq.'SCHAN')xpt(k)=schan(j1)
      if(iptc(k,j).eq.'SCTP')xpt(k)=sctp(j1,j2)
      if(iptc(k,j).eq.'SCVR')xpt(k)=scvr(j1,j2)
      if(iptc(k,j).eq.'SD')xpt(k)=sd(j1,j2)
      if(iptc(k,j).eq.'SD1')xpt(k)=sd1(j1,j2)
      if(iptc(k,j).eq.'SD3')xpt(k)=sd3(j1,j2)
      if(iptc(k,j).eq.'SDMNCT')xpt(k)=sdmnct(j1)
      if(iptc(k,j).eq.'SDMNSA')xpt(k)=sdmnsa(j1)
      if(iptc(k,j).eq.'SDMX')xpt(k)=sdmx(j1,j2)
```

```
      if(iptc(k,j).eq.'SDPTH')xpt(k)=sdpth(j1,j2)
      if(iptc(k,j).eq.'SEDGAT')xpt(k)=sedgat
      if(iptc(k,j).eq.'SEDMNS')xpt(k)=sedmns(j1)
      if(iptc(k,j).eq.'SFAC')xpt(k)=sfac(j1,j2)
      if(iptc(k,j).eq.'SFFP')xpt(k)=sffp(j1)
      if(iptc(k,j).eq.'SG')xpt(k)=sg(j1,j2,j3)
      if(iptc(k,j).eq.'SILCN')xpt(k)=silcn(j1,j2)
      if(iptc(k,j).eq.'SILT')xpt(k)=silt(j1,j2,j3)
      if(iptc(k,j).eq.'SINK')xpt(k)=sink(j1,j2)
      if(iptc(k,j).eq.'SIR')xpt(k)=sir(j1,j2)
      if(iptc(k,j).eq.'SLICE')xpt(k)=slice(j1,j2,j3)
      if(iptc(k,j).eq.'SLOPX')xpt(k)=slopx(j1)
      if(iptc(k,j).eq.'SLOSS')xpt(k)=sloss(j1,j2)
      if(iptc(k,j).eq.'SLUNG')xpt(k)=slung(j1)
      if(iptc(k,j).eq.'SNFRC')xpt(k)=snfrc(j1)
      if(iptc(k,j).eq.'SNIT')xpt(k)=snit(j1,j2)
      if(iptc(k,j).eq.'SNM')xpt(k)=snm(j1)
      if(iptc(k,j).eq.'SOILF')xpt(k)=soilf(j1,j2)
      if(iptc(k,j).eq.'SOLRAD')xpt(k)=solrad
      if(iptc(k,j).eq.'SORGB')xpt(k)=sorgb(j1,j2,j3)
      if(iptc(k,j).eq.'SORGD')xpt(k)=sorgd(j1,j2,j3)
      if(iptc(k,j).eq.'SOTOTZ')xpt(k)=sototz(j1)
      if(iptc(k,j).eq.'SPAA')xpt(k)=spaa(j1,j2,j3,j4)
      if(iptc(k,j).eq.'SPAC')xpt(k)=spac(j1,j2,j3)
      if(iptc(k,j).eq.'SPAR')xpt(k)=spar(j1,j2,j3)
      if(iptc(k,j).eq.'SPAS')xpt(k)=spas(j1,j2,j3,j4)
      if(iptc(k,j).eq.'SPNCAN')xpt(k)=spncan(j1,j2)
      if(iptc(k,j).eq.'SRUN')xpt(k)=srun(j1)
      if(iptc(k,j).eq.'STEM12')xpt(k)=stem12(j1,j2)
      if(iptc(k,j).eq.'STMTT')xpt(k)=stmtt
      if(iptc(k,j).eq.'STRANS')xpt(k)=strans(j1,j2)
      if(iptc(k,j).eq.'SUMPUT')xpt(k)=sumput(j1)
      if(iptc(k,j).eq.'SUNMN')xpt(k)=sunmn(j1)
      if(iptc(k,j).eq.'SUNMX')xpt(k)=sunmx(j1)
      if(iptc(k,j).eq.'SUSFC')xpt(k)=susfc(j1,j2)
      if(iptc(k,j).eq.'SYAV')xpt(k)=syav(j1,j2)
      goto543
   20 if(iptc(k,j).eq.'T')xpt(k)=t(j1)
      if(iptc(k,j).eq.'T1')xpt(k)=t1(j1)
      if(iptc(k,j).eq.'T2')xpt(k)=t2(j1)
      if(iptc(k,j).eq.'T3')xpt(k)=t3(j1)
      if(iptc(k,j).eq.'T4')xpt(k)=t4(j1)
      if(iptc(k,j).eq.'T5P')xpt(k)=t5p(j1,j2)
      if(iptc(k,j).eq.'TAV')xpt(k)=tav(j1)
      if(iptc(k,j).eq.'TBBNZ')xpt(k)=tbbnz(j1)
      if(iptc(k,j).eq.'TBCH')xpt(k)=tbch(j1)
      if(iptc(k,j).eq.'TBKDZ')xpt(k)=tbkdz(j1)
      if(iptc(k,j).eq.'TBLVZ')xpt(k)=tblvz(j1)
      if(iptc(k,j).eq.'TBRM')xpt(k)=tbrm(j1,j2)
      if(iptc(k,j).eq.'TBWBZ')xpt(k)=tbwbz(j1)
      if(iptc(k,j).eq.'TCH')xpt(k)=tch(j1)
      if(iptc(k,j).eq.'TCHAN')xpt(k)=tchan(j1)
      if(iptc(k,j).eq.'TCN')xpt(k)=tcn(j1,j2,j3,j4)
      if(iptc(k,j).eq.'TCNAT')xpt(k)=tcnat(j1,j2,j3)
```

```
      if(iptc(k,j).eq.'TCREL')xpt(k)=tcrel(j1,j2,j3,j4)
      if(iptc(k,j).eq.'TCUT')xpt(k)=tcut(j1,j2,j3)
      if(iptc(k,j).eq.'TCVR')xpt(k)=tcvr(j1)
      if(iptc(k,j).eq.'TF')xpt(k)=tf(j1,j2,j3)
      if(iptc(k,j).eq.'THR')xpt(k)=thr(j1,j2)
      if(iptc(k,j).eq.'THZ')xpt(k)=thz(j1)
      if(iptc(k,j).eq.'TIM')xpt(k)=tim(j1,j2)
      if(iptc(k,j).eq.'TLAI')xpt(k)=tlai(j1,j2)
      if(iptc(k,j).eq.'TLLV')xpt(k)=tllv(j1,j2)
      if(iptc(k,j).eq.'TM')xpt(k)=tm(j1,j2)
      if(iptc(k,j).eq.'TMN')xpt(k)=tmn(j1)
      if(iptc(k,j).eq.'TMPDAY')xpt(k)=tmpday(j1,j2)
      if(iptc(k,j).eq.'TMS')xpt(k)=tms(j1,j2)
      if(iptc(k,j).eq.'TMX')xpt(k)=tmx(j1)
      if(iptc(k,j).eq.'TMXV')xpt(k)=tmxv(j1)
      if(iptc(k,j).eq.'TN')xpt(k)=tn(j1)
      if(iptc(k,j).eq.'TNDLX')xpt(k)=tndlx(j1)
      if(iptc(k,j).eq.'TNDTN')xpt(k)=tndtn(j1)
      if(iptc(k,j).eq.'TNIT')xpt(k)=tnit(j1,j2)
      if(iptc(k,j).eq.'TNTR')xpt(k)=tntr(j1,j2)
      if(iptc(k,j).eq.'TNYLX')xpt(k)=tnylx(j1)
      if(iptc(k,j).eq.'TNYTN')xpt(k)=tnytn(j1)
      if(iptc(k,j).eq.'TOFH')xpt(k)=tofh(j1,j2)
      if(iptc(k,j).eq.'TOFL')xpt(k)=tofl(j1,j2)
      if(iptc(k,j).eq.'TOPM')xpt(k)=topm(j1,j2)
      if(iptc(k,j).eq.'TOPT')xpt(k)=topt(j1,j2)
      if(iptc(k,j).eq.'TOTAC')xpt(k)=totac(j1,j2)
      if(iptc(k,j).eq.'TOTSA')xpt(k)=totsa(j1,j2)
      if(iptc(k,j).eq.'TOTSF')xpt(k)=totsf(j1,j2,j3)
      if(iptc(k,j).eq.'TPGNW')xpt(k)=tpgnw(j1,j2)
      if(iptc(k,j).eq.'TPOP')xpt(k)=tpop(j1,j2,j3)
      if(iptc(k,j).eq.'TPSM')xpt(k)=tpsm(j1)
      if(iptc(k,j).eq.'TQX')xpt(k)=tqx(j1,j2,j3)
      if(iptc(k,j).eq.'TRE')xpt(k)=tre(j1,j2)
      if(iptc(k,j).eq.'TREES')xpt(k)=trees(j1,j2)
      if(iptc(k,j).eq.'TSRF')xpt(k)=tsrf
      if(iptc(k,j).eq.'TTZ')xpt(k)=ttz(j1)
      if(iptc(k,j).eq.'TULV')xpt(k)=tulv(j1,j2)
      if(iptc(k,j).eq.'TX')xpt(k)=tx(j1,j2)
      goto543
   21 if(iptc(k,j).eq.'UGRN')xpt(k)=ugrn
      if(iptc(k,j).eq.'ULEF')xpt(k)=ulef(j1,j2)
      if(iptc(k,j).eq.'ULV')xpt(k)=ulv
      if(iptc(k,j).eq.'UMNT')xpt(k)=umnt(j1,j2)
      if(iptc(k,j).eq.'UMPH')xpt(k)=umph(j1,j2)
      if(iptc(k,j).eq.'UMSI')xpt(k)=umsi(j1,j2)
      if(iptc(k,j).eq.'UP')xpt(k)=up(j1,j2)
      if(iptc(k,j).eq.'UPK')xpt(k)=upk(j1)
      if(iptc(k,j).eq.'URNLOS')xpt(k)=urnlos(j1)
      if(iptc(k,j).eq.'USV')xpt(k)=usv
      goto543
   22 if(iptc(k,j).eq.'V')xpt(k)=v(j1,j2,j3)
      if(iptc(k,j).eq.'V4')xpt(k)=v4
      if(iptc(k,j).eq.'V8')xpt(k)=v8(j1,j2)
```

```
      if(iptc(k,j).eq.'VA')xpt(k)=va(j1,j2,j3)
      if(iptc(k,j).eq.'VAQF')xpt(k)=vaqf(j1)
      if(iptc(k,j).eq.'VAQT')xpt(k)=vaqt(j1)
      if(iptc(k,j).eq.'VBN1')xpt(k)=vbn1(j1,j2,j3)
      if(iptc(k,j).eq.'VBN2')xpt(k)=vbn2(j1,j2,j3)
      if(iptc(k,j).eq.'VCHAN')xpt(k)=vchan(j1)
      if(iptc(k,j).eq.'VCUT')xpt(k)=vcut(j1,j2)
      if(iptc(k,j).eq.'VFAT')xpt(k)=vfat(j1,j2,j3)
      if(iptc(k,j).eq.'VFEC')xpt(k)=vfec(j1,j2,j3)
c ** new
      if(iptc(k,j).eq.'VGUT')xpt(k)=vgut(j1,j2,j3)
      if(iptc(k,j).eq.'VIIIN')xpt(k)=viiin(j1)
      if(iptc(k,j).eq.'VINTK')xpt(k)=vintk(j1,j2,j3)
      if(iptc(k,j).eq.'VKD1')xpt(k)=vkd1(j1,j2,j3)
      if(iptc(k,j).eq.'VKD2')xpt(k)=vkd2(j1,j2,j3)
      if(iptc(k,j).eq.'VLAT')xpt(k)=vlat(j1)
      if(iptc(k,j).eq.'VLOS')xpt(k)=vlos(j1,j2,j3)
      if(iptc(k,j).eq.'VLOSS')xpt(k)=vloss(j1)
c ** new
      if(iptc(k,j).eq.'VLUG')xpt(k)=vlug(j1,j2,j3)
      if(iptc(k,j).eq.'VLV')xpt(k)=vlv(j1,j2,j3)
      if(iptc(k,j).eq.'VMK')xpt(k)=vmk(j1,j2,j3)
      if(iptc(k,j).eq.'VMS')xpt(k)=vms(j1,j2,j3)
c ** new
      if(iptc(k,j).eq.'VPELT')xpt(k)=vpelt(j1,j2,j3)
      if(iptc(k,j).eq.'VPLN')xpt(k)=vpln(j1,j2,j3)
c ** fin
      if(iptc(k,j).eq.'VR')xpt(k)=vr(j1,j2,j3)
      if(iptc(k,j).eq.'VRFL')xpt(k)=vrfl(j1)
      if(iptc(k,j).eq.'VRT')xpt(k)=vrt(j1,j2,j3)
      if(iptc(k,j).eq.'VRT1')xpt(k)=vrt1(j1,j2,j3)
c ** new
      if(iptc(k,j).eq.'VSOL')xpt(k)=vsol(j1,j2,j3)
      if(iptc(k,j).eq.'VTMBR')xpt(k)=vtmbr(j1,j2,j3)
      if(iptc(k,j).eq.'VTRND')xpt(k)=vtrnd(j1)
      if(iptc(k,j).eq.'VTRNDP')xpt(k)=vtrndp(j1)
      if(iptc(k,j).eq.'VUPTK')xpt(k)=vuptk(j1,j2,j3)
      if(iptc(k,j).eq.'VURIN')xpt(k)=vurin(j1,j2,j3)
      if(iptc(k,j).eq.'VVTM')xpt(k)=vvtm(j1,j2,j3)
      goto543
   23 if(iptc(k,j).eq.'W')xpt(k)=w(j1,j2,j3)
c ** new ***
c extend glossary to include w(1-8), delete others
      if(iptc(k,j).eq.'W1')xpt(k)=w1(j1,j2,j3)
      if(iptc(k,j).eq.'W1BAR')xpt(k)=w1bar(j1,j2)
      if(iptc(k,j).eq.'WA')xpt(k)=wa(j1,j2,j3)
      if(iptc(k,j).eq.'WASUMR')xpt(k)=wasumr(j1,j2)
      if(iptc(k,j).eq.'WASUMS')xpt(k)=wasums(j1,j2)
      if(iptc(k,j).eq.'WATAKD')xpt(k)=watakd(j1,j2,j3)
      if(iptc(k,j).eq.'WATAKU')xpt(k)=wataku(j1,j2,j3)
      if(iptc(k,j).eq.'WATCN')xpt(k)=watcn(j1,j2)
      if(iptc(k,j).eq.'WATCNH')xpt(k)=watcnh(j1,j2)
      if(iptc(k,j).eq.'WATCNI')xpt(k)=watcni(j1,j2)
      if(iptc(k,j).eq.'WATCNR')xpt(k)=watcnr(j1,j2,j3)
```

```
      if(iptc(k,j).eq.'WATER')xpt(k)=water(j1,j2)
      if(iptc(k,j).eq.'WATFOD')xpt(k)=watfod
      if(iptc(k,j).eq.'WATMP')xpt(k)=watmp(j1,j2)
      if(iptc(k,j).eq.'WATUPA')xpt(k)=watupa(j1,j2,j3)
      if(iptc(k,j).eq.'WATUPB')xpt(k)=watupb(j1,j2,j3)
      if(iptc(k,j).eq.'WATUPT')xpt(k)=watupt
      if(iptc(k,j).eq.'WATVOL')xpt(k)=watvol
      if(iptc(k,j).eq.'WATX')xpt(k)=watx(j1)
      if(iptc(k,j).eq.'WBURD')xpt(k)=wburd(j1)
      if(iptc(k,j).eq.'WBURI')xpt(k)=wburi(j1,j2,j3)
      if(iptc(k,j).eq.'WCHAN')xpt(k)=wchan(j1)
      if(iptc(k,j).eq.'WD1')xpt(k)=wd1(j1,j2)
      if(iptc(k,j).eq.'WD3')xpt(k)=wd3(j1,j2)
      if(iptc(k,j).eq.'WFR')xpt(k)=wfr(j1,j2,j3)
      if(iptc(k,j).eq.'WMK')xpt(k)=wmk(j1)
      if(iptc(k,j).eq.'WMS')xpt(k)=wms(j1,j2)
      if(iptc(k,j).eq.'WND')xpt(k)=wnd(j1,j2)
      if(iptc(k,j).eq.'WNDMAX')xpt(k)=wndmax(j1,j2)
      if(iptc(k,j).eq.'WSRP')xpt(k)=wsrp(j1,j2)
      if(iptc(k,j).eq.'WT1')xpt(k)=wt1(j1,j2)
      if(iptc(k,j).eq.'WTBN')xpt(k)=wtbn(j1,j2)
      if(iptc(k,j).eq.'WT')xpt(k)=wt(j1,j2)
      if(iptc(k,j).eq.'WTBL')xpt(k)=wtbl(j1,j2)
      if(iptc(k,j).eq.'WTF')xpt(k)=wtf(j1)
      if(iptc(k,j).eq.'WTKD')xpt(k)=wtkd(j1,j2)
      if(iptc(k,j).eq.'WTLV')xpt(k)=wtlv(j1,j2)
      if(iptc(k,j).eq.'WTLU')xpt(k)=wtlu(j1,j2)
      if(iptc(k,j).eq.'WTMS')xpt(k)=wtms(j1,j2)
      if(iptc(k,j).eq.'WTRS')xpt(k)=wtrs(j1,j2)
      if(iptc(k,j).eq.'WTX')xpt(k)=wtx(j1)
      if(iptc(k,j).eq.'WURAT')xpt(k)=wurat(j1,j2)
      if(iptc(k,j).eq.'WUTP')xpt(k)=wutp(j1,j2)
      if(iptc(k,j).eq.'WYAV')xpt(k)=wyav(j1,j2)
      if(iptc(k,j).eq.'WZ3')xpt(k)=wz3(j1,j2,j3)
      if(iptc(k,j).eq.'WZ4')xpt(k)=wz4(j1,j2,j3)
      if(iptc(k,j).eq.'WZ42')xpt(k)=wz42(j1,j2,j3)
      if(iptc(k,j).eq.'WZ43')xpt(k)=wz43(j1,j2,j3)
      if(iptc(k,j).eq.'WZ4C')xpt(k)=wz4c(j1,j2,j3)
      if(iptc(k,j).eq.'WZ5')xpt(k)=wz5(j1,j2,j3)
      if(iptc(k,j).eq.'WZ5C')xpt(k)=wz5c(j1,j2,j3)
      if(iptc(k,j).eq.'WZ5X')xpt(k)=wz5x(j1,j2,j3)
      if(iptc(k,j).eq.'WZ7')xpt(k)=wz7(j1,j2,j3)
      if(iptc(k,j).eq.'WZ7MAX')xpt(k)=wz7max(j1,j2,j3)
      if(iptc(k,j).eq.'WZ8')xpt(k)=wz8(j1,j2,j3)
      if(iptc(k,j).eq.'WZZ4C')xpt(k)=wzz4c(j1,j2,j3)
      goto543
   24 if(iptc(k,j).eq.'X')xpt(k)=x(j1,j2)
      if(iptc(k,j).eq.'X8')xpt(k)=x8(j1,j2,j3)
      if(iptc(k,j).eq.'XCHAN')xpt(k)=xchan(j1)
      if(iptc(k,j).eq.'XFAC')xpt(k)=xfac(j1,j2)
      if(iptc(k,j).eq.'XJ1')xpt(k)=xj1(j1,j2)
      if(iptc(k,j).eq.'XX')xpt(k)=xx(j1,j2)
      goto543
   25 if(iptc(k,j).eq.'Y1')xpt(k)=y1(j1,j2)
```

```
      if(iptc(k,j).eq.'YADLT')xpt(k)=yadlt(j1,j2)
      if(iptc(k,j).eq.'YADLTS')xpt(k)=yadlts
      if(iptc(k,j).eq.'YAZ')xpt(k)=yaz(j1,j2)
      if(iptc(k,j).eq.'YM')xpt(k)=ym(j1,j2,j3)
      if(iptc(k,j).eq.'YNDV')xpt(k)=yndv(j1,j2)
      if(iptc(k,j).eq.'YNYPH')xpt(k)=ynyph(j1,j2)
      if(iptc(k,j).eq.'YNYPHS')xpt(k)=ynyphs
      if(iptc(k,j).eq.'YRFED')xpt(k)=yrfed(j1,j2)
      if(iptc(k,j).eq.'YSRT')xpt(k)=ysrt(j1,j2)
      if(iptc(k,j).eq.'YUP')xpt(k)=yup(j1,j2)
      if(iptc(k,j).eq.'YZF')xpt(k)=yzf(j1,j2)
      goto543
   26 if(iptc(k,j).eq.'Z')xpt(k)=z(j1,j2)
      if(iptc(k,j).eq.'Z1')xpt(k)=z1(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z1AT')xpt(k)=z1at(j1,j2)
      if(iptc(k,j).eq.'Z1AVE')xpt(k)=z1ave(j1,j2)
      if(iptc(k,j).eq.'Z1DL')xpt(k)=z1dl(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z2')xpt(k)=z2(j1,j2,j3)
      if(iptc(k,j).eq.'Z3')xpt(k)=z3(j1,j2,j3)
      if(iptc(k,j).eq.'Z4')xpt(k)=z4(j1,j2,j3)
      if(iptc(k,j).eq.'Z42')xpt(k)=z42(j1,j2,j3)
      if(iptc(k,j).eq.'Z43')xpt(k)=z43(j1,j2,j3)
      if(iptc(k,j).eq.'Z4C')xpt(k)=z4c(j1,j2,j3)
      if(iptc(k,j).eq.'Z4G')xpt(k)=z4g(j1,j2,j3)
      if(iptc(k,j).eq.'Z4G2')xpt(k)=z4g2(j1,j2,j3)
      if(iptc(k,j).eq.'Z4G3')xpt(k)=z4g3(j1,j2,j3)
      if(iptc(k,j).eq.'Z4GC')xpt(k)=z4gc(j1,j2,j3)
      if(iptc(k,j).eq.'Z4R')xpt(k)=z4r(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z4RC')xpt(k)=z4rc(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z4TT')xpt(k)=z4tt(j1,j2)
      if(iptc(k,j).eq.'Z4TTS')xpt(k)=z4tts(j1,j2)
      if(iptc(k,j).eq.'Z4TTSW')xpt(k)=z4ttsw(j1,j2)
      if(iptc(k,j).eq.'Z4TTW')xpt(k)=z4ttw(j1,j2)
      if(iptc(k,j).eq.'Z5')xpt(k)=z5(j1,j2,j3)
      if(iptc(k,j).eq.'Z5C')xpt(k)=z5c(j1,j2,j3)
      if(iptc(k,j).eq.'Z5R')xpt(k)=z5r(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z5RC')xpt(k)=z5rc(j1,j2,j3,j4)
      if(iptc(k,j).eq.'Z5RTT')xpt(k)=z5rtt(j1,j2)
      if(iptc(k,j).eq.'Z5TT')xpt(k)=z5tt(j1,j2)
      if(iptc(k,j).eq.'Z5X')xpt(k)=z5x(j1,j2,j3)
      if(iptc(k,j).eq.'Z6')xpt(k)=z6(j1,j2,j3)
      if(iptc(k,j).eq.'Z6MAX')xpt(k)=z6max(j1,j2,j3)
      if(iptc(k,j).eq.'Z7')xpt(k)=z7(j1,j2,j3)
      if(iptc(k,j).eq.'Z77')xpt(k)=z77(j1,j2)
      if(iptc(k,j).eq.'Z77A')xpt(k)=z77a(j1,j2)
      if(iptc(k,j).eq.'Z78')xpt(k)=z78(j1,j2)
      if(iptc(k,j).eq.'Z79')xpt(k)=z79(j1,j2)
      if(iptc(k,j).eq.'Z7A')xpt(k)=z7a(j1,j2)
      if(iptc(k,j).eq.'Z7AR')xpt(k)=z7ar(j1,j2)
      if(iptc(k,j).eq.'Z7MAX')xpt(k)=z7max(j1,j2,j3)
      if(iptc(k,j).eq.'Z8')xpt(k)=z8(j1,j2,j3)
      if(iptc(k,j).eq.'Z9')xpt(k)=z9(j1,j2,j3)
      if(iptc(k,j).eq.'Z9R')xpt(k)=z9r(j1,j2,j3)
      if(iptc(k,j).eq.'ZF')xpt(k)=zf(j1,j2,j3)
```

```
      if(iptc(k,j).eq.'ZFF')xpt(k)=zff(j1,j2,j3)
      if(iptc(k,j).eq.'ZGR')xpt(k)=zgr(j1)
      if(iptc(k,j).eq.'ZGRN')xpt(k)=zgrn(j1)
      if(iptc(k,j).eq.'ZGRNI')xpt(k)=zgrni(j1,j2,j3)
      if(iptc(k,j).eq.'ZKQNT')xpt(k)=zkqnt(j1,j2)
      if(iptc(k,j).eq.'ZKQPH')xpt(k)=zkqph(j1,j2)
      if(iptc(k,j).eq.'ZLV')xpt(k)=zlv(j1)
      if(iptc(k,j).eq.'ZLVI')xpt(k)=zlvi(j1,j2,j3)
      if(iptc(k,j).eq.'ZMAX')xpt(k)=zmax(j1,j2)
      if(iptc(k,j).eq.'ZMEAT')xpt(k)=zmeat(j1)
      if(iptc(k,j).eq.'ZMEATI')xpt(k)=zmeati(j1,j2,j3)
      if(iptc(k,j).eq.'ZMILK')xpt(k)=zmilk(j1)
      if(iptc(k,j).eq.'ZMILKI')xpt(k)=zmilki(j1,j2,j3)
      if(iptc(k,j).eq.'ZMIN')xpt(k)=zmin(j1,j2)
      if(iptc(k,j).eq.'ZMNT')xpt(k)=zmnt(j1,j2)
      if(iptc(k,j).eq.'ZMPH')xpt(k)=zmph(j1,j2)
      if(iptc(k,j).eq.'ZNITRO')xpt(k)=znitro(j1,j2,j3)
      if(iptc(k,j).eq.'ZOBIO')xpt(k)=zobio(j1,j2,j3)
      if(iptc(k,j).eq.'ZOO')xpt(k)=zoo(j1)
      if(iptc(k,j).eq.'ZOOPC')xpt(k)=zoopc(j1,j2,j3)
      if(iptc(k,j).eq.'ZOOPM')xpt(k)=zoopm(j1,j2,j3)
      if(iptc(k,j).eq.'ZOSUM')xpt(k)=zosum(j1,j2)
      if(iptc(k,j).eq.'ZP')xpt(k)=zp(j1,j2)
      if(iptc(k,j).eq.'ZPHOSP')xpt(k)=zphosp(j1,j2,j3)
      if(iptc(k,j).eq.'ZSD')xpt(k)=zsd(j1,j2,j3)
      if(iptc(k,j).eq.'ZSM')xpt(k)=zsm(j1,j2,j3)
      if(iptc(k,j).eq.'ZSV')xpt(k)=zsv(j1)
      if(iptc(k,j).eq.'ZSVI')xpt(k)=zsvi(j1,j2,j3)
      if(iptc(k,j).eq.'ZTMN')xpt(k)=ztmn(j1,j2)
      if(iptc(k,j).eq.'ZTX')xpt(k)=ztx(j1,j2)
      if(iptc(k,j).eq.'ZU')xpt(k)=zu(j1,j2,j3,j4)
      if(iptc(k,j).eq.'ZUP')xpt(k)=zup(j1,j2)
      if(iptc(k,j).eq.'ZZ3')xpt(k)=zz3(j1,j2,j3)
  543 continue
c ** end variable loop
c
      xpt(12)=xptc(1,j)
      isyrt=styr
c     resets day and year numbers to coincide with start of output
      iai1=ai1
      iaiii=aiii
      idmd=dmd
      if(iai1.eq.isyrt.and.iaiii.eq.idmd)tdyt=dmd
      if(isyrt.eq.iai1)tyrt=1.0
      xpt(13)=tdyt
      if(xptc(2,j).gt.0.0)xpt(13)=tdyt/365.0
c
c excludes formatted output (for plotting purposes)
      if (gmm(j)) 51,50,51
c allows only 1 formatted output cycle /time period if allowed at all
   51 if (is(j,1).gt.0) go to 50
      is(k,1)=1
c labels written; acronyms and units
c
```

```
      write (5,580) (lptc(jjj,j),jjj=1,10)
      write (5,590) (lunt(jjj,j),jjj=1,10)
c
c variables and identifiers printed
   50 write (5,570) (xpt(ll),ll=1,15)
      print *, (xpt(ll),ll=1,10),tdyt,tyrt
c
      if(nplots.eq.0)goto656
      do659jp=1,nplots
      call truplt (j,jp,xpt,lx,nplts,iflag,rdidx)
  659 continue
c
  656 continue
      do658kk=1,20
      do658l=1,10
      if(nv(kk,l,j).gt.0)goto657
  658 continue
c
      if(nplots.eq.0)goto540
      dmdr=int(365./dmd)
      aexit=(ai1-1.0)*365.+aiii
      aout=(y2-1.0)*365.+dmdr*dmd
      if(aexit.lt.aout)goto540
      if(regrc(j).lt.1.0.and.pltrc(j).lt.1.0)goto545
      if(regrc(j).ge.1.0.and.pltrc(j).ge.1.0)goto567
      if(regrc(j).lt.1.0)goto 566
      if(jct.eq.ipx)goto545
      goto540
  566 if(jct.eq.ipl)goto545
      goto540
  567 if(jct.ge.ipx.and.jct.ge.ipl)goto545
      goto540
c
c excercises plotting after all records have been generated
  545 if(j.ne.irec)goto540
      do544jp=1,nplots
      call tplotr(j,jp,lx,ngrap,iflag,rdidx)
  544 continue
      lx=0
c
  540 continue
      return
  570 format(10(e10.4,1x),5f4.0)
  580 format(12a10)
  590 format(11a10)
      end
```

14.7. CLIMAT (bgm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine climat (lx,bpx,rdidx,y2,iseed)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  climat estimates climatic parameters stochastically on a
c    daily basis
c
      include 'main.inc'
c
      real l1, l2, l3
c
      common/pc/ anday(12), clm(n), dnn(n), drmm(n), ffp(n), g1(n),g4(n)
     1 , g5(n), mmx(n), nx(24,n), pm(12,n), pms(12,n), pn(n),
     2 px(12,n), q(40,n), q8(n), r(n), t3(n), t4(n), tm(12,n), tms(12,n)
     3 , tn(n), tre(12,n), tx(12,n), x(12,n),relv(n),l2(n),l1(n),l3(n),
     4 prcv(n),tmxv(n),t1(n),t2(n),alpsr(n),alatd(n),rada(n),sunmn(n),
     5 sunmx(n)
c
c  excludes all but first access to subroutine
      if (rdidx-0.0) 30,31,30
   31 if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-555555.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','green')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE CLIMAT                    ')
      call wnoust('******************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('***************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                 ')
      call wnoust('ENTER yes=1, no=0                   ')
      call wnoust('***************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
```

```
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 555555.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c ** regional loop ******************************************************
      do 10 lz=1,ipx
c enters yearly minimum and maximum solar insolation, ly/day; enters lag
c period to maximum insolation,days; yearly minimum and maximum average daily
c temperature, deg. F., and lag period to maximum temperature, days
      print *,'the following entries are for region ',lz
 5002 print *,'ENTER min. and max. solar insolation, ly/d, for region'
      if(nu.eq.0)read(*,*,err=5002)t1(lz),t2(lz)
      if(nu.eq.1)read(7,*)t1(lz),t2(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*)t1(lz),t2(lz)
      if(iu.eq.1)print *,t1(lz),t2(lz)
 5003 print *,'ENTER lag period, days, to maximum insolation'
      if(nu.eq.0)read(*,*,err=5003)l1(lz)
      if(nu.eq.1)read(7,*)l1(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)l1(lz)
      if(iu.eq.1)print *,l1(lz)
 5004 print *,'ENTER yearly min. and max. average daily temp.,deg. F'
      if(nu.eq.0)read(*,*,err=5004)t3(lz),t4(lz)
      if(nu.eq.1)read(7,*)t3(lz),t4(lz)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)t3(lz),t4(lz)
      if(iu.eq.1)print *,t3(lz),t4(lz)
 5005 print *,'ENTER lag period, days, to maximum temperature'
      if(nu.eq.0)read(*,*,err=5005)l2(lz)
      if(nu.eq.1)read(7,*)l2(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*)l2(lz)
      if(iu.eq.1)print *,l2(lz)
c enters mean monthly temperature standard deviation, deg. F.,coefficient of
c variation for each precipitation event, site elevation, feet a.m.s.l.;
c lapse rate, deg. F./foot; random number seed, day and year modulus
 5006 print *,'ENTER mean monthly temperature stnd. dev., deg. F.'
      if(nu.eq.0)read(*,*,err=5006)tmxv(lz)
      if(nu.eq.1)read(7,*)tmxv(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*)tmxv(lz)
      if(iu.eq.1)print *,tmxv(lz)
 5007 print *,'ENTER coefficient of variation for each precip. event'
      if(nu.eq.0)read(*,*,err=5007)prcv(lz)
      if(nu.eq.1)read(7,*)prcv(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*)prcv(lz)
      if(iu.eq.1)print *,prcv(lz)
 5008 print *,'ENTER reference site elevation, feet, a.m.s.l.'
      if(nu.eq.0)read(*,*,err=5008)relv(lz)
      if(nu.eq.1)read(7,*)relv(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*)relv(lz)
      if(iu.eq.1)print *,relv(lz)
 5009 print *,'ENTER region temperature lapse rate, deg. F./ft '
      if(nu.eq.0)read(*,*,err=5009)alpsr(lz)
      if(nu.eq.1)read(7,*)alpsr(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,*)alpsr(lz)
      if(iu.eq.1)print *,alpsr(lz)
c enters total number of precipitation events per month for each region
 5010 print *,'ENTER number of rain events/month (12 values)'
      if(nu.eq.0)read(*,*,err=5010) (tre(j,lz),j=1,12)
      if(nu.eq.1)read(7,*) (tre(j,lz),j=1,12)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
```

```
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*) (tre(j,lz),j=1,12)
      if(iu.eq.1)print *, (tre(j,lz),j=1,12)
c enters monthly precipitation fractions for each region
 5011 print *,'ENTER precipitation fraction/month (12 values)'
      if(nu.eq.0)read(*,*,err=5011) (x(j,lz),j=1,12)
      if(nu.eq.1)read(7,*) (x(j,lz),j=1,12)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*) (x(j,lz),j=1,12)
      if(iu.eq.1)print *, (x(j,lz),j=1,12)
c enters mean annual precipitation,elevation,frost free period ,days, and
c site latitude, deg.
 5012 print *,'ENTER mean regional annual precipitation, inches'
      if(nu.eq.0)read(*,*,err=5012)r(lz)
      if(nu.eq.1)read(7,*)r(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5012) ans
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*)r(lz)
      if(iu.eq.1)print *,r(lz)
 5013 print *,'ENTER study area site elevation, feet, a.m.s.l.'
      if(nu.eq.0)read(*,*,err=5013)g1(lz)
      if(nu.eq.1)read(7,*)g1(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5013) ans
      if(ans.gt.0.and.nu.eq.0)goto5013
      write(6,*)g1(lz)
      if(iu.eq.1)print *,g1(lz)
 5014 print *,'ENTER temp. correction, deg. F.,for latitude '
      print *,' difference from reference site'
      if(nu.eq.0)read(*,*,err=5014)l3(lz)
      if(nu.eq.1)read(7,*)l3(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*)l3(lz)
      if(iu.eq.1)print *,l3(lz)
 5015 print *,'ENTER mean regional frost-free period, days'
      if(nu.eq.0)read(*,*,err=5015)ffp(lz)
      if(nu.eq.1)read(7,*)ffp(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*)ffp(lz)
      if(iu.eq.1)print *,ffp(lz)
 5016 print *,'ENTER regional latitude, deg.'
      if(nu.eq.0)read(*,*,err=5016)alatd(lz)
      if(nu.eq.1)read(7,*)alatd(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*)alatd(lz)
```

```
      if(iu.eq.1)print *,alatd(lz)
   10 continue
 5018 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN CLIMAT                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for CLIMAT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5018) ans
      if(ans)11,5500,11
   11 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-555555.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c ** end regional loop ***************************************************
c
c ** region loop ********************************************************
      pi=3.1416
      do 20 lz=1,ipx
c establishes frost free boundaries
      sffp(lz)=210.-ffp(lz)/2.
      effp(lz)=210.+ffp(lz)/2.
c initializes precipitation and temperature parameters
      f7(lz)=0.0
      tn(lz)=0.0
      pn(lz)=0.0
c estimates minimum and maximum solar inclination angles
      sunmn(lz)=90.-alatd(lz)-23.5
      sunmx(lz)=90.-alatd(lz)+23.5
c angle of latitude in radians
      rada(lz)=alatd(lz)/57.3
   20 continue
c ** end region loop *****************************************************
      do22j=1,iseed
      xrand=random(2)
   22 continue
      return
```

```
c
  30 continue
     iii=aiii
     if (iii.ne.1) go to 50
c initialize temperature and day parameters
     tav(lx)=0.0
     dnn(lx)=0.0
c
c ** plant loop ********************************************************
     do 40 jj=1,ipl
c initializes cumulative moisture additions to soil from all sources
     p1(jj,lx)=0.0
  40 continue
c ** end plant loop ****************************************************
c  tests day number for beginning of new month,and assigns month number
  50 if (iii.eq.1.or.iii.eq.32.or.iii.eq.60.or.iii.eq.91.or.iii.eq.121
    1 .or.iii.eq.152.or.iii.eq.182.or.iii.eq.213.or.iii.eq.244.or.iii
    2 .eq.274.or.iii.eq.305.or.iii.eq.335) go to 60
     go to 120
  60 mmx(lx)=0
     mon=12
     if (iii.le.334) mon=11
     if (iii.le.304) mon=10
     if (iii.le.273) mon=9
     if (iii.le.243) mon=8
     if (iii.le.212) mon=7
     if (iii.le.181) mon=6
     if (iii.le.151) mon=5
     if (iii.le.120) mon=4
     if (iii.le.90) mon=3
     if (iii.le.59) mon=2
     if (iii.le.31) mon=1
c average annual precipitation for month(mon) and region(lx)
     drmm(lx)=x(mon,lx)*r(lx)*25.4
c number of days in month(mon)
     anday(mon)=nday(mon)
c number of precipitation events in January
     ire1=tre(mon,1)
c number of precipitation events in month(mon)
     ire=tre(mon,lx)
c generates precipitation event dates and amounts using random numbers
c initializes random number arrays
     q8(lx)=0.0
c
c ** monthly precipitation event event loop ******************************
     do 70 l=1,ire
c day or month multiplier initialized
     q(l,lx)=0.0
  70 continue
c ** end monthly precip. event loop ***************************************
c
c ** monthly precipitation event loop for all regions *********************
     do 100 l=1,ire
c excludes all but first region
```

```
      if (lx.ne.1) go to 90
c selects day numbers for precipition events for month(mon) and region(lx)
   80 q(l,lx)=random(2)*anday(mon)
c generates random number multiplier for each precipitation event
      q(l+ire,lx)=random(2)
c sums random numbers
      q8(lx)=q8(lx)+q(l+ire,lx)
c assigns day numbers for precipitation events
      aint=int(q(l,lx))
      abc=q(l,lx)-aint
      nx(l,lx)=int(q(l,lx))+1
      if (abc.lt.0.5) nx(l,lx)=int(q(l,lx))
      go to 100
   90 continue
c sets all precipitation and random number assignments to all regions
c with the same number of events in a given month.
      if (l.gt.ire1) go to 80
      q(l,lx)=q(l,1)
      q(l+ire,lx)=q(l+ire,1)
      q8(lx)=q8(lx)+q(l+ire,lx)
      nx(l,lx)=nx(l,1)
  100 continue
c ** end monthly precipitation loop ***************************************
c initializes random number arrays to 0.0
      g4(lx)=0.0
      g5(lx)=0.0
c ** random number selection loop ******************************************
      do 110 l=1,24
c creates two sets of summed random numbers drawn from an even distribution
      if (lx.eq.1) g4(lx)=g4(lx)+random(2)
      g5(lx)=g5(lx)+random(2)
  110 continue
c ** end random number selection loop ***************************************
c creates two random numbers drawn from a normal distribution
      g4(lx)=g4(lx)-12.0
      if (lx.gt.1) g4(lx)=g4(1)
      g5(lx)=g5(lx)-12.0
c sums number of days in month
  120 mmx(lx)=mmx(lx)+1
      ire=tre(mon,lx)
c sums up days in a given year
      dnn(lx)=dnn(lx)+1.0
c day number ratio corrected for solar radiation lag(l1) to total year
      dsol=(dnn(lx)-l1(lx))/365.0
c solar radiation constant mean of maximum(t2) and minimum(t1)
      asol=(t2(lx)-t1(lx))/2.0
c ratio of day number corrected for temperature lag(l2) to total year
      e=(dnn(lx)-l2(lx))/365.
c mean of maximum(t4) and minimum(t3) annual regional temperatures
      btemp=(t4(lx)-t3(lx))/2.0
c ratio of day number to total days in year
      f=dnn(lx)/365.0
c solar radiation,ly, for given day
      akk=asol*(sin(dsol*2.0*pi)+1.0)+t1(lx)
```

```
      solrad=akk
c sets solar radiation saturation maximum for plants in north latitudes
      if (akk.gt.450.) akk=450.
c sets photosynthetic transformation rate as a function of solar radiation
      sa=0.001*akk
c estimates average daily temoerature, degrees F.
      t(lx)=btemp*(sin(e*2.*pi)+1.)+t3(lx)
c corrects temperature for elevation relative to reference elevation
c and latitude difference from reference site
      t(lx)=t(lx)+(relv(lx)-g1(lx))*alpsr(lx)+l3(lx)
c adjusts temperature for variance from mean using 1-sigma stnd. dev.(tmxv)
c and convert to degrees C.
      t(lx)=t(lx)+g4(lx)*tmxv(lx)
      t(lx)=(t(lx)-32.)*5./9.
c estimate daily declination angle for regions, degrees
      declin=47.*(sin(dsol*2.*3.1416)+1.)*.5-23.5
c estimates daily zenith angle for regions, degrees
      radang=(sunmx(lx)-sunmn(lx))*(sin(dsol*2.*3.1416)+1.0)*0.5+
     1sunmn(lx)
      deca=declin/57.3
c hour angle estimated for region , radians
      hrang=-tan(rada(lx))*tan(deca)
      if (hrang.gt.0.999999) hrang=0.999999
      if (hrang.lt.-0.999999) hrang=-0.999999
c photoperiod estimated for regions, hours
      phohr=acos(hrang)
      photpr=phohr*57.3*24./180.
c fraction of 24 hour day that is daylight
      ipp=photpr
      app=ipp*10
      bpp=photpr*10.
      photpr=ipp
      if (bpp-ipp.ge.5.) photpr=ipp+1.
      dayfrc=photpr/24.
c 24 hour maximum and minimum temperature is estimated
      txx=t(lx)
      tmx(lx)=t(lx)+.53*(abs(txx)+1.0)
      tmn(lx)=t(lx)-.45*(abs(txx)+1.0)
      if (tmx(lx).le.0.0) tmx(lx)=0.0
      if (tmn(lx).le.0.0) tmn(lx)=0.0
      if (t(lx).le.0.0) t(lx)=0.0
c fraction of 24 hour day that is night
      anite=24.-photpr
c mean nightly temperature initialized for estimation
      antav(lx)=0.0
      hrct=0.0
c
c ** 24 hour loop ********************************************************
      do 140 j=1,24
      aj=j
c lag to minimum temperature (sets minimum to 5am)
      dhr=(aj-6.)/photpr
      if (aj.gt.photpr) go to 130
      thr(j,lx)=(tmx(lx)-tmn(lx))*(sin(dhr*2.*3.1416)+1.)*0.5+tmn(lx)
```

```
      thrp=thr(j,lx)
      go to 140
  130 hrct=hrct+1.0
      thr(j,lx)=thrp-hrct*(thrp-tmn(lx))/anite
c mean nightly temperature estimated
      antav(lx)=antav(lx)+thr(j,lx)/(24.-photpr)
  140 continue
c ** end 24 hour loop ******************************************************
c daily precipitation estimate initialized to 0.0
      p(lx)=0.0
c estimates mean annual temperature
      tav(lx)=tav(lx)+t(lx)/365.
c
c  assigns precipitation amounts to specified days,and determines
c     snowfall and snowmelt based on temperature
c
c *** precipitation event loop *********************************************
      do 160 l=1,ire
      lo=l
      if (mmx(lx).eq.nx(l,lx)) go to 150
      go to 160
  150 nxx=lo+ire
      pe=q(nxx,lx)*drmm(lx)/q8(lx)
c estimates daily precipitation adjusted for variance(prcv)
      p(lx)=p(lx)+pe+pe*g5(lx)*prcv(lx)
  160 continue
c ** end precipitation event loop ******************************************
c estimates cummulative daily precipitation
      pn(lx)=pn(lx)+p(lx)
c estimates cummulative temperature
      tn(lx)=tn(lx)+t(lx)
      if (iii.eq.31.or.iii.eq.59.or.iii.eq.90.or.iii.eq.120.or.iii.eq.15
     1 1.or.iii.eq.181.or.iii.eq.212.or.iii.eq.243.or.iii.eq.273.or.iii
     2 .eq.304.or.iii.eq.334.or.iii.eq.365) go to 170
      go to 180
c mean monthly temperature for climograph generation
  170 tm(mon,lx)=tn(lx)/anday(mon)
      tn(lx)=0.0
c mean monthly precipitation for climograph generation
      pm(mon,lx)=pn(lx)
      pn(lx)=0.0
  180 if (t(lx).gt.2.0) go to 190
c snowfall accumulation
      f7(lx)=f7(lx)+p(lx)*10.
      p(lx)=0.0
  190 continue
      tt=t(lx)
      prec(lx)=0.0
c potential snowmelt thickness,mm
      snm(lx)=11.28*tt+1.1128
c actual snowmelt thickness,mm
      if (f7(lx).le.0.0) snm(lx)=0.0
      if (f7(lx).lt.snm(lx)) snm(lx)=f7(lx)
c snow thickness adjusted for snowmelt
```

```
      f7(lx)=f7(lx)-snm(lx)
      if (f7(lx).le.0.0) f7(lx)=0.0
c soil moisture addition incremented for snowmelt
      prec(lx)=p(lx)+snm(lx)/10.
c
c           potential evapotranspiration module
c
c the slope of the saturation pressure estimated
      deltt=(5304./(tt+273.)**2)*exp(21.255-5304./(tt+273.))
c albedo of land surface set at 0.2
      albd=0.2
c net daily solar radiation absorbed by land surface at mean air temp.
      ho=(1.0-albd)*solrad/58.3
c barometric pressure adjusted for elevation, mbars
      pmbar=1013.-.1152*(.3048*g1(lx))+5.44e-06*(g1(lx)*.3048)**2
c psychrometric  constant estimated
      apsyc=.0006595*pmbar
c potential evapotranspiration ,mm/day from land surface
      e2(lx)=(1.28*deltt*ho)/(deltt+apsyc)
      ammm=mon
c excludes all but last day of year
      if (aiii.lt.365.) go to 220
c subroutine index number for CLIMAT
      saj=201.
c
c ** month loop ******************************************************
      do 210 k=1,12
      ak=k
c cummulative monthly  temperature and precipitation over year(ai1)
      tms(k,lx)=tms(k,lx)+tm(k,lx)
      pms(k,lx)=pms(k,lx)+pm(k,lx)
      if (ai1-y2) 200,201,200
c mean yearly precipitation and temperature for year(ai1)
  201 px(k,lx)=pms(k,lx)/ai1
      tx(k,lx)=tms(k,lx)/ai1
  200 continue
  210 continue
c ** end month loop ******************************************************
  220 continue
      return
      end
```

14.8. EROSON (bhm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine eroson (lpg,lg,wchx,schxx,rchxx,qflow)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
c  excludes channels
      if (lg.gt.ipx) go to 10
c crown cover fraction ratio to total crown cover of plant(lg) in region(lg)
      scvr(lpg,lg)=((ccvv(lpg,lg)+.01/aplx(lg))/tcvr(lg))
c crown cover fraction ratio to regional area for irrigated crop
      if (acres(lpg,lg)-rarea(lg)) 11,12,11
   11 scvr(lpg,lg)=acres(lpg,lg)/rarea(lg)
c regional runoff estimate,inches,from all plant groups(lpg)
   12 rndlx(lg)=rndlx(lg)+rnofds(lpg,lg)*scvr(lpg,lg)/25.4
c peak runoff,inches/hour, regional area ,square miles, and peak runoff
c ,cfs, from regions or channels
   10 qpkd(lg)=c5hr(lg)*qflow/duhr(lg)
      rasmi=rarea(lg)/640.
c qflow =region runoff,inches. vtrnd channel transmission,cfd
      cfspd(lg)=1.008*(c5hr(lg)/c1hr(lg))*qflow*rasmi**(1.-c2hr(lg))
      if (lg.gt.ipx) cfspd(lg)=c5hr(lg)*vtrnd(lg)/(duhr(lg)*3600.)
      if (qflow.le.0.0) return
      if (lg.gt.ipx) go to 20
c tonflw= tons/acre of sediment eroded from region and accumulator
c aksoil,csoil,psoil,and alsoil = soil erosion parameters
      tonflw=95.*((qflow*cfspd(lg))**.56)*aksoil(lg)*csoil(lg)*psoil(lg)
     1 *alsoil(lg)*scvr(lpg,lg)
      tndlx(lg)=tndlx(lg)+tonflw
c total eroded soil,tons, and total runoff,cfd, from region
      tndtn(lg)=tndtn(lg)+tonflw*rarea(lg)
      rdcf(lg)=rdcf(lg)+rnofds(lpg,lg)*(scvr(lpg,lg)/(25.4*12.))*rarea
     1 (lg)*43562.
c regional and channel depth,feet
   20 dchan(lg)=qflow/12.
      if (lg.gt.ipx) dchan(lg)=qflow/(12.*duhr(lg)*3600.)
c region width,feet
      if (lg.le.ipx) wchan(lg)=xchan(lg)*5280.
c default channel width,feet
      if (wchx) 22,21,22
   21 wchan(lg)=2.17*(cfspd(lg)/3.)**0.5
c rhyrc=hyraulic radius of channel or regions,feet, for surface moisture
   22 rhyrc=wchan(lg)*dchan(lg)/(wchan(lg)+2.0*dchan(lg))
c vchan = channel or region velocity(ft/sec) of surface flow
c schan= slope of channel or region(rise/run)
c default slope
      if (schxx) 24,23,24
   23 schan(lg)=.000771*(cfspd(lg)/3.)**(-.3)
c rchan = roughness or manning coefficient for regions or channels
```

```
c default roughness
  24 if (rchxx) 26,25,26
  25 rchan(lg)=.0301*(cfspd(lg)/3.)**(-.05)
  26 vchan(lg)=(1.49/rchan(lg))*(schan(lg)**(0.5))*(rhyrc**(0.66))
c hydraulic roughness of channel or regional beds
    rchanb=(rchan(lg)**(1.5)+(2.*dchan(lg)/wchan(lg))*(rchan(lg)**(.66
   1 )-(1.32*rchan(lg))**(1.5)))**(0.66)
    if (rchanb.le.0.63*rchan(lg)) rchanb=0.63*rchan(lg)
    wdr=((wchan(lg)+4.*dchan(lg))/(4.*dchan(lg)))**(0.66)
    if (rchanb.ge.wdr*rchan(lg)) rchanb=wdr*rchan(lg)
c hydraulic radius for bed resistence,feet
    rhyrb=rhyrc*(rchanb/rchan(lg))**(1.5)
c grain or particle resistence coefficient
    rchang=0.132*dsz(8,lg)**(0.166)
c hydraulic radius for grain resistence,feet, for regions/channels
    rhyrg=rhyrb*(rchang/rchanb)**(1.5)
c effective shear stress for sediment transport,lbs/ft2, for regions/channels
    sstres=62.43*rhyrg*schan(lg)
c particle transport coefficient for median diameter diameter,cf/lb-sec
    bsd50=40./(dsz(8,lg)**(1.5))
c critical shear stress for median particle diameter, lbs/ft2
    sstr50=.0022+.01*dsz(8,lg)
    if (dsz(8,lg).gt.1.0) sstr50=-.0078+.02*dsz(8,lg)
c particle transport capacity per unit width of channel/region for median
c diameter, lbs/sec-ft
    gsbd50=bsd50*sstres*(sstres-sstr50)
    if (gsbd50.le.0.0) gsbd50=0.0
c particle transport capacity,lbs/sec-ft,for diameters less than .063 mm
    gsbd(1,lg)=dsf(1,lg)*2.0*sstres*vchan(lg)**2
    if (gsbd(1,lg).le.0.0) gsbd(1,lg)=0.0
c transport capacity accumulator initialized
    gsbdt=0.0
c
c ** particle size loop ************************************************
    do 30 ls=2,7
c particle transport coeficient,critical shear stress, and transport
c capacity of particles greater than 0.063 mm
    bsdi=40./(dsz(ls,lg)**(1.5))
    sstri=.0022+.01*dsz(ls,lg)
    if (dsz(ls,lg).gt.1.0) sstri=-.0078+.02*dsz(ls,lg)
    gsbd(ls,lg)=dsf(ls,lg)*bsdi*sstres*(sstres-sstri)
    if (gsbd(ls,lg).le.0.0) gsbd(ls,lg)=0.0
    gsbdt=gsbdt+gsbd(ls,lg)
  30 continue
c ** end particle size loop ********************************************
c particle transport capacities normalized to median particle value
    if (gsbdt.le.0.0) gsbdt=0.0
    alphi=0.0
    if (gsbdt.gt.0.0) alphi=gsbd50/gsbdt
    gsbdtt(lg)=gsbd(1,lg)
    if(gsbdtt(lg).le.0.0)gsbdtt(lg)=0.0
c
c ** particle size loop greater than .063 mm diameter ******************
    do 40 ls=2,7
```

```
      gsbd(ls,lg)=alphi*gsbd(ls,lg)
      if(gsbd(ls,lg).le.0.0)gsbd(ls,lg)=0.0
      gsbdtt(lg)=gsbdtt(lg)+gsbd(ls,lg)
   40 continue
      if(gsbdtt(lg).le.0.0)gsbdtt(lg)=0.0
c ** end particle size loop ***********************************************
c
c **particle diameter loop including .063 mm diameter ********************
      if(gsbdtt(lg).gt.0.0)sedmns(lg)=0.0
      if(gsbdtt(lg).gt.0.0)sdmnsa(lg)=0.0
      do 50 ls=1,7
      if (gsbdtt(lg).le.0.0) go to 50
      alphz=gsbd(ls,lg)/gsbdtt(lg)
c channel flow is strapped to surface runoff sediment yield to
c estimate losses according to particle diameter, lbs/day
      if (lg.le.ipx) gsbd(ls,lg)=tonflw*rarea(lg)*alphz*2000.
      if (lg.gt.ipx) gsbd(ls,lg)=alphz*gsbdtt(lg)*duhr(lg)*3600.*wchan
     1 (lg)
      if(gsbd(ls,lg).le.0.0)gsbd(ls,lg)=0.0
c limits daily transport to total inventory of specific diameter class,lbs
      if (gsbd(ls,lg).ge.dsms(ls,lg)) gsbd(ls,lg)=dsms(ls,lg)
      if(gsbd(ls,lg).le.0.0)gsbd(ls,lg)=0.0
c increments daily loss(lbs);adjusts current inventory for diameter(ls)
      dsm(ls,lg)=gsbd(ls,lg)+dsm(ls,lg)
      if(dsm(ls,lg).le.0.0)dsm(ls,lg)=0.0
      dsms(ls,lg)=dsms(ls,lg)-gsbd(ls,lg)
      if (dsms(ls,lg).le.0.0) dsms(ls,lg)=0.0
c adjusts total inventory(all particle sizes) for loss,lbs
      sedmns(lg)=sedmns(lg)+dsms(ls,lg)
c daily activity/mass removal, pci,...ug..etc;adjustment of total inventory
      dsact=gsbd(ls,lg)*dzas(ls,lg)*454.
      if(dsact.le.0.0)dsact=0.0
      dsma(ls,lg)=dsma(ls,lg)-dsact
      if(dsms(ls,lg).le.0.0)dsma(ls,lg)=0.0
      if (dsma(ls,lg).le.0.0) dsma(ls,lg)=0.0
c adjustment of total activity/mass inventory for losses
      sdmnsa(lg)=sdmnsa(lg)+dsma(ls,lg)
   50 continue
c ** end particle size loop ***********************************************
      sdmnct(lg)=0.0
      if(sedmns(lg).gt.0.0)sdmnct(lg)=sdmnsa(lg)/(sedmns(lg)*454.)
c
c ** particle size loop, all sizes ***************************************
      do 60 ls=1,7
c total sediment removal, all sizes combined,lbs/day
      dsmt(lg)=dsmt(lg)+gsbd(ls,lg)
      if(dsmt(lg).le.0.0)dsmt(lg)=0.0
c initialize specific activity; recalculate specific activity and particle
c diameter fractions adjusted for losses and gains
      dzas(ls,lg)=0.0
      if (dsms(ls,lg).gt.0.0) dzas(ls,lg)=dsma(ls,lg)/(dsms(ls,lg)*454.)
      if(dzas(ls,lg).le.0.0)dzas(ls,lg)=0.0
      dsf(ls,lg)=0.0
      if (sedmns(lg).gt.0.0) dsf(ls,lg)=dsms(ls,lg)/sedmns(lg)
```

```
      if(dsf(ls,lg).le.0.0)dsf(ls,lg)=0.0
   60 continue
c ** end particle size loop ***********************************************
      return
      end
c
```


454

14.9. FORCUT (bim.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine forcut (jk,lx,dc,iipx,iijk,rdidx,bpx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
      common/pd/ acut(i,2,n), bcut(i,2,n), ccut(i,2,n), datct(i,10,n),
     1 frt(n), racut(i,n), rbcut(i,n), rccut(i,n), satrs(m,n), tcut(i,10
     2 ,n), vcut(kt,n), z3wa2f(m,n)
c
c   forcut simulates different stand cutting practices or fire
c   events(k) for the stands developed in the forman subroutine
c
c  for each cutting or fire event (tbrm is total number of events) varia
c  ble tcut is set to type of cut or fire.  for choice cut (tcut=1.0),
c  reverse "j" cut (tcut=2.0),and forest fire (tcut=3.0) use tcut to
c  specify.  for tcut=1.0  must specify acut,bcut,ccut, and
c  racut,rbcut, and rccut.
c
c excludes all but first access for parameter input and initialization
      if (rdidx.gt.0.) go to 10
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
      ict1=0
      ict2=0
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-666666.)5702,5704,5702
 5704 if(ict1.ne.iipx)ict1=ict1+1
      if(ict2.ne.iijk)ict2=ict2+1
      if(ict1.eq.iipx.and.ict2.eq.iijk)goto5050
      goto5702
 5555 nu=0
 5050 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,6)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE FORCUT                        ')
      call wnoust('                                    ')
      call wnoust('                                    ')
      call wnoust('******************************                    ')
      call wnouce(6,'press any key')
      call inkey(key)
      call wncuxy(1,3)
      print *,'management number=',iijk
      print *,'region number=',iipx
c the user may change mode of input for this subroutine
c
```

```
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 666666.                     ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
      ibrm=tbrm(jk,lx)
c
c reads year number(s) for timber management(DATCT) for plant(JK); enters
c type of cutting option for each mangement scenario(TCUT); there are a
c total of 10 managements possible/simulation
c
 5003 print *,'ENTER ',ibrm,'timber harvests for plant',jk,'in'
      print *,'region',lx
      if(nu.eq.0)read(*,*,err=5003)(datct(jk,k,lx),k=1,ibrm)
      if(nu.eq.1)read(7,*)(datct(jk,k,lx),k=1,ibrm)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)(datct(jk,k,lx),k=1,ibrm)
      if(iu.eq.1)print *,(datct(jk,k,lx),k=1,ibrm)
```

```
      print *,'1.=sawtimber(12-40),2.=rev.J.(above 26),3.=fire(all)'
      print *,'values in parentheses. are diameters in inches'
 5004 print *,'ENTER',ibrm,' type of cuts(1.,2.,3.) for plant',jk
      print *,'in region',lx
      if(nu.eq.0)read(*,*,err=5004)(tcut(jk,k,lx),k=1,ibrm)
      if(nu.eq.1)read(7,*)(tcut(jk,k,lx),k=1,ibrm)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)(tcut(jk,k,lx),k=1,ibrm)
      if(iu.eq.1)print *,(tcut(jk,k,lx),k=1,ibrm)
c
c     1=choice cut,sawtimber(12-40),pulpwood(6-10)precomm(0-4),clear cut
c     2=reverse-j cut,and all sizes above 26 inches dia. removed
c     3=fire occurance,all classes affected
c
c reads upper(ACUT1,BCUT1,CCUT1) and lower (ACUT2..CCUT2) diameters selected
c for cutting or fire losses
 5005 print *,'ENTER lower diameter tree cutting dia. interval'
      print *,',inches, for plant',jk,'in region',lx,'; for rvj'
      print *,'cut(2.) enter 0.0; 2 entries (lower,upper);'
      print *,'0.0 may be entered for empty intervals(0.,0.)'
      if(nu.eq.0)read(*,*,err=5005)acut(jk,1,lx),acut(jk,2,lx)
      if(nu.eq.1)read(7,*)acut(jk,1,lx),acut(jk,2,lx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*)acut(jk,1,lx),acut(jk,2,lx)
      if(iu.eq.1)print *,acut(jk,1,lx),acut(jk,2,lx)
 5006 print *,'ENTER middle diam. interval as above(2 entries)'
      if(nu.eq.0)read(*,*,err=5006)bcut(jk,1,lx),bcut(jk,2,lx)
      if(nu.eq.1)read(7,*)bcut(jk,1,lx),bcut(jk,2,lx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*)bcut(jk,1,lx),bcut(jk,2,lx)
      if(iu.eq.1)print *,bcut(jk,1,lx),bcut(jk,2,lx)
 5007 print *,'ENTER upper diam. interval as above(2 entries)'
      if(nu.eq.0)read(*,*,err=5007)ccut(jk,1,lx),ccut(jk,2,lx)
      if(nu.eq.1)read(7,*)ccut(jk,1,lx),ccut(jk,2,lx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*)ccut(jk,1,lx),ccut(jk,2,lx)
      if(iu.eq.1)print *,ccut(jk,1,lx),ccut(jk,2,lx)
c estimates fraction remaining for each cutting class
 5008 print *,'ENTER remaining fraction for each (3) cutting'
      print *,'class for plant',jk,'in region',lx
      if(nu.eq.0)read(*,*,err=5008)racut(jk,lx),rbcut(jk,lx),
     1rccut(jk,lx)
      if(nu.eq.1)read(7,*)racut(jk,lx),rbcut(jk,lx),rccut(jk,lx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
```

```
      write(6,*)racut(jk,lx),rbcut(jk,lx),rccut(jk,lx)
      if(iu.eq.1)print *,racut(jk,lx),rbcut(jk,lx),rccut(jk,lx)

 5009 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN FORCUT                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for FORCUT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5009) ans
      if(ans) 11,5500,11
   11 rewind 6
      ict1=0
      ict2=0
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-666666.)5501,5503,5501
 5503 if(ict1.ne.iipx)ict1=ict1+1
      if(ict2.ne.iijk)ict2=ict2+1
      if(ict1.ne.iipx.or.ict2.ne.iijk)goto5501
      backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
      return
c
c  if cutting or fire occurs during this year (datct equals ai1) then
c  forcut is accessed.
c
   10 continue
c sets number of cuttings(managements) for plant(JK) in region(LX)
      ibrm=tbrm(jk,lx)
c sets time to current year
      atme=ai1
c if accessed daily (RGC=3) time set to current julian day number
      if (rgc(jk,lx)-3.0) 13,12,13
   12 atme=aiii
c
c ** tree management loop ******************************************
   13 do 20 k=1,ibrm
      kk=k
```

```
c selects cutting dates
      if (datct(jk,k,lx)-atme) 20,30,20
   20 continue
c ** end tree management loop ****************************************
      ictpl=0
      return
c
c  sets type of cut
   30 itc=tcut(jk,kk,lx)
c initializes timber volume removed (cubic feet)
      vtot=0.0
c
c ** diameter class loop ***********************************************
      do 40 k=1,kt
c initializes volume of timber removed from given diameter class
   40 vcut(k,lx)=0.0
c ** end diameter class loop *******************************************
c excludes reverse-j cut
      if (itc.eq.2) goto110
c
c  for choice cut acut,bcut,ccut,racut,rbcut, and rccut are used to
c  estimate the remaining stand biomass parameters.
c  choice cuts(1) and forest fires(3); selects upper and lower diameters and
c fraction removed for upm to three diameter groupings
c
      ia1=acut(jk,1,lx)
      ia2=acut(jk,2,lx)
      ib1=bcut(jk,1,lx)
      ib2=bcut(jk,2,lx)
      ic1=ccut(jk,1,lx)
      ic2=ccut(jk,2,lx)
c excludes a-cuts
      if (ia1.eq.0) go to 60
c
c ** a-cut loop ******************************************************
      do 50 k=ia1,ia2
c excludes empty diameter classes
      if (vtmbr(jk,k,lx).le.0.0) go to 50
c volume of timber removed ,cubic feet,(VCUT)
      vcut(k,lx)=vtmbr(jk,k,lx)*(1.-racut(jk,lx))
c number of trees remaining estimated
      dtbr(jk,k,lx)=dtbr(jk,k,lx)*(vtmbr(jk,k,lx)-vcut(k,lx))/vtmbr(jk,k
     1 ,lx)
c volume removed summed
      vtot=vtot+vcut(k,lx)
   50 continue
c ** end a-cut loop ***************************************************
c excludes b-cuts
   60 if (ib1.eq.0) go to 80
c
c ** b-cut loop ******************************************************
      do 70 k=ib1,ib2
c same as for a-cuts (above)
      if (vtmbr(jk,k,lx).le.0.0) go to 70
```

```
      vcut(k,lx)=vtmbr(jk,k,lx)*(1.-rbcut(jk,lx))
      dtbr(jk,k,lx)=dtbr(jk,k,lx)*(vtmbr(jk,k,lx)-vcut(k,lx))/vtmbr(jk,k
     1 ,lx)
      vtot=vtot+vcut(k,lx)
   70 continue
c ** end b-cut loop ***************************************************
c excludes c-cuts
   80 if (ic1.eq.0) go to 100
c
c ** c-cut loop *******************************************************
      do 90 k=ic1,ic2
c same as for a-cuts (above)
      if (vtmbr(jk,k,lx).le.0.0) go to 90
      vcut(k,lx)=vtmbr(jk,k,lx)*(1.-rccut(jk,lx))
      dtbr(jk,k,lx)=dtbr(jk,k,lx)*(vtmbr(jk,k,lx)-vcut(k,lx))/vtmbr(jk,k
     1 ,lx)
      vtot=vtot+vcut(k,lx)
   90 continue
c ** end c-cut loop ***************************************************
c selects for  forest fire scenario
  100 if (itc.eq.3) go to 140
      go to 230
c diverts to biomass reduction adjustments
c
c ** diameter class loop **********************************************
  110 do 130 k=1,kt
c  reverse "j" cutting is estimated by cutting trees  greater than
c  25 inches in diameter
      ak=2.0*k
c excludes empty diameter classes
      if (vtmbr(jk,k,lx).le.0.0) go to 130
c excludes tree dia. classes greater than 26 inches
      if (ak.ge.26.) go to 120
c estimates cubic feet remaining for dia. class(AK)
      rvj=126.0*exp(-.197*ak)
      rvj=rvj*z(k,lx)
c excludes dia. class if projected cut volume exceeds that available
      if (vtmbr(jk,k,lx).le.rvj) go to 130
c estimates volume removed, cubic feet; number of trees remaining
      vcut(k,lx)=vtmbr(jk,k,lx)-rvj
      dtbr(jk,k,lx)=dtbr(jk,k,lx)*(vtmbr(jk,k,lx)-vcut(k,lx))/vtmbr(jk,k
     1 ,lx)
c sums volume removed
      vtot=vtot+vcut(k,lx)
      go to 130
c totally removes all trees equal to or greater than 26 inches in diameter
  120 vcut(k,lx)=vtmbr(jk,k,lx)
c adjusts the number of trees remaining to 0.0
      dtbr(jk,k,lx)=dtbr(jk,k,lx)*(vtmbr(jk,k,lx)-vcut(k,lx))/vtmbr(jk,k
     1 ,lx)
c sums total timber volume removed
      vtot=vtot+vcut(k,lx)
  130 continue
c ** end diameter class loop ******************************************
```

```
      go to 230
c diverts to biomass density adjustments
c
c ** diameter class loop (excludes 1st tree diameter) ***************
  140 do 160 k=2,kt
c forest fire scenario; excludes empty dia. classes
      if (vtmbr(jk,k,lx).le.0.0.or.vcut(k,lx).le.0.0) go to 160
c estimates volume remaining; VCUT estimated above(cut types 1,3 above)
      vtmbr(jk,k,lx)=vtmbr(jk,k,lx)-vcut(k,lx)
c conversion factors; cubic feet to gdwt; mass(g) of top 20 mm soil
      vtbmc=.142/c9(jk,lx)
      sl14=slice(jk,1,lx)*1400.
c
c ** contaminant loop ************************************************
      do 150 j=1,inz
c total contaminant concentration (per m2)  removed from both internal and
c external contamination(SATRS) and transported to soil surface
      xsatb=satb(jk,k,j,lx)*vcut(k,lx)/(vtbmc*4047.)
      satrs(j,lx)=xsatb+a2(jk,j,lx)+a3(jk,j,lx)+afr(jk,j,lx)
     1 +afs(jk,j,lx)+zsd(jk,j,lx)+wz3(jk,j,lx)
      satrs(j,lx)=satrs(j,lx)+b2(jk,j,lx)+bb2(jk,j,lx)+b3(jk,j,lx)+bb3
     1 (jk,j,lx)
c soil contaminant increment (per gdwt) assumes 70% deposition of ash
      z1(jk,1,j,lx)=z1(jk,1,j,lx)+0.7*satrs(j,lx)/sl14
c air contaminant increment(per m3) assumes 10% dispersal of ash and
c a 30 meter mixing layer
      aaa(jk,j,lx)=aaa(jk,j,lx)+0.1*satrs(j,lx)/30.
c extimates total live above ground contaminant content(per m2);bounds
c for use determining contaminant transport to dead standing biomass
      xz3wz3= z3(jk,j,lx)+wz3(jk,j,lx)
      if(xz3wz3.le.0.0)xz3wz3=1.0
c dead standing contaminant content(per m2) for bole(Z4) and leaves
c (WZ4) assumes 20% conversion of live standing biomass as a result
c of fire
      z4(jk,j,lx)=z4(jk,j,lx)+0.2*satrs(j,lx)*(z3(jk,j,lx)/xz3wz3)
      wz4(jk,j,lx)=wz4(jk,j,lx)+0.2*satrs(j,lx)*(wz3(jk,j,lx)/xz3wz3)
c live standing above ground bole, leaves, and fruits adjusted for fire
c losses
c adjusts live above ground biomass(Z3),fruits(ZSD),and leaf(WZ3) internal
c contaminant concentration(per m2)
      z3(jk,j,lx)=z3(jk,j,lx)*(c7-vtot)/c7
      zsd(jk,j,lx)=zsd(jk,j,lx)*(c7-vtot)/c7
      wz3(jk,j,lx)=wz3(jk,j,lx)*(c7-vtot)/c7
  150 continue
c ** end contaminant loop ********************************************
  160 continue
c ** end diameter class loop *****************************************
c
c ** plant loop ******************************************************
      do 220 kx=1,ipl
      ictpl=ictpl+1
c excludes plants that are managed
      if (rgc(kx,lx)-1.0) 180,181,180
c
```

```
c ** contaminant loop ***************************************************
 181 do 170 k=1,inz
c sums total contaminant content(per m2) of live above ground biomass
c for both internal and external components from plant(KX) on plant(K)
     z3wa2f(k,lx)=wz3(kx,k,lx)+a2(kx,k,lx)+afr(kx,k,lx)+z3(kx,k,lx)+a3
    1 (kx,k,lx)+afs(kx,k,lx)+zsd(kx,k,lx)
     z3wa2f(k,lx)=z3wa2f(k,lx)+b2(kx,k,lx)+b3(kx,k,lx)+bb3(kx,k,lx)+bb2
    1 (kx,k,lx)
c adjusts soil concentrations(per gram) for forest fire scenario for
     z1(kx,1,k,lx)=z1(kx,1,k,lx)+0.7*(vtot/c7)*z3wa2f(k,lx)/(slice(kx,1
    1 ,lx)*1400.)
c removes saltation-creep contaminant components for resuspension estimate
     z3wa2f(k,lx)=z3wa2f(k,lx)-a3(kx,k,lx)-afs(kx,k,lx)-b3(kx,k,lx)-bb3
    1 (kx,k,lx)
c adjusts internal and external contaminant concentrations(per m2) assuming
c proportional destruction of understory plants as with trees.
c adjustments are made on both live and dead standing vegetation
     aaa(kx,k,lx)=0.1*(vtot/c7)*z3wa2f(k,lx)/1.0+aaa(kx,k,lx)
     z4(kx,k,lx)=z4(kx,k,lx)+0.2*(vtot/c7)*(z3(kx,k,lx)+zsd(kx,k,lx))
     wz4(kx,k,lx)=wz4(kx,k,lx)+.2*(vtot/c7)*wz3(kx,k,lx)
     wz42(kx,k,lx)=wz42(kx,k,lx)*(c7-vtot)/c7+.2*a2(kx,k,lx)*vtot/c7
c estimates made on basis of integral average saltation-creep values
     wz43(kx,k,lx)=wz43(kx,k,lx)*(c7-vtot)/c7+.3*a3av(kx,k,lx)*vtot/c7
     bsav=b3av(kx,k,lx)+bb3av(kx,k,lx)+afsav(kx,k,lx)
     z43(kx,k,lx)=z43(kx,k,lx)*(c7-vtot)/c7+.3*(vtot/c7)*bsav
     z43(kx,k,lx)=z43(kx,k,lx)*(c7-vtot)/c7+.3*(b3(kx,k,lx)+bb3(kx,k,lx
    1 )+afs(kx,k,lx))
c
c estimates of external resuspension deposit includes that present on bole,
c branches, and leaves assuming 20% contribution from fire minus losses
c
     z42(kx,k,lx)=z42(kx,k,lx)*(c7-vtot)/c7+.2*(b2(kx,k,lx)+bb2(kx,k,lx
    1 )+afr(kx,k,lx))
c adjusts both internal and external parameters based on losses proportional
c to tree losses due to fire
     z3(kx,k,lx)=z3(kx,k,lx)*(c7-vtot)/c7
     wz3(kx,k,lx)=wz3(kx,k,lx)*(c7-vtot)/c7
     a2(kx,k,lx)=a2(kx,k,lx)*(c7-vtot)/c7
     afr(kx,k,lx)=afr(kx,k,lx)*(c7-vtot)/c7
     b3(kx,k,lx)=b3(kx,k,lx)*(c7-vtot)/c7
     bb3(kx,k,lx)=bb3(kx,k,lx)*(c7-vtot)/c7
     b2(kx,k,lx)=b2(kx,k,lx)*(c7-vtot)/c7
     bb2(kx,k,lx)=bb2(kx,k,lx)*(c7-vtot)/c7
     a3(kx,k,lx)=a3(kx,k,lx)*(c7-vtot)/c7
     afs(kx,k,lx)=afs(kx,k,lx)*(c7-vtot)/c7
     zsd(kx,k,lx)=zsd(kx,k,lx)*(c7-vtot)/c7
 170 continue
c ** end contaminant loop ***********************************************
c excludes all trees except tree under consideration and understory plants
 180 if (rgc(kx,lx)-1.0) 222,221,222
 222 if (jk.ne.kx) goto220
c excludes tree not under consideration ;allows adjustment for first tree
c type in fire only; adjusts current tree under consideration
 221 if (ictpl.gt.ipl.and.jk.ne.kx) go to 220
```

```
c converts live root to dead root based on total timber volume removal
      rd1(kx,lx)=rd1(kx,lx)+r1(kx,lx)*vtot/c7
c calculates number of soil layers
      iicde=rcode(kx,lx)
c
c ** soil layer loop ***************************************************
      do 200 j=1,iicde
c estimates root fraction/layer; assumes total fraction entered for annuals
      rotfr=zff(kx,j,lx)*zf(kx,j,lx)
      if (hf(kx,lx)-1.0) 202,201,202
  201 rotfr=zf(kx,j,lx)
c
c ** contaminant loop **************************************************
  202 do 190 k=1,inz
c estimates dead root contaminant content(per m2) from internal and
c adsorbed contaminant on root bole and branches based on total volume
c removed
      z4r(kx,j,k,lx)=z4r(kx,j,k,lx)+rotfr*(z2(kx,k,lx)+rb3(kx,k,lx)+rbb3
     1 (kx,k,lx))*vtot
  190 continue
c ** end contaminant loop **********************************************
  200 continue
c ** end soil layer loop **************************************************
c adjusts live standing above and below ground biomass(per m2) based on
c total volume removed
      r1(kx,lx)=r1(kx,lx)-r1(kx,lx)*vtot/c7
      b1(kx,lx)=b1(kx,lx)*(c7-vtot)/c7
      bbp(kx,lx)=bbp(kx,lx)*(c7-vtot)/c7
      wnd(kx,lx)=wnd(kx,lx)*(c7-vtot)/c7
c
c ** contaminant loop **************************************************
      do 210 k=1,inz
c internal and external contaminant concentrations(per m2) adjusted based
c on timber removed for both above and below ground live standing biomass
      z2(kx,k,lx)=z2(kx,k,lx)-z2(kx,k,lx)*vtot/c7
      rb3(kx,k,lx)=rb3(kx,k,lx)-rb3(kx,k,lx)*vtot/c7
      rbb3(kx,k,lx)=rbb3(kx,k,lx)-rbb3(kx,k,lx)*vtot/c7
      afr(kx,k,lx)=afr(kx,k,lx)*(c7-vtot)/c7
      afs(kx,k,lx)=afs(kx,k,lx)*(c7-vtot)/c7
      wz42(kx,k,lx)=wz42(kx,k,lx)*(c7-vtot)/c7+.2*(vtot/c7)*a2(kx,k,lx)
c
c some estimates(those using variables ending in ..av) are based on integral
c average values of saltation-creep deposition estimates as a function of
c height above the soil surface; for root surface contaminant adsorption
c
      wz43(kx,k,lx)=wz43(kx,k,lx)*(c7-vtot)/c7+.3*(vtot/c7)*a3av(kx,k,lx
     1 )
      bsav=b3av(kx,k,lx)+bb3av(kx,k,lx)+afsav(kx,k,lx)
      z43(kx,k,lx)=z43(kx,k,lx)*(c7-vtot)/c7+.3*(vtot/c7)*bsav
      z42(kx,k,lx)=z42(kx,k,lx)*(c7-vtot)/c7+.2*(vtot/c7)*(b2(kx,k,lx)
     1 +bb2(kx,k,lx)+afr(kx,k,lx))
      a2(kx,k,lx)=a2(kx,k,lx)*(c7-vtot)/c7
      a3(kx,k,lx)=a3(kx,k,lx)*(c7-vtot)/c7
      b2(kx,k,lx)=b2(kx,k,lx)*(c7-vtot)/c7
```

```
      bb2(kx,k,lx)=bb2(kx,k,lx)*(c7-vtot)/c7
      b3(kx,k,lx)=b3(kx,k,lx)*(c7-vtot)/c7
      bb3(kx,k,lx)=bb3(kx,k,lx)*(c7-vtot)/c7
  210 continue
c ** end contaminant loop ***********************************************
  220 continue
c ** end plant loop ****************************************************
      go to 290
c
c continuation of non-fire cutting strategies
c converts 25% of removed timber to dead standing biomass, g/m2
c
  230 d1(jk,lx)=d1(jk,lx)+b1(jk,lx)*.25*vtot*((b1(jk,lx)-wnd(jk,lx))/b1
     1 (jk,lx))/c7
c converts 25% of removed leaf biomass to dead standing leaf biomass, g/m2
c converts all of live root biomass to dead standing root biomass(RD1),g/m2
      wd1(jk,lx)=wd1(jk,lx)+b1(jk,lx)*.25*vtot*wnd(jk,lx)/(b1(jk,lx)*c7)
      rd1(jk,lx)=rd1(jk,lx)+r1(jk,lx)*vtot/c7
c
c ** tree diameter class loop ******************************************
      do 250 k=2,kt
c excludes empty dia. classes or zero timber removals
      if (vtmbr(jk,k,lx).le.0.0.or.vcut(k,lx).le.0.0) go to 250
c adjusts timber volume,ft3, for timber removal(VCUT)
      vtmbr(jk,k,lx)=vtmbr(jk,k,lx)-vcut(k,lx)
c
c ** contaminant loop **************************************************
      do 240 j=1,inz
c adjusts dead standing bole(Z4) and leaf(WZ4) internal contaminant
c content,per m2, by a weighted 25% of contaminant removed by cutting
      vtbmc=.142/c9(jk,lx)
      xsatb=satb(jk,k,j,lx)*vcut(k,lx)/vtbmc
      xz3wz3=z3(jk,j,lx)+wz3(jk,j,lx)
      z4(jk,j,lx)=z4(jk,j,lx)+0.25*xsatb*z3(jk,j,lx)/xz3wz3
      wz4(jk,j,lx)=wz4(jk,j,lx)+0.25*xsatb*wz3(jk,j,lx)/xz3wz3
c adjusts internal contaminant content,per m2, for live above ground
c biomass(Z3), fruits(ZSD), and leaves(WZ3) for timber cutting losses
      z3(jk,j,lx)=z3(jk,j,lx)*(c7-vtot)/c7
      zsd(jk,j,lx)=zsd(jk,j,lx)*(c7-vtot)/c7
      wz3(jk,j,lx)=wz3(jk,j,lx)*(c7-vtot)/c7
  240 continue
c ** end contaminant loop ***********************************************
  250 continue
c ** end diameter class loop ********************************************
c determines number of soil layers below plant(JK) in region(LX)
      iicde=rcode(jk,lx)
c
c ** soil layer loop ***************************************************
      do 270 j=1,iicde
c determines root fraction in each layer based on penetration fraction(ZZF)
c and maximum fraction(ZF); assumes maximum for annuals
      rotfr=zff(jk,j,lx)*zf(jk,j,lx)
      if (hf(jk,lx)-1.0) 272,271,272
  271 rotfr=zf(jk,j,lx)
```

```
c
c ** contaminant loop ***************************************************
 272 do 260 k=1,inz
c converts all internal(Z2),and external root branch(RBB3),and root bole(RB3)
c to dead standing root concentration, per m2, adjusted for timber removal
c (vtot) and total for all tree types(C7) cut
    z4r(jk,j,k,lx)=z4r(jk,j,k,lx)+rotfr*(z2(jk,k,lx)+rbb3(jk,k,lx)+rb3
   1 (jk,k,lx))*vtot/c7
 260 continue
c ** end contaminant loop ***********************************************
 270 continue
c ** end soil layer loop *************************************************
c adjusts above and below ground wood biomass for timber removal
    wnd(jk,lx)=wnd(jk,lx)*(c7-vtot)/c7
    r1(jk,lx)=r1(jk,lx)-r1(jk,lx)*vtot/c7
    b1(jk,lx)=b1(jk,lx)*(c7-vtot)/c7
    bbp(jk,lx)=bbp(jk,lx)*(c7-vtot)/c7
c
c ** contaminant loop ***************************************************
    do 280 k=1,inz
c
c adjusts internal and external contaminant contents, per m2, of roots,
c leaves,stems, and branches for timber removal. a 25% weighted conversion
c of removed contaminant by cutting is returned as dead standing biomass.
c saltation-creep contamination uses integral average values above ground
c surface
c
    z2(jk,k,lx)=z2(jk,k,lx)-z2(jk,k,lx)*vtot/c7
    rb3(jk,k,lx)=rb3(jk,k,lx)-rb3(jk,k,lx)*vtot/c7
    rbb3(jk,k,lx)=rbb3(jk,k,lx)-rbb3(jk,k,lx)*vtot/c7
    wz42(jk,k,lx)=wz42(jk,k,lx)*(c7-vtot)/c7+.25*a2(jk,k,lx)*vtot/c7
    wz43(jk,k,lx)=wz43(jk,k,lx)*(c7-vtot)/c7+.25*a3av(jk,k,lx)*vtot/c7
    bsav=b3av(jk,k,lx)+bb3av(jk,k,lx)+afsav(jk,k,lx)
    z43(jk,k,lx)=z43(jk,k,lx)*(c7-vtot)/c7+.25*bsav*vtot/c7
    z42(jk,k,lx)=z42(jk,k,lx)*(c7-vtot)/c7+.25*(b2(jk,k,lx)+bb2(jk,k
   1 ,lx)+afr(jk,k,lx))*vtot/c7
c leaf resuspension, and saltation-creep contamination adjusted for timber
c removal. (A2,AFR,B2,BB2 = resuspension;(A3,AfS,B3,BB3,BB3AV) = saltation-
c creep contamination; (A2,A3,AFR,AFS) = external leaf contamination
    a2(jk,k,lx)=a2(jk,k,lx)*(c7-vtot)/c7
    a3(jk,k,lx)=a3(jk,k,lx)*(c7-vtot)/c7
    afr(jk,k,lx)=afr(jk,k,lx)*(c7-vtot)/c7
    afs(jk,k,lx)=afs(jk,k,lx)*(c7-vtot)/c7
    b2(jk,k,lx)=b2(jk,k,lx)*(c7-vtot)/c7
    bb2(jk,k,lx)=bb2(jk,k,lx)*(c7-vtot)/c7
    b3(jk,k,lx)=b3(jk,k,lx)*(c7-vtot)/c7
    bb3(jk,k,lx)=bb3(jk,k,lx)*(c7-vtot)/c7
    bb3av(jk,k,lx)=bb3av(jk,k,lx)*(c7-vtot)/c7
 280 continue
c ** end contaminant loop ***********************************************
 290 continue
    return
    end
c
```

14.10. FORMAN (bjm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine forman (jk,lx,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'

      common/pe/ acts(m,n), actsm(m,n), frm(n), totac(m,n), age(i,n),v8
     1 (i,n)
c
c   forman estimates forest structural characteristics for a given tree(jk)
c   for different diameter classes(l)
c
c  for each tree stand simulated forman estimates tree volumes, diam-
c  eters, heights, etc.  input required are the stand parameters
c  relating tree diameter to biomass(V8),crown area to bole diameter
c  (C6),and specific gravity(C9). if yearly output from forman is
c  wanted set RGC=2.0 for that tree, if daily output is wanted set
c  RGC=3.0 for that plant; this will send the proper data to BIOPLT
c  for output the particular days or years the user wants to print
c  TBRM is the  maximum number  of cutting or fire events selected
c  in FORCUT; the day or year when the cutting  or fire  event occurs
c  is input (DATCT).
c
c excludes all but first access for parameter input and initialization
      if (rdidx-0.) 30,31,30
   31 if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5015
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-777777.)5702,5015,5702
 5555 nu=0
 5015 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE FORMAN                    ')
      call wnoust('****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
```

```
      call wnoust('ENTER yes=1, no=0                          ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 777777.                  ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c enters regional output selector(FRM), day and year modulus(DMOD,YMOD)
c
c ** regional loop ******************************************************
      iipx=0
      iijk=0
      do 20 lz=1,ipx
c identifies number of plants simulated in region(lz)
      jpl=apl(lz)
c
c ** plant loop *********************************************************
      do 10 jk=1,jpl
c selects for plants not using plant structure model
      if (rgc(jk,lz)-1.0) 5003,10,5003
c
c reads number of timber managements(TBRM) for each plant specified, diameter
c class intervals(XJ1),tree diameter to dry biomass regression coefficient
c (V8), specific gravity of dry wood(C9), tree diameter to crown diameter
c regression coefficient(C6), and number trees/acre(TNTR, default=0.0)
c
 5003 print *,'ENTER number of timber cuttings for plant',jk,'in'
```

```
      print *,'region',lz
      if(nu.eq.0)read(*,*,err=5003)tbrm(jk,lz)
      if(nu.eq.1)read(7,*)tbrm(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)tbrm(jk,lz)
      if(iu.eq.1)print *,tbrm(jk,lz)
 5004 print *,'ENTER dia. class intervals,inches, for plant',jk,'in'
      print *,'region',lz
      if(nu.eq.0)read(*,*,err=5004)xj1(jk,lz)
      if(nu.eq.1)read(7,*)xj1(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)xj1(jk,lz)
      if(iu.eq.1)print *,xj1(jk,lz)
 5005 print *,'ENTER tree dia.(inches) to dry biomass(kg) regression'
      print *,'coefficient for plant',jk,'in region',lz
      if(nu.eq.0)read(*,*,err=5005)v8(jk,lz)
      if(nu.eq.1)read(7,*)v8(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*)v8(jk,lz)
      if(iu.eq.1)print *,v8(jk,lz)
 5006 print *,'ENTER specific gravity of wood for plant',jk,'in region'
      print *,lz
      if(nu.eq.0)read(*,*,err=5006)c9(jk,lz)
      if(nu.eq.1)read(7,*)c9(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*)c9(jk,lz)
      if(iu.eq.1)print *,c9(jk,lz)
 5007 print *,'ENTER  crown dia.(feet) to stem dia.(inches) regression'
      print *,'coefficient for plant',jk,'in region',lz
      if(nu.eq.0)read(*,*,err=5007)c6(jk,lz)
      if(nu.eq.1)read(7,*)c6(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*)c6(jk,lz)
      if(iu.eq.1)print *,c6(jk,lz)
 5008 print *,'ENTER number of trees/acre for plant',jk,'in region',lz
      print *,'enter 0.0 for natural growth'
      if(nu.eq.0)read(*,*,err=5008)tntr(jk,lz)
      if(nu.eq.1)read(7,*)tntr(jk,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*)tntr(jk,lz)
      if(iu.eq.1)print *,tntr(jk,lz)
c stand volume, ft3, age, years, and tree ring with, inches initialized
```

```
      bbp(jk,lz)=0.0
      age(jk,lz)=0.0
      ringt(jk,lz)=0.0
c selects for timber management(cutting or harvesting strategies)
      if(tbrm(jk,lz).le.0.0)goto10
      iipx=iipx+1
      iijk=iijk+1
      call sleep@(2.0)
      call clear_screen@
      nuf=nu
      nu=nuu
      call forcut (jk,lz,dc,iipx,iijk,rdidx,bpx)
      nu=nuf
      call sleep@(2.0)
      call clear_screen@
   10 continue
c ** end of plant loop ******************************************************
   20 continue
c ** end of regional loop ****************************************************
 5009 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN FORMAN                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for FORMAN?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5009) ans
      if(ans)11,5500,11
   11 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-777777.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto 5015
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
      return
   30 continue
c volume, ft3/acre for tree stand estimated(V4) from live standing biomass
      v4=(.142/c9(jk,lx))*b1(jk,lx)
c competition factor initialized; stand age incremented
      c8(jk,lx)=1.0
```

```
      age(jk,lx)=age(jk,lx)+1.0
c net annual volume increment, cubic feet dry wood
      g4t=v4-bbp(jk,lx)
c fractional volume increase initialized; estimated; and previous volume
c reset
      dgrwth=1.0
      if (bbp(jk,lx).gt.0.0) dgrwth=v4/bbp(jk,lx)
      bbp(jk,lx)=v4
c forest structure parameters initialized during first year
      if (age(jk,lx).gt.1.0) go to 60
c
c tree diameter class loop; KT classifications
c
c ** diameter size loop ***************************************************
      do 50 l=1,kt
c diameter class: volume(VTMBR), number of trees(DTBR), volume of mean tree
c (Z), entering age(AGTB), mean tree age(AATB), and age difference between
c youngest and oldest trees in class(GA) are initialized
      vtmbr(jk,l,lx)=0.0
      dtbr(jk,l,lx)=0.0
      z(l,lx)=0.0
      agtb(jk,l,lx)=0.0
      aatb(jk,l,lx)=0.0
      ga(l,lx)=0.0
c
c ** contaminant loop ****************************************************
      do 40 k=1,inz
c diameter class: contaminant concentration, pCi/g,...g/g..(SATB), areal
c concentration, pCi/m2...g/m2..(VA), and areal concentration of previous
c year
      satb(jk,l,k,lx)=0.0
      va(l,k,lx)=0.0
      zz3(jk,k,lx)=0.0
   40 continue
c ** contaminant loop ****************************************************
   50 continue
c ** end tree diameter loop ***********************************************
   60 continue
c
c  estimates the first diameter class (seedlings). also estimates the
c  number of trees(jk)/acre and competition factors for the first diameter
c  class.  the volume of seedlings decreases with stand age.
c
c seedling volume,cf, (first diameter class)
      vtmbr(jk,1,lx)=v4*exp(-.0037*v4)*(1.0-ccvv(jk,lx))
c volume of mean first diameter class
      z(1,lx)=((.035/c9(jk,lx))*v8(jk,lx)*(2.54*2.*xj1(jk,lx)/2.)**2)/2.
c selects for uneven aged tree stand
      if (tntr(jk,lx)) 71,70,71
c even aged tree stand; returns if volume(V4) insufficient for number of
c trees in stand(TNTR)
   71 if (v4.lt.tntr(jk,lx)*z(1,lx)) return
c estimates current volume of first diameter class, cubic feet; volume
c reduction based on square of ratio (volume of 1st class/stand volume)
```

```
      vtmbr(jk,1,lx)=tntr(jk,lx)*z(1,lx)
      vtmbr(jk,1,lx)=vtmbr(jk,1,lx)*(tntr(jk,lx)*z(1,lx)/v4)**2
c number of trees in diameter class esstimated
   70 dtbr(jk,1,lx)=vtmbr(jk,1,lx)/z(1,lx)
c competiton factor of 1st diameter class estimated based on bole diameter to
c crown diameter regression coefficient, number of trees in class, and
c reduction constant(0.2212, tuned value)
      xx(1,lx)=0.2212*(c6(jk,lx)**2)*dtbr(jk,1,lx)
c initializes logarithmic sum of diameter class ages
      agsum=0.0
c
c ** contaminant loop ************************************************
      do 80 k=1,inz
c initializes total contaminant content in specific stand of region(lx)
      actsm(k,lx)=0.0
   80 continue
c ** end contaminant loop ********************************************
c initializes stand: number of trees(TREES),mean height,ft(HTAV), and bole
c diameter(BMN)
      trees(jk,lx)=0.0
      htav(jk,lx)=0.0
      bmn(jk,lx)=0.0
c initializes mean crown cover fraction(COVAV), bole diameter(BAAV), height
c (HAAV), height sum of squares(HTSQ), crown cover fraction sum of squares
c (CVSQ), and bole diameter sum of squares(BSQ)
      covav=0.0
      baav=0.0
      haav=0.0
      htsq=0.0
      cvsq=0.0
      bsq=0.0
c competition factor sum initialized(C8); basal area cover, m2, initialized
      c8(jk,lx)=xx(1,lx)
      cvtr=0.0
c
c ** tree diameter class loop ****************************************
      do 120 nn=2,kt
      an=nn
c tree crown cover, m2(COV), tree basal area, m2(BAR), mean tree height(HGT),
c ft, and competition factor initialized
      cov(nn,lx)=0.0
      bar(nn,lx)=0.0
      hgt(nn,lx)=0.0
      xx(nn,lx)=0.0
c excludes empty diameter classes
      if (vtmbr(jk,nn,lx).le.0.0) go to 110
c diameter class calculated
      an=2.0*an-1.0
      an=an*xj1(jk,lx)/2.
c competition factor estimated based crown to bole regression coeff.(C6),
c bole diameter(AN), numberr of trees in class(DTBR), growth factor(GRWFAC)
c and diameter class increment(XJ1)
      xx(nn,lx)=(c6(jk,lx)*an**2)*dtbr(jk,nn,lx)
      grwfac=(1.-exp(-.50*an*2./xj1(jk,lx)))
```

```
      xx(nn,lx)=xx(nn,lx)*grwfac
c crown cover,m2, of individual tree in class
      cov(nn,lx)=.0929*3.1416*((an*c6(jk,lx))/2.)**2
c mean tree height,ft, of individual tree in class
      hgt(nn,lx)=(307.*v8(jk,lx)/c6(jk,lx))*(an/24.)**0.9
      hgtt=hgt(nn,lx)
c tree diameter,inches
      bar(nn,lx)=an
c total crown cover,m2, all trees in class
      cov(nn,lx)=dtbr(jk,nn,lx)*cov(nn,lx)
c total logarithmic height sum of trees in class
      hgt(nn,lx)=dtbr(jk,nn,lx)*alog(hgt(nn,lx))
c total logarithmic diameter sum of trees in class
      bar(nn,lx)=dtbr(jk,nn,lx)*alog(bar(nn,lx))
c total number of trees/acre in stand
      trees(jk,lx)=trees(jk,lx)+dtbr(jk,nn,lx)
c crown cover sum of squares
      covsq(nn,lx)=dtbr(jk,nn,lx)*(cov(nn,lx)/dtbr(jk,nn,lx))**2
c logarithmic height sum of squares
      hgtsq(nn,lx)=dtbr(jk,nn,lx)*(hgt(nn,lx)/dtbr(jk,nn,lx))**2
c logarithmic bole diameter sum of squares
      barsq(nn,lx)=dtbr(jk,nn,lx)*(bar(nn,lx)/dtbr(jk,nn,lx))**2
c excludes empty age diameter classes
      if (aatb(jk,nn,lx).le.0.0) go to 90
c logarithmic mean age sum of squares
      agsum=agsum+dtbr(jk,nn,lx)*alog(aatb(jk,nn,lx))
   90 continue
c
c ** contaminant loop ***************************************************
      do 100 k=1,inz
c bounds concentration of contaminant
      if (satb(jk,nn,k,lx).le.0.0) satb(jk,nn,k,lx)=0.0
c contaminant concentration sum over entire stand
      actsm(k,lx)=actsm(k,lx)+dtbr(jk,nn,lx)*satb(jk,nn,k,lx)
  100 continue
c ** end contaminant loop ***********************************************
c total stand sum: crown cover(COVAV),log diameter(BAAV),log height(HAAV)
c log squares height(HTSQ), cover squares(CVSQ),and log squares bole
c diameter(BSQ)
      covav=covav+cov(nn,lx)
      baav=baav+bar(nn,lx)
      haav=haav+hgt(nn,lx)
      htsq=htsq+hgtsq(nn,lx)
      cvsq=cvsq+covsq(nn,lx)
      bsq=bsq+barsq(nn,lx)
c tree height,ft.,for dia. class(nn), mean crown cover(m2) of class(nn), dia.
c of tree(in.) in class(nn)
      hgt(nn,lx)=hgtt
      cov(nn,lx)=cov(nn,lx)/dtbr(jk,nn,lx)
      bar(nn,lx)=an
c total stand basal(bole) area, ft2
      cvtr=cvtr+dtbr(jk,nn,lx)*3.1416*(bar(nn,lx)/24.)**2
c sum of competition factors
  110 c8(jk,lx)=c8(jk,lx)+xx(nn,lx)
```

```
 120 continue
c ** end diameter class loop ********************************************
c excludes null trees/acre case
    if (trees(jk,lx).le.0.0) go to 140
c degrees freedom for total stand;boundary
    df=trees(jk,lx)-1.0
    if (df.lt.1.0) df=1.0
c stand geometric variance: bole diameter(BSD),height(HTSD),variance crown
c cover(cvsd)
    bsd=(bsq-baav**2/trees(jk,lx))/df
    htsd=(htsq-haav**2/trees(jk,lx))/df
    cvsd=(cvsq-covav**2/trees(jk,lx))/df
    if(cvsd.le.0.0)cvsd=0.0
c geometric variance coverted to normal variance; converted to normal stnd.
c deviation; conditional; stnd. dev. crown dia.,m2, estimated(CVSD)
    bsd=exp(bsd)
    htsd=exp(htsd)
    if (bsd.gt.0.0) bsd=sqrt(bsd)
    if (htsd.gt.0.0) htsd=sqrt(htsd)
    if (cvsd.gt.0.0) cvsd=sqrt(cvsd)
c mean stand value: crown cover,m2(CVMN),height,ft(HTAV)
    cvmn=covav/trees(jk,lx)
    bmn(jk,lx)=exp(baav/trees(jk,lx))
    htav(jk,lx)=exp(haav/trees(jk,lx))
c fraction crown cover
    covav=cvmn*trees(jk,lx)/4047.
c stand mean tree height,ft, using geometric mean
    batr=(307.*v8(jk,lx)/c6(jk,lx))*(bmn(jk,lx)/24.)**0.9
c geometric mean age of stand
    agave=exp(agsum/trees(jk,lx))
c
c ** contaminant loop ***********************************************
    do 130 k=1,inz
c stand mean contaminant concentration ,pCi/g...g/g..
    actav(jk,k,lx)=actsm(k,lx)/trees(jk,lx)
 130 continue
c ** end contaminant loop ********************************************
c crown cover fraction bounded
    if (covav.gt.1.0) covav=1.0
 140 continue
c crown cover fraction for use in other subroutines
    ccvv(jk,lx)=covav
c
c ** contaminant loop ***********************************************
    do 150 k=1,inz
c total stand internal live above ground contaminant content, pCi..g../m2
c ;bounded
    totsa(k,lx)=z3(jk,k,lx)+wz3(jk,k,lx)+zsd(jk,k,lx)
    if (totsa(k,lx).le.0.0) totsa(k,lx)=0.0
c annual contaminant accumulation in bole and leaves, pCi..g../m2
    acts(k,lx)=z3(jk,k,lx)+wz3(jk,k,lx)-zz3(jk,k,lx)
 150 continue
c ** end contaminant loop ********************************************
c
```

```
c ** diameter class loop ************************************************
      do 170 l=2,kt
c excludes empty dia. classes
      if (xx(l-1,lx).le.0.0.and.xx(l,lx).le.0.0) go to 170
c calculates diameter of class,inches(AL9);calc. 1/2 diameter incr.,in(DD)
      al=l
      al9=al*xj1(jk,lx)-xj1(jk,lx)/2.
      dd=xj1(jk,lx)/2.
c estimates mean tree dia.,in.(Z) for diameter class(l)
      z(l,lx)=(.035/c9(jk,lx))*(v8(jk,lx))*((al9-dd)*2.54)**2
      z(l,lx)=(z(l,lx)+(.035/c9(jk,lx))*(v8(jk,lx))*(((al9+dd)*2.54)**2)
     1 )/2.
c excludes null annual stand volume growth increments
      if (g4t.le.0.0) go to 170
c allocates volume to previous and current dia. class based on competiton
c factor ratio with sum of stand competition factors
      g3=g4t*(xx(l-1,lx)/c8(jk,lx))
      g3o=g4t*(xx(l,lx)/c8(jk,lx))
c updates wood volume in dia. class;gains from prev. class;losses to next
c diameter class; boundary at 1/100 individual tree volume(Z)
      vtmbr(jk,l,lx)=vtmbr(jk,l,lx)+g3-g3o
      if (vtmbr(jk,l,lx).gt.0.01*z(l,lx)) go to 170
c re-initializes structural parameters: current volume(VTMBR), previous
c volume(VVTM),number of trees for specific diameter class(l)
      vtmbr(jk,l,lx)=0.0
      vvtm(jk,l,lx)=0.0
      dtbr(jk,l,lx)=0.0
c
c contaminant loop ******************************************************
      do 160 k=1,inz
c re-initialize dia. class areal contaminant conc.,pCi...g../m2(VA),
c and class contaminant conc.,pCi...g../g(SATB)
      va(l,k,lx)=0.0
      satb(jk,l,k,lx)=0.0
  160 continue
c ** end contaminant loop ***********************************************
c re-initialize dia. class: age-span,yr(GA), minimum age,yr(AGTB), and
c average age,yr(AATB)
      ga(l,lx)=0.0
      agtb(jk,l,lx)=0.0
      aatb(jk,l,lx)=0.0
  170 continue
c ** end diameter class loop ********************************************
c initialize stand volume sum(V6), and tree sum(C7G)
      v6=0.0
      c7g=0.0
c
c ** diameter class loop ************************************************
      do 180 k=1,kt
c excludes empty diameter classes
      if (vtmbr(jk,k,lx).le.0.0) go to 180
c annual volume loss reduces class dia. volume using fractional growth factor
c (DGRWTH)
      if (g4t.lt.0.0) vtmbr(jk,k,lx)=vtmbr(jk,k,lx)*dgrwth
```

```
c class dia. volume and number of trees updated
      v6=v6+vtmbr(jk,k,lx)
      c7g=c7g+vtmbr(jk,k,lx)/z(k,lx)
  180 continue
c ** end diameter class loop ********************************************
c excludes uneven aged stands
      if (tntr(jk,lx)) 211,210,211
c even aged stand; excludes null wood volume
  211 if (v6.le.0.0) go to 210
c
c ** diameter class loop ************************************************
      do 200 k=1,kt
c excludes number of trees equal to specified(TNTR)
      if (c7g-tntr(jk,lx)) 212,210,212
c tree diameter index
  212 kk=kt-k
c excludes tree count greater than TNTR
      if (c7g.gt.tntr(jk,lx)) go to 190
c excludes first dia. class; excludes empty class dia.
      if (kk-1.le.0.0) go to 200
      if (vtmbr(jk,kk,lx)) 201,200,201
c removes higher and second higheet diameter classes;reduces stand tree
c count; converts removed volume to second highest dia.class
  201 c7g=c7g-vtmbr(jk,kk,lx)*(v4/v6)/z(kk,lx)-vtmbr(jk,kk-1,lx)*(v4/v6)
     1 /z(kk-1,lx)
      vtmbr(jk,kk-1,lx)=vtmbr(jk,kk-1,lx)+vtmbr(jk,kk,lx)
c increments number of trees;sets higher diameter class volume to 0.0
      c7g=c7g+vtmbr(jk,kk-1,lx)*(v4/v6)/z(kk-1,lx)
      vtmbr(jk,kk,lx)=0.0
c loops again if tree count(C7G) less than specified(TNTR)
      if (c7g.lt.tntr(jk,lx)) go to 200
c estimates tree count in second highest layer
      dtkm1=vtmbr(jk,kk-1,lx)*(v4/v6)/z(kk-1,lx)
c estimates new tree count
      c78=tntr(jk,lx)-c7g+vtmbr(jk,kk-1,lx)*(v4/v6)/z(kk-1,lx)
c estimates excess tree volume in lower class;adds excess to upper layer;
c bounds
      res=(c78-dtkm1)/((v4/v6)*(1./z(kk,lx)-1./z(kk-1,lx)))
      if (res.le.0.0) res=0.0
      vtmbr(jk,kk,lx)=res
      if (vtmbr(jk,kk,lx).le.0.0) vtmbr(jk,kk,lx)=0.0
c updates lower layer volume;bounds
      vtmbr(jk,kk-1,lx)=vtmbr(jk,kk-1,lx)-res
      if (vtmbr(jk,kk-1,lx).le.0.0) vtmbr(jk,kk-1,lx)=0.0
      go to 210
c even aged stands;excludes dia. class above kt;excludes null dia. classes
  190 if (k+1.gt.kt) go to 200
      if (vtmbr(jk,k,lx)) 202,200,202
c removes current and higher diameter class tree counts and volumes; adds
c removed volume to higher diameter class;estimates new tree count; removes
c current dia. class
  202 c7g=c7g-vtmbr(jk,k,lx)*(v4/v6)/z(k,lx)-vtmbr(jk,k+1,lx)*(v4/v6)/z
     1 (k+1,lx)
      vtmbr(jk,k+1,lx)=vtmbr(jk,k+1,lx)+vtmbr(jk,k,lx)
```

```
      c7g=c7g+vtmbr(jk,k+1,lx)*(v4/v6)/z(k+1,lx)
      vtmbr(jk,k,lx)=0.0
c loops again if tree count exceeds or equals specified count(TNTR)
      if (c7g.ge.tntr(jk,lx)) go to 200
c estimates residual number of trees in stand;bounds;adds to current dia.
c class;updates higher dia. class;bounds
      c78=tntr(jk,lx)-c7g+vtmbr(jk,k+1,lx)*(v4/v6)/z(k+1,lx)
      dtkp1=vtmbr(jk,k+1,lx)*(v4/v6)/z(k+1,lx)
      res=(c78-dtkp1)/((v4/v6)*(1./z(k,lx)-1./z(k+1,lx)))
      if (res.le.0.0) res=0.0
      vtmbr(jk,k,lx)=res
      if (vtmbr(jk,k,lx).le.0.0) vtmbr(jk,k,lx)=0.0
      vtmbr(jk,k+1,lx)=vtmbr(jk,k+1,lx)-res
      if (vtmbr(jk,k+1,lx).le.0.0) vtmbr(jk,k+1,lx)=0.0
      go to 210
  200 continue
c ** end diameter class loop ********************************************
  210 c7=0.0
      trees(jk,lx)=0.0
c
c  estimates tree ring daily growth increment for each diameter class
c  using  the diameter class mid diameter (al9).
c
c ** diameter class loop ************************************************
      do 240 l=1,kt
c bounds dia. class vol.(VTMBR);excludes null stand vol.;excludes null dia.
c class volume
      if (vtmbr(jk,l,lx).le.0.0) vtmbr(jk,l,lx)=0.0
      if (v6.le.0.0) go to 240
      if (vtmbr(jk,l,lx).le.0.0) go to 240
c estimates mid-diameter for dia. class(l)
      al9=l
      al9=(al9*xj1(jk,lx)-xj1(jk,lx)/2.0)/2.0
c sets previous vol;estimates current value;current tree count in dia. class
      vvtm(jk,l,lx)=vtmbr(jk,l,lx)
      vtmbr(jk,l,lx)=vtmbr(jk,l,lx)*v4/v6
      dtbr(jk,l,lx)=vtmbr(jk,l,lx)/z(l,lx)
c
c ** contaminant loop ***************************************************
      do 220 k=1,inz
c initializes dia. class areal conc.,per m2;mass conc,per g;excludes null
c dia. class volume
      if (vtmbr(jk,l,lx).le.0.0) va(l,k,lx)=0.0
      if (vtmbr(jk,l,lx).le.0.0) satb(jk,l,k,lx)=0.0
      if (vtmbr(jk,l,lx).le.0.0) go to 240
  220 continue
c ** end contaminant loop ***********************************************
c increments stand volume and tree count
      c7=c7+vtmbr(jk,l,lx)
      trees(jk,lx)=trees(jk,lx)+dtbr(jk,l,lx)
c estimates dia. class volume increment,ft3(VDTBM);bounds prev. volume
      vdtbm=vtmbr(jk,l,lx)-vvtm(jk,l,lx)
      if (vvtm(jk,l,lx).le.0.0) vvtm(jk,l,lx)=vtmbr(jk,l,lx)
      if (vvtm(jk,l,lx).gt.vtmbr(jk,l,lx)) vvtm(jk,l,lx)=vtmbr(jk,l,lx)
```

```
c estimates dia. class total thickness,in(RADIWT);estimates thickness
c increment,in.,(RINGT) using prev. thickness;bounds
      radiwt=al9*sqrt(vtmbr(jk,l,lx)/vvtm(jk,l,lx))*dtbr(jk,l,lx)
      ringt(jk,lx)=ringt(jk,lx)+radiwt-al9*dtbr(jk,l,lx)
      if(ringt(jk,lx).le.0.0)ringt(jk,lx)=0.0
c
c ** contaminant loop ************************************************
      do 230 k=1,inz
c estimates areal(VA, per m2) and internal(SATB, per gdwt) contaminant
c concentration for each diameter group of plant(JK);bounds
      va(l,k,lx)=vtmbr(jk,l,lx)*satb(jk,l,k,lx)/(.142/c9(jk,lx))
      va(l,k,lx)=va(l,k,lx)+abs(g4t)*acts(k,lx)*xx(l,lx)/c8(jk,lx)
      satb(jk,l,k,lx)=va(l,k,lx)/(vtmbr(jk,l,lx)/(.142/c9(jk,lx)))
      va(l,k,lx)=va(l,k,lx)+(vdtbm/(.142/c9(jk,lx)))*satb(jk,l,k,lx)
      if (va(l,k,lx).lt.0.0) va(l,k,lx)=0.0
  230 continue
c ** end contaminant loop ********************************************
  240 continue
c ** end diameter class loop ******************************************
c estimates mean tree ring growth, inches, for entire stand
      if (trees(jk,lx).gt.0.0) ringt(jk,lx)=ringt(jk,lx)/trees(jk,lx)
c
c ** contaminant loop ************************************************
      do 250 k=1,inz
c re-initializes total contaminant inventory in stand to first diameter
c group;bounds
      totac(k,lx)=satb(jk,1,k,lx)*vtmbr(jk,1,lx)/(.142/c9(jk,lx))
      if (totsa(k,lx)-1.0) 252,251,252
  251 totac(k,lx)=1.0
  252 if (totac(k,lx).le.0.0) totac(k,lx)=1.0
  250 continue
c ** end contaminant loop ********************************************
c
c ** diameter class loop ************************************************
      do 290 l=1,kt
c initial age of trees, years, entering diameter class(l)
      if (agtb(jk,l,lx).gt.0.0) go to 260
      if (vtmbr(jk,l+1,lx).gt.0.0) agtb(jk,l,lx)=ai1
c excludes 1st dia. class and empty dia. classes
  260 if (l.eq.1) go to 290
      if (agtb(jk,l-1,lx)) 271,270,271
c estimates mean age of 2nd dia. class ; diff. btw. oldest and youngest
  271 if (l.eq.2) aatb(jk,l-1,lx)=agtb(jk,l-1,lx)/2.0
      if (l.eq.2) ga(l-1,lx)=agtb(jk,l-1,lx)
      if (l.eq.2) go to 270
c estimates previous dia. class mean age from current and previous entering
c means(AGTB);sets to lower mean if current dia. class has not been set
      aatb(jk,l-1,lx)=(agtb(jk,l,lx)+agtb(jk,l-1,lx))/2.0
      if (agtb(jk,l,lx)) 273,272,273
  272 if (l.gt.2) aatb(jk,l-1,lx)=2.*aatb(jk,l-1,lx)
c estimates diff. in age betw. youngest and oldest tree in previous dia.
c class; bounds to entering age if current entering age equals 0.0
  273 ga(l-1,lx)=agtb(jk,l,lx)-agtb(jk,l-1,lx)
      if (ga(l-1,lx).lt.0.0) ga(l-1,lx)=agtb(jk,l-1,lx)
```

```
 270 continue
c excludes empty dia. classes
    if (vtmbr(jk,l,lx).le.0.0) go to 290
c
c ** contaminant loop ***********************************************
    do 280 k=1,inz
c estimates total stand contaminant content
    totac(k,lx)=totac(k,lx)+satb(jk,l,k,lx)*vtmbr(jk,l,lx)/(.142/c9(jk
   1 ,lx))
 280 continue
c ** end contaminant loop *******************************************
 290 continue
c ** end diameter class loop *****************************************
c
c ** contaminant loop ************************************************
    do 300 k=1,inz
c estimates contaminant concentration (per gram dwt) of 1st dia. class
    if (totac(k,lx)-1.0) 302,301,302
 301 totsa(k,lx)=1.0
 302 satb(jk,1,k,lx)=satb(jk,1,k,lx)*totsa(k,lx)/totac(k,lx)
    if (satb(jk,1,k,lx).lt.0.0) satb(jk,1,k,lx)=0.0
 300 continue
c ** end contaminant loop *******************************************
c
c ** diameter class loop **********************************************
    do 330 l=1,kt
c
c ** contaminant loop ************************************************
    do 320 k=1,inz
c estimates total contaminant concentration(per m2) in live above ground
c biomass
    zz3(jk,k,lx)=z3(jk,k,lx)+wz3(jk,k,lx)
c estimates contaminant concentration (per gram dwt) in all diameter
c classes adjusted total stand activity
    satb(jk,l,k,lx)=satb(jk,l,k,lx)*totsa(k,lx)/totac(k,lx)
 320 continue
c ** end contaminant loop *******************************************
 330 continue
c ** end diameter class loop *****************************************
c excludes trees not managed( harvested)
    if (tbrm(jk,lx)) 332,331,332
 331 return
c call forest management subroutine  FORCUT
 332 call forcut (jk,lx,dc,iipx,iijk,rdidx,bpx)
    return
c
    end
```

14.11. GEOFLX (bkm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine geoflx (lx,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c       watershed,sediment transport,and aquifer model
c
      include 'main.inc'
c
      common/pf/ chlat(30,20), chne(30,20), entchn(30), regent(n),regx(n
     1 ,10), rfdcf(n), rfycf(n), rfylx(n), rmchn(30), rnylx(n), rycf(n),
     2 sarea(30), tnylx(n), tnytn(n), vtrndp(30)
c
c excludes all but first access to subroutine
      if (rdidx) 140,5705,140
c parameters for runoff and sediment yield are entered(Tr,C1,C2,C5,K
c C,P,LS). if entered as 0.0,then default values will be used(see
c appropriate section of code). constants correspond to SPUR .
c
 5705 nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-888888.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','yellow')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('*****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE GEOFLX                    ')
      call wnoust('*****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
```

```
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 888888.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c number of channels
      ich=chnnu
c channels are numbered after regions
      isch=ipx+1
c total number of regions + channels
      icht=ich+ipx
      icnumb=icht-ipx
c     read in appropriate constants for channels;default=0.0
c hydraulic conductivity, inches/hour
 5001 print *,'Enter hydraulic conductivity, inches/hour, for'
      print *,icnumb,'channels; default=0.0 sets to 3"/hr'
      if(nu.eq.0)read(*,*,err=5001) (akond(ic),ic=isch,icht)
      if(nu.eq.1)read(7,*) (akond(ic),ic=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write(6,*) (akond(ic),ic=isch,icht)
      if(iu.eq.1)print *, (akond(ic),ic=isch,icht)
c constants for determining flow duration, c1hr*mi2**c2hr, hours
 5002 print *,'ENTER flow duration constant,c1hr, for the'
      print *,'relationship: dur(hrs)=c1hr*mi2**c2hr'
      print *,'for ',icnumb,'channels;default(0.0) sets to 2.53'
      if(nu.eq.0)read(*,*,err=5002) (c1hr(ic),ic=isch,icht)
      if(nu.eq.1)read(7,*) (c1hr(ic),ic=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*) (c1hr(ic),ic=isch,icht)
```

```
      if(iu.eq.1)print *, (c1hr(ic),ic=isch,icht)
 5003 print *,'ENTER flow duration exponent,c2hr, for'
      print *,icnumb,' channels;default(0.0) sets to 0.2'
      if(nu.eq.0)read(*,*,err=5003) (c2hr(ic),ic=isch,icht)
      if(nu.eq.1)read(7,*) (c2hr(ic),ic=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*) (c2hr(ic),ic=isch,icht)
      if(iu.eq.1)print *, (c2hr(ic),ic=isch,icht)
c constant for determining peak flow in channels, ft3/sec
 5004 print *,'ENTER constant, c5hr, for determining peak flow'
      print *,'in the relationship peak flow(ft3/s)=c1hr*v/dur'
      print *,'for ',icnumb,'channels; v= daily runoff(inches)'
      print *,'default(0.0) sets to 4.82'
      if(nu.eq.0)read(*,*,err=5004) (c5hr(ic),ic=isch,icht)
      if(nu.eq.1)read(7,*) (c5hr(ic),ic=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*) (c5hr(ic),ic=isch,icht)
      if(iu.eq.1)print *, (c5hr(ic),ic=isch,icht)
c     read in sediment diameters and fractions(dsz,mm;dsf,fraction)
c
c ** channel loop ***************************************************
      do 10 ic=isch,icht
      icm=ic
 5005 print *,'ENTER 7 diameter classes,mm, for channel ',icm
      print *,'lower diameter=0.062 mm; maxiumum up to 4.0 mm'
      print *,'8th entree =  median diameter,mm, of aluvium'
      if(nu.eq.0)read(*,*,err=5005) (dsz(iz,ic),iz=1,8)
      if(nu.eq.1)read(7,*) (dsz(iz,ic),iz=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*) (dsz(iz,ic),iz=1,8)
      if(iu.eq.1)print *, (dsz(iz,ic),iz=1,8)
 5006 print *,'ENTER 7 mass fractions for each diameter class'
      print *,'in channel ',icm,'entered above'
      if(nu.eq.0)read(*,*,err=5006) (dsf(iz,ic),iz=1,7)
      if(nu.eq.1)read(7,*) (dsf(iz,ic),iz=1,7)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*) (dsf(iz,ic),iz=1,7)
      if(iu.eq.1)print *, (dsf(iz,ic),iz=1,7)
c dzas=act/gram for each diameter class in channel soil
 5007 print *,'ENTER 7 concentrations, activity/g, ppm..etc for'
      print *,'each diameter class entered  above in channel ',icm
      print *,'if interfacing with WATFLX soil concentrations,'
      print *,'use consistent units:unless specifed otherwise,'
      print *,'ppm in soils are converted to grams/gram'
      if(nu.eq.0)read(*,*,err=5007) (dzas(iz,ic),iz=1,7)
      if(nu.eq.1)read(7,*) (dzas(iz,ic),iz=1,7)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*) (dzas(iz,ic),iz=1,7)
      if(iu.eq.1)print *, (dzas(iz,ic),iz=1,7)
c default values for channel parameters
      if (akond(ic)) 12,11,12
   11 akond(ic)=3.0
   12 if (c1hr(ic)) 14,13,14
   13 c1hr(ic)=2.53
   14 if (c2hr(ic)) 16,15,16
   15 c2hr(ic)=0.20
   16 if (c5hr(ic)) 10,17,10
   17 c5hr(ic)=4.82
   10 continue
c ** end channel loop ******************************************************
c for each channel specify number of channels entering
 5008 print *,'ENTER number of channels tributary to each of'
      print *,icnumb,'channels; 0.0 if none'
      if(nu.eq.0)read(*,*,err=5008) (entchn(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (entchn(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*) (entchn(lc),lc=isch,icht)
      if(iu.eq.1)print *, (entchn(lc),lc=isch,icht)
c for each channel specify number of lateral flows entering channel
 5009 print *,'ENTER number of lateral flow surfaces attached to'
      print *,'each of ',icnumb,' channels;0.0 if none'
      if(nu.eq.0)read(*,*,err=5009) (rnchn(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (rnchn(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,*) (rnchn(lc),lc=isch,icht)
      if(iu.eq.1)print *, (rnchn(lc),lc=isch,icht)
c
c ** channel loop ******************************************************
      do 20 lch=isch,icht
      lchm=lch-1
c initialize channel ice depth,mm, and mean temperature
      aice(lch)=0.0
      tchan(lch)=0.0
c designates number of tributaries to channel(lch)
      ichx=entchn(lch)
      if (ichx.eq.0) go to 20
c read in specific channels entering channel(lch)
 5010 print *,'ENTER ',ichx,' channels (by number 1,2,..etc)'
      print *,'tributary to channel',lchm
      if(nu.eq.0)read(*,*,err=5010) (chne(lch,lc),lc=1,ichx)
      if(nu.eq.1)read(7,*) (chne(lch,lc),lc=1,ichx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
```

```
      write(6,*) (chne(lch,lc),lc=1,ichx)
      if(iu.eq.1)print *, (chne(lch,lc),lc=1,ichx)
   20 continue
c ** end channel loop ***************************************************
c
c ** channel loop ****************************************************
      do 30 lch=isch,icht
      lchm=lch-1
c number of lateral areas(regions) draining into channel(lch)
      ilatx=rnchn(lch)
      if (ilatx.eq.0) go to 30
c read in specific areas(laterals) entering channel
 5011 print *,'ENTER ',ilatx,' lateral areas (by number)'
      print *,'bordering channel ',lchm
      if(nu.eq.0)read(*,*,err=5011) (chlat(lch,lc),lc=1,ilatx)
      if(nu.eq.1)read(7,*) (chlat(lch,lc),lc=1,ilatx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*) (chlat(lch,lc),lc=1,ilatx)
      if(iu.eq.1)print *, (chlat(lch,lc),lc=1,ilatx)
   30 continue
c ** end channel loop ***************************************************
c specify which region is above aquifer for given channel.set at 0.0
c if none exists
 5012 print *,'ENTER region number above each of ',icnumb,'channels'
      print *,'which contains an open aquifer;otherwise set to 0.0'
      if(nu.eq.0)read(*,*,err=5012) (chnaq(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (chnaq(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5012) ans
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*) (chnaq(lc),lc=isch,icht)
      if(iu.eq.1)print *, (chnaq(lc),lc=isch,icht)
c
c read in aquifer dimensions,ft,tch,bch,htp,hbt, corresponding to top
c and bottom width,top height,and bottom height ,respectively.
c tbch and bbch correspond to bottom aquifer top and bottom dimensions
c the channel segment aquifers are assumed to be regular or irregular
c trapezoidal volumes.
c
      if (aqugat) 5013,40,5013
 5013 print *,'ENTER top posterior channel widths,ft, for each of'
      print *,icnumb,' channels; enter 0.0 if no aquifer present'
      if(nu.eq.0)read(*,*,err=5013) (tch(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (tch(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5013) ans
      if(ans.gt.0.and.nu.eq.0)goto5013
      write(6,*) (tch(lc),lc=isch,icht)
      if(iu.eq.1)print *, (tch(lc),lc=isch,icht)
 5014 print *,'ENTER bottom posterior channel widths,ft, for each of'
      print *,icnumb,' channels; enter 0.0 if no aquifer present'
      if(nu.eq.0)read(*,*,err=5014) (bch(lc),lc=isch,icht)
```

```
      if(nu.eq.1)read(7,*) (bch(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*) (bch(lc),lc=isch,icht)
      if(iu.eq.1)print *, (bch(lc),lc=isch,icht)
 5015 print *,'ENTER posterior channel heights, ft, for each of'
      print *,icnumb,' channels; enter 0.0 if no aquifer present'
      if(nu.eq.0)read(*,*,err=5015) (htp(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (htp(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*) (htp(lc),lc=isch,icht)
      if(iu.eq.1)print *, (htp(lc),lc=isch,icht)
 5016 print *,'ENTER anterior channel heights,ft, as above'
      if(nu.eq.0)read(*,*,err=5016) (hbt(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (hbt(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*) (hbt(lc),lc=isch,icht)
      if(iu.eq.1)print *, (hbt(lc),lc=isch,icht)
 5017 print *,'ENTER top anterior channel widths,ft, as above'
      if(nu.eq.0)read(*,*,err=5017) (tbch(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (tbch(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,*) (tbch(lc),lc=isch,icht)
      if(iu.eq.1)print *, (tbch(lc),lc=isch,icht)
 5018 print *,'ENTER bottom anterior channel widths,ft, as above'
      if(nu.eq.0)read(*,*,err=5018) (bbch(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (bbch(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
      write(6,*) (bbch(lc),lc=isch,icht)
      if(iu.eq.1)print *, (bbch(lc),lc=isch,icht)
c read in initial volume of aquifer, cubic feet
 5019 print *,'ENTER initial volume,cubic feet, of each of'
      print *,icnumb,' channels; enter 0.0 if no aquifer present'
      if(nu.eq.0)read(*,*,err=5019) (vaqf(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (vaqf(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
      write(6,*) (vaqf(lc),lc=isch,icht)
      if(iu.eq.1)print *, (vaqf(lc),lc=isch,icht)
c for each region(lz),read in number of regions contributing sediment
c to that region
   40 print *,'ENTER number of regions contributing sediment from'
      print *,'runoff to each of ',ipx,' regions; 0.0 if none'
      if(nu.eq.0)read(*,*,err=40) (regent(lz),lz=1,ipx)
```

```
      if(nu.eq.1)read(7,*)(regent(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=40) ans
      if(ans.gt.0.and.nu.eq.0)goto40
      write(6,*)(regent(lz),lz=1,ipx)
      if(iu.eq.1)print *,(regent(lz),lz=1,ipx)
c
c ** region loop ******************************************************
      do 50 lz=1,ipx
      ireg=regent(lz)
      if (ireg) 5020,50,5020
c read in specific region(s) contributing sediment to region lz
 5020 print *,'ENTER ',ireg,' specific region(s) contributing'
      print *,'sediment from runoff to region ',lz
      if(nu.eq.0)read(*,*,err=5020) (regx(lz,lr),lr=1,ireg)
      if(nu.eq.1)read(7,*) (regx(lz,lr),lr=1,ireg)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
      write(6,*) (regx(lz,lr),lr=1,ireg)
      if(iu.eq.1)print *, (regx(lz,lr),lr=1,ireg)
   50 continue
c ** end region loop ***************************************************
c read in area of each region,acres
 5021 print *,'ENTER area, acres, of each of ',ipx,'regions'
      if(nu.eq.0)read(*,*,err=5021) (rarea(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (rarea(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*) (rarea(lz),lz=1,ipx)
      if(iu.eq.1)print *, (rarea(lz),lz=1,ipx)
c read specific soil dia(mm)sizes for each site: 0-.062,.062-.125,
c .125-.250,.250-.500,.500-1.0,1.0-2.0, 2.0-4.0,where iz=1,2,
c 3...7,lz=number of sections 1,2,...ipx.  dsz(8,lz)=d50(mm).
c dsf(iz,lz)=mass fraction of particles in size category
c
c ** region loop *******************************************************
      do 70 lz=1,ipx
 5022 print *,'ENTER 7 soil diameter sizes,mm,for region ',lz
      print *,'8th entree is median soil diameter,mm'
      print *,'1st diameter category=0.062mm; maximum =4.0 mm'
      if(nu.eq.0)read(*,*,err=5022) (dsz(iz,lz),iz=1,8)
      if(nu.eq.1)read(7,*) (dsz(iz,lz),iz=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5022) ans
      if(ans.gt.0.and.nu.eq.0)goto5022
      write(6,*) (dsz(iz,lz),iz=1,8)
      if(iu.eq.1)print *, (dsz(iz,lz),iz=1,8)
 5023 print *,'ENTER 7 mass fractions for each diameter class'
      print *,'in region ',lz
      if(nu.eq.0)read(*,*,err=5023) (dsf(iz,lz),iz=1,7)
      if(nu.eq.1)read(7,*) (dsf(iz,lz),iz=1,7)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
```

```
      if(nu.eq.0)read(*,*,err=5023) ans
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*) (dsf(iz,lz),iz=1,7)
      if(iu.eq.1)print *, (dsf(iz,lz),iz=1,7)
c dzas= act/gram for each diameter class regional soil
 5024 print *,'ENTER 7 soil concentrations, activity/gram, ppm..'
      print *,'for region ',lz,' Make sure units are consistent'
      print *,'with soil conc. units; ppm in soils converted to'
      print *,'grams/gram unless specified otherwise'
      if(nu.eq.0)read(*,*,err=5024) (dzas(iz,lz),iz=1,7)
      if(nu.eq.1)read(7,*) (dzas(iz,lz),iz=1,7)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*) (dzas(iz,lz),iz=1,7)
      if(iu.eq.1)print *, (dzas(iz,lz),iz=1,7)
c initialize total lbs soil in top 20mm, and specific activity of
      sedmns(lz)=0.0
      sdmnsa(lz)=0.0
c ** particle size loop ***********************************************
      do 60 ls=1,7
c estimate lbs of each diameter size in a given region
c assuming a 20mm thickness of soil at 1.5 bulk density
      dsms(ls,lz)=dsf(ls,lz)*273433.*rarea(lz)*10.
c estimate conc. of contaminant/diameter size in a given region
      dsma(ls,lz)=dzas(ls,lz)*dsms(ls,lz)*454.
c sedmns=total soil (lbs) in a given lateral area(region)
      sedmns(lz)=sedmns(lz)+dsms(ls,lz)
c sdmnsa=total activity of soil in a given lateral area(region)
      sdmnsa(lz)=sdmnsa(lz)+dsma(ls,lz)
   60 continue
c ** end particle size loop *******************************************
   70 continue
c ** end region loop ***************************************************
c enter runoff constants for regions as defined for channels above
 5025 print *,'ENTER flow duration constant, c1hr, for the'
      print *,'relationnship: dur(hrs)=c1hr*mi2**c2hr, for each'
      print *,'of ',ipx,'regions; default(0.0) sets to 2.53'
      if(nu.eq.0)read(*,*,err=5025) (c1hr(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (c1hr(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5025) ans
      if(ans.gt.0.and.nu.eq.0)goto5025
      write(6,*) (c1hr(lz),lz=1,ipx)
      if(iu.eq.1)print *, (c1hr(lz),lz=1,ipx)
 5026 print *,'ENTER flow duration exponent, c2hr, for relationship'
      print *,'above for each of ',ipx,'regions'
      print *,'default(0.0) sets to 0.2'
      if(nu.eq.0)read(*,*,err=5026) (c2hr(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (c2hr(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5026) ans
      if(ans.gt.0.and.nu.eq.0)goto5026
      write(6,*) (c2hr(lz),lz=1,ipx)
```

```
      if(iu.eq.1)print *, (c2hr(lz),lz=1,ipx)
 5027 print *,'ENTER peak flow constant,c5hr,for the relationship'
      print *,'flow(cfps)=c5hr*v/dur, for each of ',ipx,'regions'
      print *,'default(0.0) sets to 4.84'
      if(nu.eq.0)read(*,*,err=5027) (c5hr(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (c5hr(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5027) ans
      if(ans.gt.0.and.nu.eq.0)goto5027
      write(6,*) (c5hr(lz),lz=1,ipx)
      if(iu.eq.1)print *, (c5hr(lz),lz=1,ipx)
c the following constants represent reasonable estimates of(K,C,P,&L
c of the MUSLE equation used in SPUR. site specific
c parameters should be substituted when available.
 5028 print *,'ENTER constant, K, of the MUSCLE eq. for each of'
      print *,ipx,' regions;default(0.0) sets to 0.2'
      if(nu.eq.0)read(*,*,err=5028) (aksoil(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (aksoil(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5028) ans
      if(ans.gt.0.and.nu.eq.0)goto5028
      write(6,*) (aksoil(lz),lz=1,ipx)
      if(iu.eq.1)print *, (aksoil(lz),lz=1,ipx)
 5029 print *,'ENTER constant, C, of the MUSCLE eq. for each of'
      print *,ipx,' regions;default(0.0) sets to 0.1'
      if(nu.eq.0)read(*,*,err=5029) (csoil(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (csoil(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5029) ans
      if(ans.gt.0.and.nu.eq.0)goto5029
      write(6,*) (csoil(lz),lz=1,ipx)
      if(iu.eq.1)print *, (csoil(lz),lz=1,ipx)
 5030 print *,'ENTER constant, P, of MUSCLE eq. for each of'
      print *,ipx,' regions;default(0.0) sets to 1.0'
      if(nu.eq.0)read(*,*,err=5030) (psoil(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (psoil(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5030) ans
      if(ans.gt.0.and.nu.eq.0)goto5030
      write(6,*) (psoil(lz),lz=1,ipx)
      if(iu.eq.1)print *, (psoil(lz),lz=1,ipx)
 5031 print *,'ENTER constant, L, of MUSCLE eq. for each of'
      print *,ipx,' regions;default(0.0) sets to 1.0'
      if(nu.eq.0)read(*,*,err=5031) (alsoil(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (alsoil(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5031) ans
      if(ans.gt.0.and.nu.eq.0)goto5031
      write(6,*) (alsoil(lz),lz=1,ipx)
      if(iu.eq.1)print *, (alsoil(lz),lz=1,ipx)
 5032 print *,'ENTER hydraulic conductivity, inches/hr, for'
      print *,'each of ',ipx,' regions;default(0.0) sets to 2.0'
      if(nu.eq.0)read(*,*,err=5032) (akond(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (akond(lz),lz=1,ipx)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5032) ans
      if(ans.gt.0.and.nu.eq.0)goto5032
      write(6,*) (akond(lz),lz=1,ipx)
      if(iu.eq.1)print *, (akond(lz),lz=1,ipx)
c if values for wchan,schan,rchan: are read in(not 0.0),
c then accompanying indices:schxi,rchxi,wchi,schi,rchi are =1.0,
c otherwise they are equal to 0.0
c slope of region, fraction
 5033 print *,'ENTER region slope(s),fraction, for each of'
      print *,ipx,'regions;default(0.0) sets internally'
      print *,'additional entree set to 0.0 if default;'
      print *,'otherwise set to 1.0 if slopes entered'
      if(nu.eq.0)read(*,*,err=5033) (schan(lz),lz=1,ipx),schxi
      if(nu.eq.1)read(7,*) (schan(lz),lz=1,ipx),schxi
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5033) ans
      if(ans.gt.0.and.nu.eq.0)goto5033
      write(6,*) (schan(lz),lz=1,ipx),schxi
      if(iu.eq.1)print *, (schan(lz),lz=1,ipx),schxi
c manning coefficient of region
 5034 print *,'ENTER Manning coefficient for each of ',ipx
      print *,'regions;default sets internally; set'
      print *,'additional entree entered as above'
      if(nu.eq.0)read(*,*,err=5034) (rchan(lz),lz=1,ipx),rchxi
      if(nu.eq.1)read(7,*) (rchan(lz),lz=1,ipx),rchxi
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5034) ans
      if(ans.gt.0.and.nu.eq.0)goto5034
      write(6,*) (rchan(lz),lz=1,ipx),rchxi
      if(iu.eq.1)print *, (rchan(lz),lz=1,ipx),rchxi
c length of regions and channels, miles
 5035 print *,'ENTER length of region, miles, along channel or '
      print *,'along border with another region for each of'
      print *,ipx,'regions;default(0.0) sets to 1 mile'
      if(nu.eq.0)read(*,*,err=5035) (xchan(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (xchan(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5035) ans
      if(ans.gt.0.and.nu.eq.0)goto5035
      write(6,*) (xchan(lz),lz=1,ipx)
      if(iu.eq.1)print *, (xchan(lz),lz=1,ipx)
 5036 print *,'ENTER channel lengths,miles, for each of ',icnumb
      print *,'channels;default(0.0) sets to 1.0 mile'
      if(nu.eq.0)read(*,*,err=5036) (xchan(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (xchan(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5036) ans
      if(ans.gt.0.and.nu.eq.0)goto5036
      write(6,*) (xchan(lc),lc=isch,icht)
      if(iu.eq.1)print *, (xchan(lc),lc=isch,icht)
c channel width, feet,slope, and manning coefficient
 5037 print *,'ENTER channel widths, ft, for each of ',icnumb
      print *,'channels;default(0.0) sets internally;'
```

```
      print *,'set additional entree to 0.0 if default,'
      print *,'set to 1.0 if widths entered'
      if(nu.eq.0)read(*,*,err=5037) (wchan(lc),lc=isch,icht),wchi
      if(nu.eq.1)read(7,*) (wchan(lc),lc=isch,icht),wchi
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5037) ans
      if(ans.gt.0.and.nu.eq.0)goto5037
      write(6,*) (wchan(lc),lc=isch,icht),wchi
      if(iu.eq.1)print *, (wchan(lc),lc=isch,icht),wchi
 5038 print *,'ENTER channel slopes, fraction, for each of'
      print *,icnumb,'channels;default(0.0) set internally'
      print *,'additional entry 0.0 or 1.0 as above'
      if(nu.eq.0)read(*,*,err=5038) (schan(lc),lc=isch,icht),schi
      if(nu.eq.1)read(7,*) (schan(lc),lc=isch,icht),schi
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5038) ans
      if(ans.gt.0.and.nu.eq.0)goto5038
      write(6,*) (schan(lc),lc=isch,icht),schi
      if(iu.eq.1)print *, (schan(lc),lc=isch,icht),schi
 5039 print *,'ENTER Manning coefficient for each of ',icnumb
      print *,'channels;default(0.0) set internally;'
      print *,'additional entree 0.0 or 1.0 as above'
      if(nu.eq.0)read(*,*,err=5039) (rchan(lc),lc=isch,icht),rchi
      if(nu.eq.1)read(7,*) (rchan(lc),lc=isch,icht),rchi
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5039) ans
      if(ans.gt.0.and.nu.eq.0)goto5039
      write(6,*) (rchan(lc),lc=isch,icht),rchi
      if(iu.eq.1)print *, (rchan(lc),lc=isch,icht),rchi
c if any channel is perennial,then vtrndp=cfs of flow
 5040 print *,'ENTER perennial flow rate(cfs) for each of'
      print *,icnumb,' channels; enter 0.0 if no-flow'
      if(nu.eq.0)read(*,*,err=5040) (vtrndp(lc),lc=isch,icht)
      if(nu.eq.1)read(7,*) (vtrndp(lc),lc=isch,icht)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5040) ans
      if(ans.gt.0.and.nu.eq.0)goto5040
      write(6,*) (vtrndp(lc),lc=isch,icht)
      if(iu.eq.1)print *, (vtrndp(lc),lc=isch,icht)
 5041 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN GEOFLX                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for GEOFLX?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
```

```
      read(*,*,err=5041)ans
      if(ans) 18,5500,18
   18 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-888888.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c
c ** channel loop ******************************************************
      do 100 lc=isch,icht
      if (xchan(lc)) 102,101,102
  101 xchan(lc)=1.0
  102 if (entchn(lc)) 103,100,103
c scans tributaries
  103 xchant=0.0
      itrib=entchn(lc)
c
c ** tributary loop ****************************************************
      do 80 jl=1,itrib
c identifies tributaries and sums lengths
      ici=chne(lc,jl)
      xchant=xchant+xchan(ici)
   80 continue
c ** end tributary loop ************************************************
  100 continue
c ** end channel loop **************************************************
c
c ** channel loop *******************************************************
      do 120 lc=isch,icht
c sedmns=total soil(lbs) in defined channel
c sdmnsa=total conc. of contaminant in defined channel
      sedmns(lc)=0.0
      sdmnsa(lc)=0.0
c
c ** particle size loop ************************************************
      do 110 ls=1,7
c estimate weight in lbs of channel alluvium 200 mm thick assuming
c a bulk density of 1.5
      dsms(ls,lc)=dsf(ls,lc)*xchan(lc)*324284.*wchan(lc)
c estimate total activity of  channel as above
      dsma(ls,lc)=dzas(ls,lc)*dsms(ls,lc)*454.
      als=ls
      sedmns(lc)=sedmns(lc)+dsms(ls,lc)
      sdmnsa(lc)=sdmnsa(lc)+dsma(ls,lc)
  110 continue
c ** end particle size loop ********************************************
  120 continue
```

```
c ** end channel loop ******************************************************
c
c ** region loop ***********************************************************
      do 130 lz=1,ipx
c default values for regional runoff parameters
      if (c1hr(lz)) 132,131,132
  131 c1hr(lz)=2.53
  132 if (c2hr(lz)) 134,133,134
  133 c2hr(lz)=0.20
  134 if (c5hr(lz)) 136,135,136
  135 c5hr(lz)=4.84
c runoff duration, hours, for region (lz)
  136 duhr(lz)=c1hr(lz)*(rarea(lz)/640.)**(c2hr(lz))
c default runoff and erosion parameters for region(lz)
      if (akond(lz)) 138,137,138
  137 akond(lz)=2.0
  138 if (aksoil(lz)) 141,139,141
  139 aksoil(lz)=0.2
  141 if (csoil(lz)) 143,142,143
  142 csoil(lz)=0.1
  143 if (psoil(lz)) 145,144,145
  144 psoil(lz)=1.0
  145 if (alsoil(lz)) 147,146,147
  146 alsoil(lz)=1.0
  147 if (xchan(lz)) 130,148,130
  148 xchan(lz)=1.0
  130 continue
c ** end region loop ******************************************************
      return
  140 if (sedgat.gt.ai1) return
c
c rndlx,rnylx equal daily,yearly runoff from region lx in inches.
c rdcf,rycf equal daily,yearly runoff from region lx in cubic feet.
c tndlx,tnylx equal daily,yearly sediment yield tons/acre
c tndtn,tnytn equals daily,yearly sediment yield tons/region lx
c rfdlx,rfylx,rfdcf,rfycf equal daily -yearly return subsurface flow
c in inches-cubic feet,respectively.
c dsz,dsf,dsm equal diameter(mm),fraction of particles,and lbs
c eroded for 7 diameter classes.  1st dia.=suspended sediment.
c
c initialize annual runoff parameters
      if (aiii-1.0) 151,149,151
  149 rfylx(lx)=0.0
      tnylx(lx)=0.0
      tnytn(lx)=0.0
      rfycf(lx)=0.0
      rnylx(lx)=0.0
      rycf(lx)=0.0
c initialize daily runoff parameters
  151 rfdlx(lx)=0.0
      tndlx(lx)=0.0
      tndtn(lx)=0.0
      rfdcf(lx)=0.0
      rndlx(lx)=0.0
```

```
      rdcf(lx)=0.0
c initialize total wt. of soil (lbs) eroded/day from region
      dsmt(lx)=0.0
c
c ** particle size loop ***********************************************
      do 150 ls=1,7
c initializes lbs soil eroded by dia. group from region
c initializes particle transport capacity,lbs/sec-ft, by dia. class/region
      dsm(ls,lx)=0.0
      gsbd(ls,lx)=0.0
  150 continue
c ** end particle size loop ********************************************
c initializes total plant crown cover of region
      tcvr(lx)=0.01
c initializes plant counter
      aplx(lx)=ipl
c
c ** plant loop ******************************************************
      do 160 lpr=1,ipl
c excludes irrigated crops from crown cover determination
      if (acres(lpr,lx)-rarea(lx)) 163,164,163
  163 aplx(lx)=aplx(lx)-1.0
  164 if (acres(lpr,lx)-rarea(lx)) 160,165,160
c sums crown covers (ccvv)
  165 tcvr(lx)=tcvr(lx)+ccvv(lpr,lx)
  160 continue
c ** end plant loop ****************************************************
      if (aplx(lx)) 162,161,162
  161 aplx(lx)=1.0
c
c ** plant loop for region (LX) ****************************************
  162 do 420 lpr=1,ipl
      wchr=1.0
c estimates daily runoff from plant(lpr) in region(lx),inches
      qflowx=rnofds(lpr,lx)/25.4
      call eroson (lpr,lx,wchr,schxi,rchxi,qflowx)
c
c return flow(rfdcf,cubic feet),and(rfdlx,inches)are estimated on a
c daily basis from akond,and the crossection (ft) of soil depth
c exceeding field capacity using darcies equation
c
c initialize soil depth,feet,and total moisture content,cf,of soil layers
c potentially contributing to return flow
      dpx=0.0
      watvx=0.0
c estimated crown cover area,ft2,of plant(lpr) in region(lx)
      arx=scvr(lpr,lx)*rarea(lx)*43562.
      iicde=rcode(lpr,lx)
c ** soil layer loop omitting top layer(20 mm) *************************
      do 170 ly=2,iicde
c allows return flow from layers exceeding field capacity
      if (wa(lpr,ly,lx).le.f1(lpr,ly,lx)) go to 170
      dpx=dpx+slice(lpr,ly,lx)/(25.4*12.)
      watvx=watvx+(wa(lpr,ly,lx)-f1(lpr,ly,lx))*arx/(25.4*12.)
```

```
 170 continue
c ** end soil layer loop ************************************************
    if (watvx) 191,190,191
c estimates crossectional area for return flow for plant(lpr) in region(lx)
 191 arrx=scvr(lpr,lx)*dpx*xchan(lx)*5280.
c estimates potential return flow assuming dh/dl=0.1
    watva=2.0*arrx*akond(lx)*0.1
c limits flow that available
    if (watva.ge.watvx) watva=watvx
c return flow in cubic feet and inches/day
    rfdcf(lx)=rfdcf(lx)+watva
    rfdlx(lx)=rfdlx(lx)+12.*watva/arx
c
c ** soil layer loop ****************************************************
    do 180 ly=2,iicde
    if (wa(lpr,ly,lx).le.f1(lpr,ly,lx)) go to 180
c total moisture residual above field capacity,cubic feet, from layer(ly)
c of plant(lpr) in region(lx)
    watz=(wa(lpr,ly,lx)-f1(lpr,ly,lx))*arx/(25.4*12.)
c weighted removal of moisture,mm, from layer(ly) of plant(lpr)
c to account for return flow losses
    wa(lpr,ly,lx)=wa(lpr,ly,lx)-(watva*12.*25.4/arx)*watz/watvx
 180 continue
c ** end soil layer loop ************************************************
 190 continue
c excludes all but last cycle in plant and region loops
    if (lpr.ne.ipl) go to 420
    if (lx.ne.ipx) go to 420
c
c regional loops
c
c ** region loop ********************************************************
    do 290 lz=1,ipx
c number of regions supplying runoff to a given region(lz)
    ireg=regent(lz)
c temperature, precipitation,inches, surface area, and total available
c moisture in top layer, mm
    tchan(lz)=t(lz)
    prch(lz)=p(lz)/25.4
    sarea(lz)=rarea(lz)
    precp=prec(lz)
    if (ireg.eq.0) go to 280
    atct=regent(lz)+1.0
    tchan(lz)=tchan(lz)/atct
    prch(lz)=prch(lz)/atct
c ** runoff regions loop ************************************************
    do 220 lr=1,ireg
c identifies specific contributing region
    irgx=regx(lz,lr)
c estimates weighted temperature and precipitation in contributing region
c receiver region complex
    tchan(lz)=tchan(lz)+t(irgx)/atct
    prch(lz)=prch(lz)+p(irgx)/(atct*25.4)
c estimates total area of contributing and receiver regions
```

```
      sarea(lz)=sarea(lz)+rarea(irgx)
c regional loop for estimating runoff from contributing sites
c if runoff from a given region(lz)is receiving runoff from another
c region(s)(irgx),then infiltration is estimated
      ipz=apl(lz)
c estimates runoff contribution,mm, from region(irgx)to(lz) weighted by
c areal differences
      prec(lz)=25.4*rndlx(irgx)*rarea(irgx)/rarea(lz)
c
c ** plant loop ********************************************************
      do 210 lpz=1,ipz
      qflowx=0.0
c increments top soil layer with runoff received from region(irgx),mm
      wa(lpz,1,lz)=wa(lpz,1,lz)+prec(lz)
      if (t(lz).le.2.0) go to 200
c calls RUNOFF subroutine
      call runoff (lpz,lz)
c sums daily runoff from plant(lpz) of region(lx)
      rnodx(lpz,lz)=rnodx(lpz,lz)+rnofds(lpz,lz)
c converts runoff from mm to inches/day
      qflowx=rnofds(lpz,lz)/25.4
  200 wchr=1.0
c calls EROSON subroutine
      call eroson (lpz,lz,wchr,schxi,rchxi,qflowx)
  210 continue
c ** end plant loop ****************************************************
  220 continue
c ** end runoff regions loop ********************************************
c resets precipitation to original regional value
      prec(lz)=precp
c
c ** plant loop ********************************************************
      do 230 lpz=1,ipz
      rnofds(lpz,lz)=rnodx(lpz,lz)
  230 continue
c ** end plant loop ****************************************************
c
c ** contributing region loop ********************************************
      do 270 lr=1,ireg
      irgx=regx(lz,lr)
c
c ** particle size loop ************************************************
      do 240 ls=1,7
c estimates sediment deposition-erosion from contributing regions(irgx)
c to region(lz)
      dsms(ls,lz)=dsms(ls,lz)+dsm(ls,irgx)
      if(dsms(ls,lz).le.0.0)dsms(ls,lz)=0.0
      dsma(ls,lz)=dsma(ls,lz)+dsm(ls,irgx)*dzas(ls,irgx)*454.
      if(dsma(ls,lz).le.0.0)dsma(ls,lz)=0.0
      sedmns(lz)=sedmns(lz)+dsm(ls,irgx)
      sdmnsa(lz)=sdmnsa(lz)+dsm(ls,irgx)*dzas(ls,irgx)*454.
  240 continue
      if(sedmns(lz).le.0.0)sedmns(lz)=0.0
      if(sdmnsa(lz).le.0.0)sdmnsa(lz)=0.0
```

```
c ** particle size loop ************************************************
    ipll=apl(lz)
c total moisture,mm,added to region(lz) from region(irgx)
    rfxt=25.4*12.*rfdcf(irgx)/(rarea(lz)*43562.)
c regional loop for coupling subsurface flow from contributing region(s)
c
c ** plant loop  ******************************************************
    do 260 lprr=1,ipll
    jicde=rcode(lprr,lz)
c weighted moisture allocation to plant(lprr) based on crown cover
    rfxxt=rfxt*scvr(lprr,lz)
c
c ** soil layer loop excluding top (20 mm) layer ************************
    do 250 jj=2,jicde
    if (rfxxt.le.0.0) go to 250
c commences with deepest soil layer
    ldc=jicde-jj+2
c estimates maximum moisture,mm,deficit to reach saturation
    rfdx=h(lprr,ldc,lz)-wa(lprr,ldc,lz)
    if (rfdx.le.0.0) rfdx=0.0
    if (rfdx.gt.rfxxt) rfdx=rfxxt
c increments soil moisture and readjusts available moisture(rfxxt) for next
c soil layer
    wa(lprr,ldc,lz)=wa(lprr,ldc,lz)+rfdx
    rfxxt=rfxxt-rfdx
    if (rfxxt.le.0.0) rfxxt=0.0
 250 continue
c ** end soil layer loop ***********************************************
c if all layers saturated, then water added and/or ponded in top layer
    wa(lprr,1,lz)=wa(lprr,1,lz)+rfxxt
 260 continue
c ** end plant loop ****************************************************
 270 continue
c ** end contributing region loop ***************************************
 280 continue
 290 continue
c ** end region loop ***************************************************
c
c ** channel loop for determining individual losses(gains) ****************
    do 410 lc=isch,icht
c initializes channel transmission loss,lateral inflow,return flow inflow,
c and losses from aquifer (cfd)
    vtrnd(lc)=0.0
    vlat(lc)=0.0
    vrfl(lc)=0.0
    vloss(lc)=0.0
c initializes channel drainage area,mean drainage precipitation and
c temperature,sediment mass(lbs),peak flow(cfs),and peak flow(in./hr)
    rarea(lc)=0.0
    prch(lc)=0.0
    tchan(lc)=0.0
    dsmt(lc)=0.0
    cfspd(lc)=0.0
    qpkd(lc)=0.0
```

```
c
c ** particle size loop ************************************************
      do 300 ls=1,7
c initialize eroded sediment(lbs/day),and particle transport capacity
c per unit channel width(lbs/sec-ft)
      dsm(ls,lc)=0.0
      gsbd(ls,lc)=0.0
  300 continue
c ** end particle size loop ********************************************
c number of regions supplying lateral runoff to channel(lc)
      ilatx=rnchn(lc)
      alatx=ilatx
      if (ilatx.eq.0) go to 340
c
c ** lateral addition loop *********************************************
      do 330 ilt=1,ilatx
c
c ** region loop ******************************************************
      do 320 jx=1,ipx
      ajx=jx
      if (ajx-chlat(lc,ilt)) 320,321,320
c increments channel(lc) with return and surface flow from region(jx)
  321 vrfl(lc)=vrfl(lc)+rfdcf(jx)
      vlat(lc)=vlat(lc)+rdcf(jx)
c average precipitation for region(s) entering channel,inches
      prch(lc)=prch(lc)+prch(jx)/alatx
c mean channel air temperature
      tchan(lc)=tchan(lc)+tchan(jx)/alatx
c estimates total drainage area for channel(lc) in acres
      rarea(lc)=rarea(lc)+sarea(jx)
c
c ** particle size loop ************************************************
      do 310 ls=1,7
c
c increments channel particle size(ls),cumulative for particle(ls), total
c mass(lbs) all particle sizes, total activity of particles(ls), and total
c activity all particle sizes
c
      dsm(ls,lc)=dsm(ls,lc)+dsm(ls,jx)
      if(dsm(ls,lc).le.0.0)dsm(ls,lc)=0.0
      dsms(ls,lc)=dsms(ls,lc)+dsm(ls,jx)
      if(dsms(ls,lc).le.0.0)dsms(ls,lc)=0.0
      sedmns(lc)=sedmns(lc)+dsm(ls,jx)
      if(sedmns(lc).le.0.0)sedmns(lc)=0.0
      dsma(ls,lc)=dsma(ls,lc)+dsm(ls,jx)*dzas(ls,jx)*454.
      if(dsma(ls,lc).le.0.0)dsma(ls,lc)=0.0
      sdmnsa(lc)=sdmnsa(lc)+dsm(ls,jx)*dzas(ls,jx)*454.
      if(sdmnsa(lc).le.0.0)sdmnsa(lc)=0.0
  310 continue
c ** end particle size loop ********************************************
  320 continue
c ** end region loop ***************************************************
  330 continue
c ** end lateral addition loop *****************************************
```

```
 340 continue
c
c initialize transmission of moisture from other channels to channel(lc),
c number of channels entering channel(lc), mean temperature and
c precipitation parameter for channel(lc)
c
    vtrndj=0.0
    ichx=entchn(lc)
    if (ichx.eq.0) go to 380
    achx=ichx+1.0
    tchan(lc)=tchan(lc)/achx
    prch(lc)=prch(lc)/achx
c
c ** contributing channel loop ******************************************
    do 370 jch=1,ichx
c
c ** total channel loop ****************************************************
    do 360 jc=isch,icht
    ajc=jc
    if (ajc-chne(lc,jch)) 360,361,360
c
c increment channel(lc) for transmission from channel(jc),increment
c aquifer volume(ft3) for loss from channel(jc), increment channel(lc)
c drainage area for drainage area of channel(jc), and estimate weighted
c mean precipitation and temperature from channel drainage area(jc)
c
 361 vtrndj=vtrndj+vtrnd(jc)
    vaqf(lc)=vaqf(lc)+vloss(jc)
    rarea(lc)=rarea(lc)+rarea(jc)
    tchan(lc)=tchan(lc)+tchan(jc)/achx
    prch(lc)=prch(lc)+prch(jc)/achx
c
c ** particle size loop ****************************************************
    do 350 ls=1,7
c increments particle size(ls) mass,total soil mass, particle activity,
c and total soil activity for contributions from  channel(jc)
    dsms(ls,lc)=dsms(ls,lc)+dsm(ls,jc)
    if(dsms(ls,lc).le.0.0)dsms(ls,lc)=0.0
    sedmns(lc)=sedmns(lc)+dsm(ls,jc)
    if(sedmns(lc).le.0.0)sedmns(lc)=0.0
    dsma(ls,lc)=dsma(ls,lc)+dsm(ls,jc)*dzas(ls,jc)*454.
    if(dsma(ls,lc).le.0.0)dsma(ls,lc)=0.0
    sdmnsa(lc)=sdmnsa(lc)+dsm(ls,jc)*dzas(ls,jc)*454.
    if(sdmnsa(lc).le.0.0)sdmnsa(lc)=0.0
 350 continue
c ** end particle size loop ************************************************
 360 continue
c ** end channel loop ******************************************************
 370 continue
c ** end contributing channel loop ******************************************
 380 continue
c initializes channel ice melt, cfd
    amelt=0.0
    vtrnp=vtrndp(lc)
```

```
c estimates area of aquifer surface crossection, ft2
      charex=(tch(lc)+tbch(lc))*0.5*xchan(lc)*5280.
      if (tchan(lc).le.2.0) go to 390
      if (aice(lc).le.0.0) go to 400
c estimates potential snowmelt from surface, cfd
      amelt=(11.28*tchan(lc)+1.1128)*charex/(25.4*12.)
c estimates amount of icemelt available, ft3
      avmlt=aice(lc)*charex/(25.4*12.)
      if (amelt.ge.avmlt) amelt=avmlt
c adjusts ice thickness,mm, for icemelt loss
      aice(lc)=aice(lc)-amelt*25.4*12./charex
      if (aice(lc).le.0.0) aice(lc)=0.0
      go to 400
c estimates channel flow(ft3) for ice formation
  390 volbed=vlat(lc)+vtrndj+vtrnp
c estimates ice thickness increment,mm, from channel flow
      aice(lc)=aice(lc)+(volbed/charex)*12.*25.4
c resets liquid channel flow contributors to 0.0
      vlat(lc)=0.0
      vtrndj=0.0
      vtrnp=0.0
c resultant water transmission along channel(lc) estimated
  400 vtrnd(lc)=amelt+vtrndj+vtrnp
c transmission -loss parameters initialized
      alnc=0.0
      btrn=0.0
      atrn=0.0
      if (aqugat.gt.0.0) call aquifer (lc)
c if aquifer not present,then return flow added to lateral flow(cfd),and
c total channel flow estimated(cfd)
      if (aqugat) 402,401,402
  401 vlat(lc)=vlat(lc)+vrfl(lc)
  402 tflowx=vtrnd(lc)+vlat(lc)
      if (tflowx.le.0.0) go to 410
c channel runoff duration,hours,and mean velocity of flow,ft/sec estimated
      duhr(lc)=c1hr(lc)*(rarea(lc)/640.)**(c2hr(lc))
      vmean=0.05*(rarea(lc)/640.)**(-0.2)
c unit channel intercept for transmission loss,acre-feet,and parameter
c for estmating channel decay factor,inches/acre-foot estimated
      auci=-.00465*akond(lc)*duhr(lc)
      alnc=0.00545*akond(lc)*duhr(lc)/(vmean*rarea(lc)/12.)
      if (alnc.ge.0.999999) alnc=0.999999
c akdcf(mile/foot) = decay factor
      akdcf=-1.09*alog(1.-alnc)
      bctr=exp(-akdcf)
c btrn =regression slope
      btrn=exp(-akdcf*xchan(lc)*wchan(lc))
c atrn(acre-ft) = regression intercept
      atrn=(auci/(1.-bctr))*(1.-btrn)
c ftrn(acre-ft) = lateral flow parameter for channel
      ftrn=(1.-btrn)/(akdcf*wchan(lc))
c vtrnd = outflow volume from channel, acre-ft
      vtrnd(lc)=atrn+btrn*(vtrnd(lc)/43562.)
      vtrnd(lc)=vtrnd(lc)+ftrn*(vlat(lc))/(43562.*xchan(lc))
```

```
c converts transmission loss to cfd
      vtrnd(lc)=vtrnd(lc)*43562.
      if (vtrnd(lc).le.0.0) vtrnd(lc)=0.0
c bounds transmission loss to current channel flow ,cfd
      if (vtrnd(lc).ge.tflowx) vtrnd(lc)=tflowx
c if aquifer present non-transmitted channel water enters aquifer
      if (aqugat.gt.0.0) vaqf(lc)=vaqf(lc)+(tflowx-vtrnd(lc))
c transmitted channel flow converted to inches
      qflowx=vtrnd(lc)*12./(xchan(lc)*5280.*wchan(lc))
      ddd=0.0
c calls EROSON subroutine
      call eroson (idd,lc,wchi,schi,rchi,qflowx)
c
  410 continue
c ** end channel loop ***************************************************
  420 continue
c yearly runoff parameters incremented for daily additions
      rnylx(lx)=rnylx(lx)+rndlx(lx)
      tnylx(lx)=tnylx(lx)+tndlx(lx)
      tnytn(lx)=tnytn(lx)+tndtn(lx)
      rycf(lx)=rycf(lx)+rdcf(lx)
      rfylx(lx)=rfylx(lx)+rfdlx(lx)
      rfycf(lx)=rfycf(lx)+rfdcf(lx)
      return
      end
c
```

14.12. HUMTRN (blm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine humtrn (in,lx,ii,fedh,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
c      real lvtbl, kdtbl1, kdtbl2, mstbl
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),wt1(kr,ko),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      common/pw/ gaslos(ko),urnlos(ko),wtbn(kr,ko),
     1wtkd(kr,ko),wtlv(kr,ko),wtms(kr,ko),wtrs(kr,ko),wtlu(kr,ko)
c
      common/pa/ absw(120,ko), adia(120,ko), al4(m), al5(m),
     1bddt(120,ko),c66(m),degg(120,ko),egg(120,ko),
     2hatce(120,ko), hatch(120,ko),hatmp(120,ko),hattmp(120,ko),
     3ondv(30,ko), onyph(30,ko), pdiape(120,ko), thz(24), tmpday(30,ko)
     4,watcnr(ko,m,n),yadlt(30,ko),yndv(30,ko),
     5 ynyph(30,ko),hemlos(ko,m,n),bltgt(ko,m),fdtmp(ko),tpsm(ko),
     6 dvtmp(ko),adlt(2,ko),iegg(ko),iegd(ko),iegh(ko),eggs,deggs,
     7 hatchs, pdiaps,aha(ko),yadlts,ynyphs,onyphs,eglay,degday(ko),
     8 colday(ko),diaday(ko),eggmrt(ko),aii,summrt(ko,2)
c
      common/pj/ awt(m), cdrp(ks,n), dsrt(m,n), hbar(ks,n), ph(i,ks,n),
     1 rock(i,ks,n), sand(i,ks,n), silt(i,ks,n), soilf(i,n), tim(i,n),
     2 w1bar(ks,n), watcn(m,n), wtf(n), yaz(m,n), ysrt(m,n),
     3 yup(m,n), z1dl(i,ks,m,n)
c
      common/ha/ popt(ko),tqx0(kr,2,ko),jage(ko),aiif(ko,2),ifm(kr),
     1 vurlfj(ko),vmljk(kr,ko),popp(2,ko),ikn(2,ko,8),sfpop(ko),
     2 gst(ko,2),gstn(ko),gestp(ko),aiiir(ko,2),agem(kr,ko),cagm(ko,2),
     3 cntk(2),brts(ko,2,2),frpr(2,ko),agcor(ko),tpppt(ko),sfd(8,ko),
     4tppcs(ko),tppps(ko),ca1(ko),clf(ko),lagf(ko),lua(ko),offsp(ko,3,2)
     5 ,affsp(ko,3),rat12(ko),tqp(kr,ko),tqs(kr,ko),vpelt(kr,ko,m),
     6vgut(kr,ko,m),cnmus(kr,ko,m),cnliv(kr,ko,m),cnkid(kr,ko,m),
     7cnfat(kr,ko,m),cnlun(kr,ko,m),cnbon(kr,ko,m),cnres(kr,ko,m),
     8cnblo(kr,ko,m),cnbod(kr,ko,m),cngut(kr,ko,m),cnpel(kr,ko,m),
     9vpln(kr,ko,m),vsol(kr,ko,m),vlug(kr,ko,m),cnpln(kr,ko,m),
     1cnsol(kr,ko,m),cnlug(kr,ko,m),rtpls(ko),rtmus(ko),rtkid(ko),
```

```
     2rtliv(ko),rtgut(ko),rthrt(ko),rtlun(ko),rtspl(ko),rtmar(ko),
     3rtfat(ko),rtres(ko),cbp(kr,ko,m),cbf(ko,m),cbw(ko,m),tacres(n),
     4vplt(kr,ko),c31plt(kr,ko,m),cnmlk(kr,ko,m),cnurnc(kr,ko,m),
     5cnfec(kr,ko,m),cnurn(kr,ko,m)
      dimension poppi(2,ko),pop2r(ko)
c
c  humtrn subroutine simulates growth and metabolism of male and
c  female humans. food is specified by input identifying which
c  plants simulated are to be consumed
c
      nhm=2
      nto=nan+nhm
      nts=nan+1
      if (rdidx.gt.0.) go to 190
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5070
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-999999.)5702,5070,5702
 5555 nu=0
 5070 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE HUMTRN                    ')
      call wnoust('****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 999999.                    ')
```

```
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c enters male and/or female indicies
 5001 print *,'ENTER male and female gates(1,mature weight);'
      print *,'a value of 1 specifies that default population'
      print *,'parameters will be used; if mature weight is'
      print *,'specified, then population structure will be created; if'
      print *,'a negative value is entered, then model parameters for'
      print *,'contaminant transport will be entered, otherwise they'
      print *,'will default to model estimates; 2 values required'
      if(nu.eq.0)read(*,*,err=5001)(rr(l),l=nts,nto)
      if(nu.eq.1)read(7,*)(rr(l),l=nts,nto)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write(6,*)(rr(l),l=nts,nto)
      if(iu.eq.1)print *,(rr(l),l=nts,nto)
c enters work activity level, total yearly and/or daily food schedules, soft
c tissue, bone,and endosteal bone deposition ,Mev-Rem/dis-Rad;and day and
c year modulus
c
      do11j=nts,nto
c ** excludes sex not simulated
 5002 print *,'ENTER human activity level(1.=normal) for',j
      if(nu.eq.0)read(*,*,err=5002)acti(j)
      if(nu.eq.1)read(7,*)acti(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*)acti(j)
      if(iu.eq.1)print *,acti(j)
 5003 print *,'ENTER total yearly and/or daily food schedules'
      print *,'to be specified by user, default =1.0'
      if(nu.eq.0)read(*,*,err=5003)rzoon(j)
      if(nu.eq.1)read(7,*)rzoon(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
```

```
      write(6,*)rzoon(j)
      if(iu.eq.1)print *,rzoon(j)
      do12k=1,inz
      if(k.gt.1.and.k.lt.7)goto12
      ian=atno(k)
      cbf(j,k)=0.0
      cbw(j,k)=0.0
      if(ian.ge.101)goto12
 5004 print *,'ENTER soft tiss. dose conv. fac.,Mev-rem/dis-Rad'
      print *,'for human',j,'and contaminant',k
      if(nu.eq.0)read(*,*,err=5004)evs(j,k)
      if(nu.eq.1)read(7,*)evs(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)evs(j,k)
      if(iu.eq.1)print *,evs(j,k)
 5005 print *,'ENTER bone tiss. dose conv.fac. as above'
      if(nu.eq.0)read(*,*,err=5005)evb(j,k)
      if(nu.eq.1)read(7,*)evb(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*)evb(j,k)
      if(iu.eq.1)print *,evb(j,k)
 5006 print *,'ENTER endosteal bone tiss. conv. fac. as above'
      if(nu.eq.0)read(*,*,err=5006)eve(j,k)
      if(nu.eq.1)read(7,*)eve(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*)eve(j,k)
      if(iu.eq.1)print *,eve(j,k)
   12 continue
c
 5099 print *,'for accessing of the EFFECTS subroutine:'
      print *,'ENTER number of effects years simulated in this'
      print *,'run; a value of 0.0 will not allow access to'
      print *,'the EFFECTS subroutine'
c
      if(nu.eq.0)read(*,*,err=5099) ydos(j)
      if(nu.eq.1)read(7,*) ydos(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5099) ans
      if(ans.gt.0.and.nu.eq.0)goto5099
      write(6,*)ydos(j)
      if(iu.eq.1)print *, ydos(j)
   11 continue
c
c ** region loop ****************************************************
      do 10 lz=1,ipx
c enters regional contaminant conc., per liter, in drinking water
      print *,'For region',lz
 5008 print *,'ENTER contaminant conc., ug/l, for',inz,'contaminants'
```

```
      if(nu.eq.0)read(*,*,err=5008)(watcni(k,lz),k=1,inz)
      if(nu.eq.1)read(7,*)(watcni(k,lz),k=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*)(watcni(k,lz),k=1,inz)
      if(iu.eq.1)print *,(watcni(k,lz),k=1,inz)
   10 continue
c ** end region loop ************************************************
c
c ** region loop ****************************************************
      do 30 lz=1,ipx
c
c ** contaminant loop ***********************************************
      do 20 k=1,inz
c converts stable contaminants, ug/liter to g/liter
      if (hflif(k)) 20,21,20
   21 watcni(k,lz)=watcni(k,lz)/1.0e06
   20 continue
c ** end contaminant loop *******************************************
   30 continue
c ** end region loop ************************************************
c
c ** male-female loop ************************************************
      do 120 j=nts,nto
c
 5023 print *,'ENTER fraction of meat consumed from study area'
      print *,'for human group',j
      if(nu.eq.0)read(*,*,err=5023)p5(j)
      if(nu.eq.1)read(7,*)p5(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5023) ans
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*)p5(j)
      if(iu.eq.1)print *,p5(j)
c
 5024 print *,'for human',j,'ENTER population size, and population'
      print *,'structure type: 1=none,2=parabolic; a value of 1'
      print *,'indicates structure will be entered by user, or default'
      print *,'values will be utilized'
c
      if(nu.eq.0)read(*,*,err=5024) zoo(j),popt(j)
      if(nu.eq.1)read(7,*) zoo(j),popt(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*) zoo(j),popt(j)
      if(iu.eq.1)print *, zoo(j),popt(j)
c
 5025 print *,'ENTER fraction of study area utilized '
      print *,'by human group',j
      if(nu.eq.0)read(*,*,err=5025) aa2(j)
      if(nu.eq.1)read(7,*) aa2(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
```

```
      if(nu.eq.0)read(*,*,err=5025) ans
      if(ans.gt.0.and.nu.eq.0)goto5025
      write(6,*) aa2(j)
      if(iu.eq.1)print *, aa2(j)
c
 5026 print *,'ENTER fraction non-meat food consumed from study'
      print *,'area by human group',j
      if(nu.eq.0)read(*,*,err=5026)fsupf(j)
      if(nu.eq.1)read(7,*)fsupf(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5026) ans
      if(ans.gt.0.and.nu.eq.0)goto5026
      write(6,*)fsupf(j)
      if(iu.eq.1)print *,fsupf(j)
c
 5028 print *,'ENTER the number of the yearly food schedule'
      print *,'changes for human group',j,'minimum value=1.0'
      if(nu.eq.0)read(*,*,err=5028)eatin(j)
      if(nu.eq.1)read(7,*)eatin(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5028) ans
      if(ans.gt.0.and.nu.eq.0)goto5028
      write(6,*)eatin(j)
      if(iu.eq.1)print *,eatin(j)
c
 5029 print *,'ENTER the number of the daily food schedule'
      print *,'changes for human group',j,'minimum value=1.0'
      if(nu.eq.0)read(*,*,err=5029)fodin(j)
      if(nu.eq.1)read(7,*)fodin(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5029) ans
      if(ans.gt.0.and.nu.eq.0)goto5029
      write(6,*)fodin(j)
      if(iu.eq.1)print *,fodin(j)
c
 5030 print *,'for human sex',j,'ENTER oldest age group, years'
      print *,'may be set larger or equal to life-span (ALSP) as'
      print *,'complete years;85 years is maximum age allowed'
      print *,'enter 0.0 for default to 85 yr life-span'
c
      print *,'also ENTER when gestation period is initiated:'
      print *,'1=Spring, 2=Fall; also enter number of gestation'
      print *,'periods/year,and fraction of offspring produced in'
      print *,'first reproductive period (equals 1.0 if only one'
      print *,'reproductive cycle is specified); 4 entries required'
c
      if(nu.eq.0)read(*,*,err=5030)agea(j),gst(j,1),gstn(j),
     1frpr(1,j)
      if(nu.eq.1)read(7,*)agea(j),gst(j,1),gstn(j),frpr(1,j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5030) ans
      if(ans.gt.0.and.nu.eq.0)goto5030
      write(6,*)agea(j),gst(j,1),gstn(j),frpr(1,j)
      if(iu.eq.1)print *,agea(j),gst(j,1),gstn(j),frpr(1,j)
```

```
      if(gst(j,1).le.1.0)gst(j,1)=92.0
      if(gst(j,1).lt.92.0)gst(j,1)=270.0
      frpr(2,j)=1.0-frpr(1,j)
c
c year schedule index (upper-lower year interval pairs)
      iiet=eatin(j)*2.0
      iieth=iiet/2
c daily food schedules throughout year (upper-lower day number pairs)
      iifd=fodin(j)*2.0
      iifdh=iifd/2
c maximum age index
      iage=agea(j)
c feeding intervals by yearly intervals
 5031 print *,'ENTER',iieth,'yearly food sub-intervals for human'
      print *,'group',j,'; 2 values/interval'
      if(nu.eq.0)read(*,*,err=5031)(yrfed(j,l),l=1,iiet)
      if(nu.eq.1)read(7,*)(yrfed(j,l),l=1,iiet)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5031) ans
      if(ans.gt.0.and.nu.eq.0)goto5031
      write(6,*)(yrfed(j,l),l=1,iiet)
      if(iu.eq.1)print *,(yrfed(j,l),l=1,iiet)
c
c feeding intervals by day number intervals
 5032 print *,'ENTER',iifdh,'daily food sub-intervals for human'
      print *,'group',j,'; 2 values/interval'
      if(nu.eq.0)read(*,*,err=5032)(onfod(j,l),l=1,iifd)
      if(nu.eq.1)read(7,*)(onfod(j,l),l=1,iifd)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5032) ans
      if(ans.gt.0.and.nu.eq.0)goto5032
      write(6,*)(onfod(j,l),l=1,iifd)
      if(iu.eq.1)print *,(onfod(j,l),l=1,iifd)
c
c ** region loop ******************************************************
      do 80 lz=1,ipx
      if(lz.gt.1)goto5036
c number of daily food schedules for each yearly interval
c
 5035 print *,'ENTER number of daily food schedules for each'
      print *,'yearly interval for human group',j
      print *,iieth,'intervals; 2 values/interval required'
      if(nu.eq.0)read(*,*,err=5035)(eatrec(j,l),l=1,iiet)
      if(nu.eq.1)read(7,*)(eatrec(j,l),l=1,iiet)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5035) ans
      if(ans.gt.0.and.nu.eq.0)goto5035
      write(6,*)(eatrec(j,l),l=1,iiet)
      if(iu.eq.1)print *,(eatrec(j,l),l=1,iiet)
c
c total number of daily and/or yearly food schedules
 5036 kherb=rzoon(j)
c diet type sequence, 1,2,3....kherb
      print *,'ENTER',kherb,'diet sequence..1.,2.,3.,...etc.'
```

```
      print *,'for human group',j,'in region',lz
      if(nu.eq.0)read(*,*,err=5036)(pltind(j,l,lz),l=1,kherb)
      if(nu.eq.1)read(7,*)(pltind(j,l,lz),l=1,kherb)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5036) ans
      if(ans.gt.0.and.nu.eq.0)goto5036
      write(6,*)(pltind(j,l,lz),l=1,kherb)
      if(iu.eq.1)print *,(pltind(j,l,lz),l=1,kherb)
c
c ** feed schedules loop ***********************************************
      do60l=1,kherb
      if(lz.gt.1)goto5037
c
 5033 print *,'ENTER number plant types used for schedule',l
c
      if(nu.eq.0)read(*,*,err=5033) aisch
      if(nu.eq.1)read(7,*) aisch
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5033) ans
      if(ans.gt.0.and.nu.eq.0)goto5033
      write(6,*) aisch
      if(iu.eq.1)print *, aisch
 5037 isch=aisch
c
c plant type numbers used in schedule
c
      print *,'ENTER new plant type numbers used in schedule for'
      print *,'human group',j,'in region',lz,isch,'inputs required'
      print *,'a maximum of eight are possible'
      if(nu.eq.0)read(*,*,err=5037)(pltusd(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltusd(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5037) ans
      if(ans.gt.0.and.nu.eq.0)goto5037
      write(6,*)(pltusd(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltusd(j,k,l,lz),k=1,isch)
c
c plant food type...leafy,grains..etc. used in schedule
c
 5038 print *,'ENTER new plant food types for human group',j
      print *,'in region',lz,isch,'inputs required; a maximum of'
      print *,'8 inputs possible; 4=leafy veg;5=grains;6=seed veg'
c
      if(nu.eq.0)read(*,*,err=5038)(pltrnk(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltrnk(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5038) ans
      if(ans.gt.0.and.nu.eq.0)goto5038
      write(6,*)(pltrnk(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltrnk(j,k,l,lz),k=1,isch)
c
c utilization fraction of plants used in schedule (must equal 1.0)
c
 5039 print *,'ENTER new utilization fractions of plants used'
```

```
      print *,'in schedule for human group',j,'in region',lz
      print *,isch,'inputs required; a maimum of 8 inputs possible'
c
      if(nu.eq.0)read(*,*,err=5039)(pltf(j,k,l,lz),k=1,isch)
      if(nu.eq.1)read(7,*)(pltf(j,k,l,lz),k=1,isch)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5039) ans
      if(ans.gt.0.and.nu.eq.0)goto5039
      write(6,*)(pltf(j,k,l,lz),k=1,isch)
      if(iu.eq.1)print *,(pltf(j,k,l,lz),k=1,isch)
c
      do51k=1,isch
      pusd=pltusd(j,k,l,lz)
      do52ll=1,ipl
      if(pusd-pltid(ll,lz)) 52,53,52
   52 continue
      goto60
   53 prnk=pltrnk(j,k,l,lz)
      putzd(j,ll,lz)=prnk
   51 continue
c
   60 continue
c ** end schedule loop ***************************************************
   80 continue
c ** end region loop ****************************************************
c
c a negative value for RR signifiies user specification of transport
c coefficients

c*** perfusion parameters for 70 kg standard man****
c
      wtgrm=70000.
c plasma vol, liters
      volpls=4.29e-05*wtgrm**0.992
c muscle vol, liters
      volmus=4.63e-04*wtgrm**1.009
c kidney vol, liters
      volkid=2.18e-05*wtgrm**0.843
c liver vol, liters
      volliv=8.59e-05*wtgrm**0.885
c gut vol, liters
      volgut=8.65e-05*wtgrm**0.909
c gut lumen vol, liters
      vollum=5.78e-05*wtgrm**0.934
c heart vol, liters
      volhrt=2.0e-06*wtgrm**1.043
c lungs vol, liters
      vollun=3.16e-06*wtgrm**1.104
c spleen vol, liters
      volspl=4.52e-06*wtgrm**0.901
c marrow vol, liters
      volmar=3.3e-05*wtgrm**0.936
c fat vol, liters
      volfat=1.1e-04*wtgrm
```

```
c current residual volume
c     volres=volgut+volhrt+volspl
      volres=volgut+volhrt+volspl+vollun
c
c ** plasma flow rates **
c
c plasma flow, liters/day
      flopls=1.44*.601*wtgrm**0.778
c muscle flow, liters/day
      flomus=1.44*.0817*wtgrm**0.81
c kidney flow, liters/day
      flokid=1.44*0.0845*wtgrm**0.802
c liver flow, liters/day
      floliv=1.44*0.0944*wtgrm**0.792
c gut flow, liters/day
      flogut=1.44*0.102*wtgrm**0.787
c heart flow, liters/day
      flohrt=1.44*5.77e-03*wtgrm**0.965
c lung flow, liters/day
      flolun=1.44*0.215*wtgrm**0.82
c spleen flow, liters/day
      flospl=1.44*1.79e-03*wtgrm**1.028
c marrow flow, liters/day
      flomar=1.44*0.0164*wtgrm**0.804
c fat flow, liters/day
      flofat=0.75*flomus
c flow to residual
c     flores=flogut+flohrt+flospl
      flores=flohrt+flospl+0.1*flogut+0.1*flolun
c urine production, liters/day
      flourn=0.024*.0064*wtgrm**0.82
c maximum milk production from a 60 kg woman, liters/day
      flomlk=0.163*60.0**0.71
c
c *** end of perfusion parameters ****
c
      if (rr(j).ge.0.0) go to 96

c ** user specified contaminant parameters
c
 5040 print *,'ENTER lung (N-P) deposition frac. for human group',j
      if(nu.eq.0)read(*,*,err=5040) plcm1(j)
      if(nu.eq.1)read(7,*) plcm1(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5040) ans
      if(ans.gt.0.and.nu.eq.0)goto5040
      write(6,*) plcm1(j)
      if(iu.eq.1)print *, plcm1(j)
 5041 print *,'ENTER lung (T-B) deposition frac. for human group',j
      if(nu.eq.0)read(*,*,err=5041) plcm2(j)
      if(nu.eq.1)read(7,*) plcm2(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5041) ans
      if(ans.gt.0.and.nu.eq.0)goto5041
```

```
      write(6,*) plcm2(j)
      if(iu.eq.1)print *, plcm2(j)
 5042 print *,'ENTER lung (P) deposition frac. for human group',j
      if(nu.eq.0)read(*,*,err=5042) plcm3(j)
      if(nu.eq.1)read(7,*) plcm3(j)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5042) ans
      if(ans.gt.0.and.nu.eq.0)goto5042
      write(6,*) plcm3(j)
      if(iu.eq.1)print *, plcm3(j)
c
c ** contaminant loop ********************************************
   96 do 100 k=1,inz
c atomic number of contaminant, N,P,K,Pu,..,or contaminant index number
      ian=atno(k)
c gives default values for nutrients
      if (k.gt.1.and.k.lt.7) go to 90
c
      print *,'FOR human group',j,'and contaminant',k
 5043 print *,'ENTER GI-blood daily transport fraction; a value of 0.0'
      print *,'will default to UCRL estimate if contaminant is a'
      print *,'nuclide; if not it will default to 0.5; enter now'
c
      if(nu.eq.0)read(*,*,err=5043)gtbl(j,k)
      if(nu.eq.1)read(7,*)gtbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5043) ans
      if(ans.gt.0.and.nu.eq.0)goto5043
      write(6,*)gtbl(j,k)
      if(iu.eq.1)print *,gtbl(j,k)
      if(ian.le.100.and.gtbl(j,k).le.0.0)gtbl(j,k)=fbwbz(ian)
      if(ian.gt.100.and.gtbl(j,k).le.0.0)gtbl(j,k)=0.5
c
 5047 print *,'ENTER blood-urine daily transport fraction; a'
      print *,'default value of 0.0 will set both contaminants and'
      print *,'organic contaminants to a 1 day biological half-time;'
      print *,'enter now'
c
      if(nu.eq.0)read(*,*,err=5047)bltur(j,k)
      if(nu.eq.1)read(7,*)bltur(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5047) ans
      if(ans.gt.0.and.nu.eq.0)goto5047
      write(6,*)bltur(j,k)
      if(iu.eq.1)print *,bltur(j,k)
      if(bltur(j,k).le.0.0)bltur(j,k)=0.693
c
 5049 print *,'ENTER blood to milk daily fractional transport;a default'
      print *,'value of 0.0 will set nuclides to UCRL values; for'
      print *,'organic contaminants the value is set to the square of'
      print *,'the gut absorption. Enter value now.'
c
      if(nu.eq.0)read(*,*,err=5049) bltmk(j,k)
      if(nu.eq.1)read(7,*)bltmk(j,k)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5049) ans
      if(ans.gt.0.and.nu.eq.0)goto5049
      write(6,*)bltmk(j,k)
      if(iu.eq.1)print *,bltmk(j,k)
      bmk=bltmk(j,k)
      if(ian.le.100.and.bmk.le.0.0)bltmk(j,k)=fmk(ian)/gtbl(j,k)
      if(ian.gt.100.and.bmk.le.0.0)bltmk(j,k)=gtbl(j,k)**2
c
 5046 print *,'ENTER lung(P)-blood daily fractional transport; from'
      print *,'particulates; a value of 0.0 will default to 0.1/day;'
      print *,'for all contaminants; enter value now'
c
      if(nu.eq.0)read(*,*,err=5046)pltbl(j,k)
      if(nu.eq.1)read(7,*)pltbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5046) ans
      if(ans.gt.0.and.nu.eq.0)goto5046
      write(6,*)pltbl(j,k)
      if(iu.eq.1)print *,pltbl(j,k)
      if(pltbl(j,k).le.0.0)pltbl(j,k)=0.1
c
 5048 print *,'ENTER alveoli-to-blood from gas phase partition'
      print *,'coefficient; if contaminant is non-volatile, enter'
      print *,'a value of -1.0; if user enters 0.0, the the'
      print *,'partition coefficient will default to 10 for volatile'
      print *,'contaminants; enter value now'
c
      if(nu.eq.0)read(*,*,err=5048)pltblg(j,k)
      if(nu.eq.1)read(7,*)pltblg(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5048) ans
      if(ans.gt.0.and.nu.eq.0)goto5048
      write(6,*)pltblg(j,k)
      if(iu.eq.1)print *,pltblg(j,k)
      ilbg=pltblg(j,k)
      if(ilbg.eq.0)pltblg(j,k)=10.
      if(ilbg.le.-1.0)pltblg(j,k)=0.0
c
 5044 print *,'ENTER lung(N-P)-GI daily transport fraction from'
      print *,'particulates; a value of 0.0 will default to to a'
      print *,'biological half-time of 1 day; enter value now'
c
      if(nu.eq.0)read(*,*,err=5044)pltg1(j,k)
      if(nu.eq.1)read(7,*)pltg1(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5044) ans
      if(ans.gt.0.and.nu.eq.0)goto5044
      write(6,*)pltg1(j,k)
      if(iu.eq.1)print *,pltg1(j,k)
      if(pltg1(j,k).le.0.0)pltg1(j,k)=0.693
c
 5045 print *,'ENTER lung(T-B)-GI daily transport fraction from'
      print *,'particulates; a value of 0.0 will default to a'
```

```
      print *,'biological half-time of 10 days; enter value now'
c
      if(nu.eq.0)read(*,*,err=5045)pltg2(j,k)
      if(nu.eq.1)read(7,*)pltg2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5045) ans
      if(ans.gt.0.and.nu.eq.0)goto5045
      write(6,*)pltg2(j,k)
      if(iu.eq.1)print *,pltg2(j,k)
      if(pltg2(j,k).le.0.0)pltg2(j,k)=0.0693
c
 5058 print *,'ENTER fat to blood daily transport fraction or partition'
      print *,'coefficient for contaminant',k,'; enter a positive'
      print *,'number if value is a transport coefficient; or the'
      print *,'negative of the value if it is a partition coefficent;'
      print *,'a value of 0.0 will default to a 4 day half-time; a'
      print *,'value of -1000000.0 will default to a partition'
      print *,'coefficient of 10; enter value now'
c
      if(nu.eq.0)read(*,*,err=5058) fttbl(j,k)
      if(nu.eq.1)read(7,*)fttbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5058) ans
      if(ans.gt.0.and.nu.eq.0)goto5058
      write(6,*)fttbl(j,k)
      if(iu.eq.1)print *,fttbl(j,k)
      ftb=fttbl(j,k)
      if(ftb.le.-1000000.0)fttbl(j,k)=-10.
      if(ftb.gt.0.0)fttbl(j,k)=volpls*flofat/(flopls*volfat*ftb)
      if(ftb.lt.0.0)fttbl(j,k)=-fttbl(j,k)
      if(fttbl(j,k).le.0.0)fttbl(j,k)=volpls*flofat/(flopls*volfat*.173)
c
      if (rr(j).ge.0.0) go to 90
c
 5050 print *,'ENTER bone(1)-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5050)bntbl1(j,k)
      if(nu.eq.1)read(7,*)bntbl1(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5050) ans
      if(ans.gt.0.and.nu.eq.0)goto5050
      write(6,*)bntbl1(j,k)
      if(iu.eq.1)print *,bntbl1(j,k)
      bn1=bntbl1(j,k)
      if(bn1.gt.0.0)bntbl1(j,k)=volpls*flomar/(flopls*volmar*bn1)
      if(bn1.lt.0.0)bntbl1(j,k)=-bn1
c
 5051 print *,'ENTER bone(2)-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
```

```
c
      if(nu.eq.0)read(*,*,err=5051)bntbl2(j,k)
      if(nu.eq.1)read(7,*)bntbl2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5051) ans
      if(ans.gt.0.and.nu.eq.0)goto5051
      write(6,*)bntbl2(j,k)
      if(iu.eq.1)print *,bntbl2(j,k)
      bn2=bntbl2(j,k)
      if(bn2.gt.0.0)bntbl2(j,k)=volpls*flomar/(flopls*volmar*bn2)
      if(bn2.lt.0.0)bntbl2(j,k)=-bn2
c
 5053 print *,'ENTER liver-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5053)lvtbl(j,k)
      if(nu.eq.1)read(7,*)lvtbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5053) ans
      if(ans.gt.0.and.nu.eq.0)goto5053
      write(6,*)lvtbl(j,k)
      if(iu.eq.1)print *,lvtbl(j,k)
      alv=lvtbl(j,k)
      if(alv.gt.0.0)lvtbl(j,k)=volpls*floliv/(flopls*volliv*alv)
      if(alv.lt.0.0)lvtbl(j,k)=-alv
c
 5056 print *,'ENTER kidney(1)-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5056)kdtbl1(j,k)
      if(nu.eq.1)read(7,*)kdtbl1(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5056) ans
      if(ans.gt.0.and.nu.eq.0)goto5056
      write(6,*)kdtbl1(j,k)
      if(iu.eq.1)print *,kdtbl1(j,k)
      akd=kdtbl1(j,k)
      if(akd.gt.0.0)kdtbl1(j,k)=volpls*flokid/(flopls*volkid*akd)
      if(akd.lt.0.0)kdtbl1(j,k)=-akd
c
 5057 print *,'ENTER kidney(2)-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5057)kdtbl2(j,k)
      if(nu.eq.1)read(7,*)kdtbl2(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5057) ans
      if(ans.gt.0.and.nu.eq.0)goto5057
```

```
      write(6,*)kdtbl2(j,k)
      if(iu.eq.1)print *,kdtbl2(j,k)
      akd=kdtbl2(j,k)
      if(akd.gt.0.0)kdtbl2(j,k)=volpls*flokid/(flopls*volkid*akd)
      if(akd.lt.0.0)kdtbl2(j,k)=-akd
c
 5059 print *,'ENTER muscle-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5059) mstbl(j,k)
      if(nu.eq.1)read(7,*) mstbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5059) ans
      if(ans.gt.0.and.nu.eq.0)goto5059
      write(6,*) mstbl(j,k)
      if(iu.eq.1)print *, mstbl(j,k)
      ams=mstbl(j,k)
      if(ams.gt.0.0)mstbl(j,k)=volpls*flomus/(flopls*volmus*ams)
      if(ams.lt.0.0)mstbl(j,k)=-ams
c
 5060 print *,'ENTER residual tissue-blood daily transport fraction or'
      print *,'partition coefficient; if a partition coefficient'
      print *,'is entered, then enter the negative of the value; a'
      print *,'0.0 entree will ignore this component; enter now'
c
      if(nu.eq.0)read(*,*,err=5060)rstbl(j,k)
      if(nu.eq.1)read(7,*)rstbl(j,k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5060) ans
      if(ans.gt.0.and.nu.eq.0)goto5060
      write(6,*)rstbl(j,k)
      if(iu.eq.1)print *,rstbl(j,k)
      rsb=rstbl(j,k)
      if(rsb.gt.0.0)rstbl(j,k)=volpls*flores/(flopls*volres*rsb)
      if(rsb.lt.0.0)rstbl(j,k)=-rsb
c
      go to 100
c
c default values of transport and deposition fractions are estimated from
c information in DATA statements
c
   90 tbb=1.0
      tbl=1.0
      tbk=1.0
      tbw=0.6
c **** end partition coefficents ***
c
c *** estimated biological half-lives of radionuclide contaminants
c
c real variables from UCRL data if specified
      fbww=1.0
      if(ian.le.100)fbww=fbwbz(ian)
```

```
      tbbb=1.0
      if(ian.le.100)tbbb=0.693/tbbnz(ian)
      tbll=1.0
      if(ian.le.100)tbll=0.693/tblvz(ian)
      tbkk=1.0
      if(ian.le.100)tbkk=0.693/tbkdz(ian)
      tbww=1.0
      if(ian.le.100)tbww=0.693/tbwbz(ian)
c
c *** end nuclide biological half-times estimates from UCRL data***
      if(k.gt.1.and.k.lt.7)pltbl(j,k)=0.1
      if(k.gt.1.and.k.lt.7)pltblg(j,k)=0.0
      plcm1(j)=0.25*0.4
      plcm2(j)=0.25*.4
      plcm3(j)=0.25*0.2
      if(k.gt.1.and.k.lt.7)gtbl(j,k)=fbwbz(ian)
      if(k.gt.1.and.k.lt.7)bltmk(j,k)=fmk(ian)
      if(k.gt.1.and.k.lt.7)bltur(j,k)=0.693
      bntbl1(j,k)=volpls*flomar/(flopls*volmar*tbbb)
      if(ian.gt.100)bntbl1(j,k)=tbb
      bntbl2(j,k)=volpls*flomar/(flopls*volmar*tbbb)
      if(ian.gt.100)bntbl2(j,k)=tbb
      lvtbl(j,k)=volpls*floliv/(flopls*volliv*tbll)
      if(ian.gt.100)lvtbl(j,k)=tbl
      kdtbl1(j,k)=volpls*flokid/(flopls*volkid*tbkk)
      if(ian.gt.100)kdtbl1(j,k)=tbk
      kdtbl2(j,k)=volpls*flokid/(flopls*volkid*tbkk)
      if(ian.gt.100)kdtbl2(j,k)=tbk
      mstbl(j,k)=volpls*flomus/(flopls*volmus*tbww)
      if(ian.gt.100)mstbl(j,k)=tbw
      rstbl(j,k)=volpls*flores/(flopls*volres*tbww)
      if(ian.gt.100)rstbl(j,k)=tbw
      if(k.gt.1.and.k.lt.7)fttbl(j,k)=
     1volpls*flofat/(flopls*volfat*0.1733)
c
c blood-milk transport coefficient set for ages 18-40 year-old females
c from UCRL data;
c
      if(j-nan.eq.1)bltmk(j,k)=0.0
      if(k.gt.1.and.k.lt.7)goto100
      call clear_screen@
      print *,'*******************************************************'
      print *,'default partition coefficients estimated from UCRL'
      print *,'biological half-life data'
      print *,'*******************************************************'
      print *,'bone 1,2=',bntbl1(j,k),bntbl2(j,k),j
      print *,'liver=',lvtbl(j,k)
      print *,'kidney 1,2=',kdtbl1(j,k),kdtbl2(j,k)
      print *,'muscle=',mstbl(j,k)
      print *,'residual=', rstbl(j,k)
      print *,'*****************************************************'
      print *,'CAUTION, UCRL specific organ data not always reliable'
      print *,'*****************************************************'
      print *,'fat(partition coefficient)',fttbl(j,k)
```

```
      print *,'gut absorption coefficient',gtbl(j,k)
      print *,'transport fraction to urine from blood,d-1',bltur(j,k)
      print *,'transport fraction to milk, from blood,d-1',bltmk(j,k)
      print *,'*****************************************'
      pause
      call clear_screen@
c
  100 continue
c ** end contaminant loop ****************************************
c default value for max age(0.0) assumes default age and weight values
      if (iage.eq.0) go to 110
c excludes simulations where population structure will be estimated
      if(popt(j).gt.1.0)goto110
c enters fraction of males-females in each age group to total in group
c
 5062 print *,'ENTER fraction of population in',iage,'age groups'
      print *,'for human group',j
      if(nu.eq.0)read(*,*,err=5062) (a(k,ko,j-nan),k=1,iage)
      if(nu.eq.1)read(7,*) (a(k,ko,j-nan),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5062) ans
      if(ans.gt.0.and.nu.eq.0)goto5062
      write(6,*) (a(k,ko,j-nan),k=1,iage)
      if(iu.eq.1)print *, (a(k,ko,j-nan),k=1,iage)
c
 5068 print *,'ENTER specified population',j,'mortality rate for'
      print *,iage,'age groups; enter 0.0 when not known or used'
      if(nu.eq.0)read(*,*,err=5068)(tqx0(k,j-nan,ko),k=1,iage)
      if(nu.eq.1)read(7,*)(tqx0(k,j-nan,ko),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5068) ans
      if(ans.gt.0.and.nu.eq.0)goto5068
      write(6,*)(tqx0(k,j-nan,ko),k=1,iage)
      if(iu.eq.1)print *,(tqx0(k,j-nan,ko),k=1,iage)
c
      if(j-nan.eq.1)goto5063
c
 5082 print *,'assuming that the reproductive age interval for FEMALES'
      print *,'is bounded between 0.15 and 0.70 of the life-span:'
      print *,'divide reproductive female population into up to 8 age'
      print *,'sub-intervals:(TRAILING zeros may be added if not all'
      print *,'sub-intervals are needed) depending on differential'
      print *,'reproductive rate with age; ENTER 8 inputs now'
c
      if(nu.eq.0)read(*,*,err=5082) (brty(k,ko),k=1,8)
      if(nu.eq.1)read(7,*) (brty(k,ko),k=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5082) ans
      if(ans.gt.0.and.nu.eq.0)goto5082
      write(6,*) (brty(k,ko),k=1,8)
      if(iu.eq.1)print *, (brty(k,ko),k=1,8)
c
c enters expected weights, kgfwt, in each male-female age group
c
```

```
 5063 print *,'ENTER mean weights,kgfwt,for',iage,'population'
      print *,'age groups for human group',j
      if(nu.eq.0)read(*,*,err=5063)(wt1(k,j),k=1,iage)
      if(nu.eq.1)read(7,*)(wt1(k,j),k=1,iage)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5063) ans
      if(ans.gt.0.and.nu.eq.0)goto5063
      write(6,*)(wt1(k,j),k=1,iage)
      if(iu.eq.1)print *,(wt1(k,j),k=1,iage)
  110 continue
c default maximum age utilized in simulation, years
      jage(j)=iage
      if(iage.eq.0) agea(j)=kr-1
      iage=agea(j)
      wt1(iage,j)=70
      if(rr(j).gt.1.0)wt1(iage,j)=rr(j)
c
c input utilized in EFFECTS model
c
      mm=j-nan
      if(ydos(j).le.0.0)goto120
      iian=iian+1
      call sleep@(2.0)
      call clear_screen@
      nue=nu
      nu=nuu
      call effects(j,lx,ii,mm,rdidx)
      nu=nue
      call sleep@(2.0)
      call clear_screen@
  120 continue
c ** end male-female loop ***********************************************
c
 5064 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN HUMTRN                        ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for HUMTRN?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5064)ans
      if(ans) 121,5500,121
  121 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-999999.)5501,5503,5501
```

```
5503 backspace 6
    nu=nuu
    call clear_screen@
    goto 5070
5501 continue
5500 continue
    nu=nuu
    call clear_screen@
c
c ** male-female loop ***************************************************
    do 180 j=nts,nto
    iage=agea(j)
    do281kg=1,iage
    if(popt(j).gt.1.0)goto281
    a(kg,j,j-nan)=a(kg,ko,j-nan)
    if(jage(j).gt.0)tqx0(kg,j-nan,j)=tqx0(kg,j-nan,ko)
    if(jage(j).eq.0)tqx0(kg,j-nan,j)=tqx(kg,j-nan,ko)
    if(kg.gt.8)goto281
    brty(kg,j)=brty(kg,ko)
 281 continue
c initial total population by sex 1=male, 2=female
    pop(j-nan,j)=zoo(j)
    popp(j-nan,j)=pop(j-nan,j)
    poppi(j-nan,j)=pop(j-nan,j)
    affsp(j,1)=0.0
    affsp(j,2)=0.0
    affsp(j,3)=0.0
    offsp(j,1,j-nan)=0.0
    offsp(j,2,j-nan)=0.0
    offsp(j,3,j-nan)=0.0
    sfd(1,j)=.0001
    sfd(2,j)=.0001
    sfd(3,j)=.0001
    sfd(4,j)=.0001
    sfd(5,j)=.0001
    sfd(6,j)=.0001
    sfd(7,j)=.0001
    sfd(8,j)=.0001
c
    alsp(j)=72.0*365.
c estimated gestation period, days estimated from mean life-span
    gestp(j)=0.08*(alsp(j))**0.8
c time delay for newborns in a given year
    cagm(j,1)=gst(j,1)+14.+gestp(j)
    gst(j,2)=0.0
    if(gstn(j).ge.2.0)gst(j,2)=cagm(j,1)+90.
    cagm(j,2)=0.0
    if(gstn(j).ge.2.0) cagm(j,2)=gst(j,2)+14.+gestp(j)
    agcor(j)=0.0
    if(gstn(j).ge.2.0) agcor(j)=(14.+90.+gestp(j))/2.0
c
c  population structure and natality parameters
    ipopt=popt(j)
    trep=0.55*alsp(j)/8.0
```

```
      srep=0.15*alsp(j)
      iaged=iage*365
      femgrp=0.0
      akdy=1.0
      do 626 kk=1,iaged
      akd=kk
c excludes age groups below .15, and greater than .70 of the life-span
      if (akd.lt..15*alsp(j).or.akd.gt..70*alsp(j)) go to 626
      femgrp=femgrp+1.0
      kdy=akdy
      if(akdy.le.1.0)goto131
      if(akd.lt.srep+trep)goto626
      ageo=agea(j)*365.
c estimates fraction of reproductive females in 8 age groups
  131 goto (133,132), ipopt
c parabolic population structure
  132 sfd(kdy,j)=((ageo-srep)/ageo)**1.5-((ageo-(srep+trep)
     1)/ageo)**1.5
      goto134
  133 isrp=srep/365.
      if(isrp.lt.1)isrp=1
      istrp=(srep+trep)/365.
      if(istrp.lt.1)istrp=1
      sfd(kdy,j)=(a(isrp,j,j-nan)+a(istrp,j,j-nan))/2.0
  134 akdy=akdy+1.0
      if(akdy.gt.8.0)goto136
c
  626 continue
c determines upper age for 8 age categories of reproductive females
  136 kn=1
      akn=kn
      femgpi=0.0
      do627kk=1,iaged
      if(kn.gt.8)goto627
      akd=kk
      if (akd.lt..15*alsp(j).or.akd.gt..70*alsp(j)) go to 627
      femgpi=femgpi+1.0
      femc=femgrp*akn/8.0
      if(femgpi.lt.femc)goto627
      brty(kn,j)=akd
      kn=kn+1
      akn=kn
  627 continue
c
      sfp=1.0
      agemi=0.0
c ** human age loop ***************************************************
      do 112 kg=1,iage
c estimates initial age , days, of human at start of simulation
      agemi=agemi+365.
      aka=kg
      if(agemi.gt.alsp(j))agemi=alsp(j)
      if(kg.eq.1)agemri=agemi-cagm(j,1)-agcor(j)
      if(agemri.lt.0.0)agemri=-agemri
```

```
      agemrf=365.
c estimates age stucture,fraction, for each yearly age group
      goto (124,123), ipopt
c parabolic
  123 sfc=((ageo-agemi)/ageo)**1.5
      if(kg.eq.1)sfr=((ageo-agemri)/ageo)**1.5
      if(kg.eq.1)sfrf=((ageo-agemrf)/ageo)**1.5
      a(kg,j,j-nan)=(sfp-sfc)
      if(kg.eq.iage)a(kg,j,j-nan)=sfp
      sfp=sfc
      if(kg.ne.1)goto111
      sfpop(j)=(1.0-sfc)
      rat12(j)=sfrf/(sfr+sfrf)
      goto127
  124 if(kg.ne.1)goto111
      sfpop(j)=a(kg,j,j-nan)
      sfr=a(1,j,j-nan)*agemri/agemrf
      if(jage(j).eq.0)sfr=a(1,j,j-nan)
      sfrf=a(2,j,j-nan)
      if(jage(j).eq.0)sfrf=0.0
      rat12(j)=sfr/(sfr+sfrf)
c
c differential birth rates for males and females for stable population
c given as number of births/year/female in 8 reproductive age groups
c
  127 jn=j-nan
      brrt(1,j,jn)=.003*sfpop(j)*pop(jn,j)/(sfd(1,j)*zoo(nto)*1.5)
      brrt(2,j,jn)=.138*sfpop(j)*pop(jn,j)/(sfd(2,j)*zoo(nto)*1.5)
      brrt(3,j,jn)=.296*sfpop(j)*pop(jn,j)/(sfd(3,j)*zoo(nto)*1.5)
      brrt(4,j,jn)=.295*sfpop(j)*pop(jn,j)/(sfd(4,j)*zoo(nto)*1.5)
      brrt(5,j,jn)=.207*sfpop(j)*pop(jn,j)/(sfd(5,j)*zoo(nto)*1.5)
      brrt(6,j,jn)=.050*sfpop(j)*pop(jn,j)/(sfd(6,j)*zoo(nto)*1.5)
      brrt(7,j,jn)=.010*sfpop(j)*pop(jn,j)/(sfd(7,j)*zoo(nto)*1.5)
      brrt(8,j,jn)=.001*sfpop(j)*pop(jn,j)/(sfd(8,j)*zoo(nto)*1.5)
c
  111 tpop(kg,j-nan,j)=a(kg,j,j-nan)*pop(j-nan,j)
      tpopi(kg,j-nan,j)=tpop(kg,j-nan,j)
      if(kg.eq.1)tpop(kg,j-nan,j)=(sfr-sfrf)*pop(j-nan,j)
      if(tpop(kg,j-nan,j).lt.0.0)tpop(kg,j-nan,j)=-tpop(kg,j-nan,j)
      if(kg.eq.1)offsp(j,1,j-nan)=tpop(kg,j-nan,j)
      alspi=alsp(j)
      agem(kg,j)=(aka-1.0)*365.
      if(kg.eq.1)agem(kg,j)=0.006*alspi
      iage=agea(j)
      wt1(kg,j)=wt1(iage,j)*(1.-exp(-9.0*agem(kg,j)/alspi))
      wt(kg,j)=wt1(kg,j)
      wtgrm=wt1(kg,j)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
      amw(kg,j)=5.6234e-03*wtgrm**0.75
  112 continue
c
c ** human age loop ***************************************************
      iage=agea(j)
      do 170 kg=1,iage
```

```
c number of individuals in each population group estimated
      pop(j-nan,j)=pop(j-nan,j)+tpop(kg,j-nan,j)
c
      lf=iage+1
      tqp(kg,j)=0.0
      tqs(kg,j)=0.0
c
c ** contaminant loop ************************************************
      do130l=1,inz
      rkd(lf,j,l)=0.0
      rbn(lf,j,l)=0.0
      rndo(lf,j,l)=0.0
      rlv(lf,j,l)=0.0
      rms(lf,j,l)=0.0
c Rem cumulative doses to internal organs initialized
c kidneys
      rkd(kg,j,l)=0.0
c bone
      rbn(kg,j,l)=0.0
c endosteal tissue of bone
      rndo(kg,j,l)=0.0
c liver
      rlv(kg,j,l)=0.0
c lung
      rlu(kg,j,l)=0.0
c muscle
      rms(kg,j,l)=0.0
c** internal organ concentrations initialized
c bone
      vbn1(kg,j,l)=0.0
      vbn2(kg,j,l)=0.0
c liver
      vlv(kg,j,l)=0.0
c muscle
      vms(kg,j,l)=0.0
c kidneys
      vkd1(kg,j,l)=0.0
      vkd2(kg,j,l)=0.0
c blood
      v(kg,j,l)=0.0
c lungs
      q1(kg,j,l)=0.0
      q2(kg,j,l)=0.0
      q3(kg,j,l)=0.0
c adipose tissue
      vfat(kg,j,l)=0.0
c residual tissue
      vr(kg,j,l)=0.0
c milk contaminant content
      vmk(kg,j,l)=0.0
      vgut(kg,j,l)=0.0
      vpelt(kg,j,l)=0.0
      vpln(kg,j,l)=0.0
      vsol(kg,j,l)=0.0
```

```
      vlug(kg,j,l)=0.0
  130 continue
c ** end contaminant loop ****************************************
c
c initializes weight, kgfwt, of male-female in age group
      ak=kg
      if(jage(j).eq.0) wt1(kg,j)=10.
c routes to male or female section
      go to (150,140), j-nan
c
c female section
c
c female weight, kgfwt, depending on age
  140 if(jage(j).gt.0)goto141
      if (kg.gt.1.and.kg.le.10) wt1(kg,j)=10.+2.4*(ak-1.0)
      if (kg.gt.10.and.kg.le.13) wt1(kg,j)=31.6+(ak-10.)*5.47
      if (kg.gt.13) wt1(kg,j)=48.+.611*(ak-13.)
      if (kg.ge.32) wt1(kg,j)=59.+.20*(ak-31.)
  141 rff=1.0
      wt1(kg,j)=wt1(kg,j)*rff
c female total fat content, kg, depending on age
      fatb(kg,j)=.0125*exp(-(.693/7.2)*ak)*wt1(kg,j)
      if (kg.gt.14) fatb(kg,j)=.003732*(1.+.0055*(ak-14.))*wt1(kg,j)
      fatb(kg,j)=fatb(kg,j)*wt1(kg,j)
c female total body water, liters, depending on age
      tbwk=.511*wt1(kg,j)+1.244
      if (kg.gt.16) tbwk=.6981*wt1(kg,j)-.0026*wt1(kg,j)**2-.0012*ak*
     1wt1(kg,j)
      go to 160
c
c male section
c
c male weight, kgfwt, depending on age
  150 if(jage(j).gt.0)goto151
      if (kg.gt.1.and.kg.le.10) wt1(kg,j)=10.+2.4*(ak-1.0)
      if (kg.gt.10.and.kg.le.18) wt1(kg,j)=31.+4.75*(ak-10.)
      if (kg.gt.18.and.kg.le.32) wt1(kg,j)=70.+.36*(ak-18.)
      if (kg.gt.32) wt1(kg,j)=75.-.11*(ak-32.)
  151 rfm=1.0
      wt1(kg,j)=wt1(kg,j)*rfm
c male total body fat, kg, depending on age
      fatb(kg,j)=1.25*exp(-(.693/7.2)*ak)*wt1(kg,j)/100.
      if (kg.gt.18) fatb(kg,j)=.21+.000307*(ak-18.)*wt1(kg,j)/100.
      fatb(kg,j)=fatb(kg,j)*wt1(kg,j)
c male total body water, liters, depending on age
      tbwk=.6111*wt1(kg,j)+0.251
      if (kg.gt.18) tbwk=.7945*wt1(kg,j)-.0024*wt1(kg,j)-.0015*ak*
     1wt1(kg,j)
c male-female crystalline bone fraction of total body weight; conditional
  160 bnfr=0.032
      if (kg.gt.18) bnfr=0.041
c male-female protein-carbohydrate body weight, kg
      wtres=wt1(kg,j)-bnfr*wt1(kg,j)-tbwk-fatb(kg,j)
c male-female body protein weight, kg
```

```
      protb(kg,j-nan)=0.97*wtres
      prot1(kg,j-nan)=protb(kg,j-nan)
c male-female duplicate expected body weight
      wt(kg,j)=wt1(kg,j)
c male-female carbohydrate weight, kg
      ch20b(kg,j-nan)=.03*wtres
c*******
      wtgrm=wt1(kg,j)*1000.
c weight of blood, kg
      wtbl(kg,j)=5.5e-05*wtgrm**0.99
c weight of bone, kg
      wtbn(kg,j)=5.5e-05*wtgrm
c weight of kidneys, kg
      wtkd(kg,j)=2.12e-05*wtgrm**0.85
c weight of liver, kg
      wtlv(kg,j)=8.2e-05*wtgrm**0.87
c weight of lung, kg
      wtlu(kg,j)=1.24e-05*wtgrm**0.99
c weight of muscle, kg
      wtms(kg,j)=4.5e-04*wtgrm
c fat weight, kg
      fatb(kg,j)=1.1e-04*wtgrm
c weight of residual soft tissue, kg
      wtrs(kg,j)=wt1(kg,j)-wtbl(kg,j)-wtbn(kg,j)-wtkd(kg,j)-
     1wtlv(kg,j)-wtms(kg,j)-wtlu(kg,j)-fatb(kg,j)
c
      do172l=1,inz
c atomic number of contaminant...N,P,K,Pu, ..etc, or index number
      ian=atno(l)
c initializes urine contaminant content
      vurin(kg,j,l)=0.0
c initializes metabolic waste transport to parent from fetus
      cbp(kg,j,l)=0.0
c excludes organic contaminants
      if(ian.gt.100)goto172
      cbz=1.0e6
c preserves units, pCi, ug, uCi, ...etc.
      if(yaz(l,lx).ge.2)cbz=1.0
c converts to pCi
      if(hflif(l).gt.0.0)cbz=1.0
c
c initial total contaminant  content, grams, pCi, ug, etc. of organs
c for nutrients only
      if(l.lt.2.or.l.gt.6)goto172
      v(kg,j,l)=(cbwbz(ian)/cbz)*wtbl(kg,j)*1000.
      vr(kg,j,l)=(cbwbz(ian)/cbz)*wtrs(kg,j)*1000.
      vbn1(kg,j,l)=(cbbnz(ian)/cbz)*wtbn(kg,j)*1000.
      vkd1(kg,j,l)=(cbkdz(ian)/cbz)*wtkd(kg,j)*1000.
      vlv(kg,j,l)=(cblvz(ian)/cbz)*wtlv(kg,j)*1000.
      vms(kg,j,l)=(cbwbz(ian)/cbz)*wtms(kg,j)*1000.
  172 continue
  170 continue
c ** end age loop ****************************************************
c
```

```
c ** human fetal age relative to gestation period initialized
      aiif(j,1)=0.0
      aiif(j,2)=0.0
      aiiir(j,1)=0.0
      aiiir(j,2)=0.0
      pop2r(j)=1.0
  180 continue
c ** end male-female loop ********************************************
c daily meat intake(ICRP), gfwt, for adult American males
      ameat=275.*.19+379.*.095+527.*.032+193.*.032+193.*.012
c same for carbohydrate intake, gdwt
      choh=379.*.758+527.*.046+193.*.037+193.*.03
c same for fat intake, grams
      fat=275.*.21+379.*.03+527.*.05+193.*.003+193.*.001
      return
  190 continue
c male-female index
c
c male-female age group
      k=ii
      aii=ii
      j=in-nan
      print *,'age=',ii,'sex=',j,aiii,ai1
      iage=agea(in)
      if(aiii.gt.1.0)goto194
      if(ii.gt.iage)goto194
      agem(ii,in)=(aii-1.0)*365.
      if(ii.ne.1)goto194
      if(ai1.gt.yrfed(in,1))goto194
      agem(ii,in)=365.-cagm(in,1)-agcor(in)
      affsp(in,1)=agem(ii,in)
      if(affsp(in,1).lt.1.0)affsp(in,1)=1.0
c
c excludes all but first age group
  194 if(ii.gt.1)goto191
c resets fetus age counter
      lagf(in)=0
c age of first newborn stage initialized
      if(aiiir(in,1).ge.cagm(in,1)+90.)goto196
      if(aiiir(in,2).ge.cagm(in,2)+90.)goto196
      goto197
  196 agem(ii,in)=365.-cagm(in,1)-agcor(in)
      affsp(in,1)=agem(ii,in)
      if(affsp(in,1).lt.1.0)affsp(in,1)=1.0
c
  197 aiiir(in,1)=aiiir(in,1)+1.0
      if(gst(in,2).ge.2.0)aiiir(in,2)=aiiir(in,2)+1.0
c
c indicates that gestation period has ended; resumes regular time
      if(aiiir(in,1).gt.cagm(in,1)+90.) aiiir(in,1)=aiii
      if(aiiir(in,2).gt.cagm(in,2)+90.) aiiir(in,2)=aiii
c
c gestation still active
      if(aiiir(in,1).le.aiii)goto191
```

```
      if(gstn(in).lt.2.0) agem(ii,in)=agem(ii,in)+1.0
      if(gstn(in).lt.2.0) goto 703
      if(aiiir(in,2).gt.aiii)agem(ii,in)=agem(ii,in)+1.0
      if(aiiir(in,2).gt.aiii)goto 703
c
  191 if(ii.ne.1)goto188
      if(gstn(in).ge.2.0)goto192
      iagem=affsp(in,2)
      if(iagem.ne.2)goto188
c sums up total births in one year
      pop2r(in)=pop(in-nan,in)/poppi(in-nan,in)
      goto193
  192 iagem=affsp(in,3)
      if(iagem.ne.2)goto188
  193 continue
c
  188 if(ii.le.iage)goto189
c
c weaning constant initialized
      cntk(1)=0.0
      cntk(2)=0.0
c estimates age of newborns
      agem(ii,in)=aiiir(in,1)-cagm(in,1)
c
c weaning constant set to 1.0 after weaning period
      if(affsp(in,2).gt.cagm(in,1)+90.)cntk(1)=1.0
      if(gstn(in).lt.2.0)goto703
      if(affsp(in,3).gt.cagm(in,2)+90.)cntk(2)=1.0
      goto703
  189 agem(ii,in)=agem(ii,in)+1.0
      if(ii.ne.1)goto195
      affsp(in,1)=agem(ii,in)
      if(affsp(in,1).lt.1.0)affsp(in,1)=1.0
      if(affsp(in,2).ge.1.0)affsp(in,2)=affsp(in,2)+1.0
      if(affsp(in,3).ge.1.0)affsp(in,3)=affsp(in,3)+1.0
  195 tqx(ii,in-nan,in)=0.0
      tqx1=0.0
      tqx2=0.0
      tqx3=0.0
      if(agem(ii,in).le.0.0)goto703
      if(jage(in).eq.0)goto184
      of1=affsp(in,1)
      of2=affsp(in,2)
      of3=affsp(in,3)
      agemh=agem(ii,in)
      ageo=agea(in)*365
      ipopt=popt(in)
      goto (182,183), ipopt
c parabolic population structure
  183 if(ii.gt.1)goto283
      tqx1=(0.5/ageo)*(1.-of1/ageo)**(-0.5)/pop2r(in)
      tqx2=(0.5/ageo)*(1.-of2/ageo)**(-0.5)/pop2r(in)
      tqx3=(0.5/ageo)*(1.-of3/ageo)**(-0.5)/pop2r(in)
      goto184
```

```
 283 tqx(ii,in-nan,in)=tqx(ii-1,in-nan,in)
     if(agem(ii,in).ge.ageo-2.0)goto184
     tqx(ii,in-nan,in)=(0.5/ageo)*(1.-agemh/ageo)**(-0.5)/pop2r(in)
     goto184
 182 tqxp1=0.0
     if(ii-1.gt.0)tqxp1=tqx0(ii-1,in-nan,in)/365.
     dqxp1=tqx0(ii,in-nan,in)/365.-tqxp1
     tqx(ii,in-nan,in)=tqx0(ii,in-nan,in)/365.+dqxp1*(aiii/365.)
     if(ii.gt.1)goto184
     tqx1=tqx0(ii,in-nan,in)/365.
     tqx2=tqx0(ii,in-nan,in)/365.
     tqx3=tqx0(ii,in-nan,in)/365.
c
 184 a(ii,in,in-nan)=tpop(ii,in-nan,in)/pop(j,in)
c
c male-female metabolic weight, kgfwt
     amw(k,in)=wt(k,in)**0.75
c male-female inhalation rate, m3/day
     rsrt(k,in)=1.4*amw(k,in)
c male-female blood volume, liters
     wtbl(k,in)=.07*wt(k,in)**0.94
c male-female total body water, liters; conditional depending on age and sex
     tbwk=.6111*wt(k,in)+.251
     if (j.eq.2) tbwk=0.511*wt(k,in)+1.244
     if (k.gt.18) tbwk=.7945*wt(k,in)-.0024*wt(k,in)**2-.0015*ak*wt(k
    1 ,in)
     if (j.eq.2.and.k.gt.18) tbwk=.6981*wt(k,in)-.0026*wt(k,in)**2-.001
    1 2*ak*wt(k,in)
c male -female crystalline bone fraction to total body depending on age
     bnfr=0.032
     if (k.gt.18) bnfr=0.041
c male-female work energy expenditure(American), Kcal/day, for moderate
c activity
     worke=12.0*exp(-(.693/70.)*(ak))*wt(k,in)
c same except at activity level(ACTI) different from 1.0
     workt=worke*acti(in)
c anabolic energy, Kcal/day, utilized for muscle development; conditional
c depending ratio between estimated and expected protein weight
     amusi=200.*(workt-worke)/worke
     if (protb(k,j)/prot1(k,j).gt.1.2) amusi=0.0
     if (protb(k,j)/prot1(k,j).lt.0.8) amusi=0.0
c routing to female and male sections
     go to (220,200), j
c
c female section
c
c age groups younger than 14 years excluded
 200 if (k.lt.14) go to 210
c daily intake of protein, grams
     protk=1.343*(1.-.0077*(ak-14.))*wt(k,in)
c daily intake of fat, grams
     fatik=1.676*(1.-.0077*(ak-14.))*wt(k,in)
c daily intake of carbohydrate, grams
     ch20k=4.965*(1.-.0077*(ak-14.))*wt(k,in)
```

```
c basal metabolic rate, Kcal/day
      bmk=26.7*(1.-.0063*(ak-14.))*wt(k,in)
c drinking water intake, liters/day
      drnkk=.000767*(1.-.00303*(ak-1.))*(bmk+workt)
c body surface area, cm2
      sak=288.*(1.-.00204*(ak-14.))*wt(k,in)
c body height, cm
      hlnk=hln13*(1.-(ak-13.)*.00047)
c body cell mass, kgfwt
      bcmk=.4861*(1.-.0053*(ak-14.))*wt(k,in)
c heat emissivity for body heat flux estimates
      eta=(.0148-(.00021*(ak-14.)))*wt(k,in)*.88/.905
      go to 240
c estimates as above for female age groups less than 14 years old
  210 protk=4.1*exp(-(.693/10.)*ak)*wt(k,in)
      fatik=4.8*exp(-(.693/10.)*ak)*wt(k,in)
      ch20k=14.0*exp(-(.693/10.)*ak)*wt(k,in)
      bmk=54.18*exp(-(.693/11.85)*ak)*wt(k,in)
      drnkk=.0006731*(workt+bmk)
      sak=470.*exp(-(.693/20.5)*ak)*wt(k,in)
      bcmk=.35*exp((.693/24.)*ak)*wt(k,in)
      hlnk=5.5*exp(-(.693/18.5)*ak)*wt(k,in)
      if (k.eq.13) hln13=hlnk
      eta=(.78-.01*(ak-1.))
      go to 240
c
c male section
c
c age groups younger than 18 years excluded
  220 if (k.lt.18) go to 230
c daily protein intake, grams
      protk=1.386*(1.-.0019*(ak-18.))*wt(k,in)
c daily fat intake, grams
      fatik=1.786*(1.-.0019*(ak-18.))*wt(k,in)
c daily carbohydrate intake, grams
      ch20k=4.643*(1.-.0019*(ak-18.))*wt(k,in)
c basal metabolic rate, Kcal/day
      bmk=25.*(1.-.0038*(ak-18.))*wt(k,in)
c drinking water uptake, liters/day
      drnkk=.000767*(1.-.00303*(ak-1.))*(bmk+workt)
c body surface area, cm2
      sak=252.*wt(k,in)
c body height, cm
      hlnk=hln17*(1.-(ak-17.)*.00047)
c body cell mass, kgfwt
      bcmk=.58*(1.-.0019*(ak-18.))*wt(k,in)
c emissivity estimate for body heat flux estimates; conditional
      cs1=.01
      cs2=.00010
      cs3=18.
      if (j.eq.2) cs1=.0148
      if (j.eq.2) cs2=.00021
      if (j.eq.2) cs3=14.
      eta=(cs1-(cs2*(ak-cs3)))*wt(k,in)
```

```
      go to 240
c same for males younger than 18 years
 230 protk=4.1*exp(-(.693/13.5)*ak)*wt(k,in)
      fatik=4.5*exp(-(.693/13.5)*ak)*wt(k,in)
      ch20k=10.7*exp(-(.693/15.5)*ak)*wt(k,in)
      bmk=61.19*exp(-(.693/13.5)*ak)*wt(k,in)
      drnkk=.0006731*(workt+bmk)
      sak=470.*exp(-(.693/20.5)*ak)*wt(k,in)
      bcmk=.35*exp((.693/24.)*ak)*wt(k,in)
      hlnk=5.5*exp(-(.693/15.)*ak)*wt(k,in)
      if (k.eq.17) hln17=hlnk
      eta=.82-.01*(ak-1.)
 240 continue
c emissivity coefficients estimated for both sexes for ages greater than 32
c years of age
      cs1=.00855
      cs2=.000035
c female coefficients
      if (j.eq.2) cs1=.0109
      if (j.eq.2) cs2=.00011
c male-female emissivity  estimated
      if (k.gt.32) eta=(cs1+(cs2*(ak-33.)))*wt(k,in)
c female emissivities
      if (j.eq.2) eta=eta*0.905
c male-female daily urine production , liters/day
      urink=0.74*drnkk
c fourth power of body temperature, deg. C.**4
      t4b=(36.+273.)**4
c fourth power of environmental temperature, deg.C.**4
      t4e=(21.+273.)**4
c body surface heat loss, Kcal/cm2/day
      hflux=eta*8.132e-11*60.*24.*(t4b-t4e)/1000.
c body heat production, Kcal/cm2/day
      heatb=(.85*workt+bmk)/sak
c net heat gain/loss, Kcal/cm2/day; absolute value;bounded
      htdf=hflux-heatb
      errht=abs(htdf)
      if (errht/heatb.le..02) htdf=0.0
c feces water loss, ml/day
      fcwt=.072*urink*1000.
c insensible water loss from lungs, ml/day
      evpwt=.61*urink*1000.
c perspiration water loss, ml/day
      swtwt=.465*urink*1000.
c perspiration loss required to adjust heat loss, ml/day
      swta=-1000.*htdf*sak/580.
c maximum potential respiration loss above normal, ml/day
      swtc=swtwt*2.
c net perspiration loss, ml/day
      swtwt=swtwt+swta
      swtd=swtwt
c limits perspiration to twice normal loss, ml/day; bounded
      if (swtwt.gt.swtc) swtwt=swtc
      if (swtwt.le.0.0) swtwt=0.0
```

```
c net increase/decrease in perspiration volume, ml/day
    swta=swtwt-swtd
c heat loss residual, Kcal/cm2/day
    htdf=swta*580./(1000.*sak)
c daily urine losses, ml/day
    urnwt=urink*1000.
c total energy dissipated through perspiration and insensible lung losses,
c Kcal/day
    evpeng=(evpwt+swtwt)*(580./1000.)
c factor to adjust energy use due to work above moderate levels
    fcp=(workt-worke+htdf*sak)/(worke+bmk)
c protein energy losses to urine, Kcal/day
    upreng=(.01111*urink*1000.-2.)*2.5
c fat energy losses in feces, Kcal/day
    ufteng=.042*fatik*9.3
c carbohydrate energy losses to urine, Kcal/day
    ucheng=.001*urink*1000.*4.1
c protein, fat, and carbohydrate energy losses adjusted for work activity
c above moderate(ACTI=1.0)
    upreng=upreng+fcp*upreng
    ufteng=ufteng+fcp*ufteng
    ucheng=ucheng+fcp*ucheng
c total maintenance energy required by body, Kcal/day, including anabolic
c requirements(AMUSI), and energy im-balance(HTDF)
    toteng=heatb*sak+evpeng+upreng+ufteng+ucheng+0.15*workt
    toteng=toteng+amusi+htdf*sak
c total energy available from food intake, Kcal/day, from carbohydrate,
c protein, and fat
    fodeng=ch20k*4.1+protk*5.6+fatik*9.3
c carbohydrate,protein and fat intake adjusted for energy requirement,
c Kcal/day; energy from food re-calculated
    ch20k=ch20k*toteng/fodeng
    protk=protk*toteng/fodeng
    fatik=fatik*toteng/fodeng
    fodeng=ch20k*4.1+protk*5.6+fatik*9.3
c meat, grain, milk, and leafy vegetable intake, grams(ml for milk)/day
    gmeat=0.0
    ggrain=0.0
    gmilk=0.0
    glveg=0.0
c meat, carbohydrate, and fat intake, gfwt/day, from ingested food
c initialized
    bmeat=0.0
    bchoh=0.0
    bfat=0.0
c caloric content of ingested food, Kcal/day, initialized
    pteng=0.0
c leafy vegetable, grain, and seeds(fruits) intake, gdwt/day, intialized
    ulv=0.0
    ugrn=0.0
    usv=0.0
c leafy vegetable, grain, and seed(fruit) required intake, gfwt/day,
c initialized
    glvu=0.0
```

```
      ggrn=0.0
      gsvg=0.0
c total energy ingested food by standard man, Kcal/day
      engfod=0.0
c
c ** contaminant loop ******************************************
      do 260 l=1,inz
c contaminant intake from food, per day, initialized
      c3(l)=0.0
c
c ** region loop ******************************************************
      do 250 lz=1,ipx
c contaminant intake from drinking water and inhaled air initialized
      watcnh(l,lz)=0.0
      a1th(l,lz)=0.0
      a1thg(l,lz)=0.0
  250 continue
c ** end region loop ******************************************************
c contaminant intake from meat, milk, leafy vegetables, grains, and
c seeds(fruits), per day, initialized
      zmeat(l)=0.0
      zmilk(l)=0.0
      zlv(l)=0.0
      zgrn(l)=0.0
      zsv(l)=0.0
  260 continue
c ** end contaminant loop ******************************************
c excludes simulated food from BIOTRAN.2
      if (fedh) 411,410,411
c
c ** region loop ******************************************************
  411 do 280 lz=1,ipx
c
c ** contaminant loop ******************************************
      do 270 ll=1,inz
c
c contaminant intake, per day, set to input water concentration, per
c liter, and to simulated air concentration, per m3
c
      watcnh(ll,lz)=watcni(ll,lz)
      a1th(ll,lz)=a1tt(ll,lz)
      a1thg(ll,lz)=a1tg(ll,lz)
  270 continue
c ** end contaminant loop ******************************************
  280 continue
c ** end region loop ******************************************************
c standard man total energy intake, Kcal/day
      engfod=ameat*5.6+choh*4.1+fat*9.3
c total meat, grain, milk, leafy vegetable,and fresh seed(fruit) intake,
c gfwt(ml for milk)/day
      gmeat=275.*fodeng/engfod
      ggrain=379.*fodeng/engfod
      gmilk=527.*fodeng/engfod
      glveg=193.*fodeng/engfod
```

```
      gsveg=193.*fodeng/engfod
c protein, carbohydrate, and fat intake, gfwt/day, from ingested food
      bmeat=(gmeat*.19+ggrain*.095+gmilk*.032+glveg*.032+gsveg*.012)
      bchoh=(ggrain*.758+gmilk*.046+glveg*.037+gsveg*.03)
      bfat=(gmeat*.21+ggrain*.03+gmilk*.05+glveg*.003+gsveg*.001)
c total energy intake from food intake, Kcal/day
      pteng=bmeat*5.6+bchoh*4.1+bfat*9.3
c meat, grain, milk, leafy vegetable, and fresh seed(fruits) intake, gdwt(ml)
c for milk)/day, adjusted for required energy intake
      gmeat=gmeat*fodeng/pteng
      ggrain=0.88*ggrain*fodeng/pteng
      gmilk=gmilk*fodeng/pteng
      glveg=0.1*glveg*fodeng/pteng
      gsveg=0.1*gsveg*fodeng/pteng
c
c ** region loop ******************************************************
      do 380 lz=1,ipx
c fraction of food as grain, leafy vegetables, and fresh seeds(fruits)
      putgrt=0.0
      putlvg=0.0
      putsvg=0.0
c
c ** plant loop *******************************************************
      do 290 ll=1,ipl
c excludes range crops(used in ANIMAL subroutine only)
      if (putzd(in,ll,lz).le.3.) go to 290
      iputz=putzd(in,ll,lz)-3.
c sums utilization fraction by plant(LL) and region(LZ) for types above
      if (iputz.eq.1) putlvg=putlvg+putfrc(in,ll,lz)
      if (iputz.eq.2) putgrt=putgrt+putfrc(in,ll,lz)
      if (iputz.eq.4) putsvg=putsvg+putfrc(in,ll,lz)
  290 continue
c ** end plant loop ***************************************************
c number of individuals age group by sex
      hmct=tpop(k,j,in)
      hmct1=hmct
      if(hmct.le.1.0)hmct=1.0
c
c ** plant loop *******************************************************
      do 370 ll=1,ipl
c excludes range plants
      if (putzd(in,ll,lz).le.3.) go to 370
      iputz=putzd(in,ll,lz)-3.
      go to (300,320,340,340), iputz
c excludes empty leafy vegetable food source
  300 if (putlvg) 361,360,361
c required intake, gdwt/day, obtained from plant(LL) in region(LZ); bounded to
c available food(HRVST)
  361 glvu=glveg*putfrc(in,ll,lz)/putlvg
      if (glvu.ge.(hrvst(in,ll,lz)/hmct)) glvu=(hrvst(in,ll,lz)/hmct)
c
c ** contaminant loop *************************************************
      do 310 l=1,inz
c contaminant intake, per day, from leafy vegetables, all plants and
```

```
c regions;from internal and external contamination
      zlv(l)=zlv(l)+fsupf(in)*glvu*(spas(in,ll,l,lz)+0.5*spaa(in,ll,l,lz
     1 ))
  310 continue
c ** end contaminant loop ***********************************************
c leafy vegetable intake expressed as fresh weight
      ulv=ulv+glvu/.10
c available leafy vegetables adjusted for intake
      hrvst(in,ll,lz)=hrvst(in,ll,lz)-glvu*hmct1
c required leafy vegetables intake, gdwt/day, by age and sex
      totsf(in,ll,lz)=glvu
      go to 360
c excludes unavailable grain source
  320 if (putgrt) 362,360,362
c total grain required from plant(LL) in region(LZ), gdwt/day; bounded to
c available to available inventory(HRVST)
  362 ggrn=ggrain*putfrc(in,ll,lz)/putgrt
      if (ggrn.ge.(hrvst(in,ll,lz)/hmct)) ggrn=(hrvst(in,ll,lz)/hmct)
c
c ** contaminant loop ***************************************************
      do 330 l=1,inz
c contaminant intake from grain crops, per day, from both internal and
c surficial contamination
      zgrn(l)=zgrn(l)+fsupf(in)*ggrn*(spas(in,ll,l,lz)+0.25*spaa(in,ll,l
     1 ,lz))
  330 continue
c ** end contaminant loop ***********************************************
c grain crops intake expressed as gfwt/day
      ugrn=ugrn+ggrn/.88
c available grain inventory adjusted for consumption, gdwt
      hrvst(in,ll,lz)=hrvst(in,ll,lz)-ggrn*hmct1
c required grain intake, gdwt/day, by age and sex
      totsf(in,ll,lz)=ggrn
      go to 360
c excludes non-utilized fresh seed(fruit) food source
  340 if (putsvg) 363,360,363
c required seed(fruit) uptake, gdwt/day from plant(LL) inregion(LZ); bounded
  363 gsvg=gsveg*putfrc(in,ll,lz)/putsvg
      if (gsvg.ge.(hrvst(in,ll,lz)/hmct)) gsvg=(hrvst(in,ll,lz)/hmct)
c
c ** contaminant loop ***************************************************
      do 350 l=1,inz
c contaminant intake from seeds(fruits), per day, from both internal and
c surficial contamination; bounded
      zsv(l)=zsv(l)+fsupf(in)*gsvg*(spas(in,ll,l,lz)+0.20*spaa(in,ll,l
     1 ,lz))
      if (zsv(l).le.0.0) zsv(l)=0.0
  350 continue
c ** end contaminant loop ***********************************************
c seeds(fruits) expressed as fresh weight
      usv=usv+gsvg/.10
c available seeds(fruits) adjusted for consumption losses
      hrvst(in,ll,lz)=hrvst(in,ll,lz)-gsvg*hmct1
c required seeds(fruit) intake, gdwt/day, by age and sex
```

```
      totsf(in,ll,lz)=gsvg
  360 continue
  370 continue
c ** end plant loop ******************************************************
  380 continue
c ** end region loop *****************************************************
c selects beef cows as source of meat
      imt=1
c defaults to dairy cows as source of meat if beef cows not simulated
      if (rr(1)) 382,381,382
  381 imt=2
c bounds meat consumption, gfwt/day, to available amount(W)
  382 if(gmeat*p5(in).ge.wms(imt,1)*1000./hmct)
     1 gmeat=wms(imt,1)*1000./hmct
c
c ** contaminant loop ***************************************************
      do 390 l=1,inz
c contaminant intake, per day, based on contaminant concentration(W),
c and adjusted for fraction of meat consumed from study area(P5); bounded
      zmeat(l)=(gmeat/1000.)*w(13,imt,l)*p5(in)
      if (zmeat(l).le.0.0) zmeat(l)=0.0
  390 continue
c ** end contaminant loop ***********************************************
c available meat inventory(W) adjusted for food consumption by age group
      wms(imt,1)=wms(imt,1)-gmeat*hmct1/1000.
c sets dairy cows as source of milk
      imk=2
c defaults to beef cows as source of milk if dairy cows not simulated
      if (rr(2)) 392,391,392
  391 imk=1
c bounds milk consumption, ml/day, to available milk inventory(W)
  392 if (gmilk.ge.wmk(imk)*1000./hmct) gmilk=wmk(imk)*1000./hmct
c
c ** contaminant loop ***************************************************
      do 400 l=1,inz
c contaminant intake, per day, from milk based on contaminant concentration,
c and for fraction of milk consumed from the study area
      zmilk(l)=(gmilk/1000.)*w(18,imk,l)*p5(in)
c total contaminant intake, per day, from all sources of food
      c3(l)=zlv(l)+zsv(l)+zgrn(l)+zmeat(l)+zmilk(l)
  400 continue
c ** end contaminant loop ***********************************************
c protein, carbohydrate, and fat intake, gdwt/day, from all food sources
      protk=gmeat*.19+ugrn*.095+gmilk*.032+ulv*.032+usv*.012
      ch20k=ugrn*.758+gmilk*.046+ulv*.037+usv*.03
      fatik=gmeat*.21+ugrn*.03+gmilk*.05+ulv*.003+usv*.001
c total food weight ingested, gdwt/day
  410 fodwt=ch20k+protk+fatik
c total water intake, ml/day, from drinking water and that contained in food
      watrin=drnkk*1000.+2.04*fodwt
c net water balance (loss/gain)
      watnet=watrin-urnwt-fcwt-evpwt-swtwt
c adjusted urine loss, ml/day, based on negative water balance; bounded to
c 66% of normal urine output
```

```
      urina=0.0
      if (watnet.lt.0.0) urina=-watnet
      if (urina.gt.0.66*urnwt) urina=0.66*urnwt
c urine loss is decreased if water balance negative
      urnwt=urnwt-urina
c urine loss expressed as liters/day
      urink=urnwt/1000.
c water balance re-determined after accounting for urine output loss
      watnet=watnet+urina
c drinking water consumption, liters/day, adjusted for water balance; bounded
      drnkk=drnkk-watnet/1000.
      if (drnkk.le.0.0) drnkk=0.0
c number of individuals in age group(K) of sex(J)
      zott=tpop(k,j,in)
c
c ** region loop ******************************************************
      do 430 lz=1,ipx
c
c ** contaminant loop *************************************************
      do 420 l=1,inz
      if(lz.eq.1)c56(l)=0.0
c total contaminant intake, per day, from drinking water from all regions
      wburd(l)=1000.*drnkk*watcnh(l,lz)*rfrz(in,lz)*sptfrf(lz)+wburd(l)
c same for region(lz)
      c56z=1000.*drnkk*watcnh(l,lz)*rfrz(in,lz)*sptfrf(lz)
c total contaminant entering blood from ingested drinking water
      c56(l)=c56(l)+c56z*gtbl(in,l)
  420 continue
c ** end contaminant loop *********************************************
  430 continue
c ** end region loop ******************************************************
c extracellular water, liters based on body weight(HWT)
      ecwk=0.24*wt(k,in)
c intracellular water, liters, based on total body water(TBWK)
      aicwk=tbwk-ecwk
c essential fat, grams, in membranes and cells
      essfat=bcmk*.11
c other non-essential fat based ont total fat content(FATB)
      ofat=fatb(k,in)-essfat
c initial daily protein, fat, and carbohydrate increment, grams/day, for
c 1 year-olds
      prtdf=.2427*wt(k,in)*.60
      fatdf=.1657*wt(k,in)*.60
      chodf=.0074*wt(k,in)*.60
c 1 year-olds excluded
      if (k.eq.1) go to 470
c individuals greater than 10 years of age excluded
      if (k.gt.10) go to 440
c
c 1-10 year-olds
c
      cx=0.5
c female coefficient adjusted
      if (j.eq.2) cx=0.66
```

```
c adjusted protein,fat,and carbohydrate increments
      prtdf=.1781*exp(-(.693/6.)*(ak-1.))*wt(k,in)*cx
      fatdf=.1254*exp(-(.693/10.)*(ak-1.))*wt(k,in)*cx
      chodf=.0057*exp(-(.693/7.)*(ak-1.))*wt(k,in)*cx
      go to 470
c
c individuals older than 10 years
c
c  age index
  440 icx=17
c female index adjusted
      if (j.eq.2) icx=12
c excludes males older than 17 years and females older than 12 years
      if (k.gt.icx) go to 450
      cx=0.7
c female coefficient adjusted
      if (j.eq.2) cx=1.3
c adjusted protein,fat, and carbohydrate increments, g/day
      prtdf=.1043*exp(-(.693/6.)*(ak-11.))*wt(k,in)*cx
      fatdf=.0996*exp(-(.693/5.)*(ak-11.))*wt(k,in)*cx
      chodf=.0031*exp(-(.693/7.)*(ak-11.))*wt(k,in)*cx
      go to 470
c excludes males and females older than 32 years
  450 if (k.gt.32) go to 460
c
c 18-32 years males, 13-32 years females
c
c male age index and growth coefficient
      cxx=18.
      cx=0.8
c female age index and growth coefficient
      if (j.eq.2) cxx=12.
      if (j.eq.2) cx=2.8
c adjusted protein, fat, and carbohydrate increments, g/day
      prtdf=.00248*exp(-(.693/13.)*(ak-cxx))*wt(k,in)*cx
      fatdf=.0021*exp((.693/20.)*(ak-cxx))*wt(k,in)*cx
      chodf=.0000768*exp(-(.693/14.)*(ak-cxx))*wt(k,in)*cx
      go to 470
c
c males-females over 33 years old
c
c male coefficents
  460 cx=2.0
      cx2=.0003
c female coefficents
      if (j.eq.2) cx=1.8
      if (j.eq.2) cx2=.004
c adjusted protein, fat, and carbohydrate increments, g/day
      prtdf=-.00037*wt(k,in)*cx
      fatdf=cx2*wt(k,in)*cx
      chodf=-.000012*wt(k,in)*cx
  470 continue
c protein increment adjusted for anabolic muscle growth, g/day
      prtdf=prtdf+amusi/5.6
```

```
c available protein ,fat, and carbohydrate energy, Kcal/day, after adjustment
c for losses in urine and feces, and for anabolic processes
      apreng=protk*5.6-upreng-prtdf*5.6
      afteng=fatik*9.3-ufteng-fatdf*9.3
      acheng=ch20k*4.1-ucheng-chodf*4.1
c daily maintenance energy requirements, Kcal/day, including basal metabolic
c rate(BMK), work, evaporation,and surface heat losses
      uteng=bmk+workt+evpeng+htdf*sak
c surplus energy from food, Kcal/day
      enrgy=apreng+acheng+afteng
c fat, carbohydrate, and protein body compartments incremented, kg
      fatb(k,in)=fatb(k,in)+fatdf/1000.
      ch20b(k,j)=ch20b(k,j)+chodf/1000.
      protb(k,j)=protb(k,j)+prtdf/1000.
c net energy balance between surplus and maintenance energy requirements
      reseng=enrgy-uteng
c excludes negative and positive balances less than or equal to 15 Kcal/day
      absval=abs(reseng)
      if (absval.le.15.) reseng=0.0
c excludes 15 Kcal/day positive or negative energy balances
      if (reseng) 491,490,491
c excludes positive balances above 15 Kcal/day
  491 if (reseng.gt.0.0) go to 480
c residual energy need, Kcal/day above 15 Kcal/day
      reseng=reseng+15.
      engres=reseng
c fat stores mobilized to produce energy, Kcal/day, equal to deficit; bounded
c to essential fat content
      fatb(k,in)=fatb(k,in)+engres/(9.3*1000.)
      if (fatb(k,in).lt.essfat) engres=1000.*9.3*(fatb(k,in)-
     1essfat)
      if (fatb(k,in).lt.essfat) fatb(k,in)=essfat
c adjusted residual energy need, Kcal/d;
      reseng=reseng-engres
c excludes satisfied energy need
      if (reseng.gt.0.0) reseng=0.0
      if (reseng) 492,490,492
c residual energy need, Kcal/day
  492 engres=reseng
c carbohydrate stores mobilized to produce energy, Kcal/day, equal to
c deficit; bounded to available amount
      ch20b(k,j)=ch20b(k,j)+engres/(1000.*4.1)
      if (ch20b(k,j).lt.0.0) engres=ch20b(k,j)*1000.*4.1
      if (ch20b(k,j).lt.0.0) ch20b(k,j)=0.0
c adjusted residual energy need, Kcal/day, excludes satisfied energy need
      reseng=reseng-engres
      if (reseng.gt.0.0) reseng=0.0
      if (reseng) 493,490,493
c residual energy need, Kcal/day
  493 engres=reseng
c protein in soft tissue mobilized to produce energy, Kcal/day, equal to
c deficit; bounded to available amout
      protb(k,j)=protb(k,j)+engres/(1000.*4.1)
      if (protb(k,j).lt.0.0) engres=protb(k,j)*1000.*5.6
```

```
      if (protb(k,j).lt.0.0) protb(k,j)=0.0
c adjusted residual energy need, Kcal/day, excludes satisfied energy need
      reseng=reseng-engres
      if (reseng.gt.0.0) reseng=0.0
      if (reseng) 494,490,494
c print *,'individual DIES'
  494 call exit
c surplus energy available above 15 Kcal/day
  480 reseng=reseng-15.
c fat stores incremented ,kg, equal to energy surplus
      fatb(k,in)=fatb(k,in)+reseng/(1000.*9.3)
c total body weight increase, Kg, from protein, fat, and carbohydrate
c increases, kg/day
  490 grwth=prtdf/1000.+chodf/1000.+fatdf/1000.
c total body water ratio to to total weight
      tbwr=tbwk/wt(k,in)
c total body weight increase adjusted for total body water and bone fraction,
c kg (de-watered protein, fat, and carbohydrate increase)
      wt(k,in)=wt(k,in)+grwth/(1.-tbwr-bnfr)
c total body water, liters
      tbwk=tbwr*wt(k,in)
      tbwpr=tbwk
c water requirement, ml/day, required for body water increase
      dtbw=(tbwk-tbwpr)*1000.
c water balance adjusted for total body water increase
      if (dtbw.gt.0.0) watnet=watnet-dtbw
      wtgrm=wt(k,in)*1000.
c weight of blood, kg
      wtbl(k,in)=5.5e-05*wtgrm**0.99
c weight of bone, kg
      wtbn(k,in)=5.5e-05*wtgrm
c weight of kidneys, kg
      wtkd(k,in)=2.12e-05*wtgrm**0.85
c weight of liver, kg
      wtlv(k,in)=8.2e-05*wtgrm**0.87
c weight of lung/trachae
      wtlu(k,in)=1.24e-05*wtgrm**0.99
c weight of muscle, kg
      wtms(k,in)=4.5e-04*wtgrm
c inhalation rate, m3/day
      rsrt(k,in)=1.4*amw(k,in)
c weight of residual soft tissue, kg
      wtrs(k,in)=wt(k,in)-wtbl(k,in)-wtbn(k,in)-wtkd(k,in)-
     1wtlv(k,in)-wtms(k,in)-wtlu(k,in)-fatb(k,in)
c cardiac output, liters/day
      cardrt=1.44*0.7081*wtgrm**0.79
c renal flow, liters/day
      renflo=1.44*0.2109*wtgrm**0.77
c milk volume production in mature female
      flomlk=1.915e-03*wtgrm**0.7
c
c** compartment volumes **
c
c body weight, grams
```

```
      wtgrm=wt(k,in)*1000.
c
c plasma vol, liters
      volpls=4.29e-05*wtgrm**0.992
c muscle vol, liters
      volmus=4.63e-04*wtgrm**1.009
c kidney vol, liters
      volkid=2.18e-05*wtgrm**0.843
c liver vol, liters
      volliv=8.59e-05*wtgrm**0.885
c gut vol, liters
      volgut=8.65e-05*wtgrm**0.909
c gut lumen vol, liters
      vollum=5.78e-05*wtgrm**0.934
c heart vol, liters
      volhrt=2.0e-06*wtgrm**1.043
c lungs vol, liters
      vollun=3.16e-06*wtgrm**1.104
c spleen vol, liters
      volspl=4.52e-06*wtgrm**0.901
c marrow vol, liters
      volmar=3.3e-05*wtgrm**0.936
c fat vol, liters
      volfat=fatb(k,in)/0.9
c current residual volume
      volres=volgut+volhrt+volspl
c
c ** plasma flow rates **
c
c plasma flow, liters/day
      flopls=1.44*.601*wtgrm**0.778
c muscle flow, liters/day
      flomus=1.44*.0817*wtgrm**0.81
c kidney flow, liters/day
      flokid=1.44*0.0845*wtgrm**0.802
c liver flow, liters/day
      floliv=1.44*0.0944*wtgrm**0.792
c gut flow, liters/day
      flogut=1.44*0.102*wtgrm**0.787
c heart flow, liters/day
      flohrt=1.44*5.77e-03*wtgrm**0.965
c lung flow, liters/day
      flolun=1.44*0.215*wtgrm**0.82
c spleen flow, liters/day
      flospl=1.44*1.79e-03*wtgrm**1.028
c marrow flow, liters/day
      flomar=1.44*0.0164*wtgrm**0.804
c fat flow, liters/day
      flofat=0.75*flomus
c residual tissue flow, liters/day
      flores=flogut+flohrt+flospl
c urine production, liters/day
      flourn=1.44*.0064*wtgrm**0.82
c
```

```
c mean air concentration, per m3, all regions initialized
      a1thz=0.0
      a1thzg=0.0
c number of regions simulated
      dpx=ipx
c
c ** contaminant loop ***********************************************
      do 510 l=1,inz
c
c ** region loop ****************************************************
      do 500 lz=1,ipx
c mean regional air concentration
      a1thz=a1thz+a1th(l,lz)*sptfrf(lz)/dpx
      a1thzg=a1thzg+a1thg(l,lz)*sptfrf(lz)/dpx
  500 continue
c ** end region loop ************************************************
  510 continue
c ** end contaminant loop ********************************************
c
c total human milk production, liters, from producing female age group
c excludes females younger than 18 years, and older than 40 years
      if (j.eq.1)goto703
      if(k.lt.18)wmk(in)=0.0
      if(k.gt.17.and.k.lt.41) wmk(in)=wmk(in)+flomlk*tpop(k,j,in)
c ** contaminant loop ***********************************************
  703 do 570 l=1,inz
      c99(l)=0.0
      hl4(l)=0.0
      car(l)=0.0
      hl5(l)=0.0
      c66(l)=0.0
      if(agem(ii,in).le.0.0)goto570
      if(ii.gt.1)goto707
c
c gestation stage still active
      if(aiiir(in,1).gt.aiii)goto 704
      if(gstn(in).lt.2.0)goto707
      if(aiiir(in,2).gt.aiii)goto704
c**** mean daily human milk concentration initialized
  707 if(ii.eq.1) w(18,in,l)=0.0
c
c contaminant deposition, per day, in lung
      pburd(l)=.25*a1thz*rsrt(k,in)*(plcm1(in)+plcm2(in)+plcm3(in)+1.52)
c N-P and T-B contaminant deposition, per day
      gburd(l)=rsrt(k,in)*a1thz*0.38
c N-P, T-B, and P lung burdens incremented for deposition, and corrected for
c radioactive decay if required andN-P, T-B, and P lung burdens adjusted
c for losses to GI and blood(Q3)
      q1(k,in,l)=q1(k,in,l)+plcm1(in)*a1thz*rsrt(k,in)-disint(l)
     1*q1(k,in,l)-q1(k,in,l)*pltg1(in,l)
      q2(k,in,l)=q2(k,in,l)+plcm2(in)*a1thz*rsrt(k,in)-disint(l)
     1*q2(k,in,l)-q2(k,in,l)*pltg2(in,l)
      q3(k,in,l)=q3(k,in,l)+plcm3(in)*a1thz*rsrt(k,in)-disint(l)
     1*q3(k,in,l)-q3(k,in,l)*pltbl(in,l)
```

```
c
      if(q1(k,in,l).le.0.0)q1(k,in,l)=0.0
      if(q2(k,in,l).le.0.0)q2(k,in,l)=0.0
      if(q3(k,in,l).le.0.0)q3(k,in,l)=0.0
c blood contaminant uptake, per day, from N-P and T-B regions via GI
      hl4(l)=q1(k,in,l)*(pltg1(in,l)+q2(k,in,l)*pltg2(in,l))*gtbl(in,l)
c contaminant vapor exchange between air and venous blood weighted by
c regional occupancy
      qae=rsrt(k,in)*a1thzg+flopls*v(k,in,l)
c
      qbe=flopls+pltblg(in,l)*rsrt(k,in)*1000.
c arterial blood concentration estimated after contaminant exchange
      if(pltblg(in,l).gt.0.0)v(k,in,l)= qae/qbe
      if(v(k,in,l).le.0.0)v(k,in,l)=0.0
c contaminant uptake from ingested plant foods
      vpln(k,in,l)=c3(l)/gtbl(in,l)
c contaminant uptake from ingested soil consumption(set to 0.0)
      vsol(k,in,l)=c66(k)/gtbl(in,l)
c contaminant consumption via clearance from respiratory tract to gut
      vlug(k,in,l)=(car(l)+hl4(l))/gtbl(in,l)
c blood contaminant uptake from ingested food, per day
      c99(l)=c3(l)*gtbl(in,l)
c total gut ingestion: all routes
      vgut(k,in,l)=(car(l)+hl4(l)+c99(l)+c56(l)+c66(l))/gtbl(in,l)
c
c blood contaminant uptake from P lung directly to pulmonary blood
      hl5(l)=q3(k,in,l)*pltbl(in,l)
      v(k,in,l)=v(k,in,l)+hl5(l)
c
c daily contaminant transport to arterial blood from pulmonary region
c and for correction for radioactive decay or persistence
      car(l)=rsrt(k,in)*a1thz*.38*gtbl(in,l)
      v(k,in,l)=v(k,in,l)-disint(l)*v(k,in,l)
      if(v(k,in,l).le.0.0)v(k,in,l)=0.0
c
c total milk production liters/day , producer age groups
      wmk(in)=0.0
      agemi=agem(ii,in)
c estimates mature age fraction of 1 year-olds based on 3 sub-divisions
      fracs=a(ii,in,j)
      if(ii.ne.1)goto712
      off2t=0.0
      off2i=0.0
      do713ig1=1,3
      off2t=off2t+offsp(in,ig1,j)
      if(affsp(in,ig1).ge..15*alsp(in).and.affsp(in,ig1).le..7*alsp(in))
     1off2i=off2i+offsp(in,ig1,j)
  713 continue
      fracs=0.0
      if(off2t.gt.0.0) fracs=a(ii,in,j)*off2i/off2t
  712 if(agemi.ge.0.15*alsp(in).and.agemi.le.0.7*alsp(in))
     1wmk(in)=wmk(in)+flomlk*fracs
      if(j.eq.1)wmk(in)=0.0
c
```

```
c ** region loop ****************************************************
      do 520 lz=1,ipx
c humus sink adjusted for unabsorbed contaminant, per m2, from above,
c and from blood(VH) losses via urine(BLTUR)
      humus(in,l,lz)=humus(in,l,lz)+(1.0-gtbl(in,l))*vgut(k,in,l)
     1*rfrz(in,lz)*zott*sptfrf(lz)/agr(in,lz)
      humus(in,l,lz)=humus(in,l,lz)+vurin(k,in,l)*rfrz(in,lz)*
     1zott*sptfrf(lz)/agr(in,lz)
  520 continue
c ** end region loop ************************************************
c total contaminant intake, per day, from all sources: milk, meat, grains,
c fruits (seeds), leafy vegetables, drinking water, and lungs
      vintk(k,j,l)=zmilk(l)+zmeat(l)+zgrn(l)+zsv(l)+zlv(l)+wburd(l)
     1 +pburd(l)+gburd(l)
c cumulative daily contaminant intake for leafy vegetables, grains,
c fruits(seeds), meat, milk, water, and lungs(2)
      zlvi(k,j,l)=zlvi(k,j,l)+zlv(l)
      zgrni(k,j,l)=zgrni(k,j,l)+zgrn(l)
      zsvi(k,j,l)=zsvi(k,j,l)+zsv(l)
      zmeati(k,j,l)=zmeati(k,j,l)+zmeat(l)
      zmilki(k,j,l)=zmilki(k,j,l)+zmilk(l)
      wburi(k,j,l)=wburi(k,j,l)+wburd(l)
      pburi(k,j,l)=pburi(k,j,l)+pburd(l)
      gburi(k,j,l)=gburi(k,j,l)+gburd(l)
c total contaminant uptake, per day, from all sources entering blood
      vuptk(k,j,l)=hl4(l)+hl5(l)+car(l)+c99(l)+c56(l)
c total contaminant loss to feces from all sources of intake,  per day
      vfec(k,j,l)=(hl4(l)+hl5(l)+car(l)+c99(l)+c56(l))
     1*(1.-gtbl(in,l))
c
      viiin(l)=v(k,in,l)
      dbo=0.0
      vmk(k,in,l)=0.0
      vurin(k,in,l)=0.0
      do719it=1,24
c arterial perfusion of organs with contaminant
c daily bone(2) contaminant content updated for gains from arterial
c blood
      dbn2=0.0
      if(bntbl2(in,l).le.0.0)goto333
      f0=2.0*flomar
      f2=viiin(l)/volpls
      f3=vbn2(k,in,l)/(volmar*bntbl2(in,l))
      dbn2=f0*(f2-f3)/24.
c same for bone(1) compartment
c
  333 dbn1=0.0
      if(bntbl1(in,l).le.0.0)goto334
      f0=2.0*flomar
      f2=viiin(l)/volpls
      f3=vbn1(k,in,l)/(volmar*bntbl1(in,l))
      dbn1=f0*(f2-f3)/24.
      dbo=f0*f2/24.
c
```

```
c same for muscle contaminant content
  334 dbms=0.0
      if(mstbl(in,l).le.0.0)goto335
      f0=flomus
      f2=viiin(l)/volpls
      f3=vms(k,in,l)/(volmus*mstbl(in,l))
      dbms=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c same for kidney(2) compartment
  335 dbk2=0.0
      if(kdtbl2(in,l).le.0.0)goto336
      f0=flokid
      f2=viiin(l)/volpls
      f3=vkd2(k,in,l)/(volkid*kdtbl2(in,l))
      dbk2=f0*(f2-f3)/24.
c
c adipose tissue content updated for gains from arterial blood
  336 dbft=0.0
      if(fttbl(in,l).le.0.0)goto337
      f0=flofat
      f2=viiin(l)/volpls
      f3=vfat(k,in,l)/(volfat*fttbl(in,l))
      dbft=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c residual soft tissue content updated for gain from arterial blood
c and losses to mammary glands and/or milk production
  337 dbrs=0.0
      dbmk=0.0
      if(rstbl(in,l).le.0.0)goto338
      f0=flores
      f2=viiin(l)/volpls
      f3=vr(k,in,l)/(volres*rstbl(in,l))
      dbrs=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltmk(in,l).le.0.0)goto338
      f2=viiin(l)*bltmk(in,l)
      dbmk=f0*f2/(flopls*24.)
      if(vmk(k,in,l).le.0.0)dbmk=0.0
      if(dbmk.le.0.0)dbmk=0.0
c
c kidney(1) contaminant content updated for gains from arterial
c blood and losses to urine
  338 dbk1=0.0
      dbu=0.0
      if(kdtbl1(in,l).le.0.0)goto339
      f0=flokid
      f2=viiin(l)/volpls
      f3=vkd1(k,in,l)/(volkid*kdtbl1(in,l))
      dbk1=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltur(in,l).le.0.0)goto339
      f2=viiin(l)*bltur(in,l)
```

```
      dbu=f0*f2/(flopls*24.)
      if(dbu.le.0.0)dbu=0.0
      if(alsp(j).le.0.0.and.ainsm.le.0.0)dbu=0.0

c
c liver contaminant content updated for gains from arterial blood
c and gut absorption
  339 dblv=0.0
      dbi=0.0
      if(lvtbl(in,l).le.0.0)goto332
      cntki=1.0
      if(l.ne.1)goto717
      cntki=cntk(1)
      if(gstn(j).lt.2.0)goto717
      if(aiiir(j,2).gt.cagm(j,2))cntki=(cntk(1)+cntk(2))/2.
  717 dbi=(al4(l)+car(l)+cntki*(c99(l)+c66(l)+c56(l)))/24.
      if(l.eq.1)dbi=dbi+cbw(in,l)/24.
      f0=floliv
      f2=viiin(l)/volpls
      vlv(k,in,l)=vlv(k,in,l)+dbi
      f3=vlv(k,in,l)/(volliv*lvtbl(in,l))
      dblv=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(dbo.le.0.0)goto719
      flxbn2=dbn2*viiin(l)/dbo
      brto=0.99*vbn2(k,in,l)
      if(flxbn2.lt.0.0.and.-flxbn2.gt.brto)flxbn2=-brto
      vbn2(k,in,l)=vbn2(k,in,l)+flxbn2
      viiind=viiin(l)-flxbn2
      if(viiind.le.0.0)viiind=0.0
      if(vbn2(k,in,l).le.0.0)vbn2(k,in,l)=0.0
      flxbn1=dbn1*viiin(l)/dbo
      brto=0.99*vbn1(k,in,l)
      if(flxbn1.lt.0.0.and.-flxbn1.gt.brto)flxbn1=-brto
      if(bntbl2(in,l).le.0.0)viiind=viiin(l)
      vbn1(k,in,l)=vbn1(k,in,l)+flxbn1*viiind/viiin(l)
      if(flxbn2.lt.0.0)vbn1(k,in,l)=vbn1(k,in,l)-flxbn2
      if(flxbn2.lt.0.0)flxbn2=0.0
      if(vbn1(k,in,l).le.0.0)vbn1(k,in,l)=0.0
      flxms=dbms*viiin(l)/dbo
      brto=0.99*vms(k,in,l)
      if(flxms.lt.0.0.and.-flxms.gt.brto)flxms=-brto
      vms(k,in,l)=vms(k,in,l)+flxms
      if(vms(k,in,l).le.0.0)vms(k,in,l)=0.0
      flxkd2=dbk2*viiin(l)/dbo
      brto=0.99*vkd2(k,in,l)
      if(flxkd2.lt.0.0.and.-flxkd2.gt.brto)flxkd2=-brto
      vkd2(k,in,l)=vkd2(k,in,l)+flxkd2
      viiind=viiin(l)-flxkd2
      if(viiind.le.0.0)viiind=0.0
      if(vkd2(k,in,l).le.0.0)vkd2(k,in,l)=0.0
      flxkd1=dbk1*viiin(l)/dbo
      brto=0.99*vkd1(k,in,l)
      if(flxkd1.lt.0.0.and.-flxkd1.gt.brto)flxkd1=-brto
```

```
      if(kdtbl2(in,l).le.0.0)viiind=viiin(l)
      vkd1(k,in,l)=vkd1(k,in,l)+flxkd1*viiind/viiin(l)
      if(flxkd2.lt.0.0)vkd1(k,in,l)=vkd1(k,in,l)-flxkd2
      if(flxkd2.lt.0.0)flxkd2=0.0
      if(vkd1(k,in,l).le.0.0)vkd1(k,in,l)=0.0
      flxft=dbft*viiin(l)/dbo
      brto=0.99*vfat(k,in,l)
      if(flxft.lt.0.0.and.-flxft.gt.brto)flxft=-brto
      vfat(k,in,l)=vfat(k,in,l)+flxft
      if(vfat(k,in,l).le.0.0)vfat(k,in,l)=0.0
      flxrs=dbrs*viiin(l)/dbo
      brto=0.99*vr(k,in,l)
      if(flxrs.lt.0.0.and.-flxrs.gt.brto)flxrs=-brto
      vr(k,in,l)=vr(k,in,l)+flxrs
      if(vr(k,in,l).le.0.0)vr(k,in,l)=0.0
      vmk(k,in,l)=vmk(k,in,l)+dbmk
      vurin(k,in,l)=vurin(k,in,l)+dbu
      flxlv=dblv*viiin(l)/dbo
      flxlvp=flxlv
      brto=0.99*vlv(k,in,l)
      if(flxlv.lt.0.0.and.-flxlv.gt.brto)flxlv=-brto
      vlv(k,in,l)=vlv(k,in,l)+flxlv
      if(vlv(k,in,l).le.0.0)vlv(k,in,l)=0.0
      v(k,in,l)=v(k,in,l)+cbp(k,in,l)/24.-flxbn1-flxbn2-flxkd1-flxkd2
     1-flxft-flxrs-flxlv-flxms-dbu-dbmk*(1.-cblc)-cblc*dbmk/
     2(0.1*wt(k,in)/flomlk)
      if(v(k,in,l).le.0.0)v(k,in,l)=0.0
      viiin(l)=v(k,in,l)
  719 continue

c
c bone(1), bone(2), liver, muscle, kidney(1), and kidney(2) corrected for
c radioactive decay or persistence
  332 vbn1(k,in,l)=vbn1(k,in,l)-disint(l)*vbn1(k,in,l)
      vbn2(k,in,l)=vbn2(k,in,l)-disint(l)*vbn2(k,in,l)
      vlv(k,in,l)=vlv(k,in,l)-disint(l)*vlv(k,in,l)
      vms(k,in,l)=vms(k,in,l)-disint(l)*vms(k,in,l)
      vkd1(k,in,l)=vkd1(k,in,l)-disint(l)*vkd1(k,in,l)
      vkd2(k,in,l)=vkd2(k,in,l)-disint(l)*vkd2(k,in,l)
      vfat(k,in,l)=vfat(k,in,l)-disint(l)*vfat(k,in,l)
      vr(k,in,l)=vr(k,in,l)-disint(l)*vr(k,in,l)
c

c
c daily blood contaminant content updated for losses;transport to fetal
c blood assumed equivalent to blood transport to milk
      vmkc=0.0
c** excludes immature or old age female mammals from milk production
      agemi=agem(k,in)
      if(k.ne.1)goto714
      if(iage.ge.6)goto714
      off2i=0.0
      do715ig1=1,3
      if(affsp(in,ig1).ge..15*alsp(in).and.affsp(in,ig1).le..7*alsp(in))
```

```
      1off2i=off2i+offsp(in,ig1,j)
  715 continue
      if(off2i.gt.0.0.and.agemi.lt.0.16*alsp(in))agemi=0.16*alsp(in)
c
  714 if(agemi.lt.0.15*alsp(in).or.agemi.gt.0.7*alsp(in))goto716
      vmkc=1.0
c ** excludes milk production during non-reproductive period
      if(aiiir(in,1).lt.gst(in,1)+14.)vmkc=0.0
c ** excludes milk production after weaning period
      if(aiiir(in,1).gt.cagm(in,1)+90.) vmkc=0.0
      if(gstn(in).lt.2.0)goto716
      if(aiiir(in,2).le.cagm(in,1)+90.)goto716
      vmkc=1.0
      if(aiiir(in,2).lt.gst(in,2)+14.)vmkc=0.0
      if(aiiir(in,2).gt.cagm(in,2)+90.)vmkc=0.0
  716 vmk(k,in,l)=vmkc*vmk(k,in,l)
c factor for excluding immature or older females as milk producers and
c reproducers
  704 afmt=0.0
      lf=iage+1
      if(k.eq.1)gsfc=0.0
      if(k.eq.1)gsst=0.0
c
c** excludes non-reproductive period; allows 2 weeks for mating and
c** embryo development
      if(aiiir(in,1).lt.gst(in,1)+14.)cbf(in,l)=0.0
      if(aiiir(in,1).lt.gst(in,1)+14.)cbp(k,in,l)=0.0
      if(aiiir(in,1).lt.gst(in,1)+14.)goto702
      if(gstn(in).lt.2.0)goto801
      if(aiiir(in,2).le.cagm(in,1)+90.)goto801
      if(aiiir(in,2).lt.gst(in,2)+14.)cbf(in,l)=0.0
      if(aiiir(in,2).lt.gst(in,2)+14.)cbp(k,in,l)=0.0
      if(aiiir(in,2).lt.gst(in,2)+14.)goto702
c
c ** excludes period after weaning
  801 if(aiiir(in,1).le.cagm(in,1)+90.)goto802
      if(gstn(in).lt.2.0)cbw(in,l)=0.0
      if(gstn(in).lt.2.0)goto702
      if(aiiir(in,2).gt.cagm(in,2)+90.)cbw(in,l)=0.0
      if(aiiir(in,2).gt.cagm(in,2)+90.)goto702
c
c** excludes immature animals from milk production and reproduction
  802 agemi=agem(k,in)
      fracs=1.0
      if(k.ne.1)goto721
      off2t=0.0
      off2i=0.0
      do722ig1=1,3
      off2t=off2t+offsp(in,ig1,j)
      if(affsp(in,ig1).ge..15*alsp(in).and.affsp(in,ig1).le..7*alsp(in))
      1off2i=off2i+offsp(in,ig1,j)
  722 continue
      fracs=0.0
      if(off2t.gt.0.0) fracs=off2i/off2t
```

```
      if(fracs.gt.0.0.and.agemi.lt.0.16*alsp(in))agemi=0.16*alsp(in)
c
  721 if(agemi.le.0.15*alsp(in))goto702
      gsst=gsst+1.0
      lua(in)=0
      if(agem(k+1,in).ge.0.7*alsp(in))lua(in)=k
      if(k.eq.iage)lua(in)=k
      if(agemi.gt.0.7*alsp(in))gsfc=gsfc+1.0
      if(agemi.gt.0.7*alsp(in))goto902
c** obtains weighted reproducer fractions
c
      do711ia=1,iage
      ai=ia
      ifm(ia)=agemi-0.15*alsp(in)/(0.55*alsp(in)/9.0)
      if(ifm(ia).lt.1)ifm(ia)=1
      if(ifm(ia).gt.8)ifm(ia)=8
      if(ia.eq.1)ifa=ifm(ia)
      ifb=ifm(ia)
      afmt=afmt+sfd(ifb,in)*fracs
  711 continue
c
c** initiates fetus production during Spring/Fall time frame
c
c *** weighted parental contaminant addition to fetal blood ***
c
c *** changes transport from fetal blood to GI(liver) after birth
      cblc=1.0
      if(aiiir(in,1).gt.cagm(in,1)) cblc=0.0
      if(aiiir(in,1).gt.cagm(in,1)) aiif(in,1)=0.0
      if(gstn(in).lt.2.0)goto803
      if(aiiir(in,1).le.cagm(in,1)+90.)goto803
      cblc=1.0
      if(aiiir(in,2).gt.cagm(in,2)) cblc=0.0
      if(aiiir(in,2).gt.cagm(in,2)) aiif(in,2)=0.0
c
c contaminant transport to fetus blood
  803 if(cblc.le.0.0)v(lf,in,l)=0.0
      if(cblc.le.0.0)cbf(in,l)=0.0
      if(cblc.ge.1.0)cbw(in,l)=0.0
      if(gsst.gt.1.0)goto903
      v(lf,in,l)=0.0
      vbn1(lf,in,l)=0.0
      vbn2(lf,in,l)=0.0
      vkd1(lf,in,l)=0.0
      vkd2(lf,in,l)=0.0
      vms(lf,in,l)=0.0
      vr(lf,in,l)=0.0
      vfat(lf,in,l)=0.0
      vurin(lf,in,l)=0.0
      q1(lf,in,l)=0.0
      q2(lf,in,l)=0.0
      q3(lf,in,l)=0.0
      rkd(lf,in,l)=0.0
      rbn(lf,in,l)=0.0
```

```
      rndo(lf,in,l)=0.0
      rlv(lf,in,l)=0.0
      rlu(lf,in,l)=0.0
      rms(lf,in,l)=0.0
c
  903 vmilk=1.1915e-03*(wt(k,in)*1000.)**0.78
      cbf(in,l)=cbf(in,l)+cblc*vmk(k,in,l)*sfd(ifa,in)*fracs/
     1(afmt*vmilk)
c
c contaminant transport via milk during weaning
      cbw(in,l)=cbw(in,l)+(1.-cblc)*vmk(k,in,l)*gtbl(in,l)*sfd(ifa,in)
     1*fracs/(afmt*vmilk)
c
c weighted addition of fetal metabolic wastes to female parent
      cbp(k,in,l)=cblc*vurin(lf,in,l)*sfd(ifa,in)*fracs/afmt
      if(j.eq.1)cbp(k,in,l)=0.0
      goto702
  902 if(gsfc.gt.1.0)goto702
      if(cblc.le.0.0)goto702
      icagm1=cagm(in,1)
      iiir1=aiiir(in,1)
      iiir2=0
      icagm2=0
      igst2=gst(in,2)
      iiir2=aiiir(in,2)
      icagm2=cagm(in,2)
      if(k.ne.lua(in))goto702
c fetus index counter
      lagf(in)=lagf(in)+1
      if(lagf(in).gt.1)goto702
c fetal age, days
      if(iiir1.lt.icagm1) aiif(in,1)=aiif(in,1)+1.0
      if(iiir2.gt.icagm1.and.iiir2.lt.icagm2)
     1aiif(in,2)=aiif(in,2)+1.0
      if(aiif(in,1).gt.gestp(in))aiif(in,1)=gestp(in)
      if(aiif(in,2).gt.gestp(in))aiif(in,2)=gestp(in)
      aiifg=aiif(in,1)
      if(aiif(in,2).ge.1.0)aiifg=aiif(in,2)
c ** estimated fetus live weight and organ/tissue weights from beginning
c ** of growth period
      wt(lf,in)=0.05256*wt1(iage,in)*(1.0-exp(-9.*(aiifg/gestp(in))))
c ** estimated metabolic biomass of fetus
      wtgrm=wt(lf,in)*1000.
      if(wtgrm.le.1.0)wtgrm=1.0
      amw(lf,in)=5.6234e-03*wtgrm**0.75
c
c** contaminant content of fetus organs estimated
c
c fetal total body water, liters
      tbwlf=9.077e-04*wtgrm**0.963
c fetal cardiac output, liters/day
      cardrf=1.0197*wtgrm**0.79
c
c** compartment volumes **
```

```
c
c fetal body weight limited to a minimum of 1 gram
      if(wtgrm.lt.1.0)wtgrm=1.0
c
c fetal plasma vol, liters
      volplf=4.29e-05*wtgrm**0.992
c fetal muscle vol, liters
      volmuf=4.63e-04*wtgrm**1.009
c fetal kidney vol, liters
      volkif=2.18e-05*wtgrm**0.843
c fetal liver vol, liters
      vollif=8.59e-05*wtgrm**0.885
c fetal gut vol, liters
      volguf=8.65e-05*wtgrm**0.909
c fetal gut lumen vol, liters
      vollum=5.78e-05*wtgrm**0.934
c fetal heart vol, liters
      volhrf=2.0e-06*wtgrm**1.043
c fetal lungs vol, liters
      volluf=3.16e-06*wtgrm**1.104
c fetal spleen vol, liters
      volspf=4.52e-06*wtgrm**0.901
c fetal marrow vol, liters
      volmaf=3.3e-05*wtgrm**0.936
c fetal fat vol, liters
      volfaf=1.1e-04*wtgrm
c fetal current residual volume
      volref=volguf+volhrf+volspf+volluf
c
c ** fetal plasma flow rates **
c
c fetal plasma flow, liters/day
      floplf=1.44*.601*wtgrm**0.778
c fetal muscle flow, liters/day
      flomuf=1.44*.0817*wtgrm**0.81
c fetal kidney flow, liters/day
      flokif=1.44*0.0845*wtgrm**0.802
c fetal liver flow, liters/day
      flolif=1.44*0.0944*wtgrm**0.792
c fetal gut flow, liters/day
      floguf=1.44*0.102*wtgrm**0.787
c fetal heart flow, liters/day
      flohrf=1.44*5.77e-03*wtgrm**0.965
c fetal lung flow, liters/day
      floluf=1.44*0.215*wtgrm**0.82
c fetal spleen flow, liters/day
      flospf=1.44*1.79e-03*wtgrm**1.028
c fetal marrow flow, liters/day
      flomaf=1.44*0.0164*wtgrm**0.804
c fetal fat flow
      flofaf=0.75*flomuf
c ** fetal flow to residual tissues
      floref=flohrf+flospf+0.1*floguf+0.1*floluf
c urine production, liters/day
```

```
      flourf=0.024*.0064*wtgrm**0.82
c  fetal arterial blood content
      viiin(k)=v(lf,in,l)
      dbo=0.0
c
c fetal arterial perfusion of organs with contaminant
c fetal bone(2) contaminant content updated for gains from arterial
c blood
      do720it=1,24
      dbn2=0.0
      if(bntbl2(in,l).le.0.0)goto433
      f0=2.0*flomaf
      f2=viiin(l)/volplf
      f3=vbn2(lf,in,l)/(volmaf*bntbl2(in,l))
      dbn2=f0*(f2-f3)/24.
c same for bone(1) compartment
c
  433 dbn1=0.0
      if(bntbl1(in,l).le.0.0)goto434
      f0=2.0*flomaf
      f2=viiin(l)/volplf
      f3=vbn1(lf,in,l)/(volmaf*bntbl1(in,l))
      dbn1=f0*(f2-f3)/24.
      dbo=f0*f2/24.
c
c same for muscle contaminant content
  434 dbms=0.0
      if(mstbl(in,l).le.0.0)goto435
      f0=flomuf
      f2=viiin(l)/volplf
      f3=vms(lf,in,l)/(volmuf*mstbl(in,l))
      dbms=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c same for kidney(2) compartment
  435 dbk2=0.0
      if(kdtbl2(in,l).le.0.0)goto436
      f0=flokif
      f2=viiin(l)/volplf
      f3=vkd2(lf,in,l)/(volkif*kdtbl2(in,l))
      dbk2=f0*(f2-f3)/24.
c
c adipose tissue content updated for gains from arterial blood
  436 dbft=0.0
      if(fttbl(in,l).le.0.0)goto437
      f0=flofaf
      f2=viiin(l)/volplf
      f3=vfat(lf,in,l)/(volfaf*fttbl(in,l))
      dbft=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c residual soft tissue content updated for gain from arterial blood
  437 dbrs=0.0
      if(rstbl(in,l).le.0.0)goto438
```

```
      f0=floref
      f2=viiin(l)/volplf
      f3=vr(lf,in,l)/(volref*rstbl(in,l))
      dbrs=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
c kidney(1) contaminant content updated for gains from arterial
c blood and losses to urine
  438 dbk1=0.0
      dbu=0.0
      if(kdtbl1(in,l).le.0.0)goto439
      f0=flokif
      f2=viiin(l)/volplf
      f3=vkd1(lf,in,l)/(volkif*kdtbl1(in,l))
      dbk1=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
      if(bltur(in,l).le.0.0)goto439
      f2=viiin(l)*bltur(in,l)
      dbu=f0*f2/(24.*floplf)
      if(dbu.le.0.0)dbu=0.0
      if(ainsm.le.0.0)dbu=0.0
c
c liver contaminant content updated for gains from arterial blood
c and gut absorption
  439 dblv=0.0
      if(lvtbl(in,l).le.0.0)goto718
      f0=flolif
      f2=viiin(l)/volplf
      f3=vlv(lf,in,l)/(vollif*lvtbl(in,l))
      dblv=f0*(f2-f3)/24.
      dbo=dbo+f0*f2/24.
c
  718 if(dbo.le.0.0)goto720
      flxbn2=dbn2*viiin(l)/dbo
      brto=0.99*vbn2(lf,in,l)
      if(flxbn2.lt.0.0.and.-flxbn2.gt.brto)flxbn2=-brto
      vbn2(lf,in,l)=vbn2(lf,in,l)+flxbn2
      viiind=viiin(l)-flxbn2
      if(viiind.le.0.0)viiind=0.0
      if(vbn2(lf,in,l).le.0.0)vbn2(lf,in,l)=0.0
      flxbn1=dbn1*viiin(l)/dbo
      brto=0.99*vbn1(lf,in,l)
      if(flxbn1.lt.0.0.and.-flxbn1.gt.brto)flxbn1=-brto
      if(bntbl2(in,l).le.0.0)viiind=viiin(l)
      vbn1(lf,in,l)=vbn1(lf,in,l)+flxbn1*viiind/viiin(l)
      if(flxbn2.lt.0.0)vbn1(lf,in,l)=vbn1(lf,in,l)-flxbn2
      if(flxbn2.lt.0.0)flxbn2=0.0
      if(vbn1(lf,in,l).le.0.0)vbn1(lf,in,l)=0.0
      flxms=dbms*viiin(l)/dbo
      brto=0.99*vms(lf,in,l)
      if(flxms.lt.0.0.and.-flxms.gt.brto)flxms=-brto
      vms(lf,in,l)=vms(lf,in,l)+flxms
      if(vms(lf,in,l).le.0.0)vms(lf,in,l)=0.0
      flxkd2=dbk2*viiin(l)/dbo
```

```
      brto=0.99*vkd2(lf,in,l)
      if(flxkd2.lt.0.0.and.-flxkd2.gt.brto)flxkd2=-brto
      vkd2(lf,in,l)=vkd2(lf,in,l)+flxkd2
      viiind=viiin(l)-flxkd2
      if(viiind.le.0.0)viiind=0.0
      if(vkd2(lf,in,l).le.0.0)vkd2(lf,in,l)=0.0
      flxkd1=dbk1*viiin(l)/dbo
      brto=0.99*vkd1(lf,in,l)
      if(flxkd1.lt.0.0.and.-flxkd1.gt.brto)flxkd1=-brto
      if(kdtbl2(in,l).le.0.0)viiind=viiin(l)
      vkd1(lf,in,l)=vkd1(lf,in,l)+flxkd1*viiind/viiin(l)
      if(flxkd2.lt.0.0)vkd1(lf,in,l)=vkd1(lf,in,l)-flxkd2
      if(flxkd2.lt.0.0)flxkd2=0.0
      if(vkd1(lf,in,l).le.0.0)vkd1(lf,in,l)=0.0
      flxft=dbft*viiin(l)/dbo
      brto=0.99*vfat(lf,in,l)
      if(flxft.lt.0.0.and.-flxft.gt.brto)flxft=-brto
      vfat(lf,in,l)=vfat(lf,in,l)+flxft
      if(vfat(lf,in,l).le.0.0)vfat(lf,in,l)=0.0
      flxrs=dbrs*viiin(l)/dbo
      brto=0.99*vr(lf,in,l)
      if(flxrs.lt.0.0.and.-flxrs.gt.brto)flxrs=-brto
      vr(lf,in,l)=vr(lf,in,l)+flxrs
      if(vr(lf,in,l).le.0.0)vr(lf,in,l)=0.0
      vurin(lf,in,l)=vurin(lf,in,l)+dbu
      flxlv=dblv*viiin(l)/dbo
      flxlvp=flxlv
      brto=0.99*vlv(lf,in,l)
      if(flxlv.lt.0.0.and.-flxlv.gt.brto)flxlv=-brto
      vlv(lf,in,l)=vlv(lf,in,l)+flxlv
      if(vlv(lf,in,l).le.0.0)vlv(lf,in,l)=0.0
      v(lf,in,l)=v(lf,in,l)+cbf(in,l)/24.-flxbn1-flxbn2-flxkd1-flxkd2
     1-flxft-flxrs-flxlv-flxms-dbu
      if(v(lf,in,l).le.0.0)v(lf,in,l)=0.0
      viiin(l)=v(lf,in,l)
  720 continue
c  contaminant corrected for contaminant decay or persistence
      v(lf,in,l)=v(lf,in,l)-disint(l)*v(lf,in,l)
      vbn1(lf,in,l)=vbn1(lf,in,l)-disint(l)*vbn1(lf,in,l)
      vbn2(lf,in,l)=vbn2(lf,in,l)-disint(l)*vbn2(lf,in,l)
      vlv(lf,in,l)=vlv(lf,in,l)-vlv(lf,in,l)*disint(l)
      vms(lf,in,l)=vms(lf,in,l)-disint(l)*vms(lf,in,l)
      vkd1(lf,in,l)=vkd1(lf,in,l)-disint(l)*vkd1(lf,in,l)
      vkd2(lf,in,l)=vkd2(lf,in,l)-disint(l)*vkd2(lf,in,l)
      vr(lf,in,l)=vr(lf,in,l)-disint(l)*vr(lf,in,l)
      vfat(lf,in,l)=vfat(lf,in,l)-disint(l)*vfat(lf,in,l)
c
c**** weighted contaminant dose assessment for human fetus****
c
c** excludes nutrients
      ian=atno(l)
      if(l.gt.1.and.l.lt.7)goto723
      if(wt(lf,in).le.0.00105)wt(lf,in)=0.00105
      wtgrm=wt(lf,in)*1000.
```

```
c
      wtbn(lf,in)=5.5e-05*wtgrm
      wtkd(lf,in)=2.12e-05*wtgrm**0.85
      wtlv(lf,in)=8.2e-05*wtgrm**0.87
      wtms(lf,in)=4.5e-04*wtgrm
      wtlu(lf,in)=1.24e-05*wtgrm**0.99
c
      if(ian.ge.101.)goto723
c fetal cummulative kidney dose, rems/gfwt
      rkd(lf,in,l)=rkd(lf,in,l)+5.12e-05*evs(in,l)*(vkd1(lf,in,l)
     1+vkd2(lf,in,l))/(wtkd(lf,in)*1000.)
c fetal cumulative bone dose, rems/gfwt
      rbn(lf,in,l)=rbn(lf,in,l)+5.12e-05*evb(in,l)*(vbn1(lf,in,l)
     1+vbn2(lf,in,l))/(wtbn(lf,in)*1000)
c fetal contaminant burden in bone(1,2)
      vndof=(vbn1(lf,in,l)+vbn2(lf,in,l))
c fetal weight of endosteal bone tissue, gfwt
      endwtf=0.025*wtbn(lf,in)*1000.
c fetal cummulative endosteal bone dose, rems/gfwt
      rndo(lf,in,l)=rndo(lf,in,l)+5.12e-05*eve(in,l)*vndof/endwtf
c fetal cummulative liver dose, rems/gfwt
      rlv(lf,in,l)=rlv(lf,in,l)+5.12e-05*evs(in,l)*vlv(lf,in,l)/
     1(wtlv(lf,in)*1000.)
c fetal cummulative muscle dose, rems/gfwt
      rms(lf,in,l)=rms(lf,in,l)+5.12e-05*evs(in,l)*vms(lf,in,l)/
     1(wtms(lf,in)*1000.)
c
c*******end fetal burden and dose calculations
c
c contaminant dose assessment for live humans
  702 continue
c excludes nutrients
      if(l.gt.1.and.l.lt.7)goto723
      ian=atno(l)
      if(ian.ge.101)goto723
      if(k.ne.1)goto706
      if(aiiir(in,1).le.aiii)goto706
      if(gstn(in).lt.2.0) goto570
      if(aiiir(in,2).gt.aiii)goto570
c
c cummulative kidney dose, rems/gfwt
  706 rkd(k,in,l)=rkd(k,in,l)+5.12e-05*evs(in,l)*(vkd1(k,in,l)+
     1vkd2(k,in,l))/wtkd(k,in)
c cummulative bone dose, rems/gfwt
      rbn(k,in,l)=rbn(k,in,l)+5.12e-05*evb(in,l)*(vbn1(k,in,l)+
     1vbn2(k,in,l))/wtbn(k,in)
c contaminant burden in bone(1,2)
      vndoh=(vbn1(k,in,l)+vbn2(k,in,l))
c weight of endosteal bone tissue, gfwt
      endwt=wtbn(k,in)*125./5000.
c cummulative endosteal bone dose, rems/gfwt
      rndo(k,in,l)=rndo(k,in,l)+5.12e-05*eve(in,l)*vndoh/endwt
c cummulative liver dose, rems/gfwt
      rlv(k,in,l)=rlv(k,in,l)+5.12e-05*evs(in,l)*vlv(k,in,l)/wtlv(k,in)
```

```
c cummulative lung dose, rems/gfwt
      rlu(k,in,l)=rlu(k,in,l)+5.12e-05*evs(in,l)*(q1(k,in,l)+q2(k,in,l)
     1+q3(k,in,l))/wtlu(k,in)
c cummulative muscle dose, rems/gfwt
      rms(k,in,l)=rms(k,in,l)+5.12e-05*evs(in,l)*vms(k,in,l)/wtms(k,in)
c
c kidney(1,2), bone(1,2), and lung contaminant pools (burdens)
  723 vkd(k,in,l)=vkd1(k,in,l)+vkd2(k,in,l)
      vbn(k,in,l)=vbn1(k,in,l)+vbn2(k,in,l)
      vlu(k,in,l)=q1(k,in,l)+q2(k,in,l)+q3(k,in,l)
c cummulative urine, fecal, and milk contaminant losses
      vlos(k,j,l)=vfec(k,j,l)+vurin(k,in,l)
      if(j.eq.2)vlos(k,j,l)=vlos(k,j,l)+vmk(k,in,l)
c total body contaminant content in bone, kidneys,muscle, and liver
      vrt1(k,in,l)=vbn(k,in,l)+vkd(k,in,l)+vms(k,in,l)+vlv(k,in,l)
c total body contaminant including above plus lung and blood contents
      vrt(k,in,l)=vrt1(k,in,l)+vlu(k,in,l)+v(k,in,l)
c
c excludes males
c     if(j.eq.1)goto570
c excludes immature and older females from milk production
      agemi=agem(k,in)
      fracs=1.0
      if(k.ne.1)goto724
      off2t=0.0
      off2i=0.0
      do725ig1=1,3
      off2t=off2t+offsp(in,ig1,j)
      if(affsp(in,ig1).ge..15*alsp(in).and.affsp(in,ig1).le..7*alsp(in))
     1off2i=off2i+offsp(in,ig1,j)
  725 continue
      fracs=0.0
      if(off2t.gt.0.0) fracs=off2i/off2t
      if(fracs.gt.0.0.and.agemi.lt.0.16*alsp(in)) agemi=0.16*alsp(in)
  724 if(agemi.lt.0.15*alsp(in).or.agemi.gt.0.7*alsp(in))goto570
      if(afmt.le.0.0)goto570
c mean daily activity per liter in milk, producer age groups

      if(j.eq.2) w(18,in,l)=w(18,in,l)+vmk(k,in,l)*sfd(ifa,in)
     1*fracs/(afmt*flomlk)
  570 continue
c ** end contaminant loop ******************************************
      aiin=in
      if(ydos(in).gt.0.0)call effects(in,lx,k,j,rdidx)
c
c excludes all but last julian day number, and oldest age group in loop
      if(k.lt.iage)goto 790
c initializes population parameters daily
      if(k.gt.iage)goto831
      zoo(in)=0.0
      pop(j,in)=0.0
      if(aiii.gt.1.0)goto742
c initializes daily mortalities for all age groups combined
      summrt(in,j)=0.0
```

```
  742 ikn(j,in,1)=0
      ikn(j,in,2)=0
      ikn(j,in,3)=0
      ikn(j,in,4)=0
      ikn(j,in,5)=0
      ikn(j,in,6)=0
      ikn(j,in,7)=0
      ikn(j,in,8)=0
c ** age loop ********************************************************
      iage=agea(in)
      do 770 kg=1,iage
c estimates population losses on a daily basis
c does not allow mortality losses to exceed predation losses
      if(kg.gt.1)goto772
      tqx1=tqx1+tqp(kg,in)+tqs(kg,in)
      tqx2=tqx2+tqp(kg,in)+tqs(kg,in)
      tqx3=tqx3+tqp(kg,in)+tqs(kg,in)
c
      summrt(in,j)=summrt(in,j)+tqx1*offsp(in,1,j)+tqx2*offsp(in,2,j)+
     1tqx3*offsp(in,3,j)
c
      offsp(in,1,j)=offsp(in,1,j)-tqx1*offsp(in,1,j)
      offsp(in,2,j)=offsp(in,2,j)-tqx2*offsp(in,2,j)
      offsp(in,3,j)=offsp(in,3,j)-tqx3*offsp(in,3,j)
c
      tpop(kg,j,in)=offsp(in,1,j)+offsp(in,2,j)+offsp(in,3,j)
      pop(j,in)=pop(j,in)+tpop(kg,j,in)
      zoo(in)=zoo(in)+tpop(kg,j,in)
      goto773
  772 tqx(kg,j,in)=tqx(kg,j,in)+tqp(kg,in)+tqs(kg,in)
c
      summrt(in,j)=summrt(in,j)+tqx(kg,j,in)*tpop(kg,j,in)
      tpop(kg,j,in)=tpop(kg,j,in)-tqx(kg,j,in)*tpop(kg,j,in)
c
      pop(j,in)=pop(j,in)+tpop(kg,j,in)
      zoo(in)=zoo(in)+tpop(kg,j,in)
c
c ** births age loop *************************************************
  773 if(kg.ne.1)goto770
      iagem=aiiir(in,1)
      icagm=cagm(in,1)
      if(iagem.eq.icagm)irp=1
      if(iagem.eq.icagm)goto805
c
      if(gstn(in).lt.2.0)goto770
      icagm=cagm(in,2)
      iagem=aiiir(in,2)
      if(iagem.eq.icagm)irp=2
      if(iagem.eq.icagm)goto805
      goto770
c
c initializes daily births for each reproductive cycle
  805 births(j,in)=0.0
c
```

```
      iaged=iage*365.
      do621kk=1,iaged
      akd=kk
      if (akd.lt..15*alsp(in).or.akd.gt..70*alsp(in)) go to 621
c sets initial male-female birth rates for youngest females
      brate=brrt(1,in,j)*sfd(1,in)*popp(2,nto)*frpr(irp,in)
      knp=1
c ** birth rate loop ***************************************************
      do 623 kn=1,7
c natality rate changed according female age groupings specified in (BRTY)
      if(akd.le.brty(kn,in))goto623
      knp=kn+1
      brate=brrt(knp,in,j)*sfd(knp,in)*popp(2,nto)*frpr(irp,in)
  623 continue
c * end birth rate loop ************************************************
c
      if(ikn(j,in,knp).gt.0)goto624
c new-borns of males-females estimated on a daily basis
      births(j,in)=births(j,in)+brate
      brts(in,j,irp)=births(j,in)
      ikn(j,in,knp)=ikn(j,in,knp)+1
  624 continue
  621 continue
c
      affsp(in,irp+1)=1.0
      if(jage(in).ne.0)goto625
      if(irp.eq.1) offsp(in,1,j)=births(j,in)
      if(irp.eq.2) offsp(in,1,j)=offsp(in,1,j)+births(j,in)
      aiif(in,1)=0.0
      aiif(in,2)=0.0
      goto806
  625 offsp(in,irp,j)=offsp(in,irp,j)+offsp(in,irp+1,j)
      offsp(in,irp+1,j)=births(j,in)
      aiif(in,1)=0.0
      aiif(in,2)=0.0
c newborn ratio to other 1 year-olds estimated
c
  806 tppcs(in)=births(j,in)
  770 continue
      c12f=1.0
      offsm=offsp(in,1,j)+offsp(in,2,j)+offsp(in,3,j)
      if(offsm.le.0.0)goto774
      c12f=rat12(in)
      off1=1.0
      off2=1.0
      off3=1.0
      off1=offsp(in,1,j)/offsm
      off2=offsp(in,2,j)/offsm
      off3=offsp(in,3,j)/offsm
      if(jage(in).ne.0)offsp(in,1,j)=tpop(1,j,in)*(1.0-c12f)*off1
      if(jage(in).ne.0)offsp(in,2,j)=tpop(1,j,in)*(1.0-c12f)*off2
      if(jage(in).ne.0)offsp(in,3,j)=tpop(1,j,in)*(1.0-c12f)*off3
c
  774 if(aiii.lt.365)goto777
```

```
      iage=agea(in)
      do771 kg=1,iage
      jj=kg-1
c
c the following statements move animal parameters into next age group
c at the end of each Julian year
c
      if(kg.eq.iage)goto771
      if(iage-kg.ne.1)goto776
c
c updates population structure if mortality parameters are specified
c
  776 if(jage(in).ne.0) tpop(iage-jj,j,in)=tpop(iage-kg,j,in)*c12f
c updates other population parameters
c
c total weight,carbohydrate, protein, and fat contents moved into succeeding
c age group
      wt(iage-jj,in)=wt(iage-kg,in)
      ch20b(iage-jj,j)=ch20b(iage-kg,j)
      protb(iage-jj,j)=protb(iage-kg,j)
      fatb(iage-jj,in)=fatb(iage-kg,in)
c
c ** contaminant loop ************************************************
      do 760 l=1,inz
c updates contaminant compartments
c blood
      v(iage-jj,in,l)=v(iage-kg,in,l)
c residual soft tissue
      vr(iage-jj,in,l)=vr(iage-kg,in,l)
c bone(1 & 2)
      vbn1(iage-jj,in,l)=vbn1(iage-kg,in,l)
      vbn2(iage-jj,in,l)=vbn2(iage-kg,in,l)
c liver
      vlv(iage-jj,in,l)=vlv(iage-kg,in,l)
c kidneys(1 & 2)
      vkd1(iage-jj,in,l)=vkd1(iage-kg,in,l)
      vkd2(iage-jj,in,l)=vkd2(iage-kg,in,l)
c muscle
      vms(iage-jj,in,l)=vms(iage-kg,in,l)
c fat
      vfat(iage-jj,in,l)=vfat(iage-kg,in,l)
c lung compartment(1,2, & 3)
      q1(iage-jj,in,l)=q1(iage-kg,in,l)
      q2(iage-jj,in,l)=q2(iage-kg,in,l)
      q3(iage-jj,in,l)=q3(iage-kg,in,l)
c kidney(1,2), bone(1,2), endosteal tissue, liver, lung, and muscle
c cummulative doses moved into succeeding age group
      rkd(iage-jj,in,l)=rkd(iage-kg,in,l)
      rbn(iage-jj,in,l)=rbn(iage-kg,in,l)
      rndo(iage-jj,in,l)=rndo(iage-kg,in,l)
      rlv(iage-jj,in,l)=rlv(iage-kg,in,l)
      rlu(iage-jj,in,l)=rlu(iage-kg,in,l)
      rms(iage-jj,in,l)=rms(iage-kg,in,l)
c cummulative contaminant uptakes in consumed leafy vegetables, seeds
```

```
c (fruits),grains, meats and milk moved into succeeding age group
      zlvi(iage-jj,j,l)=zlvi(iage-kg,j,l)
      zsvi(iage-jj,j,l)=zsvi(iage-kg,j,l)
      zgrni(iage-jj,j,l)=zgrni(iage-kg,j,l)
      zmeati(iage-jj,j,l)=zmeati(iage-kg,j,l)
      zmilki(iage-jj,j,l)=zmilki(iage-kg,j,l)
c cummulative contaminant uptake from drinking water moved into succeeding
c age group
      wburi(iage-jj,j,l)=wburi(iage-kg,j,l)
c N-P,T-B,and P cumulative contaminant uptake moved into succeeding age
c group; same for N-P and T-B cummulative contaminant uptake(GBURI)
      pburi(iage-jj,j,l)=pburi(iage-kg,j,l)
      gburi(iage-jj,j,l)=gburi(iage-kg,j,l)
  760 continue
c ** end contaminant loop ********************************************
  771 continue
c ** end age loop ****************************************************
  777 aoffsp=affsp(in,2)
      if(aiiir(in,1).ge.cagm(in,1)+90.) goto792
      if(gstn(in).le.1.0)goto790
      aoffsp=affsp(in,3)
      if(aiiir(in,2).ge.cagm(in,2)+90.)goto792
      goto790
  792 prwt=wt(1,in)
      iage=agea(in)
      wt(1,in)=wt1(iage,in)*(1.0-exp(-9.0*aoffsp/alsp(in)))
      wtrat=wt(1,in)/prwt
c population number, body weight, protein weight, and fat weight of new-born
c males-females estimated
      protb(1,j)=prot1(1,j)*wtrat
      fatb(1,in)=.0125*wtrat*exp(-(.693/7.2))*wt1(1,in)**2
c new-born male total body water
      tbwk=(.6111*wt1(1,in)+0.25)*wtrat
c new-born female total body water
      if (j.eq.2) tbwk=(.511*wt1(1,in)+1.244)*wtrat
c new-born male-female carbohydrate weight
      ch20b(1,j)=0.968*wtrat*wt1(1,in)-fatb(1,in)-
     1protb(1,j)-tbwk
      do791l=1,inz
c blood
      v(1,in,l)=v(1,in,l)*wtrat
c residual soft tissue
      vr(1,in,l)=vr(1,in,l)*wtrat
c bone(1 & 2)
      vbn1(1,in,l)=vbn1(1,in,l)*wtrat
      vbn2(1,in,l)=vbn2(1,in,l)*wtrat
c liver
      vlv(1,in,l)=vlv(1,in,l)*wtrat
c kidneys(1 & 2)
      vkd1(1,in,l)=vkd1(1,in,l)*wtrat
      vkd2(1,in,l)=vkd2(1,in,l)*wtrat
c muscle
      vms(1,in,l)=vms(1,in,l)*wtrat
c lung compartment(1,2, & 3)
```

```
      q1(1,in,l)=q1(1,in,l)*wtrat
      q2(1,in,l)=q2(1,in,l)*wtrat
      q3(1,in,l)=q3(1,in,l)*wtrat
  791 continue
c
c ** contaminant loop ************************************************
  790 iage=agea(in)
      if(ii.lt.iage)goto831
      do 630 l=1,inz
c mean age weighted population characteristics initialized
      w(1,in,l)=0.0
      w(2,in,l)=0.0
      w(3,in,l)=0.0
      w(4,in,l)=0.0
      w(5,in,l)=0.0
      w(6,in,l)=0.0
      w(7,in,l)=0.0
      w(8,in,l)=0.0
      w(9,in,l)=0.0
      w(10,in,l)=0.0
      w(11,in,l)=0.0
      w(12,in,l)=0.0
      w(13,in,l)=0.0
      w(14,in,l)=0.0
      w(15,in,l)=0.0
      w(16,in,l)=0.0
      w(17,in,l)=0.0
      w(19,in,l)=0.0
  630 continue
c ** end contaminant loop ********************************************
c
c ** age loop ************************************************************
      wms(in,2)=0.0
      wtrsa=0.0
      wtbla=0.0
      wtfff=0.0
      wtkdd=0.0
      wtmuu=0.0
      wtlvv=0.0
      wtbnn=0.0
      wtluu=0.0
      wtbod=0.0
      xinf=0.0
c ** age loop ************************************************************
      iage=agea(in)
      do 811 kg=1,iage
c estimates fraction of milk producing females in female population
      do 813 l=1,inz
      if(vmk(kg,in,l).gt.0.0)xinf=xinf+a(kg,in,2)
  813 continue
c
c mean values of mass parameters estimated
c
c mean human weight, kgfwt
```

```
      xins=a(kg,in,j)*pop(j,in)/zoo(in)
      wms(in,2)=wms(in,2)+wt(kg,in)*xins
      wtrsa=wtrsa+wtrs(kg,in)*xins
      wtbla=wtbla+wtbl(kg,in)*xins
      wtfff=wtfff+fatb(kg,in)*xins
      wtkdd=wtkdd+wtkd(kg,in)*xins
      wtmuu=wtmuu+wtms(kg,in)*xins
      wtlvv=wtlvv+wtlv(kg,in)*xins
      wtbnn=wtbnn+wtbn(kg,in)*xins
      wtluu=wtluu+wtlu(kg,in)*xins
      wtbod=wtbod+wt(kg,in)*xins
  811 continue
c ** age loop ***********************************************************
      iage=agea(in)
      do 650 kg=1,iage
c ** contaminant loop ****************************************************
      do 640 l=1,inz
c mean weighted concentrations of various organs and contaminant sources
c and same for each age group, per kgfwt organ or body
      xins=a(kg,in,j)*pop(j,in)/zoo(in)
      xinf2=0.0
      if(vmk(kg,in,l).gt.0.0)xinf2=a(kg,in,2)
c milk clearance, per kgfwt body
      if(xinf.gt.0.) w(1,in,l)=w(1,in,l)+vmk(kg,in,l)*xinf2/(xinf*wtbod)
      cnmlk(kg,in,l)=vmk(kg,in,l)/wt(kg,in)
c urine clearance, per kgfwt body
      w(2,in,l)=w(2,in,l)+vurin(kg,in,l)*xins/wtbod
      cnurnc(kg,in,l)=vurin(kg,in,l)/wt(kg,in)
c contaminant ingestion from lungs,per kgfwt body
      w(3,in,l)=w(3,in,l)+vlug(kg,in,l)*xins/wtbod
      cnlug(kg,in,l)=vlug(kg,in,l)/wt(kg,in)
c contaminant ingestion from soil ingestion, per kgfwt body (set to 0.0)
      w(4,in,l)=w(4,in,l)+vsol(kg,in,l)*xins/wtbod
      cnsol(kg,in,l)=vsol(kg,in,l)/wt(kg,in)
c contaminant ingestion from plant foods, per kgfwt body
      w(5,in,l)=w(5,in,l)+vpln(kg,in,l)*xins/wtbod
      cnpln(kg,in,l)=vpln(kg,in,l)/wt(kg,in)
c internal body concentration
      bbr=vkd1(kg,in,l)+vkd2(kg,in,l)+vbn1(kg,in,l)+vbn2(kg,in,l)+
     1vr(kg,in,l)+vms(kg,in,l)+vlv(kg,in,l)+v(kg,in,l)+vfat(kg,in,l)
      w(6,in,l)=w(6,in,l)+bbr*xins/wtbod
      cnbod(kg,in,l)=bbr/wt(kg,in)
c skin concentration, per kgfwt body (set to 0.0)
      w(7,in,l)=w(7,in,l)+vpelt(kg,in,l)*xins/wtbod
      cnpel(kg,in,l)=vpelt(kg,in,l)/wt(kg,in)
c total ingested contaminant concentration, per kgfwt body
      w(8,in,l)=w(8,in,l)+vgut(kg,in,l)*xins/wtbod
      cngut(k,in,l)=vgut(kg,in,l)/wt(kg,in)
c kidney concnentration
      w(9,in,l)=w(9,in,l)+(vkd1(kg,in,l)+vkd2(kg,in,l))*xins/wtkdd
      cnkid(kg,in,l)=(vkd1(kg,in,l)+vkd2(kg,in,l))/wtkd(kg,in)
c mean bone contaminant concentration
      w(10,in,l)=w(10,in,l)+(vbn1(kg,in,l)+vbn2(kg,in,l))*xins/wtbnn
      cnbon(kg,in,l)=(vbn1(kg,in,l)+vbn2(kg,in,l))/wtbn(kg,in)
```

```
c mean residual soft tissue contaminant concentration
      w(11,in,l)=w(11,in,l)+vr(kg,in,l)*xins/wtrsa
      cnres(kg,in,l)=vr(kg,in,l)/wtrs(kg,in)
c mean muscle contaminant concentration
      w(13,in,l)=w(13,in,l)+vms(kg,in,l)*xins/wtmuu
      cnmus(kg,in,l)=vms(kg,in,l)/wtms(kg,in)
c mean liver contaminant concentration
      w(14,in,l)=w(14,in,l)+vlv(kg,in,l)*xins/wtlvv
      cnliv(kg,in,l)=vlv(kg,in,l)/wtlv(kg,in)
c mean blood contaminant concentration
      w(15,in,l)=w(15,in,l)+v(kg,in,l)*xins/wtbla
      cnblo(kg,in,l)=v(kg,in,l)/wtbl(kg,in)
c lung contaminant concentration from particulates
      slung(l)=q1(kg,in,l)+q2(kg,in,l)+q3(kg,in,l)
c mean lung contaminant concentration
      w(16,in,l)=w(16,in,l)+slung(l)*xins/wtluu
      cnlun(kg,in,l)=slung(l)/wtlu(kg,in)
c mean fat contaminant concentration
      w(17,in,l)=w(17,in,l)+vfat(kg,in,l)*xins/wtfff
      cnfat(kg,in,l)=vfat(kg,in,l)/fatb(kg,in)
c mean urine concentration, per liter
      flourn=0.024*.0064*(wt(kg,in)*1000.)**0.82
      w(19,in,l)=w(19,in,l)+(vurin(kg,in,l)/flourn)*xins
      cnurn(kg,in,l)=vurin(kg,in,l)/flourn
c mean feces concentration, per liter
      vollum=5.78e-05*(wt(kg,in)*1000.)**0.934
      w(20,in,l)=w(20,in,l)+(1.0-gtbl(in,l))*vgut(kg,in,l)/vollum
      cnfec(kg,in,l)=(1.0-gtbl(in,l))*vgut(kg,in,l)/vollum
  640 continue
c ** end contaminant loop ******************************************
  650 continue
c ** end age loop ****************************************************
c population mean muscle mass, kgfwt
      if(wms(in,2).le.0.0)goto831
c total population muscle mass
      wms(in,1)=0.45*wms(in,2)*zoo(in)
c
c ** establishes contaminant concentrations in newborns
  831 aiin=in
c ** establishes contaminant concentrations in newborns ****
c ** excludes other age groups
      if(ii.lt.iage+1)goto860
      iagem=aiiir(in,1)
      icagm1=cagm(in,1)
      if(iagem.ne.icagm1)goto851
      goto852
  851 if(gstn(in).lt.2.0)goto860
      icagm2=cagm(in,2)
      iagem=aiiir(in,2)
      if(iagem.ne.icagm2)goto860
c
c ** contaminant loop ***********************************************
c
  852 tppps(in)=tpop(1,in-nan,in)
```

```
      tpppt(in)=tppps(in)+tppcs(in)
      clf(in)=tppcs(in)/tpppt(in)
      ca1(in)=tppps(in)/tpppt(in)
      lf=iage+1
c newborn and fetal contaminant parameters initialized
      wt(1,in)=clf(in)*wt(iage+1,in)+ca1(in)*wt(1,in)
c ** contaminant loop ********************************************
      do 690 l=1,inz
      v(1,in,l)=clf(in)*v(lf,in,l)+ca1(in)*v(1,in,l)
      vbn1(1,in,l)=clf(in)*vbn1(lf,in,l)+ca1(in)*vbn1(1,in,l)
      vbn2(1,in,l)=clf(in)*vbn2(lf,in,l)+ca1(in)*vbn2(1,in,l)
      vlv(1,in,l)=clf(in)*vlv(lf,in,l)+ca1(in)*vlv(1,in,l)
      vkd1(1,in,l)=clf(in)*vkd1(lf,in,l)+ca1(in)*vkd1(1,in,l)
      vkd2(1,in,l)=clf(in)*vkd2(lf,in,l)+ca1(in)*vkd2(1,in,l)
      vms(1,in,l)=clf(in)*vms(lf,in,l)+ca1(in)*vms(1,in,l)
      vr(1,in,l)=clf(in)*vr(lf,in,l)+ca1(in)*vr(1,in,l)
      vfat(1,in,l)=clf(in)*vfat(lf,in,l)+ca1(in)*vfat(1,in,l)
c
      q1(1,in,l)=clf(in)*q1(lf,in,l)+ca1(in)*q1(1,in,l)
      q2(1,in,l)=clf(in)*q2(lf,in,l)+ca1(in)*q2(1,in,l)
      q3(1,in,l)=clf(in)*q2(lf,in,l)+ca1(in)*q3(1,in,l)
c
      rbn(1,in,l)=clf(in)*rbn(lf,in,l)+ca1(in)*rbn(1,in,l)
      rndo(1,in,l)=clf(in)*rndo(lf,in,l)+ca1(in)*rndo(1,in,l)
      rlv(1,in,l)=clf(in)*rlv(lf,in,l)+ca1(in)*rlv(1,in,l)
      rlu(1,in,l)=clf(in)*rlu(lf,in,l)+ca1(in)*rlu(1,in,l)
      rms(1,in,l)=clf(in)*rms(lf,in,l)+ca1(in)*rms(1,in,l)
      rkd(1,in,l)=clf(in)*rkd(lf,in,l)+ca1(in)*rkd(1,in,l)
c
  690 continue
c ** end contaminant loop ****************************************
  860 ain=in
      return
      end
```

14.13. PLTGRO (bmm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine pltgro (lpn,lx,rdidx,bpx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c pltgro simulates plant growth and mortality
c
      include 'main.inc'
c
      common/ph/ adsuc(n), ahrv(i,n), arig(i,n), pgr(n), rtrans(i,n),
     1 syav(i,n), xfac(i,n)
      dimension hfa(i,n),wutpa(i,n),sctpa(i,n),astpa(i,n),rxppa(i,n),
     1rfpa(i,n),psepa(i,n),bcpa(i,n),strana(i,n),rtrana(i,n),topta(i,n),
     2tofla(i,n),fleafa(i,n),ulefa(i,n),respfa(i,n),rspf2a(i,n),
     3tofha(i,n)
c
c excludes all but first access for parameter input and initialization
      if (rdidx) 100,101,100
  101 if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-121212.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','green')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE PLTGRO                    ')
      call wnoust('******************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
```

```
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 121212.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c ** region loop ****************************************************
      do 90 lz=1,ipx
c succession, niche competiton, and advanced succession gates entered
 5003 print *,'ENTER plant succession gate(0.,1.) for region ',lz
      if(nu.eq.0)read(*,*,err=5003)rscde(lz)
      if(nu.eq.1)read(7,*)rscde(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)rscde(lz)
      if(iu.eq.1)print *,rscde(lz)
 5004 print *,'ENTER plant niche-competition gate(0.,1.) for '
      print *,'region ',lz
      if(nu.eq.0)read(*,*,err=5004)aiche(lz)
      if(nu.eq.1)read(7,*)aiche(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)aiche(lz)
      if(iu.eq.1)print *,aiche(lz)
 5005 print *,'ENTER plant advanced succession gate(0.,1.) for'
      print *,'region ',lz,'(RSCDE must equal 1.0)'
      if(nu.eq.0)read(*,*,err=5005) adsuc(lz)
      if(nu.eq.1)read(7,*) adsuc(lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*) adsuc(lz)
```

```
      if(iu.eq.1)print *, adsuc(lz)
c
c plant simulation number, irrigation row index multiplier, number of
c harvest events, plant output selector, irrigation index, niche competition
c selector,and the simulation area (acres) for each plant type are entered
c
      print *,'*******************************************************'
      print *,'BIOTRAN.2 currently identifies 22 plant types: plants'
      print *,'1 to 10 are warm season plants; 11-22 are cool season'
      print *,'plants. 1=corn or sorghum,2=ann. veg. and grains,'
      print *,'3=pasture grass,4=fruit trees,5=pinyon-juniper,'
      print *,'6=dec. shrubs,7=ann. grasses and herbs,8=per. grasses'
      print *,'9=evergreen shrubs,10=dec. trees'
      print *,'11=alfalfa,12=ann. veg. and grains,13=pasture grass,'
      print *,'14=douglas fir,15=ponderosa pine,16=dec. shrubs,'
      print *,'17=spruce-fir,18=per. grasses,19=ann.grass and herbs'
      print *,'20=evergreen shrubs,21=dec. trees,22=cheatgrass'
      print *,'*******************************************************'
      print *,'USER may duplicate plant type in run by adding 22 '
      print *,'to duplicates: ex.. 1,23,45..for successive choices of'
      print *,'plant #1 (corn and sorghum) for a given run'
      print *,'USER may create own plant type by entering a negative'
      print *,'number such as -1,-2.....-22 for a given plant number'
      print *,'USER must be prepared to enter plant growth parameters'
      print *,'for each plant created. Parameters required are'
      print *,'described in later in this subroutine '
      print *,'*******************************************************'
 5006 print *,'ENTER plant id for ',ipl,'plants',' in region',lz
      if(nu.eq.0)read(*,*,err=5006) (pltid(k,lz),k=1,ipl)
      if(nu.eq.1)read(7,*) (pltid(k,lz),k=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*) (pltid(k,lz),k=1,ipl)
      if(iu.eq.1)print *, (pltid(k,lz),k=1,ipl)
      if (pltid(1,lz)) 5007,20,5007
 5007 print *,'ENTER number of water applications for ',ipl,'plants'
      print *,' in region ',lz
      if(nu.eq.0)read(*,*,err=5007) (arig(k,lz),k=1,ipl)
      if(nu.eq.1)read(7,*) (arig(k,lz),k=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*) (arig(k,lz),k=1,ipl)
      if(iu.eq.1)print *, (arig(k,lz),k=1,ipl)
 5008 print *,'ENTER number of harvest day for ',ipl,'plants'
      print *,'in region ',lz,' A minimum of 2.0 is required'
      print *,'even if only 1 harvest is specified'
      if(nu.eq.0)read(*,*,err=5008) (ahrv(k,lz),k=1,ipl)
      if(nu.eq.1)read(7,*) (ahrv(k,lz),k=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*) (ahrv(k,lz),k=1,ipl)
```

```
      if(iu.eq.1)print *, (ahrv(k,lz),k=1,ipl)
 5010 print *,'ENTER plant code(1.,2.,3.) for ',ipl,'plants in'
      print *,'region ',lz
      print *,' 1=forest structure not enabled, 2=forest structure'
      print *,' enabled yearly, 3=forest structure enabled daily'
      if(nu.eq.0)read(*,*,err=5010) (rgc(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (rgc(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*) (rgc(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (rgc(l,lz),l=1,ipl)
 5011 print *,'ENTER harvest-irrigation gate(0,1,2) for ',ipl,'plants'
      print *,'in region ',lz,' 0=none,1=yes,2=yes with succession'
      if(nu.eq.0)read(*,*,err=5011) (rirrc(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (rirrc(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*) (rirrc(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (rirrc(l,lz),l=1,ipl)
 5012 print *,'ENTER niche competition matches(0.0 for none)'
      print *,'for ',ipl,'plants in region ',lz,' Matched '
      print *,'plants must have same value...1.0,1.0, 2.0,2.0..etc.'
      if(nu.eq.0)read(*,*,err=5012) (rss(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (rss(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5012) ans
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*) (rss(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (rss(l,lz),l=1,ipl)
 5013 print *,'ENTER coverage area, acres, for ',ipl,'plants in'
      print *,'region ',lz,' A value of 0.0 defaults to model value'
      print *,'of 92162 acres (144 square miles)'
      if(nu.eq.0)read(*,*,err=5013) (acres(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (acres(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5013) ans
      if(ans.gt.0.and.nu.eq.0)goto5013
      write(6,*) (acres(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (acres(l,lz),l=1,ipl)
c
c irrigation input must be carefully constructed. if no irrigation of
c a crop or plant is wanted then set rirrc=0.0 which causes cuta,
c awata, and dira not to be read. refer to the irrigation module in
c the watflx subroutine for irrigation logic flow. multiple harvesting
c and perennials are specified by special input in column 1 for cuta.
c the first column of dira is reserved for planting date. succeeding
c columns contain ascending irrigation dates. in the same relative
c columns awata has the irrigation amounts for each date given by
c dira. for annuals the first column of cuta is reserved for the
c last harvest date. the succeeding columns contain harvest dates in
c ascending order. for perennials a special day counter is used so
c harvest dates into the second and succeeding growing years are  made
```

```
c by adding 365 for each year. also for perennials the first column
c of cuta must be 1.0 to signal the new day counter (attt) for awata
c and dira.
c
c ** plant loop ******************************************************
      do 10 l=1,ipl
c default value of acreage for each plant type
      if (acres(l,lz)) 12,11,12
   11 acres(l,lz)=92162.
c excludes non-harvested crops
   12 if (rirrc(l,lz)) 13,10,13
   13 jrig=arig(l,lz)
c number of harvests
      khrv=ahrv(l,lz)
c harvest dates,acres-inches added, and irrigation times entered
 5014 print *,'ENTER ',khrv,'harvesting dates for plant ',l,'in'
      print *,'region ',lz,' The first column is reserved for'
      print *,'final harvest;others including final in ascending'
      print *,'order. If 1.0 entered for column, then harvest days'
      print *,'greater than 365. may be used for multiyear plants'
      print *,'if not harvested, enter 0.0 twice'
      if(nu.eq.0)read(*,*,err=5014) (cuta(l,k,lz),k=1,khrv)
      if(nu.eq.1)read(7,*) (cuta(l,k,lz),k=1,khrv)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*) (cuta(l,k,lz),k=1,khrv)
      if(iu.eq.1)print *, (cuta(l,k,lz),k=1,khrv)
 5015 print *,'ENTER irrigation application, inches, added for ',jrig
      print *,'irrigations for plant ',l,'in region ',lz,'Enter 0.0'
      print *,'first column'
      if(nu.eq.0)read(*,*,err=5015) (awata(l,k,lz),k=1,jrig)
      if(nu.eq.1)read(7,*) (awata(l,k,lz),k=1,jrig)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*) (awata(l,k,lz),k=1,jrig)
      if(iu.eq.1)print *, (awata(l,k,lz),k=1,jrig)
 5016 print *,'ENTER irrigation day numbers for ',jrig,'irrigations'
      print *,'for plant ',l,'in region ',lz,' First column reserved'
      print *,'for planting date'
      if(nu.eq.0)read(*,*,err=5016) (dira(l,k,lz),k=1,jrig)
      if(nu.eq.1)read(7,*) (dira(l,k,lz),k=1,jrig)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*) (dira(l,k,lz),k=1,jrig)
      if(iu.eq.1)print *, (dira(l,k,lz),k=1,jrig)
   10 continue
c ** end plant loop ******************************************************
c
c    if multiples of the same plant are desired,then choose PLTID for a
c    plant n as n,22+n,44+n,66+n, etc....
c    if user wishes to create own plant n,then read in -n for PLTID,and
```

```
c     then prepare to read in WUTP,SCTP,etc...,as shown below
c     if PLTID is set to 0.0,then run will bypass PLTGRO subroutine,as
c     required for use of other subroutines. only one 0.0 may be read in
c      without any other input in this case for PLTID
c
c ** plant loop ********************************************************
   20 do 80 k=1,ipl
c excludes null plant index
      if (pltid(1,lz)) 41,40,41
c excludes all but user-created plant types
   41 if (pltid(k,lz).ge.1.0) go to 30
      pltid(k,lz)=-pltid(k,lz)
c user specified plant growth parameters entered; see below for definition
      print *,'*******************************************************'
      print *,'USER specified plant growth parameters; example for corn'
      print *,'or sorghum (plant #1) given to aid user: hf=1.,wutp=200.'
      print *,'sctp=1.,astp=0.,rxpp=.161,rfp=.0013,psep=0.5,bcp=0.6'
      print *,'strans=0.75,rtrans=0.25,topt=30.,tofl=20.,tofh=9.,'
      print *,'fleaf=0.,ulef=0.,respf=.05,respf2=0.05; other ex. can'
      print *,'be found in DATA statements in this subroutine'
      print *,'*******************************************************'
 5017 print *,'ENTER mean life-span, yrs, of plant ',k,'in region ',lz
      if(nu.eq.0)read(*,*,err=5017) hf(k,lz)
      if(nu.eq.1)read(7,*) hf(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,*) hf(k,lz)
      if(iu.eq.1)print *, hf(k,lz)
 5018 print *,'ENTER water util. efficiency, ml-H2O/gdwt-photosynthate'
      print *,'for plant ',k,' in region ',lz
      if(nu.eq.0)read(*,*,err=5018)wutp(k,lz)
      if(nu.eq.1)read(7,*)wutp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
      write(6,*)wutp(k,lz)
      if(iu.eq.1)print *,wutp(k,lz)
 5019 print *,'ENTER type plant: woody=0.,herbaceous=1. for plant ',k
      print *,'in region ',lz
      if(nu.eq.0)read(*,*,err=5019)sctp(k,lz)
      if(nu.eq.1)read(7,*)sctp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
      write(6,*)sctp(k,lz)
      if(iu.eq.1)print *,sctp(k,lz)
 5020 print *,'ENTER type plant: deciduous=0.,evergreen=1. for plant',k
      print *,'in region ',lz
      if(nu.eq.0)read(*,*,err=5020)astp(k,lz)
      if(nu.eq.1)read(7,*)astp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
```

```
      write(6,*)astp(k,lz)
      if(iu.eq.1)print *,astp(k,lz)
 5021 print *,'ENTER root humus loss coeff. #1,rxpp, for plant ',k
      print *,'in region ',lz,':(gdwt/m2-d)=.1*RFP*exp(RXPP*temp)*rh1'
      print *,'where temp=soil layer temp(deg.C),rh1= rapidly '
      print *,'decomposing fraction of root humus, gdwt/m2'
      if(nu.eq.0)read(*,*,err=5021)rxpp(k,lz)
      if(nu.eq.1)read(7,*)rxpp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*)rxpp(k,lz)
      if(iu.eq.1)print *,rxpp(k,lz)
 5022 print *,'ENTER root humus loss coeff. #2,rfp, for plant ',k
      print *,'in region ',lz
      if(nu.eq.0)read(*,*,err=5022)rfp(k,lz)
      if(nu.eq.1)read(7,*)rfp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5022) ans
      if(ans.gt.0.and.nu.eq.0)goto5022
      write(6,*)rfp(k,lz)
      if(iu.eq.1)print *,rfp(k,lz)
 5023 print *,'ENTER C4(0.5) or C3(1.0) plant type for plant',k
      print *,'in region',lz
      if(nu.eq.0)read(*,*,err=5023)psep(k,lz)
      if(nu.eq.1)read(7,*)psep(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5023) ans
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*)psep(k,lz)
      if(iu.eq.1)print *,psep(k,lz)
 5024 print *,'ENTER net photosynthate fraction after respiration'
      print *,'losses for plant',k,' in region',lz
      if(nu.eq.0)read(*,*,err=5024)bcp(k,lz)
      if(nu.eq.1)read(7,*)bcp(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*)bcp(k,lz)
      if(iu.eq.1)print *,bcp(k,lz)
 5025 print *,'ENTER fraction of photosyntahate shunted to above'
      print *,'ground biomass for plant',k,' in region',lz
      if(nu.eq.0)read(*,*,err=5025)strans(k,lz)
      if(nu.eq.1)read(7,*)strans(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5025) ans
      if(ans.gt.0.and.nu.eq.0)goto5025
      write(6,*)strans(k,lz)
      if(iu.eq.1)print *,strans(k,lz)
 5026 print *,'ENTER fraction of photosynthate shunted to below'
      print *,'ground biomass for plant',k,'in region',lz
      if(nu.eq.0)read(*,*,err=5026)rtrans(k,lz)
      if(nu.eq.1)read(7,*)rtrans(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
```

```
      if(nu.eq.0)read(*,*,err=5026) ans
      if(ans.gt.0.and.nu.eq.0)goto5026
      write(6,*)rtrans(k,lz)
      if(iu.eq.1)print *,rtrans(k,lz)
5027  print *,'ENTER optimum temperature(deg. C.) for photosynthesis'
      print *,'for plant',k,'in region',lz
      if(nu.eq.0)read(*,*,err=5027)topt(k,lz)
      if(nu.eq.1)read(7,*)topt(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5027) ans
      if(ans.gt.0.and.nu.eq.0)goto5027
      write(6,*)topt(k,lz)
      if(iu.eq.1)print *,topt(k,lz)
5028  print *,'ENTER absolute difference between optimum and lowest'
      print *,'temperature tolerance for photosynthesis for plant',k
      print *,'in region',lz
      if(nu.eq.0)read(*,*,err=5028)tofl(k,lz)
      if(nu.eq.1)read(7,*)tofl(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5028) ans
      if(ans.gt.0.and.nu.eq.0)goto5028
      write(6,*)tofl(k,lz)
      if(iu.eq.1)print *,tofl(k,lz)
5029  print *,'ENTER absolute difference between optimum and highest'
      print *,'temperature tolerance for photosynthesis for plant',k
      print *,'in region',lz
      if(nu.eq.0)read(*,*,err=5029)tofh(k,lz)
      if(nu.eq.1)read(7,*)tofh(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5029) ans
      if(ans.gt.0.and.nu.eq.0)goto5029
      write(6,*)tofh(k,lz)
      if(iu.eq.1)print *,tofh(k,lz)
5030  print *,'ENTER initial fraction of above ground biomass assigned'
      print *,'to leaves for plant',k,' in region',lz,' ; applies to'
      print *,'trees and shrubs only; enter 0.0 for all other types'
      if(nu.eq.0)read(*,*,err=5030)fleaf(k,lz)
      if(nu.eq.1)read(7,*)fleaf(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5030) ans
      if(ans.gt.0.and.nu.eq.0)goto5030
      write(6,*)fleaf(k,lz)
      if(iu.eq.1)print *,fleaf(k,lz)
5031  print *,'ENTER running average net productivity frac. assigned'
      print *,'to leaves for plant',k,' in region',lz,' ; applies to'
      print *,'trees and shrubs only; enter 0.0 for all other types'
      if(nu.eq.0)read(*,*,err=5031)ulef(k,lz)
      if(nu.eq.1)read(7,*)ulef(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5031) ans
      if(ans.gt.0.and.nu.eq.0)goto5031
      write(6,*)ulef(k,lz)
      if(iu.eq.1)print *,ulef(k,lz)
5032  print *,'ENTER live-to-dead stem and branch conversion coeff.'
```

```
      print *,'for plant',k,' in region',lz,' :sd3=.0055*respf/hf'
      print *,'where sd3=daily conversion,g/m2-day,hf=mean life-span'
      if(nu.eq.0)read(*,*,err=5032)respf(k,lz)
      if(nu.eq.1)read(7,*)respf(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5032) ans
      if(ans.gt.0.and.nu.eq.0)goto5032
      write(6,*)respf(k,lz)
      if(iu.eq.1)print *,respf(k,lz)
 5033 print *,'ENTER live-to-dead leaf conversion coefficient for'
      print *,'plant',k,' in region',lz,' for herbaceous plants'
      print *,'wd3=(1.1-1/tofl-5.)*respf*wnd (below opt. temp.)'
      print *,'wd3=(rspf2*(t-topt)+0.1)*respf*wnd (above opt. temp.)'
      print *,'where t=mean daily temp.(deg. C.),wnd=leaf biomass,g/m2'
      print *,'wd3= daily conversion,g/m2-day'
      if(nu.eq.0)read(*,*,err=5033)rspf2(k,lz)
      if(nu.eq.1)read(7,*)rspf2(k,lz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5033) ans
      if(ans.gt.0.and.nu.eq.0)goto5033
      write(6,*)rspf2(k,lz)
      if(iu.eq.1)print *,rspf2(k,lz)
      go to 40
   30 kpn=pltid(k,lz)
      ipn=kpn/22
      if (kpn.gt.22) kpn=kpn-22*ipn
c
c ** user specified input in above read statement; default values are entered
c ** in DATA statements for different plant types to give user guidence:
c
c mean lifespan (years), photosyn. use efficiency, woody/herbaceous plant
c index, evergreen/deciduous plant index, root humus respiration coefficient,
c root humus respiration coefficient, C3/C4 plant index, net daily photosyn.
c fraction, stem photosynthate fraction, root photosynthate fraction, optimum
c temperature for photosynthesis C., lower temperature tolerance C., upper
c temperature tolerance C., initial fraction of above ground biomass assigned
c to leaves, final fraction as above, live-to-dead mortality rate, same as
c above
c sets parameters for all regions to those of region 1
      hfa(k,lz)=hf(kpn,1)
      wutpa(k,lz)=wutp(kpn,1)
      sctpa(k,lz)=sctp(kpn,1)
      astpa(k,lz)=astp(kpn,1)
      rxppa(k,lz)=rxpp(kpn,1)
      rfpa(k,lz)=rfp(kpn,1)
      psepa(k,lz)=psep(kpn,1)
      bcpa(k,lz)=bcp(kpn,1)
      strana(k,lz)=strans(kpn,1)
      rtrana(k,lz)=1.-strans(k,lz)
      topta(k,lz)=topt(kpn,1)
      tofla(k,lz)=tofl(kpn,1)
      tofha(k,lz)=tofh(kpn,1)
      fleafa(k,lz)=fleaf(kpn,1)
      ulefa(k,lz)=ulef(kpn,1)
```

```
      respfa(k,lz)=respf(kpn,1)
      rspf2a(k,lz)=rspf2(kpn,1)
c
c dead standing biomass g/m2, dead standing bole(stem) biomass, dead standing
c branch biomass, leaf area index m2/m2, live above ground biomass g/m2, live
c below ground biomass,live biomass of bole, live biomass of branches, live
c biomass of root bole, live biomass of root branches, live biomass of
c leaves, standing dead root biomass, daily photosynthate production
c g/m2/day, photosynthate flow to roots, biomass of fruit g/m2, dead leaf
c biomass, dry fruit biomass, net productivity g/m2/year, year counter for
c estimating yearly weighted productivities---- values initialized----
   40 continue
      d1(k,lz)=0.0
      d1gb(k,lz)=0.0
      d1gbb(k,lz)=0.0
      d1gt(k,lz)=0.0
      tlai(k,lz)=0.0
      b1(k,lz)=1.0
      r1(k,lz)=1.0
      gm2t(k,lz)=1.0
      gm2b(k,lz)=1.0
      gm2r(k,lz)=1.0
      gm2rb(k,lz)=1.0
      wnd(k,lz)=1.0
      rd1(k,lz)=0.0
      b5(k,lz)=0.0
      b7(k,lz)=0.0
      sd(k,lz)=1.0
      wd1(k,lz)=0.0
      sd1(k,lz)=0.0
      wyav(k,lz)=0.0
      syav(k,lz)=0.0
c effect of temperature on photosynthesis initialized
      e5(k,lz)=0.0
c cummulative evapotranspiration initialized
      ea(k,lz)=0.0
c irrigation and harvest counters initialized
      irg(k,lz)=2
      ica(k,lz)=2
c day counter for perennial plants initialized
      attt(k,lz)=0.0
      if (pltid(k,lz)) 50,51,50
   51 rirrc(k,lz)=0.0
      arig(k,lz)=0.0
      rtrans(k,lz)=1.0
c crown cover, niche and succession factors initialized
   50 ccvv(k,lz)=0.0
      cxfac(k,lz)=0.0
      sfac(k,lz)=1.0
      cfac(k,lz)=1.0
      c8(k,lz)=1.0
      c6(k,lz)=1.0
      rank(k,lz)=0.0
c last harvest and irrigation dates initialized
```

```
      cutmax(k,lz)=0.0
      dirmax(k,lz)=0.0
      if (rirrc(k,lz)) 81,80,81
c final harvest and irrigation dates selected; required because first date
c has a dual function if perennial plants are simulated to allow julian dates
c to exceed 365 (see above for CUTMAX=1.0)
   81 cutmax(k,lz)=cuta(k,1,lz)
      dirmax(k,lz)=dira(k,1,lz)
      jrig=arig(k,lz)*16.
c
c ** irrigation loop ******************************************************
      do 60 jj=1,jrig
      if (dira(k,jj,lz).gt.dirmax(k,lz)) dirmax(k,lz)=dira(k,jj,lz)
   60 continue
c ** end irrigation loop **************************************************
      khrv=ahrv(k,lz)
c
c ** harvest loop *********************************************************
      do 70 kk=1,khrv
      if (cuta(k,kk,lz).gt.cutmax(k,lz)) cutmax(k,lz)=cuta(k,kk,lz)
   70 continue
c ** end harvest loop *****************************************************
   80 continue
      do84k=1,ipl
      hf(k,lz)=hfa(k,lz)
      wutp(k,lz)=wutpa(k,lz)
      sctp(k,lz)=sctpa(k,lz)
      astp(k,lz)=astpa(k,lz)
      rxpp(k,lz)=rxppa(k,lz)
      rfp(k,lz)=rfpa(k,lz)
      psep(k,lz)=psepa(k,lz)
      bcp(k,lz)=bcpa(k,lz)
      strans(k,lz)=strana(k,lz)
      rtrans(k,lz)=1.-strana(k,lz)
      topt(k,lz)=topta(k,lz)
      tofl(k,lz)=tofla(k,lz)
      tofh(k,lz)=tofha(k,lz)
      fleaf(k,lz)=fleafa(k,lz)
      ulef(k,lz)=ulefa(k,lz)
      respf(k,lz)=respfa(k,lz)
   84 continue
c ** end plant loop ******************************************************
c fruit tree productivity period initialized
      frtct=1.0
   90 continue
 5034 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN PLTGRO                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
```

```
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for PLTGRO?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5034)ans
      if(ans) 91,5500,91
   91 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-121212.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c ** end region loop ******************************************************
      afan=0.0
      afpr=0.0
      return
  100 continue
      if (aiii-1.0) 120,121,120
c
c ** plant loop ***********************************************************
  121 do 110 jj=1,ipl
c
c maximum yearly biomass densities of standing above ground, fruit, below
c ground, and leaves initialized
c
      bsmax(jj,lx)=0.0
      sdmx(jj,lx)=0.0
      r1max(jj,lx)=0.0
      wndmax(jj,lx)=0.0
      ccmax(jj,lx)=0.0
c annual net productivity g/m2/year initialized
      wurat(jj,lx)=0.0
  110 continue
c ** end plant loop *******************************************************
c selects plant type
  120 kpn=pltid(lpn,lx)
      if (kpn.gt.22) kpn=kpn-22
c returns if simulation does not require the PLTGRO subroutine
      if (kpn.eq.0) return
c
c  the effects of temperature on photosynthesis is estimated. The standing
c  live leaf biomass g/m2, is estimated for herbaceous and immature woody
c  plants (generally below 1000 g/m2).
c
      e5(lpn,lx)=1.0-(t(lx)-topt(lpn,lx))**2/tofl(lpn,lx)**2
      if (t(lx).gt.topt(lpn,lx)) e5(lpn,lx)=1.0-(t(lx)-topt(lpn,lx))**2
```

```
     1 /tofh(lpn,lx)**2
      if (e5(lpn,lx).le.0.0) e5(lpn,lx)=0.0
      if (kpn.eq.22) e5(lpn,lx)=1.0
      b11=b1(lpn,lx)/1000.
      wnd(lpn,lx)=928.*b11*exp(-.311*b11)+72.*b11
c selects for irrigated plants
      if (dira(lpn,1,lx).gt.0.0) go to 140
c non-irrigated plants
c selects for cool season plants
      if (kpn.gt.11) go to 130
c warm season plants
c increments photosynthate fraction to fruits of warm season plants during
c the growing season to day 240; reduces to day 270
      if (aiii.gt.sffp(lx).and.aiii.lt.240.) etrans(lpn,lx)=etrans(lpn
     1 ,lx)+.004
      if (aiii.ge.240..and.aiii.lt.270.) etrans(lpn,lx)=etrans(lpn,lx)-.
     1 004
c photsynthesis gate opened
      bap=1.
c selects for evergreen plants
      if (astp(lpn,lx)-1.0) 151,150,151
c selects for perennial plants
  151 if (hf(lpn,lx).gt.1.0) go to 150
c closes annual plants photosynthesis gate relative to frost free period
      if (aiii.lt.sffp(lx)-14..or.aiii.gt.effp(lx)-30.) bap=0.0
c eliminates fruit photsynthate shunt
      if (bap)153,152,153
  152 etrans(lpn,lx)=0.0
  153 go to 150
c cool season plants
c photosynthate shunt to fruits incremented relative to beginning of frost
c free period or decremented( between 180-210)
  130 cffp=sffp(lx)-30.
      if (aiii.gt.cffp.and.aiii.lt.180.) etrans(lpn,lx)=etrans(lpn,lx)+.
     1 004
      if (aiii.ge.180..and.aiii.lt.210.) etrans(lpn,lx)=etrans(lpn,lx)-.
     1 004
c opens photosynthesis gate
      bap=1.
c selects for evergreen perennial plants
      if (astp(lpn,lx)-1.0) 154,150,154
c selects for perennial plants
  154 if (hf(lpn,lx).gt.1.0) go to 150
c closes photsynthesis gate for cool season annuals relative to the frost
c free period
      if (aiii.lt.sffp(lx)-28..or.aiii.gt.effp(lx)-15.) bap=0.0
c eliminates photsynthate shunt to fruits
      if (bap) 150,155,150
  155 etrans(lpn,lx)=0.0
c irrigated plants photosynthesis gate opened
  140 bap=1.
c photosynthesis gate closed before planting and after last harvest for
c irrigated annual plants
      if(attt(lpn,lx).lt.dira(lpn,1,lx).or.attt(lpn,lx).gt.cutmax(lpn
```

```
     1 ,lx)) bap=0.0
      if (hf(lpn,lx).gt.1.0) bap=1.0
c conditionally closes photosynthesis gate
      if (bap) 142,141,142
  141 etrans(lpn,lx)=0.0
c increments photosynthate shunt to fruits relative to planting date
  142 if (attt(lpn,lx).gt.dira(lpn,1,lx)) etrans(lpn,lx)=etrans(lpn,lx)+
     1 .004
c selects for non-harvested annuals and for irrigated plants before final
c harvest
      if (cutmax(lpn,lx)) 156,150,156
  156 if (attt(lpn,lx).le.cutmax(lpn,lx)) go to 150
c resets photosynthate shunt to fruits and their biomass density
      etrans(lpn,lx)=0.0
      sd(lpn,lx)=1.0
c all plant types selected
c immature fruit trees not allowed fruit until after 5 years of growth
  150 if (kpn.eq.4.and.frtct.le.5.) etrans(lpn,lx)=0.0
c excludes herbaceous plants
      if (sctp(lpn,lx)-1.) 171,170,171
c selects for evergreen plants
  171 if (astp(lpn,lx)-1.0) 161,160,161
c woody-deciduous plants; estimates leaf biomass density
  161 if (aiii.lt.sffp(lx).or.aiii.gt.effp(lx)) wnd(lpn,lx)=1.0
      if (wnd(lpn,lx)-1.0) 160,170,160
c woody-evergreen plants; estimates leaf biomass density
  160 wnd(lpn,lx)=fleaf(lpn,lx)*b1(lpn,lx)
      if(wnd(lpn,lx).le.1.0)wnd(lpn,lx)=1.0
      tpg=tpgnw(lpn,lx)
      if (tpg.le.0.0) go to 170
      if (wnd(lpn,lx).ge.ulef(lpn,lx)*tpg) wnd(lpn,lx)=ulef(lpn,lx)*tpg
c selects for deciduous and evergreen trees ; limits fruiting to trees older
c than 10 years
  170 if (ai1.le.10..and.hf(lpn,lx).gt.125.) etrans(lpn,lx)=0.0
      if (hf(lpn,lx).gt.125.) go to 180
c selects for deciduous and evergreen shrubs; limits fruiting to shrubs older
c than 5 years
      if (ai1.le.5..and.hf(lpn,lx).gt.45.) etrans(lpn,lx)=0.0
c
c for warm and cool season plants the yearly succession factor is calculated
c for those plants set in succession; factor set to 1.0 for irrigated plants
c (RIRRC), or for plants not in succession (RSCDE)
c
  180 if (rscde(lx)) 182,181,182
  182 if (rirrc(lpn,lx)-1.0) 183,181,183
  181 sfac(lpn,lx)=1.0
      rank(lpn,lx)=1.0
c excludes all but first year and 1st day of 1st year from succession
  183 if (ai1-1.0) 210,212,210
  212 if (aiii.gt.1.0) goto210
c excludes irrigated and non-succession plants
      if (rscde(lx)) 185,184,185
  185 if (rirrc(lpn,lx)-1.0) 186,184,186
  184 go to 210
```

```
c secondary succession factor initialized for plants with different life
c spans
 186 sfac(lpn,lx)=1.0
     if(hf(lpn,lx).ge.30.0)sfac(lpn,lx)=0.0
     if(hf(lpn,lx).ge.30.0)rank(lpn,lx)=0.0
     xfac(lpn,lx)=0.0
c selects for normal secondary succession
     if (adsuc(lx)) 211,210,211
c advanced secondary succession factor estimated for grasses, shrubs and
c trees
 211 xfac(lpn,lx)=0.7
     if (hf(lpn,lx).le.70.) xfac(lpn,lx)=0.2
     if (hf(lpn,lx).le.30.) xfac(lpn,lx)=0.1
c succession factor sum initialized and estimated
     sumxf=0.0
     if (lpn.ne.ipl) go to 210
c
c ** plant loop ***********************************************************
     do 190 jj=1,ipl
     sumxf=sumxf+xfac(jj,lx)
 190 continue
c ** end plant loop *******************************************************
c adjusts advanced succession factors if SUMXF less than 1.0
     if (sumxf.ge.1.0) go to 210
c
c ** plant loop ************************************************************
     do 200 jj=1,ipl
     xfac(jj,lx)=xfac(jj,lx)/sumxf
 200 continue
c ** end plant loop *******************************************************
c
c the root respiration factor (AC), and the plant mortality factor (AP)
c are estimated.
c
 210 ac=.10*rfp(lpn,lx)*exp(rxpp(lpn,lx)*tsrf)
     ap=2.0/(365.0*hf(lpn,lx))
c
c leaf area index (TLAI) is estimated (includes dead leaf component, WD1)
     tlai(lpn,lx)=0.013*(wnd(lpn,lx)+wd1(lpn,lx))
c estimates photsynthetic efficiency for C4 plants, g/ly/day
     if (psep(lpn,lx).lt.1.0) sa=.001*solrad
c nitrate and phosphate growth increments initialized
     untr=0.0
     untp=0.0
c estimates fraction of growing period expended
     groper=(aiii-sffp(lx))/(effp(lx)-sffp(lx))
c sets fraction of growing period to 1.0 outside of growth period
     if (aiii.gt.effp(lx)) groper=1.0
     if (aiii.lt.sffp(lx)) groper=1.0
c estimates early growth nitrate and phosphate increments for 1st 20% of
c growth period
     if (groper.le.0.2) untr=.024-.120*groper
     if (groper.le.0.2) untp=.005-.025*groper
c estimates nitrate concentration in root, g/g-dry root tissue
```

```
      sncn=z6(lpn,3,lx)*b1(lpn,lx)/wnd(lpn,lx)+untr
      if (sncn.le..006) sncn=0.006
      if (sncn.gt..06) sncn=.06
c estimates soluble phosphate concentration in root, g/g-dry root tissue
      spcn=z6(lpn,4,lx)*b1(lpn,lx)/wnd(lpn,lx)+untp
      if (spcn.gt..005) spcn=.005
      if (spcn.le..0005) spcn=.0005
c effect of nitrate and phosphate root concentrations on photosynthesis
      unitef=1.32*(1.0-exp((-.693/.014)*sncn))
      phoef=1.32*(1.0-exp((-.693/.0014)*spcn))
      efonut=unitef*phoef
      if (efonut.le.0.001) efonut=0.001
c potential gross daily photosynthesis, g/m2/day
      b5(lpn,lx)=wnd(lpn,lx)*bap*sa*e5(lpn,lx)
c selects for normal secondary succession
      if (adsuc(lx)) 261,262,261
c excludes all julian days except 1.0
  261 if (aiii.gt.1) go to 260
c selects for perennials
      if (hf(lpn,lx).gt.2.0) go to 220
c limits annuals succession factor below or equal to pre-set value (XFAC)
      if (sfac(lpn,lx).ge.xfac(lpn,lx)) sfac(lpn,lx)=xfac(lpn,lx)
      go to 230
c limits perennial succession factor above or equal to pre-set value (XFAC)
  220 if (sfac(lpn,lx).le.xfac(lpn,lx)) sfac(lpn,lx)=xfac(lpn,lx)
c normal succession strategy is initialized after last plant in region
  230 if (lpn.ne.ipl) go to 260
c succession factor sum initialized and totaled
  262 if(adsuc(lx).ge.1.0)goto260
      if(rscde(lx).le.0.0)goto260
      if(lpn.ne.ipl)goto260
      sumfac=0.0
c
c ** plant loop ********************************************************
      do 240 jj=1,ipl
      if(rank(jj,lx).le.0.0)goto240
      sumfac=sumfac+sfac(jj,lx)
  240 continue
c ** end plant loop ****************************************************
      sfan =0.0
      sfpr=0.0
      sfsh=0.0
      sftr=0.0
c
c ** plant loop ********************************************************
      do 250 jj=1,ipl
      ihf=hf(jj,lx)
      if(ai1.gt.1.0.and.aiii.gt.1.0)sfac(jj,lx)=sfac(jj,lx)/sumfac
        do 540 k=1,ipl
c selects for ranked members
      if (rank(k,lx)) 541,540,541
c
c ** plant loop ********************************************************
  541 do 530 l=1,ipl
```

```
c selects for niche partners
      if (rss(l,lx)) 531,530,531
  531 if (k.eq.l) go to 530
      if (rss(l,lx)-rss(k,lx)) 530,532,530
c sets succession factor fraction for other members of niche set
  532 sfac(l,lx)=sfac(k,lx)
  530 continue
c ** end plant loop ********************************************************
  540 continue
c
      if(ihf.lt.30)sfan=sfac(jj,lx)
      if(ihf.ge.30.and.ihf.lt.70)sfpr=sfac(jj,lx)
      if(ihf.ge.70.and.ihf.lt.150)sfsh=sfac(jj,lx)
      if(ihf.ge.150)sftr=sfac(jj,lx)
  250 continue
      if(adsuc(lx).gt.0.0)goto260
      do251jj=1,ipl
      ihf=hf(jj,lx)
c perennial grasses
      if(ihf.ge.30.and.ihf.lt.70.and.ai1.le.1.0)sfac(jj,lx)=0.0
c shrubs
      if(afan.ge.1.0)goto252
      if(ihf.ge.70.and.ihf.lt.150.and.3.0*sfan.gt.sfpr)sfac(jj,lx)=0.0
      if(ihf.ge.70.and.ihf.lt.150.and.3.0*sfan.lt.sfpr)afan=1.0
c trees
  252 if(afpr.ge.1.0)goto251
      if(ihf.ge.150.and.3.0*sfpr.gt.sfsh)sfac(jj,lx)=0.0
      if(ihf.ge.150.and.3.0*sfpr.lt.sfsh)afpr=1.0
  251 continue
c     if(aiii.ge.365)pause
c ** end plant loop ********************************************************
c transpiration, ml/m2/day adjusted for niche competition (CFAC), and
c secondary sucession (SFAC) factors
  260 eamn=1.0
      tla=0.013*wnd(lpn,lx)
      if(tla.le.1.0)eamn=1.0/tla
      t5p(lpn,lx)=ea(lpn,lx)*1000.*sfac(lpn,lx)*cfac(lpn,lx)*eamn
c plant water requirement (WUTP), ml/g-photosynthate synthesized (B5)
c adjusted for nitrate and phosphate availability
      watrq=wutp(lpn,lx)*b5(lpn,lx)/efonut
      if (b5(lpn,lx).le.0.0) go to 270
c gross photosynthesis limited by water availability (T5P)
      if(watrq.gt.t5p(lpn,lx))b5(lpn,lx)=(t5p(lpn,lx)/watrq)*b5(lpn,lx)
c net photosynthesis estimated after adjustment for respiration fraction
c (BCP), and photosynthetic efficiency (PSEP) for C3 and C4 plants
  270 b5(lpn,lx)=b5(lpn,lx)*bcp(lpn,lx)/psep(lpn,lx)
c cumulative daily net productivity (WURAT)
      wurat(lpn,lx)=wurat(lpn,lx)+b5(lpn,lx)
c crown cover for annuals estimated (CCVV) based on (WURAT)
      if (hf(lpn,lx)-1.0) 272,271,272
  271 ccvv(lpn,lx)=(1.-exp(-.00154*wurat(lpn,lx)))
c fraction of photsynthate transferred to fruits bounded
  272 if (etrans(lpn,lx).gt.0.95) etrans(lpn,lx)=0.95
      if (etrans(lpn,lx).le.0.0) etrans(lpn,lx)=0.0
```

```
      etrn=etrans(lpn,lx)
c fruit transfer reset for perennials (excluding biennials)
      if (hf(lpn,lx).gt.3.0) etrn=0.5*etrn
c fruit transfer reset for evergreen shrubs and trees
      if (astp(lpn,lx)-1.0) 274,273,274
  273 etrn=2.*etrn
  274 if (etrn.ge..95) etrn=.95
c photosynthate transported to roots (B7) estimated
      b7(lpn,lx)=rtrans(lpn,lx)*(1.-etrn)*b5(lpn,lx)
c
c the biomass variables are incremented for the daily net productivity
c for live and dead compartments.
c
c above ground biomass (B1), below ground biomass (R1) and fruit (SD)
c incremented, g/m2
      b1(lpn,lx)=b1(lpn,lx)+b5(lpn,lx)*(1.0-etrn)*strans(lpn,lx)
      r1(lpn,lx)=r1(lpn,lx)+b7(lpn,lx)
      sd(lpn,lx)=sd(lpn,lx)+b5(lpn,lx)*etrn
c dead standing biomass of fruit (SD3), above ground (D3), below ground
c (RD3), and leaves (WD3)  of annuals and biennials
      sd3(lpn,lx)=.005*sd(lpn,lx)
      d3(lpn,lx)=.005*b1(lpn,lx)*(1.-wnd(lpn,lx)/b1(lpn,lx))
      rd3(lpn,lx)=r1(lpn,lx)*.005
      wd3(lpn,lx)=.005*b1(lpn,lx)*wnd(lpn,lx)/b1(lpn,lx)
c selects for herbaceous plants
      if (sctp(lpn,lx)-1.0) 291,290,291
c selects for deciduous plants
  291 if (astp(lpn,lx)) 281,280,281
c woody evergreen plants; dead standing above ground, dead standing leaf
c biomass, and dead standing below ground biomass estimated
  281 decfc=365.
      d3(lpn,lx)=(b1(lpn,lx)-wnd(lpn,lx))*ap*respf(lpn,lx)
      wd3(lpn,lx)=0.693*wnd(lpn,lx)/decfc
      rd3(lpn,lx)=r1(lpn,lx)*ap*8.
      go to 300
c woody deciduous plants; dead standing above ground, leaf, fruit, and below
c ground biomass estimated
  280 d3(lpn,lx)=b1(lpn,lx)*ap*respf(lpn,lx)*(1.-wnd(lpn,lx)/b1(lpn,lx))
      wd3(lpn,lx)=b1(lpn,lx)*ap*respf(lpn,lx)*wnd(lpn,lx)/b1(lpn,lx)
c leaf and fruit biomass limited to frost free period; also dead standing
      if (aiii-effp(lx)) 283,282,283
  282 wd3(lpn,lx)=wnd(lpn,lx)
      sd3(lpn,lx)=sd(lpn,lx)
  283 if (aiii.lt.sffp(lx).or.aiii.gt.effp(lx)) wnd(lpn,lx)=1.0
      if (aiii.lt.sffp(lx).or.aiii.gt.effp(lx)) sd(lpn,lx)=1.0
c dead standing below ground biomass estimated
      rd3(lpn,lx)=r1(lpn,lx)*ap*15.
c all but fruit trees excluded
      if (kpn.ne.4) go to 300
c age counter for fruit trees
      if (aiii-365.) 285,284,285
  284 frtct=frtct+1.
c excludes fruit trees younger than 25 years
  285 if (frtct.lt.25.) go to 300
```

```
c 25 year trees removed, 1/10th converted to above ground and leaf dead
c standing biomass; 1/2 of live roots converted to dead standing roots
      d3(lpn,lx)=d3(lpn,lx)*0.1*b1(lpn,lx)*(1.-wnd(lpn,lx)/b1(lpn,lx))
      wd3(lpn,lx)=wd3(lpn,lx)*.1*b1(lpn,lx)*wnd(lpn,lx)/b1(lpn,lx)
      rd3(lpn,lx)=rd3(lpn,lx)+0.5*r1(lpn,lx)
c live above ground parameters initilized, tree age re-initialized
      b1(lpn,lx)=1.0
      r1(lpn,lx)=1.0
      frtct=0.0
      go to 310
c herbaceous plants; dead standing leaf and above ground biomass is estimated
c based on lower (TOFL) tolerance and optimum temperature (TOPT) for
c photosynthesis
  290 tls=1.0/(tofl(lpn,lx)-5.)
      rab=b1(lpn,lx)-wnd(lpn,lx)
      if (rab.le.0.0) rab=0.0
      if (t(lx).lt.tofl(lpn,lx)-5.) wd3(lpn,lx)=(1.1-tls*t(lx))*respf
     1 (lpn,lx)*wnd(lpn,lx)
      if (t(lx).lt.tofl(lpn,lx)-5.) d3(lpn,lx)=wd3(lpn,lx)*rab/wnd(lpn
     1 ,lx)
      tz=rspf2(lpn,lx)
      rf=respf(lpn,lx)
      if (t(lx).gt.topt(lpn,lx)) wd3(lpn,lx)=wnd(lpn,lx)*(tz*(t(lx)-topt
     1 (lpn,lx))+.1)*rf
      if (t(lx).gt.topt(lpn,lx)) d3(lpn,lx)=wd3(lpn,lx)*rab/wnd(lpn,lx)
c excludes cheatgrass (B. tectorum)
      if (lpn.eq.22) go to 300
c excludes all but annuals
      if (aiii-effp(lx)) 300,302,300
  302 if (hf(lpn,lx)-1.0) 300,303,300
  303 d3(lpn,lx)=b1(lpn,lx)-wnd(lpn,lx)
      wd3(lpn,lx)=wnd(lpn,lx)
      rd3(lpn,lx)=r1(lpn,lx)
      sd3(lpn,lx)=sd(lpn,lx)
  300 continue
      if (aiii.le.sffp(lx).or.aiii.ge.effp(lx)) etrans(lpn,lx)=0.0
      if (b1(lpn,lx).le.1.0) d3(lpn,lx)=0.0
      if (r1(lpn,lx).le.1.0) rd3(lpn,lx)=0.0
      if (sd(lpn,lx).le.1.0) sd3(lpn,lx)=0.0
      if (b1(lpn,lx).le.1.0) wd3(lpn,lx)=0.0
      if (d3(lpn,lx).le.0.0) d3(lpn,lx)=0.0
      if (wd3(lpn,lx).le.0.0) wd3(lpn,lx)=0.0
      if (sd3(lpn,lx).le.0.0) sd3(lpn,lx)=0.0
      if (rd3(lpn,lx).le.0.0) rd3(lpn,lx)=0.0
c
c  this section of code assures that  biomass is only incremented to
c  crops after they have been planted. biomass is taken to be equal to
c  1 g/m2 for otherwise.  after harvest biomass is reset to 20
c  g/m2 for cropped plants such as alfalfa
c
c excludes non-irrigated plants
  310 if (rirrc(lpn,lx)) 401,400,401
c excludes irrigated perrenials
  401 if (cutmax(lpn,lx)-dirmax(lpn,lx)) 402,400,402
```

```
c excludes perennial plants (includes biennials)
 402 if (hf(lpn,lx).gt.2.0) go to 360
c    irrigated annuals
c excludes non-harvest days
    if (attt(lpn,lx)-cutmax(lpn,lx)) 350,351,350
c
c ** contaminant loop ***************************************************
 351 do 320 k=1,inz
c
c transports 1/10 of live standing internal contaminant concentrations to dead
c compartment (Z4) ; 1/10th external concentrations on stems, branches, and
c seeds from resuspension to external dead compartment (Z42)
c
    z4(lpn,k,lx)=z4(lpn,k,lx)+.1*(z3(lpn,k,lx)+zsd(lpn,k,lx))
    z42(lpn,k,lx)=z42(lpn,k,lx)+.1*(b2(lpn,k,lx)+bb2(lpn,k,lx)+afr(lpn
   1 ,k,lx))
c
c transports 1/10 of external concentrations on stems,branches, and seeds
c from saltation-creep to external dead compartment (Z43)


c
    z43(lpn,k,lx)=z43(lpn,k,lx)+.1*(b3av(lpn,k,lx)+bb3av(lpn,k,lx)
   1 +afsav(lpn,k,lx))
c
c transports 1/10 of internal live leaf contaminant concentrations to dead leaf
c compartment (WZ4); 1/10 resuspension and saltation-creep on live leafs
c transported to dead leaf compartments (WZ42,WZ43), respectively
c
    wz4(lpn,k,lx)=wz4(lpn,k,lx)+.1*wz3(lpn,k,lx)
    wz42(lpn,k,lx)=wz42(lpn,k,lx)+.1*a2(lpn,k,lx)
    wz43(lpn,k,lx)=wz43(lpn,k,lx)+.1*a3av(lpn,k,lx)
 320 continue
c ** end contaminant loop ***************************************************
c
c transports 1/10 live above standing biomass (excluding leaves) to dead
c standing biomass (D3); 1/10 live leaf biomass transported to standing dead
c leaf compartment (WD3)
c
    d3(lpn,lx)=b1(lpn,lx)*0.1*(1.-wnd(lpn,lx)/b1(lpn,lx))
    wd3(lpn,lx)=b1(lpn,lx)*0.1*wnd(lpn,lx)/b1(lpn,lx)
c
c ** soil layer loop ****************************************************
    iicde=rcode(lpn,lx)
    do 340 j=1,iicde
c total root fraction in layer(j)
    rotfr=zff(lpn,j,lx)*zf(lpn,j,lx)
c total root penetrance(ZFF=1) for annuals
    if (hf(lpn,lx)-1.0) 342,341,342
 341 rotfr=zf(lpn,j,lx)
c
c ** contaminant loop ***************************************************
 342 do 330 k=1,inz
c transports live standing root internal and external contaminant content to
```

```
c dead standing contaminant content (Z4R)
     z4r(lpn,j,k,lx)=z4r(lpn,j,k,lx)+(z2(lpn,k,lx)+rb3(lpn,k,lx)+rbb3
    1 (lpn,k,lx))*rotfr
 330 continue
c ** end contaminant loop ******************************************************
 340 continue
c ** end soil layer loop ****************************************************
c
c converts live standing root biomass to dead standing biomass(RD3);resets
c standing root biomass
c
     rd3(lpn,lx)=r1(lpn,lx)
     r1(lpn,lx)=1.0
     go to 400
c
c resets above and below ground standing biomass; sets productivity to
c 0.0 outside of growing period
c
 350 if (attt(lpn,lx).ge.dira(lpn,1,lx).and.attt(lpn,lx).lt.cutmax(lpn
    1 ,lx)) go to 360
     b1(lpn,lx)=1.
     sd(lpn,lx)=1.
     wnd(lpn,lx)=1.
     r1(lpn,lx)=1.
     b5(lpn,lx)=0.0
c initializes dead standing biomasses to 0.0;also annual productivity and
c fruit transport factor reset
     d3(lpn,lx)=0.0
     wd3(lpn,lx)=0.0
     wurat(lpn,lx)=0.0
     etrans(lpn,lx)=0.0
     go to 400
c perennial irrigated plants; harvest date set
 360 icaaa=ica(lpn,lx)
c multiple or singly harvested perrenials
c excludes all julian days but harvest dates
     if (attt(lpn,lx)-cuta(lpn,icaaa,lx)) 400,403,400
c resets harvest date to succeeeding value
 403 ica(lpn,lx)=ica(lpn,lx)+1
c estimates total above ground live and dead biomass
     b1sdt=b1(lpn,lx)+sd(lpn,lx)+wd1(lpn,lx)+d1(lpn,lx)
c excludes harvesting below 20 g/m2 (stubble)
     if (b1sdt.le.20.) go to 400
c excludes woody plants
     if (sctp(lpn,lx)) 381,380,381
c
c ** contaminant loop ******************************************************
 381 do 370 k=1,inz
c all contaminant internal and external parameters reset to ratio 20/b1sdt
     z3(lpn,k,lx)=z3(lpn,k,lx)*20./b1sdt
     wz3(lpn,k,lx)=wz3(lpn,k,lx)*20./b1sdt
     zsd(lpn,k,lx)=zsd(lpn,k,lx)*20./b1sdt
     a2(lpn,k,lx)=a2(lpn,k,lx)*20./b1sdt
     a3(lpn,k,lx)=a3(lpn,k,lx)*20./b1sdt
```

```
      b3(lpn,k,lx)=b3(lpn,k,lx)*20./b1sdt
      bb3(lpn,k,lx)=bb3(lpn,k,lx)*20./b1sdt
      b2(lpn,k,lx)=b2(lpn,k,lx)*20./b1sdt
      bb2(lpn,k,lx)=bb2(lpn,k,lx)*20./b1sdt
      afr(lpn,k,lx)=afr(lpn,k,lx)*20./b1sdt
      afs(lpn,k,lx)=afs(lpn,k,lx)*20./b1sdt
      z4(lpn,k,lx)=z4(lpn,k,lx)*20./b1sdt
      z42(lpn,k,lx)=z42(lpn,k,lx)*20./b1sdt
      z43(lpn,k,lx)=z43(lpn,k,lx)*20./b1sdt
      wz4(lpn,k,lx)=wz4(lpn,k,lx)*20./b1sdt
      wz42(lpn,k,lx)=wz42(lpn,k,lx)*20./b1sdt
      wz43(lpn,k,lx)=wz43(lpn,k,lx)*20./b1sdt
  370 continue
c ** end contaminant loop ****************************************************
c live and dead standing biomass parameters reset to ratio 20/b1sdt
      d1(lpn,lx)=d1(lpn,lx)*20./b1sdt
      sd1(lpn,lx)=sd1(lpn,lx)*20./b1sdt
      wd1(lpn,lx)=wd1(lpn,lx)*20./b1sdt
      wnd(lpn,lx)=wnd(lpn,lx)*20./b1sdt
      b1(lpn,lx)=b1(lpn,lx)*20./b1sdt
      sd(lpn,lx)=sd(lpn,lx)*20./b1sdt
c fruit transport fraction initialized
      etrans(lpn,lx)=0.0
      go to 400
c woody irrigated plants; dry and live fruit biomasses initialized
  380 continue
      sd(lpn,lx)=1.0
      sd1(lpn,lx)=0.0
c
c ** contaminant loop ********************************************************
      do 390 k=1,inz
c transports internal contaminant fruit content to internal dead standing
c biomass (Z4) according dry seed/live seed biomass ratio
      z4(lpn,k,lx)=z4(lpn,k,lx)+sd3(lpn,lx)*zsd(lpn,k,lx)/sd(lpn,lx)
c transports external resuspension and saltation-creep external contaminant
c content from fruits to external dead standing compartments (Z42,Z43)
      z42(lpn,k,lx)=z42(lpn,k,lx)+sd3(lpn,lx)*afr(lpn,k,lx)/sd(lpn,lx)
      z43(lpn,k,lx)=z43(lpn,k,lx)+sd3(lpn,lx)*afsav(lpn,k,lx)/sd(lpn,lx)
c internal contaminant content in fruits initialized
      zsd(lpn,k,lx)=0.0
  390 continue
c ** end contaminant loop ****************************************************
c all irrigated and non-irrigated plants
  400 continue
c
c ** contaminant loop ********************************************************
      do 410 k=1,inz
c all internal and external contaminant concentrations bounded
      if (b1(lpn,lx).le.1.0) z3(lpn,k,lx)=0.0
      if (r1(lpn,lx).le.1.0) z2(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) wz3(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) a2(lpn,k,lx)=0.0
      if (wd1(lpn,lx).le.1.0) wz42(lpn,k,lx)=0.0
      if (wd1(lpn,lx).le.1.0) wz43(lpn,k,lx)=0.0
```

```
      if (b1(lpn,lx).le.1.0) a3(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) zsd(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) a3(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) b3(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) bb3(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) b2(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) bb2(lpn,k,lx)=0.0
      if (d1(lpn,lx)+sd1(lpn,lx).le.2.0) z42(lpn,k,lx)=0.0
      if (d1(lpn,lx)+sd1(lpn,lx).le.2.0) z43(lpn,k,lx)=0.0
      if (d1(lpn,lx)+sd1(lpn,lx).le.2.0) z4(lpn,k,lx)=0.0
      if (wd1(lpn,lx).le.1.0) wz4(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) afr(lpn,k,lx)=0.0
      if (b1(lpn,lx).le.1.0) afs(lpn,k,lx)=0.0
      if (z4(lpn,k,lx).le.0.0) z4(lpn,k,lx)=0.0
      if (z4g(lpn,k,lx).le.0.0) z4g(lpn,k,lx)=0.0
      if (z42(lpn,k,lx).le.0.0) z42(lpn,k,lx)=0.0
      if (z43(lpn,k,lx).le.0.0) z43(lpn,k,lx)=0.0
      if (wz4(lpn,k,lx).le.0.0) wz4(lpn,k,lx)=0.0
      if (wz42(lpn,k,lx).le.0.0) wz42(lpn,k,lx)=0.0
      if (wz43(lpn,k,lx).le.0.0) wz43(lpn,k,lx)=0.0
      if (z3(lpn,k,lx).le.0.0) z3(lpn,k,lx)=0.0
      if (wz3(lpn,k,lx).le.0.0) wz3(lpn,k,lx)=0.0
      if (a2(lpn,k,lx).le.0.0) a2(lpn,k,lx)=0.0
      if (a3(lpn,k,lx).le.0.0) a3(lpn,k,lx)=0.0
      if (z2(lpn,k,lx).le.0.0) z2(lpn,k,lx)=0.0
      if (rb3(lpn,k,lx).le.0.0) rb3(lpn,k,lx)=0.0
      if (rbb3(lpn,k,lx).le.0.0) rbb3(lpn,k,lx)=0.0
  410 continue
c ** end contaminant loop ***************************************************
c above ground, below ground, and fruit standing biomasses adjusted for
c losses from increases in dead standing compartments
      b1lpn=b1(lpn,lx)
      b1(lpn,lx)=b1(lpn,lx)-d3(lpn,lx)-wd3(lpn,lx)
      r1(lpn,lx)=r1(lpn,lx)-rd3(lpn,lx)
      sd(lpn,lx)=sd(lpn,lx)-sd3(lpn,lx)
c above ground, dead standing, and fruit live biomasses bounded
      if (b1(lpn,lx).le.1.0) b1(lpn,lx)=1.0
      if (r1(lpn,lx).le.1.0) r1(lpn,lx)=1.0
      if (sd(lpn,lx).le.1.0) sd(lpn,lx)=1.0
c dead standing compartments incremented for live standing losses from
c stems,fruits,leaves,bole,and branch biomasses
      d1lp=d1(lpn,lx)
      sd1lpn=sd1(lpn,lx)
      wd1lpn=wd1(lpn,lx)
      dgblp=d1gb(lpn,lx)
      dgbblp=d1gbb(lpn,lx)
c dead bole biomass bounded
      if (d3(lpn,lx).le.0.0) d3(lpn,lx)=0.0
c dead standing stem, seeds,and leaves incremented
      d1(lpn,lx)=d1(lpn,lx)+d3(lpn,lx)
      sd1(lpn,lx)=sd1(lpn,lx)+sd3(lpn,lx)
      wd1(lpn,lx)=wd1(lpn,lx)+wd3(lpn,lx)
c dead standing root biomass divided between decomposable and resistant
c to decay fractions (RD1,RD2), respectively
```

```
      rd1(lpn,lx)=rd1(lpn,lx)+rd3(lpn,lx)*(wnd(lpn,lx)/b1lpn)
      rd2(lpn,lx)=rd2(lpn,lx)+rd3(lpn,lx)*(1.0-wnd(lpn,lx)/b1lpn)
c leaf area index estimated and bounded
      tlainu=0.013*wnd(lpn,lx)
      tlaina=tlainu
      if (tlaina.le.1.0) tlaina=1.0
c actual to potential evaporation ratio estimated and bounded
      evrat=(1.-(ea(lpn,lx)*tlainu/tlaina+evap(lpn,lx))/e2(lx))
      if (evrat.ge.1.0) evrat=1.0
      if (evrat.le.0.0) evrat=0.0
c surface temperature of soil estimated, deg. C.
      tsrf=t(lx)-4.*(wnd(lpn,lx)-300.)/600.
      if(wnd(lpn,lx).le.300.)tsrf=t(lx)+15.*evrat*(1.-wnd(lpn,lx)/300.)
      if(tsrf.le.0.0)tsrf=0.0
c effect of soil surface temperature on litter decomposition
      h2=.004+.0032*tsrf
c effect of soil moisture on litter decomposition estimated and bounded
      h3=1.+(1./(f1(lpn,1,lx)-w1(lpn,1,lx)))*(wa(lpn,1,lx)-w1(lpn,1,lx))
      if (h3.le.1.0) h3=1.0
c humus production from dead leaves
      rh1(lpn,1,lx)=rh1(lpn,1,lx)+h2*h3*(wd1lpn+sd1lpn)
c humus producton from dead stems,branches, and fruits
      rh2(lpn,1,lx)=rh2(lpn,1,lx)+.1*h2*h3*(dgblp+dgbblp)
c weighting constant for estimating decomposition of bole(stems) and branches
c dead standing biomass (bounded)
      bbbs=(100.*gm2b(lpn,lx)+10.*gm2t(lpn,lx))/(gm2t(lpn,lx)+gm2b(lpn
     1 ,lx))
      if (bbbs.le.1.0) bbbs=1.0
c decomposition of dead standing stem-branches estimated
      d1(lpn,lx)=d1(lpn,lx)-(.693*d1lp/(hf(lpn,lx)*365.))*bbbs
c total stem-branch biomass density estimated; bounded
      bbs=gm2t(lpn,lx)+gm2b(lpn,lx)
      if (bbs.le.1.0) bbs=1.0
c dcomposition of dead stem(bole) estimated;bounded
      d1gb(lpn,lx)=d1gb(lpn,lx)-h2*h3*dgblp/10.
      if (d1gb(lpn,lx).le.1.0) d1gb(lpn,lx)=1.0
c decompositionn of dead branches(twigs) estimated;bounded
      d1gbb(lpn,lx)=d1gbb(lpn,lx)-h2*h3*dgbblp/10.
      if (d1gbb(lpn,lx).le.1.0) d1gbb(lpn,lx)=1.0
c dead stem and branches incremented for dead standing stem-branch losses
      d1gb(lpn,lx)=d1gb(lpn,lx)+(6.93*d1lp/(hf(lpn,lx)*365.))*gm2t(lpn
     1 ,lx)/bbs
      d1gbb(lpn,lx)=d1gbb(lpn,lx)+(69.3*d1lp/(hf(lpn,lx)*365.))*gm2b(lpn
     1 ,lx)/bbs
      if (d1gb(lpn,lx).le.1.0) d1gb(lpn,lx)=1.0
      if (d1gbb(lpn,lx).le.1.0) d1gbb(lpn,lx)=1.0
      d1gt(lpn,lx)=d1gb(lpn,lx)+d1gbb(lpn,lx)
c dead standing leaves and fruits adjusted for decomposition losses
      wd1(lpn,lx)=wd1(lpn,lx)-h2*h3*wd1lpn
      sd1(lpn,lx)=sd1(lpn,lx)-h2*h3*sd1lpn/10.
c biomass density parameters bounded
      if (d1(lpn,lx).le.1.0) d1(lpn,lx)=1.0
      if (sd1(lpn,lx).le.1.0) sd1(lpn,lx)=1.0
      if (wd1(lpn,lx).le.1.0) wd1(lpn,lx)=1.0
```

```
      if (d1gb(lpn,lx).le.1.0) d1gb(lpn,lx)=1.0
      if (d1gbb(lpn,lx).le.1.0) d1gbb(lpn,lx)=1.0
      if (b1(lpn,lx).le.1.0) b1(lpn,lx)=1.0
      if (b1(lpn,lx).le.1.0) wnd(lpn,lx)=1.0
      if (r1(lpn,lx).le.1.0) r1(lpn,lx)=1.0
      if (b1(lpn,lx).le.1.0) sd(lpn,lx)=1.0
c live standing biomass less leaf biomass estimated;bounded
      b1mwnd=b1(lpn,lx)-wnd(lpn,lx)
      if (b1mwnd.le.1.0) b1mwnd=1.0
c total stem and branch biomass estimated;bounded
      bbs=gm2t(lpn,lx)+gm2b(lpn,lx)
      if (bbs.le.1.0) bbs=1.0
c total standing above ground biomass estimated
      b1sd=b1(lpn,lx)+sd(lpn,lx)
c
c ** contaminant loop ******************************************************
      do 420 k=1,inz
c live root internal contaminant content (Z2), and saltation-creep external
c contaminant contents for root bole (RB3), and root branches (RBB3) estimated
      z2(lpn,k,lx)=z2(lpn,k,lx)-rd3(lpn,lx)*z2(lpn,k,lx)/r1(lpn,lx)
      rb3(lpn,k,lx)=rb3(lpn,k,lx)-rd3(lpn,lx)*rb3(lpn,k,lx)/r1(lpn,lx)
      rbb3(lpn,k,lx)=rbb3(lpn,k,lx)-rd3(lpn,lx)*rbb3(lpn,k,lx)/
     1r1(lpn,lx)
c leaf internal contaminant content (WZ3), external resuspension content (A2),
c saltation-creep external content(A3), and dead standing leaf external
c resuspension, and saltation-creep contents (WZ42,WZ43) adjusted for
c mortality losses
      wz3(lpn,k,lx)=wz3(lpn,k,lx)-wz3(lpn,k,lx)*wd3(lpn,lx)/wnd(lpn,lx)
      a2(lpn,k,lx)=a2(lpn,k,lx)-h2*h3*a2(lpn,k,lx)
      a2x(lpn,k,lx)=a2x(lpn,k,lx)-h2*h3*a2x(lpn,k,lx)
      a3(lpn,k,lx)=a3(lpn,k,lx)-h2*h3*a3(lpn,k,lx)
c live stem (B3) saltation-creep, live branches(BB3) saltation-creep
c mortality losses; same for live stem(B2) and branches(BB2) resuspension
c losses
      b3(lpn,k,lx)=b3(lpn,k,lx)-b3(lpn,k,lx)*d3(lpn,lx)*gm2t(lpn,lx)/
     1(b1mwnd*bbs)
      bb3(lpn,k,lx)=bb3(lpn,k,lx)-bb3(lpn,k,lx)*d3(lpn,lx)*gm2b(lpn,lx)/
     1(b1mwnd*bbs)
      b2(lpn,k,lx)=b2(lpn,k,lx)-b2(lpn,k,lx)*d3(lpn,lx)*gm2t(lpn,lx)/
     1(b1mwnd*bbs)
      bb2(lpn,k,lx)=bb2(lpn,k,lx)-bb2(lpn,k,lx)*d3(lpn,lx)*gm2b(lpn,lx)/
     1(b1mwnd*bbs)
c boundary conditions set for saltation-creep and resuspension contents
      if (a2(lpn,k,lx).le.0.0) a2(lpn,k,lx)=0.0
      if (a2x(lpn,k,lx).le.0.0) a2x(lpn,k,lx)=0.0
      if (z42(lpn,k,lx).le.0.0) z42(lpn,k,lx)=0.0
      if (a3(lpn,k,lx).le.0.0) a3(lpn,k,lx)=0.0
      if (b3(lpn,k,lx).le.0.0) b3(lpn,k,lx)=0.0
      if (bb3(lpn,k,lx).le.0.0) bb3(lpn,k,lx)=0.0
      if (b2(lpn,k,lx).le.0.0) b2(lpn,k,lx)=0.0
      if (bb2(lpn,k,lx).le.0.0) bb2(lpn,k,lx)=0.0
      if (z42(lpn,k,lx).le.0.0) z42(lpn,k,lx)=0.0
      if (z43(lpn,k,lx).le.0.0) z43(lpn,k,lx)=0.0
c live stem internal contaminant loss due to mortality
```

```
      z3(lpn,k,lx)=z3(lpn,k,lx)-z3(lpn,k,lx)*d3(lpn,lx)/b1mwnd
c boundary conditions on internal stem and biomass contaminant contemt
      if (z3(lpn,k,lx).le.0.0) z3(lpn,k,lx)=0.0
      if (wz3(lpn,k,lx).le.0.0) wz3(lpn,k,lx)=0.0
c live seed contaminant content due to mortality(drying); boundary condition
      zsd(lpn,k,lx)=zsd(lpn,k,lx)-zsd(lpn,k,lx)*sd3(lpn,lx)/sd(lpn,lx)
      if (zsd(lpn,k,lx).le.0.0) zsd(lpn,k,lx)=0.0
c total plant contaminant content (live and dead standing) from all sources
c and boundary
      zsum=z3(lpn,k,lx)+wz3(lpn,k,lx)+a3(lpn,k,lx)+a2(lpn,k,lx)+afr(lpn
     1,k,lx)+wz43(lpn,k,lx)+zsd(lpn,k,lx)+wz42(lpn,k,lx)+afs(lpn,k,lx)
      if (zsum.le.0.0) zsum=0.0
c selects highest concentration of total contaminant content(Z7MAX), internal
c live leaf(WZ7MAX), maximum root concentration(Z6MAX), fruit maximum
c biomass(SDMX); total live biomass maximum(BSMAX); live root biomass
c maximum(R1MAX), and maximum leaf biomass(WNDMAX) at total live standing
c maximum biomass
      if (b1sd.ge.bsmax(lpn,lx)) z7max(lpn,k,lx)=zsum/b1sd
      if (b1sd.ge.bsmax(lpn,lx)) wz7max(lpn,k,lx)=wz3(lpn,k,lx)/wnd(lpn
     1 ,lx)
      if (b1sd.ge.bsmax(lpn,lx)) z6max(lpn,k,lx)=z2(lpn,k,lx)/r1(lpn,lx)
  420 continue
c ** end contaminant loop ***********************************************
      if (b1sd.ge.sdmx(lpn,lx)) sdmx(lpn,lx)=sd(lpn,lx)
      if (b1sd.ge.bsmax(lpn,lx)) bsmax(lpn,lx)=b1sd
      if (b1sd.ge.bsmax(lpn,lx)) r1max(lpn,lx)=r1(lpn,lx)
      if (wnd(lpn,lx).ge.wndmax(lpn,lx)) wndmax(lpn,lx)=wnd(lpn,lx)
      if (ccvv(lpn,lx).ge.ccmax(lpn,lx)) ccmax(lpn,lx)=ccvv(lpn,lx)
      if (lpn.ne.ipl) return
c niche competition selector
      if (aiche(lx)) 471,470,471
c skips first year of simulation
  471 if (ai1-1.0) 472,470,472
c
c niche competition
c
c  niche competition factor  cfac is calculated based on the net daily
c  productivity and leaf biomass. for those plants set into niche  com-
c  petition  t5p (water transpired) is affected for each plant.
c
c ** plant loop *****************************************************
  472 do 450 k=1,ipl
c initializes niche competiton factor
      cfac(k,lx)=1.0
c excludes plants not in competition
      if (rss(k,lx)) 451,450,451
c calculates competition factor based on ratio of gross photosynthesis to
c live leaf biomass
  451 cfac(k,lx)=b5(k,lx)/(wnd(k,lx))
      cxfac(k,lx)=cfac(k,lx)
c
c ** plant loop *****************************************************
      do 440 l=1,ipl
c selects niche competitors and sums fractions
```

```
      if (rss(l,lx)) 445,440,445
  445 if (k.eq.l) goto 440
      if (rss(l,lx)-rss(k,lx)) 440,441,440
  441 cfac(k,lx)=cfac(k,lx)+b5(l,lx)/(wnd(l,lx))
  440 continue
c ** end plant loop ******************************************************
  450 continue
c ** end plant loop ******************************************************
c
c ** plant loop ********************************************************
      do 460 k=1,ipl
c selects niche competitors and estimates individual fractions;and boundaries
      if (rss(k,lx)) 461,460,461
  461 if (cfac(k,lx).gt.0.0) cfac(k,lx)=cxfac(k,lx)/cfac(k,lx)
      if (wnd(k,lx).le.1.0) cfac(k,lx)=1.0
      if (cfac(k,lx).le.0.1) cfac(k,lx)=0.1
  460 continue
c ** end plant loop ******************************************************
c selects for secondary succession module yearly
  470 if (aiii.lt.365.)return
      if(lpn.lt.ipl)return
c selects for plants under succession
      if (rscde(lx)) 551,550,551
c
c succession module
c
c  succession factor is calculated for each plant set by input into
c  succession by rscde. plants are ranked by the value of their
c  structural biomass ratio . a succession factor is then calculated
c (sfac) for use in the t5p equation .
c
c ** plant loop ******************************************************
  551 do 480 k=1,ipl
c assumes plants under irrigation not under succession
      if (rirrc(k,lx).gt.0.0) go to 480
c initializes plant rank(RANK), estimates the individual plant succession
c factor(SFAC) based on maximum annual live standing above ground biomass
c (BSMAX), end-of-year root biomass(R1), and the bole to crown diameter
c regression coefficient(C6). The latter set to 1.0 if subroutine FORMAN
c not accessed for a given plant; the factor is bounded
      rank(k,lx)=1.0
c mean height for trees,feet
      htavx=htav(k,lx)
c maximum height for grasses,herbs, and shrubs,feet
      if(rgc(k,lx).le.1.0)htavx=bsmax(k,lx)/(304.8)
      if(htavx.le.0.1)htavx=0.1
      htavx=htavx**0.9
      ccvvx=ccmax(k,lx)
      if(ccvvx.le.0.1)ccvvx=0.1
      ccvvx=ccvvx
       sfac(k,lx)=alog(bsmax(k,lx)+r1(k,lx)+0.7183-1.0)*ccvvx*htavx
  480 continue
c ** end plant loop ******************************************************
c
```

```
c ** plant loop ********************************************************
      do 500 l=1,ipl
c selects for successsion based on irrigation and rank; also determines
c whether a specific plant has a niche partner for summation
      if (rirrc(l,lx).gt.0.0) go to 500
      if (rank(l,lx)) 502,501,502
  502 if (rss(l,lx))  503,501,503
  501 go to 500
c
c plant loop ***********************************************************
  503 do 490 k=1,ipl
c selects for ranking or niche competitor of specific plant
      if (rank(k,lx)) 505,504,505
  505 if (rss(k,lx)) 506,504,506
  504 go to 490
  506 if (rss(k,lx)-rss(l,lx)) 490,491,490
  491 if (k.eq.l) go to 490
c sums factor for plants in niche competiton... cool season annual
c grasses..etc; sets ranking of partner to 0.0 for further processing
      if(sfac(k,lx).gt.sfac(l,lx))sfac(l,lx)=sfac(k,lx)
c      sfac(l,lx)=(sfac(l,lx)+sfac(k,lx))
      rank(k,lx)=0.0
  490 continue
c ** end plant loop ****************************************************
  500 continue
c ** end plant loop ****************************************************
c initializes succession factor sum
      sffb=0.0
c
c ** plant loop ********************************************************
      do 510 k=1,ipl
c selects only one member of niche composited plants
      if (rank(k,lx)) 511,510,511
c sums succession factors
  511 sffb=sffb+sfac(k,lx)
  510 continue
c ** end plant loop ****************************************************
c
c ** plant loop ********************************************************
      do 520 k=1,ipl
c estimates succession factor fraction of ranked members
      if (rank(k,lx)) 521,520,521
  521 sfac(k,lx)=sfac(k,lx)/sffb
  520 continue
c ** end plant loop ****************************************************
c
c  estimates a ten year running average productivity to estimate
c  the crown cover fraction; the average is used to dampen the
c  otherwise wide variability seen from extreme climatic conditions
c  in some years. this is especially important for trees and plants
c  which store nutrients and water over long periods.
c
  550 continue
c
```

```
c ** plant loop ************************************************************
      do 570 k=1,ipl
c excludes annual plants
      if (hf(k,lx)-1.0) 561,560,561
c year counter(SYAV) is reset every 10 years; mean annual productivty is
c estimated every 10 years
  561 syav(k,lx)=syav(k,lx)+1.0
      if (syav(k,lx)-10.0) 563,562,563
  562 wyav(k,lx)=tpgnw(k,lx)
  563 if (syav(k,lx)-10.0) 565,564,565
  564 syav(k,lx)=2.0
c within each 10 year interval, a running average net annual productivity
c (TPGNW) is estimated and bounded
  565 wyav(k,lx)=wyav(k,lx)+wurat(k,lx)
      tpgnw(k,lx)=wyav(k,lx)/syav(k,lx)
      if (tpgnw(k,lx).le.0.0) tpgnw(k,lx)=0.0
c estimates fractional crown cover(CCVV) based on TPGNW for plants that do
c not access subroutine FORMAN ; access based on value of RGC
      cccc=(1.-exp(-.00154*tpgnw(k,lx)))
      if (rgc(k,lx)-1.0) 567,566,567
  566 ccvv(k,lx)=cccc
c bounds crown cover fraction(CCVV) to greater value of parameters(CCCC,CCVV)
c when FORMAN subroutine is accessed
  567 if (cccc.gt.ccvv(k,lx))goto569
      goto560
  569 if (rgc(k,lx)-1.0) 568,560,568
  568 ccvv(k,lx)=cccc
  560 continue
  570 continue
c ** end plant loop ******************************************************
      return
      end
c
c
```

14.14. RUNOFF (bnm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine runoff (lpg,lg)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
c set number of soil layers; initialize daily surface runoff,cfd
      iicde=rcode(lpg,lg)
      rnofds(lpg,lg)=0.0
c test for available surface water,mm
      if (prec(lg).le.0.0) return
c irrigated land is assumed to pond
      if (rirrc(lpg,lg).gt.0.0) go to 30
c soil thickness,mm, and normalizing constant initialized
      slcsm=slice(lpg,1,lg)
      awis=0.0
c
c ** soil layer loop ***************************************************
      do 10 l=2,iicde
      slcsm=slcsm+slice(lpg,l,lg)
      awis=awis+exp(-4.16*slcsm/sdpth(lpg,lg))
   10 continue
c ** end soil layer loop ***********************************************
      awis=1.0/awis
c
c acn2=estimated intercept for regression:normal moisture condition,
c CNII;bcn2,bcn3= regression coefficients for same.
c relationship based on crown cover,infiltration rate of soil, and the
c persistence or average life-span of vegetation type.
c ccvv =crown cover, fraction, of plant(lpg)in region(lg)
c
      acn2=96.-16.*ccvv(lpg,lg)
      bcn2=9.0+5.9*ccvv(lpg,lg)
      bcn3=.08+.04*ccvv(lpg,lg)
c average life-span, years, of plant(lpg) in region(lg)
      hfav=hf(lpg,lg)/2.0
      if (hfav.le.15.) hfav=15.
c normal moisture condition
      cn2=acn2-bcn2*akond(lg)-bcn3*hfav
c cn1 =dry antecedent moisture condition,CNI,curve number.
      cn1=-16.91+1.348*cn2-.01379*cn2**2+.0001177*cn2**3
c bdrk =bedrock or exposed rock surface, fraction, in region(lg)
c factor has been excluded because it has been accounted for in
c in WATFLX routine
c     cn1=(cn1*100.*(1.-bdrk(lg))+99.*bdrk(lg)*100.)/100.
c smx= maximum value for retention parameter,sir.
      smx=25.4*((1000./cn1)-10.)
      slcsm=slice(lpg,1,lg)
c initialize water storage parameter,inches
```

```
      sir(lpg,lg)=0.0
c
c ** soil layer loop ***************************************************
      do 20 l=2,iicde
      slcsm=slcsm+slice(lpg,l,lg)
c initialize maximum and actual moisture in soil layer column
      uli=h(lpg,l,lg)
      swi=wa(lpg,l,lg)
      if (l.eq.2) uli=uli+h(lpg,1,lg)
      if (l.eq.2) swi=swi+wa(lpg,1,lg)
c weighting constant used to estimate SIR
      wis=awis*exp(-4.16*slcsm/sdpth(lpg,lg))
      sir(lpg,lg)=sir(lpg,lg)+smx*wis*((uli-swi)/uli)
   20 continue
c ** end soil layer loop ***************************************************
      if (sir(lpg,lg).le.0.0) sir(lpg,lg)=0.0
c limits runoff if available surface moisture less than 20% SIR
      if (prec(lg).le.0.2*sir(lpg,lg)) go to 30
c estimates runoff parameter ratio components
      rnin=(prec(lg)-0.2*sir(lpg,lg))**2
      rnid=prec(lg)+0.8*sir(lpg,lg)
c estimates and limits runoff betweem 0.0 and available surface moisture,mm
      rnofds(lpg,lg)=rnin/rnid
      if (rnofds(lpg,lg).ge.prec(lg)) rnofds(lpg,lg)=prec(lg)
      if (rnofds(lpg,lg).le.0.0) rnofds(lpg,lg)=0.0
c estmates water available for runoff,mm, to bound maximum runoff
c f1 =field capacity of soil layer for plant(lpg)in region(lg)
      avwat=wa(lpg,1,lg)-f1(lpg,1,lg)
      if (avwat.le.0.0) avwat=0.0
      if (rnofds(lpg,lg).gt.avwat) rnofds(lpg,lg)=avwat
   30 rnodx(lpg,lg)=rnofds(lpg,lg)
c yearly runoff accumulator,and adjustment of 1st soil layer for loss,mm
      rnofys(lpg,lg)=rnofys(lpg,lg)+rnofds(lpg,lg)
      wa(lpg,1,lg)=wa(lpg,1,lg)-rnofds(lpg,lg)
c infiltrates excess non-runoff water into soil profile(wsrp) if available
c moisture exceeds saturation
      wsrp(lpg,lg)=wa(lpg,1,lg)-h(lpg,1,lg)
      if (wsrp(lpg,lg).le.0.0) wsrp(lpg,lg)=0.0
c resets 1st soil layer moisture to saturation under conditions above
      if (wsrp(lpg,lg).gt.0.0) wa(lpg,1,lg)=h(lpg,1,lg)
c
c ** soil layer loop exclusive of 1st layer ****************************
      do 40 k=2,iicde
      if (wsrp(lpg,lg)) 41,40,41
   41 if (wsrp(lpg,lg).le.0.0) wsrp(lpg,lg)=0.0
c infiltrates excess moisture,mm, to lower layers
      wa(lpg,k,lg)=wa(lpg,k,lg)+wsrp(lpg,lg)
      wsrp(lpg,lg)=0.0
c limits current soil layer to saturation,mm, and adjust surplus moisture
      if (wa(lpg,k,lg).gt.h(lpg,k,lg)) wsrp(lpg,lg)=wa(lpg,k,lg)-h(lpg,k
     1 ,lg)
      if (wa(lpg,k,lg).gt.h(lpg,k,lg)) wa(lpg,k,lg)=h(lpg,k,lg)
   40 continue
c ** end soil layer loop ***************************************************
```

```
      if (wsrp(lpg,lg).le.0.0) wsrp(lpg,lg)=0.0
c ponds non-runoff, non-infiltrated moisture in 1st soil layer
c assumes ponded water for irrigated plants infiltrates laterally as well
      if(rirrc(lpg,lg).le.0.0)wa(lpg,1,lg)=wa(lpg,1,lg)+wsrp(lpg,lg)
      wsrp(lpg,lg)=0.0
      return
      end
c
```

14.15. UPTAKE (bom.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine uptake (dn,lx,rdidx,bpx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
      common/pi/ a1hum(m,n), a1z(m,n), a1zs(m,n), a2hum(m,n), a2lp(m,n),
     1 a2tt(m,n), a2xlp(m,n), a3lp(m,n), a3tt(m,n), afrlp(m,n), afrtt(m
     2 ,n), afstt(m,n), aslp(m,n), b2lp(m,n), b3lp(m,n), bb2lp(m,n),
     3 bb3lp(m,n), cvtt(n), rb3lp(m,n), rbb3lp(m,n), rgain(m,n), sloss(m
     4 ,n), stem12(i,n), susfc(m,n), upk(n), wasumr(m,n), wasums(m,n),
     5 wz3lp(m,n), wz42lp(m,n), wz43lp(m,n), wz4lp(m,n), wz5lp(m,n),
     6 wz5x(i,m,n), wz5xlp(m,n), z1at(m,n), z1ave(m,n), z1lp1(m,n), z2lp
     7 (m,n), z3lp(m,n), z42lp(m,n), z43lp(m,n), z4g2(i,m,n), z4g2lp(m,n
     8 ), z4g3(i,m,n), z4g3lp(m,n), z4glp(m,n), z4lp(m,n), z4rlpl(m,n),
     9 z4tt(m,n), z4tts(m,n), z4ttsw(m,n), z4ttw(m,n), z5lp(m,n), z5rlpl
     1 (m,n), z5rtt(m,n), z5tt(m,n), z5x(i,m,n), z5xlp(m,n), zp(m,n),
     1 zsdlp(m,n), zup(m,n), dose(nq), evsp(m), egeop(m), excp(m),nppts
c
      common/pj/ awt(m), cdrp(ks,n), dsrt(m,n), hbar(ks,n), ph(i,ks,n),
     1 rock(i,ks,n), sand(i,ks,n), silt(i,ks,n), soilf(i,n), tim(i,n),
     2 w1bar(ks,n), watcn(m,n), wtf(n), yaz(m,n), ysrt(m,n),
     3 yup(m,n), z1dl(i,ks,m,n)
c
      dimension airlos(m,n),tcont(m),soilat(ks,m),vfe(m),vfc(m),
     1aircnc(m,n),sacre(n)
c     uptake subrountine calculates contaminant uptake and external
c     contamination for plants simulated
c
c excludes all but first access for parameter input and initialization
      if (rdidx.gt.0.0) go to 40
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-131313.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','red')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE UPTAKE                    ')
      call wnoust('******************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
```

```
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 131313.                     ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
 5004 print *,'ENTER number of plant organs used in dose measurements'
      print *,'if not calculating doses, enter 0 (enter integer value)'
      if(nu.eq.0)read(*,*,err=5004) nppts
      if(nu.eq.1)read(7,*)nppts
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*) nppts
      if(iu.eq.1)print *,nppts
      if(nppts.eq.0)goto5003
c
      do5060knz=1,inz
c**** excludes 5 nutrients NH4,NO3,PO4(s),PO4(l), and K
      if(knz.gt.1.and.knz.lt.7)goto5060
```

```
c atomic number of contaminant;101 or greater for organic contaminants
      ian=atno(knz)
c**** excludes organic contaminants
      if(ian.ge.101)goto5060
 5005 print *,'for nuclide',knz,'with atomic number',atno(knz)
      print *,'ENTER soft tissue dose factor, MeV-Rem/dis-rad (other),'
      print *,'exposure geometry(0.5 or 1.0), and exposure constant:'
      print *,'3 values required; if not calculating doses, enter three'
      print *,'0.0"s'
      if(nu.eq.0)read(*,*,err=5005) evsp(knz),egeop(knz),excp(knz)
      if(nu.eq.1)read(7,*) evsp(knz),egeop(knz),excp(knz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*) evsp(knz),egeop(knz),excp(knz)
      if(iu.eq.1)print *, evsp(knz),egeop(knz),excp(knz)
 5060 continue
 5003 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN UPTAKE                       ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for UPTAKE?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5003)ans
      if(ans)11,5500,11
   11 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-131313.)5501,5503,5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c
c ** regional loop ******************************************************
      do 30 lz=1,ipx
c
c ** plant loop *******************************************************
      apll=ipl
      do 20 l=1,ipl
c
```

```
c ** contaminant loop ****************************************************
      do 10 k=1,inz
c initializes  internal and external contaminant concentrations, per m2,
c for all plant components to be defined as needed
      a2(l,k,lz)=0.0
      a2x(l,k,lz)=0.0
      wz42(l,k,lz)=0.0
      afs(l,k,lz)=0.0
      afsav(l,k,lz)=0.0
      a3(l,k,lz)=0.0
c bole and branches
      b3(l,k,lz)=0.0
      b2(l,k,lz)=0.0
      bb3(l,k,lz)=0.0
      bb2(l,k,lz)=0.0
c root bole and branches
      rb3(l,k,lz)=0.0
      rbb3(l,k,lz)=0.0
c temporary values of same
      a3c(l,k,lz)=0.0
      afsc(l,k,lz)=0.0
      b3c(l,k,lz)=0.0
      bb3c(l,k,lz)=0.0
      rb3c(l,k,lz)=0.0
      rbb3c(l,k,lz)=0.0
c air concentration parameters, per m3
      a1(l,k,lz)=0.0
      a1g(l,k,lz)=0.0
      aaa(l,k,lz)=0.0
      aag(l,k,lz)=0.0
c internal live and dead standing concentration parameters, per m2
      z3(l,k,lz)=0.0
      wz3(l,k,lz)=0.0
      zsd(l,k,lz)=0.0
      z4(l,k,lz)=0.0
      z4g(l,k,lz)=0.0
c external dead concentration parameters, per m2
      z4g2(l,k,lz)=0.0
      z4g3(l,k,lz)=0.0
      z42(l,k,lz)=0.0
      z43(l,k,lz)=0.0
c humus concentration parameters, per m2, for leaves and bole
      z5(l,k,lz)=0.0
      wz5(l,k,lz)=0.0
      z5x(l,k,lz)=0.0
      wz5x(l,k,lz)=0.0
c specific concentration parameters, per gdwt, for above and below ground
c components
      z6(l,k,lz)=0.0
      z7(l,k,lz)=0.0
      wz7(l,k,lz)=0.0
      z8(l,k,lz)=0.0
      z9(l,k,lz)=0.0
      z9r(l,k,lz)=0.0
```

```
      x8(l,k,lz)=0.0
c**** jy=plant exposure time, years
c**** jp=plant part number (leaves, stems, roots..etc.
      kaa=ka
      if(nppts.eq.0)kaa=1
      nqq=nppts
      if(nppts.eq.0)nqq=1
      do21jy=1,kaa
      do21jp=1,nqq
      rpt(jy,l,jp,k,lz)=0.0
   21 continue
c initializes soil temperature, deg. C, for second soil layer
      stem12(l,lz)=0.0
   10 continue
c ** end contaminant loop ***************************************************
   20 continue
c ** end plant loop *********************************************************
   30 continue
c ** end region loop ********************************************************
      return
c initialize components below on julian day 1 of each year
   40 if (aiii-1.0) 70,71,70
c
c ** plant loop ************************************************************
   71 do 60 jj=1,ipl
c
c ** contaminant loop ******************************************************
      do 50 k=1,inz
c annual average air concentration(per m3) from resuspension(A1AV), from
c saltation-creep(ASAV)
      a1av(jj,k,lx)=0.0
      asav(jj,k,lx)=0.0
c maximum annual bole,leaf,and root contaminant concentrations(per gdwt)
      z7max(jj,k,lx)=0.0
      wz7max(jj,k,lx)=0.0
      z6max(jj,k,lx)=0.0
c**** initializes current year plant organ doses to previous year for
c**** perennial plants
      ll=ai1-1.0
      if(ll.eq.0)goto50
      do51nqq=1,nppts
c**** excludes nutrients from dosimetric estimates
      if(k.gt.1.and.k.lt.7)goto51
c**** sets cummulative dose of annual plants to old seed dose
      if(hf(jj,lx)-1.0) 53,52,53
   52 rpt(ll+1,jj,nqq,k,lx)=rpt(ll,jj,1,k,lx)
      goto51
c**** resets cummulative dose of current perrenial plants to previous yr
   53 rpt(ll+1,jj,nqq,k,lx)=rpt(ll,jj,nqq,k,lx)
   51 continue
   50 continue
c ** end contaminant loop ***************************************************
   60 continue
c ** end plant loop *********************************************************
```

```
c mean crown cover for plants in region(LX) initialized
  70 cvtt(lx)=0.0
c if plants growth not being simulated, then returns to main
     if (pltid(1,lx)) 72,73,72
  73 return
c initializes mean soil temperature for second soil layer,deg.C,(STMTT),soil
c matrix potential,bars,(PHITT), and total acreage for all plants in region
c (SACRE)
  72 stmtt=0.0
     phitt(1)=0.0
     sacre(lx)=0.0
c
c ** contaminant loop ****************************************************
     do 80 k=1,inz
c initializes mean air,leaf,dead standing leaf and bole; and humus contaminant
c contents
c mean air concentration, per m3
     a1tt(k,lx)=0.0
     a1tg(k,lx)=0.0
c mean external leaf concentrations, per m2 initialized
     a2tt(k,lx)=0.0
     a3tt(k,lx)=0.0
     afstt(k,lx)=0.0
     afrtt(k,lx)=0.0
c mean internal and external concentrations,per m2, on dead standing biomass
c above and below ground and humus
     z4tt(k,lx)=0.0
     z4ttsw(k,lx)=0.0
     z5tt(k,lx)=0.0
     z5rtt(k,lx)=0.0
  80 continue
c ** end contaminant loop ************************************************
c
c ** plant loop ******************************************************
     do 90 l=1,ipl
c sums total acreage of plants in region(LX)
     sacre(lx)=sacre(lx)+acres(l,lx)
  90 continue
c ** end plant loop ******************************************************
c
c ** plant loop ******************************************************
     do 130 l=1,ipl
     areaa=acres(l,lx)
c areally weighted mean second soil layer temperature and matrix potential
     stmtt=stmtt+stem12(l,lx)*areaa/sacre(lx)
     phitt(1)=phitt(1)+phi(l,2,lx)*areaa/sacre(lx)
c
c ** contaminant loop ****************************************************
     do 100 k=1,inz
c areally weighted mean air concnetration;bounded
     a1tt(k,lx)=a1tt(k,lx)+a1(l,k,lx)*areaa/sacre(lx)
     a1tg(k,lx)=a1tg(k,lx)+a1g(l,k,lx)*areaa/sacre(lx)
     if (a1tt(k,lx).le.0.0) a1tt(k,lx)=0.0
     if (a1tg(k,lx).le.0.0) a1tg(k,lx)=0.0
```

```
 100 continue
c ** end contaminant loop ***************************************************
c excludes trees greater than 60 cm in height
    if (rgc(l,lx).gt.1.0.and.htav(l,lx)*12.*2.54.ge.60.) go to 110
c estimates areally weighted mean crown cover, fraction of total area
    cvtt(lx)=cvtt(lx)+ccvv(l,lx)*areaa/sacre(lx)
 110 continue
c
c ** contaminant loop *******************************************************
    do 120 k=1,inz
c estimates total resuspension contaminant content,per m2, on above ground
c dead and live plant bole and branches
    bb10=.1*(z42(l,k,lx)+bb2(l,k,lx)+b2(l,k,lx))
    if(bb10.le.0.0)bb10=0.0
c areally weighted mean resuspension contaminant content,per m2, for standing
c above ground biomass(g/m2);bounded
    a2tt(k,lx)=a2tt(k,lx)+(a2(l,k,lx)+afr(l,k,lx)+bb10)*areaa
    1/sacre(lx)
    if (a2tt(k,lx).le.0.0) a2tt(k,lx)=0.0
c same as above for saltation-creep contaminant content,per m2
    bb10=.1*(b3(l,k,lx)+bb3(l,k,lx)+z43(l,k,lx))
    if(bb10.le.0.0)bb10=0.0
c
c areally weighted means for saltation-creep contaminant content,per m2, for
c above ground standing(A3TT),fruit saltation-creep(AFSTT),and fruit
c resuspension(AFRTT)
c
    a3tt(k,lx)=a3tt(k,lx)+(a3(l,k,lx)+afs(l,k,lx)+bb10)*areaa
    1/sacre(lx)
    afstt(k,lx)=afstt(k,lx)+afs(l,k,lx)*areaa/sacre(lx)
    afrtt(k,lx)=afrtt(k,lx)+afr(l,k,lx)*areaa/sacre(lx)
c
c areally weighted mean above ground dead biomass contaminant content,per m2,
c for internal+resuspension, saltation-creep,leaf internal+resuspension,leaf
c saltation-creep(Z4TTSW), and root humus(Z5RTT)
c
    z4tt(k,lx)=z4tt(k,lx)+z4g2(l,k,lx)*areaa/sacre(lx)
    if(z4tt(k,lx).le.0.0)z4tt(k,lx)=0.0
    z4tts(k,lx)=z4tts(k,lx)+z4g3(l,k,lx)*areaa/sacre(lx)
    if(z4tts(k,lx).le.0.0)z4tts(k,lx)=0.0
    z4ttw(k,lx)=z4ttw(k,lx)+wz42(l,k,lx)*areaa/sacre(lx)
    if(z4ttw(k,lx).le.0.0)z4ttw(k,lx)=0.0
    z4ttsw(k,lx)=z4ttsw(k,lx)+wz43(l,k,lx)*areaa/sacre(lx)
    if(z4ttsw(k,lx).le.0.0)z4ttsw(k,lx)=0.0
    z5rtt(k,lx)=z5rtt(k,lx)+z5r(l,1,k,lx)*areaa/sacre(lx)
    if(z5rtt(k,lx).le.0.0)z5rtt(k,lx)=0.0
 120 continue
c ** end contaminant loop ***************************************************
 130 continue
c ** end plant loop *********************************************************
c
c ** plant loop ***************************************************************
    do 310 lp=1,ipl
c
```

```
c ** contaminant loop ******************************************************
      do 140 k=1,inz
c temporary variables
      z2lp(k,lx)=z2(lp,k,lx)
      rb3lp(k,lx)=rb3(lp,k,lx)
      rbb3lp(k,lx)=rbb3(lp,k,lx)
      z3lp(k,lx)=z3(lp,k,lx)
      wz3lp(k,lx)=wz3(lp,k,lx)
      a2lp(k,lx)=a2(lp,k,lx)
      a2xlp(k,lx)=a2x(lp,k,lx)
      afrlp(k,lx)=afr(lp,k,lx)
      a3lp(k,lx)=a3(lp,k,lx)
      zsdlp(k,lx)=zsd(lp,k,lx)
      z4lp(k,lx)=z4(lp,k,lx)
      z4glp(k,lx)=z4g(lp,k,lx)
      z42lp(k,lx)=z42(lp,k,lx)
      z4g2lp(k,lx)=z4g2(lp,k,lx)
      z4g3lp(k,lx)=z4g3(lp,k,lx)
      z43lp(k,lx)=z43(lp,k,lx)
      z5lp(k,lx)=z5(lp,k,lx)
      wz5lp(k,lx)=wz5(lp,k,lx)
      wz5xlp(k,lx)=wz5x(lp,k,lx)
      z5xlp(k,lx)=z5x(lp,k,lx)
      wz4lp(k,lx)=wz4(lp,k,lx)
      wz42lp(k,lx)=wz42(lp,k,lx)
      wz43lp(k,lx)=wz43(lp,k,lx)
      b2lp(k,lx)=b2(lp,k,lx)
      b3lp(k,lx)=b3(lp,k,lx)
      bb2lp(k,lx)=bb2(lp,k,lx)
      bb3lp(k,lx)=bb3(lp,k,lx)
c initialize the mean contaminant content in the effective root zone,per gdwt
      z1ave(k,lx)=0.0
  140 continue
c ** end contaminant loop ****************************************************
c number of soil layers for plant(LP) in region(LX)
      iicde=rcode(lp,lx)
      aicde=iicde
      z1smf=0.0
c
c ** soil layer loop *********************************************************
      do 160 kk=1,iicde
c
c ** contaminant loop ********************************************************
      do 150 k=1,inz
c sum of contaminant contents in effective root zone by layer
      z1ave(k,lx)=z1ave(k,lx)+z1(lp,kk,k,lx)*zf(lp,kk,lx)*zff(lp,kk,lx)
  150 continue
c ** end contaminant loop ****************************************************
c sum of fractions of roots in each soil layer
      z1smf=z1smf+zf(lp,kk,lx)*zff(lp,kk,lx)
  160 continue
c ** end soil layer loop *****************************************************
c bounds Z1SMF to fraction in top soil layer
      if (z1smf.le.zf(lp,1,lx)) z1smf=zf(lp,1,lx)
```

```
c
c ** contaminant loop ************************************************************
      do 170 k=1,inz
c mean contaminant content in effective root zone, per gdwt
      z1ave(k,lx)=z1ave(k,lx)/z1smf
  170 continue
c ** end contaminant loop ********************************************************
c root respiration rate and woody plant mortality rate
      ac=.10*rfp(lp,lx)*exp(rxpp(lp,lx)*tsrf)
      ap=2.0/(365.*hf(lp,lx))
c excludes plants in succession or plants in niche competition
      if (rscde(lx)) 190,171,190
  171 if (aiche(lx)) 190,172,190
c soil temperature(deg. C) and matrix potential(bars) equated the second
c layer values
  172 stmtt=stem12(lp,lx)
      phitt(2)=phi(lp,2,lx)
c mean crown cover equated to individual plant(LP) cover
      cvtt(lx)=ccvv(lp,lx)
c
c ** contaminant loop ************************************************************
      do 180 k=1,inz
c mean dead plant contaminant content, per m2, equated to individual plant
c value
      z4tt(k,lx)=z4g2(lp,k,lx)
      z4tts(k,lx)=z4g3(lp,k,lx)
      z4ttw(k,lx)=wz42(lp,k,lx)
      z4ttsw(k,lx)=wz43(lp,k,lx)
c mean root humus contaminant content equated to humus content of 1st
c soil layer
      z5rtt(k,lx)=z5r(lp,1,k,lx)
c mean resuspendable deposit on plant equated to individual plant including
c leaf, fruit, +1/10 of content in dead standing,live branch, and bole
c surficial deposit(A2TT); same for mean saltation-creep deposit(A3TT);bound
      a2tt(k,lx)=a2(lp,k,lx)+afr(lp,k,lx)+.1*(z42(lp,k,lx)+bb2(lp,k,lx)
     1 +b2(lp,k,lx))
      if (a2tt(k,lx).le.0.0) a2tt(k,lx)=0.0
      a3tt(k,lx)=a3(lp,k,lx)+afs(lp,k,lx)+.1*(b3(lp,k,lx)+bb3(lp,k,lx)
     1 +z43(lp,k,lx))
      if (a3tt(k,lx).le.0.0) a3tt(k,lx)=0.0
c mean fruit saltation-creep and resuspendable deposits equated to individual
c plant values,  per m2
      afstt(k,lx)=afs(lp,k,lx)
      afrtt(k,lx)=afr(lp,k,lx)
  180 continue
c ** end contaminant loop ********************************************************
  190 continue
c bounds mean crown cover
      if (cvtt(lx).ge.1.0) cvtt(lx)=1.0
c mass of top 20 mm of soil, gdwt, assumming bulk density of 1.4
      soilms=1400.*slice(lp,1,lx)
c precipitation and dry NH4-NO3 deposition on top(20mm) soil layer
      z1(lp,1,3,lx)=z1(lp,1,3,lx)+0.5*(.00055+.0006*p(lx))/soilms
      z1(lp,1,2,lx)=z1(lp,1,2,lx)+0.5*(.00055+.0006*p(lx))/soilms
```

```
c biomass, gdwt/m2, of live above biomass less leaf biomass;bound
      b1mwnd=b1(lp,lx)-wnd(lp,lx)
      if (b1mwnd.le.1.0) b1mwnd=1.0
c estimated U.S. mean daily dry deposition velocity, m/day
      dpvel=.0021
c estimated deposition velocity(m/day) as a function of precipitation(P),mm;
c saturates at about 10 mm or about a 0.4 inch rainfall
      if (p(lx).gt.2.0) dpvel=.7*(1.-exp(-.6518*p(lx)))
c estimated deposition velocity for saltation-creep events, m/day
      saldp=0.4
c mass of top 20 mm soil, gdwt, assuming 1.4 bulk density
      soilms=slice(lp,1,lx)*1400.
c wash-off half-lives for resuspended(WASOFR) and saltation-creep(WASOFS)
c deposits, days
      wasofr=14.
      wasofs=8.0
c
c ** contaminant loop ********************************************************
      do 200 k=1,inz
      z1lp1(k,lx)=z1(lp,1,k,lx)
c temporary variable for top layer contaminant concentration, per gdwt
c estimated resuspension factor x 10**8
      susfc(k,lx)=2.5
c estimated resuspension fraction as a function of mean crown cover
      rsusp(lp,k,lx)=susfc(k,lx)*exp(-4.6*cvtt(lx))*1.0e-08
c bounds resuspension to rainfall(P) or snowfall(F7) accum. in mm
      if (p(lx).gt.2.0.or.f7(lx).gt.1.0) rsusp(lp,k,lx)=0.0
c estimated saltation-creep fraction as a function of mean crown cover
      salcrp(lp,k,lx)=susfc(k,lx)*exp(-3.2*cvtt(lx))*1.0e-04
c increases SALCRP x 10 for rainfall events above 5 mm or about 0.2 inches;
c fraction equal to zero for snow accumulationabove 1 mm
      if(p(lx).gt.5..and.t(lx).ge.0.)salcrp(lp,k,lx)=salcrp(lp,k,lx)*10.
      if (f7(lx).gt.1.0) salcrp(lp,k,lx)=0.0
c temporary variable for saltation-creep air concnetration, per m3;bounds
      if (aslp(k,lx).le.0.0) aslp(k,lx)=0.0
      aslp(k,lx)=as(lp,k,lx)
c corrects contaminant concentrations for radioactive decay or DISINT=0.0
c (stable)
      a2(lp,k,lx)=a2(lp,k,lx)-disint(k)*a2(lp,k,lx)
      a2x(lp,k,lx)=a2x(lp,k,lx)-disint(k)*a2x(lp,k,lx)
      z4g2(lp,k,lx)=z4g2(lp,k,lx)-disint(k)*z4g2(lp,k,lx)
      a3(lp,k,lx)=a3(lp,k,lx)-disint(k)*a3(lp,k,lx)
      z4g(lp,k,lx)=z4g(lp,k,lx)-disint(k)*z4g(lp,k,lx)
      z4g3(lp,k,lx)=z4g3(lp,k,lx)-disint(k)*z4g3(lp,k,lx)
      b3(lp,k,lx)=b3(lp,k,lx)-disint(k)*b3(lp,k,lx)
      bb3(lp,k,lx)=bb3(lp,k,lx)-disint(k)*bb3(lp,k,lx)
      b2(lp,k,lx)=b2(lp,k,lx)-disint(k)*b2(lp,k,lx)
      bb2(lp,k,lx)=bb2(lp,k,lx)-disint(k)*bb2(lp,k,lx)
      afr(lp,k,lx)=afr(lp,k,lx)-disint(k)*afr(lp,k,lx)
      afs(lp,k,lx)=afs(lp,k,lx)-disint(k)*afs(lp,k,lx)
c eluviation of contaminants into soil set equal to about 0.2mm/year as with
c humus from top soil layer surface
      sloss(k,lx)=.000025/slice(lp,1,lx)
c cesium and potassium is assumed to enter clay lattice and delayed in upper
```

```
c soil layer by a factor of 10
    if (atno(k)-55.) 202,201,202
 201 sloss(k,lx)=sloss(k,lx)/10.
 202 if (atno(k)-19.) 204,203,204
 203 sloss(k,lx)=sloss(k,lx)/10.
c contaminant loss from top layer due to eluviation of attached particles
 204 z1(lp,1,k,lx)=z1(lp,1,k,lx)-sloss(k,lx)*z1lp1(k,lx)
    if(z1(lp,1,k,lx).le.0.0)z1(lp,1,k,lx)=0.0
c resuspended dead biomass set to 20% RESUSP assuming 1/10 of non-leaf and
c 100 % of dead leaf biomass is available;bounds
    a1hum(k,lx)=0.2*rsusp(lp,k,lx)*(.1*z4tt(k,lx)+z4ttw(k,lx))
    if (a1hum(k,lx).le.0.0) a1hum(k,lx)=0.0
c same as above for saltation-creep events(A2HUM)
    a2hum(k,lx)=0.2*salcrp(lp,k,lx)*(0.1*z4tts(k,lx)+z4ttsw(k,lx))
    if(a2hum(k,lx).le.0.0)a2hum(k,lx)=0.0
c top layer of soil incremented for deposition of resuspended contaminant
c containing material in air(AAA): adjusted for mean crown cover and
c retention constant(0.67), per gdwt
    z1(lp,1,k,lx)=z1(lp,1,k,lx)+dpvel*aaa(lp,k,lx)*(1.-cvtt(lx))*.67
   1 /soilms
    if(z1(lp,1,k,lx).le.0.0)z1(lp,1,k,lx)=0.0
c same as above for contributions from bole-branch and leaf humus
c decomposition
    z1(lp,1,k,lx)=z1(lp,1,k,lx)+(acl*wz5xlp(k,lx)+acll*z5xlp(k,lx))
   1 /soilms
    if(z1(lp,1,k,lx).le.0.0)z1(lp,1,k,lx)=0.0
c increments 2nd soil layer for eluviated contaminant from top layer,per gdwt
    z1(lp,2,k,lx)=z1(lp,2,k,lx)+sloss(k,lx)*soilms*z1lp1(k,lx)/(slice
   1 (lp,2,lx)*1400.)
    if(z1(lp,2,k,lx).le.0.0)z1(lp,2,k,lx)=0.0
c estimates saltation-creep and rainsplash energy  air concentration
c increment of contaminant(K), per m3
    a1zs(k,lx)=salcrp(lp,k,lx)*(z1lp1(k,lx)*soilms)*(1.-cvtt(lx))
    if(a1zs(k,lx).le.0.0)a1zs(k,lx)=0.0
c estimates resuspension  air concentration increment, per m3, from
c resuspended bare soil as a function of mean crown cover(CVTT)
    a1z(k,lx)=(rsusp(lp,k,lx))*(z1lp1(k,lx)*soilms)*(1.-cvtt(lx))
    if(a1z(k,lx).le.0.0)a1z(k,lx)=0.0
c
c estimates total air concentration increment from particles attached to
c leaf surfaces(A2TT),on soil surface, and resuspended humic matter, per m3
c
c estimates total air concentration increment from gas vapor attached to
c soil particles assuming 2 meter mixing height above 1 m2, a mean wind
c velocity of 2.25 m/s, and 10 deg C mean soil profile temperature,
c soil porosity of 0.35, a soil density of 2.65 g/cm3, 2% organic
c matter content of soil, a one day exposure (86,400 seconds)
c
    ian=atno(k)
    vfc(k)=0.0
    vfe(k)=0.0
    if(akoc(ian).le.0.0)goto 1004
    if(henrc(ian).le.0.0)goto 1004
c estimates diffusion coefficient from benzene when not known
```

```
      if(difmol(ian).le.0.0)difmol(ian)=0.08195*(78./awt(k))**0.5
      dei=difmol(ian)*0.35**0.33
      ckds=0.02*akoc(ian)
      ckas=41.*henrc(ian)/ckds
      calph=0.35*dei/(0.35+2.65*0.65*ckds/henrc(ian))
      calp2=3.1416*calph*86400.0
c emission rate ug/m2/sec averaged over 1 day for 1 ug/gm in soil
      vfe(k)=ckas*0.35*2.0*dei*1.0e-02/sqrt(calp2)
      if(vfe(k).le.0.0)vfe(k)=0.0
      vfc(k)=vfe(k)*sqrt(sacre(lx)*4047.0)/(2.0*2.25)
      if(vfc(k).le.0.0)vfc(k)=0.0
c
 1004 a1(lp,k,lx)=(.2*rsusp(lp,k,lx))*a2tt(k,lx)+(a1z(k,lx)+a1hum(k,lx))
      if(a1(lp,k,lx).le.0.0)a1(lp,k,lx)=0.0
c air concentration accumulator incremented for resuspended particles on
c plant surface for specific plant(LP) and for other sources of particles
      aaa(lp,k,lx)=(.2*rsusp(lp,k,lx))*a2xlp(k,lx)+aaa(lp,k,lx)
      if(aaa(lp,k,lx).le.0.0)aaa(lp,k,lx)=0.0
      a3tt(k,lx)=0.0
c saltation-creep-rainsplash air concentration due to all sources, per m3
      as(lp,k,lx)=0.2*salcrp(lp,k,lx)*a3tt(k,lx)+a1zs(k,lx)+a2hum(k,lx)
      if (as(lp,k,lx).le.0.0) as(lp,k,lx)=0.0
c mean annual saltation-creep-rainsplash air concentration, per m3
      asav(lp,k,lx)=asav(lp,k,lx)+as(lp,k,lx)/365.
c resuspension air concentration from all sources, per m3, for contaminant(K)
      a1(lp,k,lx)=a1(lp,k,lx)+aaa(lp,k,lx)
      a1g(lp,k,lx)=aag(lp,k,lx)
      if(a1(lp,k,lx).le.0.0)a1(lp,k,lx)=0.0
      if(a1g(lp,k,lx).le.0.0)a1g(lp,k,lx)=0.0
c mean annual resuspension air concentration, per m3, for contaminant(K)
      a1av(lp,k,lx)=a1av(lp,k,lx)+a1(lp,k,lx)/365.
c resuspension leaf deposit adjusted for losses due to reentrainment
c and washoff(WASOFR); excluding external sources of particles
      a2(lp,k,lx)=a2(lp,k,lx)-0.2*rsusp(lp,k,lx)*a2(lp,k,lx)-.693*a2(lp
     1 ,k,lx)/wasofr
      if(a2(lp,k,lx).le.0.0)a2(lp,k,lx)=0.0
      a2x(lp,k,lx)=a2x(lp,k,lx)-.2*rsusp(lp,k,lx)*a2x(lp,k,lx)-.693*a2x
     1 (lp,k,lx)/wasofr
      if(a2x(lp,k,lx).le.0.0)a2x(lp,k,lx)=0.0
c resuspension deposit on fruits corrected for resuspension and washoff
c losses;bounded
      afr(lp,k,lx)=afr(lp,k,lx)-0.2*rsusp(lp,k,lx)*afr(lp,k,lx)-.693*afr
     1 (lp,k,lx)/wasofr
      if (afr(lp,k,lx).le.0.0) afr(lp,k,lx)=0.0
c saltation-creep-rainsplash deposit on leaves adjusted for reentrainment
c and washoff losses
      a3(lp,k,lx)=a3(lp,k,lx)-0.2*salcrp(lp,k,lx)*a3(lp,k,lx)-.693*a3(lp
     1 ,k,lx)/wasofs
      if(a3(lp,k,lx).le.0.0)a3(lp,k,lx)=0.0
c same as above for dead standing biomass, per m2, for contaminant(K)
      z43(lp,k,lx)=z43(lp,k,lx)-.02*salcrp(lp,k,lx)*z43(lp,k,lx)-.693
     1 *z43(lp,k,lx)/(2.*wasofs)
      if(z43(lp,k,lx).le.0.0)z43(lp,k,lx)=0.0
c same as above for live bole saltation-creep-rainspalsh deposit
```

```
      b3(lp,k,lx)=b3(lp,k,lx)-.02*salcrp(lp,k,lx)*b3(lp,k,lx)-.693*b3(lp
     1 ,k,lx)/(3.*wasofs)
      if(b3(lp,k,lx).le.0.0)b3(lp,k,lx)=0.0
c resuspension deposit on live bole adjusted for reentrainment and washoff
c losses, per m2
      b2(lp,k,lx)=b2(lp,k,lx)-.02*rsusp(lp,k,lx)*b2(lp,k,lx)-.693*b2(lp
     1 ,k,lx)/(3.*wasofr)
      if(b2(lp,k,lx).le.0.0)b2(lp,k,lx)=0.0
c same as above for live standing branch deposits, per m2
      bb3(lp,k,lx)=bb3(lp,k,lx)-.02*salcrp(lp,k,lx)*bb3(lp,k,lx)-.693
     1 *bb3(lp,k,lx)/(3.*wasofs)
      if(bb3(lp,k,lx).le.0.0)bb3(lp,k,lx)=0.0
c same as above, but for resuspension deposit on live standing branches
      bb2(lp,k,lx)=bb2(lp,k,lx)-.02*rsusp(lp,k,lx)*bb2(lp,k,lx)-.693*bb2
     1 (lp,k,lx)/(wasofr*3.)
      if(bb2(lp,k,lx).le.0.0)bb2(lp,k,lx)=0.0
c resuspension deposit on dead standing biomass adjusted for reentrainment
c and washoff losses, per m2
      z42(lp,k,lx)=z42(lp,k,lx)-.02*rsusp(lp,k,lx)*z42(lp,k,lx)-.693*z42
     1 (lp,k,lx)/(wasofr*3.)
      if(z42(lp,k,lx).le.0.0)z42(lp,k,lx)=0.0
c same as above but for saltationn-creep-rainsplash deposit on fruits
      afs(lp,k,lx)=afs(lp,k,lx)-0.2*salcrp(lp,k,lx)*afs(lp,k,lx)-.693
     1 *afs(lp,k,lx)/wasofs
      if(afs(lp,k,lx).le.0.0)afs(lp,k,lx)=0.0
c daily loss rate of resuspension deposit from plant(LP), per m2 assuming
c that deposit on bole,branches, and dead standing biomass deposit is 1/3
c that of leaf deposit;bounded, per m2 per day
      wasumr(k,lx)=.693*(a2(lp,k,lx)+.33*(b2(lp,k,lx)+bb2(lp,k,lx)+z42
     1 (lp,k,lx))+afr(lp,k,lx))/wasofr
      if(wasumr(k,lx).le.0.0)wasumr(k,lx)=0.0
c same as above but for saltation-creep-rainsplash deposit, per m2 per day
      wasums(k,lx)=.693*(a3av(lp,k,lx)+.33*(b3av(lp,k,lx)+bb3av(lp,k,lx)
     1 +z43(lp,k,lx)))/wasofs
      wasums(k,lx)=wasums(k,lx)+0.693*afsav(lp,k,lx)/wasofs
      if (wasums(k,lx).le.0.0) wasums(k,lx)=0.0
c resuspension deposit on leaves incremented for deposition from resuspended
c particles in air for mean air concentration(A1TT), per m3, including the
c the effect leaf area index(TLAI)
      a2(lp,k,lx)=a2(lp,k,lx)+dpvel*a1(lp,k,lx)*tlai(lp,lx)*0.67
      if (a2(lp,k,lx).le.0.0) a2(lp,k,lx)=0.0
c same as above except other sources of air contaminants included other than
c local reentrainment of particles...global fallout..etc.
      a2x(lp,k,lx)=a2x(lp,k,lx)+dpvel*aaa(lp,k,lx)*tlai(lp,lx)*0.67
c deposition of resuspended particles on fallen dead biomass on ground
c surface assuming dead branches+stems occupy 1/10 surface litter; bounded
      z4g2(lp,k,lx)=z4g2(lp,k,lx)+dpvel*a1(lp,k,lx)*cvtt(lx)*.1
      if (z4g2(lp,k,lx).le.0.0) z4g2(lp,k,lx)=0.0
c dead leaf resuspension deposit incremented for deposition; decremented for
c washoff and reentrainment losses; bounded
      wz42(lp,k,lx)=wz42(lp,k,lx)+dpvel*a1(lp,k,lx)*cvtt(lx)*.9
      if(wz42(lp,k,lx).le.0.0)wz42(lp,k,lx)=0.0
      wz42(lp,k,lx)=wz42(lp,k,lx)-.693*wz42(lp,k,lx)/(wasofr*1.)
      if(wz42(lp,k,lx).le.0.0)wz42(lp,k,lx)=0.0
```

```
      wz42(lp,k,lx)=wz42(lp,k,lx)-0.2*rsusp(lp,k,lx)*wz42(lp,k,lx)
      if(wz42(lp,k,lx).le.0.0)wz42(lp,k,lx)=0.0
c resuspension deposit on fruits incremented for deposition from air
c particulates; bounded
      afr(lp,k,lx)=afr(lp,k,lx)+dpvel*a1(lp,k,lx)*cvtt(lx)*0.67
      if (afr(lp,k,lx).le.0.0) afr(lp,k,lx)=0.0
  200 continue
c ** end contaminant loop ***********************************************
c
c plant allometric parameters module
c
c assumes height of plant, cm, is 1/10 of biomass density, g/m2; saltation-
c creep-rainsplash contamination on leaves, boles and branches of plant based
c on cumulative fraction of dust flux accumulation with height above soil
c surface: ft = 76.45*HTAV**(-2.21) with a minimum height of 7 cm; This
c relationship is used to simulate samples for analysis drawn at random
c heights above the canopy
c
c leaf-to-leaf+fruit biomass density ratio
      wndtsd=wnd(lp,lx)/(wnd(lp,lx)+sd(lp,lx))
c plants per m2 initalized
      trm2=1.0
c estimated mean diameter of plant bole(stem), meters, based on biomass
c density
      bmnm=0.003*b1(lp,lx)/100.
c estimated specific gravity of plant stem tissue, dry weight
      c9den=0.2
c estimated plant height in meters(HTAVM),and in cm(HTAVV) based on biomass
c density
      htavm=0.1*b1(lp,lx)/100.
      if(rgc(lp,lx).le.1.0)htav(lp,lx)=htavm/.3048
      htavv=0.1*b1(lp,lx)
c estimated stem diameter, cm; bounded
      bmncm=0.003*b1(lp,lx)
      if (bmncm.le.0.1) bmncm=0.1
c height of plant ,cm, initialized
      ht3=7.0
c plants not entering forest structure subroutine (FORMAN) excluded; plants
c with zero bole diameters, plant heights, and number of trees excluded
      if (rgc(lp,lx).le.1.0) go to 210
      if (bmn(lp,lx).le.0.0) go to 210
      if (htav(lp,lx).le.0.0) go to 210
      if (trees(lp,lx).le.0.0) go to 210
c tree/m2, bole diameter(meters), height(meters), height(cm), and bole
c diameter(cm) estimated from parameters estimated in FORMAN;bounded
      trm2=trees(lp,lx)/4047.
      bmnm=bmn(lp,lx)*.0254
      htavm=htav(lp,lx)*.3048
      htavv=htavm*100.
      bmncm=bmn(lp,lx)*2.54
      if (bmncm.le.0.1) bmncm=0.1
c wood specific activity  estimated from FORMAN
      c9den=c9(lp,lx)
c mean height of plant set at 1/3 that estimated based on biomass density
```

```
      ht3=htavv*0.33
c mean height of deciduous plants set at 2/3 that estimated based on biomass
c density, cm; both evergreen and deciduous plants bounded at 7 cm
      if (astp(lp,lx)) 212,211,212
  211 ht3=htavv*0.66
  212 if (ht3.le.7.) ht3=7.0
c total surface area, m2, of all plant boles combined based on surface area
c of a cone with base(BMNM, meters) and height(HTAVM, meters), m2 per m2
  210 cvtre=3.1416*(bmnm/2.)*trm2*sqrt((bmnm/2.)**2+htavm**2)
c total surface area of bole(ARTRE) that of a cone with base(BMNCM, cm);
c height(HTAVV, cm); units are cm2 per m2.
      artre=3.1416*bmncm*.5*sqrt(htavv**2+(bmncm*.5)**2)
c surface area of all plants combined
      artre=artre*trm2
c total volume of plants, cm3 per m2, for all plants based on the volume a
c cone for each plant stem, cm3, and plants per m2(TRM2)
      vcm3=3.1416*trm2*((bmncm/2.)**2)*htavv/3.
c biomass density of bole(GM2T) and branches(GM2B), gdwt per m2, based on
c plant stem and branch volume(VCM3) estimated above;bounded
      gm2t(lp,lx)=vcm3*c9den
      if (gm2t(lp,lx).le.1.0) gm2t(lp,lx)=1.0
      gm2b(lp,lx)=b1mwnd-gm2t(lp,lx)
      if (gm2b(lp,lx).le.1.0) gm2b(lp,lx)=1.0
c surface area of branches based on assumption that mean branch diameter
c is 1/10 that of mean bole diameter, cm2 per m2; bounded
      cm2b=40.*gm2b(lp,lx)/(bmncm*c9den)
      if (gm2b(lp,lx).le.1.0) cm2b=0.0
c projected branch cover,m2/m2 surface, assuming branches parallel to ground
c surface; deposition flux perpendicular to surface
      cm2pr=cm2b/31416.
c biomass density of branch,root branch,bole,root bole; based on assumption
c that 1/10 stem radius is bark at density 1/2 that of wood
c gdwt of branch bark, g/m2
      gdbbrk=cm2b*c9den*bmncm/30.
c total sum of bole and branch biomass density, g/m2; bounded
      bbs=gm2t(lp,lx)+gm2b(lp,lx)
      if (bbs.le.1.0) bbs=1.0
c estimated total surface area, m2/m2, of dead standing boles
      cvtred=cvtre*(d1(lp,lx)/(d1(lp,lx)+bbs))*(gm2t(lp,lx)/bbs)
c estimated projected area , m2/m2, of dead standing branches perpendicular
c to soil surface
      cm2prd=cm2pr*(d1(lp,lx)/(d1(lp,lx)+bbs))*(gm2b(lp,lx)/bbs)
c temporary variable for contaminant concentration(K) in top soil layer,
c per gdwt
      z1plk=z1(lp,1,k,lx)
      if (z1plk.le.0.0) z1plk=1.0
c increments branch bark biomass for saltation-creep-rainsplash deposited
c on surface;bounded
      gdbbrk=gdbbrk+saldp*aslp(k,lx)*cm2pr*gm2b(lp,lx)/(bbs*z1plk)
      if(gdbbrk.le.1.0)gdbbrk=1.0
c gdbrk is gdwt of tree bark/m2, g/m2, also incremented for saltation-creep-
c rainsplash deposit on surface;bounded
      gdbrk=artre*c9den*bmncm/30.
      gdbrk=gdbrk+saldp*aslp(k,lx)*cvtre*gm2t(lp,lx)/(bbs*z1plk)
```

```
      if (gdbrk.le.1.0) gdbrk=1.0
c root structure modeled analogous to stem;root bole estmated from r1/b1
c root bole length, meters
      rthtm=htavm*(r1(lp,lx)/b1(lp,lx))
c temporary variable for extent of root penetration, mm; bounded to maximum
c soil depth(SDPTH)
      rpnx=rpntp(lp,lx)
      if (rpnx.ge.sdpth(lp,lx)) rpnx=sdpth(lp,lx)
c root bole length limited total root penetration, meters
      if (rthtm.ge.rpnx/1000.) rthtm=rpnx/1000.
c total surface area, m2/m2, for all root boles, m2/m2
      cvrot=3.1416*trm2*(bmnm/2.)*sqrt(rthtm**2+(bmnm/2.)**2)
c total root bole length, cm
      rthtav=rthtm*100.
c total surface area, cm2/m2, of all root boles
      arrot=3.1416*bmncm*.5*sqrt(rthtav**2+(bmncm*.5)**2)
      arrot=arrot*trm2
c volume ,cm3/m2, of all root boles
      vrcm3=3.1416*trm2*((bmncm/2.)**2)*rthtav/3.
c biomass density, gdwt/m2, of root boles ; bounded
      gm2r(lp,lx)=vrcm3*c9den
      if (gm2r(lp,lx).ge.r1(lp,lx)) gm2r(lp,lx)=r1(lp,lx)
      if (gm2r(lp,lx).le.1.0) gm2r(lp,lx)=1.0
c biomass density of root branches, gdwt/m2; bounded
      if(gm2rb(lp,lx).le.1.0)gm2rb(lp,lx)=1.0
      gm2rb(lp,lx)=r1(lp,lx)-gm2r(lp,lx)
c total surface area of root branches,cm2/m2, assuming root branch diameter
c equals 1/10 of root bole diameter; bounded
      cm2rb=40.*gm2rb(lp,lx)/(bmncm*c9den)
      if (gm2rb(lp,lx).le.1.0) cm2rb=0.0
c biomass density of root branch bark, g/m2, (see GDBBRK above)
      gdrrrk=cm2rb*c9den*bmncm/30.
c soil contamination on root bark based on mean potassium concentration in
c root zone(Z1AVE), second layer(Z1), and that 1/10 of root bark biomass is
c soil mass
      gdrrrk=gdrrrk+z1ave(6,lx)*gdrrrk/(10.*z1(lp,2,6,lx))
c biomass density of root bole and soil contamination as above for root
c branches,gdwt-biomass + soil/m2
      gdrbrk=arrot*c9den*bmncm/30.
      gdrbrk=gdrbrk+z1ave(6,lx)*gdrbrk/(10.*z1(lp,2,6,lx))
c number of plant types simulated in region(LX)
      alpp=lp
c total surface area, m2/m2, of combined bole and branches of plant
      cvtreb=cvtre+cm2pr*3.1416
c projected surface area, m2/m2, of combined bole and branches of a plant
c assuming parallel branches and perpendicular boles relative to the soil
c surface
      cvtrep=cvtre+cm2pr
c branch bark, root bole bark, and root branch bark bounded
      if (gdbbrk.le.1.0) gdbbrk=1.0
      if (gdrbrk.le.1.0) gdrbrk=1.0
      if (gdrrrk.le.1.0) gdrrrk=1.0
c integral average deposition fraction, of plant bole(B3AVI) and plant
c branches (BB3AVI) for estimating mean saltation-creep rainsplash
```

```
c contamination from sampling plant at random height above soil surface; 7 cm
c minimum set for bole; 1/3 (evergreens), 2/3 (deciduous) mean height minimum
c for branches
    b3avi=-(76.45/1.21)*(htavv**(-1.21)-7.**(-1.21))/(htavv-7.)
    bb3avi=-(76.45/1.21)*(htavv**(-1.21)-ht3**(-1.21))/(htavv-ht3)
c estimates total bole area, m2/m2
    cvbm=trm2*3.1416*(bmnm/2.)**2
c estimates total leaf-branch area parallel to ground surface;bounded
    cvtt(lx)=cvtt(lx)-cvbm
    if (cvtt(lx).le.0.0) cvtt(lx)=0.0
c mean crown cover of live green leaves
    cvtt(lx)=cvtt(lx)*(wnd(lp,lx)/(wnd(lp,lx)+wd1(lp,lx)))
c mean integral average deposition fraction for saltation-creep-rainsplash
c contamination based on height limits described as for branches above
c (BB3AVI); bounded
    a3avi=-(76.45/1.21)*(htavv**(-1.21)-ht3**(-1.21))/(htavv-ht3)
    if (a3avi.le.0.0) a3avi=0.0
    if (a3avi.ge.1.0) a3avi=1.0
c same as above for fruit contamination, cm
    afsavi=-(76.45/1.21)*(htavv**(-1.21)-ht3**(-1.21))/(htavv-ht3)
c
c ** contaminant loop ******************************************************
    do 230 k=1,inz
    akv=k
c estimates maximum concentration, per m2, on bole(bark) below 7.0 cm height
c for saltation-creep-rainsplash contamination; bounded
    b3(lp,k,lx)=b3(lp,k,lx)+saldp*aslp(k,lx)*(cvtre-cvtred)*gm2t(lp,lx
    1 )/bbs
c estimates resuspension deposition increment on  bole (bark) based on mean
c air concentration (A1TT) and total surface area of bole(CVTRE) assuming
c 67 %  instantaneous retention; bounded
    b2(lp,k,lx)=b2(lp,k,lx)+dpvel*a1(lp,k,lx)*cvtre*0.67
c same for dead standing bole and dead standing fruit, per m2 ;bounded
    z42(lp,k,lx)=z42(lp,k,lx)+dpvel*a1(lp,k,lx)*cvtreb*((d1(lp,lx)+sd1
    1 (lp,lx))/b1mwnd)*0.67
    if (gdbrk.le.1.0) b2(lp,k,lx)=0.0
c estimates maximum saltation-creep-rainsplash concentration increment, per
c m2, branches using projected areas to the soil surface; bounded
    bb3(lp,k,lx)=bb3(lp,k,lx)+saldp*aslp(k,lx)*(cm2pr-cm2prd)*gm2b(lp
    1 ,lx)/bbs
c estimates resuspension concentration increment, per m2, based on total
c surface area of branches;bounded
    bb2(lp,k,lx)=bb2(lp,k,lx)+dpvel*a1(lp,k,lx)*cm2pr*3.1416*0.67
c bounding conditionns for above parameters
    if (gdbbrk.le.1.0) bb3(lp,k,lx)=0.0
    if (gdbrk.le.1.0) b3(lp,k,lx)=0.0
    if (gdbbrk.le.1.0) bb2(lp,k,lx)=0.0
    if (gdbrk+gdbbrk.le.2.0) z42(lp,k,lx)=0.0
c estimates root bole and root branch bark soil contamination, per m2 based
c on 1/15 of soil contaminant effective root zone concentration; bounded
    rb3(lp,k,lx)=z1ave(k,lx)*gdrbrk/15.
    if (gdrbrk.le.1.0) rb3(lp,k,lx)=0.0
    rbb3(lp,k,lx)=z1ave(k,lx)*gdrrrk/15.
    if (gdrrrk.le.1.0) rbb3(lp,k,lx)=0.0
```

```
c estimates integral average of bole(B3AV) and branches(BB3AV) of samples;
c bole samples picked at random between 7 cm to top of tree, per gdwt bark
c based on maximum values; bounded
      b3av(lp,k,lx)=0.0
      bb3av(lp,k,lx)=0.0
      b3av(lp,k,lx)=b3avi*b3(lp,k,lx)
      bb3av(lp,k,lx)=bb3avi*bb3(lp,k,lx)
      if (htavv.le.7.) b3av(lp,k,lx)=b3(lp,k,lx)
      if (htavv.le.7.) bb3av(lp,k,lx)=bb3(lp,k,lx)
c saltation-creep-rainsplash increment, per m2, for live leaves
      a3(lp,k,lx)=a3(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*wndtsd
c ratio of integral average saltation-creep-rainsplash contamination on bole
c and branches combined to maximum value; bounded
      brat=1.0
      if (b3(lp,k,lx)+bb3(lp,k,lx).gt.0.) brat=(b3av(lp,k,lx)+bb3av(lp,k
     1 ,lx))/(b3(lp,k,lx)+bb3(lp,k,lx))
c integral average saltation-creep-rainsplash contamination, per m2, based on
c total dead bole and projected dead branch surface areas
      z43(lp,k,lx)=z43(lp,k,lx)+saldp*aslp(k,lx)*(cvtred+cm2prd)*brat
c dead leaf,bole, lying dead bole, and lying dead branches biomass bounded
      if (wd1(lp,lx).le.1.0) wd1(lp,lx)=1.0
      if (d1(lp,lx).le.1.0) d1(lp,lx)=1.0
      if (d1gb(lp,lx).le.1.0) d1gb(lp,lx)=1.0
      if (d1gbb(lp,lx).le.1.0) d1gbb(lp,lx)=1.0
c sum of dead leaf,lying branches and boles(BRATW), and sum of dead lying
c branches and boles(D1GG); bounded
      bratw=wd1(lp,lx)+d1gb(lp,lx)+d1gbb(lp,lx)
      d1gg=d1gb(lp,lx)+d1gbb(lp,lx)
      if (d1gg.le.1.0) d1gg=1.0
      if (bratw.le.1.0) bratw=1.0
c estimated lying dead biomass contamination increment, per m2, based on
c residual after correction for integral average deposition fraction for
c saltation-creep-rainsplash deposition events
      z4g3(lp,k,lx)=z4g3(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*(1.0-a3avi)
     1 *d1gg/bratw
      if(z4g3(lp,k,lx).le.0.0)z4g3(lp,k,lx)=0.0
c same as above for lying dead leaves, per m2
      wz43(lp,k,lx)=wz43(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*(1.-a3avi)
     1 *wd1(lp,lx)/bratw
      if(wz43(lp,k,lx).le.0.0)wz43(lp,k,lx)=0.0
c estimates integral average saltation-creep-rainsplash contamination for
c leaf deposits; bounded to maximum below 7 cm plant heights
      a3av(lp,k,lx)=a3avi*a3(lp,k,lx)
      if (htavv.le.7.0) a3av(lp,k,lx)=a3(lp,k,lx)
c maximum saltation-creep-rainsplash contamination on fruits, per m2
      afs(lp,k,lx)=afs(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*sd(lp,lx)/(wnd
     1 (lp,lx)+sd(lp,lx))
c integral average fruit contamination for saltation-creep-rainsplash events
c based on maximum value ;bounded
      afsav(lp,k,lx)=afsavi*afs(lp,k,lx)
      if (htavv.le.7.) afsav(lp,k,lx)=afs(lp,k,lx)
      if (afsav(lp,k,lx).le.0.0) afsav(lp,k,lx)=0.0
c estimates maximum concentration on leaves below 7 cm above soil for
c saltation-creep-rainsplash events
```

```
      a3(lp,k,lx)=a3(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*wndtsd
c estimates maximum concentration on fruits(seeds) below 7 cm above soil as
c above leaves
      afs(lp,k,lx)=afs(lp,k,lx)+saldp*aslp(k,lx)*cvtt(lx)*sd(lp,lx)/(wnd
     1 (lp,lx)+sd(lp,lx))
c concentration of contaminant in top soil layer, per gdwt soil, bounded
      if (z1(lp,1,k,lx).le.0.0) z1(lp,1,k,lx)=0.0
c soil mass density parameters, gdwt-soil/m2 on plant components initialized
c and defined below for each contaminant considered(K)
      dirt=0.0
      dirtb=0.0
      dirtbb=0.0
      dirts=0.0
      adrt=0.0
      adrtb=0.0
      adrtbb=0.0
      adrts=0.0
      dz4c=0.0
      dwz4c=0.0
      dz4gc=0.0
      dwzz4t=0.0
      dwz5c=0.0
      dwz5t=0.0
      dz5c=0.0
      dz4rc=0.0
      dz5rc=0.0
      dz4c=0.0
c excludes zero contaminant concentrations
      if (z1(lp,1,k,lx)) 221,220,221
c areal soil density on plant components based on deposition ratio to
c contaminant activity in top 20 mm of soil
c
c total saltation-creep-rainsplash(SCRS) and resuspension(RSP) soil areal
c density for leaves, tree boles, tree branches, and fruits based on maximum
c deposition below 7 cm
  221 dirt=(a3(lp,k,lx)+a2(lp,k,lx))/z1plk
      dirtb=(b3(lp,k,lx)+b2(lp,k,lx))/z1plk
      dirtbb=(bb3(lp,k,lx)+bb2(lp,k,lx))/z1plk
      dirts=(afs(lp,k,lx)+afr(lp,k,lx))/z1plk
c integral average SCRS and RSP soil areal density on leaves, boles,
c branches, and fruits soil areal densities
      adrt=(a3av(lp,k,lx)+a2(lp,k,lx))/z1plk
      adrtb=(b3av(lp,k,lx)+b2(lp,k,lx))/z1plk
      adrtbb=(bb3av(lp,k,lx)+bb2(lp,k,lx))/z1plk
      adrts=(afr(lp,k,lx)+afsav(lp,k,lx))/z1plk
c dead leaves, dead lying bole and branch biomass,and total for all dead
c components of both SCRS and RSP soil areal densities, gdwt-soil/m2
      dwz4c=(wz42(lp,k,lx)+wz43(lp,k,lx))/z1plk
      dz4gc=(z4g2(lp,k,lx)+z4g3(lp,k,lx))/z1plk
      dwzz4t=(wz43(lp,k,lx)+wz42(lp,k,lx)+z4g2(lp,k,lx)+z4g3(lp,k,lx))/
     1z1plk
c leaf humus,leaf+bole+branch humus, and bole+branch humus soil areal
c densities for internal concentrations
      dwz5c=wz5(lp,k,lx)/z1plk
```

```
      dwz5t=(wz5(lp,k,lx)+z5(lp,k,lx))/z1plk
      dz5c=z5(lp,k,lx)/z1plk
c dead root, and root humus equivalent areal soil concentrations, per m2
      dz4rc=z4r(lp,1,k,lx)/z1plk
      dz5rc=z5r(lp,1,k,lx)/z1plk
c dead standing biomass internal equaivalent soil areal concentrations
      dz4c=(z42(lp,k,lx)+z43(lp,k,lx))/z1plk
c boundary conditions set on various parameters estimated in previous
c calculations
  220 if (afs(lp,k,lx).le.0.0) afs(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) afs(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) afr(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) a2(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) a2x(lp,k,lx)=0.0
      if (d1gb(lp,lx)+d1gbb(lp,lx).le.2.0) z4g2(lp,k,lx)=0.0
      if (wd1(lp,lx).le.1.0) wz42(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) a3(lp,k,lx)=0.0
      if (d1gb(lp,lx)+d1gbb(lp,lx).le.2.0) z4g3(lp,k,lx)=0.0
      if (d1gb(lp,lx)+d1gbb(lp,lx).le.1.0) z4g(lp,k,lx)=0.0
      if (z4g(lp,k,lx).le.0.0) z4g(lp,k,lx)=0.0
      if (d1(lp,lx)+sd1(lp,lx).le.2.0) z43(lp,k,lx)=0.0
      if (wd1(lp,lx).le.1.0) wz43(lp,k,lx)=0.0
      if (wd1(lp,lx).le.1.0) wz4(lp,k,lx)=0.0
      if (d1(lp,lx)+sd1(lp,lx).le.2.0) z4(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) a3av(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) b3(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) bb3(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) b2(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) bb2(lp,k,lx)=0.0
      if (d1(lp,lx)+sd1(lp,lx).le.2.0) z42(lp,k,lx)=0.0
      if (r1(lp,lx).le.1.0) rb3(lp,k,lx)=0.0
      if (r1(lp,lx).le.1.0) rbb3(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) b3av(lp,k,lx)=0.0
      if (b1(lp,lx).le.1.0) bb3av(lp,k,lx)=0.0
      if (a3(lp,k,lx).le.0.0) a3(lp,k,lx)=0.0
      if (z4g3(lp,k,lx).le.0.0) z4g3(lp,k,lx)=0.0
      if (z4g3(lp,k,lx)) 224,223,224
  223 z4g3lp(k,lx)=0.0
  224 if (z4g2(lp,k,lx).le.0.0) z4g2(lp,k,lx)=0.0
      if (z4g2(lp,k,lx)) 226,225,226
  225 z4g2lp(k,lx)=0.0
  226 if (z43(lp,k,lx).le.0.0) z43(lp,k,lx)=0.0
      if (wz43(lp,k,lx).le.0.0) wz43(lp,k,lx)=0.0
      if (a3av(lp,k,lx).le.0.0) a3av(lp,k,lx)=0.0
      if (b3(lp,k,lx).le.0.0) b3(lp,k,lx)=0.0
      if (bb3(lp,k,lx).le.0.0) bb3(lp,k,lx)=0.0
      if (b2(lp,k,lx).le.0.0) b2(lp,k,lx)=0.0
      if (bb2(lp,k,lx).le.0.0) bb2(lp,k,lx)=0.0
      if (z42(lp,k,lx).le.0.0) z42(lp,k,lx)=0.0
      if (rb3(lp,k,lx).le.0.0) rb3(lp,k,lx)=0.0
      if (rbb3(lp,k,lx).le.0.0) rbb3(lp,k,lx)=0.0
      if (b3av(lp,k,lx).le.0.0) b3av(lp,k,lx)=0.0
      if (bb3av(lp,k,lx).le.0.0) bb3av(lp,k,lx)=0.0
c
```

```
c contaminant concentrations, per gwdt-plant+soil, of parameters based on total
c mass of plant and of soil on plant surfaces; both internal and external
c contaminant contamination considered; SCRS and RSP contaminaton included
c
c maximum estimate for leaves
      a3c(lp,k,lx)=(a3(lp,k,lx)+a2(lp,k,lx)+wz3(lp,k,lx))/(wnd(lp,lx)
     1 +dirt)
c integral average estimate for leaves
      a3ac(lp,k,lx)=(a3av(lp,k,lx)+a2(lp,k,lx)+wz3(lp,k,lx))/(wnd(lp,lx)
     1 +adrt)
c maximum estimate for fruits (seeds)
      afsc(lp,k,lx)=(afs(lp,k,lx)+afr(lp,k,lx)+zsd(lp,k,lx))/(sd(lp,lx)
     1 +dirts)
c integral average estimate for fruits (seeds)
      afsac(lp,k,lx)=(afsav(lp,k,lx)+afr(lp,k,lx)+zsd(lp,k,lx))/(sd(lp
     1 ,lx)+adrts)
c maximum estimate for plant bole (stems)
      b3c(lp,k,lx)=(b3(lp,k,lx)+b2(lp,k,lx)+z3(lp,k,lx))/(gdbrk+dirtb)
c integral average estimate for plant bole (stems)
      b3ac(lp,k,lx)=(b3av(lp,k,lx)+b2(lp,k,lx)+z3(lp,k,lx))/(gdbrk+adrtb
     1 )
c maximum estimate for plant branches
      bb3c(lp,k,lx)=(bb3(lp,k,lx)+bb2(lp,k,lx)+z3(lp,k,lx))/(gdbbrk
     1 +dirtbb)
c integral average estimate for plant branches
      bb3ac(lp,k,lx)=(bb3av(lp,k,lx)+bb2(lp,k,lx)+z3(lp,k,lx))/(gdbbrk
     1 +adrtbb)
c estimate for root bole (stem)
      rb3c(lp,k,lx)=rb3(lp,k,lx)/gdrbrk
c estimate for root branches
      rbb3c(lp,k,lx)=rbb3(lp,k,lx)/gdrrrk
c estimate for standing dead biomass (stems and branches)
      z4c(lp,k,lx)=(z4(lp,k,lx)+z42(lp,k,lx)+z43(lp,k,lx))/(d1(lp,lx)+1.
     1 0+dz4c)
c estimate for dead leaves
      wz4c(lp,k,lx)=(wz4(lp,k,lx)+wz42(lp,k,lx)+wz43(lp,k,lx))/(wd1(lp
     1 ,lx)+1.+dwz4c)
c estimate for lying dead boles and branches
      z4gc(lp,k,lx)=(z4g(lp,k,lx)+z4g2(lp,k,lx)+z4g3(lp,k,lx))/(d1gb(lp
     1 ,lx)+d1gbb(lp,lx)+1.+dz4gc)
c estimate for total leaf (internal+external),and lying dead boles and
c and branches( internal +external)
      wzz4c(lp,k,lx)=(wz4(lp,k,lx)+wz43(lp,k,lx)+wz42(lp,k,lx)+z4g(lp,k
     1 ,lx)+z4g2(lp,k,lx)+z4g3(lp,k,lx))
      wzz4c(lp,k,lx)=wzz4c(lp,k,lx)/(wd1(lp,lx)+d1gb(lp,lx)+d1gbb(lp,lx)
     1 +1.+dwzz4t)
c estimate for leaf humus
      wz5c(lp,k,lx)=wz5(lp,k,lx)/(rh1(lp,1,lx)+1.0+dwz5c)
c estimate for bole and branch humus
      z5c(lp,k,lx)=z5(lp,k,lx)/(rh2(lp,1,lx)+1.0+dz5c)
c estimate for dead root bole and branches in top soil horizon
      z4rc(lp,1,k,lx)=z4r(lp,1,k,lx)/(rd1(lp,lx)*.1+1.0+dz4rc)
c estimate for root bole and branch humus
      z5rc(lp,1,k,lx)=z5r(lp,1,k,lx)/(rh2(lp,1,lx)+1.0+dz5rc)
```

```
c
      ian=atno(k)
      if(ian.ge.101)goto 230
c**** this section estimates acute/chronic radiation exposure dose to
c**** different plant components daily and integrates over current yr
c
c**** sets exposure parameters for contaminant(k)
      exc=excp(k)
      evps=evsp(k)
      egeo=egeop(k)
c**** 1=seeds, 2=leaves, 3=stems, 4=boles, 5=root stems
c**** 6=root boles,
      exx=exc*evps
c**** maximum doses
      dose(1)=exx*(zsd(lp,k,lx)/sd(lp,lx)+egeo*afsc(lp,k,lx))
      dose(2)=exx*(wz7(lp,k,lx)+egeo*a3c(lp,k,lx))
      dose(3)=exx*(z7(lp,k,lx)+egeo*bb3c(lp,k,lx))
      dose(4)=exx*(z7(lp,k,lx)+egeo*b3c(lp,k,lx))
      dose(5)=exx*rbb3c(lp,k,lx)
      dose(6)=exx*rb3c(lp,k,lx)
c**** integral average doses: seeds, leaves, branches,boles
      dose(7)=exx*(zsd(lp,k,lx)/sd(lp,lx)+egeo*afsac(lp,k,lx))
      dose(8)=exx*(wz7(lp,k,lx)+egeo*a3ac(lp,k,lx))
      dose(9)=exx*(z7(lp,k,lx)+egeo*bb3ac(lp,k,lx))
      dose(10)=exx*(z7(lp,k,lx)+egeo*b3ac(lp,k,lx))
      dose(11)=exx*(zsd(lp,k,lx)/sd(lp,lx))
      ll=ai1
      do231nqq=1,nppts
      if(k.gt.1.and.k.lt.7)goto231
      rpt(ll,lp,nqq,k,lx)=rpt(ll,lp,nqq,k,lx)+dose(nqq)
  231 continue
  230 continue
c** end contaminant loop ****************************************************
c total rooting soil depth initialized
      rtdep=0.0
c
c ** soil layer loop ********************************************************
      do 290 l=1,iicde
c mass of soil layer(l) estimated, gdwt
      soilms=1400.*slice(lp,l,lx)
c
c ** contaminant loop *******************************************************
      do 240 k=1,inz
      if(l.eq.1)tcont(k)=0.0
c total mass, activity..etc of contaminant in soil profile, grams,pci..
c or ugrams if organic compound
      tcont(k)=tcont(k)+z1(lp,l,k,lx)*soilms
c same for individual layer
      soilat(l,k)=z1(lp,l,k,lx)*soilms
c temporary variables for soil concentration(Z1AT), per gdwt-soil, dead root
c bole and branches(Z4RLPL), and root humus(Z5RLPL) concentrations, per m2
      z1at(k,lx)=z1(lp,l,k,lx)
      z4rlpl(k,lx)=z4r(lp,l,k,lx)
      z5rlpl(k,lx)=z5r(lp,l,k,lx)
```

```
      z5rlpl(k,lx)=z5r(lp,l,k,lx)
  240 continue
c ** end contaminant loop ***************************************************
c temporary variables for ammonia(Z1NH4), nitrate(Z1NO3), soluble phosphate
c (Z1PHS), and labile phosphate(Z1PHL) in soil by layer, per gdwt-soil
      z1nh4=z1(lp,l,3,lx)
      z1no3=z1(lp,l,2,lx)
      z1phs=z1(lp,l,4,lx)
      z1phl=z1(lp,l,5,lx)
c rooting depth, mm, incremented by layer; root penetration fraction by layer
c estimated(ROTFR); assumed equal to maximum if plant is an annual
      rtdep=rtdep+slice(lp,l,lx)
      rotfr=zff(lp,l,lx)*zf(lp,l,lx)
      if (hf(lp,lx)-1.0) 242,241,242
  241 rotfr=zf(lp,l,lx)
c temperature, deg. C., of each soil layer estimated
  242 templ=(t(lx)-12.78)*exp(-(.693/396.)*(rtdep-.5*slice(lp,l,lx)))
      templ=templ+12.78
c temperature of second layer stored in array form
      if (l.eq.2) stem12(l,lx)=templ
c effect of soil temperature on nutrient uptake;bounded
      tmpef=.19+.045*templ
      if (templ.le.10.) tmpef=.41+.0175*templ
      if (templ.le.2.) tmpef=.25*templ
      if (templ.gt.22.) tmpef=1.0
c effect of soil matrix potential on nutrient uptake; bounded
      wteff=.4-.0133*phi(lp,l,lx)
      if (phi(lp,l,lx).le.15.) wteff=1.-.0533*phi(lp,l,lx)
      if (phi(lp,l,lx).gt.30.) wteff=0.0
c dead root biomass to root humus conversion coefficients based on soil
c temperature and available soil moisture; bounded
      h2l=.004+.0032*templ
      h3l=1.+(1./(f1(lp,l,lx)-w1(lp,l,lx)))*(wa(lp,l,lx)-w1(lp,l,lx))
      if (wa(lp,l,lx).lt.w1(lp,l,lx)) h3l=1.0
c humus biomass decomposition conversion coefficient based on soil
c temperature (rapidly decomposing fraction)
      acl=0.1*rfp(lp,lx)*exp(rxpp(lp,lx)*templ)
c fraction total dead root biomass present in soil layer(l) estimated for
c both rapid(RDL) and slowly decomposing(RDL2) components
      rdl=rd1(lp,lx)*rotfr
      rdl2=rd2(lp,lx)*rotfr
c estimates conversion of rapidly(RD1) and slowly decomposing(RD2)
c dead root biomass to humus and decrements current values, g/m2; bounded
      rd1(lp,lx)=rd1(lp,lx)-h2l*h3l*rdl
      if (rd1(lp,lx).le.1.0) rd1(lp,lx)=1.0
      rd2(lp,lx)=rd2(lp,lx)-h2l*h3l*rdl2/10.
      if (rd2(lp,lx).le.1.0) rd2(lp,lx)=1.0
c increments corresponding humus compartments (rapid and slowly decomposing),
c g/m2
      rh1(lp,l,lx)=rh1(lp,l,lx)+h2l*h3l*rdl
      rh2(lp,l,lx)=rh2(lp,l,lx)+h2l*h3l*rdl2/10.
c estimates decomposition rate for slowly decomposing humus
      acll=0.05*acl
c decomposition of rapid and slowly decomposing root humus fractions and
```

```
c adjustment of current values, g/m2; bounded
      rh1(lp,l,lx)=rh1(lp,l,lx)-acl*rh1(lp,l,lx)
      rh2(lp,l,lx)=rh2(lp,l,lx)-acll*rh2(lp,l,lx)
      if (rh2(lp,l,lx).le.1.0) rh2(lp,l,lx)=1.0
      if (rh1(lp,l,lx).le.1.0) rh1(lp,l,lx)=1.0
c
c ** contaminant loop ******************************************************
c estimate of mean weighted contaminant content of air, grams
      do 250 k=1,inz
      if(l.eq.1)airlos(k,lx)=0.0
      if(l.eq.1)aircnc(k,lx)=0.0
c dead root contaminant concentration incremented, per m2, including internal
c and external transport (SCRS and RSP) from conversion of live biomass
      z4r(lp,l,k,lx)=z4r(lp,l,k,lx)+rotfr*(z2lp(k,lx)+rbb3lp(k,lx)+rb3lp
     1 (k,lx))*rd3(lp,lx)/r1(lp,lx)
c above decremented for dead root conversion to root humus, per m2
      z4r(lp,l,k,lx)=z4r(lp,l,k,lx)-h2l*h3l*z4rlpl(k,lx)*.1
c above decremented for persistence or decay; bounded
      z4r(lp,l,k,lx)=z4r(lp,l,k,lx)-disint(k)*z4rlpl(k,lx)
      if (z4r(lp,l,k,lx).le.0.0) z4r(lp,l,k,lx)=0.0
      if (rd1(lp,lx)+rd2(lp,lx).le.1.0) z4r(lp,l,k,lx)=0.0
c top layer incremented for decomposition of above ground leaf humus(WZ5XLP)
c and above ground bole+branch humus(Z5XLP), per gwdt-soil
      if (l.eq.1) z1(lp,l,k,lx)=z1(lp,l,k,lx)+(acl*wz5xlp(k,lx)+acll
     1 *z5xlp(k,lx))/soilms
c above ground stem+branch humus, and leaf humus losses due to decomposition,
c per m2; 2 sets of values for top 20 mm layer
      if (l.eq.1) z5(lp,k,lx)=z5(lp,k,lx)-acll*z5lp(k,lx)
      if (l.eq.1) wz5(lp,k,lx)=wz5(lp,k,lx)-acl*wz5lp(k,lx)
      if (l.eq.1) wz5x(lp,k,lx)=wz5x(lp,k,lx)-acl*wz5xlp(k,lx)
      if (l.eq.1) z5x(lp,k,lx)=z5x(lp,k,lx)-acll*z5xlp(k,lx)
c root humus adjusted for gains from decomposing dead roots, and decremented
c for root humus decomposition and persistence or decay for contaminant, per
c m2; bounded based on amount of root humus biomass(RH1 and RH2)
      z5r(lp,l,k,lx)=z5r(lp,l,k,lx)+h2l*h3l*z4rlpl(k,lx)*.1-acl*z5rlpl(k
     1 ,lx)
      if(z5r(lp,l,k,lx).le.0.0)z5r(lp,l,k,lx)=0.0
      z5r(lp,l,k,lx)=z5r(lp,l,k,lx)-disint(k)*z5r(lp,l,k,lx)
      if(z5r(lp,l,k,lx).le.0.0)z5r(lp,l,k,lx)=0.0
      if (rh1(lp,l,lx)+rh2(lp,l,lx).le.2.0) z5r(lp,l,k,lx)=0.0
c decomposing root humus accumulator, per m2
      z9r(lp,k,lx)=z9r(lp,k,lx)+acl*z5rlpl(k,lx)*rotfr
c root humus by layer(l) incremented for gains from decomposing dead roots,
c and losses from decomposition
      z5r(lp,l,k,lx)=z5r(lp,l,k,lx)+h2l*h3l*z4rlpl(k,lx)*rotfr*.1-acl
     1 *z5rlpl(k,lx)
      if(z5r(lp,l,k,lx).le.0.0)z5r(lp,l,k,lx)=0.0
c total soil contaminant dispersed from each soil layer in diffusion
c column daily above 1 m2, ug
c
      airlos(k,lx)=airlos(k,lx)+vfe(k)*z1(lp,l,k,lx)*slice(lp,l,lx)
     1*acres(lp,lx)*86400.0/(sdpth(lp,lx)*sacre(lx))
      aircnc(k,lx)=aircnc(k,lx)+vfc(k)*z1(lp,l,k,lx)*slice(lp,l,lx)
     1*acres(lp,lx)/(sdpth(lp,lx)*sacre(lx))
```

```
c
c above parameters bounded
    if (r1(lp,lx).le.1.0) z2(lp,k,lx)=0.0
    if (z5(lp,k,lx).le.0.0) z5(lp,k,lx)=0.0
    if (z5x(lp,k,lx).le.0.0) z5x(lp,k,lx)=0.0
    if (wz5(lp,k,lx).le.0.0) wz5(lp,k,lx)=0.0
    if (wz5x(lp,k,lx).le.0.0) wz5x(lp,k,lx)=0.0
    if (z5r(lp,l,k,lx).le.0.0) z5r(lp,l,k,lx)=0.0
    if(z9r(lp,k,lx).le.0.0)z9r(lp,k,lx)=0.0
    if(z9(lp,k,lx).le.0.0)z9(lp,k,lx)=0.0
 250 continue
c ** end contaminant loop ****************************************************
c mass of 1 m3 of soil in layer (l)
    slden=soilms*1000./slice(lp,l,lx)
c
c nitrate-phosphate uptake dynamics
c
c phosphate exchange fraction from labile to soluble form in soil based on
c clay fraction(CLAY) in soil layer(l)
c
    pexch=15.+185.*(1.-exp(-(.693/.3)*clay(lp,l,lx)))
    pexch=1./pexch
c estimates soluble phosphate increment to soil, g/gdwt-soil, based on
c exchange fraction and water volumetric ratio in soil(WFR)
    psolu=z1(lp,l,5,lx)*pexch*wfr(lp,l,lx)
    if(psolu.le.0.0)psolu=0.0
c soluble phosphate set to increment
    z1(lp,l,4,lx)=psolu
c conversion of NH4 to NO3 forward reaction coefficients initialized and
c estimated as a function of concentration and density g/m3-soil(SLDEN)
    sfr=0.0
    sfrp=0.0
    if (z1nh4.gt.0.) sfrp=.025+.437*exp(-.00575*z1nh4*slden)
    if (z1nh4.gt.0.) sfr=.025+.437*exp(-.00575*z1nh4*slden)
c conversion of NO3 to NH4 back reaction coefficients initialized and
c estimated as above
    sbr=0.0
    sbrp=0.0
    if (z1no3.gt.0.) sbrp=.0631/((z1no3*slden)**0.4)
    if (z1no3.gt.0.) sbr=.0631/((z1no3*slden)**0.4)
c net conversion of NH4 to NO3 estimated, g/gdwt-soil
    sbf=sfr*z1nh4-sbr*z1no3
c density of NO3 and NH4, g/m3-soil estimated; bounded
    denno3=z1no3*1.4e06
    dennh4=z1nh4*1.4e06
    if (denno3.le..0001) denno3=.0001
    if (dennh4.le..0001) dennh4=.0001
c ratio of labile phosphate in 1 m3-soil to volumetric moisture ratio
c initialized, estimated; bounded
    conphl=0.0
    if (wfr(lp,l,lx).gt.0.0) conphl=z1phl*1.4e06/wfr(lp,l,lx)
    if (conphl.le..00001) conphl=.00001
c evapotranspiration to potential evaporation ratio initialized;estimated
    wneff=0.0
```

```
      if (ea(lp,lx).gt.0.0) wneff=eaval(lp,l,lx)/ea(lp,lx)
c
c initial NO3 (UPNI) and NH4 (UPNH) daily uptake, grams Nitrogen/g-root/m2
      upni=.002/(1.+84./denno3)+.0004/(1.+4.8/denno3)
      upnh=.0002/(1.+84./dennh4)+.00004/(1.+4.8/dennh4)
c NO3 and NH4 parameters used to prevent high nitrogen concentrations in
c roots if rates exceed rate of translocation to above ground biomass
      fni1=35.*exp(-.15*denno3)+40.
      fnh1=35.*exp(-.15*dennh4)+40.
c plant number
      alp=lp
c reverse flow fraction of NO3 and NH4 to prevent excess root concentrations;
c estimated; bounded
      z2a=z2(lp,3,lx)*b1(lp,lx)/wnd(lp,lx)
      if(wnd(lp,lx).le.1.0)z2a=0.0
      fnh2=exp((z2a/r1(lp,lx)-.006)*fnh1)-1.
      fni2=exp((z2a/r1(lp,lx)-.006)*fni1)-1.
      if (fni2.lt.0.0) fni2=0.0
      if (fnh2.lt.0.0) fnh2=0.0
      if (fni2.ge.1.) fni2=1.0
      if (fnh2.ge.1.) fnh2=1.0
c reverse flow for NO3 (BPNI) and NH4 (BPNH) estimated, gN/gdwt-root/m2
      bpni=upni*fni2
      bpnh=upnh*fnh2
c net NO3 uptake adjusted for soil temperature effect, effect soil matrix
c potential, and for the effect of evapotranspiration; bounded
      upni=(upni-bpni)*tmpef*wteff*wneff
      if (upni.le.0.0) upni=0.0
      if (r1(lp,lx).le.1.0) upni=0.0
c same for NH4 uptake; bounded
      upnh=(upnh-bpnh)*tmpef*wteff*wneff
      if (upnh.le.0.0) upnh=0.0
      if (r1(lp,lx).le.1.0) upnh=0.0
c root uptake fraction of inorganic labile phosphate
      uppl=(1.635e-04/(1.+1.43/conphl)+3.27e-05/(1.+.084/conphl))
c leakage fraction of phosphate as function of root phosphate concentration;
c bounded
      z2a=z2(lp,4,lx)*b1(lp,lx)/wnd(lp,lx)
      if(wnd(lp,lx).le.1.0)z2a=0.0
      pleak=exp((z2a/r1(lp,lx)-.001)*173.29)-1.
      if (z2a/r1(lp,lx).lt..001) pleak=0.0
c effect of soil moisture potential on phosphate uptake;bounded
      smpef=10.*(wfr(lp,l,lx)-.073)
      if (wfr(lp,l,lx).ge..173) smpef=1.0
      if (wfr(lp,l,lx).lt..073) smpef=0.0
c NO3 uptake increment due to plant phenology initialized
      untr=0.0
c fraction of growth period expended;bounded
      groper=(aiii-sffp(lx))/(effp(lx)-sffp(lx))
      if (aiii.gt.effp(lx)) groper=1.0
      if (aiii.lt.sffp(lx)) groper=1.0
c NO3 uptake estimate estimated, gN/gdwt-root/m2 for growth phenology less
c than or equal to 20% of total frost free period
      if (groper.le.0.2) untr=.024-0.12*groper
```

```
c NO3 content of roots incremented:bounded
      sncn=z6(lp,3,lx)*(b1(lp,lx)/wnd(lp,lx))+untr
      if (sncn.gt..06) sncn=.06
      if (sncn.le..006) sncn=.006
c effect of nitrogen concentration on PO4 uptake;bounded
      efnit=1.21*(1.0-exp((-.693/.014)*sncn))
      if (efnit.le..001) efnit=.001
c labile phosphate uptake adjusted for effects of soil temperature, PO4
c leakage from roots, soil matrix potential, NO3 root concentration, and
c evapotranspiration;bounded
      uppl=uppl*tmpef*(1.-pleak)*smpef*efnit*wneff
      if (uppl.le.0.0) uppl=0.0
      if (r1(lp,lx).le.1.0) uppl=0.0
c soil matrix potential and soil temperature effects estimated for soil
c soil nitrification;bounded
      wteff=exp(-(.693/8.)*phi(lp,l,lx))
      tmpef=.20+.043*templ
      if (templ.le.7.) tmpef=.0114*templ
      if (templ.gt.28.) tmpef=1.0
c net conversion fraction of NH4 to NO3 estimated; bounded, second estimate
      spbf=sfrp*z1nh4-sbrp*z1no3
      if (spbf.lt.2.) spbf=2.0
c first conversion estimate(SBF); bounded
      if (sbf.ge.2..and.sbf.ge.spbf) spbfv=spbf*2.25
      if (sbf.ge.spbfv) sbf=spbfv
c effect of soil temperature on denitrification estimated;bounded
      dtmpef=-.31+.03266*templ
      if (templ.le.10.) dtmpef=.002*templ
      if (templ.gt.40.) dtmpef=1.0
c effect of soli matrix potential on denitrification estimated;bounded
      dwteff=1.25*(1.-phi(lp,l,lx))
      if (phi(lp,l,lx).gt.1.) dwteff=0.0
      if (phi(lp,l,lx).le.0.2) dwteff=1.0
c denitrification  of soil NO3  estimated, gNH4/gdwt-soil;bounded
      dsbf=.0033*sbf*dwteff*dtmpef
      if (dsbf.le.0.0) dsbf=0.0
c nitrification of soil NH4 to NO3 estimated, gNH4/gdwt-soil;bounded
      sbf=sbf*wteff*tmpef
      if (sbf.le.0.0) sbf=0.0
c
c denitrification of soil NO3 to N2, gN/gdwt-soil/day
c
      z1(lp,l,2,lx)=z1(lp,l,2,lx)-dsbf
c nitrification of soil NH4 to NO3, gN/gdwt-soil/day
      z1(lp,l,2,lx)=z1(lp,l,2,lx)+sbf
c
c adjustment of soil NH4 for conversion to NO3 ;bounded
      z1(lp,l,3,lx)=z1(lp,l,3,lx)-sbf
      if(z1(lp,l,3,lx).le.0.0)z1(lp,l,3,lx)=0.0
c nitrogen uptake coefficient; bounding dependent on plant mean life span
c (HF) for plant(LP) in region(LX)
      ufn=0.5
      if (hf(lp,lx).le.50.) ufn=1.0
      z2p=z2(lp,3,lx)
```

```
      z2up=ufn*(upni+upnh)*r1(lp,lx)*wnd(lp,lx)/b1(lp,lx)
      z2(lp,3,lx)=z2(lp,3,lx)+z2up
      if(z2up.le.0.0)goto251
      z2a=z2(lp,3,lx)
      z6p=z2(lp,3,lx)/r1(lp,lx)
c assures nitrogen contents below or equal to 6%
      if(z6p.le.0.06*wnd(lp,lx)/b1(lp,lx))goto251
      z2(lp,3,lx)=0.06*(wnd(lp,lx)/b1(lp,lx))*r1(lp,lx)
      diff=z2(lp,3,lx)-z2p
      if(diff.le.0.0)diff=0.0
      upnh=upnh*diff/z2up
      upni=upni*diff/z2up
  251 continue
      ufp=0.5
      if (hf(lp,lx).le.50.) ufp=1.0
c root labile phosphate uptake estimated based on leaf biomass, gPO4/gdwt/m2
      z2p=z2(lp,4,lx)
      z2up=ufp*uppl*r1(lp,lx)*wnd(lp,lx)/b1(lp,lx)
      z2(lp,4,lx)=z2(lp,4,lx)+z2up
      if(z2up.le.0.0)goto252
      z2a=z2(lp,4,lx)
      z6p=z2(lp,4,lx)/r1(lp,lx)
c assures phosphate contents below or equal to 0.5%
      if(z6p.le.0.005*wnd(lp,lx)/b1(lp,lx))goto252
      z2(lp,4,lx)=0.005*r1(lp,lx)*wnd(lp,lx)/b1(lp,lx)
      diff=z2(lp,4,lx)-z2p
      if(diff.le.0.0)diff=0.0
      uppl=uppl*diff/z2up
  252 continue
c soil NO3 adjusted for root uptake, gN/gdwt-soil
      z1(lp,l,2,lx)=z1(lp,l,2,lx)-ufn*upni*r1(lp,lx)*wnd(lp,lx)/(soilms*
     1b1(lp,lx))
      if(z1(lp,l,2,lx).le.0.0)z1(lp,l,2,lx)=0.0
c soil PO4 adjusted for root uptake, gP/gdwt-soil ; bounded
      z1(lp,l,5,lx)=z1(lp,l,5,lx)-ufp*uppl*r1(lp,lx)*wnd(lp,lx)/(soilms*
     1b1(lp,lx))
      if (z1(lp,l,5,lx).le.0.0) z1(lp,l,5,lx)=0.0
c soil NH4 adjusted for root uptake, gN/gdwt-soil
      z1(lp,l,3,lx)=z1(lp,l,3,lx)-ufn*upnh*r1(lp,lx)*wnd(lp,lx)/(soilms*
     1b1(lp,lx))
      if(z1(lp,l,3,lx).le.0.0)z1(lp,l,3,lx)=0.0
c
c temporary variable based on reciprocal soil mass, grams
      sajf=slice(lp,l,lx)*1400.
      sajfu=0.0
      if(l.gt.1)sajfu=slice(lp,l-1,lx)*1400.
      sajfd=0.0
      if(l.lt.iicde)sajfd=slice(lp,l+1,lx)*1400.
c total loss of moisture from soil layer(L), mm/day
      if(wataku(lp,l,lx).le.0.0)wataku(lp,l,lx)=0.0
      if(watakd(lp,l,lx).le.0.0)watakd(lp,l,lx)=0.0
c eluviation of humus into lower layers at 0.2mm/year; last layer excluded
      if (l.eq.iicde) go to 260
c both rapidly and slowly decomposing humus in soil layer(L) increments
```

```
c deeper layer; and adjusted for loss
    rh1(lp,l+1,lx)=rh1(lp,l+1,lx)+(.00055/slice(lp,l,lx))*rh1(lp,l,lx)
    rh2(lp,l+1,lx)=rh2(lp,l+1,lx)+(.00055/slice(lp,l,lx))*rh2(lp,l,lx)
    rh1(lp,l,lx)=rh1(lp,l,lx)-(.00055/slice(lp,l,lx))*rh1(lp,l,lx)
    rh2(lp,l,lx)=rh2(lp,l,lx)-(.00055/slice(lp,l,lx))*rh2(lp,l,lx)
 260 continue
c
c ** contaminant loop ****************************************************
    do 280 k=1,inz
c contaminants adjusted for losses due to radioactivity
    z1(lp,l,k,lx)=z1(lp,l,k,lx)-disint(k)*z1(lp,l,k,lx)
c mineral uptake coefficient; dependent on plant mean life span
    ufg=0.3
    if (hf(lp,lx).le.50.) ufg=1.0
c contaminant uptake of contaminant(K) by plant roots, per m2, as a function of
c uptake coefficient,transpiration, leaf area index, and soil
c concentration;bounded
    tlain= 0.013*wnd(lp,lx)
    rgain(k,lx)=ufg*zu(lp,l,k,lx)*eaval(lp,l,lx)*tlain*z1(lp,l,k,lx)
    1*1.25
    rgain(k,lx)=rgain(k,lx)*(1.0-1.0/wnd(lp,lx))
    if(rgain(k,lx).le.0.0)rgain(k,lx)=0.0
    if (b1(lp,lx).le.1.0) rgain(k,lx)=0.0
    if (r1(lp,lx).le.1.0) rgain(k,lx)=0.0
c NO3, NH4, PO4-labile, and PO4-soluble excluded
    if (k.ge.2.and.k.le.5) rgain(k,lx)=0.0
c root contaminant concentration corrected for radioactive decay
    z2(lp,k,lx)=z2(lp,k,lx)-disint(k)*z2(lp,k,lx)
c root contaminant concentration incremented for root uptake, per m2
    z2(lp,k,lx)=z2(lp,k,lx)+rgain(k,lx)
c soil contaminant concentration incremented for gains from decomposing root
c humus, and decremented for losses via plant uptake
    z1(lp,l,k,lx)=z1(lp,l,k,lx)+acl*z5rlpl(k,lx)/soilms-rgain(k,lx)
    1 /soilms
c soil contaminant concentration decremented for losses due to water efflux
c from layer(L) as a function of estimated soil water concentration; bounded
    ian=atno(k)
    if(k.eq.2)zkd(ian)=5.0
    if(k.eq.3)zkd(ian)=50.
    if(k.eq.5)zkd(ian)=1500.
    if(k.eq.6)zkd(ian)=5.5
    if(k.eq.4)zkd(ian)=50.
    watu=0.0
    watd=0.0
    difu=z1(lp,l+1,k,lx)-z1(lp,l,k,lx)
    if(difu.le.0.0)difu=0.0
    difd=z1(lp,l-1,k,lx)-z1(lp,l,k,lx)
    if(difd.le.0.0)difd=0.0
    if(z1(lp,l,k,lx).gt.0.0)difu=difu/z1(lp,l,k,lx)
    if(z1(lp,l,k,lx).gt.0.0)difd=difd/z1(lp,l,k,lx)
    watu=wataku(lp,l,lx)*1000.*z1at(k,lx)/zkd(ian)
    watu=watu*difu
    watd=watd*difd
    if(watu.le.0.0)watu=0.0
```

```
      watd=watakd(lp,l,lx)*1000.*z1at(k,lx)/zkd(ian)
      if(watd.le.0.0)watd=0.0
      if(l.gt.1)z1(lp,l-1,k,lx)=z1(lp,l-1,k,lx)+watu/sajfu
      if(l.lt.iicde)z1(lp,l+1,k,lx)=z1(lp,l+1,k,lx)+watd/sajfd
      z1(lp,l,k,lx)=z1(lp,l,k,lx)-(watu+watd)/sajf
      if (z1(lp,l,k,lx).le.0.0) z1(lp,l,k,lx)=0.0
c excludes bottom soil layer
      zup(k,lx)=zu(lp,l,k,lx)
      if (l.eq.iicde) go to 280
c eluviation of humus mineral into lower layers at 0.2mm/year; upper layer
c decremented
      z5r(lp,l,k+1,lx)=z5r(lp,l,k+1,lx)+(.00055/slice(lp,l,lx))*z5r(lp,l
     1 ,k,lx)
      if(z5r(lp,l,k,lx).le.0.0)z5r(lp,l,k,lx)=0.0
      z5r(lp,l,k,lx)=z5r(lp,l,k,lx)-(.00055/slice(lp,l,lx))*z5r(lp,l,k
     1 ,lx)
      if(z5r(lp,l,k,lx).le.0.0)z5r(lp,l,k,lx)=0.0
c sum above ground leaf and bole-branch humus contaminant content, per m2
      z5sm=wz5(lp,k,lx)+z5(lp,k,lx)
c eluviation above ground humus into second layer, per m2; bounded
      if (l.eq.1) z5r(lp,l+1,k,lx)=z5r(lp,l+1,k,lx)+(.00055/slice(lp,l
     1 ,lx))*z5sm
      if (z5r(lp,l,k,lx).le.0.0) z5r(lp,l,k,lx)=0.0
c adjustment above ground humus for eluviation out of top layer, per m2;
c bole-branch(Z5) and leaf(WZ5); bounded
      if (l.eq.1) z5(lp,k,lx)=z5(lp,k,lx)-(.00055/slice(lp,l,lx))*z5(lp
     1 ,k,lx)
      if(z5(lp,k,lx).le.0.0)z5(lp,k,lx)=0.0
      if (l.eq.1) wz5(lp,k,lx)=wz5(lp,k,lx)-(.00055/slice(lp,l,lx))*wz5
     1 (lp,k,lx)
      if(wz5(lp,k,lx).le.0.0)wz5(lp,k,lx)=0.0
      if (l.eq.1) wz5x(lp,k,lx)=wz5x(lp,k,lx)-(.00055/slice(lp,l,lx))
     1 *wz5x(lp,k,lx)
      if(wz5(lp,k,lx).le.0.0)wz5x(lp,k,lx)=0.0
      if (l.eq.1) z5x(lp,k,lx)=z5x(lp,k,lx)-(.00055/slice(lp,l,lx))*z5x
     1 (lp,k,lx)
      if(wz5(lp,k,lx).le.0.0)wz5x(lp,k,lx)=0.0
c
  280 continue
c ** end contaminant loop ***********************************************
  290 continue
c ** end soil layer loop ********************************************
c effect of soil moisture on humus decomposition, fraction; bounded
      h3=1.0+(1.0/(f1(lp,1,lx)-w1(lp,1,lx)))*(wa(lp,1,lx)-w1(lp,1,lx))
      if (h3.le.1.0) h3=1.0
c
c contaminant translocation
c
c temporary variable for leaf biomass(W111), bole-branch biomass(B111), root
c biomass(R111), and fruit(seed) biomass(S111); bounded
      w111=wnd(lp,lx)
      if (w111.le.1.0) w111=1.0
      b111=b1(lp,lx)-wnd(lp,lx)
      if (b111.le.1.0) b111=1.0
```

```
      b111=0.1*b111
      r111=r1(lp,lx)
      if (r111.le.1.0) r111=1.0
      s111=sd(lp,lx)
      if (s111.le.1.0) s111=1.0
c daily net photosynthate to leaf biomass ratio, g-photosynthate/g-leaves;
c bounded
      add=b5(lp,lx)/wnd(lp,lx)
      if (add.le.0.0) add=0.0
      if (add.gt.0.99) add=0.99
      if (b1(lp,lx).le.1.0) add=0.0
c ratio twice live above ground biomass to total live plant biomass;bounded
      cz2o=(b111+w111+s111)/(b111+w111+s111+r111)*2.
c ratio twice live leaf biommass to total above ground biomass;bounded
      cz3o=w111/(b111+w111+s111)*2.
c ratio twice live fruit biomass to total above ground biomass;bounded
      cz3o1=s111/(b111+w111+s111)*2.
c ratio live root biomass to total plant biomass;bounded
      cz3o2=r111/(r111+b111+w111+s111)
c ratio live bole-branch biomass to total above ground biomass;bounded
      wz3o=b111/(w111+b111+s111)
      if (cz2o.ge.0.99) cz2o=0.99
      if (cz3o.ge.0.99) cz3o=0.99
      if (cz3o1.ge.0.99) cz3o1=0.99
      if (cz3o2.ge.0.99) cz3o2=0.99
      if (wz3o.ge.0.50) wz3o=0.50
c sum of live leaf,live fruit, and live root fractions;bounded
      czsum=cz3o+cz3o1+cz3o2
c reciprocal of CZSUM estimated;bounded
      czfac=1.0
      if (czsum.gt.1.0) czfac=1.0/czsum
c weighted ratios for live leaf,live fruit, and live root fractions
      cz3o=cz3o*czfac
      cz3o1=cz3o1*czfac
      cz3o2=cz3o2*czfac
c weighted sum ratio of plant branch and bole biomass; bounded
      bbbs=(100.*gm2b(lp,lx)+10.*gm2t(lp,lx))/(gm2t(lp,lx)+gm2b(lp,lx))
      if (bbbs.le.1.0) bbbs=1.0
c sum of dead leaf,dead lying bole, and dead lying branch biomass, g/m2
      gdsum=wd1(lp,lx)+d1gb(lp,lx)+d1gbb(lp,lx)
c sum of dead standing bole-branch, leaf, and fruit biomass, g/m2;bounded
      drysum=d1(lp,lx)+wd1(lp,lx)+sd1(lp,lx)
      if (drysum.le.1.0) drysum=1.0
c
c ** contaminant loop ******************************************************
      do 300 k=1,inz
c temporary variable for soil contaminant concentration in top layer;bounded
c     zp(k,lx)=1.0
c     if (z1(lp,1,k,lx).gt.0.0) zp(k,lx)=z1(lp,1,k,lx)
c     if(zp(k,lx).le. 0.0)zp(k,lx)=1.0

c contaminant loss from bottom layer as a function of soil concentration and
c distribution coefficient between soil and water phases
      rnofdm(lp,k,lx)=watakd(lp,iicde,lx)*1000.*(z1at(k,lx)/zkd(k))*1.25
```

```
c annual contaminant loss from bottom layer, per/m2/yr
    rnofym(lp,k,lx)=rnofym(lp,k,lx)+rnofdm(lp,k,lx)
c live above internal contaminant concentration adjusted for radioactive decay,
c per m2
    z3(lp,k,lx)=z3(lp,k,lx)-z3(lp,k,lx)*disint(k)
    wz3(lp,k,lx)=wz3(lp,k,lx)-wz3(lp,k,lx)*disint(k)
    zsd(lp,k,lx)=zsd(lp,k,lx)-zsd(lp,k,lx)*disint(k)
c translocation factor initialized; set to 0.5 for NH4 and soluble PO4
    unp=1.0
    if (k.eq.3.or.k.eq.4) unp=0.5
c contaminant transport from root compartment, per m2
    z2(lp,k,lx)=z2(lp,k,lx)-add*z2lp(k,lx)*cz2o*unp
c contaminant translocation to bole-branch from root compartment, per m2
    z3(lp,k,lx)=z3(lp,k,lx)+add*z2lp(k,lx)*cz2o*unp
c contaminant transport from bole-branch compartment(1), per m2
    z3(lp,k,lx)=z3(lp,k,lx)-add*z3lp(k,lx)*cz3o
c contaminant transport from bole-branch compartment(2), per m2
    z3(lp,k,lx)=z3(lp,k,lx)-add*z3lp(k,lx)*cz3o1
c contaminant translocation to leaves from bole-stem compartment(1), per m2
    wz3(lp,k,lx)=wz3(lp,k,lx)+add*z3lp(k,lx)*cz3o
c contaminant transport from leaf compartment, per m2
    wz3(lp,k,lx)=wz3(lp,k,lx)-add*wz3lp(k,lx)*wz3o
c contaminant translocation to fruits from bole-branch compartment(2), per m2
    zsd(lp,k,lx)=zsd(lp,k,lx)+add*z3lp(k,lx)*cz3o1
c contaminant transport from fruit compartment, per m2
    zsd(lp,k,lx)=zsd(lp,k,lx)-add*zsdlp(k,lx)*wz3o
c contaminant translocation to bole-branch from leaf compartment, per m2
    z3(lp,k,lx)=z3(lp,k,lx)+add*wz3lp(k,lx)*wz3o
c contaminant translocation to bole-branch from fruit compartment, per m2
    z3(lp,k,lx)=z3(lp,k,lx)+add*zsdlp(k,lx)*wz3o
c contaminant transport from bole-branch compartment(3), per m2; bounded
    z3(lp,k,lx)=z3(lp,k,lx)-add*z3lp(k,lx)*cz3o2
    if (z3(lp,k,lx).le.0.0) z3(lp,k,lx)=0.0
c contaminant translocation to roots from bole-branch compartment, per m2;
c bounded
    z2(lp,k,lx)=z2(lp,k,lx)+add*z3lp(k,lx)*cz3o2
c contaminant areal concentrations bounded
    if (z2(lp,k,lx).le.0.0) z2(lp,k,lx)=0.0
    if (r1(lp,lx).le.1.0) z2(lp,k,lx)=0.0
    if (z3(lp,k,lx).le.0.0) z3(lp,k,lx)=0.0
    if (b1(lp,lx).le.1.0) z3(lp,k,lx)=0.0
c labile PO4 concentration in bole-branch compartment set to 0.0
    if (k.eq.5) z3(lp,k,lx)=0.0
c leaf and fruit areal concentrations bounded
    if (wz3(lp,k,lx).le.0.0) wz3(lp,k,lx)=0.0
    if (b1(lp,lx).le.1.0) wz3(lp,k,lx)=0.0
    if (zsd(lp,k,lx).le.0.0) zsd(lp,k,lx)=0.0
    if (sd(lp,lx).le.1.0) zsd(lp,k,lx)=0.0
c
c internal and external areal concentrations corrected for radioactive decay
    z4(lp,k,lx)=z4(lp,k,lx)-disint(k)*z4(lp,k,lx)
    z42(lp,k,lx)=z42(lp,k,lx)-disint(k)*z42(lp,k,lx)
    z43(lp,k,lx)=z43(lp,k,lx)-disint(k)*z43(lp,k,lx)
    wz4(lp,k,lx)=wz4(lp,k,lx)-disint(k)*wz4(lp,k,lx)
```

```
      wz42(lp,k,lx)=wz42(lp,k,lx)-disint(k)*wz42(lp,k,lx)
      wz43(lp,k,lx)=wz43(lp,k,lx)-disint(k)*wz43(lp,k,lx)
      z5(lp,k,lx)=z5(lp,k,lx)-disint(k)*z5(lp,k,lx)
      wz5(lp,k,lx)=wz5(lp,k,lx)-disint(k)*wz5(lp,k,lx)
      wz5x(lp,k,lx)=wz5x(lp,k,lx)-disint(k)*wz5x(lp,k,lx)
      z5x(lp,k,lx)=z5x(lp,k,lx)-disint(k)*z5x(lp,k,lx)
c internal areal dead standing bole-branch contaminant concentration
c incremented for mortality losses of live bole-branch, per m2
      z4(lp,k,lx)=z4(lp,k,lx)+z3lp(k,lx)*d3(lp,lx)/b1mwnd
c same for resuspendable surface deposit, per m2
      z42(lp,k,lx)=z42(lp,k,lx)+b2lp(k,lx)*d3(lp,lx)*gm2t(lp,lx)/(b1mwnd
     1 *bbs)
      z42(lp,k,lx)=z42(lp,k,lx)+bb2lp(k,lx)*d3(lp,lx)*gm2b(lp,lx)/
     1 (b1mwnd*bbs)
c dead leaf contaminant areal concentration incremented for mortality
c losses of live leaves, per m2; resuspension and washoff component included
      wz4(lp,k,lx)=wz4(lp,k,lx)+wz3lp(k,lx)*wd3(lp,lx)/wnd(lp,lx)
      wz4(lp,k,lx)=wz4(lp,k,lx)+a2xlp(k,lx)*wd3(lp,lx)/wnd(lp,lx)
      wz4(lp,k,lx)=wz4(lp,k,lx)+.693*a2xlp(k,lx)/wasofr
c dead leaf external resuspendable incremented for losses from live
c component, per m2
      wz42(lp,k,lx)=wz42(lp,k,lx)+a2lp(k,lx)*wd3(lp,lx)/wnd(lp,lx)
      if(wz42(lp,k,lx).le.0.0)wz42(lp,k,lx)=0.0
c same for saltation-creep-rainsplash component, per m2
      wz43(lp,k,lx)=wz43(lp,k,lx)+a3av(lp,k,lx)*wd3(lp,lx)/wnd(lp,lx)
      if(wz43(lp,k,lx).le.0.0)wz43(lp,k,lx)=0.0
c bole-branch external resuspendable incremented for dead fruit production
      z42(lp,k,lx)=z42(lp,k,lx)+afrlp(k,lx)*sd3(lp,lx)/sd(lp,lx)
c same for saltation-creep-rainsplash; including integral average surficial
c concentrations on leaves, bole, and branches, per m2
      z43(lp,k,lx)=z43(lp,k,lx)+afsav(lp,k,lx)*sd3(lp,lx)/sd(lp,lx)
      z43(lp,k,lx)=z43(lp,k,lx)+b3av(lp,k,lx)*d3(lp,lx)*gm2t(lp,lx)/
     1 (b1mwnd*bbs)
      z43(lp,k,lx)=z43(lp,k,lx)+bb3av(lp,k,lx)*d3(lp,lx)*gm2b(lp,lx)/
     1 (b1mwnd*bbs)
c dead bole-branch internal contaminant concentration decremented for losses
c due to decomposition; bounded
      z4(lp,k,lx)=z4(lp,k,lx)-(.693*z4lp(k,lx)/(hf(lp,lx)*365.))*bbbs
      if (z4(lp,k,lx).le.0.0) z4(lp,k,lx)=0.0
c same for resuspension and saltation-creep-rainsplash decomposition losses;
c bounded
      z42(lp,k,lx)=z42(lp,k,lx)-(.693*z42lp(k,lx)/(hf(lp,lx)*365.))*bbbs
      if (z42(lp,k,lx).le.0.0) z42(lp,k,lx)=0.0
      z43(lp,k,lx)=z43(lp,k,lx)-(.693*z43lp(k,lx)/(hf(lp,lx)*365.))*bbbs
      if (z43(lp,k,lx).le.0.0) z43(lp,k,lx)=0.0
c lying internal and external areal concentrations incremented for conversion
c of standing dead biomass to lying dead biomass; bounded
      z4g2(lp,k,lx)=z4g2(lp,k,lx)+(.693*z42lp(k,lx)/(hf(lp,lx)*365.))
     1 *bbbs
      z4g3(lp,k,lx)=z4g3(lp,k,lx)+(.693*z43lp(k,lx)/(hf(lp,lx)*365.))
     1 *bbbs
      z4g(lp,k,lx)=z4g(lp,k,lx)+(.693*z4lp(k,lx)/(hf(lp,lx)*365.))*bbbs-
     1 .1*h2*h3*z4glp(k,lx)
      if (z4g2(lp,k,lx).le.0.0) z4g2(lp,k,lx)=0.0
```

```
      if (z4g3(lp,k,lx).le.0.0) z4g3(lp,k,lx)=0.0
      if (z4g(lp,k,lx).le.0.0) z4g(lp,k,lx)=0.0
c lying external contaminant concentration for resuspension and saltation-
c creep-rainsplash deposits decremented for decomposition losses
      z4g2(lp,k,lx)=z4g2(lp,k,lx)-h2*h3*z4g2lp(k,lx)/10.
      z4g3(lp,k,lx)=z4g3(lp,k,lx)-h2*h3*z4g3lp(k,lx)/10.
      if(z4g2(lp,k,lx).le.0.0)z4g2(lp,k,lx)=0.0
      if(z4g3(lp,k,lx).le.0.0)z4g3(lp,k,lx)=0.0
c lying dead above ground biomass sum bounded, g/m2
      if (gdsum.le.1.0) gdsum=1.0
c dead leaf resuspension deposit incremented for washoff from dead standing
c component(WASUMR), per m2
      wz42(lp,k,lx)=wz42(lp,k,lx)+cvtt(lx)*wasumr(k,lx)*(wd1(lp,lx)
     1 /gdsum)
      if(wz42(lp,k,lx).le.0.0)wz42(lp,k,lx)=0.0
c same for lying dead biomass of bole-branches, per m2
      z4g2(lp,k,lx)=z4g2(lp,k,lx)+cvtt(lx)*wasumr(k,lx)*(d1gb(lp,lx)
     1 +d1gbb(lp,lx))/gdsum
      if(z4g2(lp,k,lx).le.0.0)z4g2(lp,k,lx)=0.0
c lying dead bole-branch contaminant resuspension deposit decremented for
c washoff and resuspension losses
      z4g2(lp,k,lx)=z4g2(lp,k,lx)-.693*z4g2(lp,k,lx)/(2.*wasofr)
      if(z4g2(lp,k,lx).le.0.0)z4g2(lp,k,lx)=0.0
      z4g2(lp,k,lx)=z4g2(lp,k,lx)-.02*rsusp(lp,k,lx)*z4g2(lp,k,lx)
      if(z4g2(lp,k,lx).le.0.0)z4g2(lp,k,lx)=0.0
c dead leaf saltation-creep-rainsplash incremented for washoff from dead
c standing component(WASUMS), per m2
      wz43(lp,k,lx)=wz43(lp,k,lx)+cvtt(lx)*wasums(k,lx)*(wd1(lp,lx)
     1 /gdsum)
      if(wz43(lp,k,lx).le.0.0)wz43(lp,k,lx)=0.0
c same adjusted for losses due to washoff and for saltation-creep-rainsplash
c removals, per m2
      wz43(lp,k,lx)=wz43(lp,k,lx)-.693*wz43(lp,k,lx)/(wasofs*2.)
      if(wz43(lp,k,lx).le.0.0)wz43(lp,k,lx)=0.0
      wz43(lp,k,lx)=wz43(lp,k,lx)-.2*salcrp(lp,k,lx)*wz43(lp,k,lx)
      if(wz43(lp,k,lx).le.0.0)wz43(lp,k,lx)=0.0
c lying dead bole-branch saltation-creep-rainsplash deposit incremented for
c washoff from dead standing components(WASUMS), per m2
      z4g3(lp,k,lx)=z4g3(lp,k,lx)+cvtt(lx)*wasums(k,lx)*(d1gb(lp,lx)
     1 +d1gbb(lp,lx))/gdsum
      if(z4g3(lp,k,lx).le.0.0)z4g3(lp,k,lx)=0.0
c same adjusted for washoff and removal losses due to saltation-creep-
c rainsplash losses
      z4g3(lp,k,lx)=z4g3(lp,k,lx)-.693*z4g3(lp,k,lx)/(wasofs*2.)
      if(z4g3(lp,k,lx).le.0.0)z4g3(lp,k,lx)=0.0
      z4g3(lp,k,lx)=z4g3(lp,k,lx)-.02*salcrp(lp,k,lx)*z4g3(lp,k,lx)
      if(z4g3(lp,k,lx).le.0.0)z4g3(lp,k,lx)=0.0
c dead leaf internal and external contaminant concentrations decremented for
c decomposition losses
      wz4(lp,k,lx)=wz4(lp,k,lx)-h2*h3*wz4lp(k,lx)
      wz42(lp,k,lx)=wz42(lp,k,lx)-h2*h3*wz42lp(k,lx)
      wz43(lp,k,lx)=wz43(lp,k,lx)-h2*h3*wz43lp(k,lx)
c contaminant internal and external concentrations bounded
      if (z4(lp,k,lx).le.0.0) z4(lp,k,lx)=0.0
```

```
      if (z4g(lp,k,lx).le.0.0) z4g(lp,k,lx)=0.0
      if (z42(lp,k,lx).le.0.0) z42(lp,k,lx)=0.0
      if (z43(lp,k,lx).le.0.0) z43(lp,k,lx)=0.0
      if (wz4(lp,k,lx).le.0.0) wz4(lp,k,lx)=0.0
      if (wz42(lp,k,lx).le.0.0) wz42(lp,k,lx)=0.0
      if (wz43(lp,k,lx).le.0.0) wz43(lp,k,lx)=0.0
c cumulative contaminant losses from bole-branch humus decomposition corrected
c for radioactive decay; same for root humus decomposition
      z9(lp,k,lx)=z9(lp,k,lx)-disint(k)*z9(lp,k,lx)+ac*z5lp(k,lx)
      z9r(lp,k,lx)=z9r(lp,k,lx)-disint(k)*z9r(lp,k,lx)
c humus contaminant concentration incremented for decomposition of lying dead
c component
      z5(lp,k,lx)=z5(lp,k,lx)+.1*h2*h3*(z4glp(k,lx)+z4g2lp(k,lx)+z4g3lp
     1 (k,lx))
c same for leaf humus component
      wz5(lp,k,lx)=wz5(lp,k,lx)+h2*h3*(wz4lp(k,lx)+wz42lp(k,lx))+wz43lp
     1 (k,lx)
c same as above except external contamination components excluded
      wz5x(lp,k,lx)=wz5x(lp,k,lx)+h2*h3*wz4lp(k,lx)
      z5x(lp,k,lx)=z5x(lp,k,lx)+.1*h2*h3*z4glp(k,lx)
c humus contaminant concentrations bounded
      if (z5(lp,k,lx).le.0.0) z5(lp,k,lx)=0.0
      if (wz5(lp,k,lx).le.1.0e-20) wz5(lp,k,lx)=0.0
      if (wz5x(lp,k,lx).le.0.0) wz5x(lp,k,lx)=0.0
      if (z5x(lp,k,lx).le.0.0) z5x(lp,k,lx)=0.0
      if (rh1(lp,1,lx).le.1.0) wz5(lp,k,lx)=0.0
      if (rh1(lp,1,lx).le.1.0) wz5x(lp,k,lx)=0.0
      if (rh2(lp,1,lx).le.1.0) z5x(lp,k,lx)=0.0
      if (rh2(lp,1,lx).le.1.0) z5(lp,k,lx)=0.0
c root internal contaminant concentration, per gdwt-root; bounded
      z6(lp,k,lx)=z2(lp,k,lx)/r1(lp,lx)
      if (z6(lp,k,lx).le.0.0) z6(lp,k,lx)=0.0
c root concentration factor (plant/top soil layer)
      x8(lp,k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10) x8(lp,k,lx)=z6(lp,k,lx)/zp(k,lx)
c bole-branch internal contaminant concentration, per gdwt (bole-branch)
      z7(lp,k,lx)=z3(lp,k,lx)/b1mwnd
c leaf internal contaminant concentration, per gdwt-leaf
      wz7(lp,k,lx)=wz3(lp,k,lx)/wnd(lp,lx)
c concentration factor for bole-branch (plant/top soil layer)
      z8(lp,k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10) z8(lp,k,lx)=z7(lp,k,lx)/zp(k,lx)
c same for leaves
      wz8(lp,k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10) wz8(lp,k,lx)=wz7(lp,k,lx)/zp(k,lx)
c external contaminant concentration on dead above biomass and dead roots
c initialized
      z7a(k,lx)=0.0
      z7ar(k,lx)=0.0
c total contaminant content in and on leaves from all sources, per m2
      wz432=wz4(lp,k,lx)+wz42(lp,k,lx)+wz43(lp,k,lx)
c same for bole-branch component, per m2
      z432=z4(lp,k,lx)+z42(lp,k,lx)+z43(lp,k,lx)
c external contaminant concentration on dry standing above ground biomass,
```

```
c per m2
      if (drysum.gt.0.0) z7a(k,lx)=(z432+wz432)/drysum
c same for dead root biomass
      if ((rd1(lp,lx)+rd2(lp,lx)).gt.0.) z7ar(k,lx)=z4r(lp,1,k,lx)/(rd1
     1 (lp,lx)+rd2(lp,lx))
c total external contaminant concentration of live and dead leaves combined,
c per m2; for both resuspension and saltation-creep-rainsplash components,
c per m2
      a22(k,lx)=(a2(lp,k,lx)+wz42(lp,k,lx))/(wnd(lp,lx)+wd1(lp,lx))
      a23(k,lx)=a3(lp,k,lx)/wnd(lp,lx)
c concentration factor for total contaminant in and on live above standing
c biomass (plant/top soil layer)
      z77(k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10)
     1z77(k,lx)=((z3(lp,k,lx)+wz3(lp,k,lx)+a2(lp,k,lx)+a3(lp,k,lx)+wz42
     1(lp,k,lx)+wz43(lp,k,lx))/b1(lp,lx))/zp(k,lx)
c contaminant concentration of same, per gdwt-plant
      z77a(k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10)z77a(k,lx)=z77(k,lx)*zp(k,lx)
c concentration factor live and dead leaves combined(plant/topsoil layer)
c considering total above ground biomass as denominator
      z78(k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10)
     1z78(k,lx)=((a2(lp,k,lx)+wz42(lp,k,lx))/(b1(lp,lx))*zp(k,lx))
c same as above for saltation-creep rainsplash deposit
      z79(k,lx)=0.0
      if(zp(k,lx).ge.1.0e-10)
     1z79(k,lx)=(a3(lp,k,lx)/(b1(lp,lx))*zp(k,lx))
  300 continue
c ** end contaminant loop ***************************************************
      do312l=1,iicde
      soilms=1400.*slice(lp,l,lx)
      do313k=1,inz
c air concentration adjusted for diffusion, contaminant weighted,ug/m3
      if(l.eq.1)a1g(lp,k,lx)=a1g(lp,k,lx)+aircnc(k,lx)
      if(vfe(k).le.0.0)goto313
      if(tcont(k).le.0.0)goto313
c soil concentration adjusted for diffusion losses
      cnloss=airlos(k,lx)*soilat(l,k)*z1(lp,l,k,lx)/tcont(k)
      z1(lp,l,k,lx)=z1(lp,l,k,lx)-cnloss/soilms
      if(z1(lp,l,k,lx).le.0.0)z1(lp,l,k,lx)=0.0
  313 continue
  312 continue
  310 continue
c ** end plant loop *********************************************************
      return
      end
```

14.16. WATFLX (bpm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine watflx (lpn,lx,bpx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  watflx estimates soil layer hydrology for a specific soil layer
c  profile
c
      include 'main.inc'
c
      common/pj/ awt(m), cdrp(ks,n), dsrt(m,n), hbar(ks,n), ph(i,ks,n),
     1 rock(i,ks,n), sand(i,ks,n), silt(i,ks,n), soilf(i,n), tim(i,n),
     2 w1bar(ks,n), watcn(m,n), wtf(n), yaz(m,n), ysrt(m,n),
     3 yup(m,n), z1dl(i,ks,m,n)
      common/ph/ adsuc(n), ahrv(i,n), arig(i,n), pgr(n), rtrans(i,n),
     1 syav(i,n), xfac(i,n)
      dimension rsi(i,n)
c
c excludes all but first access to subroutine for input and initialization
      if (rdidx) 240,241,240
  241 if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-141414.)5702,5050,5702
 5555 nu=0
 5050 call atcol('white','blue')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE WATFLX                    ')
      call wnoust('****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
```

```
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 141414.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c  if user specifies each soil layer thickness(slice),then rcode
c  must be specified as the number of layers.
c  if soil layer thickness is not specified(rcode=0.0),then default
c  is set at 250mm layers.uptke will be used to estimate zu(yup=0.0).
c  z1 is set internally(yaz=0.0),and zf is estimated when rcode=0.0.
c  z1 will be set for all layers by default assuming 250 mm layers.
c  sdpth should be divisible by 250 to calculate the number
c  of soil layers when default is desired. z1,zf,and zu must be read in
c  when yaz,rcode,or yup are not set to zero totally or in part.
c  zu is read in as the concentration factor (amt/g veg)/(amt/g soil)
c  both vegetation and soil concentrations are based on dry weight,
c  and are in ppm or contaminant concentrations units. Units read in
c  ppm are converted to grams/gram plant tissue. A value of -1.0 for
c  RCODE equates soil parameters current plant tp previous plant.
c
c ** region loop ***********************************************************
      do 10 lz=1,ipx
c regional selector
      if (watx(lz).gt.0.0) go to 20
   10 continue
c ** end region loop *******************************************************
      return
c chnnu=number of channels in watershed
c sedgat= year when watershed module becomes operational
c aqugat (0.0,1.0) = gate for aquifer module,bypass(0.0)
   20 print *,'ENTER gates(0.,1.) for simulating the effects'
```

```
      print *,'of soil pH,and clay content (2 entries)'
      if(nu.eq.0)read(*,*,err=20)pheff,clayef
      if(nu.eq.1)read(7,*)pheff,clayef
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=20) ans
      if(ans.gt.0.and.nu.eq.0)goto20
      write(6,*)pheff,clayef
      if(iu.eq.1)print *,pheff,clayef
 5002 print *,'ENTER number of channels in watershed'
      print *,'if watershed not simulated, enter 0.0'
      if(nu.eq.0)read(*,*,err=5002)chnnu
      if(nu.eq.1)read(7,*)chnnu
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*)chnnu
      if(iu.eq.1)print *,chnnu
 5003 print *,'ENTER year number when watershed simulation starts'
      print *,'must enter 0.0 if watershed not simulated'
      if(nu.eq.0)read(*,*,err=5003)sedgat
      if(nu.eq.1)read(7,*)sedgat
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*)sedgat
      if(iu.eq.1)print *,sedgat
 5004 print *,'ENTER unconfined aquifer gate(0.,1.)'
      print *,'must enter 0.0 if watershed not simulated'
      if(nu.eq.0)read(*,*,err=5004)aqugat
      if(nu.eq.1)read(7,*)aqugat
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*)aqugat
      if(iu.eq.1)print *,aqugat
c     bdrk = fraction of region's surface not permeable to water
c     it is assumed that moisture enters soil between impermeable
 5006 print *,'ENTER impermeable frac. of surface for',ipx,'region(s)'
      if(nu.eq.0)read(*,*,err=5006) (bdrk(lz),lz=1,ipx)
      if(nu.eq.1)read(7,*) (bdrk(lz),lz=1,ipx)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*) (bdrk(lz),lz=1,ipx)
      if(iu.eq.1)print *, (bdrk(lz),lz=1,ipx)
c regional and channel erosion model  selection evokes geoflx input
 5032 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,7)
      call wnoust('*****************************************************')
      call wnoust('You may go to get GEOFLX input before returning to')
      call wnoust('WATFLX; Do you wish to RE-ENTER input to WATFLX?  ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnoust('*****************************************************')
```

```
      call wnouce(6,'press any key')
      call inkey(key)
      call wncuxy(1,7)
      read(*,*,err=5032)ans
      call wnclos(1)
      if(ans) 11,5500,11
   11 rewind 6
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-141414.)5501,5505,5501
 5505 backspace 6
      nu=nuu
      goto 5050
 5501 continue
 5500 continue
      call clear_screen@
      nug=nu
      nu=nuu
      if (chnnu) 12,5814,12
   12 call geoflx (lx,bpx,rdidx)
      nu=nug
      call clear_screen@
c
c the user may change mode of input for this subroutine
c
 5814 if(nu.eq.0)goto5815
      rewind 7
 5802 continue
      read(7,*,err=5802,end=5556)achek2
      if(achek2-151515.)5802,5815,5802
 5556 nu=0
 5815 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('*************************************          ')
      call wnoust(' YOU ARE IN SUBROUTINE WATFLX, PART 2          ')
      call wnoust('*************************************          ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
C
c the user may change mode of input for this subroutine
c
 5800 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                          ')
      call wnoust('ENTER yes=1, no=0                          ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
```

```
      call wncuxy(1,8)
      read(*,*,err=5800)ans
      if(ans.le.0.0)goto5801
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5801 call wnclos(1)
      if(nu.eq.1)goto5806
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 151515.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5801)achek2
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',achek2
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5801) ans
      if(ans.gt.0.0)goto5801
 5806 write(6,*)achek2
      if(iu.eq.1.and.nu.eq.1)print *,achek2
c
 5007 print *,'** all contaminant input requires',inz,'values/line**'
      print *,'ENTER atomic numbers of elements simulated:'
      print *,'for contaminants 2-6 use 7.,7.,15.,15.,and 19.'
      print *,'remember contaminants 1,7,8..etc. are contaminants'
      print *,'other than nutrients, NO3,NH3,PO4(l),PO4(s),and K'
c enters atomic number of ionic component of compound or element
      if(nu.eq.0)read(*,*,err=5007) (atno(j),j=1,inz)
      if(nu.eq.1)read(7,*) (atno(j),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,*) (atno(j),j=1,inz)
      if(iu.eq.1)print *, (atno(j),j=1,inz)
c enters atomic weight of cationic component of compound or element
 5008 print *,'ENTER atomic weights of elements simulated'
      print *,'for contaminants 2-6 use 14.,14.,31.,31.,and 39.1'
      if(nu.eq.0)read(*,*,err=5008) (awt(j),j=1,inz)
      if(nu.eq.1)read(7,*) (awt(j),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5008) ans
      if(ans.gt.0.and.nu.eq.0)goto5008
      write(6,*) (awt(j),j=1,inz)
      if(iu.eq.1)print *, (awt(j),j=1,inz)
c enters halflife or persisitence halflife in years , otherwise 0.0
 5009 print *,'ENTER half-life, years, for ',inz,'contaminants'
      print *,'for nutrients(2-6) or stable contaminants and long-lived'
```

```
      print *,'contaminants enter 0.0'
      if(nu.eq.0)read(*,*,err=5009) (hflif(j),j=1,inz)
      if(nu.eq.1)read(7,*) (hflif(j),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5009) ans
      if(ans.gt.0.and.nu.eq.0)goto5009
      write(6,*) (hflif(j),j=1,inz)
      if(iu.eq.1)print *, (hflif(j),j=1,inz)
c ** regional loop ********************************************************
      do 230 lz=1,ipx
      if (watx(lz)) 5010,230,5010
 5010 print *,'ENTER gate(1,,0.) for entering root fractions'
      print *,'in soil horizons for',ipl,'plants in region',lz
      print *,'a value of 0.0 defaults to estimated values'
c root fraction gate(0.0=default,1.0=values entered)
      if(nu.eq.0)read(*,*,err=5010) (yzf(j,lz),j=1,ipl)
      if(nu.eq.1)read(7,*) (yzf(j,lz),j=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5010) ans
      if(ans.gt.0.and.nu.eq.0)goto5010
      write(6,*) (yzf(j,lz),j=1,ipl)
      if(iu.eq.1)print *, (yzf(j,lz),j=1,ipl)
 5011 print *,'ENTER gate(0,1,2,3,4) for entering',inz,' contaminant'
      print *,'and/or organic compound concentrations in soil in'
      print *,'region',lz
c
      print *,'a value of 0.0 defaults to internal values:'
      print *,'ppm converted to g/gdwt soil,or pci/gdwt if'
      print *,'half-life is specified'
c
      print *,'a value of 1.0: ppm converted to g/gdwt soil,'
      print *,'or pci/gdwt if half-life is specified'
c
      print *,'a value of 2.0: units not converted'
c
      print *,'a value of 3.0: same as 2 above, except that all soil'
      print *,'layers will have the same concentration, activity, etc..'
      print *,'as the second layer (first layer is resuspension layer)'
c
      print *,'a value of 4.0 indicates an organic compound in'
      print *,'ug/gdwt soil: units not converted'
c
c soil contaminant concentration gate
      if(nu.eq.0)read(*,*,err=5011) (yaz(j,lz),j=1,inz)
      if(nu.eq.1)read(7,*) (yaz(j,lz),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5011) ans
      if(ans.gt.0.and.nu.eq.0)goto5011
      write(6,*) (yaz(j,lz),j=1,inz)
      if(iu.eq.1)print *, (yaz(j,lz),j=1,inz)
c
 5012 print *,'ENTER year number when each contaminant conc. takes'
      print *,'effect for',inz,'contaminants in region',lz
      print *,'a value 1.0 assumes effect takes place in year 1'
```

```
c year of initial soil contamination
      if(nu.eq.0)read(*,*,err=5012) (ysrt(j,lz),j=1,inz)
      if(nu.eq.1)read(7,*) (ysrt(j,lz),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5012) ans
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*) (ysrt(j,lz),j=1,inz)
      if(iu.eq.1)print *, (ysrt(j,lz),j=1,inz)
c
 5013 print *,'ENTER day number when each contaminant conc. takes'
      print *,'effect for',inz,'contaminants in region',lz
      print *,'a value of 1.0 assumes effect takes place on day 1'
c
c day of initial soil contamination
      if(nu.eq.0)read(*,*,err=5013) (dsrt(j,lz),j=1,inz)
      if(nu.eq.1)read(7,*) (dsrt(j,lz),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5013) ans
      if(ans.gt.0.and.nu.eq.0)goto5013
      write(6,*) (dsrt(j,lz),j=1,inz)
      if(iu.eq.1)print *, (dsrt(j,lz),j=1,inz)
c
 5014 print *,'ENTER H2O contaminant conc. for',inz,'contaminants in'
      print *,'region',lz
      print *,'units in g/l if soil conc. entered in ppm; otherwise in'
      print *,'ug/l; if radioactive in pCi/l if soil ppm converted to'
      print *, ' pCi/g; otherwise as above for stable elements'
c
c contaminant concentration in drinking or irrigation water
      if(nu.eq.0)read(*,*,err=5014) (watcn(j,lz),j=1,inz)
      if(nu.eq.1)read(7,*) (watcn(j,lz),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*) (watcn(j,lz),j=1,inz)
      if(iu.eq.1)print *, (watcn(j,lz),j=1,inz)
c
 5015 print *,'ENTER plant uptake ratio gates: (amt. per gdwt-plant/'
      print *,'amt. per gdwt-soil) for',inz,' contaminants or organic'
      print *,'compounds in region',lz
      print *,'a value of 0.0 enters values from UCRL, Yook et.al'
      print *,'for contaminants; or octanol/water ratios from EPA for'
      print *,'organic compounds'
c
c contaminant/organic compound uptake coefficient soil to plant gate(0,1)
c
      if(nu.eq.0)read(*,*,err=5015) (yup(j,lz),j=1,inz)
      if(nu.eq.1)read(7,*) (yup(j,lz),j=1,inz)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*) (yup(j,lz),j=1,inz)
      if(iu.eq.1)print *, (yup(j,lz),j=1,inz)
c
```

```
 5016 print *,'ENTER number of soil horizons for',ipl,'plants in'
      print *,'region',lz,'; a value of 0.0 defaults to 250mm'
      print *,'horizons. If 0.0 used make sure soil depth is'
      print *,'a multiple of 250mm; if RCODE set to -1.0'
      print *,'value set to previous RCODE value; 1st RCODE value'
      print *,'must not be set equal to -1.0; others can be'
c
c number of soil layers for each plant specified in a given region
      if(nu.eq.0)read(*,*,err=5016) (rcode(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (rcode(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*) (rcode(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (rcode(l,lz),l=1,ipl)
c
 5017 print *,'ENTER horizon depth,mm, for',ipl,'plants in'
      print *,'region',lz,';if RCODE=0.0, then value must'
      print *,'be a multiple of 250mm'
c
c depth, mm, for each soil layer specified for a given plant
      if(nu.eq.0)read(*,*,err=5017) (sdpth(l,lz),l=1,ipl)
      if(nu.eq.1)read(7,*) (sdpth(l,lz),l=1,ipl)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,*) (sdpth(l,lz),l=1,ipl)
      if(iu.eq.1)print *, (sdpth(l,lz),l=1,ipl)
c
c  ** contaminant loop ********************************************************
      do 30 k=1,inz
c converts stable contaminant concentrations to grams contaminant/ml
      if (hflif(k)) 32,31,32
   31 watcn(k,lx)=watcn(k,lx)/1.0e06
c calculates decay or persisitence constant day-1 for contaminants
   32 if (hflif(k).gt.0.0) disint(k)=.693/(hflif(k)*365.)
   30 continue
c ** end contaminant loop ****************************************************
c ** plant loop ******************************************************************
      do 80 l=1,ipl
c a value of -1.0 for rcode sets equal current soil values to previous
      if (rcode(l,lz)+1.0) 41,40,41
c determines number of soil layers
   41 iicde=rcode(l,lz)
      if (rcode(l,lz)) 43,42,43
   42 iicde=int(sdpth(l,lz)/250.)+1
      rcode(l,lz)=iicde
   43 continue
 5018 print *,'ENTER horizon thickness,mm, for plant',l,'for'
      print *,iicde,'horizons in region',lz
c enters soil layer thickness, mm, for all layers
c
      if(nu.eq.0)read(*,*,err=5018) (slice(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*) (slice(l,k,lz),k=1,iicde)
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
      write(6,*) (slice(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *, (slice(l,k,lz),k=1,iicde)
c
 5019 print *,'ENTER rock fraction for plant',l,'for'
      print *,iicde,'horizons in region',lz
c sets rock,sand,silt,and clay fractions for each soil layer
      if(nu.eq.0)read(*,*,err=5019)(rock(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*)(rock(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
      write(6,*)(rock(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *,(rock(l,k,lz),k=1,iicde)
c
 5020 print *,'ENTER sand fraction for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5020) (sand(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*) (sand(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
      write(6,*) (sand(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *, (sand(l,k,lz),k=1,iicde)
c
 5021 print *,'ENTER silt fraction for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5021) (silt(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*) (silt(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*) (silt(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *, (silt(l,k,lz),k=1,iicde)
c
 5022 print *,'ENTER clay fraction for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5022) (clay(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*) (clay(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5022) ans
      if(ans.gt.0.and.nu.eq.0)goto5022
      write(6,*) (clay(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *, (clay(l,k,lz),k=1,iicde)
c
c sets wilting point,field and holding capacity for each soil layer
 5023 print *,'ENTER wilt point,percent, for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5023)(w1(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*)(w1(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5023) ans
```

```
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*)(w1(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *,(w1(l,k,lz),k=1,iicde)
c
 5024 print *,'ENTER holding capacity,percent, for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5024)(h(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*)(h(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*)(h(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *,(h(l,k,lz),k=1,iicde)
c
 5025 print *,'ENTER field capacity,percent,for plant',l,'for'
      print *,iicde,'horizons in region',lz
      if(nu.eq.0)read(*,*,err=5025) (f1(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*) (f1(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5025) ans
      if(ans.gt.0.and.nu.eq.0)goto5025
      write(6,*) (f1(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *, (f1(l,k,lz),k=1,iicde)
c
c conditionally sets ph values for each soil layer
 5026 if(pheff-1.0) 82,81,82
   81 print *,'ENTER pH for plant',l,'for',iicde,'horizons'
   82 if(pheff-1.0) 40,44,40
   44 print *,'in region',lz
c
      if (nu.eq.0)read(*,*,err=5026)(ph(l,k,lz),k=1,iicde)
      if(nu.eq.1)read(7,*)(ph(l,k,lz),k=1,iicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5026) ans
      if(ans.gt.0.and.nu.eq.0)goto5026
      write(6,*)(ph(l,k,lz),k=1,iicde)
      if(iu.eq.1)print *,(ph(l,k,lz),k=1,iicde)
c
   40 continue
c conditionally enters of soil contaminant concentrations or activities
      do90jzz=1,inz
      izz=yaz(jzz,lz)
      if(izz.eq.0)goto91
      jicde=iicde
      if(izz.eq.3)jicde=2
 5027 print *,'ENTER soil conc., act., ppm.. for plant',l,'for'
      print*,'first',jicde,'horizons for contaminant',jzz,'in region',lz
      if (nu.eq.0) read(*,*,err=5027) (z1(l,k,jzz,lz),k=1,jicde)
      if (nu.eq.1) read(7,*) (z1(l,k,jzz,lz),k=1,jicde)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5027) ans
      if(ans.gt.0.and.nu.eq.0)goto5027
      write(6,*) (z1(l,k,jzz,lz),k=1,jicde)
      if(iu.eq.1) print *, (z1(l,k,jzz,lz),k=1,jicde)
```

```
 91 continue
    if(l.eq.1)goto50
    if(jzz.gt.1)goto50
c conditionally equates soil characteristics of different plants
    if (rcode(l,lz)+1.0) 90,60,90
 60 rcode(l,lz)=rcode(l-1,lz)
    iicde=rcode(l,lz)
c
c ** soil layer loop ***************************************************
 50 do 70 j=1,iicde
    if(l.eq.1)goto72
    if(jzz.gt.1.0)goto72
    slice(l,j,lz)=slice(l-1,j,lz)
    rock(l,j,lz)=rock(l-1,j,lz)
    sand(l,j,lz)=sand(l-1,j,lz)
    silt(l,j,lz)=silt(l-1,j,lz)
    clay(l,j,lz)=clay(l-1,j,lz)
    w1(l,j,lz)=w1(l-1,j,lz)
    h(l,j,lz)=h(l-1,j,lz)
    f1(l,j,lz)=f1(l-1,j,lz)
    if (pheff-1.0) 72,71,72
 71 ph(l,j,lz)=ph(l-1,j,lz)
 72 if(j.le.2)goto70
    if (izz.eq.3) z1(l,j,jzz,lz)=z1(l,2,jzz,lz)
 70 continue
c ** end soil layer loop ***********************************************
 90 continue
c ** end contaminant loop ***********************************************
 80 continue
c ** end plant loop ****************************************************
c
c ** plant loop *********************************************************
    do 160 j=1,ipl
    rpntp(j,lz)=0.0
    iicde=rcode(j,lz)
c conditionally sets number of soil layers
    if (rcode(j,lz)) 162,161,162
161 iicde=int(sdpth(j,lz)/250.)+1
162 soilf(j,lz)=(1.0-exp(-.004*sdpth(j,lz)))/.632
c
c ** soil layer loop ***************************************************
    do 120 k=1,iicde
c
c ** contaminant loop ***************************************************
    do 110 jz=1,inz
    iatno=atno(jz)
    if (hflif(jz)) 101,100,101
c converts ppm concentrations to pci/g activity
101 pcicnx=.693*6.023e17/(2.22*1440.*365.*awt(jz)*hflif(jz))
100 continue
c used to convert ppm to grams contaminant/gram soil for stable contaminants
    pcicon=1.0e-06
    if (hflif(jz).gt.0.0) pcicon=pcicnx
c conditionally assumes that input units correct for long halflife contaminants
```

```
c or stable contaminants
      if (yaz(jz,lz)-2.0) 103,102,103
  103 if (yaz(jz,lz)-3.0) 104,102,104
  102 pcicon=1.0
c assumes input units correct for organic compound concentrations, mg/kg
  104 if(yaz(jz,lz).ge.4.0)pcicon=1.0
      z1(j,k,jz,lz)=z1(j,k,jz,lz)*pcicon
c conditionally sets concentrations and uptake coefficients to default values
      if (yaz(jz,lz)) 106,105,106
  105 z1(j,k,jz,lz)=y1(iatno,2)*pcicon
  106 if(yaz(jz,lz).lt.4.0)uptke=y1(iatno,1)/y1(iatno,2)
c organic uptake coefficient converted to natural log units
      if(yaz(jz,lz).ge.4.0)uptke=2.30259*(1.588-0.578*y1(iatno,1))
c
      if(yaz(jz,lz).ge.4.0)uptke=exp(uptke)
      if(yup(jz,lz).le.0.0)zu(j,k,jz,lz)=uptke
  110 continue
c ** end contaminant loop ******************************************************
c sets defaulted soil layer thickness to 250 mm except first and second
      if (rcode(j,lz)) 112,111,112
  111 slice(j,k,lz)=250.
  112 if (rcode(j,lz)) 115,114,115
  114 if (k.ne.1) goto 115
      slice(j,k,lz)=20.
  115 if (rcode(j,lz)) 117,116,117
  116 if (k.ne.2) goto 117
      slice(j,k,lz)=230.
c initializes soil moisture content in all layers to the wilt point
  117 wa(j,k,lz)=w1(j,k,lz)*slice(j,k,lz)/100.
  120 continue
c ** end soil layer loop ******************************************************
c conditionally enters root fractions in each soil layer, otherwise default
 5028 if(yzf(j,lz).gt.0.0)
      lprint *,'ENTER root fractions for plant',j,'for'
      if(yzf(j,lz).gt.0.0)
      lprint *,iicde,'horizons in region',lz
      if (yzf(j,lz).gt.0.0.and.nu.eq.0)
      lread(*,*,err=5028) (zf(j,l,lz),l=1,iicde)
      if (yzf(j,lz).gt.0.0.and.nu.eq.1)
      lread(7,*) (zf(j,l,lz),l=1,iicde)
      if(yzf(j,lz).gt.0.0.and.nu.eq.0)
      lprint *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(yzf(j,lz).gt.0.0.and.nu.eq.0)
      lread(*,*,err=5028) ans
      if(yzf(j,lz).gt.0.0.and.nu.eq.0.and.ans.gt.0.0)
      lgoto5028
      if(yzf(j,lz).gt.0.0)write(6,*) (zf(j,l,lz),l=1,iicde)
      if(iu.eq.1.and.yzf(j,lz).gt.0.)print *,(zf(j,l,lz),l=1,iicde)
c
c ** contaminant loop ************************************************************
      do 130 jz=1,inz
c conditionally enters soil-plant uptake coefficients, otherwise default
 5029 if(yup(jz,lz).gt.0.0)
      lprint *,'ENTER soil-plant uptake coefficients for plant',j
```

```
      if(yup(jz,lz).gt.0.0)
      lprint *,'for',iicde,'horizons,and contaminant',jz,'in region',lz
      if (yup(jz,lz).gt.0.0.and.nu.eq.0)
      lread(*,*,err=5029)(zu(j,l,jz,lz),l=1,iicde)
      if (yup(jz,lz).gt.0.0.and.nu.eq.1)
      lread(7,*)(zu(j,l,jz,lz),l=1,iicde)
      if(nu.eq.0.and.yup(jz,lz).gt.0.0)
      lprint *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0.and.yup(jz,lz).gt.0.0)
      lread(*,*,err=5029) ans
      if(ans.gt.0.0.and.nu.eq.0.and.yup(jz,lz).gt.0.0)
      lgoto5029
      if(yup(jz,lz).gt.0.)write(6,*)(zu(j,l,jz,lz),l=1,iicde)
      if(iu.eq.1.and.yup(jz,lz).gt.0.0)
      lprint *,(zu(j,l,jz,lz),l=1,iicde)
  130 continue
c ** end contaminant loop ******************************************************
      rfc=0.0
      sdpsm=0.0
c
c ** soil layer loop ***********************************************************
      do 150 l=1,iicde
c conditionally calculates root fractions in each soil layer
      sdpsm=sdpsm+slice(j,l,lz)
      zfll=(1.0-exp(-.004*sdpsm))/.632
      if (yzf(j,lz)) 132,131,132
  131 zf(j,l,lz)=(zfll-rfc)/soilf(j,lz)
  132 rfc=zfll
  150 continue
c ** end soil layer loop *******************************************************
c resets all defaulted soil layer arrays to estimated value
      rcode(j,lz)=iicde
  160 continue
c ** end plant loop ************************************************************
c
c converts h,w1,and f1 from percent water to mm of water in each soil
c layer
c
c ** plant loop ****************************************************************
      do 190 k=1,ipl
      iicde=rcode(k,lz)
c
c ** soil layer loop ***********************************************************
      do 180 j=1,iicde
      rh1(k,j,lz)=0.0
      rh2(k,j,lz)=0.0
      h(k,j,lz)=0.01*h(k,j,lz)*slice(k,j,lz)
      w1(k,j,lz)=0.01*w1(k,j,lz)*slice(k,j,lz)
      f1(k,j,lz)=0.01*f1(k,j,lz)*slice(k,j,lz)
c
c ** contaminant loop **********************************************************
      do 170 jz=1,inz
c conditionally holds specific contaminant concentration at 0.0 for delayed
c soil contamination events. z1dl is temporary array, z1 is permanent
```

```
      if (ysrt(jz,lz).gt.0.0) z1dl(k,j,jz,lz)=z1(k,j,jz,lz)
      if (ysrt(jz,lz).gt.0.0) z1(k,j,jz,lz)=0.0
  170 continue
c ** end contaminant loop ********************************************************
  180 continue
c ** end soil layer loop ****************************************************
  190 continue
c ** end plant loop ********************************************************
c
c  estimates the effects of clay content and ph on the uptake coeffi-
c    cient zu based on cs-137 data
c
c  estimates the effects of calcium and ph on the uptake coefficient
c  based on sr-90 data
c
c ** plant loop ************************************************************
      do 220 k=1,ipl
      iicde=rcode(k,lz)
c
c ** soil layer loop ******************************************************
      do 210 l=1,iicde
c effect of clay
      efocl=12.92*exp(-.12792*clay(k,l,lz)*100.)
c effect of ph
      efoph=39.81*exp(-.4605*ph(k,l,lz))
c effect of calcium not in use
c     if (caleff.eq.1.0) efoca=7.95*(z1(k,l,7,lz)*5000.)**(-0.748)
      if (clayef) 222,221,222
  221 efocl=1.0
  222 if (pheff) 224,223,224
  223 efoph=1.0
c estimates volumetric water content at wilt point and holding capacity
c from estimates of rock,sand,silt,and clay soil fractions
  224 rockj=rock(k,l,lz)
      sandj=sand(k,l,lz)
      clayj=clay(k,l,lz)
      siltj=silt(k,l,lz)
      w1bar(l,lz)=slice(k,l,lz)*(sandj*.02+siltj*.023+clayj*.049+rockj*.
     1 002)
      hbar(l,lz)=slice(k,l,lz)*(sandj*.44+siltj*.48+clayj*.52+rockj*.044
     1 )
c ** contaminant loop ****************************************************
      do 200 jz=1,inz
c resets soil-plant uptake coefficients adjusted for ph and clay
      zu(k,l,jz,lz)=zu(k,l,jz,lz)*efoph*efocl
c initializes contaminant concentrations in dead roots and root humus
      z4r(k,l,jz,lz)=0.0
      z5r(k,l,jz,lz)=0.0
  200 continue
c ** end contaminant loop ************************************************
  210 continue
c ** end soil layer loop ***************************************************
  220 continue
c ** end contaminant loop ************************************************
```

```
 230 continue
c ** end region loop ******************************************************
    print *,'END OF INPUT FOR WATFLX SUBROUTINE'
    if(nu.eq.1)goto5503
5030 print *,'Do you wish to RE-ENTER input to WATFLX?'
    print *,'yes(to beginning of subroutine): ENTER 1.0'
    print *,'yes(but after GEOFLX input if any): ENTER 2.0'
    print *,'no: ENTER 0.0; ENTER now: 1.,2.,or 0.'
    read(*,*,err=5030)ans
    if(ans)165,5503,165
 165 rewind 6
    achek3=151515.
    if(ans-1.) 163,166,163
 166 achek3=141414.
 163 do5504l=1,20000
    read(6,*,err=5504)achek2
    if(achek2-achek3)5504,5507,5504
5507 if(ans-2.) 164,5508,164
5508 backspace 6
    nu=nuu
    call clear_screen@
    goto5815
 164 if(ans-1.) 5504,5506,5504
5506 backspace 6
    nu=nuu
    call clear_screen@
    goto5050
5504 continue
5503 continue
    nu=nuu
    call clear_screen@
c
c estimates hydraulic conductivity coefficients for rock,sand,silt,and
c clay soil fractions. cz1 represents regression coefficient,cz2
c represents exponent with volumetric water content as independent variable
c
    cz1rk=1.56e06
    cz1sd=1.56e05
    cz1st=6.99e04
    cz1cl=1.04e04
    cz2rk=5.27
    cz2sd=5.27
    cz2st=6.4
    cz2cl=9.2
c
c estimates matrix potential coefficients for rock,sand,silt,and clay soil
c fractions. ym1 represents regression coefficient,ym2 represents exponent
c with volumetric water content as independent variable
c
    ym1rk=50.
    ym1sd=50.
    ym1st=40.
    ym1cl=41.4
    ym2rk=0.5
```

```
      ym2sd=1.5
      ym2st=2.6
      ym2cl=4.1
c initializes channel and region parameters for hydrological simulations
      ic=chnnu
      ich=ipx+1
      icht=ich+ipx
c
      return
  240 continue
c
c ** contaminant loop ******************************************************
      do 270 jz=1,inz
c conditionally sets soil concentrations to specific values after lag
c period (day and year) has ended
c year lag
      if (ysrt(jz,lx)-ai1) 270,271,270
c day lag
  271 if (dsrt(jz,lx)-aiii) 270,272,270
c
c ** plant loop from main ***************************************************
  272 do 260 k=lpn,lpn
      iicde=rcode(k,lx)
c
c ** soil layer loop ********************************************************
      do 250 j=1,iicde
c set to temporary array value
      z1(k,j,jz,lx)=z1(k,j,jz,lx)+z1dl(k,j,jz,lx)
  250 continue
c ** end contaminant loop ***************************************************
  260 continue
c ** end plant loop *********************************************************
  270 continue
c ** end contaminant loop ***************************************************
c annual runoff parameters initialized  on julian day 1 of each year
      if (aiii-1.0) 300,301,300
  301 if (lpn.ne.1) goto 300
c
c ** plant loop ************************************************************
      do 290 jj=1,ipl
      mofys(jj,lx)=0.0
c
c ** contaminant loop ******************************************************
      do 280 jz=1,inz
      mofym(jj,jz,lx)=0.0
  280 continue
c ** end contaminant loop ***************************************************
      mofyg(jj,lx)=0.0
c
c  if irrigation is specified (rirrc>0.0) then the input for ahrv,
c  cuta, awata, dira and rirrc must be made for pltgro to be used
c  here. there are four types of crop practices:
c        1. irrigate and harvest annual crop
c        2. irigate and harvest perennial crop
```

```
c        3. irrigate  and not harvest (annual or perennial can be
c           used)
c        4. no irrigation and harvest dryland crop (annual or
c           perennial)
c
c  the amount of irrigation water, precipitation, and snow melt are
c  added to the top soil layer.
c
c a value of 1.0 for cuta indicates greater than 365 day irrigation-
c harvesting schedule
      if (cuta(jj,1,lx)-1.0) 291,290,291
c resets irrigation and harvesting daily counters
  291 irg(jj,lx)=2
      ica(jj,lx)=2
  290 continue
c ** end plant loop ***************************************************
c initializes daily runoff parameters
  300 rnofds(lpn,lx)=0.0
      rnofdg(lpn,lx)=0.0
c
c ** contaminant loop *****************************************************
      do 310 jz=1,inz
      rnofdm(lpn,jz,lx)=0.0
  310 continue
c ** end contaminant loop ***************************************************
c checks for water surplus "ponding" in top soil layer
      wsrp(lpn,lx)=wa(lpn,1,lx)-h(lpn,1,lx)
      if (wsrp(lpn,lx).le.0.0) wsrp(lpn,lx)=0.0
c initializes available moisture added to top soil layer
      prec(lx)=wsrp(lpn,lx)
      wsrp(lpn,lx)=0.0
c precipitation and snowmelt are added to top soil layer
      pppt=p(lx)+snm(lx)/10.
c average daily temperature must exceed 2 deg. C for addition to occur
      if (t(lx)-2..gt.0.0) wa(lpn,1,lx)=wa(lpn,1,lx)+pppt/(1.-bdrk(lx))
c
c irrigation module
c
c irrigation water accumulator initialized
      addwat=0.0
c specific plant day counter set to current julian day number
      attt(lpn,lx)=aiii
c bypasses irrigation module if rirrc equals to 0.0 for a given plant
      if (rirrc(lpn,lx)) 361,360,361
c identifies annually harvested plants if cuta not equal to 1.0
  361 if (cuta(lpn,1,lx)-1.0) 320,330,320
c day number for annual harvested plants set to current day number
  320 tim(lpn,lx)=aiii
      irggg=irg(lpn,lx)
c sets specific plant day number to current day number
      attt(lpn,lx)=tim(lpn,lx)
      go to 350
c increments day number beyond julian day 365 for multiyear plants
  330 tim(lpn,lx)=tim(lpn,lx)+1.0
```

```
c checks to see if last harvest for multiyear plant is exceeded
      if (tim(lpn,lx).le.cutmax(lpn,lx)) go to 340
c resets time and irrigation parameters for new crop growth
      tim(lpn,lx)=aiii
      ica(lpn,lx)=2
      irg(lpn,lx)=2
c sets plant specific day counter for multiyear plants
  340 attt(lpn,lx)=tim(lpn,lx)
      irggg=irg(lpn,lx)
c tests all irrigated plants for irrigation day numbers
  350 if (attt(lpn,lx)-dira(lpn,irggg,lx)) 360,362,360
c adds specified irrigation water amount, mm, top soil layer
  362 wa(lpn,1,lx)=wa(lpn,1,lx)+awata(lpn,irggg,lx)*25.4/(1.-bdrk(lx))
c increments accumulated irrigation water parameter
      addwat=awata(lpn,irggg,lx)*25.4
      irg(lpn,lx)=irg(lpn,lx)+1
  360 continue
c increments cumulative soil moisture additions from irrigation,
c snowmelt,and rainfall; prec is same from all sources including ponding
      p1(lpn,lx)=p1(lpn,lx)+pppt+addwat
      prec(lx)=prec(lx)+pppt+addwat
c number of soil layers specified for a given plant
      iicde=rcode(lpn,lx)
c
c ** contaminant loop ***************************************************
      do 370 jz=1,inz
c top soil layer contaminant conc. adjusted for contaminated water addition
      z1(lpn,1,jz,lx)=z1(lpn,1,jz,lx)+addwat*watcn(jz,lx)/(slice(lpn,1
     1 ,lx)*1400.)
  370 continue
c ** end contaminant loop ***********************************************
c
c initializes evapotranspiration accumulator
      if(aiche(lx).gt.0.0.and.rss(lpn,lx).gt.0.0)goto371
      if(rscde(lx).gt.0.0.or.adsuc(lx).gt.0.0)goto371
      ea(lpn,lx)=0.0
  371 rsi(lpn,lx)=0.0
c
c  estimates  root growth and root death for each plant simulated.
c  RPNTP is the root penetration total depth and ZFF is the fraction
c  the roots have penetrated the lowest root layer.
c
c estimates root penetration rate, mm/day, for plants with different
c mean life-spans, years (HF)
      ai1p=5.*2.54
      if (hf(lpn,lx).gt.1.0) ai1p=3.0*2.54
      if (hf(lpn,lx).gt.50.) ai1p=2.54
      if (hf(lpn,lx).gt.150.) ai1p=1.00
c cheatgrass (Bromus tectorum) is given special treatment
      if (pltid(lpn,lx)-22.) 422,421,422
  421 if (aiii-180.) 422,424,422
  424 rpntp(lpn,lx)=0.0
  422 if (pltid(lpn,lx)-22.) 423,420,423
c isolates annuals from perennials
```

```
 423 if (hf(lpn,lx)-1.0) 381,380,381
c does not allow perennial plant root growth until one month before to one
c month after the frost-free period
 381 if (aiii.lt.sffp(lx)-30.or.aiii.gt.effp(lx)+30) ai1p=0.0
     go to 420
c separates irrigated harvested plants from other annuals(dirmax=0,cutmax=0)
 380 if (dirmax(lpn,lx)) 383,382,383
 382 if (cutmax(lpn,lx)) 383,410,383
 383 if (dirmax(lpn,lx)) 391,390,391
c does not allow root growth until first irrigation schedule
 391 if (attt(lpn,lx).lt.dira(lpn,1,lx)) rpntp(lpn,lx)=0.0
     if (cutmax(lpn,lx)) 401,400,401
c does not allow root growth after final harvest
 401 if (attt(lpn,lx).ge.cutmax(lpn,lx)) rpntp(lpn,lx)=0.0
     go to 420
c sets root depth to 0.0 30 days prior to the frost-free period or if the
c final harvest day number is exceeded (non-irrigated harvested plants)
 390 if (attt(lpn,lx).lt.sffp(lx)-30..or.attt(lpn,lx).ge.cutmax(lpn,lx)
    1 ) rpntp(lpn,lx)=0.
     go to 420
c sets root depth to 0.0 prior to the first irrigation day, and before
c 30 days beyond the frost-free period is reached (unharvested plants
c irrigated plants)
 400 if (attt(lpn,lx).lt.dira(lpn,1,lx).or.attt(lpn,lx).gt.effp(lx)+30.
    1 ) rpntp(lpn,lx)=0.
     go to 420
c sets non-irrigated, non-harvested annual plant root penetration to 0.0
c up to 30 days prior to 30 days after the frost-free period
 410 if (aiii.lt.sffp(lx)-30..or.aiii.gt.effp(lx)+30.) rpntp(lpn,lx)=0.
c increments root penetration and limits to total soil depth
 420 rpntp(lpn,lx)=rpntp(lpn,lx)+ai1p
     if (pltid(lpn,lx)) 426,425,426
 425 rpntp(lpn,lx)=0.0
c determines root penetration fraction(zff) for each soil layer excluding
c the top layer
 426 depsum=20.
     iicde=rcode(lpn,lx)
c
c ** soil layer loop ***************************************************
     do 430 l=1,iicde
     if (l.ge.2) depsum=depsum+slice(lpn,l-1,lx)


c estimates penetration fraction to be twice the actual amount estimated
c to allow for root hair moisture boundary effects(zff=1 is maximum)
     zff(lpn,l,lx)=2.0*rpntp(lpn,lx)/depsum
     if (zff(lpn,l,lx).ge.1.0) zff(lpn,l,lx)=1.0
 430 continue
c ** end soil layer loop ***********************************************
c
c estimates infiltration from precip. and snowmelt;and runoff/day,mm
     if (t(lx).le.2.0) go to 440
c calls RUNOFF subroutine
     call runoff (lpn,lx)
```

```
 440 continue
    if (lpn.ne.ipl) go to 450
c calls GEOFLX subroutine
    if (chnnu) 451,450,451
 451 call geoflx (lx,bpx,rdidx)
c
c  estimates evapotranspiration loss from each soil layer from
c  roots and adjusts the soil layer water content daily.
c
c potential evapotranspiration
 450 e22=e2(lx)
c leaf area index
    tlain=.013*wnd(lpn,lx)
    eaval(lpn,1,lx)=0.0
c
c ** soil layer loop ***********************************************
    do 480 k=2,iicde
c initialize evapotranspiration moisture withdrawal from each layer
    eaval(lpn,k,lx)=0.0
    if (e22.le.0.0) go to 465
    cdrp(k,lx)=(1.0/(h(lpn,k,lx)-w1(lpn,k,lx)))
    if(e5(lpn,lx).le.0.0)goto480
    if (wa(lpn,k,lx).le.w1(lpn,k,lx)) go to 465
    walpn=wa(lpn,k,lx)
    if (walpn.gt.h(lpn,k,lx)) walpn=h(lpn,k,lx)
c
c estimate evapotranspiration, mm/day, from each soil layer initially
c withdrawing required moisture from layer 2, and if not adequate, from
c lower layers until satisfied. Layer 1 (20mm) is not included
c
    eaval(lpn,k,lx)=cdrp(k,lx)*((walpn-w1(lpn,k,lx))**2)*zff(lpn,k,lx)
    eaval(lpn,k,lx)=eaval(lpn,k,lx)*tlain
c estimates soil moisture in layer available for evapotranspiration
    avwt=wa(lpn,k,lx)-w1(lpn,k,lx)
    if (eaval(lpn,k,lx).ge.avwt) eaval(lpn,k,lx)=avwt
    if (eaval(lpn,k,lx).ge.e22) eaval(lpn,k,lx)=e22
    e22=e22-eaval(lpn,k,lx)
    if (e22.le.0.0) e22=0.0
    if(rirrc(lpn,lx).gt.0.0)goto473
    if(aiche(lx).gt.0.0.and.rss(lpn,lx).gt.0.0)goto476
    if(rscde(lx).gt.0..or. adsuc(lx).gt.0.) goto465
c accumulates total moisture withdrawal from all layers required
 473 ea(lpn,lx)=ea(lpn,lx)+eaval(lpn,k,lx)
    wa(lpn,k,lx)=wa(lpn,k,lx)-eaval(lpn,k,lx)
    if (wa(lpn,k,lx).le.w1(lpn,k,lx)) wa(lpn,k,lx)=w1(lpn,k,lx)
    goto460
 465 if(lpn.ne.ipl)goto460
 476 if(aiche(lx).gt.0.0.and.lpn.ne.ipl)goto460
    easum=0.0
    wndt=0.0
    wndtt=0.0
    ipll=ipl
    if(rscde(lx).gt.0.0.or.adsuc(lx).gt.0.0)ipll=1
    do478lp=1,ipll
```

```
      if(ipll.eq.1)goto477
      if(rss(lp,lx).le.0.0)goto478
      if(rsi(lp,lx).gt.0.0)goto478
      rsi(lp,lx)=1.0
  477 do463jj=1,ipl
      if(rirrc(jj,lx).gt.0.0)goto463
      if(rscde(lx).gt.0.0.or.adsuc(lx).gt.0.0)goto479
      if(rss(jj,lx).le.0.0)goto463
      if(lp.eq.jj)goto463
      if(rss(lp,lx).gt.rss(jj,lx))goto463
      if(rss(lp,lx).lt.rss(jj,lx))goto463
      rsi(jj,lx)=1.0
  479 iicde=rcode(jj,lx)
      kki=0
      kkj=0
      do467kk=2,iicde
      if(eaval(jj,kk,lx).le.0.0)goto468
      kki=kki+1
      easum=easum+eaval(jj,kk,lx)*wnd(jj,lx)
      if(kki.eq.1)wndt=wndt+wnd(jj,lx)
  468 if(rscde(lx).gt.0.0.or.adsuc(lx).gt.0.0)goto467
      if(eaval(lp,kk,lx).le.0.0)goto467
      kkj=kkj+1
      if(kkj.eq.1)wndtt=wnd(lp,lx)
      easum=easum+eaval(lp,kk,lx)*wnd(lp,lx)
  467 continue
  463 continue
      if(easum.le.0.0)goto478
      wndt=wndt+wndtt
      easum=easum/wndt
      easump=easum
      do464jj=1,ipl
      if(rirrc(jj,lx).gt.0.0)goto464
      if(rscde(lx).gt.0.0.or.adsuc(lx).gt.0.0)goto475
      if(rss(lp,lx).gt.rss(jj,lx))goto464
      if(rss(lp,lx).lt.rss(jj,lx))goto464
  475 easum=easump
      iicde=rcode(jj,lx)
      ea(jj,lx)=0.0
      do466 kk=2,iicde
      if(eaval(jj,kk,lx).le.0.0)goto466
      avwt=(wa(jj,kk,lx)-w1(jj,kk,lx))
      if(easum.ge.avwt)easum=avwt
      eaval(jj,kk,lx)=easum
      ea(jj,lx)=ea(jj,lx)+eaval(jj,kk,lx)
      wa(jj,kk,lx)=wa(jj,kk,lx)-easum
      if(wa(jj,kk,lx).le.w1(jj,kk,lx))wa(jj,kk,lx)=w1(jj,kk,lx)
      easum=easump-easum
      if(easum.le.0.0)easum=0.0
  466 continue
  464 continue
  478 continue
c     this module determines whether root penetration has reached the top
c     of an aquifer if present to allow transpiration for overlying plants.
```

```
 460 if (k.ne.iicde) go to 480
    if (chnnu) 481,480,481
 481 if (e22.le.0.0) goto480
    alx=lx
c ** regional loop for regions containing an aquifer below a channel *****
    do 470 lc=ich,icht
    if (chnaq(lc)-alx) 470,471,470
c converts root depth to feet
 471 rpnft=rpntp(lpn,lx)/304.8
    dtaqu=sdpth(lpn,lx)/304.8+hbt(lc)-htx(lc)
    if (rpnft.lt.dtaqu) go to 470
c estimates volume ,cubic feet, required to satisfy p.e.t adjusted for
c that supplied from upper soil levels of region lx
    e22v=(e22/304.8)*rarea(lx)*43562.
c limits available moisture to one-half present in aquifer
    if (e22v.ge.0.5*vaqf(lc))e22v=0.5*vaqf(lc)
    e22a=304.8*e22v/(rarea(lx)*43562.)
c increments plant last soil layer and total evapotranspiration
    eaval(lpn,k,lx)=eaval(lpn,k,lx)+e22a
    ea(lpn,lx)=ea(lpn,lx)+e22a*tlain
    vaqf(lc)=vaqf(lc)-e22v
    if(vaqf(lc).le.0.0)vaqf(lc)=0.0
    e22=e22-e22a
    if(e22.le.0.0)e22=0.0
 470 continue
c ** end region loop with aquifers ****************************************
 480 continue
c ** end soil layer loop ***************************************************
c
c  estimates surface evaporation in top 20mm soil layer based on
c  potential evapotranspiration (e2) and leaf area index (tlai).
c
    evap(lpn,lx)=0.0
    if (e22.le.0.0) go to 490
c limits available water for evaporation down to 5% soil moisture
    if (wa(lpn,1,lx).le..05*(1.-bdrk(lx))*slice(lpn,1,lx)) go to 490
    wav=wa(lpn,1,lx)-.05*(1.-bdrk(lx))*slice(lpn,1,lx)
    wav=0.5*wav
    if (e22.ge.wav) e22=wav
c estimates evaporation, mm, as a function of leaf area index
    evap(lpn,lx)=e22*exp(-.4*tlain)
    if (evap(lpn,lx).gt.wav) evap(lpn,lx)=wav
    evap(lpn,lx)=0.5*evap(lpn,lx)
c adjust soil moisture in top layer for evaporation loss
    wa(lpn,1,lx)=wa(lpn,1,lx)-evap(lpn,lx)
 490 continue
c
c  for each soil layer the volumetric wetness change is calculated
c  using maximum and minimum water holding capacities based on
c  data obtained by Hillel. Conductivity and matrix potential
c  are then calculated for each soil layer.
c  it is assumed that soil particles larger than sand have a conduc.
c  10 x that of sand,and a matrix potential 1/10 that of sand ,
c  and that a 1.16 m depth was utilized for making the measurements.
```

```
c
c ** soil layer loop ************************************************
      iicde=rcode(lpn,lx)
      do 500 j=1,iicde
c converts moisture thickness, mm, to moisture volume, ml, based on soil
c bulk density of 1.4
      wa(lpn,j,lx)=wa(lpn,j,lx)*1.4
c
c estimates volume availability, availability at field capacity,and at
c wilt point
c
      thel=wa(lpn,j,lx)/slice(lpn,j,lx)
      thef=1.4*f1(lpn,j,lx)/slice(lpn,j,lx)
      thew=1.4*w1(lpn,j,lx)/slice(lpn,j,lx)
      if (thef.ge.0.6) thef=.59
      if (thew.ge..16) thew=.15
c estimates pressure deficit in bars for each soil layer
      if (thel.gt.thef) phi(lpn,j,lx)=(.6-thel)*.3/(.6-thef)
      if (thel.le.thef) phi(lpn,j,lx)=.3+(thef-thel)*2.7/(thef-.16)
      if (thel.lt.0.16) phi(lpn,j,lx)=3.+(.16-thel)*12./(.16-thew)
      if (thel.le.thew) phi(lpn,j,lx)=15.+(thew-thel)*2000.
c
      if (phi(lpn,j,lx).le.0.0) phi(lpn,j,lx)=0.0
c limits pressure deficit to 15 bars at wilt point
      if (phi(lpn,j,lx).ge.15.) phi(lpn,j,lx)=15.
c soil moisture content, mm ; bounded
      walpnj=wa(lpn,j,lx)
      if (walpnj.gt.hbar(j,lx)) walpnj=hbar(j,lx)
      if (walpnj.lt.w1bar(j,lx)) walpnj=w1bar(j,lx)
c estimates soil moisture potential (volumetric)
      wfr(lpn,j,lx)=(walpnj/slice(lpn,j,lx))
c estimates hydraulic conductivity from rock,sand,silt,and clay soil frac.,
c mm/mm-soil/day, for each soil layer
      cz(lpn,j,lx)=sand(lpn,j,lx)*cz1sd*wfr(lpn,j,lx)**cz2sd
      cz(lpn,j,lx)=cz(lpn,j,lx)+rock(lpn,j,lx)*cz1rk*wfr(lpn,j,lx)*
     1 *cz2rk
      cz(lpn,j,lx)=cz(lpn,j,lx)+silt(lpn,j,lx)*cz1st*wfr(lpn,j,lx)*
     1 *cz2st
      cz(lpn,j,lx)=cz(lpn,j,lx)+clay(lpn,j,lx)*cz1cl*wfr(lpn,j,lx)*
     1 *cz2cl
c estimates soil moisture potential,mm/mm-soil,from rock,sand,silt,and clay
c fractionns for each soil layer
      ym(lpn,j,lx)=sand(lpn,j,lx)*ym1sd*wfr(lpn,j,lx)**(-ym2sd)
      ym(lpn,j,lx)=ym(lpn,j,lx)+rock(lpn,j,lx)*ym1rk*wfr(lpn,j,lx)**(
     1 -ym2rk)
      ym(lpn,j,lx)=ym(lpn,j,lx)+silt(lpn,j,lx)*ym1st*wfr(lpn,j,lx)**(
     1 -ym2st)
      ym(lpn,j,lx)=ym(lpn,j,lx)+clay(lpn,j,lx)*ym1cl*wfr(lpn,j,lx)**(
     1 -ym2cl)
  500 continue
c ** end soil layer loop ************************************************
c
c  water flux and direction is calculated using the matric potential
c  (ym) difference between layers. a negative flux indicates upward
```

```
c  movement and positive downward. cz= hydraulic conductivity
c  first soil layer:
c
      watakd(lpn,1,lx)=cz(lpn,1,lx)+cz(lpn,1,lx)*0.5*ym(lpn,2,lx)
      wataku(lpn,1,lx)=0.0
      if (watakd(lpn,1,lx).le.0.0) watakd(lpn,1,lx)=0.0
c water flux converted to mm/mm assuming 1.16 meter column from which
c regression equations were derived
      watakd(lpn,1,lx)=watakd(lpn,1,lx)/1160.
c
c ** soil layer loop ***************************************************
      do 510 l=2,iicde
      ymp1=0.0
      if (l+1.le.iicde) ymp1=ym(lpn,l+1,lx)
      dym=ymp1-ym(lpn,l-1,lx)
      watakd(lpn,l,lx)=cz(lpn,l,lx)+cz(lpn,l,lx)*0.5*dym
      wataku(lpn,l,lx)=0.0
      if (watakd(lpn,l,lx).lt.0.0) wataku(lpn,l,lx)=-watakd(lpn,l,lx)
      if (watakd(lpn,l,lx).lt.0.0) watakd(lpn,l,lx)=0.0
      watakd(lpn,l,lx)=watakd(lpn,l,lx)/1160.
      wataku(lpn,l,lx)=wataku(lpn,l,lx)/1160.
  510 continue
c ** end soil layer loop ***********************************************
c
c  estimates the loss of water from the bottom of each soil layer
c
c ** soil layer loop ***************************************************
      do 530 j=1,iicde
c compares available soil moisture, mm/mm-soil, with estimated flux
c in either direction
      wajp1=wajp1/slice(lpn,j,lx)
      flodn=watakd(lpn,j,lx)
      floup=wataku(lpn,j,lx)
      flosm=flodn+floup
      flosmp=flosm
      if (flosm) 521,520,521
  521 if (flosmp.gt.wajp1) flosmp=wajp1
c adjust flux. mm/mm-soil for moisture availabilty, mm/mm
      flodn=flodn*flosmp/flosm
      floup=floup*flosmp/flosm
c sets losses below root zone to 0.005/day or to 0.0 when channel is present
c
  520 if (j.eq.iicde) flodn=0.005*flodn
      if (j.eq.iicde.and.chnnu.gt.0.0) flodn=0.0
      if (j+1.gt.iicde) go to 531
c accumulates moisture loss, mm in extra (sink) soil compartment
      wajp1=wa(lpn,j+1,lx)+flodn*slice(lpn,j,lx)
c upward or downward flux from last soil layer, mm/mm-soil
  531 watakd(lpn,j,lx)=flodn
      wataku(lpn,j,lx)=floup
  530 continue
c ** end soil layer loop ***********************************************
c
c  checks and adjusts flux between layers and calculates the water con-
```

```
c  tent in each layer to not exceed holding capacity (except for first
c  layer, and not drain below the minimum 0f 5 %
c
c ** soil layer loop ******************************************************
      do 560 j=1,iicde
c starts with bottom layer
      jj=iicde-j+1
      flxi=0.0
c checks for layer above layer jj; calculates downward flux from it,mm
      if (jj-1.gt.0) flxi=watakd(lpn,jj-1,lx)*slice(lpn,jj-1,lx)
      watu=0.0
c checks for layer above layer jj; calculates upward flux from it,mm
      if (jj+1.le.iicde) watu=wataku(lpn,jj+1,lx)*slice(lpn,jj+1,lx)
c estimates total flux into current layer,mm
      flxi=flxi+watu
c calculates and sums upward and downward flux from current layer,mm
      flxout=watakd(lpn,jj,lx)*slice(lpn,jj,lx)
      flxout=flxout+wataku(lpn,jj,lx)*slice(lpn,jj,lx)
c calculates change in moisture content, mm/mm-soil for current layer
      delta=(flxi-flxout)/slice(lpn,jj,lx)
      if (delta) 561,560,561
c converts summed upward and downward fluxes from upper,current, and
c lower layers, mm/mm-soil
  561 flxi=flxi/slice(lpn,jj,lx)
      flxout=flxout/slice(lpn,jj,lx)
      flxia=flxi
      flxota=flxout
c estimates actual mm/mm-soil of current soil layer
      wajj=wa(lpn,jj,lx)/slice(lpn,jj,lx)
      hbj=hbar(jj,lx)
      w1jj=w1bar(jj,lx)
c limits summed fluxes for all layers ,to not exceed holding capacity or to
c less than minimum moisture content (0.05),mm/mm-soil
      if (wajj+delta.gt.hbj) flxia=flxi-(wajj+delta-hbj)
      if (wajj+delta.lt.w1jj) flxota=flxout-(w1jj-wajj-delta)
      if (flxia.le.0.0) flxia=0.0
      if (flxota.le.0.0) flxota=0.0
      if (flxi.le.0.0) go to 550
      if (jj-1.le.0) go to 540
c adjusts down-flux from upper layer to current layer.mm/mm-soil
      if (jj-1.gt.0) watakd(lpn,jj-1,lx)=watakd(lpn,jj-1,lx)*flxia/flxi
      if (watakd(lpn,jj-1,lx).le.0.0) watakd(lpn,jj-1,lx)=0.0
  540 if (jj+1.gt.iicde) go to 550
c adjusts up-flux from lower layer into cuurent layer, mm/mm-soil
      wataku(lpn,jj+1,lx)=wataku(lpn,jj+1,lx)*flxia/flxi
      if (wataku(lpn,jj+1,lx).le.0.0) wataku(lpn,jj+1,lx)=0.0
  550 if (flxout.le.0.0) go to 560
c adjusts upward and downward fluxes out of soil current layer, mm/mm-soil
      wataku(lpn,jj,lx)=wataku(lpn,jj,lx)*flxota/flxout
      watakd(lpn,jj,lx)=watakd(lpn,jj,lx)*flxota/flxout
      if (watakd(lpn,jj,lx).le.0.0) watakd(lpn,jj,lx)=0.0
      if (wataku(lpn,jj,lx).le.0.0) wataku(lpn,jj,lx)=0.0
  560 continue
c ** end soil layer loop ************************************************
```

```
c
c ** soil layer loop starting with soil layer 3 ************************
      do 570 j=3,iicde
      if (wataku(lpn,j,lx).le.0.0) wataku(lpn,j,lx)=0.0
      if (wataku(lpn,j,lx)) 571,570,571
  571 waj1=wa(lpn,j-1,lx)/slice(lpn,j-1,lx)
      wajj=wa(lpn,j,lx)/slice(lpn,j,lx)
c estimates net potential moisture, mm/mm-soil, in upper (j-1) layer from
c addition of extreme upper(j-2) layer, and losses from current(j) layer
c in both directions
      watop=waj1+watakd(lpn,j-2,lx)+wataku(lpn,j,lx)-watakd(lpn,j-1,lx)
      watop=watop-wataku(lpn,j-1,lx)
c sets influx from lower(j+1) layer ,mm/mm-soil,to current(j) layer
      watu=0.0
      if (j+1.le.iicde) watu=wataku(lpn,j+1,lx)
c estimates net potential moisture in current(j) layer as above
      wabot=wajj+watakd(lpn,j-1,lx)+watu-watakd(lpn,j,lx)
      wabot=wabot-wataku(lpn,j,lx)
c finds ratio with holding capacity
      watrr=watop/hbar(j-1,lx)
      wabrr=wabot/hbar(j,lx)
c does not layer j-1 ratio to exceed that of current(j) layer by adjusting
c upward flux from (j) if necessary.
      if (wabrr.gt.watrr) go to 570
      wataku(lpn,j,lx)=wataku(lpn,j,lx)-(watop-wabot)
      if (wataku(lpn,j,lx).le.0.0) wataku(lpn,j,lx)=0.0
  570 continue
c ** end soil layer loop ***********************************************
c
c soil layer loop ******************************************************
      do 580 j=1,iicde
c converts fluxes from mm-water/mm-soil to mm-water
      watakd(lpn,j,lx)=watakd(lpn,j,lx)*slice(lpn,j,lx)
      wataku(lpn,j,lx)=wataku(lpn,j,lx)*slice(lpn,j,lx)
  580 continue
c ** end soil layer loop ***********************************************
c
c ** soil layer loop ******************************************************
      do 600 j=1,iicde
      watupa(lpn,j,lx)=0.0
      watupb(lpn,j,lx)=0.0
c increments moisture,mm,in current layer(j) from (j-1) and (j+1) layers
      if (j-1.gt.0) wa(lpn,j,lx)=wa(lpn,j,lx)+watakd(lpn,j-1,lx)
      if (j+1.le.iicde) wa(lpn,j,lx)=wa(lpn,j,lx)+wataku(lpn,j+1,lx)
      walpj=wa(lpn,j,lx)
      walm=.05*(1.-bdrk(lx))*slice(lpn,j,lx)
c calculates moisture leaving current layer(j),mm
      waflx=watakd(lpn,j,lx)+wataku(lpn,j,lx)
      waflxt=waflx
      if (waflxt.le.0.0) go to 590
c does not allow outflux to reduce moisture,mm, in (j) below 5% minimum
      if (walpj-waflx.lt.walm) waflx=walpj-walm
      wa(lpn,j,lx)=wa(lpn,j,lx)-waflx
      if (wa(lpn,j,lx).le.walm) wa(lpn,j,lx)=walm
```

```
      watakd(lpn,j,lx)=watakd(lpn,j,lx)*waflx/waflxt
      if (watakd(lpn,j,lx).le.0.0) watakd(lpn,j,lx)=0.0
      wataku(lpn,j,lx)=wataku(lpn,j,lx)*waflx/waflxt
      if (wataku(lpn,j,lx).le.0.0) wataku(lpn,j,lx)=0.0
c converts soil moisture to mm/mm-soil and does not allow below 5% minimum
  590 wfr(lpn,j,lx)=wa(lpn,j,lx)/slice(lpn,j,lx)
      w1barn=w1bar(j,lx)/slice(lpn,j,lx)
      if (wfr(lpn,j,lx).le.w1barn) wfr(lpn,j,lx)=w1barn
c converts soil moisture from volumetric ml to gravimetric mm water
      wa(lpn,j,lx)=wa(lpn,j,lx)/1.4
c creates new arrays for moisture fluxes from upper and lower layers,mm
      if (j-1.gt.0) watupa(lpn,j,lx)=watakd(lpn,j-1,lx)
      if (j+1.le.iicde) watupb(lpn,j,lx)=wataku(lpn,j+1,lx)
      if (j.lt.iicde) go to 600
c     couples excess infiltration water to ground water losses
c     if ground water flow option is utilized(see flodn above)
      rnofdg(lpn,lx)=watakd(lpn,j,lx)
      rnofyg(lpn,lx)=rnofyg(lpn,lx)+rnofdg(lpn,lx)
  600 continue
c ** end soil layer loop ************************************************
      return
c
      end
```

14.17. ZOOGRO (bqm.for)

```
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      subroutine zoogro (in,lx,bpx,humx,rdidx)
c
c ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      include 'main.inc'
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),wt1(kr,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      dimension qsm(ko,n)
c
c   zoogro subroutine calculates simulated food inventories from feed
c   and growing crops available to zoons and humans as specified
c
      if(ai1.gt.1)goto70
      if(aiii.gt.1)goto70
c ** zoon loop ************************************************************
      do 60 l=1,ko
c
c ** contaminant loop *****************************************************
      do 20 k=1,inz
c contaminant parameters initialized (defined below where used)
      w(18,l,k)=0.0
      w(19,l,k)=0.0
      w(20,l,k)=0.0
c
c ** regional loop ********************************************************
      do 10 lz=1,ipx
c contaminant concentrations in and on feed consumed by zoon, per gdwt,
c initialized; humus biomass initialized
      spac(l,k,lz)=0.0
      spar(l,k,lz)=0.0
      humus(l,k,lz)=0.0
      sact(l,k,lz)=0.0
      sair(l,k,lz)=0.0
   10 continue
c ** end region loop ******************************************************
   20 continue
c ** end contaminant loop **************************************************
```

```
c
c ** regional loop ********************************************************
      do 50 lz=1,ipx
c
c ** plant loop ***********************************************************
      do 40 j=1,ipl
c harvest storage, kg-dwt, of plant(J) from region(LZ) for zoon(L)
      hrvst(l,j,lz)=0.0
c
c ** contaminant loop *****************************************************
      do 30 k=1.inz
c contaminant(K) concentration in and on plant(J) parts in region(LZ)
c initialized, per gdwt, in supplementary feed
      spas(l,j,k,lz)=0.0
      spaa(l,j,k,lz)=0.0
   30 continue
c ** end contaminant loop *************************************************
   40 continue
c ** end plant loop ********************************************************
   50 continue
c ** end of region loop ****************************************************
   60 continue
c ** end of zoon loop ******************************************************
      return
   70 continue
c enters only once per simulation day
      if (lx.ne.1) return
c
c ** zoon loop ************************************************************
      nto=nan+nhm
c total area utilized by zoon type(L) initialized
      do 680 in=1,nto
      if(rr(in).le.0.0) return
      agz(in)=0.0
c zoon id
      aiin=in
      sptrt=0.0
      sptft=0.0
c ** regional loop ********************************************************
      do 55 lz=1,ipx
c initializes frac. of food consumed from lz for range and feed plants
      sptfrr(lz)=0.0
      sptfrf(lz)=0.0
c
c ** plant loop ***********************************************************
      areamx=0.0
      do 45 j=1,ipl
c area used set by area of first plant food preference; AGR is total area, m2
c utilized by zoon(l) in region(lz) from plant(j) in that region
c
c sets maximum acreage for zoon grazing; default area for range
c zoon, m2, (144 sections)
      if(putzd(in,j,lz).le.0.0)goto45
      if(putzd(in,j,lz).gt.3.0)sptfrf(lz)=sptfrf(lz)+putfrc(in,j,lz)
```

```
      if(putzd(in,j,lz).gt.3.0)sptft=sptft+putfrc(in,j,lz)
      if(putzd(in,j,lz)-3.0) 41,41,45
   41 areamp=acres(j,lz)*4047.
      sptfrr(lz)=sptfrr(lz)+putfrc(in,j,lz)
      sptrt=sptrt+putfrc(in,j,lz)
      if(acres(j,lz).le.0.0)areamp=3.73e08
      areamx=areamx+areamp
c acreage of plants under niche competition should add up to single area
      if(rscde(lz).gt.0.0)agr(in,lz)=areamp
      if(rscde(lz).le.0.0)agr(in,lz)=areamx
   45 continue
c ************** end of plant loop ************************************
c
c total area utilized from all regions, m2, as a function of regional
c use fraction(RFRZ); bounded
      if(agr(in,lz).le.0.0)agr(in,lz)=3.73e08
      agz(in)=agz(in)+agr(in,lz)*rfrz(in,lz)
   55 continue
c ************** end of region loop ********************************
c
      if (agz(in).le.0.0) agz(in)=3.73e08
c
c plant harvest biomass, grams, from dry feed fractions, gdwt/m2
      do 681lz=1,ipx
      do 681l=1,ipl
      if (putzd(in,l,lz).le.3.0)goto 681
      ixx=ica(l,lz)
      if (cuta(l,ixx,lz)-(attt(l,lz)+3.0)) 681,682,681
c total live and dry plant biomass, g/m2
  682 hrvst(in,l,lz)=0.0
      b1sd=b1(l,lz)+sd(l,lz)+sd1(l,lz)+wd1(l,lz)+d1(l,lz)
c limits harvest to biomass densities above 20 g/m2
      if(b1sd.le.20.)goto681
      b1sdt=b1sd
c designates fruits and dry fruits(5,7), and above ground biomass(6) less
c fruits(seeds)
      if (putzd(in,l,lz)-5.0) 272,271,272
c seeds and grains
  271 b1sd=sd(l,lz)+sd1(l,lz)
  272 if (putzd(in,l,lz)-6.0) 274,273,274
c green and dry forage
  273 b1sd=b1(l,lz)+d1(l,lz)+wd1(l,lz)
  274 if (putzd(in,l,lz)-7.0) 276,275,276
c seeds and grains
  275 b1sd=sd(l,lz)+sd1(l,lz)
c estimates harvest of plant(L), grams, fed to zoon(K)
  276 hrvst(in,l,lz)=(b1sd-20.*b1sd/b1sdt)*4047.*acres(l,lz)
      if(hrvst(in,l,lz).le.0.0)hrvst(in,l,lz)=0.0
  681 continue
      sumput(in)=0.0
c initializes green plant fraction and nitrogen fraction in feed
      grnas(in)=0.0
      antros(in)=0.0
      do 683lz=1,ipx
```

```
      do 683l=1,ipl
      if(putzd(in,l,lz).le.3.0)goto683
      if(putfrc(in,l,lz).le.0.0)goto683
c
c ** contaminant loop ******************************************************
      do 160 j=1,inz
c contaminant concentration on(SPAA), and on and in(SPAS) plant(l) initialized,
c per gdwt for zoon(k),contaminant(j), and region(lz) in supplementary feed
      spaa(in,l,j,lz)=0.0
      spas(in,l,j,lz)=0.0
  160 continue
c ** end contaminant loop ***************************************************
      if(hrvst(in,l,lz).le.0.0)goto683
      iputz=putzd(in,l,lz)-3.0
      go to (171,181,191,201), iputz
  171 b1sd=b1(l,lz)+sd(l,lz)+sd1(l,lz)+wd1(l,lz)+d1(l,lz)
      d1l=d1(l,lz)*wnd(l,lz)/b1(l,lz)
      dgg=(b1(l,lz)-wnd(l,lz))*wnd(l,lz)/b1(l,lz)
c estimates weighted fraction nitrogen due to all plants(l) in feed
      antros(in)=antros(in)+spas(in,l,2,lz)*putfrc(in,l,lz)
c estimates weighted green or live plant fraction in feed due to all
c plants(l)
      grnas(in)=grnas(in)+(b1(l,lz)+sd(l,lz)+sd1(l,lz))*putfrc(in,l,lz)
     1 /b1sd
      sumput(in)=sumput(in)+putfrc(in,l,lz)
      goto 211
  181 b1sd=sd(l,lz)+sd1(l,lz)
c estimates weighted nitrogen fraction in seeds
      antros(in)=antros(in)+spas(in,l,2,lz)*putfrc(in,l,lz)
c increments weighted green plant fraction due to fruits
      grnas(in)=grnas(in)+putfrc(in,l,lz)
      sumput(in)=sumput(in)+putfrc(in,l,lz)
      goto 211
  191 b1sd=b1(l,lz)+wd1(l,lz)+d1(l,lz)
      d1l=d1(l,lz)*wnd(l,lz)/b1(l,lz)
      dgg=(b1(l,lz)-wnd(l,lz))*wnd(l,lz)/b1(l,lz)
c estimates weighted fraction nitrogen contribution for plant parts included
      antros(in)=antros(in)+spas(in,l,2,lz)*putfrc(in,l,lz)
c estimates weighted green plant fraction  for plant parts included
      grnas(in)=grnas(in)+(b1(l,lz))*putfrc(in,l,lz)/b1sd
      sumput(in)=sumput(in)+putfrc(in,l,lz)
      goto 211
  201 b1sd=sd(l,lz)+sd1(l,lz)
c estimates weighted percent nitrogen contribution for plant parts included
      antros(in)=antros(in)+spas(in,l,2,lz)*putfrc(in,l,lz)
c estimates weighted green plant fraction for plant parts included
      grnas(in)=grnas(in)+putfrc(in,l,lz)
      sumput(in)=sumput(in)+putfrc(in,l,lz)
c
c ** contaminant loop ******************************************************
c
  211 do 210 j=1,inz
c feed selector
      go to (170,180,190,200), iputz
```

```
c estimates contaminant concentration, per gdwt-plant, for total activity in
c and on plant that is consumed
 170 zza=d11*(z4(l,j,lz)+z42(l,j,lz)+z43(l,j,lz))/d1(l,lz)+(wz4(l,j,lz)
    1+wz42(l,j,lz)+wz43(l,j,lz))
     b1sd=d11+wd1(l,lz)
     zza=zza+(wz3(l,j,lz)+a2(l,j,lz)+a3av(l,j,lz))
     b1sd=b1sd+wnd(l,lz)
     zza=zza+dgg*(z3(l,j,lz)+b2(l,j,lz)+b3(l,j,lz))/b1(l,lz)
     b1sd=b1sd+dgg
     zza=zza+(zsd(l,j,lz)+afr(l,j,lz)+afsav(l,j,lz))
     b1sd=b1sd+sd(l,lz)+sd1(l,lz)
     spas(in,l,j,lz)=zza/b1sd
c
c estimates surficial contaminant concentration, per gdwt-plant for live and
c dead leaves and live fruits adjusted total plant biomass, per gdwt-plant
     b1sd=wnd(l,lz)+wd1(l,lz)+sd(l,lz)
     spaa(in,l,j,lz)=(a2(l,j,lz)+a3av(l,j,lz)+afr(l,j,lz)+
    1afsav(l,j,lz)+wz42(l,j,lz)+wz43(l,j,lz))/b1sd
     go to 210
c estimates contaminant concentration in and on live and dry fruits(seeds),
c per gdwt-seeds that are not sheathed
 180 zza=(zsd(l,j,lz)+afr(l,j,lz)+afsav(l,j,lz))
     b1sd=sd(l,lz)
     spas(in,l,j,lz)=zza/b1sd
c estimates surficial contamination on seeds, per gdwt-seed
     spaa(in,l,j,lz)=(afr(l,j,lz)+afs(l,j,lz))/sd(l,lz)
     go to 210
c estimates contaminant concentration in and on above ground biomass less live
c and dead fruits, per gdwt-plant
 190 zza=d11*(z4(l,j,lz)+z42(l,j,lz)+z43(l,j,lz))/d1(l,lz)+(wz4(l,j,lz)
    1+wz42(l,j,lz)+wz43(l,j,lz))
     b1sd=d11+wd1(l,lz)
     zza=zza+(wz3(l,j,lz)+a2(l,j,lz)+a3av(l,j,k))
     b1sd=b1sd+wnd(l,lz)
     zza=zza+dgg*(z3(l,j,k)+b2(l,j,k)+b3(l,j,k))/b1(l,lz)
     b1sd=b1sd+dgg
     spas(in,l,j,lz)=zza/b1sd
c estimates contaminant surficial contamination on dead and live leaves
     spaa(in,l,j,lz)=(a2(l,j,lz)+a3av(l,j,lz)+wz42(l,j,lz)+
    1wz43(l,j,lz))/(wnd(l,lz)+wd1(l,lz))
     go to 210
c estimates contaminant concentration in and on live and dry fruits(seeds)
c that are sheathed during development..corn..etc, set to 1/10
 200 zza=(zsd(l,j,lz)+afr(l,j,lz)+afsav(l,j,lz))
     b1sd=sd1(l,lz)
     spas(in,l,j,lz)=0.1*zza/b1sd
c estimates contaminant surficial contamination, per gdwt, on live and dry
c fruits
     spaa(in,l,j,lz)=((afr(l,j,lz)+afsav(l,j,lz))/sd(l,lz))
 210 continue
c ** end contaminant loop ***************************************************
 683 continue
c ** end region and plant loop nest ************************************
c
```

```
      if(sumput(in).le.0.0)goto 684
      grnas(in)=grnas(in)/sumput(in)
      antros(in)=antros(in)/sumput(in)
c
c number of yearly feeding intervals for zoon(IN) times 2
  684 iiet=eatin(in)*2.0
c
c ** yearly feeding interval loop (uses upper interval value..2,4,.******
      do 300 l=2,iiet,2
c last year number in interval
      ieat=l
c selects year numbers within first and last year of interval
      if (ai1.ge.yrfed(in,l-1).and.ai1.le.yrfed(in,l)) go to 310
  300 continue
c ** end yearly feeding intervals ************************************
c exits ZOOGRO if not on feed
      go to 680
  310 continue
c initializes human food index
      fedh=0.0
c initializes upper day index number in daily feeding interval
      aeat=0.0
c
c ** year interval feed loop for given upper interval(IEAT) **********
      do 320 j=2,ieat,2
      jkk=j/2
c upper day number index in feed schedule(last day)
      aeat=aeat+eatrec(in,jkk)*2.0
c lower day number index in feed schedule(last day)
      beat=aeat-eatrec(in,jkk)*2.0+2.0
  320 continue
c ** end interval feed loop ******************************************
c lower and upper day number indicies
      meat=beat
      jeat=aeat
c
c ** regional loop ***************************************************
      do 360 lz=1,ipx
c
c ** daily feed interval loop ****************************************
      do 350 j=meat,jeat,2
c preserves former food selection ,utilization, and ranking in interval
      if (onfod(in,j)) 351,350,351
  351 keat=j
c
c resets plants used for food and ranking food preference
c
c selects specific feeding interval (first day) only
      if (aiii-onfod(in,keat-1)) 350,353,350
c food schedule sequence index
  353 leat=pltind(in,keat/2,lz)
c resets fraction of food obtained on range
      rfrz(in,lz)=rnfrc(in,keat-1,lz)
c
```

```
c ** plant loop ******************************************************
      do 340 l=1,ipl
c initializes fraction of feed(L) in diet of zoon(IN)
      putfrc(in,l,lz)=0.0
c
c ** food selection loop *********************************************
      do 330 k=1,8
c excludes null plant index numbers
      if (pltusd(in,k,leat,lz)) 331,330,331
c selects specific plant type to be used in new feed schedule
  331 pusd=pltusd(in,k,leat,lz)
c specific plant identification
      pltd=pltid(l,lz)
c excludes all but matching set from plant selection
      if (pusd-pltd) 330,332,330
c resets feed type(ranking) and utilization fraction
  332 ppfrc=pltf(in,k,leat,lz)
      putfrc(in,l,lz)=ppfrc
  330 continue
c ** end food selection loop *****************************************
  340 continue
c ** end plant loop ******************************************************
  350 continue
c ** end daily feed interval *****************************************
  360 continue
c ** end region loop ******************************************************
c
c initializes and estimates range mean daily precipitation and mean daily
c temperature as a function of utilization fractions for each region
c simulated; (PAV, mm), (TTZ, deg. C.),respectively
c
      pav(in)=0.0
      ttz(in)=0.0
c
c ** region loop *******************************************************
      do 80 lzz=1,ipx
      pav(in)=pav(in)+p(lzz)*rfrz(in,lzz)
      ttz(in)=ttz(in)+t(lzz)*rfrz(in,lzz)
   80 continue
c ** end region loop ******************************************************
c overall range plant utilization fraction  for zoon(IN)
      rfrac(in)=0.0
c
c ** region loop ********************************************************
      do 100 lz=1,ipx
c
c ** plant loop *********************************************************
      do 90 jj=1,ipl
c excludes non-utilized and non-range plants
      if (putzd(in,jj,lz)) 90,90,91
   91 if (putzd(in,jj,lz).gt.3.0) goto90
c total fraction of diet obtained from grazing on range calculated from input
      rfrac(in)=rfrac(in)+putfrc(in,jj,lz)
   90 continue
```

```
c ** end plant loop ******************************************************
  100 continue
c ** end region loop ******************************************************
c
c ** region loop ***********************************************************
      sptat=0.0
      do 120 lz=1,ipx
c initiallizes fraction of animal diet from region lz
      sptfra(lz)=0.0
c ** zoon loop ***************************************************************
      do 110 jj=1,nan
c excludes strict herbivores and humans as range predators
      if (autzd(in,jj,lz)) 111,110,111
c fraction of diet from range is incremented for animal consumption
  111 rfrac(in)=rfrac(in)+autzd(in,jj,lz)
      sptfra(lz)=sptfra(lz)+autzd(in,jj,lz)
      sptat=sptat+autzd(in,jj,lz)
  110 continue
c *** end zoon loop ***************************************************
  120 continue
c *** end region loop ************************************************
c
c total zoon group intake rate, kg/day, adjusted for total number of zoons
c in age group(j), and total zoon population(ZOO)
c
      iage=agea(in)
      otot(in)=0.0
      do 130 j=1,iage
      xins=(a(j,in,1)*pop(1,in)+a(j,in,2)*pop(2,in))/zoo(in)
      otot(in)=otot(in)+fdin(j,in)*xins*zoo(in)
  130 continue
c *** end zoon loop ****************************************************
c
c estimates non-range food intake rate, kg/day, for zoon group(IN);bounded
      otots(in)=otot(in)*(1.-rfrac(in))
c estimates food intake rate from range, kg/day, for zoon group(IN);bounded
      otot(in)=otot(in)*rfrac(in)
      if (otot(in).le.0.0) otot(in)=0.0
      if (otots(in).le.0.0) otots(in)=0.0
c initializes available food for consumption(OTOTBS),kg, from non-range
c (feeds); initializes food type counter(OTTSM);and fraction of diet counter
c (SUMPUT)
      ototbs(in)=0.0
      ottsm(in)=0.0
  240 sumput(in)=0.0
c starts food type counter; limited to 10 cycles of feed supply
      ottsm(in)=ottsm(in)+1.0
      if (ottsm(in).ge.10.) go to 290
c ** region loop *******************************************************
      do 260 lz=1,ipx
c
c ** plant loop ********************************************************
      do 250 jj=1,ipl
c excludes range plants(graze)
```

```
      if (putzd(in,jj,lz).le.3.0) go to 250
c excludes low harvests
      if (hrvst(in,jj,lz).le.20.) go to 250
c starts diet fraction accumulator
      sumput(in)=sumput(in)+putfrc(in,jj,lz)
  250 continue
c ** end plant loop ***************************************************
  260 continue
c ** end region loop **************************************************
c excludes null diet fraction (SUMPUT) and  null non-graze food(OTOTS)
      if (sumput(in)) 291,290,291
  291 if (otots(in)) 290,290,292
c
c ** region loop ******************************************************
  292 do 280 lz=1,ipx
c ** plant loop *******************************************************
      do 270 jj=1,ipl
c excludes humans
      if (in.gt.nan) go to 270
c excludes range plants(graze)
      if (putzd(in,jj,lz).le.3.0) go to 270
c excludes low harvests
      if (hrvst(in,jj,lz).le.20.) go to 270
c estimates plant(JJ) feed contribution,kg, of total feed consumed; bounds
      ototx=otots(in)*putfrc(in,jj,lz)/sumput(in)
      if (ototx.ge.hrvst(in,jj,lz)/1000.) ototx=hrvst(in,jj,lz)/1000.
c adjusts harvest stores of plant(JJ) feed;bounds
      hrvst(in,jj,lz)=hrvst(in,jj,lz)-ototx*1000.
      if (hrvst(in,jj,lz).le.0.0) hrvst(in,jj,lz)=0.0
c sums total feed amount,kg, all plants(feeds) in diet
      ototbs(in)=ototbs(in)+ototx
  270 continue
c ** end plant loop ***************************************************
  280 continue
c ** end region loop **************************************************
c recycles to available feeds if feed requirement(OTOTS) not satisfied
      if (ototbs(in).lt.otots(in)) go to 240
  290 continue
c
c ** regional loop ****************************************************
      do 380 lz=1,ipx
c **** set food prefrerence ratios for regions
      if(sptrt.gt.0.0)sptfrr(lz)=sptfrr(lz)/sptrt
      if(sptft.gt.0.0)sptfrf(lz)=sptfrf(lz)/sptft
      if(sptat.gt.0.0)sptfra(lz)=sptfra(lz)/sptat
c
c ** plant loop *******************************************************
      do 370 l=1,ipl
c excludes unutilized plants
      if (putzd(in,l,lz)) 371,370,371
c increments human index number (males and females)
  371 if (in.gt.nan.and.putzd(in,l,lz).gt.3.0) fedh=fedh+1.0
  370 continue
c ** end plant loop ***************************************************
```

```
c
 380 continue
c ** end region loop ****************************************************
c
c* excludes humans
    if (in.gt.nan) go to 640
c
c ** regional loop ******************************************************
    do 400 lz=1,ipx
c
c ** plant loop *********************************************************
    do 390 l=1,ipl
c excludes non-range plants(graze) and un-utilized plants
    if (putzd(in,l,lz).le.0.0) go to 390
    if (putzd(in,l,lz).gt.3.0) go to 390
c initializes fraction of green (live) (G), and dry(D) biomass ingested
    g(l,lz)=0.0
    d(l,lz)=0.0
c green(live) biomass factor, g/m2, leaves, shoots, seeds
c assumes that etible stems(shoots) equals mass of leaves at maximum
    stems=(b1(l,lz)-wnd(l,lz))*wnd(l,lz)/b1(l,lz)
    if(stems.le.0.0)stems=0.0
    p2=(wnd(l,lz)+sd(l,lz)+sd1(l,lz)+stems)
c dry biomass factor, g/m2
    p3=wd1(l,lz)+d1(l,lz)*wnd(l,lz)/b1(l,lz)
c fraction of green biomass ingested (assumes 3x ratio green over dry)
    g(l,lz)=3.0*p2/(3.0*p2+p3)
c fraction of dry standing biomass ingested
    d(l,lz)=1.0-g(l,lz)
c assumes that etible stems equals mass of leaves at maximum
    stemz=z3(l,2,lz)*stems/b1(l,lz)
c estimates percent nitrogen in green(live) biomass
    snit(l,lz)=100.*(wz3(l,2,lz)+zsd(l,2,lz)+stemz)/(wnd(l,lz)+sd(l,lz
   1 )+stems)
c calculates percent nitrogen in consumed food assuming dry feed contains one
c half live plant nitrogen content
    cnit(l,lz)=g(l,lz)*snit(l,lz)+d(l,lz)*snit(l,lz)/2.
c estimates percentage of water in forage
    water(l,lz)=30.0*cnit(l,lz)
 390 continue
c ** end plant loop *****************************************************
c
 400 continue
c ** end region loop ****************************************************
c
c ** regional loop ******************************************************
    do 430 lz=1,ipx
c
c ** food category loop *************************************************
c
    do 420 l=1,7
c total biomass available as food,g/m2, for seven food categories:
c 1,2,3 which are grazed plants, 4-7 are for supplementary  feed
c and animal prey
```

```
      qsum(l,lz)=0.0
      qsm(l,lz)=qsum(l,lz)
c green content of forage, gdwt/gdwt plant, initialized
      gtdr(l,lz)=0.0
c total nitrogen in food, g/gdwt, initialized
      tnit(l,lz)=0.0
c food availability, kgdwt/d, food group(L) initialized
      up(l,lz)=0.0
c fraction consumed initialized
      qfrac(l,lz)=0.0
c
c ** contaminant loop ******************************************************
      do 410 j=1,inz
c contaminant concentration in (SACT), and on surface(SAIR) of plant, per gdwt
c initialized
      sact(l,j,lz)=0.0
      sair(l,j,lz)=0.0
  410 continue
c ** end contaminant loop ***************************************************
  420 continue
c ** end plant loop *********************************************************
  430 continue
c ** end region loop ********************************************************
c
c ** regional loop ***********************************************************
c ** initializes total range area, acres
      sacre=0.0
c ** initializes range soil moisture potential, bars
      phitt(3)=0.0
      do 470 lz=1,ipx
c
c ** plant loop ************************************************************
      do 460 l=1,ipl
c excludes non-utilized plants
      if (putzd(in,l,lz)) 461,460,461
  461 k=putzd(in,l,lz)
c excludes non-range (forage) plants and prey
      go to (440,440,440,460,460,460,460), k
  440 continue
c ** estimates soil moisture potential, bars, for range soils
      phitt(3)=phitt(3)+phi(l,2,lz)*acres(l,lz)
      sacre=sacre+acres(l,lz)
c estimates individual plant availabilty of both green and dry biomass,
c g/m2
      stems=(b1(l,lz)-wnd(l,lz))*wnd(l,lz)/b1(l,lz)
      if(stems.le.0.0)stems=0.0
      q10(l,lz)=(wnd(l,lz)+sd(l,lz)+sd1(l,lz)+stems)*g(l,lz)+
     1d(l,lz)*(wd1(l,lz)+d1(l,lz))
c total biomass availabilty from all plants utilized estimated, gdwt/m2
      qsum(k,lz)=qsum(k,lz)+q10(l,lz)
      qsm(k,lz)=qsum(k,lz)
c temporary variable, green biomass fraction in forage
      gtdr(k,lz)=gtdr(k,lz)+g(l,lz)*q10(l,lz)
c temporary variable, nitrogen content in consumed range plants
```

```
      tnit(k,lz)=tnit(k,lz)+cnit(l,lz)*q10(l,lz)/100.
c etible dry stem biomass, g/m2
      ddd=d1(l,lz)*wnd(l,lz)/b1(l,lz)
c etible live stem biomass
      dgg=(b1(l,lz)-wnd(l,lz))*wnd(l,lz)/b1(l,lz)
c
c ** contaminant loop ******************************************************
      do 450 j=1,inz
c temporary variable, biomass weighted live leaf contaminant conc.
      wndz=(wz3(l,j,lz)+a2(l,j,lz)+a3av(l,j,lz))
c temporary variable, biomass weighted live and dry seeds(fruits)
      wsdz=(zsd(l,j,k)+afr(l,j,k)+afsav(l,j,lz))
c temporary variable, biomass weighted live stem contaminant conc.
      wbdz=(z3(l,j,k)+b2(l,j,k)+b3(l,j,k))*dgg/b1(l,lz)
c temporary variable, biomass weighted dry leaf contaminant conc.
      wndd=(wz4(l,j,lz)+wz42(l,j,lz)+wz43(l,j,lz))
c temporary variable, biomass weighted dry stem contaminant conc.
      wbzz=(z4(l,j,lz)+z42(l,j,lz)+z43(l,j,lz))*ddd/d1(l,lz)
c total live biomass
      b1sg=wnd(l,lz)+sd(l,lz)+dgg
c total dry biomass
      b1sd=wd1(l,lz)+ddd
c temporary variable, contaminant concentration in and on consumed forage
      sactg=g(l,lz)*(wndz+wsdz+wbdz)/b1sg
      sactd=d(l,lz)*(wndd+wbzz)/b1sd
      sact(k,j,lz)=sact(k,j,lz)+g(l,lz)*(wndz+wsdz+wbdz)/b1sg
      sact(k,j,lz)=sact(k,j,lz)+d(l,lz)*(wndd+wbzz)/b1sd
c temporary variable, surficial contaminant contamination on live and dry
c leaves and fruits (seeds) in consumed forage
      airdep=(a2(l,j,lz)+a3av(l,j,lz))*g(l,lz)+(wz42(l,j,lz)+
     1wz43(l,j,lz))
      airdep=airdep+(afr(l,j,lz)+afs(l,j,lz))*(sd(l,lz))*g(l,lz)
c temporary variable, cumulative sum of above
      b1sd=wnd(l,lz)+wd1(l,lz)+sd(l,lz)
      sair(k,j,lz)=sair(k,j,lz)+airdep/b1sd
  450 continue
c ** end contaminant loop ***************************************************
  460 continue
c ** end plant loop *********************************************************
  470 continue
c ** end region loop ********************************************************
      if(sacre.gt.0.0)phitt(3)=phitt(3)/sacre
      if(sacre.le.0.0)phitt(3)=15.
      if(phitt(3).gt.15.)phitt(3)=15.
c total food available for consumption for zoon group (IN)
      ototb(in)=0.0
c
c ** region loop ***********************************************************
      do 570 lz=1,ipx
c daily food required from range by zoon(IN)
      otota(in)=ototo(in)
c excludes null intake requirement from range, kg/d
      if (otota(in).le.0.0) go to 550
c
```

```
c ** range plant food preference loop ************************************
      do 490 j=1,3
c excludes empty food preference  group
      if (qsum(j,lz).le.1.0) go to 490
c
c ** contaminant loop ********************************************************
c     do 480 k=1,inz
c contaminant concentration in and on plant, per gdwt
c     sact(j,k,lz)=sact(j,k,lz)/qsum(j,lz)**2
c contaminant concentration on live and dead leaves and fruits, per gdwt
c     sair(j,k,lz)=sair(j,k,lz)/qsum(j,lz)**2
c  480 continue
c ** end contaminant loop ****************************************************
c nitrogen content of plant , gN/gdwt plant
      tnit(j,lz)=tnit(j,lz)/qsum(j,lz)
c green fraction of forage, gdwt/gdwt
      gtdr(j,lz)=gtdr(j,lz)/qsum(j,lz)
  490 continue
c ** end range food preference loop **************************************
c initializes total animal tissue utilization fraction from all sources
      cnsm=0.0
c animal sources index initialized
      ij=0
c percent nitrogen in food initialized
      antro(in)=0.0
c
c **zoon loop ************************************************************
      do 510 j=1,nan
c excludes non-utilized animals
      if (autzd(in,j,lz)) 510,510,511
c animal source index incremented
  511 ij=ij+1
c animal food tissue category initialized for each animal type consumed, g/m2
      qsum(ij+3,lz)=0.0
      qsm(ij+3,lz)=qsum(ij+3,lz)
c oldest animal age in zoon group(J)
      kage=agea(j)
c
c ** zoon age group loop *************************************************
c
      do 500 ll=1,kage
c estimates animal tissue available for zoons in population
c being consumed, gfwt/m2, adjusts for population losses from original
      xins=(a(ll,j,1)*pop(1,j)+a(ll,j,2)*pop(2,j))/zoo(j)
      tpopc=xins*zoo(j)
      tpopr=tpopi(ll,1,j)+tpopi(ll,2,j)
      tpopf=tpopc/tpopr
      if(tpopf.ge.0.0)tpopf=1.0
      if(alsp(j).gt.0.0)xins=xins*tpopf
      if(alsp(j).le.0.0.and.ainsm.lt.1.0)xins=xins*ainsm
      qsum(ij+3,lz)=qsum(ij+3,lz)+wt(ll,j)*xins*zoo(j)*1000.
     1/agr(j,lz)
      ij3=ij+3
      qsm(ij+3,lz)=qsum(ij+3,lz)
```

```
 500 continue
c
c ** end zoon age loop ***************************************************
c with exception of insects, assumes that only 1/2 of available
c animal tissue can be utilized
c
    if (alsp(j)) 502,501,502
c
 502 qsum(ij+3,lz)=0.5*qsum(ij+3,lz)
 501 qsm(ij+3,lz)=qsum(ij+3,lz)
c
c estimates fraction of zoon in diet from utilization factor(AUTZD) and
c fraction of time consumer spends in region(LZ)
c
    qfrac(ij+3,lz)=autzd(in,j,lz)*rfrz(in,lz)
c
c estimates utilization fraction from all animal tissue consumption
c
    cnsm=cnsm+qfrac(ij+3,lz)
 510 continue
c ** end zoon loop ***************************************************
c adjusts plant range preferences classes(1-3) for animal consumption
c
    qfrac(1,lz)=0.7*(1.-cnsm)*rfrz(in,lz)
    qfrac(2,lz)=0.2*(1.-cnsm)*rfrz(in,lz)
    qfrac(3,lz)=0.1*(1.-cnsm)*rfrz(in,lz)
c total food available all sources, kg/day, initialized
    sototz(lz)=0.0
c food cycling index initialized
    sosum=0.0
c total food utilization fraction initialized each cycle
 520 fsum=0.0
c increments food cycling index
    sosum=sosum+1.0
c limits cycling to 9 passes
    if (sosum.ge.10.) go to 550
c
c ** food category loop ***********************************************
    do 530 j=1,7
c excludes empty food categories; limits vegetation to 2 g/m2 and
c animal tissue to .00001 g/m2
c
    bmlim=0.00001
    if(alsp(in).le.0.0)bmlim=0.02
    if(j.le.3.and.qsm(j,lz).le.2.0) qfrac(j,lz)=0.0
    if(j.ge.4.and.qsm(j,lz).le.bmlim) qfrac(j,lz)=0.0
c
c sums utilization fractions
 530 fsum=fsum+qfrac(j,lz)
c ** end food category loop *******************************************
c excludes null fraction total
    if (fsum) 551,550,551
c initializes forage food categories available
 551 qfrct=0.0
```

```
c
c ** forage food loop ***************************************************
      do 540 j=1,7
c excludes null utilization fraction
      if (qfrac(j,lz)) 541,540,541
c sums available groups
  541 qfrct=qfrct+1.0
c adjusts utilization fraction for fraction of range on region(LZ),and for
c total utilization fraction(FSUM)
      qfrac(j,lz)=qfrac(j,lz)*rfrz(in,lz)/fsum
      afrac=qfrac(j,lz)
c estimate portion of required intake(OTOT), kg/d, from food category(J) and
c region(LZ)
c estimates ratio of available food in category to that required
      if(j.le.3)qsmf=qsm(j,lz)-2.0
      bmlim=0.00001
      if(alsp(in).le.0.0)bmlim=0.02
      if(j.gt.3)qsmf=qsm(j,lz)-bmlim
      if(qsmf.le.0.0)qsmf=0.0
      tfrac=(qsmf*agr(in,lz)/1000.)/(otota(in)*qfrac(j,lz))
      if(j.le.3.and.sptfrr(lz).gt.0.0)tfrac=tfrac/sptfrr(lz)
      if(j.gt.3.and.sptfra(lz).gt.0.0)tfrac=tfrac/sptfra(lz)
c bounds utilization fraction for category(J) if availability ratio(TFRAC) is
c less than 1.0
c
      if(tfrac.lt.1.0)afrac=afrac*tfrac
      fadj=sptfrr(lz)
      if(j.gt.3)fadj=sptfra(lz)
      up(j,lz)=up(j,lz)+otota(in)*afrac*fadj
c decrements available food availability for consumption losses; bounds
c
      qsm(j,lz)=qsm(j,lz)-fadj*otota(in)*afrac*1000.*aa2(in)/agr(in,lz)
      if (qsm(j,lz).le.0.0) qsm(j,lz)=0.0
c sums total food available from all categories
c
      sototz(lz)=sototz(lz)+otota(in)*afrac*fadj
  540 continue
c ** end forage food loop ***********************************************
c
c recycles if available food not equal to that required by zoon group(IN)
      fadjs=sptfrr(lz)+sptfra(lz)
      if (sototz(lz).lt.otota(in)*fadjs) go to 520
c
c sums all food available for consumption, kg/d, for zoon group(IN)
  550 ototb(in)=up(1,lz)+up(2,lz)+up(3,lz)+up(4,lz)+up(5,lz)+up(6,lz)+up
     1(7,lz)+ototb(in)
      if(ototb(in).le.0.0)ototb(in)=0.0
c
c ** contaminant loop ***************************************************
      do 560 k=1,inz
c initializes contaminant contamination in and on plants(SPAC), and on the
c surface of the plant(SPAR), per gdwt, for etible portions(except roots)
      spac(in,k,lz)=0.0
      spar(in,k,lz)=0.0
```

```
 560 continue
c ** end contaminant loop ************************************************
 570 continue
c ** end region loop ****************************************************
c green plant consumption initialized, kg/d
    grna(in)=0.0
c
c ** regional loop ******************************************************
    do 620 lz=1,ipx
c
c ** food category loop *************************************************
    do 610 k=1,7
c excludes empty food category and null food availability
    if (up(k,lz).le.0.0) go to 610
    if (ototb(in).le.0.0) go to 610
c excludes all but range food categories (1-3)
c
    if (k.gt.3) go to 590
c
c ** contaminant loop ***************************************************
    do 580 j=1,inz
c contaminant concentration in and on food(SPAC), and on dead and live fruits
c and seeds(SPAR)
    spac(in,j,lz)=spac(in,j,lz)+up(k,lz)*sact(k,j,lz)/ototb(in)
    spar(in,j,lz)=spar(in,j,lz)+up(k,lz)*sair(k,j,lz)/ototb(in)
 580 continue
c ** end contaminant loop ***********************************************
c nitrogen fraction in available combined food
c
    antro(in)=antro(in)+up(k,lz)*tnit(k,lz)/ototb(in)
c
c green(live) fraction in available combined food
    grna(in)=grna(in)+up(k,lz)*gtdr(k,lz)/ototb(in)
    go to 610
c oldest age in zoon group, yrs
 590 kage=agea(in)
c** animal loop
    do601j=1,nan
    if(autzd(in,j,lz).le.0.0)goto601
c
c ** zoon age loop ******************************************************
    do 600 ll=1,kage
c excludes unavailable food
    if (ototb(in).le.0.0) go to 600
c nitrogen fraction in diet from animal tissue
    xins=(a(ll,j,1)*pop(1,j)+a(ll,j,2)*pop(2,j))/zoo(j)
    antro(in)=antro(in)+up(k,lz)*xins*(vms(ll,j,3)*autzd(in,j,lz)/
   1(0.45*wt(j,in)*ototb(in)))
 600 continue
c ** end zoon age loop **************************************************
 601 continue
c ** end animal loop ******************************************************
c assumes that grna=1.0 for consumed animal tissue
    grna(in)=grna(in)+up(k,lz)/ototb(in)
```

```
 610 continue
c ** end food category loop ***********************************************
 620 continue
c ** end region loop ****************************************************
c estimates combined nitrogen fraction (forage,feed,and animal) food
c
    sumfor=ototbs(in)+ototb(in)
    if(sumfor.le.0.0)print *,'animal',in,'  is out of food'
    if(sumfor.le.0.0)call exit
    antro(in)=(antro(in)*ototb(in)+antros(in)*ototbs(in))/sumfor
    if(antro(in).ge.1.0)antro(in)=1.0
    if(antro(in).le.0.0)antro(in)=0.0
c
c estimates combined green(live) fraction (forage,feed,animal) food
c
    grna(in)=(grna(in)*ototb(in)+grnas(in)*ototbs(in))/sumfor
    if (grna(in).ge.1.0) grna(in)=1.0
    if (grna(in).le.0.0) grna(in)=0.0
c
c sums total food required kg/day for all zoons in population(IN)
c
    otota(in)=otota(in)+otots(in)
    if(otota(in).le.0.0)otota(in)=0.0
c
c sums total food available, kg/day for all zoons in population(IN);bounds
c
    ototb(in)=ototb(in)+ototbs(in)
    if(ototb(in).le.0.0)ototb(in)=0.0
c
c ** animal and human age loop ******************************************
 640 do 660 ii=1,iage+1
c insects and animals excluded
    if (alsp(in)) 651,650,651
 651 if (in.le.nan) go to 650
c
c human model HUMTRN accessed
    call humtrn (in,lx,ii,fedh,bpx,rdidx)
    go to 660
c
c animal model accessed
 650 call animal (ii,in,lx,bpx,rdidx)
c
 660 continue
c ** end animal and human age loop ***************************************
 680 continue
c ** end zoon loop ******************************************************
    return
    end
```

14.18. BLOCK ARRAYS

```
      block data bloc1
c
c**** block data for BIOTRAN.2 MODEL
c
c**** i=number of plant types, ks=number of soil layers
c**** kt=number of tree diameters, kw=number of water layers
c**** ka= number of exposure-years for plants, kr=exposure age
c**** m=number of contaminants, n=number of regions, nq=number of plant
c****  organs for dosimetry, ko= number of zoons including humans
c**** ke=maximum age for animals
      parameter (i=8,ks=10,kt=40,kw=10,ka=200,kr=86,m=10,n=8,
     1nq=36, ko=10, ke=25)
c
      common/xb/ zmax(2,n),ztmn(2,n),ztx(2,n),celki(2,n),aiopt(5,n),
     1aksph(5,n),aksnt(5,n),akssi(5,n),umph(5,n),umnt(5,n),umsi(5,n),
     2akqph(5,n),akqnt(5,n),akqsi(5,n),zkqph(2,n),zkqnt(2,n),qmph(5,n),
     3qmsi(5,n),qmnt(5,n),zmnt(2,n),zmph(2,n),resp(5,n),sink(5,n),
     4cmax(5,n),topm(5,n),tulv(5,n),tllv(5,n),divd(5,n),exip(5,n),
     5y1(110,2),psep(22,n),bcp(22,n),strans(22,n),topt(22,n),tofl(22,n),
     6tofh(22,n),fleaf(22,n),ulef(22,n),respf(22,n),rspf2(22,n),
     7a(85,ko,2),brty(8,ko),brrt(8,ko,2),tqx(85,2,ko),nday(12),
     8tbbnz(100),tbkdz(100),fbwbz(100),
     9fmk(100),wutp(22,n),hf(22,n),sctp(22,n),astp(22,n),rxpp(22,n),
     1rfp(22,n),cbbnz(100),cbkdz(100),cblvz(100),cbwbz(100),tblvz(100),
     2tbwbz(100),difmol(110),henrc(110),akoc(110),zkd(110)
c
c **** C6H6=1 CCl4=2 CHCl3=3 HCClCClH=4 HCHO=5 Cl2CHCHCl2=6 PCBs=7 ****
c
c molecular diffusivity, cm2/sec
      data (difmol(j),j=1,110) /100*0.0,0.081995,9*0.0/
c henry's law constant, atm-m3/mol
      data (henrc(j),j=1,110) /100*0.0,5.59e-3,1.23e-2,2.87e-3,7.58e-3,
     19.87e-7,2.59e-2,1.07e-03,3*0.0/
c **** end ******
      data zmax(1,1), zmax(2,1), ztmn(1,1), ztmn(2,1), ztx(1,1), ztx(2,1
     1 ) /60.,30.,5.0,5.0,17.0,17.0/
      data (celki(j,1),j=1,2) /1.e02,1.e02/
      data (aiopt(j,1),j=1,5) /0.045,0.03,0.1,0.2,0.06/
      data (aksph(j,1),j=1,5) /1.0,2.0,1.0,3.0,0.6/
      data (aksnt(j,1),j=1,5) /2.0,5.0,1.5,5.0,0.75/
      data (akssi(j,1),j=1,5) /0.0,0.0,7.0,0.0,0.0/
      data (umph(j,1),j=1,5) /10.,1.,3.,10.,1./
      data (umnt(j,1),j=1,5) /30.,10.,20.,100.,10./
      data (umsi(j,1),j=1,5) /0.0,0.0,700.,0.0,0.0/
      data (akqph(j,1),j=1,5) /4.5,1.,2.,10.,0.5/
      data (akqnt(j,1),j=1,5) /90.,20.,40.,200.,10./
      data (akqsi(j,1),j=1,5) /0.0,0.0,2000.,0.0,0.0/
      data (zkqph(j,1),j=1,2) /37.2,18.6/
      data (zkqnt(j,1),j=1,2) /750.,375./
      data (qmph(j,1),j=1,5) /7.2,1.6,3.2,16.,0.8/
      data (qmsi(j,1),j=1,5) /0.0,0.0,1000.,0.0,0.0/
```

```
      data (qmnt(j,1),j=1,5) /72.,16.,32.,160.,8./
      data (zmnt(j,1),j=1,2) /600.,300./
      data (zmph(j,1),j=1,2) /60.,30./
      data (resp(j,1),j=1,5) /1.4,0.3,0.6,3.0,0.36/
      data (sink(j,1),j=1,5) /0.8,0.2,2.5,0.5,0.5/
      data (cmax(j,1),j=1,5) /7.,1.5,3.,15.,1.8/
      data (topm(j,1),j=1,5) /20.,20.,20.,20.,20./
      data (tulv(j,1),j=1,5) /35.,35.,35.,35.,35./
      data (tllv(j,1),j=1,5) /0.,0.,0.,0.,0./
      data (divd(j,1),j=1,5) /3.,2.5,3.,1.5,1.5/
      data (exip(j,1),j=1,5)/1.2,1.0,3.0,30.,0.3/
c
      data (y1(k,1),k=1,100) /9.7e04,5.e-06,2.5e-02,2.5e-03,2.5,1.1e05,7
     1 .5e03,7.6e05,1.3e-01,1.4e-04,3.3e02,8.e02,1.3e01,5.e01,8.8e02,5
     2 .e02,5.e02,6.e-01,5.e03,5.e02,7.5e-03,2.5e-01,1.3e-01,5.e-02,2.5e
     3 01,2.5e01,7.5e-02,7.5e-01,2.5,2.0e01,7.5e-03,1.e-01,5.e-02,1.3e-0
     4 1,3.8,3.e-04,1.3e01,5.,1.3e-01,5.e-02,5.e-02,2.5e-01,2.5e-14,5.e-
     5 05,1.3e-02,5.e-03,1.5e-02,1.5e-01,2.5e-02,2.5e-02,2.5e-02,1.3e-03
     6 ,1.e-01,1.e-04,5.e-02,2.5,1.e-01,1.3e-01,1.7e-02,6.e-02,2.5e-21,1
     7 .6e-02,2.5e-03,1.1e-02,1.8e-03,7.5e-03,1.8e-03,2.e-03,.0018,.002,
     8 1.8e-03,1.e-03,6.3e-03,1.8e-02,2.5e-04,5.e-05,1.3e-04,.005,2.5e-0
     9 4,3.8e-03,2.5e-02,.68,1.5e-02,1.e-11,2.5e-20,3.5e-13,1.e-20,2.5e-
     1 10,2.5e-13,2.5e-02,2.5e-10,2.5e-03,2.5e-15,2.5e-17,2.5e-34,2.5e-3
     1 3,2.5e-33,2.5e-33,2.5e-33,2.5e-33/
c
c C6H6=1 CCl4=2 CHCl3=3 HCClCClH=4 HCHO=5 Cl2CHCHCl2=6 PCB's=7..log(kow)
      data (y1(k,1),k=101,110)/2.12,2.64,1.97,0.7,0.0,2.6,6.04,3*0./
c
      data (y1(k,2),k=1,100) /2.e04,1.e-04,3.e+01,6.,20.,2.e04,1.e03,4.9
     1 e05,2.e02,1.e-03,6.3e03,6.3e03,7.14e04,3.3e05,8.e02,8.5e02,1.e02,
     2 1.,1.36e04,1.37e04,7.,4.6e03,100.,20.,850.,3.8e04,8.,40.,20.,50.,
     3 30.,1.,5.,.1,5.,1.e-04,100.,300.,50.,300.,5.3,2.,1.e-13,1.e-03,1
     4 .e-03,1.e-03,.1,.5,.1,10.,2.3,1.e-03,5.,1.e-05,5.,500.,40.,53.,6.
     5 7,25.,1.e-18,6.4,1.,4.2,.7,3.,.7,.8,.7,.8,.7,6.,1.,1.,1.e-03,1.e-
     6 03,1.e-05,1.e-02,1.e-01,1.e-02,.1,10.,.1,1.e-11,1.e-19,1.e-13,1.e
     7 -18,8.e-07,1.e-10,6.,1.e-07,1.,1.e-12,1.e-13,1.e-30,1.e-30,1.e-30
     8 ,1.e-30,1.e-30,1.e-30/
c
c C6H6=1 CCl4=2 CHCl3=3 HCClCClH=4 HCHO=5 Cl2CHCHCl2=6 PCB's=7
c
      data (akoc(j),j=1,110)/100*0.,83.,54.,31.,49.,3.6,364.,5.3e5,3*0./
c
      data (psep(j,1),j=1,22) /0.5,5*1.,2*0.5,8*1.,2*0.5,4*1./
c
      data (bcp(j,1),j=1,22) /3*.6,3*.4,2*.6,2*.4,3*.6,4*.4,2*.6,2*.4,.6
     1 /
c
      data (strans(j,1),j=1,22) /2*.75,.3,2*.85,.75,.5,.3,.75,.85,2*.5,.
     1 3,2*.85,.75,.85,.3,.5,.75,.85,.80/
c
      data (topt(j,1),j=1,22) /30.,2*25.,21.,25.,21.,2*30.,3*21.,2*18.,1
     1 4.,18.,15.,12.,2*18.,2*15.,18./
c
      data (tofl(j,1),j=1,22) /20.,2*15.,11.,25.,11.,2*20.,21.,2*11.,2*1
```

```
     1 0.,14.,18.,10.,12.,2*10.,15.,10.,11./
c
      data (tofh(j,1),j=1,22) /3*9.,13.,9.,13.,2*9.,3*13.,2*16.,20.,16.,
     1 19.,22.,2*16.,2*19.,16./
c
      data (fleaf(j,1),j=1,22) /3*0.,.25,.125,.25,2*0.,2*.25,3*0.,2*.125
     1 ,.25,.125,2*0.,2*.25,0./
c
      data (ulef(j,1),j=1,22) /3*0.,.4,.5,.4,2*0.,.5,.4,3*0.,2*.5,.4,.5,
     1 2*0.,.5,.4,0./
c
      data (respf(j,1),j=1,22) /3*.05,5.,2*5.,.1,.05,5.,5.,3*.05,4*5.,.0
     1 5,.10,9.,5.,.1/
c
      data (rspf2(j,1),j=1,22) /2*.05,8*.1,2*.05,10*.1/
c
      data (a(j,ko,1),j=1,85)/.01629,.01537,.01476,.01484,.01434,.01464,
     1.01431,.01489,.01545,.01708,.01708,.01662,.01623,.01664,.01720,.01
     2 830,.01913,.01923,.01915,.02006,.01982,.01955,.01932,.01920,.0186
     3 7,.01854,.01807,.01769,.01675,.01703,.01662,.01626,.01615,.01718,
     4 .01273,.01289,.01280,.01348,.01170,.01130,.01094,.01050,.01029,.0
     5 09881,.009845,.009684,.009424,.009558,.009433,.009845,.01007,.009
     6 890,.01002,.009962,.009980,.01004,.009818,.009648,.009639,.009379
     7 ,.008966,.008553,.008203,.007826,.007611,.007485,.007117,.006794,
     8 .006417,.006094,.005663,.005268,.004927,.004478,.004155,.003778,.
     9 003401,.003060,.002621,.002576,.002190,.001831,.001562,.001391,.0
     1 06336/
c
      data (a(j,ko,2),j=1,85)/.01471,.01388,.01334,.01339,.01298,.01322,
     1.01293,.01345,.01395,.01541,.01544,.01505,.01469,.01508,.01560,.01
     2 664,.01737,.01762,.01773,.01864,.01846,.01828,.01819,.01820,.0177
     3 3,.01762,.01722,.01687,.01600,.01636,.01598,.01569,.01560,.01664,
     4 .01239,.01255,.01250,.01318,.01147,.01110,.01076,.01035,.01015,.0
     5 09780,.009772,.009620,.009408,.009577,.009484,.009959,.01023,.010
     6 10,.01029,.01031,.01040,.01054,.01039,.01029,.01035,.01013,.00974
     7 7,.009374,.009077,.008780,.008678,.008687,.008415,.008186,.007881
     8 ,.007626,.007236,.006880,.006600,.006167,.005887,.005514,.005132,
     9 .004784,.004258,.004326,.003826,.003325,.002944,.002715,.01405/
c
      data (brty(j,ko),j=1,8) /14.,19.,24.,29.,34.,39.,44.,49./
c
      data (brrt(j,ko,1),j=1,8) /.000606,.0269,.0576,.0575,.0403,.0097,
     1.00198,.0001083/
c
      data (brrt(j,ko,2),j=1,8) /.000576,.0265,.0548,.0545,.0384,.0092,
     1.00198,.0001066/
c
      data (tqx(j,1,ko),j=1,85)/.02245,.00133,.00094,.00078,.00064,.0006
     1,.00054,.00051,.00046,.00041,.00036,.00035,.00042,.00059,.00084,.0
     2 0114,.00142,.00167,.00185,.00198,.00212,.00226,.00235,.00235,.002
     3 28,.00217,.00206,.00199,.00198,.00203,.00210,.00218,.00228,.00239
     4 ,.00252,.00268,.00288,.00312,.00339,.00369,.00401,.00435,.00473,.
     5 00518,.00568,.00623,.00681,.00744,.00812,.00887,.00969,.01059,.01
     6 161,.01275,.01400,.01534,.01676,.01827,.01987,.02158,.02339,.0253
```

```
     7 2..02738,.02960,.03200,.03463,.03746,.04044,.04350,.04665,.04991,
     8 .05344,.05740,.06193,.06703,.07264,.07856,.08462,.09070,.09688,.1
     9 0367,.11125,.11929,.12770,1.0/
c
      data (tqx(j,2,ko),j=1,85)/.01746,.00116,.00077,.00060,.00051,.0004
     1,.00038,.00034,.00031,.00028,.00026,.00025,.00027,.00033,.00040,.0
     2 0049,.00058,.00066,.00069,.00071,.00072,.00073,.00075,.00077,.000
     3 79,.00081,.00083,.00086,.00090,.00096,.00102,.00110,.00119,.00129
     4 ,.00140,.00152,.00165,.00180,.00197,.00215,.00233,.00251,.00273,.
     5 00297,.00325,.00354,.00384,.00416,.00449,.00484,.00523,.00565,.00
     6 611,.00660,.00712,.00768,.00829,.00894,.00962,.01035,.01113,.0120
     7 0,.01298,.01411,.01538,.01678,.01832,.02004,.02195,.02407,.02632,
     8 .02879,.03165,.03503,.03893,.04325,.04790,.05295,.05840,.06432,.0
     9 7097,.07834,.08612,.09419,1.0/
c
      data (fbwbz(k),k=1,100) /1.,1.,1.,2.e-3,9.e-9,1.,1.,1.,1.,1.,1.,1.
     1 ,1.,8.5e-1,7.5e-1,1.,1.,1.,1.,6.e-1,1.e-4,1.e-4,2.e-2,5.e-3,1.e-1
     2 ,1.e-1,3.e-1,3.e-1,2.8e-1,1.e-1,1.e-3,1.e-2,3.e-2,9.e-1,1.,1.,1.,
     3 3.e-1,1.e-4,1.e-4,5.e-2,8.e-1,1.,3.e-1,2.e-1,2.e-1,1.,2.5e-3,2.e-
     4 3,5.e-2,3.e-2,2.5e-1,1.,1.,1.,5.e-2,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4
     5 ,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,1.e-
     6 4,5.e-2,1.e-1,5.e-1,1.e-1,1.e-1,1.e-1,1.e-1,7.5e-1,4.5e-1,8.e-2,1
     7 .e-2,6.e-2,1.e0,1.,1.,3.e-1,1.e-4,1.e-4,1.e-4,1.e-4,1.e-4,3.e-5,1
     8 .e-4,1.e-4,3.e-5,3.e-5,3.e-5,3.e-5/
c
      data (nday(j),j=1,12) /30,27,30,29,30,29,2*30,29,30,29,30/
c
      data (tbbnz(k),k=1,100) /1.2e1,3.65e5,3.65e5,4.5e2,3.65e2,4.e1,9
     1 .e1,1.4e1,1.45e3,3.65e5,1.1e1,1.8e2,3.65e5,3.65e5,1.15e3,6.e2,2.9
     2 e1,3.65e5,5.8e1,1.8e4,1.6e3,3.65e5,1.35e2,3.65e5,3.6e2,1.68e3,2.4
     3 e1,8.e2,1.85e2,1.3e3,1.2e1,3.65e5,3.65e5,3.65e5,8.e0,3.65e5,8.,1.
     4 8e4,1.8e4,1.e3,1.e3,2.1e2,2.5e1,4.5e2,1.66e1,6.5e0,2.5e2,1.2e2,5.
     5 7e1,7.1e2,1.e2,3.e1,1.4e1,3.65e5,1.1e2,6.5e1,1.e3,1.5e3,1.5e3,1.5
     6 e3,1.5e3,1.5e3,1.5e3,1.e3,1.e3,1.e3,1.e3,1.e3,1.e3,1.e3,1.e3,6.e2
     7 ,3.e2,9.,3.5,3.65e5,3.65e5,3.65e5,3.65e5,8.00e0,3.6e1,2.60e3,1.33
     8 e1,2.4e1,3.65e5,3.65e5,3.65e5,1.64e4,7.3e4,7.3e4,7.3e4,2.5e3,7.3e
     9 4,7.3e4,7.3e4,7.3e4,7.3e4,7.3e4,7.3e4,7.3e4/
c
      data (tbkdz(k),k=1,100) /1.2e1,3.65e5,3.65e5,1.2e2,3.65e5,1.e1,90.
     1 ,1.4e1,3.65e5,3.65e5,1.1e1,3.65e5,3.65e5,3.65e5,1.8e1,3.65e5,2.9e
     2 1,3.65e5,5.8e1,1.7e1,1.6e3,3.65e5,7.4e1,6.16e2,2.5e1,7.2e2,2.4e1,
     3 3.65e5,9.e1,1.49e2,9.,1.2e1,5.5e2,1.1e1,8.e0,3.65e5,8.e1,1.6e1,4.
     4 2e3,9.e2,7.6e2,2.4e1,2.e1,1.8e2,2.8e1,3.e1,4.,3.e2,6.e1,6.6e2,3.6
     5 5e5,3.e1,7.,3.65e5,1.1e2,8.5,3.65e5,5.63e2,7.5e2,6.56e2,6.56e2,6.
     6 56e2,1.48e3,1.3e3,7.e2,3.65e5,8.e2,6.5e2,3.35e2,6.85e2,7.5e2,5.63
     7 e2,4.e2,3.65e5,3.65e5,5.,5.e1,6.e1,2.8e2,1.45e1,3.5e1,1.06e2,6.,7
     8 .e1,3.65e5,3.65e5,3.65e5,1.e1,2.4e4,2.2e4,5.1e4,3.0e1,6.4e4,3.2e4
     9 ,2.7e4,2.4e4,3.65e5,3.65e5,3.65e5,3.65e5/
c
      data (tblvz(k),k=1,100) /1.2e1,3.65e5,3.65e5,2.7e2,5.e-1,1.e1,9.e1
     1 ,1.4e1,3.65e5,3.65e5,1.1e1,3.65e5,3.65e5,3.65e5,3.4e1,3.65e5,2.9e
     2 1,3.65e5,5.8e1,8.,1.7e3,3.65e5,7.e1,3.65e5,7.1e1,5.54e2,2.4e1,5
     3 .e2,1.2e3,9.1e1,4.8,7.5,5.5e2,2.4e1,8.,3.65e5,8.e1,6.,1.4e3,3.2e2
     4 ,8.45e2,9.5e1,3.e1,7.e1,1.82e1,1.9e1,5.2e1,2.e2,5.8e1,1.6e3,3.8e1
```

```
     5 ,3.e1,7.,3.65e5,1.1e2,9.75e2,4.e2,2.93e2,3.75e2,1.31e2,6.56e2,1.8
     6 7e2,1.27e2,4.6e2,6.7e1,5.e2,8.75e2,4.33e2,7.e1,7.e1,3.65e5,6.25e2
     7 ,4.e2,4.,1.4e1,5.5,2.7e1,2.e1,3.e2,1.35e1,2.5e1,1.95e2,3.0e1,4.1e
     8 1,3.65e5,3.65e5,3.65e5,1.e1,2.4e3,5.7e4,5.8e4,3.0e1,5.4e4,3.e4,3
     9 .48e3,3.e3,3.65e5,3.65e5,3.65e5,3.65e5/
c
      data (tbwbz(k),k=1,100) /1.2e1,3.65e5,2.,1.8e2,5.e-1,1.e1,9.e1,1.4
     1 e1,8.08e2,3.65e5,1.1e1,1.8e2,5.5e2,6.e1,2.57e2,9.e1,2.90e1,3.65e5
     2 ,5.8e1,1.64e4,1.43e3,3.2e2,4.2e1,6.16e2,3.6e1,8.e2,2.4e1,6.67e2,2
     3 .5e2,9.33e2,6.,1.,2.8e2,1.1e1,8.,3.65e5,8.e1,1.3e4,1.4e4,4.5e2,7.
     4 6e2,1.e2,3.,1.3e2,1.04e1,5.,5.2e1,2.e2,4.8e1,2.e2,3.8e1,1.5e1,1.3
     5 8e2,3.65e5,1.1e2,6.5e1,5.e2,5.63e2,7.5e2,6.56e2,6.56e2,6.56e2,6.3
     6 5e2,5.5e2,6.7e2,7.e2,7.5e2,6.5e2,6.75e2,6.85e2,7.5e2,5.63e2,2.4e2
     7 ,1.,7.,2.,2.e1,2.4e1,1.2e2,1.e1,2.5e1,1.46e3,5.,3.e1,2.7e1,3.65e5
     8 ,7.e1,8.1e3,2.4e4,5.7e4,4.1e4,3.e2,3.9e4,6.5e4,2.e4,2.4e4,6.5e4,6
     9 .5e4,6.5e4,6.5e4/
c
      data (cbbnz(k),k=1,100) /1.e5,1.e-2,3.e-3,6.e2,3.,1.3e5,2.9e4,6.e5
     1 ,2.e2,1.e-1,1.06e4,1.42e3,2.6e1,3.3e2,8.e4,2.4e3,1.7e3,1.e2,6.24e
     2 3,9.36e4,5.e-2,1.,1.5,4.8,2.4,6.e2,5.,1.66e1,8.4,1.08e2,1.,1.e2,1
     3 .e-2,1.,3.e-1,1.e-2,1.,8.6e1,5.,1.,5.3e2,4.8e-1,1.e-11,5.e-3,1.e-
     4 1,1.e-1,1.e-1,5.e-1,1.,4.8,2.e-1,5.e-3,3.e-1,1.e-3,6.,1.6,4.e3,5
     5 .e-1,6.7e2,2.5e3,5.e-21,5.e-2,5.e-3,5.e-2,5.e-3,3.e2,7.e1,8.e1,5
     6 .e-3,5.e-3,7.e1,5.e-1,1.,1.e2,1.e-1,1.e-1,1.e-3,1.,2.5,1.,1.e-1,3
     7 .36e1,1.e1,1.e-9,1.e-17,1.e-11,1.e-16,1.7e-8,1.e-8,6.e2,1.e-5,1.e
     8 -3,1.e-10,1.e-11,1.e-28,1.e-28,1.e-28,1.e-28,1.e-28,1.e-28/
c
      data (cbkdz(k),k=1,100) /1.e5,1.e-2,8.e-3,3.e-2,1.1e-1,2.4e5,2.9e4
     1 ,6.e5,8.e-1,1.e-1,2.2e3,1.9e2,9.e-1,2.1e1,2.e3,1.e4,2.1e3,1.e2,2.
     2 3e3,1.8e2,1.e-1,5.5e-2,8.e-2,5.e-2,1.7,1.2e2,2.2e-2,1.2e-1,3.74,8
     3 .1e1,1.e-2,1.e2,1.e-2,1.8e-1,3.,1.e-2,1.4,1.2e-1,5.,1.,1.,5.3e-1,
     4 1.e-11,5.e-2,1.e-1,1.e-1,1.1e-2,5.6e1,1.e1,5.7e-1,7.e-3,1.e-1,2.e
     5 -1,1.e-3,1.8e-2,4.4e-2,7.e-5,1.e-1,6.7e2,2.5e3,5.e-21,1.e-2,1.e-2
     6 ,5.e-2,5.e-3,3.e2,7.e1,8.e1,1.e-3,5.e-3,7.e1,5.e-1,5.,1.e2,1.e-1,
     7 1.e-1,1.e-3,1.,1.1e-1,4.e-2,1.e-1,2.4,4.7e-1,1.e-9,1.e-17,1.e-11,
     8 1.e-16,8.e-5,1.e-8,6.e2,1.e-5,1.e2,1.e-10,1.e-11,1.e-28,1.e-28,1
     9 .e-28,1.e-28,1.e-28,1.e-28/
c
      data (cblvz(k),k=1,100) /1.e5,1.e-2,6.e-3,3.e-2,1.3e-1,2.4e5,2.9e4
     1 ,6.e5,2.e4,1.e-1,1.4e3,2.3e2,1.6,2.4e1,3.3e3,1.e4,1.25e3,1.e2,3.3
     2 e3,9.1e1,5.e-2,2.1e-1,7.e-2,4.8e-2,2.9,3.1e2,1.3e-1,1.4e-1,1.69e1
     3 ,7.7e1,1.1e-2,1.e2,1.6e-2,3.1e-1,4.4e0,1.e-2,3.7e1,4.6e-2,1.,1.,1
     4 .1,1.82e1,1.e-11,5.e-3,1.e-1,1.e-1,4.2e-2,4.4,1.e1,9.6e-1,1.8e-2,
     5 1.e-2,1.1e-2,1.e-3,1.2e-2,2.7e-2,1.5e-3,1.e-1,6.7e2,2.5e3,1.e-21,
     6 1.e-2,1.e-3,1.e-2,5.e-4,3.e2,7.e1,8.e1,5.e-5,1.e-4,7.e1,5.,1.e1,1
     7 .e2,1.e-1,1.e-1,1.e-3,1.,1.7e-1,3.e-2,1.1e-1,3.3,9.1e-2,1.e-9,1.e
     8 -17,1.e-11,1.e-16,8.e-5,1.e-8,6.e2,1.e-5,1.e2,1.e-10,1.e-11,1.e-2
     9 8,1.e-28,1.e-28,1.e-28,1.e-28,1.e-28/
c
      data (cbwbz(k),k=1,100) /1.e5,1.e-2,1.3e-2,1.3e-4,2.9e-1,2.4e5,2.9
     1 e4,6.e5,2.e1,1.e-1,2.e3,2.9e2,1.,3.e1,1.15e4,2.5e3,2.e3,1.e2,2.e3
     2 ,1.26e4,1.e-2,3.9e-1,1.5e-2,2.e-1,3.75e-1,6.75e1,2.45e-2,2.e-1,1.
     3 3e0,3.15e1,3.e-5,1.e2,1.4e0,5.e-1,3.9e-1,1.e-2,4.75e0,4.,5.e-1,8.
     4 6e-2,7.e-1,1.75e-1,5.e-13,1.e-3,1.e-1,1.e-1,1.9e-2,7.2e-1,1.e1,2.
```

```
     5 9e-1,5.e-2,5.e-3,5.2e-1,1.e-3,1.45e-2,7.2e-1,7.e-1,5.e-2,6.7e2,2.
     6 5e3,5.e-22,5.e-3,5.e-4,5.e-3,5.e-4,3.e2,5.e-4,8.e1,5.e-4,5.e-4,5
     7 .e-4,1.e-1,5.e-1,1.e2,1.e-1,1.e-1,1.e-5,1.,1.45e-1,5.e-2,8.6e-2,1
     8 .75,1.45e-2,1.e-9,1.e-17,1.e-11,1.e-16,4.3e-10,1.e-8,6.e2,1.e-5,2
     9 .9e-4,1.e-10,1.e-11,1.e-28,1.e-28,1.e-28,1.e-28,1.e-28,1.e-28/
c
      data (fmk(k),k=1,100) /2.e-2,2.e-2,5.e-2,1.e-4,2.7e-3,1.5e-2,2.2e-
     1 2,2.e-2,1.4e-2,2.e-2,5.e-2,1.e-2,5.e-4,1.e-4,2.5e-2,1.75e-2,5.e-2
     2 ,2.e-2,1.e-2,1.6e-2,5.e-6,5.e-6,1.e-3,2.15e-3,2.5e-4,1.2e-3,1.e-3
     3 ,6.7e-3,1.4e-2,3.9e-2,5.e-5,5.e-4,6.e-3,4.5e-2,5.e-2,2.e-2,3.e-2,
     4 2.e-3,1.e-5,5.e-6,2.5e-3,7.5e-3,2.5e-2,1.e-6,1.e-2,1.e-2,5.e-2,1.
     5 25e-4,1.e-4,2.5e-3,1.5e-3,1.e-3,2.e-2,2.e-2,1.4e-2,6.e-4,5.e-6,2
     6 .e-5,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,5.e-6,
     7 5.e-6,5.e-6,5.e-6,5.e-6,2.5e-6,5.e-4,2.5e-2,5.e-3,5.e-3,5.e-3,5.e
     8 -3,3.75e-2,2.25e-2,6.2e-4,5.e-4,3.e-3,5.e-2,2.e-2,5.e-2,1.5e-2,5
     9 .e-6,5.e-6,5.e-6,5.e-4,5.e-6,1.5e-6,5.e-6,5.e-6,1.5e-6,1.5e-6,1.5
     1 e-6,1.5e-6/
c
c**** beef cow age structure
      data (a(k,1,1),k=1,8) /.02,.4,.5,.02,.02,.02,.01,.01/
      data (a(k,1,2),k=1,8) /.02,.4,.5,.02,.02,.02,.01,.01/
c
c**** dairy cow age stucture
      data (a(k,2,1),k=1,8) /.4,.06,.12,.18,.12,.06,.036,.024/
      data (a(k,2,2),k=1,8) /.4,.06,.12,.18,.12,.06,.036,.024/
c
      data (wutp(j,1),j=1,22) /3*200.,150.,300.,150.,2*200.,300.,150.,3*
     1 200.,2*300.,150.,300.,2*200.,300.,150.,100./
c
      data (hf(j,1),j=1,22) /2*1.,30.,175.,300.,70.,1.,30.,70.,175.,30.,
     1 2.,30.,2*300.,70.,300.,30.,1.,70.,175.,1./
c
      data (sctp(j,1),j=1,22) /3*1.,3*0.,2*1.,2*0.,3*1.,4*0.,2*1.,2*0.,1
     1 ./
c
      data (astp(j,1),j=1,22) /4*0.,1.,3*0.,1.,4*0.,2*1.,0.,1.,2*0.,1.,2
     1 *0./
c
      data (rxpp(j,1),j=1,22) /11*.161,11*.12/
c
      data (rfp(j,1),j=1,22) /11*.0013,11*.0054/
c
      data (zkd(j),j=1,110) /2*1.e06,300.,650.,3.,1.e06,.5,1.e06,150.,1.
     1e6,100.,4.5,1500.,30.,3.5,7.5,.25,1.e6,5.5,4.0,3*1000.,850.,65.,25
     2.,45.,150.,35.,40.,1500.,25.,200.,300.,7.5,1.e6,60.,35.,500.,3000.
     3,350.,20.,1.5,350.,60.,60.,45.,6.5,1500.,250.,45.,300.,60.,1.e06,
     41000.,60.,15*650.,1500.,650.,150.,7.5,450.,150.,90.,25.,10.,1500.,
     5900.,200.,500.,10.,1.e06,250.,450.,1500.,1.5e05,2500.,450.,30.,
     64500.,700.,2000.,4*1.e06,10*1.e06/
      end
```

14.19. INVERT (bsm.for)

```
      subroutine invert (ii,in,o,lx,insct,bpx,rdidx)
c
*********************************************************************
c
      include 'main.inc'
c
      common/pa/ absw(120,ko), adia(120,ko), al4(m), al5(m),
     1bddt(120,ko),c66(m),degg(120,ko), egg(120,ko),
     2hatce(120,ko), hatch(120,ko),hatmp(120,ko),hattmp(120,ko),
     3ondv(30,ko), onyph(30,ko), pdiape(120,ko), thz(24), tmpday(30,ko)
     4,watcnr(ko,m,n),yadlt(30,ko), yndv(30,ko),
     5 ynyph(30,ko),hemlos(ko,m,n),bltgt(ko,m),fdtmp(ko),tpsm(ko),
     6 dvtmp(ko),adlt(2,ko),iegg(ko),iegd(ko),iegh(ko),eggs,deggs,
     7 hatchs, pdiaps,aha(ko),yadlts,ynyphs,onyphs,eglay,degday(ko),
     8 colday(ko),diaday(ko),eggmrt(ko),aii,summrt(ko,2)
c
      common/pw/ gaslos(ko),urnlos(ko),wtbn(kr,ko),
     1wtkd(kr,ko),wtlv(kr,ko),wtms(kr,ko),wtrs(kr,ko),wtlu(kr,ko)
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),wt1(kr,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      common/ha/ popt(ko),tqx0(kr,2,ko),jage(ko),aiif(ko,2),ifm(kr),
     1vurlfj(ko),vmljk(kr,ko),popp(2,ko),ikn(2,ko,8),sfpop(ko),
     2gst(ko,2),gstn(ko),gestp(ko),aiiir(ko,2),agem(kr,ko),cagm(ko,2),
     3cntk(2),brts(ko,2,2),frpr(2,ko),agcor(ko),tpppt(ko),sfd(8,ko),
     4tppcs(ko),tppps(ko),ca1(ko),clf(ko),lagf(ko),lua(ko),offsp(ko,3,2)
     5,affsp(ko,3),rat12(ko),tqp(kr,ko),tqs(kr,ko),vpelt(kr,ko,m),
     6vgut(kr,ko,m),cnmus(kr,ko,m),cnliv(kr,ko,m),cnkid(kr,ko,m),
     7cnfat(kr,ko,m),cnlun(kr,ko,m),cnbon(kr,ko,m),cnres(kr,ko,m),
     8cnblo(kr,ko,m),cnbod(kr,ko,m),cngut(kr,ko,m),cnpel(kr,ko,m),
     9vpln(kr,ko,m),vsol(kr,ko,m),vlug(kr,ko,m),cnpln(kr,ko,m),
     1cnsol(kr,ko,m),cnlug(kr,ko,m),rtpls(ko),rtmus(ko),rtkid(ko),
     2rtliv(ko),rtgut(ko),rthrt(ko),rtlun(ko),rtspl(ko),rtmar(ko),
     3rtfat(ko),rtres(ko),cbp(kr,ko,m),cbf(ko,m),cbw(ko,m),tacres(n),
     4vplt(kr,ko),c31plt(kr,ko,m),cnmlk(kr,ko,m),cnurnc(kr,ko,m),
     5cnfec(kr,ko,m),cnurn(kr,ko,m)
c
      if (rdidx.gt.0.0) go to 150
c
      nuu=nu
      if(nu.eq.0)goto5050
```

```
      rewind 7
      ict=0
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-181818.)5702,5704.5702
 5555 nu=0
 5704 ict=ict+1
      if(ict.eq.insct)goto5050
      goto 5702
 5050 call atcol('white','green')
      call wnclos(1)
      call wnopen(0,0,51,5)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE INVERT                    ')
      call wnoust('                                  ')
      call wnoust('******************************                    ')
      call wncuxy(1,3)
      print *,'invert number=',insct
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                  ')
      call wnoust('ENTER yes=1, no=0                          ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 181818.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
```

```
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c** initializes body components
      do151k=1,inz
c ** transport coefficient from malphigian tubules to gut
c
 5000 print *,'ENTER transport coefficient from malphigian tubule to'
      print *,'gut for contaminant=',k,' If 0.0 is entered then'
      print *,'coefficient will default to 0.631'
      if(nu.eq.0)read(*,*,err=5000) bltgt(in,k)
      if(nu.eq.1)read(7,*) bltgt(in,k)
      if(bltgt(in,k).le.0.0) bltgt(in,k)=0.631
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5000) ans
      if(ans.gt.0.and.nu.eq.0)goto5000
      write (6,*) bltgt(in,k)
      if(iu.eq.1) print *, bltgt(in,k)
c
c ** cumulative losses of contaminant from hemolymph to gut
      do154lz=1,ipx
      hemlos(in,k,lz)=0.0
  154 continue
c *** end of region loop ****************************************
  151 continue
c**** end of contaminant loop ****************************************
c
c**** insect growth parameters entered
c insect feeding temperature, deg. C.
c
 5001 print *,'ENTER minimum insect feeding temperature, deg. C'
      print *,'if 0.0 entered, then value will default to 16.0'
      if(nu.eq.0)read(*,*,err=5001) fdtmp(in)
      if(nu.eq.1)read(7,*) fdtmp(in)
      if(fdtmp(in).le.0.0) fdtmp(in)=16.0
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write (6,*) fdtmp(in)
      if(iu.eq.1) print *,fdtmp(in)
c
c minimum development temperature DVTMP is set to 8.0 deg. C. for
c grasshoppers in temperate zone
c
 5002 print *,'ENTER insect minimum development temperature, deg. C'
      print *,'if 0.0 entered, then value will default to 8.0'
      if(nu.eq.0)read(*,*,err=5002) dvtmp(in)
      if(nu.eq.1)read(7,*) dvtmp(in)
      if(dvtmp(in).le.0.) dvtmp(in)=8.0
```

```
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write (6,*) dvtmp(in)
      if(iu.eq.1) print *, dvtmp(in)
c
 5003 print *,'ENTER insect minimum degree day accumulation before'
      print *,'hatching will occur; if 0.0 is entered, then value '
      print *,'will default to 250 deg-days'
      if(nu.eq.0)read(*,*,err=5003) degday(in)
      if(nu.eq.1)read(7,*) degday(in)
      if(degday(in).le.0.0) degday(in)=250.0
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write (6,*) degday(in)
      if(iu.eq.1) print *, degday(in)
c
 5004 print *,'ENTER insect minimum day accumulation for cold treatment'
      print *,'of diapause eggs below minimum developmental temperature'
      print *,'if 0.0 is entered, then value will default to 40 days'
      if(nu.eq.0)read(*,*,err=5004) colday(in)
      if(nu.eq.1)read(7,*) colday(in)
      if(colday(in).le.0.) colday(in)=40.0
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write (6,*) colday(in)
      if(iu.eq.1) print *, colday(in)
c
 5005 print *,'ENTER julian day for 100% production of diapause eggs'
      print *,'at temperatures below minimum developmental temperature'
      print *,'if 0.0 is entered, then value will default to day 240'
      if(nu.eq.0)read(*,*,err=5005) diaday(in)
      if(nu.eq.1)read(7,*) diaday(in)
      if(diaday(in).le.0.) diaday(in)=240.0
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write (6,*) diaday(in)
      if(iu.eq.1) print *, diaday(in)
c
 5006 print *,'ENTER mortality rate, day-1, of regular, diapause, and'
      print *,'post-diapause eggs which is constant througout the year'
      print *,'if 0.0 is entered, then value will default to 0.003'
      if(nu.eq.0)read(*,*,err=5006) eggmrt(in)
      if(nu.eq.1)read(7,*) eggmrt(in)
      if(eggmrt(in).le.0.) eggmrt(in)=0.003
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write (6,*) eggmrt(in)
      if(iu.eq.1) print *, eggmrt(in)
 5007 if(nu.eq.1)goto5500
```

```
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN INVERT                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for INVERT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5007)ans
      if(ans) 11,5500,11
   11 rewind 6
      ict=0
      do5501l=1,20000
      read(6,*,err=5501)bcheck
      if(bcheck-181818.)5501,5503,5501
 5503 ict=ict+1
      if(ict.ne.insct)goto5501
      backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
c
c insect blood weight(XB), and oldest age group(A) set to 1.0
      wt(ii,in)=rr(in)
      wtpr=wt(ii,in)
c biomass of adult insects initialized, g/m2
      adlt(1,in)=0.0
      adlt(2,in)=0.0
c food available for insects initialized
      afodm=0.0
c
c ** nymph and young adult stage loop ********************************
      do 160 k=1,30
c 30 developmental stages of old and young nymphs, and young adult numbers
c initialized, g/m2
      ynyph(k,in)=0.0
      onyph(k,in)=0.0
      yadlt(k,in)=0.0
c associated developmental parameters initialized(to be defined)
      tmpday(k,in)=0.0
      yndv(k,in)=0.0
      ondv(k,in)=0.0
  160 continue
c ** end nymph and young adult loop ************************************
```

```
c ** egg stage loop ***************************************************
      do 170 k=1,120
c 120 developmental stages of diapause eggs(DEGG), and non-diapause eggs
c initialized ,g/m2
      degg(k,in)=0.0
      egg(k,in)=0.0
c associated developmental parameters initialized( to be defined)
      bddt(k,in)=0.0
      hatmp(k,in)=0.0
      hatce(k,in)=0.0
      hatch(k,in)=0.0
      absw(k,in)=0.0
      adia(k,in)=0.0
      hattmp(k,in)=0.0
      pdiape(k,in)=0.001
  170 continue
c ** end egg stage loop ************************************************
c maximum soil matrix potential set
      phimx=15.
      eglay=0.0
      aha(in)=0.0
      tqp(1,in)=0.0
      return
  150 continue
c
c ** 24 hour loop ******************************************************
      do 190 l=1,24
c initializes hourly temperatures, deg C.
      thz(l)=0.0
  190 continue
c ** end 24 hour loop **************************************************
c intializes regionally weighted maximum(TMXZ), and minimum temperature(TMNZ)
      tmxz=0.0
      tmnz=0.0
c
c ** region loop *******************************************************
      do 210 lz=1,ipx
c estimates regionally weighted maximum and minimum daily temperatures based
c on a 24 hour clock; RFRZ=fraction time spent in region(LZ)
      tmxz=tmxz+tmx(lz)*rfrz(in,lz)*sptfrr(lz)
      tmnz=tmnz+tmn(lz)*rfrz(in,lz)*sptfrr(lz)
c
c ** end 24 hour clock *************************************************
      do 200 l=1,24
c 24 hour temperature, deg C., for day AIII
      thz(l)=thr(l,lz)*rfrz(in,lz)*sptfrr(lz)+thz(l)
  200 continue
c ** end 24 hour clock *************************************************
  210 continue
c ** end region loop ***************************************************
c excludes all but day one of each year(AI1)
      if (aiii.gt.1.0)goto220
c initializes egg developmental stage parameters
c     iegd(in)=1
```

```
      iegg(in)=1
      iegh(in)=1
      aha(in)=0.0
c
c *** insect stage loop ************************************************
      do221j=1,120
c initializes immature insect stages
      if(j.gt.30)goto222
      ynyph(j,in)=0.0
      onyph(j,in)=0.0
      yadlt(j,in)=0.0
      yndv(j,in)=0.0
      ondv(j,in)=0.0
      tmpday(j,in)=0.0
  222 hatch(j,in)=0.0
      hatce(j,in)=0.0
      hatmp(j,in)=0.0
      absw(j,in)=0.0
c     adia(j,in)=0.0
      hattmp(j,in)=0.0
      bddt(j,in)=0.0
      egg(j,in)=0.0
      degg(j,in)=0.0
      eglay=0.0
      if(pdiape(j,in).le.0.0005)pdiape(j,in)=0.0005
  221 continue
c ** end immature insect loop ******************************************
c initializes mature insect stage
      adlt(1,in)=0.0
      adlt(2,in)=0.0
c imitializes egg developmental temperature gate
  220 tpsm(in)=0.0
c
c 24 hour day loop *****************************************************
      do 230 j=1,24
c sums number of hours/day above developmental temperature(FDTMP) where
c feeding occurs
      if (thz(j).ge.fdtmp(in)) tpsm(in)=tpsm(in)+1.0
  230 continue
c ** end 24 hour day loop **********************************************
c TPFAC is the 24 hour fraction where feeding occurs adjusted for available
c to required food ratio
      tpfac=(tpsm(in)/24.)
      tpfat=tpfac
      if(ototb(in).lt.otota(in))tpfat=0.0
c initializes hatching, biomass, and mortality parameters
      aha(in)=0.0
      ainsm=0.0
      summrt(in,1)=0.0
      summrt(in,2)=0.0
c population numbers of egg stages of insects initialized
      eggs=0.0
      deggs=0.0
      pdiaps=0.0
```

```
      hatchs=0.0
c
c ** egg stage loop ******************************************************
      do 380 kg=1,120
c egg stage populations over all age cohorts estimated
      eggs=eggs+egg(kg,in)
      if(eggs.le.0.0)iegg(in)=1
      deggs=deggs+degg(kg,in)
      if(deggs.le.0.0)iegd(in)=1
      pdiaps=pdiaps+pdiape(kg,in)
      hatchs=hatchs+hatch(kg,in)
  380 continue
      ainsme=eggs+deggs+pdiaps
c ** end egg stage loop ****************************************************
c
c ** egg development loop ***********************************************
      do 290 k=1,120
c diapause and post-diapause and egg stages adjusted for mortality
      summrt(in,2)=summrt(in,2)+eggmrt(in)*(egg(k,in)+degg(k,in)+
     10.5*pdiape(k,in))
      summrt(in,1)=summrt(in,2)*(1.0-sxrt(in))
      summrt(in,2)=summrt(in,2)*sxrt(in)
      degg(k,in)=degg(k,in)-eggmrt(in)*degg(k,in)
      pdiape(k,in)=pdiape(k,in)-0.5*eggmrt(in)*pdiape(k,in)
      egg(k,in)=egg(k,in)-eggmrt(in)*egg(k,in)
c excludes empty diapause egg stages
      if (degg(k,in).le.0.0) go to 260
c excludes daily temperatures above minimum developmental temperature
      if (ttz(in).le.dvtmp(in)) go to 260
c excludes high soil matrix potential conditions
      if (phitt(2).lt.phimx) go to 240
c increments soil water diapause development gate
      absw(k,in)=absw(k,in)+1.0
c ends degree-day accumulation
      if (absw(k,in).ge.3.0) go to 260
      go to 250
c initializes diapause development gate
  240 absw(k,in)=0.0
c accumulates degree-days above minimum(DVTMP)
  250 bddt(k,in)=bddt(k,in)+(ttz(in)-dvtmp(in))
c excludes degree-day accumulation below 0.75
      if (bddt(k,in).lt.75) go to 260
c converts diapause eggs to post-diapause stage
      pdiape(k,in)=pdiape(k,in)+degg(k,in)
c initializes spent diapause egg compartment and degree-day accumulator
      degg(k,in)=0.0
      bddt(k,in)=0.0
      absw(k,in)=0.0
c excludes days with mean daily greater than 5 deg. C.
  260 if (ttz(in).gt.5.0) go to 270
c increments diapause development gate (cold treatment)
      adia(k,in)=adia(k,in)+1.0
c excludes accumulation below minimum number of days cold treatment
  270 if (adia(k,in).lt.colday(in)) go to 280
```

```
c excludes hatching gate increment below minimum temperature(DVTMP)
      if (ttz(in).lt.dvtmp(in)) go to 280
c increments hatching degree-day gate above minimum temperature
      hatch(k,in)=hatch(k,in)+(ttz(in)-dvtmp(in))
c excludes hatching below required degree-days
      if (hatch(k,in).lt.degday(in)) go to 280
c excludes hatch temperature gate incrementing below 24 deg. C.
      if (tmxz.ge.24.) hattmp(k,in)=hattmp(k,in)+1.
c excludes hatch gate accumulations below 4
      if (hattmp(k,in).lt.4.) go to 280
c estimates difference between mean daily temperature and minimum(DVTMP)
      hateg=ttz(in)-dvtmp(in)
c estimates conversion fraction of post-diapause eggs of stage(K) to young
c estimates fraction of diapause eggs not entering young nymph stage
      egh=exp(-.693*hateg/30.0)
c estimates mass, g/m2 of young nymphs produced from diapause eggs in
c stage(K)
      ieg=iegh(in)
      ynyph(ieg,in)=pdiape(k,in)*(1.0-egh)+ynyph(ieg,in)
c adjusts un-developed diapause egg mass(g/m2) in stage(K)
      pdiape(k,in)=pdiape(k,in)*egh
c resets hatching parameter
      aha(in)=1.0
      if(pdiape(k,in).le.0.0001)pdiape(k,in)=0.0
c resets hatching parameter
      if(pdiape(k,in).gt.0.0)goto280
c resets post-diapause development parameters
      aha(in)=0.0
      adia(k,in)=0.0
      hatch(k,in)=0.0
      hattmp(k,in)=0.0
c excludes empty egg stage
  280 if (egg(k,in).le.0.0) go to 290
c excludes mean daily temperatures below minimum(DVTMP)
      if (ttz(in).lt.dvtmp(in)) go to 290
c increment hatch degree-day gate for regular eggs
      hatce(k,in)=hatce(k,in)+(ttz(in)-dvtmp(in))
c excludes degree-day accumulation below required
      if (hatce(k,in).lt.degday(in)) go to 290
c conditional increment of hatch gate at temperatures 24 deg.C. and above
      if (tmxz.ge.24.) hatmp(k,in)=hatmp(k,in)+1.
c excludes hatching below accumulator value specified
      if (hatmp(k,in).lt.4.) go to 290
c estimates difference between mean daily temperature and minimum(DVTMP)
      hateg=ttz(in)-dvtmp(in)
c estimates ratio of eggs(K) not entering young nymph stage(IEGH)
      egh=exp(-.693*hateg/30.)
c mass(g/m2) of young nymphs produced from eggs(K)
      ieg=iegh(in)
      ynyph(ieg,in)=egg(k,in)*(1.0-egh)+ynyph(ieg,in)
c adjusts un-developed egg mass(g/m2) in stage(K)
      egg(k,in)=egg(k,in)*egh
c egg hatching parameter reset
      aha(in)=1.0
```

```
      if(egg(k,in).le.0.0001)egg(k,in)=0.0
c excludes occupied egg stages
      if (egg(k,in).gt.0.0) go to 290
c egg hatching parameters re-initialized
      aha(in)=0.0
      hatce(k,in)=0.0
      hatmp(k,in)=0.0
  290 continue
c ** end egg development stage ****************************************
c nymph stage incremented
      if (aha(in)-1.0)155,153,155
  153 iegh(in)=iegh(in)+1
  155 if(iegh(in).gt.30)iegh(in)=1
c required food intake initialized
      fdin(ii,in)=1.0
c
c ** nymph loop ****************************************************
      do331j=1,30
c metamorphic index as a function of maximum daily temperature; bounded as a
c function of maximum, minimum and extreme temperatures, deg. C
      pnyd=.003*tmxz-.054
      if (tmxz.gt.38.) pnyd=.06
      if (tmxz.lt.16..or.tmxz.gt.48.) pnyd=0.0
c old nymph index set equal to young nymph index; bounded
      pnod=pnyd
c ** nymph loop ****************************************************
c young and old nymph developmental indicies
      yndv(j,in)=yndv(j,in)+pnyd
      ondv(j,in)=ondv(j,in)+pnod
c conditional conversion of young nymphs to old nymphs; YNDV re-initialized
      if (yndv(j,in).lt.0.25)goto325
      onyph(j,in)=onyph(j,in)+ynyph(j,in)
      ynyph(j,in)=0.0
      yndv(j,in)=0.0
c conditional conversion of old nymphs to young adults; ONDV re-initialized
  325 if (ondv(j,in).lt.1.0)goto327
      yadlt(j,in)=yadlt(j,in)+onyph(j,in)
      onyph(j,in)=0.0
      ondv(j,in)=0.0
c conditional temperature-day index increment for young adult development
c using mean regional temperature over all regions(TMXZ)
  327 tmxxz=tmxz
      if (tmxxz.lt.12.) tmxxz=0.0
      if (yadlt(j,in).gt.0.0) tmpday(j,in)=tmpday(j,in)+tmxxz
c conditional partial conversion of young adults to adult males(1), and adult
c females(2), g/m2, assuming 1:2 ratio of males to females
      if (tmpday(j,in).lt.100.) goto333
      yadlt1=0.33*yadlt(j,in)
      yadlt2=0.66*yadlt(j,in)
      adlt(1,in)=yadlt1+adlt(1,in)
      adlt(2,in)=yadlt2+adlt(2,in)
      yadlt(j,in)=0.0
      tmpday(j,in)=0.0
c young and old nymph, and young adult biomass, g/m2, initialized;
```

```
c temperature index for young adult developmment set to 0.0
 333 if (ynyph(j,in).le.0.0) ynyph(j,in)=0.0
     if (ynyph(j,in).le.0.0) pnyd=0.0
     if (onyph(j,in).le.0.0) onyph(j,in)=0.0
     if (onyph(j,in).le.0.0) pnod=0.0
     if (yadlt(j,in).le.0.0) yadlt(j,in)=0.0
     if (yadlt(j,in).le.0.0) tmpday(j,in)=0.0
c** sums nymph stages biomass density, g/m2
     ynyph(j,in)=ynyph(j,in)*wt(ii,in)/wtpr
     onyph(j,in)=onyph(j,in)*wt(ii,in)/wtpr
     yadlt(j,in)=yadlt(j,in)*wt(ii,in)/wtpr
     ainsm=ainsm+ynyph(j,in)+onyph(j,in)+yadlt(j,in)
 331 continue
c ** end nymph loop ******************************************************
     adlt(1,in)=adlt(1,in)*wt(ii,in)/wtpr
     adlt(2,in)=adlt(2,in)*wt(ii,in)/wtpr
     ainsm=ainsm+adlt(1,in)+adlt(2,in)
c ** biomass densities of immature insects initialized
     ynyphs=0.0
     onyphs=0.0
     yadlts=0.0
c** sums nymph stages
     do332j=1,30
     ynyphs=ynyphs+ynyph(j,in)
     if(ynyphs.le.0.0)iegh(in)=1
     onyphs=onyphs+onyph(j,in)
     yadlts=yadlts+yadlt(j,in)
 332 continue
c mortality fraction set depending on mean daily temperature of regions
     amrtp=0.0
     if (tmnz.lt.5.) amrtp=.02*(5.-tmnz)
c precipitation mortality fraction based on average precip over ranges(PAV)
c for young nymphs;bounded
     aprcfy=.00167*pav(in)
     if (aprcfy.ge.0.1) aprcfy=0.1
c precipitation mortality fraction for old nymphs and young adults;bounded
     aprcf=0.0
     if (pav(in).gt.30.) aprcf=(pav(in)-30.)*.0025
     if (aprcf.ge..05) aprcf=.05
c temperature mortality fraction for young nymphs(ATMPY), and for old nymphs
c and young adults(ATMP); bounded
     atmpy=0.0
     if (tmxz.gt.33.) atmpy=.0075
     atmp=0.0
     if (tmxz.gt.33) atmp=.0025
c young nymph(ANYMRT),old nymph(AOYMRT),and young adult(AAYMRT) mortality
c fractions for each stage(30)
     anymrt=amrtp+.0075+afodm*.0075+aprcfy+atmpy-tqp(1,in)
     aoymrt=amrtp+.0025+afodm*.0025+aprcf+atmp-tqp(1,in)
     aaymrt=amrtp+.0025+afodm*.0025+aprcf+atmp-tqp(1,in)
     if(anymrt.le.0.0)anymrt=0.0
     if(aoymrt.le.0.0)aoymrt=0.0
     if(aaymrt.le.0.0)aaymrt=0.0
c **** nymph loop ****
```

```
      do330j=1,30
c total mortality, g/m2, for young and old nymphs, and young adults over all
c ages(30) combined
      summrt(in,2)=summrt(in,2)+anymrt*ynyph(j,in)+aoymrt*onyph(j,in)+
     1aaymrt*yadlt(j,in)
      summrt(in,1)=summrt(in,2)*(1.0-sxrt(in))
      summrt(in,2)=summrt(in,2)*sxrt(in)
c net growth, g/m2 of nymph and young adult stages after adjustment for
c mortality losses; bounded
      dige(in)=(1.-grna(in))*.45+grna(in)*.75
c ** digestive fraction ajusted for wasted food which is converted to
c ** humus
      dige(in)=0.5*dige(in)
      ynp=ynyph(j,in)
      onp=onyph(j,in)
      yad=yadlt(j,in)
c ** region loop ***************************************************
      deady=0.0
      deado=0.0
      deada=0.0
      fdiny=0.0
      fdino=0.0
      fdina=0.0
      do 320 lz=1,ipx
      deady=anymrt*ynp*rfrz(in,lz)*sptfrr(lz)+deady
      deado=aoymrt*onp*rfrz(in,lz)*sptfrr(lz)+deado
      deada=aaymrt*yad*rfrz(in,lz)*sptfrr(lz)+deada
      fdiny=dige(in)*.37*tpfat*ynp*rfrz(in,lz)*sptfrr(lz)+fdiny
      fdino=dige(in)*.37*tpfat*onp*rfrz(in,lz)*sptfrr(lz)+fdino
      fdina=dige(in)*.37*tpfat*yad*rfrz(in,lz)*sptfrr(lz)+fdina
  320 continue
      ynyph(j,in)=ynyph(j,in)+fdiny-deady
      onyph(j,in)=onyph(j,in)+fdino-deado
      yadlt(j,in)=yadlt(j,in)+fdina-deada
      if (ynyph(j,in).le.0.0) ynyph(j,in)=0.0
      if (onyph(j,in).le.0.0) onyph(j,in)=0.0
      if (yadlt(j,in).le.0.0) yadlt(j,in)=0.0
      do322lz=1,ipx
      otin=0.0
c **** nuclide loop
      do321k=1,inz
      hemlos(in,k,lz)=0.0
      if(ainsm.le.0.0)goto321
c** nymph/larval stage contaminant intake
      hemlos(in,k,lz)=bltgt(in,k)*v(ii,in,k)*ynyph(j,in)*sptfrr(lz)
     1*rfrz(in,lz)/(ainsm*agr(in,lz))+hemlos(in,k,lz)
      hemlos(in,k,lz)=hemlos(in,k,lz)+bltgt(in,k)*v(ii,in,k)*onyph(j,in)
     1*sptfrr(lz)*rfrz(in,lz)/(ainsm*agr(in,lz))
      hemlos(in,k,lz)=hemlos(in,k,lz)+bltgt(in,k)*v(ii,in,k)*yadlt(j,in)
     1*sptfrr(lz)*rfrz(in,lz)/(ainsm*agr(in,lz))
  321 continue
c **** end nuclide loop
      if(ainsm.le.0.0)goto322
c total food, kgdwt, required by above three stages for each age cohort
```

```
      otin=(agr(in,lz)*tpfac*(ynyph(j,in)+onyph(j,in)+yadlt(j,in))
     1/1000.)*sptfrr(lz)*rfrz(in,lz)+otin
  322 continue
c ** end region loop ******************************************************
c total food required, kgdwt, over all stages and age cohorts
      fdin(ii,in)=fdin(ii,in)+otin
      if(ainsm.le.0.0)goto330
      v(ii,in,k)=v(ii,in,k)-bltgt(in,k)*v(ii,in,k)*ynyph(j,in)/ainsm
      v(ii,in,k)=v(ii,in,k)-bltgt(in,k)*v(ii,in,k)*onyph(j,in)/ainsm
      v(ii,in,k)=v(ii,in,k)-bltgt(in,k)*v(ii,in,k)*yadlt(j,in)/ainsm
      if(v(ii,in,k).le.0.0)v(ii,in,k)=0.0
c
  330 continue
c ** end nymph loop *******************************************************
c total insect mass, kgdwt
      wt(ii,in)=0.0
      do334lz=1,ipx
      wt(ii,in)=wt(ii,in)+(ainsm+ainsme)*agr(in,lz)*sptfrr(lz)/1000.
  334 continue
      if(wt(ii,in).le.0.0)wt(ii,in)=0.0001
      wtpr=wt(ii,in)
c temperature mortality factor for adult insects initialized and
c conditionally set(low end of survivial)
      amrtp=0.0
      if (tmnz.lt.5.) amrtp=.02*(5.-tmnz)
c same except for high end of survival range
      atmpa=0.0
      if (tmxz.gt.33.) atmpa=.00354
c total mortality fraction
      aadmrt=amrtp+.00354+afodm*.00354+aprcf+atmpa-tqp(1,in)
      if(aadmrt.le.0.0)aadmrt=0.0
c
c ** region loop ******************************************************
      afoda=0.0
      do 360 lz=1,ipx
      if(ototb(in).le.0.0)goto360
c food available for insect consumption, gdwt/m2
      afoda=ototb(in)*1000.*sptfrr(lz)/agr(in,lz)+afoda
  360 continue
c **** end region loop
      afodm=0.0
      if(ototb(in).le.0.0)goto361
c food availability factor; bounded
      afodm=10.*ainsm/afoda
      if(ainsm.le.0.0)goto361
      if (afoda/ainsm.gt.10.) afodm=1.0
      if (afoda/ainsm.le.1.0) afodm=10.
  361 dead1=0.0
      dead2=0.0
      fdin1=0.0
      fdin2=0.0
      do363lz=1,ipx
c male and female insect net mass increments adjusted for mortality losses,
c g/m2; bounded
```

```
      dead1=aadmrt*adlt(1,in)*rfrz(in,lz)*sptfrr(lz)+dead1
      dead2=aadmrt*adlt(2,in)*rfrz(in,lz)*sptfrr(lz)+dead2
      fdin1=0.6*dige(in)*.37*tpfat*adlt(1,in)*sptfrr(lz)*rfrz(in,lz)
     1+fdin1
      fdin2=0.6*dige(in)*.37*tpfat*adlt(2,in)*sptfrr(lz)*rfrz(in,lz)
     1+fdin2
  363 continue
c **** end region loop *******
      adlt(1,in)=adlt(1,in)+fdin1-dead1
      if (adlt(1,in).le.0.0)adlt(1,in)=0.0
      adlt(2,in)=adlt(2,in)+fdin2-dead2
      if (adlt(2,in).le.0.0)adlt(2,in)=0.0
c** body burdens for adult insects
      otin=0.0
c **** region loop ****
      do364lz=1,ipx
c **** contaminant loop ****
      do362k=1,inz
      if(ainsm.le.0.0)goto362
c** adult insect stage contaminant loss strategy
      hemlos(in,k,lz)=hemlos(in,k,lz)+bltgt(in,k)*v(ii,in,k)*adlt(1,in)*
     1sptfrr(lz)*rfrz(in,lz)/(ainsm*agr(in,lz))
      hemlos(in,k,lz)=hemlos(in,k,lz)+bltgt(in,k)*v(ii,in,k)*adlt(2,in)*
     1sptfrr(lz)*rfrz(in,lz)/(ainsm*agr(in,lz))
  362 continue
c **** end contaminant loop ****
c total required food,kgdwt, adjusted for adult insect requirements
      otin=(0.6*tpfac*(adlt(1,in)+adlt(2,in))*agr(in,lz)/1000.)*
     1sptfrr(lz)*rfrz(in,lz)+otin
  364 continue
c ** end region loop ***************************************************
      fdin(ii,in)=fdin(ii,in)+otin
      if(ainsm.le.0.0)goto365
      v(ii,in,k)=v(ii,in,k)-bltgt(in,k)*v(ii,in,k)*adlt(1,in)/ainsm
      v(ii,in,k)=v(ii,in,k)-bltgt(in,k)*v(ii,in,k)*adlt(2,in)/ainsm
      if(v(ii,in,k).le.0.0)v(ii,in,k)=0.0
c total mortality, g/m2, adjusted for adult insect losses
  365 summrt(in,1)=summrt(in,1)+aadmrt*adlt(1,in)
      summrt(in,2)=summrt(in,2)+aadmrt*adlt(2,in)
c ** total food consumption,kgdwt, by all insect stages combined
      o=fdin(ii,in)
c egg lay fraction initialized
      eglay=0.0
c empty female stage excluded
      if (adlt(2,in).le.0.0) go to 390
c egg laying fraction set according to maximum daily temperature(deg. C)
      eglay=.00001*exp(.396*tmxz)
      if (tmxz.ge.25..and.tmxz.le.27.) eglay=0.2
      if (tmxz.gt.27.) eglay=2.010-.0670*tmxz
      if (tmxz.lt.10..or.tmxz.gt.30.) eglay=0.0
c null egg lay fraction excluded
      if (eglay) 391,390,391
c secondary egg laying fraction based on seasonal effects initialized(julian
c day number) for production of diapause type eggs
```

```
 391 egfrac=0.0
     if (aiii.ge.220.) egfrac=1.0-(aiii-220.)/600.
     if (aiii.gt.diaday(in).and.ttz(in).lt.dvtmp(in)) egfrac=1.0
c diapause egg production ,g/m2, based on adult female population biomass
c density
     degc=0.0
     if(egfrac.gt.0.0)degc=egfrac*eglay*adlt(2,in)
     adlt(2,in)=adlt(2,in)-degc
     if(adlt(2,in).le.0.0)adlt(2,in)=0.0
     ieg=iegd(in)
     degg(ieg,in)=degg(ieg,in)+degc
     if(degg(ieg,in).le.0.0)degg(ieg,in)=0.0
c diapause egg index incremented
     iegd(in)=iegd(in)+1
     if(iegd(in).gt.120)iegd(in)=1
c empty female adult stage excluded; null normal egg production excluded
     if (adlt(2,in)) 392,390,392
 392 if (egfrac-1.0) 393,390,393
c non-diapause egg production, g/m2
 393 egc=(1.0-egfrac)*eglay*adlt(2,in)
     adlt(2,in)=adlt(2,in)-egc
     if(adlt(2,in).le.0.0)adlt(2,in)=0.0
     ieg=iegg(in)
     egg(ieg,in)=egg(ieg,in)+egc
     if(egg(ieg,in).le.0.0)egg(ieg,in)=0.0
c non-diapause egg index incremented
     iegg(in)=iegg(in)+1
     if(iegg(in).gt.120)iegg(in)=1
 390 continue
c     print *,hatchs,eggs,deggs,pdiaps,ynyphs,onyphs,yadlts,adlt(1,in),
c    1adlt(2,in),ainsm,aiii,'i'
     return
     end
```

14.20. EFFECTS (btm.for)

```
      subroutine effects(in,lx,ii,mm,rdidx)
c
      include 'main.inc'
c
      common/pg/ a1th(m,n),altnt(7,8,ko),altyr(7,8,ko),
     1births(2,ko),c3(m),ch20b(kr,2),a1thg(m,n),
     2cmorta(7,8,2,ko),cmortb(7,8,2,ko),cmorty(7,8,2,ko),wt1(kr,ko),
     3cnatr(7,18,2,ko), cnaty(7,18,2,ko),fatb(kr,ko),prot1(kr,2),
     4gburd(m),gburi(kr,2,m),hl4(m),hl5(m),pop(2,ko),
     5orgrad(7,2,ko),pburi(kr,2,m),pltea(8,ko),plteay(8,ko),sxrt(ko),
     6protb(kr,2),slung(m),tpop(kr,2,ko),vfec(kr,2,m),
     7vintk(kr,2,m),vlos(kr,2,m),vlu(kr,ko,m),vr(kr,ko,m),vrt(kr,ko,m)
     8, vrt1(kr,ko,m), vuptk(kr,2,m), vurin(kr,ko,m), watcnh(m,n),
     9watcni(m,n), wburi(kr,2,m), zgrni(kr,2,m),zlvi(kr,2,m)
     1,zmeati(kr,2,m), zmilki(kr,2,m), zsvi(kr,2,m),ydos(ko)
c
      common/pz/ tcnat(kr,2,ko),tcrel(200,kr,2,ko),
     1orgdos(6,2,ko),tcn(200,kr,2,ko),abscan(2,ko),
     2relcan(2,ko),sorgb(6,2,ko),spncan(2,ko),hasum(2,ko),popsum(2,ko),
     3orgbrd(6,2,ko),sorgd(6,2,ko),srun(ko),absr(ko),popfr(ko)
c
      dimension dmdd(ko)
c
      if (rdidx.gt.0.) go to 190
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      if(mm.eq.1.and.in.le.nan)rewind 7
      if(in.gt.nan)rewind 7
      ict=0
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-919191.)5702,5704,5702
 5555 nu=0
 5704 ict=ict+1
      if(ict.eq.iian)goto5050
      goto5702
 5050 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('******************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE EFFECTS                    ')
      call wnoust('                                    ')
      call wnoust('******************************                    ')
      call wncuxy(1,3)
      print *,'animal/human number=',in
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
```

```
5700 call wnclos(1)
     call wnopen(0,0,51,8)
     call wnoust('*****************************************************')
     call wnoust('1=file input, 0=keyboard input: you have selected  ')
     call wnoust('                                                   ')
     call wnoust('ENTER yes=1, no=0                                  ')
     call wnoust('*****************************************************')
     call wncuxy(1,3)
     print *,'mode=',nu,'; want to reverse temporarily?'
     call wnouce(7,'press any key and enter above')
     call inkey(key)
     call wncuxy(1,8)
     read(*,*,err=5700)ans
     if(ans.le.0.0)goto5701
     nux=nu
     if(nux.eq.0)nu=1
     if(nux.eq.1)nu=0
5701 call wnclos(1)
     if(nu.eq.1)goto5706
     call wnopen(0,0,51,4)
     call wnoust('ENTER subroutine code: 919191.                     ')
     call wnouce(2,'press any key and enter above')
     call inkey(key)
     call wncuxy(1,3)
     read(*,*,err=5701)acheck
     call wncuxy(1,4)
     if(iu.eq.1)print *,'value(s)=',acheck
     if(iu.eq.1)call sleep@(5.0)
     call wnclos(1)
     call wnopen(0,0,51,3)
     call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
     call wnouce(2,'press any key and enter above')
     call inkey(key)
     call wncuxy(1,3)
     read(*,*,err=5701) ans
     if(ans.gt.0.0)goto5701
5706 write(6,*)acheck
     if(iu.eq.1.and.nu.eq.1)print *,acheck
c
5011 print *,'ENTER year of initial exposure for zoon',in
     if(nu.eq.0)read(*,*,err=5011) srun(in)
     if(nu.eq.1)read(7,*) srun(in)
     if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
     if(nu.eq.0)read(*,*,err=5011) ans
     if(ans.gt.0.and.nu.eq.0)goto5011
     write(6,*) srun(in)
     if(iu.eq.1)print *, srun(in)
c
5012 print *,'ENTER day modulus for calculating effects for zoon',in
     print *,'(a modulus=365 will estimate effects yearly..etc.)'
     if(nu.eq.0)read(*,*,err=5012) dmdd(in)
     if(nu.eq.1)read(7,*) dmdd(in)
     if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
     if(nu.eq.0)read(*,*,err=5012) ans
```

```
      if(ans.gt.0.and.nu.eq.0)goto5012
      write(6,*) dmdd(in)
      if(iu.eq.1)print *, dmdd(in)
c
 5023 print *,'ENTER whether absolute(1.0), or relative(2.0) risk model'
      print *,'will be used for zoon=',in,' ENTER 1.0 or 2.0 now'
c
      if(nu.eq.0)read(*,*,err=5023) absr(in)
      if(nu.eq.1)read(7,*) absr(in)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5023) ans
      if(ans.gt.0.and.nu.eq.0)goto5023
      write(6,*) absr(in)
      if(iu.eq.1)print *, absr(in)
c
 5024 print *,'ENTER population multiplication number applied to'
      print *,'zoon',in,'from ANIMAL or HUMTRN subroutines to account'
      print *,'for cancer production in the entire population'
c
      if(nu.eq.0)read(*,*,err=5024) popfr(in)
      if(nu.eq.1)read(7,*) popfr(in)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5024) ans
      if(ans.gt.0.and.nu.eq.0)goto5024
      write(6,*) popfr(in)
      if(iu.eq.1)print *, popfr(in)
c
c ** pltea(8)=expression period for cancer induction;may vary with
c ** sex and age, 8 possible variations can be entered
c
 5014 print *,'ENTER up to 8 different cancer expression periods,'
      print *,'years, for males and females of different ages; '
      print *,'fractions of years permitted: ex. 0.5, 1.5, 3.0, etc.'
c
      if(nu.eq.0)read(*,*,err=5014)(pltea(l,in),l=1,8)
      if(nu.eq.1)read(7,*)(pltea(l,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5014) ans
      if(ans.gt.0.and.nu.eq.0)goto5014
      write(6,*)(pltea(l,in),l=1,8)
      if(iu.eq.1)print *,(pltea(l,in),l=1,8)
c
c ** plteay(8)=upper age limit where a new plateau period must be
c ** specified for a given cancer type; 8 possible .
 5015 print *,'ENTER upper age, years, where a new expression period'
      print *,'starts; up to 8 different upper ages possible, fractions'
      print *,'of years permitted:ex. 1.5, 5.0, 8.2, etc.'
c
      if(nu.eq.0)read(*,*,err=5015)(plteay(l,in),l=1,8)
      if(nu.eq.1)read(7,*)(plteay(l,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5015) ans
      if(ans.gt.0.and.nu.eq.0)goto5015
      write(6,*)(plteay(l,in),l=1,8)
```

```
      if(iu.eq.1)print *,(plteay(l,in),l=1,8)
c
c ** male-female loop ***************************************************
      ms=mm
c excludes male or female humans if desired
      if (rr(in)) 5013,40,5013
c
c enters organ exposure selector, lung, liver, kidney, bone ..etc.
 5013 print *,'For zoon sex type',ms,'and zoon type',in
      print *,'1.=kidney,2.=bone,3.=liver,4.=lung,5.=reserved,'
      print *,'6.=bone endosteal cells'
c
      print *,'ENTER organ cancer selector gates(0.,1.) for 6 organs'
      if(nu.eq.0)read(*,*,err=5013)(orgrad(l,ms,in),l=1,6)
      if(nu.eq.1)read(7,*)(orgrad(l,ms,in),l=1,6)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5013) ans
      if(ans.gt.0.and.nu.eq.0)goto5013
      write(6,*)(orgrad(l,ms,in),l=1,6)
      if(iu.eq.1)print *,(orgrad(l,ms,in),l=1,6)
c
      do 47 k=1,6
      if(orgrad(k,ms,in).le.0.0) goto47
c linear and dose squared models specific cancer rates ,per 100000 persons
c
c for each organ considered enters  cancer expression period, yrs; upper age
c limit where a new expression period must be specified, years; latent period
c before cancer expression,years; and upper age where a new latent period
c must be specified(ALTYR)
c
c ** altnt(6,8)= latent period for expression of cancer; programmed for
c ** 6 possible cancer types and 8 possible latent periods may be entered
c
 5016 print *,'ENTER up to 8 different cancer latent periods for'
      print *,'cancer type',k,'; years; fractional years permitted; '
      print *,'ex. 1.4, 5.0, 7.9, etc.'
      if(nu.eq.0)read(*,*,err=5016)(altnt(k,l,in),l=1,8)
      if(nu.eq.1)read(7,*)(altnt(k,l,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,*)(altnt(k,l,in),l=1,8)
      if(iu.eq.1)print *,(altnt(k,l,in),l=1,8)
c
c ** altyr(6,8)=upper age by cancer type for which a new value for
c ** the latent period must be specified; 8 possible.
c
 5017 print *,'ENTER upper age where a new latent period starts'
      print *,'up to 8 different upper ages possible; and fractional'
      print *,'years permitted; 0.25, 1.8, 4.0, etc..'
c
      if(nu.eq.0)read(*,*,err=5017)(altyr(k,l,in),l=1,8)
      if(nu.eq.1)read(7,*)(altyr(k,l,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
```

```
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,*)(altyr(k,l,in),l=1,8)
      if(iu.eq.1)print *,(altyr(k,l,in),l=1,8)
c
c linear and dose squared models specific cancer rates ,per 100000 persons
c
c ** cmorta(6,8,2)=specific cancer rate due to contaminant by cancer
c ** type and sex for use with the linear model; 8 possible
c
 5018 print *,'ENTER up to 8 different linear model specific cancer'
      print *,'rates, per 100000 zoons for cancer type',k,'and'
      print *,'zoon group',ms
c
      if(nu.eq.0)read(*,*,err=5018) (cmorta(k,l,ms,in),l=1,8)
      if(nu.eq.1)read(7,*) (cmorta(k,l,ms,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
      write(6,*) (cmorta(k,l,ms,in),l=1,8)
      if(iu.eq.1)print *, (cmorta(k,l,ms,in),l=1,8)
c
c ** cmortb(6,8,2)=specific cancer rate due to contaminant by cancer
c ** type and sex for use with the dose squared model
c ** in conjunction with the linear model:y=a*d+b*d**2
c
 5019 print *,'ENTER up to 8 different dose squared model specific'
      print *,'cancer rates, per 100000 zoons for cancer type',k,'and'
      print *,'zoon group',ms
      if(nu.eq.0)read(*,*,err=5019)(cmortb(k,l,ms,in),l=1,8)
      if(nu.eq.1)read(7,*)(cmortb(k,l,ms,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
      write(6,*)(cmortb(k,l,ms,in),l=1,8)
      if(iu.eq.1)print *,(cmortb(k,l,ms,in),l=1,8)
c
c ** cmorty(6,8,2)=upper age by cancer type and sex for which a new
c ** value for the specific cancer rate must be specified
c
 5020 print *,'ENTER upper age where a new cancer rate applies'
      print *,'for cancer',k,'and zoon group',ms,';up to 8 possible;'
      print *,'fractional years permitted: ex. 0.66, 2.0, 5.2, etc.'
c
      if(nu.eq.0)read(*,*,err=5020)(cmorty(k,l,ms,in),l=1,8)
      if(nu.eq.1)read(7,*)(cmorty(k,l,ms,in),l=1,8)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
      write(6,*)(cmorty(k,l,ms,in),l=1,8)
      if(iu.eq.1)print *,(cmorty(k,l,ms,in),l=1,8)
c
c ** cnatr(6,18,2)=natural cancer rate(fatal) by sex and cancer type
c ** for every 100000 population; 18 age groups specified.
```

```
c
 5021 print *,'ENTER up to 18 different natural cancer rates(fatal)'
      print *,'per 100000 persons for cancer',k,'and zoon group',ms
c
      if(nu.eq.0)read(*,*,err=5021)(cnatr(k,l,ms,in),l=1,18)
      if(nu.eq.1)read(7,*)(cnatr(k,l,ms,in),l=1,18)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*)(cnatr(k,l,ms,in),l=1,18)
      if(iu.eq.1)print *,(cnatr(k,l,ms,in),l=1,18)
c
c ** cnaty(6,18,2)=upper age limit where a new natural cancer rate must
c ** be specified as above.
c
 5022 print *,'ENTER upper age where new natural cancer rate starts'
      print *,'for cancer',k,'and zoon group',ms,';18 upper ages'
      print *,'possible, and fractional years permitted; ex. 3.5, 6.0..'
c
      if(nu.eq.0)read(*,*,err=5022)(cnaty(k,l,ms,in),l=1,18)
      if(nu.eq.1)read(7,*)(cnaty(k,l,ms,in),l=1,18)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5022) ans
      if(ans.gt.0.and.nu.eq.0)goto5022
      write(6,*)(cnaty(k,l,ms,in),l=1,18)
      if(iu.eq.1)print *,(cnaty(k,l,ms,in),l=1,18)
   47 continue
c ** end organ loop ******************************************************
   40 continue
c
 5007 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,5)
      call wnoust('************************                    ')
      call wnoust('END OF INPUT IN EFFECTS                      ')
      call wnoust('************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for EFFECTS?       ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5007)ans
      if(ans) 11,5500,11
   11 rewind 6
      ict=0
      do5501l=1,20000
      read(6,*,err=5501)bcheck
      if(bcheck-919191.)5501,5503,5501
 5503 ict=ict+1
      if(ict.ne.iian)goto5501
```

```
      backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
      iage=agea(in)
      hasum(1,in)=0.0
      hasum(2,in)=0.0
      do18k=1,6
      sorgd(k,1,in)=0.0
      sorgb(k,1,in)=0.0
      sorgd(k,2,in)=0.0
      sorgb(k,2,in)=0.0
   18 continue
      do19k=1,iage
c ** a(kr,in,2)=fraction of total population in given age group and sex
      hasum(1,in)=hasum(1,in)+a(k,in,1)
      hasum(2,in)=hasum(2,in)+a(k,in,2)
   19 continue
c ** abscan(2)=total number of cancers estimated from absolute
c ** risk model by sex.
c
      abscan(1,in)=0.0
      abscan(2,in)=0.0
c ** relcan(2)=total number of cancers estimated from relative risk
c ** model by sex.
c
      relcan(1,in)=0.0
      relcan(2,in)=0.0
c ** spncan(2)=total number of spontaneous cancers exclusive of
c ** radiation by sex.
c
      spncan(1,in)=0.0
      spncan(2,in)=0.0
c ** ydos= number of years simulated(100 years maximum)
      idos=ydos(in)
      do330k=1,iage
      do331ky=1,idos
c ** tcn(200,kr,2)=total number of projected cancers due to absolute
c ** risk model by age and sex;100 years record possible
c
      tcn(ky,k,1,in)=0.0
c ** tcrel(200,kr,2)=total number of projected cancers due to rel. risk
c ** model by age and sex(female =2);100 years record possible
c
      tcrel(ky,k,1,in)=0.0
      tcn(ky,k,2,in)=0.0
      tcrel(ky,k,2,in)=0.0
  331 continue
  330 continue
      return
```

```
c
  190 continue
      if(ai1.lt.srun(in)) return
      if(amod(aiii,dmdd(in))-0.0) 191,192,191
  191 return
  192 iage=agea(in)
      rky=ai1-srun(in)
      ky=rky
      if(ky.le.0)ky=1
      aky=ky
      if(aky.gt.ydos(in))return
c ** tpop(kr,2,in)=population age structure by age and sex for use
c ** with either the absolute or relative risk model
      if(aiii.le.1.) popsum(mm,in)=0.0
      popsum(mm,in)=popsum(mm,in)+tpop(ii,mm,in)
      do405l=1,inz
      if(l.gt.1.and.l.lt.7)goto405
      j=in
      vkds=vkd1(ii,in,l)+vkd2(ii,in,l)
      vbns=vbn1(ii,in,l)+vbn2(ii,in,l)
      vlvs=vlv(ii,in,l)
      vlus=q1(ii,in,l)+q2(ii,in,l)+q3(ii,in,l)
      vndos=vbns*0.025
c ** orgdos(6,2)=dose in rads(or rems) by cancer type and sex sustained
c ** by all age cohorts in a given year
c
      orgdos(1,mm,in)=rkd(ii,in,l)
      orgbrd(1,mm,in)=vkds
      orgdos(2,mm,in)=rbn(ii,in,l)
      orgbrd(2,mm,in)=vbns
      orgdos(3,mm,in)=rlv(ii,in,l)
      orgbrd(3,mm,in)=vlvs
      orgdos(4,mm,in)=rlu(ii,in,l)
      orgbrd(4,mm,in)=vlus
      orgdos(5,mm,in)=0.0
      orgbrd(5,mm,in)=0.0
      orgdos(6,mm,in)=rndo(ii,in,l)
      orgbrd(6,mm,in)=vndos
      ak=ii
      akd=(ak*365.+aiii)/365.
      do352jo=1,6
      sorgd(jo,mm,in)=sorgd(jo,mm,in)+orgdos(jo,mm,in)*tpop(ii,mm,in)
      sorgb(jo,mm,in)=sorgb(jo,mm,in)+orgbrd(jo,mm,in)*tpop(ii,mm,in)
      if(orgrad(jo,mm,in).le.0.0)goto352
c     cncmrt= specific canca and cancb under consideration for a given
c     age group,cancer, and sex.
      cncmrt=0.0
c     tcnrs=sum of relative  risk cancers
      tcnrs=0.0
c     tcnatr=sum of natural cancers
      tcnatr=0.0
c     relrsk= relative risk factor
      relrsk=0.0
      alatnt=altnt(jo,1,in)
```

```
c ** canca is the cancer rate/rem-person-year for linear model
c ** cancb is the cancer rate/rem-person-year for linear-quadratic model
c
      canca=cmorta(jo,1,mm,in)*1.0e-06
      cancb=cmortb(jo,1,mm,in)*1.0e-06
c     cancnk is the natural cancer rate/rem-person-year
      dfr=1.0
      if(in.le.nan)dfr=(70000.**(0.33))/((1000.*rr(in))**(0.33))
      cancnk=cnatr(jo,1,mm,in)*1.0e-05*dfr
      plteau=pltea(1,in)
      do355js=1,7
      if(plteay(js,in).le.0.0)goto359
      if(akd.gt.plteay(js,in))plteau=pltea(js+1,in)
  359 if(altyr(jo,js,in).le.0.0)goto354
      if(akd.gt.altyr(jo,js,in))alatnt=altnt(jo,js+1,in)
  354 if(cmorty(jo,js,mm,in).le.0.0)goto353
      if(akd.gt.cmorty(jo,js,mm,in))canca=cmorta(jo,js+1,mm,in)*1.0e-06
      if(akd.gt.cmorty(jo,js,mm,in))cancb=cmortb(jo,js+1,mm,in)*1.0e-06
  353 if(cnaty(jo,js,mm,in).le.0.0)goto355
      if(akd.gt.cnaty(jo,js,mm,in))cancnk=cnatr(jo,js+1,mm,in)*1.0e-05
  355 continue
c ** tcnat(kr,2)= total spontaneous cancers produced by other than
c ** ionizing radiation for a given age and sex.
c
      tcnat(ii,mm,in)=cancnk*tpop(ii,mm,in)*dmdd(in)/365.
      aip=alatnt+plteau+akd
      ail=alatnt+akd
      do307iis=1,iage
      bss=iis
      ass=(bss*365.+aiii)/365.
      kp=ky+iis-ii
      if(kp.gt.idos)goto307
      if(kp.lt.1)goto307
      dfr=1.0
      if(in.le.nan)dfr=(70000.**(0.33))/((1000.*rr(in))**(0.33))
      goto(361,362,363,365,366,368),jo
  361 cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
      goto367
  362 cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
      goto367
  363 continue
      cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
      goto367
  365 continue
      cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
      goto367
  366 continue
      cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
      goto367
  368 continue
      cncmrt=(canca*orgdos(jo,mm,in)+cancb*orgdos(jo,mm,in)**2)*dfr
  367 continue
      if(ass.lt.ail.or.ass.gt.aip)goto364
      tcn(kp,iis,mm,in)=tcn(kp,iis,mm,in)+cncmrt*dmdd(in)/365.
```

```
 364 if(cancnk.le.0.0)goto356
     dfr=1.0
     if(in.le.nan)dfr=(70000.**(0.33))/((1000.*rr(in))**(0.33))
c    cancni is the projected natural cancer rate/rem-person-year
     cancni=cnatr(jo,1,mm,in)*1.0e-05*dfr
     do372js=1,17
     if(cnaty(jo,js,mm,in).le.0.0)goto372
     if(ass.gt.cnaty(jo,js,mm,in))cancni=cnatr(jo,js+1,mm,in)*1.0e-05
    c*dfr
 372 continue
     if(ass.lt.alatnt+akd)goto356
     yrfc=1.0
     if(in.le.nan)yrfc=((1000.*rr(in))**(1./3.))/(70000.**(1./3.))
     goto(45,43,41,42,41,43),jo
  45 if(ass.gt.akd+30*yrfc)goto357
     goto50
  41 if(ass.gt.akd+45*yrfc)goto357
     goto50
  42 if(ass.gt.akd+25*yrfc)goto357
     goto50
  43 if(ass.gt.akd+24*yrfc)goto357
  50 tcrel(kp,iis,mm,in)=tcrel(kp,iis,mm,in)+cncmrt*dmdd(in)/365.
     tcnrs=tcnrs+cncmrt
     tcnatr=tcnatr+cancni
     relrsk=tcnrs/tcnatr
     goto356
 357 continue
     tcrel(kp,iis,mm,in)=tcrel(kp,iis,mm,in)+relrsk*cancni*dmdd(in)
    1/365.
 356 continue
 307 continue
     prem=orgdos(jo,mm,in)*tpop(ii,mm,in)
     orgdsm=orgdos(jo,mm,in)
 352 continue
c
c ** tqx(kr,2,in)=mortality rate by age and sex for population.
c
     tqxp=tqx(ii,mm,in)*tpop(ii,mm,in)*popfr(in)
     tpopp=tpop(ii,mm,in)*popfr(in)
     tcnats=cancnk*tpop(ii,mm,in)*popfr(in)
     tcnky=tcn(ky,ii,mm,in)*tpop(ii,mm,in)*popfr(in)
     tcrelk=tcrel(ky,ii,mm,in)*tpop(ii,mm,in)*popfr(in)
     spncan(mm,in)=spncan(mm,in)+tcnats
     abscan(mm,in)=abscan(mm,in)+tcnky
     relcan(mm,in)=relcan(mm,in)+tcrelk
c
     print *,aky,akd,orgdsm,prem,tpopp,tcnky,tqxp,tcrelk,tcnats
c
 405 continue
     if(aiii.lt.365.) return
     do371mm=1,2
     do374k=1,iage
c ** adjusts population for cancer deaths due to either absolute or
c ** relative risk estimates
```

```
c
      totcan=tcn(ky,k,mm,in)
      if(absr(in).gt.1.0)totcan=tcrel(ky,k,mm,in)
      tpop(k,mm,in)=tpop(k,mm,in)-totcan*tpop(k,mm,in)
c
  374 continue
  371 continue
      return
      end
```

14.21. TRUPLT (bum.for)

```
      subroutine truplt (krec,k,dvar,lx,nplts,iflag,rdidx)
c
      parameter (kp=8003,kv=8,kz=8)
c
      include 'main.inc'
c
      common/uv/ xpp(kp,kz),ypp(kp,kz),npts(kv,kz),varplt(kz),nobs(kz)
     1,xpx(kp,kz),srec(kz,kv)
c
      dimension xno(kv,kz),dvar(20),selct3(kv,kz),selct1(kv,kz)
     1,selct2(kv,kz),xvar(kz)
c
      character*1 iflag(14,kz)
      if (rdidx.gt.0.) go to 30
      if (lx.gt.1) return
c
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
      ict=0
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-171717.)5702,5704,5702
 5555 nu=0
 5704 ict=ict+1
      if(ict.eq.nplts)goto5050
      goto5702
 5050 call atcol('white','cyan')
      call wnclos(1)
      call wnopen(0,0,51,7)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE TRUPLT                    ')
      call wnoust('                                                 ')
      call wnoust('                                                 ')
      call wnoust('                                                 ')
      call wnoust('****************************                    ')
      call wncuxy(1,3)
      print *,'This program  will take output from BIOTRAN.2,'
      call wncuxy(1,4)
      print *,'and select and plot 2-D graphics for the user'
      call wncuxy(1,5)
      print *,'using 2-d graphics software'
      call wnouce(7,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
```

```
      call wnoust('                                        ')
      call wnoust('ENTER yes=1, no=0                        ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 171717.                  ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0       ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
c ** User selection input ***
c
 5001 print *,'ENTER the number of variables to be graphed on one plot'
      print *,'a maximum of 8 variables may be specified; ENTER now'
      if(nu.eq.0)read(*,*,err=5001) varplt(k)
      if(nu.eq.1)read(7,*) varplt(k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
      write(6,*)varplt(k)
      if(iu.eq.1) print *, varplt(k)
      nvrplt=varplt(k)
 5000 print *,'ENTER the record number selected for plotting..1,2,..etc'
      print *,'for each of the',nvrplt,'variables of plot=',k
      if(nu.eq.0)read(*,*,err=5000)(srec(k,jv),jv=1,nvrplt)
      if(nu.eq.1)read(7,*) (srec(k,jv),jv=1,nvrplt)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5000) ans
      if(ans.gt.0.and.nu.eq.0)goto5000
```

```
      write(6,*) (srec(k,jv),jv=1,nvrplt)
      if(iu.eq.1) print *, (srec(k,jv),jv=1,nvrplt)
c
c     xvar(k) is any column value or can be set to 13 for time
c     as the  x-variable
c
 5002 print *,'ENTER the row to be used as the x-axis: a value'
      print *,'of 13 designates time for this axis(most common)'
      if(nu.eq.0)read(*,*,err=5002)xvar(k)
      if(nu.eq.1)read(7,*)xvar(k)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5002) ans
      if(ans.gt.0.and.nu.eq.0)goto5002
      write(6,*)xvar(k)
      if(iu.eq.1) print *, xvar(k)
c
      print *,'the x-axis is row',xvar(k)
c
      nvrplt=varplt(k)
c
 5003 print *,'ENTER the column number(s) for',nvrplt,'variables'
      print *,'selected from a specific record of plot=',k
      if(nu.eq.0)read(*,*,err=5003) (xno(j,k),j=1,nvrplt)
      if(nu.eq.1)read(7,*) (xno(j,k),j=1,nvrplt)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,*) (xno(j,k),j=1,nvrplt)
      if(iu.eq.1) print *, (xno(j,k),j=1,nvrplt)
c
c     selct1 is equal to dvar(14),or the 14th variable from BIOTRAN.2
c     output setting the value to zero overides its control
c     selct1 is set to zero when a variable does not have specific
c     identifier
c
 5004 print *,'The value of the 14th var. from BIOTRAN.2 output can be'
      print *,'used to select a number of specific variables: plant'
      print *,'type, and animal age; the default value (0.0) will'
      print *,'override selection control for this option'
      print *,'ENTER selection(s) for',nvrplt,'variables now'
c
      if(nu.eq.0)read(*,*,err=5004) (selct1(j,k),j=1,nvrplt)
      if(nu.eq.1)read(7,*) (selct1(j,k),j=1,nvrplt)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,*) (selct1(j,k),j=1,nvrplt)
      if(iu.eq.1) print *, (selct1(j,k),j=1,nvrplt)
c
 5005 print *, 'the value of the 11th var. is an additional selector'
      print *,'which can be used to select a  specific soil layer,'
      print *,'tree dia., water lyr, contaminant, and plant organ dose,'
      print *,'the default 0.0 will override selection control'
c
```

```
      print *,'ENTER selection(s) for',nvrplt,'variables now'
      if(nu.eq.0)read(*,*,err=5005) (selct2(j,k),j=1,nvrplt)
      if(nu.eq.1)read(7,*) (selct2(j,k),j=1,nvrplt)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,*) (selct2(j,k),j=1,nvrplt)
      if(iu.eq.1) print *, (selct2(j,k),j=1,nvrplt)
c
c     selct2(j,k) is equal to dvar(11), or the 11th variable from the
c     bioplt output. setting the value to zero overrides its control
c     selct2 is needed when selecting particular soil layers within
c     particular plants. selct1 selects the plant and selct2
c     selects the layer.
c
c     selct3 is entered for array output of data and selects a
c     specfic: region, or animal type for output.  setting the
c     the value to 0.0 will override its control
c
 5006 print *,'a specific region or animal type can be selected by'
      print *,'comparison of entered values with the 15th column of'
      print *,'output; the default 0.0 overrides control'
      print *,'ENTER selct3 selector(s) for',nvrplt,'variables now'
c
      if(nu.eq.0)read(*,*,err=5006) (selct3(j,k),j=1,nvrplt)
      if(nu.eq.1)read(7,*) (selct3(j,k),j=1,nvrplt)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5006) ans
      if(ans.gt.0.and.nu.eq.0)goto5006
      write(6,*) (selct3(j,k),j=1,nvrplt)
      if(iu.eq.1) print *, (selct3(j,k),j=1,nvrplt)
c
 5012 if(nu.eq.1)goto5500
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN TRUPLT                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for TRUPLT?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      read(*,*,err=5012)ans
      if(ans)11,5500,11
   11 rewind 6
      ict=0
      do5501l=1,20000
      read(6,*,err=5501)bcheck
      if(bcheck-171717.)5501,5503,5501
```

```
 5503 ict=ict+1
      if(ict.ne.nplts)goto5501
      backspace 6
      nu=nuu
      call clear_screen@
      goto 5050
 5501 continue
 5500 continue
      nu=nuu
      call clear_screen@
      nvrplt=varplt(k)
      nobs(k)=0
      do406j=1,nvrplt
      npts(j,k)=0.0
  406 continue
      do407jo=1,kp
      xpx(jo,k)=0.0
  407 continue
c
      return
c
   30 continue
c
      idv=dvar(12)
      nvrplt=varplt(k)
      ixv=xvar(k)
c
      do20j=1,nvrplt
      jrec=srec(k,j)
      if(idv.ne.jrec)goto20
c
      if(selct1(j,k).le.0.0)goto21
      if(selct1(j,k)-dvar(14)) 20,21,20
c
   21 if(selct2(j,k).le.0.0)goto209
      if(selct2(j,k)-dvar(11)) 20,209,20
c
  209 if(selct3(j,k).le.0.0)goto310
      if(dvar(15)-selct3(j,k)) 20,310,20
c
  310 iyv=xno(j,k)
c
      npts(j,k)=npts(j,k)+1
      nobsk=nobs(k)+1
      xpp(nobsk,k)=dvar(ixv)
      ypp(nobsk,k)=dvar(iyv)
      aj=j
      xpx(nobsk,k)=aj
      nobs(k)=nobs(k)+1
   20 continue
c
      return
      end
```

14.22. TPLOTR (bvm.for)

```
      subroutine tplotr(krec,jp,lx,ngrap,iflag,rdidx)
c
      parameter (kp=8003,kv=8,kz=8)
c
      include 'main.inc'
c
      common/uv/ xpp(kp,kz),ypp(kp,kz),npts(kv,kz),varplt(kz),nobs(kz)
     1,xpx(kp,kz),srec(kz,kv)
c
      dimension itype(kv,kz),colorh(kv,kz),colors(kv,kz),colora(kv,kz)
      dimension nxl(kz),nyl(kz),ntit(kz),xpxx(kp),ypyy(kp)
c
      character*40 ixl(kz),ltitle(kz),iyl(kz)
      character*1 iflag(14,kz)
      character*12 istl(kv,kz)
c
    4 format(a)
c
      if (rdidx.gt.0.) go to 30
      if (lx.gt.1) return
      nuu=nu
      if(nu.eq.0)goto5050
      rewind 7
      ict=0
 5702 continue
      read(7,*,err=5702,end=5555)acheck
      if(acheck-191919.)5702,5704,5702
 5555 nu=0
 5704 ict=ict+1
      if(ict.eq.ngrap)goto5050
      goto5702
 5050 call atcol('white','magenta')
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('****************************                    ')
      call wnoust(' YOU ARE IN SUBROUTINE TPLOTR                    ')
      call wnoust('****************************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
c the user may change mode of input for this subroutine
c
 5700 call wnclos(1)
      call wnopen(0,0,51,8)
      call wnoust('*****************************************************')
      call wnoust('1=file input, 0=keyboard input: you have selected  ')
      call wnoust('                                                   ')
      call wnoust('ENTER yes=1, no=0                                  ')
      call wnoust('*****************************************************')
      call wncuxy(1,3)
      print *,'mode=',nu,'; want to reverse temporarily?'
```

```
      call wnouce(7,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,8)
      read(*,*,err=5700)ans
      if(ans.le.0.0)goto5701
      nux=nu
      if(nux.eq.0)nu=1
      if(nux.eq.1)nu=0
 5701 call wnclos(1)
      if(nu.eq.1)goto5706
      call wnopen(0,0,51,4)
      call wnoust('ENTER subroutine code: 191919.                    ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701)acheck
      call wncuxy(1,4)
      if(iu.eq.1)print *,'value(s)=',acheck
      if(iu.eq.1)call sleep@(5.0)
      call wnclos(1)
      call wnopen(0,0,51,3)
      call wnoust('do you wish to re-enter line? ENTER y=1, n=0      ')
      call wnouce(2,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,3)
      read(*,*,err=5701) ans
      if(ans.gt.0.0)goto5701
 5706 write(6,*)acheck
      if(iu.eq.1.and.nu.eq.1)print *,acheck
c
 5000 print *,'ENTER number of columns in x-axis label,'
      print *,'y-axis label, and title'
      print *,'label: ENTER the 3 values now (separate'
      print *,'by spaces or commas)'
c
      if(nu.eq.0)read(*,*,err=5000) nxl(jp),nyl(jp),ntit(jp)
      if(nu.eq. 1)read(7,*) nxl(jp),nyl(jp),ntit(jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5000) ans
      if(ans.gt.0.and.nu.eq.0)goto5000
      write(6,*) nxl(jp),nyl(jp),ntit(jp)
      if(iu.eq.1) print *,nxl(jp),nyl(jp),ntit(jp)
c
      ipy=varplt(jp)
      print *,'ipy',ipy,varplt(jp),jp
c
 5001 print *,'ENTER type of line markers through data points for',ipy
      print *,'variables;0=none ;1=digits ; 3=letters; 4=symbols'
      print *,'ENTER ',ipy,'selection(s) now'
      if(nu.eq.0)read(*,*,err=5001) (itype(k,jp),k=1,ipy)
      if(nu.eq.1)read(7,*) (itype(k,jp),k=1,ipy)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5001) ans
      if(ans.gt.0.and.nu.eq.0)goto5001
```

```
      write(6,*) (itype(k,jp),k=1,ipy)
      if(iu.eq.1) print *, (itype(k,jp),k=1,ipy)
c
      print *,' ****** prepare to enter 12 plotting options ******'
c
 5003 print *,'Do you want color graphics? y=yes,n=no; Enter y or n now'
      if(nu.eq.0)read(*,*,err=5003) iflag(2,jp)
      if(nu.eq.1)read(7,4) iflag(2,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5003) ans
      if(ans.gt.0.and.nu.eq.0)goto5003
      write(6,4) iflag(2,jp)
      if(iu.eq.1) print *, iflag(2,jp)
c
 5004 print *,'Do you want a legend? y=yes,n=no;ENTER y or n now'
      if(nu.eq.0)read(*,*,err=5004)iflag(3,jp)
      if(nu.eq.1)read(7,4)iflag(3,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5004) ans
      if(ans.gt.0.and.nu.eq.0)goto5004
      write(6,4)iflag(3,jp)
      if(iu.eq.1) print *, iflag(3,jp)
c
 5005 print *,'Specifiy log y-axis? y=yes,n=no;ENTER y or n now'
      if(nu.eq.0)read(*,*,err=5005)iflag(4,jp)
      if(nu.eq.1)read(7,4)iflag(4,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5005) ans
      if(ans.gt.0.and.nu.eq.0)goto5005
      write(6,4)iflag(4,jp)
      if(iu.eq.1) print *, iflag(4,jp)
c
 5007 print *,'Specify log x-axis? y=yes,n=no;ENTER y or n now'
      if(nu.eq.0)read(*,*,err=5007)iflag(6,jp)
      if(nu.eq.1)read(7,4)iflag(6,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5007) ans
      if(ans.gt.0.and.nu.eq.0)goto5007
      write(6,4) iflag(6,jp)
      if(iu.eq.1) print *, iflag(6,jp)
c
 5016 print *,'ENTER title for x-axis'
      if(nu.eq.0)read(*,*,err=5016) ixl(jp)
      if(nu.eq.1)read(7,4) ixl(jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5016) ans
      if(ans.gt.0.and.nu.eq.0)goto5016
      write(6,4) ixl(jp)
      if(iu.eq.1) print *, ixl(jp)
c
 5017 print *,'ENTER title for y-axis'
      if(nu.eq.0)read(*,*,err=5017) iyl(jp)
      if(nu.eq.1)read(7,4) iyl(jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
```

```
      if(nu.eq.0)read(*,*,err=5017) ans
      if(ans.gt.0.and.nu.eq.0)goto5017
      write(6,4) iyl(jp)
      if(iu.eq.1) print *, iyl(jp)
c
 5018 print *,'ENTER title for plot'
      if(nu.eq.0)read(*,*,err=5018) ltitle(jp)
      if(nu.eq.1)read(7,4) ltitle(jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5018) ans
      if(ans.gt.0.and.nu.eq.0)goto5018
      write(6,4) ltitle(jp)
      if(iu.eq.1) print *, ltitle(jp)
c
      if(iflag(3,jp).eq.'n')goto5020
      print *,'ENTER legend title(s) for',ipy,' y-variables'
      do6j=1,ipy
 5019 if(nu.eq.0)print *,'you are entering title for variable=',j
      if(nu.eq.0)print *,'title is limited to 12 characters'
      if(nu.eq.0)read(*,*,err=5019) istl(j,jp)
      if(nu.eq.1)read(7,4) istl(j,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5019) ans
      if(ans.gt.0.and.nu.eq.0)goto5019
    6 continue
      do7j=1,ipy
      write(6,4) istl(j,jp)
      if(iu.eq.1)print *, istl(j,jp)
    7 continue
c
 5020 print *,'Specify x-axis minimum value, and x-axis maximum value'
      print *,'a value of -111. for any value will cause code'
      print *,'calculation by default'
      print *,'ENTER these 2 parameters now'
      if(nu.eq.0)read(*,*,err=5020) xpp(kp,jp),xpp(kp-1,jp)
      if(nu.eq.1)read(7,*) xpp(kp,jp),xpp(kp-1,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5020) ans
      if(ans.gt.0.and.nu.eq.0)goto5020
      write(6,*) xpp(kp,jp),xpp(kp-1,jp)
      if(iu.eq.1) print *, xpp(kp,jp),xpp(kp-1,jp)
c
 5021 print *,'Specify these two parameters for y-axis as above'
      print *,'ENTER these 2 parameters for the y-axis now'
      if(nu.eq.0)read(*,*,err=5021) ypp(kp,jp),ypp(kp-1,jp)
      if(nu.eq.1)read(7,*) ypp(kp,jp),ypp(kp-1,jp)
      if(nu.eq.0)print *,'do you wish to re-enter line? ENTER y=1, n=0'
      if(nu.eq.0)read(*,*,err=5021) ans
      if(ans.gt.0.and.nu.eq.0)goto5021
      write(6,*) ypp(kp,jp),ypp(kp-1,jp)
      if(iu.eq.1) print *,ypp(kp,jp),ypp(kp-1,jp)
c
 5012 if(nu.eq.1)goto5500
      call wnclos(1)
```

```
      call wnopen(0,0,51,4)
      call wnoust('**********************                    ')
      call wnoust('END OF INPUT IN TPLOTR                      ')
      call wnoust('**********************                    ')
      call wnouce(4,'press any key')
      call inkey(key)
      call wnclos(1)
      call wnopen(0,0,51,4)
      call wnoust('Do you wish to RE-ENTER input for TPLOTR?        ')
      call wnoust('yes: ENTER 1.0; no: ENTER 0.0; ENTER now         ')
      call wnouce(3,'press any key and enter above')
      call inkey(key)
      call wncuxy(1,4)
      print *,'yes: ENTER 1.0; no: ENTER 0.0; ENTER now'
      read(*,*,err=5012)ans
      if(ans)11,5500,11
   11 rewind 6
      ict=0
      do5501l=1,20000
      read(6,*,err=5501)acheck
      if(acheck-191919.)5501,5504,5501
 5504 ict=ict+1
      if(ict.eq.ngrap)goto5503
      goto5501
 5503 backspace 6
      nu=nuu
      call clear_screen@
      goto5050
 5501 continue
 5500 continue
      do311jo=1,kp
      xpxx(jo)=0.0
      ypyy(jo)=0.0
  311 continue
      nu=nuu
      call clear_screen@
      return
   30 continue
      nvrplt=varplt(jp)
      nobsk=nobs(jp)
      ntitc=33-ntit(jp)/2
      nxlc=33-nxl(jp)/2
      nylc=33-nyl(jp)/2
      do312jo=1,kp
      xpxx(jo)=0.0
      ypyy(jo)=0.0
  312 continue
      iv=1
      do32jv=1,nvrplt
      npts(jv,jp)=0
      do31jo=1,nobsk
      ixpx=xpx(jo,jp)
      if(ixpx.ne.jv)goto31
      if(xpp(jo,jp).le.0.0.or.ypp(jo,jp).le.0.0)goto31
```

```
      xpxx(iv)=xpp(jo,jp)
      ypyy(iv)=ypp(jo,jp)
      iv=iv+1
      npts(jv,jp)=npts(jv,jp)+1
   31 continue
   32 continue
      write(8,*) (npts(jv,jp),jv=1,nvrplt)
      nptst=1
      nobs(jp)=0
      nobjv=0
      do35jv=1,nvrplt
      nobjv=npts(jv,jp)+nobjv
      do36jo=nptst,nobjv
      nobs(jp)=nobs(jp)+1
      nobsk=nobs(jp)
      ypyy(nobsk)=ypyy(jo)
      xpxx(nobsk)=xpxx(jo)
      write(8,*) xpxx(nobsk),ypyy(nobsk),jv,jo,nobsk
   36 continue
      nptst=nobjv+1
   35 continue
c
      call scinit(' ')
      if(infohw(20).eq.0)call iidisp(1)
      if(infogr(1).eq.0)call osexit('graphics not available',21)
      call grinit('C ',640,450,16)
      goto (301,302,303,304,305,306,307,308),jp
  301 call ghcopy('plot1.plt')
      goto309
  302 call ghcopy('plot2.plt')
      goto309
  303 call ghcopy('plot3.plt')
      goto309
  304 call ghcopy('plot4.plt')
      goto309
  305 call ghcopy('plot5.plt')
      goto309
  306 call ghcopy('plot6.plt')
      goto309
  307 call ghcopy('plot7.plt')
      goto309
  308 call ghcopy('plot8.plt')
  309 continue
      call garea(0.,0.,1.0,1.0)
      call gunit(0.0,0.0,100.,100.)
c
      call ezplot(jp,e,nvrplt,itype,iflag,ltitle,isyb,
     1ixl,iyl,istl,ierror,colorh,colors,colora,nxlc,nylc,ntitc,xpxx,
     2ypyy)
c
      call gpause(' ')
      call grquit
      call scquit('c')
      return
```

```
      end
      subroutine ezplot (jp,e,mlines,itype,iflag,ltitle,isyb,ixl,iyl
     1 ,istl,ierror,colorh,colors,colora,nxlc,nylc,ntitc,xpxx,ypyy)
      parameter (kp=8003,kv=8,kz=8)
      common/uv/ xpp(kp,kz),ypp(kp,kz),npts(kv,kz),varplt(kz),nobs(kz)
     1,xpx(kp,kz),srec(kz,kv)
      dimension itype(kv,kz),colors(kv,kz),x1(kp),y1(kp),xpxx(kp),
     1ypyy(kp),colora(kv,kz),colorh(kv,kz)
      character*40 ixl(kz),iyl(kz),ltitle(kz)
      character*60 ltitl,ixll,iyll
      character*1 iflag(14,kz)
      character*12 istl(kv,kz)
      character*(*) b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,
     1b16,b17,b18,b19,b20
      parameter (b1=' ',b2=' ',b3=' ',b4=' ',b5=' ',
     1b6=' ',b7=' ',b8=' ',b9=' ',
     2b10=' ',b11=' ',b12=' ',
     3b13=' ',b14=' ',b15=' ',
     4b16=' ',b17=' ',
     5b18=' ',b19=' ',
     6b20=' ')
      if (mlines.le.0) return
      goto (601,602,603,604,605,606,607,608,609,610,611,612,613,614,
     1615,616,617,618,619,620), ntitc
620 ltitl=b20//ltitle(jp)
      goto600
619 ltitl=b19//ltitle(jp)
      goto600
618 ltitl=b18//ltitle(jp)
      goto600
617 ltitl=b17//ltitle(jp)
      goto600
616 ltitl=b16//ltitle(jp)
      goto600
615 ltitl=b15//ltitle(jp)
      goto600
614 ltitl=b14//ltitle(jp)
      goto600
613 ltitl=b13//ltitle(jp)
      goto600
612 ltitl=b12//ltitle(jp)
      goto600
611 ltitl=b11//ltitle(jp)
      goto600
610 ltitl=b10//ltitle(jp)
      goto600
609 ltitl=b9//ltitle(jp)
      goto600
608 ltitl=b8//ltitle(jp)
      goto600
607 ltitl=b7//ltitle(jp)
      goto600
606 ltitl=b6//ltitle(jp)
      goto600
```

```
605 ltitl=b5//ltitle(jp)
   goto600
604 ltitl=b4//ltitle(jp)
   goto600
603 ltitl=b3//ltitle(jp)
   goto600
602 ltitl=b2//ltitle(jp)
   goto600
601 ltitl=b1//ltitle(jp)
600 goto (701,702,703,704,705,706,707,708,709,710,711,712,713,714,
  1715,716,717,718,719,720), nxlc
720 ixll=b20// ixl(jp)
   goto700
719 ixll=b19//ixl(jp)
   goto700
718 ixll=b18//ixl(jp)
   goto700
717 ixll=b17//ixl(jp)
   goto700
716 ixll=b16//ixl(jp)
   goto700
715 ixll=b15//ixl(jp)
   goto700
714 ixll=b14//ixl(jp)
   goto700
713 ixll=b13//ixl(jp)
   goto700
712 ixll=b12//ixl(jp)
   goto700
711 ixll=b11//ixl(jp)
   goto700
710 ixll=b10//ixl(jp)
   goto700
709 ixll=b9//ixl(jp)
   goto700
708 ixll=b8//ixl(jp)
   goto700
707 ixll=b7//ixl(jp)
   goto700
706 ixll=b6//ixl(jp)
   goto700
705 ixll=b5//ixl(jp)
   goto700
704 ixll=b4//ixl(jp)
   goto700
703 ixll=b3//ixl(jp)
   goto700
702 ixll=b2//ixl(jp)
   goto700
701 ixll=b1//ixl(jp)
700 goto (801,802,803,804,805,806,807,808,809,810,811,812,813,814,
  1815,816,817,818,819,820), nylc
820 iyll=b20//iyl(jp)
   goto800
```

```
819 iyll=b19//iyl(jp)
   goto800
818 iyll=b18//iyl(jp)
   goto800
817 iyll=b17//iyl(jp)
   goto800
816 iyll=b16//iyl(jp)
   goto800
815 iyll=b15//iyl(jp)
   goto800
814 iyll=b14//iyl(jp)
   goto800
813 iyll=b13//iyl(jp)
   goto800
812 iyll=b12//iyl(jp)
   goto800
811 iyll=b11//iyl(jp)
   goto800
810 iyll=b10//iyl(jp)
   goto800
809 iyll=b9//iyl(jp)
   goto800
808 iyll=b8//iyl(jp)
   goto800
807 iyll=b7//iyl(jp)
   goto800
806 iyll=b6//iyl(jp)
   goto800
805 iyll=b5//iyl(jp)
   goto800
804 iyll=b4//iyl(jp)
   goto800
803 iyll=b3//iyl(jp)
   goto800
802 iyll=b2//iyl(jp)
   goto800
801 iyll=b1//iyl(jp)
800 nobsk=nobs(jp)
c
   call pgarea(0.1,0.1,0.8,0.8)
   call pgraph(mlines,nobsk,' ',' ','x')
c
   if(iflag(4,jp).eq.'n'.and.iflag(6,jp).eq.'n')
  1 call pgscal('lin','lin')
   if(iflag(4,jp).eq.'y'.and.iflag(6,jp).eq.'y')
  1 call pgscal('log','log')
   if(iflag(4,jp).eq.'y'.and.iflag(6,jp).eq.'n')
  1 call pgscal('lin','log')
   if(iflag(4,jp).eq.'n'.and.iflag(6,jp).eq.'y')
  1 call pgscal('log','lin')
   call armm1(ypyy,nobsk,ymin,ymax)
   if(ypp(kp,jp).gt.0.0)ymin=ypp(kp,jp)
   if(ypp(kp-1,jp).gt.0.0)ymax=xpp(kp-1,jp)
   call armm1(xpxx,nobsk,xmin,xmax)
```

```
      if(xpp(kp,jp).gt.0.0)xmin=xpp(kp,jp)
      if(xpp(kp-1,jp).gt.0.0)xmax=xpp(kp-1,jp)
      xmax=xmax/0.90
      xmin=xmin*0.95
      ymax=ymax/0.90
      ymin=ymin*0.95
      call pgunit(xmin,ymin,xmax,ymax)
      call pgaxes
      call pgxscl(' ')
      call pgyscl(' ')
c
c
      call gchsiz(1.0,1.5)
      call pgtitl(ltitl,'L')
      call gchsiz(1.0,1.0)
      call pgxlbl(ixll,'L')
      call pgylbl(iyll,'C')
      call gchsiz(0.5,0.5)
      call gsyset('general.smb')
c
      alines=mlines
      jcol1=0
      jcol2=0
      do311jo=1,kp
      x1(jo)=0.0
      y1(jo)=0.0
  311 continue
      npt=0
      nps=0
      do30jv=1,mlines
      npt=npts(jv,jp)+npt
      lsb=itype(jv,jp)
      lpt=0
      do39jpt=1,kp
      if(jpt.le.nps)goto39
      kpt=jpt
      lpt=lpt+1
      if(jpt.gt.npt)kpt=npt
      x1(lpt)=xpxx(kpt)
      y1(lpt)=ypyy(kpt)
      if(jpt.gt.npt)y1(lpt)=ymin
   39 continue
      nps=npts(jv,jp)+nps
      goto (31,32,33,34,35,36,37,38), jv
   31 if(iflag(2,jp).eq.'y')jcol1=40
      if(iflag(2,jp).eq.'y')jcol2=40
      call pgstyl(1,0,lsb,0,jcol1,jcol2)
      if(lsb.eq.4)call gsyoua(x1,y1,23)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=26.0
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   32 if(iflag(2,jp).eq.'y')jcol1=0
```

```
      if(iflag(2,jp).eq.'y')jcol2=0
      call pgstyl(2,1,lsb,0,jcol1,jcol2)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=24.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   33 if(iflag(2,jp).eq.'y')jcol1=95
      if(iflag(2,jp).eq.'y')jcol2=95
      call pgstyl(3,2,lsb,0,jcol1,jcol2)
      if(lsb.eq.4)call gsyoua(x1,y1,15)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=22.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   34 if(iflag(2,jp).eq.'y')jcol1=127
      if(iflag(2,jp).eq.'y')jcol2=127
      call pgstyl(4,3,lsb,0,jcol1,jcol2)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=20.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   35 if(iflag(2,jp).eq.'y')jcol1=159
      if(iflag(2,jp).eq.'y')jcol2=159
      call pgstyl(5,4,lsb,0,jcol1,jcol2)
      if(lsb.eq.4)call gsyoua(x1,y1,1)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=18.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   36 if(iflag(2,jp).eq.'y')jcol1=191
      if(iflag(2,jp).eq.'y')jcol2=191
      call pgstyl(6,4,lsb,0,jcol1,jcol2)
      if(lsb.eq.4)call gsyoua(x1,y1,11)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=16.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   37 if(iflag(2,jp).eq.'y')jcol1=170
      if(iflag(2,jp).eq.'y')jcol2=170
      call pgstyl(7,4,lsb,0,jcol1,jcol2)
      if(lsb.eq.4)call gsyoua(x1,y1,13)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=14.
      call pgkey1(jv,75.,ap,istl(jv,jp))
      goto30
   38 if(iflag(2,jp).eq.'y')jcol1=20
      if(iflag(2,jp).eq.'y')jcol2=20
      call pgstyl(8,4,lsb,0,jcol1,jcol2)
```

```
      if(lsb.eq.4)call gsyoua(x1,y1,2)
      call pgxypr(x1,y1)
      if(iflag(3,jp).eq.'n')goto30
      ap=12.
      call pgkey1(jv,75.,ap,istl(jv,jp))
 30 continue
      do312jo=1,kp
      x1(jo)=0.0
      y1(jo)=0.0
312 continue
      call sleep@(5.0)
      return
      end
```

14.23. MAIN.INC

```
c**** i=number of plant types, ks=number of soil layers
c**** kt=number of tree diameters, kw=number of water layers
c**** ka= number of exposure-years for plants, kr=exposure age
c**** m=number of nuclides, n=number of regions, nq=number of plant/
c****  invertebrate organs/parts used for dosimetry, ko=number of
c**** zoons including humans, ke= maximum age for animals
c
      parameter (i=8, ks=10, kt=40, kw=10, ka=200, kr=86, m=10, n=8,
     1nq=36, ko=10, ke=25)
      real lvtbl, kdtbl1, kdtbl2, mstbl
      common /aa/ a1(i,m,n), a2(i,m,n), a3(i,m,n), aaa(i,m,n), aatb(i,kt
     1 ,n), acres(i,n), afr(i,m,n), afs(i,m,n), agtb(i,kt,n), agz(ko),as
     2 (i,m,n),atno(m),attt(i,n),autzd(ko,7,n),awata(i,36,n),a1g(i,m,n),
     3 b1(i,n),b5(i,n),b7(i,n),batr,c6(i,n),c7,c8(i,n),aag(i,m,n),ccvv
     4 (i,n), cfac(i,n), ct(m,n), cuta(i,8,n), cxfac(i,n), d1(i,n), daic
     5 (16,m,n), dira(i,36,n), dirt, dtbr(i,kt,n), ea(i,n), eatin(ko),f1
     6 (i,ks,n), fodin(ko), ga(kt,n),ica(i,n),inz,nu,ctg(m,n),
     7 ipx, irg(i,n), phi(i,ks,n), prec(n), putzd(ko,i,n), r1(i,n),rank
     8 (i,n), rd1(i,n), rgc(i,n), ringt(i,n), rirrc(i,n), rndlx(n),iu,
     9 rnofdm(i,m,n), rnofym(i,m,n), rscde(n), rss(i,n), salcrp(i,m,n),
     1 satb(i,kt,m,n), sd(i,n), sd1(i,n), sdmx(i,n), sfac(i,n),
     1 stmtt, t5p(i,n), tav(n), tbrm(i,n), tlai(i,n), tpgnw(i,n),
     1 wa(i,ks,n), wd1(i,n), wfr(i,ks,n), wnd(i,n), wurat(i,n), wyav(i,n
     1 ), wz3(i,m,n), wz4(i,m,n), wz42(i,m,n), wz43(i,m,n), wz7(i,m,n),
     1 wz8(i,m,n), x8(i,m,n), yrfed(ko,32), z(kt,n), z1(i,ks,m,n),z2(i,m
     1 ,n), z3(i,m,n), z4(i,m,n), z4r(i,ks,m,n), z5(i,m,n), z6(i,m,n),
     1 z7(i,m,n), z8(i,m,n), z9(i,m,n), z9r(i,m,n), zf(i,ks,n), zff(i,ks
     1 ,n), zsd(i,m,n), zu(i,ks,m,n), zz3(i,m,n),agr(ko,n),zgr(n)
c
      common /cd/ a1av(i,m,n), aiche(n), amw(ke,ko),asav(i,m,n),bar(kt,n
     1 ), barsq(kt,n), bbp(i,n), bsmax(i,n), c9(i,n), clay(i,ks,n), cov
     2 (kt,n), covsq(kt,n), cutmax(i,n), cz(i,ks,n), d3(i,n), e5(i,n),
     3 eaval(i,ks,n), etrans(i,n), hgt(kt,n), hgtsq(kt,n), p1(i,n), pav
     4 (n), pltind(ko,32,n), r1max(i,n), rd3(i,n), rflct, aircg(m,n),
     5 rnofdg(i,n), rnofds(i,n), rnofyg(i,n), rnofys(i,n), rpntp(i,n),
     6 rsusp(i,m,n), slice(i,ks,n), slopx(30), tntr(i,n),ccmax(i,n),
     7 vtmbr(i,kt,n), vvtm(i,kt,n), watakd(i,ks,n), wataku(i,ks,n),
     8 watupa(i,ks,n), watupb(i,ks,n), watx(n), wd3(i,n), wndmax(i,n),
     9 wz4c(i,m,n), wz5c(i,m,n), wz7max(i,m,n), wzz4c(i,m,n), xj1(i,n),
     1 xx(kt,n), ym(i,ks,n), z4gc(i,m,n), z4rc(i,ks,m,n), z5c(i,m,n),
     1 z5rc(i,ks,m,n), z6max(i,m,n), z7max(i,m,n),dchan(30), gsbdtt(30),
     1 vchan(30),vaqt(30),sir(m,n),nan,nhm,wt(kr,ko),wms(ko,2),wmk(ko)
c
      common /ef/ a1tt(m,n),a22(m,n), a23(m,n),a2x(i,m,n),vfat(kr,ko,m),
     1 a3av(i,m,n),aa2(ko),actav(i,m,n),addwat,afsav(i,m,n),a1tg(m,n),
     2 agave, agea(ko),ai1,aiii,airca(m,n),airx(n),amx(ke,ko),bmn(i,n),
     3 bsd,c2,c31(ko,m,n),c4,cnit(i,n),covav,cvmn,ac,tpopi(kr,2,ko),
     4 cvsd, cvtr, d(i,n), dburd(m), drnkk, e2(n), eatrec(ko,32),effp(n)
     5 , evap(i,n), f7(n),fdin(ke,ko),fsupf(ko),g(i,n),gmeat,
     6 gmilk, gtdr(ko,n), h2, h3, hm8, hrvst(ko,i,n), htav(i,n), htsd,
     7 humus(ko,m,n), ipl, onfod(ko,32), p(n),p5(ko),pburd(m),pltid(i,n)
```

```
     8 , pltrnk(ko,8,32,n),pltusd(ko,8,32,n),putfrc(ko,i,n),q1(kr,ko,m),
     9 q10(i,n),q2(kr,ko,m),q3(kr,ko,m),qfrac(ko,n),qsum(ko,n),rcode(i,n
     1 ), rd2(i,n), rdcf(n), rh1(i,ks,n), rr(ko),rzoon(ko),
     1 sa, sact(ko,m,n), sair(ko,m,n), sd3(i,n), sdpth(i,n),sffp(n),snit
     1 (i,n), snm(n), solrad, spaa(ko,i,m,n), spac(ko,m,n),spar(ko,m,n),
     1 spas(ko,i,m,n), t(n), thr(24,n), tnit(ko,n),totsf(ko,i,n),trees(i
     1 ,n), ttz(n), ugrn, ulv,up(ko,n),v(kr,ko,m),v4,vbn1(kr,ko,m),vbn2(
     1kr,ko,m),vkd1(kr,ko,m),vkd2(kr,ko,m),vlv(kr,ko,m),vmk(kr,ko,m),vms
     1 (kr,ko,m),w1(i,ks,n),water(i,n),watfod,watupt,watvol,wtbl(kr,ko),
     1 rsrt(kr,ko), yzf(i,n),z4c(i,m,n),z77(m,n),z78(m,n),z79(m,n),
     1 z7a(m,n), z7ar(m,n), zgrn(m), zlv(m), zmeat(n), zoo(ko), zsv(m)
c
      common /eg/ a3ac(i,m,n), a3c(i,m,n), afsac(i,m,n), afsc(i,m,n),
     1 antav(n), b2(i,m,n), b3(i,m,n), b3ac(i,m,n), b3av(i,m,n), b3c(i,m
     2 ,n), bb2(i,m,n), bb3(i,m,n), bb3ac(i,m,n), bb3av(i,m,n), bb3c(i,m
     3 ,n), c5, d1gb(i,n
     4 ), d1gbb(i,n),d1gt(i,n),
     5 gm2b(i,n), gm2r(i,n), gm2rb(i,n),h(i,ks,n), pltf(ko,8,3
     6 2,n), rb3(i,m,n), rb3c(i,m,n), rbb3(i,m,n), rbb3c(i,m,n), rh2(i
     7 ,ks,n), tsrf, usv
     8 , w(30,ko,m), wburd(m), wz5(i,m,n), z42(i,m,n), z43(i
     9 ,m,n), z77a(m,n),antro(ko),antros(ko),grna(ko),grnas(ko),dige(ko)
     1 ,rfrac(ko),ototbs(ko),otot(ko),ottsm(ko),sumput(ko),
     1 sototz(n),c31tt(ko,m),ainsm,iian,sptfrr(n),sptfra(n),sptfrf(n)
c
      common /gh/ absrad(kw,n), alsp(ko), anitr(kw,n), apl(n),
     1 calatm, calra, calri, calsub, cells(5,kw,n), dayfrc, declin,
     2 dirmax(i,n), disint(m), evtemp, extd, gm2t(i,n), hflif(m), kdi,
     3 otots(ko),otota(ko),ototb(ko),phitt(3),phosp(kw,n),photpr,phyto(5
     4 ,kw,n), radang, refrac, rfrz(ko,n), rnfrc(ko,32,n), silcn(kw,n),
     5 tmn(n), tmx(n), va(kt,m,n), watmp(kw,n), z4g(i,m,n), z5r(i,ks,m,n
     6 ), zmilk(m), zobio(2,kw,n), zoopc(2,kw,n), zoopm(2,kw,n)
c
      common /gg/ aice(30), akond(30), aksoil(n), alsoil(n), aplx(n),
     1 atx(30), bbch(30), bch(30), bdrk(n), c1hr(30), c2hr(30), c5hr(30)
     2 , cfspd(30), chnaq(30), csoil(n), dsf(7,30), dsm(7,30),dsma(7,30
     3 ), dsms(7,30), dsmt(30), dsz(8,30), duhr(30), dzas(7,30), gsbd(7,
     4 30), hbt(30), htp(30), htx(30), prch(30), psoil(n), qpkd(30),
     5 rarea(30), rchan(30), rfdlx(n), rnodx(i,n), schan(30), scvr(i,n),
     6 sdmnct(30), sdmnsa(30), sedmns(30), tbch(30), tch(30), tchan(30),
     7 tcvr(n), tndlx(n), tndtn(n), vaqf(30), vlat(30), vloss(30), vrfl(
     8 30), vtrnd(30), wchan(30), wsrp(i,n), wtx(30), xchan(30)
c
      common /oo/ c56(m), c99(m), car(m), sedgat, viiin(m)
c
      common /pq/ aqugat, bltbn1(ko,m),bltbn2(ko,m),bltkd1(ko,m),bltkd2(
     1ko,m),bltlu(ko,m),bltlv(ko,m),bltmk(ko,m),bltms(ko,m),bltrs(ko,m),
     2 bltur(ko,m), bntbl1(ko,m),bntbl2(ko,m),chnnu,gtbl(ko,m)
     3 , kdtbl1(ko,m), kdtbl2(ko,m), lvtbl(ko,m), mstbl(ko,m),plcm1(ko),
     4 plcm2(ko), plcm3(ko),pltbl(ko,m),pltg1(ko,m),pltg2(ko,m),
     5 rbn(kr,ko,m),rkd(kr,ko,m),rlu(kr,ko,m),fttbl(ko,m),pltblg(ko,m),
     6 rlv(kr,ko,m), rms(kr,ko,m),rndo(kr,ko,m),rstbl(ko,m),totsa(m
     7 ,n), vbn(kr,ko,m), vkd(kr,ko,m),rpt(ka,i,nq,m,n),
     8 evs(ko,m),eve(ko,m),evb(ko,m),acti(ko)
```

```
c
 common/xb/ zmax(2,n),ztmn(2,n),ztx(2,n),celki(2,n),aiopt(5,n),
1aksph(5,n),aksnt(5,n),akssi(5,n),umph(5,n),umnt(5,n),umsi(5,n).
2akqph(5,n),akqnt(5,n),akqsi(5,n),zkqph(2,n),zkqnt(2,n),qmph(5,n),
3qmsi(5,n),qmnt(5,n),zmnt(2,n),zmph(2,n),resp(5,n),sink(5,n),
4cmax(5,n),topm(5,n),tulv(5,n),tllv(5,n),divd(5,n),exip(5,n),
5y1(110,2),psep(22,n),bcp(22,n),strans(22,n),topt(22,n),tofl(22,n),
6tofh(22,n),fleaf(22,n),ulef(22,n),respf(22,n),rspf2(22,n),
7a(85,ko,2),brty(8,ko),brrt(8,ko,2),tqx(85,2,ko),nday(12),
8tbbnz(100),tbkdz(100),fbwbz(100),
9fmk(100),wutp(22,n),hf(22,n),sctp(22,n),astp(22,n),rxpp(22,n),
1rfp(22,n),cbbnz(100),cbkdz(100),cblvz(100),cbwbz(100),tblvz(100),
2tbwbz(100),difmol(110),henrc(110),akoc(110),zkd(110)
```

15.0. APPENDIX F

PLANT NUMBER POSITIONS FOR ECOLOGICAL TRANSPORT MODEL AND GROWTH SIMULATION DATA

15.1. Plant Types

WARM SEASON PLANTS

1. Corn and sorghum
2. Annual vegetables and grains
3. Pasture grasses
4. Fruit trees
5. Pinyon-Juniper
6. Deciduous shrubs
7. Annual grasses and herbs
8. Perennial grasses and herbs
9. Evergreen shrubs
10. Deciduous trees

COOL SEASON PLANTS

11. Alfalfa
12. Annual vegetables and grains
13. Pasture grasses
14. Douglas Fir
15. Ponderosa Pine
16. Deciduous shrubs
17. Spruce-fir
18. Perennial grasses and herbs
19. Annual grasses and herbs
20. Evergreen shrubs
21. Deciduous trees
22. Cheatgrass

15.2. Growth Simulation Data

ID	WUTP	HF	PSEP	ASTP		RFP	BCP	STRANS	TOPT	TOFL	TOFH	FLEAF	ULEF	RESPF
1	200	1	0.5	0	0.161	0.0013	0.6	0.75	30	20	9	0	0	0.05
2	200	1	1	0	0.161	0.0013	0.6	0.75	25	15	9	0	0	0.05
3	200	1	1	0	0.161	0.0013	0.6	0.3	25	15	9	0	0	0.05
4	150	0	1	0	0.161	0.0013	0.4	0.85	21	11	13	0.25	0.4	5
5	300	0	1	1	0.161	0.0013	0.4	0.85	25	25	9	0.125	0.5	5
6	150	0	1	0	0.161	0.0013	0.4	0.75	21	11	13	0.25	0.4	5
7	200	1	0.5	0	0.161	0.0013	0.6	0.5	30	20	9	0	0	0.1
8	200	1	0.5	0	0.161	0.0013	0.6	0.3	30	20	9	0	0	0.05
9	300	0	1	1	0.161	0.0013	0.4	0.75	21	21	13	0.25	0.5	5
10	150	0	1	0	0.161	0.0013	0.4	0.85	21	11	13	0.25	0.4	5
11	200	1	1	0	0.161	0.0013	0.6	0.5	21	11	13	0	0	0.05
12	200	1	1	0	0.12	0.0054	0.6	0.5	18	10	16	0	0	0.05
13	200	1	1	0	0.12	0.0054	0.6	0.3	18	10	16	0	0	0.05
14	300	0	1	1	0.12	0.0054	0.4	0.85	14	14	20	0.125	0.5	5
15	300	0	1	1	0.12	0.0054	0.4	0.85	18	18	16	0.125	0.5	5
16	150	0	1	0	0.12	0.0054	0.4	0.75	15	10	19	0.25	0.4	5
17	300	0	1	1	0.12	0.0054	0.4	0.85	12	12	22	0.125	0.5	5
18	200	1	0.5	0	0.12	0.0054	0.6	0.3	18	10	16	0	0	0.05
19	200	1	0.5	0	0.12	0.0054	0.6	0.5	18	10	16	0	0	0.1
20	300	0	1	1	0.12	0.0054	0.4	0.75	15	15	19	0.25	0.5	9
21	150	0	1	0	0.12	0.0054	0.4	0.85	15	10	19	0.25	0.4	5
22	100	1	1	0	0.12	0.0054	0.6	0.8	18	11	16	0	0	0.1

Variable Identification: **ID**=plant type, **WUTP**=photsynthesis water utilization efficiency, **HF**=mean life-span of plant, **PSEP**=distinguishes between C_4(0.5) and C_3(1.0) plant types, **SCTP**=woody(0) or evergreen(1) type of plant, **ASTP**=deciduous(0) or evergreen(1) type of plant, **RXPD**=root respiration coefficient, **RFP**=root respiration coefficient, **BCP**=net photosynthate fraction, **STRANS**=photosynthate utilization fraction for above ground biomass, **TOPT**=optimum temperature for photosynthesis, **TOFL**=optimum minus lowest temperature for photosynthesis, **TOFH**=optimum minus highest temperature for photosynthesis, **FLEAF**=initial leaf fraction for trees and shrubs, **ULEF**=running average leaf fraction, **RESPF**=live-to-dead leaf conversion fraction.

15.0. APPENDIX F

PLANT NUMBER POSITIONS FOR ECOLOGICAL TRANSPORT MODEL AND GROWTH SIMULATION DATA

15.1. Plant Types

WARM SEASON PLANTS		COOL SEASON PLANTS	
1.	Corn and sorghum	11.	Alfalfa
2.	Annual vegetables and grains	12.	Annual vegetables and grains
3.	Pasture grasses	13.	Pasture grasses
4.	Fruit trees	14.	Douglas Fir
5.	Pinon-Juniper	15.	Ponderosa Pine
6.	Deciduous shrubs	16.	Deciduous shrubs
7.	Annual grasses and herbs	17.	Spruce-fir
8.	Perennial grasses and herbs	18.	Perennial grasses and herbs
9.	Evergreen shrubs	19.	Annual grasses and herbs
10.	Deciduous trees	20.	Evergreen shrubs
		21.	Deciduous trees
		22.	Cheatgrass

15.2. Growth Simulation Data

ID	WUTP	HF	PSEP	ASTP	RXPD	RFP	BCP	STRANS	TOPT	TOFL	TOFH	FLEAF	ULEF	RESPF
1	200	1	0.5	0	0.161	0.0013	0.6	0.75	30	20	9	0	0	0.05
2	200	1	1	0	0.161	0.0013	0.6	0.75	25	15	9	0	0	0.05
3	200	1	1	0	0.161	0.0013	0.6	0.3	25	15	9	0	0	0.05
4	150	0	1	0	0.161	0.0013	0.4	0.85	21	11	13	0.25	0.4	5
5	300	0	1	1	0.161	0.0013	0.4	0.85	25	25	9	0.125	0.5	5
6	150	0	1	0	0.161	0.0013	0.4	0.75	21	11	13	0.25	0.4	5
7	200	1	0.5	0	0.161	0.0013	0.6	0.5	30	20	9	0	0	0.1
8	200	1	0.5	0	0.161	0.0013	0.6	0.3	30	20	9	0	0	0.05
9	300	0	1	1	0.161	0.0013	0.4	0.75	21	21	13	0.25	0.5	5
10	150	0	1	0	0.161	0.0013	0.4	0.85	21	11	13	0.25	0.4	5
11	200	1	1	0	0.161	0.0013	0.6	0.5	21	11	13	0	0	0.05
12	200	1	1	0	0.12	0.0054	0.6	0.5	18	10	16	0	0	0.05
13	200	1	1	0	0.12	0.0054	0.6	0.3	18	10	16	0	0	0.05
14	300	0	1	1	0.12	0.0054	0.4	0.85	14	14	20	0.125	0.5	5
15	300	0	1	1	0.12	0.0054	0.4	0.85	18	18	16	0.125	0.5	5
16	150	0	1	0	0.12	0.0054	0.4	0.75	15	10	19	0.25	0.4	5
17	300	0	1	1	0.12	0.0054	0.4	0.85	12	12	22	0.125	0.5	5
18	200	1	0.5	0	0.12	0.0054	0.6	0.3	18	10	16	0	0	0.05
19	200	1	0.5	0	0.12	0.0054	0.6	0.5	18	10	16	0	0	0.1
20	300	0	1	1	0.12	0.0054	0.4	0.75	15	15	19	0.25	0.5	9
21	150	0	1	0	0.12	0.0054	0.4	0.85	15	10	19	0.25	0.4	5
22	100	1	1	0	0.12	0.0054	0.6	0.8	18	11	16	0	0	0.1

Variable Identification: **ID**=plant type, **WUTP**=photsynthesis water utilization efficiency, **HF**=mean life-span of plant, **PSEP**=distinguishes between C_4(0.5) and C_3(1.0) plant types, **SCTP**=woody(0) or evergreen(1) type of plant, **ASTP**=deciduous(0) or evergreen(1) type of plant, **RXPD**=root respiration coefficient, **RFP**=root respiration coefficient, **BCP**=net photosynthate fraction, **STRANS**=photosynthate utilization fraction for above ground biomass, **TOPT**=optimum temperature for photosynthesis, **TOFL**=optimum minus lowest temperature for photosynthesis, **TOFH**=optimum minus highest temperature for photosynthesis, **FLEAF**=initial leaf fraction for trees and shrubs, **ULEF**=running average leaf fraction, **RESPF**=live-to-dead leaf conversion fraction.