INTEGRATED CODES FOR ESTIMATING ENVIRONMENTAL
ACCUMULATION AND INDIVIDUAL DOSE FROM PAST
HANFORD ATMOSPHERIC RELEASES

Hanford Environmental Dose Reconstruction Project

T. A. Ikenberry
R. A. Burnett
B. A. Napier
N. A. Reitz
D. B. Shipler

February 1992

Prepared for
review, evaluation, and approval
by the Technical Steering Panel

Pacific Northwest Laboratory
Richland, Washington 99352

DISCLAIMER

This report was prepared under the direction of the HANFORD ENVIRONMENTAL DOSE RECONSTRUCTION PROJECT Technical Steering Panel by Battelle Memorial Institute's Pacific Northwest Laboratories operating the Pacific Northwest Laboratory for the U.S. Department of Energy (DOE). While funding for the work was provided by DOE, the work is not under DOE direction or control. The views and opinions of the authors expressed in this document do not necessarily reflect those of the United States Government or any agency thereof. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation or favoring by the U.S. Government or any agency thereof, nor by Battelle Memorial Institute.

INTEGRATED CODES FOR ESTIMATING ENVIRONMENTAL
ACCUMULATION AND INDIVIDUAL DOSE FROM PAST
HANFORD ATMOSPHERIC RELEASES

Hanford Environmental Dose Reconstruction Project

February 1992

This document has been reviewed and
approved by the Technical Steering Panel.

J. E. Till, Chair
Technical Steering Panel

Date

PREFACE

This document is one of the deliverables required of the Pacific Northwest Laboratory (PNL) by the Technical Steering Panel (TSP) as part of the approved plans of the Hanford Environmental Dose Reconstruction (HEDR) Project. It provides the status of environmental accumulation and dose modeling. The document is being provided to the TSP and the Centers for Disease Control (CDC) for review and comment prior to approval and finalization.

In 1987, the U.S. Department of Energy (DOE) directed PNL (which is operated by Battelle Memorial Institute) to conduct the HEDR Project after receiving a recommendation by the Hanford Health Effects Review (HHER) Panel in 1986. The HHER Panel was formed to consider the potential health implications of historic Hanford Site releases of radioactive materials.

Members of the TSP were selected by the Vice Presidents for Research at the major universities of Washington and Oregon to direct the work of the project. The 18-member TSP consists of experts in the various technical fields relevant to HEDR Project work and representatives from the states of Washington, Oregon, and Idaho; Native American tribes; and the public. A December 1990 Memorandum of Understanding between the Secretaries of the DOE and U.S. Department of Health and Human Services (DHHS) transferred responsibility for managing the DOE's dose reconstruction and exposure assessment studies to the DHHS. This transfer resulted in the current process to establish a contract between Battelle and the CDC, which is an agency of the DHHS, to continue performance of project work. The TSP will continue its role as the technical director for Battelle's work when the contract with the CDC is implemented.

The HEDR Project objective is to estimate radiation doses to individuals and population groups from exposure to historical, radioactive emissions from the Hanford Site. Phase I of the project, which ended in July 1990, was designed to determine whether enough information of sufficient quality was available to develop and demonstrate a dose estimating method. The product of this phase was a set of more than 20 documents that describe the preliminary

information found or reconstructed, preliminary dose estimating models and computer codes, and preliminary estimates of doses and their uncertainties for representative individuals who may have lived near the Hanford Site during early years of operations. Phase I estimates of dose from air pathways were limited to about 20 representative individuals living within each of 99 census subdivisions in a 10-county area near Hanford. Dose estimates from Columbia River pathways were estimated for six representative individuals at each of five locations along the Hanford Reach of the Columbia River. Doses were estimated from potential exposure to iodine-131 released to the atmosphere from 1944 through 1947 and to about eight radionuclides released into the Columbia River from 1964 through 1966.

Technical work since July 1990 has consisted of evaluation of Phase I data and models; development of detailed plans for continuing work; restructuring of models to enhance their capabilities; development of detailed estimates of releases of radioactive materials; and identification, acquisition, and evaluation of additional information on plant operations, environmental monitoring, demographics, and agricultural, lifestyle, and food consumption patterns. This information is being developed for 19 counties, for major exposure pathways, and for the full history of the Hanford Site -- 1944 through 1991. In addition to the work being performed by Battelle for dose estimations for the general population, research is being conducted by eight Native American tribes near the Hanford Site to support dose estimates for their tribal members.

Work completed between July 1990 and December 1991 included

- a detailed description of the groundwater pathway and associated dose estimates
- restructuring and documentation of the stochastic atmospheric dispersion and deposition computer code
- restructuring and documentation of the stochastic environmental air pathways and dose estimating computer code
- a detailed description of the Columbia River pathway and associated environmental monitoring data and dose estimates

- reconstructed detailed release estimates of iodine-131 for the first 3 years of Site operations
- identification of all existing records of reactor operations for their lifetimes
- identification of all existing records of chemical separations plant operations for their lifetimes
- development of the stochastic computer code that models time- and activity-dependent releases of radioactive materials
- identification of environmental monitoring data for 1944 through 1957
- development of the model for the production and distribution of milk and milk products for a 19-county area for 1944 through 1991.

Project work being performed through FY 1992 will provide the basis for decisions to be made by the TSP on whether and how to expand or modify the scope of currently approved plans. Project work through FY 1992 will also generate various types of dose estimates.

The technical work required to develop databases and computer codes to estimate doses to individuals and special population groups for key radionuclides and major exposure pathways, in the 19-county area around the Hanford Site from 1944 through 1991, will be completed in early 1994.

Planned work activities for FY 1992 and 1993 are summarized in Figure P.1 and include

- A. Completion of time- and quantity-related release estimates for key radionuclides for the atmospheric and Columbia River pathways
- B. Completion of evaluation of Hanford-originated environmental monitoring data
- C. Completion of models and associated work:
 1. Milk production and distribution model for 19 counties surrounding the Hanford Site
 2. Production and distribution model of exposed fruits, vegetables, and livestock feed for the 19-county area
 3. Simple routing and decay model for concentrations of key radionuclides in Columbia River water

FIGURE P.1. HEDR Schedule Overview, FY 1992 - 1995

4. Completion of models for air dispersion, environmental accumulation, and dose.
 5. Development and implementation of a plan for sensitivity/uncertainty analysis and verification/validation activities of stochastic models for release, atmospheric transport, pathways, and dose
- D. Completion of dose estimates for
1. Hanford Thyroid Disease Study (which is conducted by the Fred Hutchinson Cancer Research Center)
 2. 19-county representative individuals and identified real individuals
 3. Native American tribes.

This report fulfills the milestone shown in Figure P.1, across from Line C.4, called Environmental/dose model report. The status of the activities shown in Figure P.1 will be documented in reports due to the TSP and CDC in FY 1992 and 1993. In FY 1994 and 1995, activities will be conducted to develop databases to estimate doses for other special population groups, minor pathways, and locations outside of the 19-county area.

SUMMARY

Preliminary radiation doses were estimated and reported during Phase I of the Hanford Environmental Dose Reconstruction (HEDR) Project. As the project has progressed, additional information regarding the magnitude and timing of past radioactive releases has been developed, and the general scope of the required calculations has been enhanced.

The overall HEDR computational model for computing doses attributable to atmospheric releases from Hanford Site operations is called HEDRIC (Hanford Environmental Dose Reconstruction Integrated Codes). It consists of four interrelated models: source term, atmospheric transport, environmental accumulation, and individual dose. The source term and atmospheric transport models are documented elsewhere.^{(a)(b)}

This report describes the initial implementation of the design specifications for the environmental accumulation model and computer code, called DESCARTES (Dynamic Estimates of Concentrations and Accumulated Radionuclides in Terrestrial Environments), and the individual dose model and computer code, called CIDER (Calculation of Individual Doses from Environmental Radionuclides). The computations required of these models and the design specifications for their codes were documented in Napier et al. (1992). Revisions to the original specifications and the basis for modeling decisions are explained. This report is not the final code documentation but gives the status of the model and code development to date. Final code documentation is scheduled to be completed in FY 1994 following additional code upgrades and refinements.

The user's guide included in this report describes the operation of the environmental accumulation and individual dose codes and associated pre- and post-processor programs. A programmer's guide describes the logical structure of the programs and their input and output files.

-
- (a) Heeb, C. M. 1992. Iodine-131 Release Model Report, milestone report 0305A (Draft)
 - (b) Ramsdell, J. V. 1992. Regional Atmospheric Transport Code for Hanford Emission Tracking (RATCHET) (Draft)

CONTENTS

PREFACE	iii
SUMMARY	vii
1.0 INTRODUCTION	1.1
1.1 BACKGROUND	1.1
1.2 RELATIONSHIPS TO OTHER MODELS, CODES, AND DATABASES	1.5
1.3 PROPOSED LIFE CYCLES FOR HEDRIC CODES	1.10
2.0 MODEL DESCRIPTION	2.1
2.1 MODEL LIMITATIONS	2.2
2.2 ENVIRONMENTAL ACCUMULATION MODEL	2.4
2.2.1 Model Development	2.4
2.2.2 Changes from the Design Specifications	2.6
2.2.3 Submodels	2.8
2.3 INDIVIDUAL DOSE MODEL	2.24
2.3.1 Changes from the Design Specifications	2.25
2.3.2 Age-Dependent Dose Factors	2.26
2.3.3 Location-Specific Doses	2.27
2.3.4 Ingestion of Locally and Commercially Available Foods	2.30
2.4 FUTURE MODELING ENHANCEMENTS	2.31
3.0 USER'S GUIDE	3.1
3.1 OVERVIEW OF CODES	3.1
3.2 CURRENT CODE STATUS AND LIMITATIONS	3.3
3.3 COMPUTER ACCESS	3.4
3.4 INPUT PARAMETER SELECTION PROGRAM: PARSEL	3.4
3.4.1 Entering Program Control Options	3.5

3.4.2	Selecting Parameter Values	3.6
3.4.3	Running PARSEL as a Background Process	3.6
3.5	ENVIRONMENTAL ACCUMULATION CODE: DESCARTES	3.6
3.5.1	Entering Program Control Options	3.7
3.5.2	Selecting Output Options	3.7
3.5.3	Running DESCARTES as a Background Process	3.8
3.6	INDIVIDUAL DOSE CODE: CIDER	3.9
3.6.1	Entering Program Control Options	3.10
3.6.2	Entering Characteristics of Individuals	3.11
3.6.3	Running CIDER as a Background Process	3.13
3.7	POST-PROCESSOR AND REPORT GENERATOR PROGRAM: REPGEN	3.14
3.7.1	Selecting Model Results for Reporting	3.14
3.7.2	Selecting Output Options	3.17
3.7.3	Running REPGEN as a Background Process	3.17
3.8	PLANNED EXPANSION AND REFINEMENT	3.18
4.0	PROGRAMMER'S GUIDE	4.1
4.1	OVERVIEW	4.1
4.1.1	Program Organization and Relationships	4.1
4.1.2	Language and Coding Standards	4.1
4.1.3	Data Input and Output	4.3
4.1.4	Computer System Requirements	4.4
4.1.5	System-Dependent Code Elements	4.5
4.2	INPUT PARAMETER SELECTION PROGRAM	4.6
4.2.1	Function	4.6
4.2.2	Input and Output Files	4.7
4.2.3	Code Logic and Structure	4.7

4.2.4	Algorithms and Special Routines	4.7
4.3	ENVIRONMENTAL ACCUMULATION CODE	4.9
4.3.1	Function	4.9
4.3.2	Input and Output Files	4.10
4.3.3	Code Logic and Structure	4.13
4.3.4	Algorithms and Special Routines	4.13
4.4	INDIVIDUAL DOSE CODE	4.16
4.4.1	Function	4.16
4.4.2	Input and Output Files	4.17
4.4.3	Code Logic and Structure	4.17
4.4.4	Algorithms and Special Routines	4.17
4.5	POST-PROCESSOR AND REPORT GENERATOR PROGRAM	4.19
4.5.1	Function	4.19
4.5.2	Input and Output Files	4.20
4.5.3	Code Logic and Structure	4.20
4.5.4	Algorithms and Special Routines	4.20
5.0	QUALITY ASSURANCE	5.1
5.1	DATA QUALITY OBJECTIVES	5.1
5.2	CODE VERIFICATION PROCEDURES	5.1
5.3	OTHER QUALITY ASSURANCE PROCEDURES	5.2
6.0	REFERENCES	6.1
APPENDIX A - LIST OF VARIABLE AND PARAMETER NAMES		A.1
APPENDIX B - INPUT DATA FILE LISTINGS		B.1
APPENDIX C - CODE LISTINGS		
C1	- PARSEL CODE LISTING	C1.1
C2	- DESCARTES CODE LISTING	C2.1
C3	- CIDER CODE LISTING	C3.1
C4	- REPGEN CODE LISTING	C4.1
APPENDIX D - EXAMPLE REPGEN REPORT		D.1

FIGURES

1.1	Expanded Model Domain for HEDRIC Codes	1.3
1.2	Detailed Grid for Calculation of Air Concentration and Deposition, Environmental Concentrations, and Individual Doses	1.4
1.3	Organization of the HEDRIC Atmospheric Pathways Codes	1.6
1.4	HEDRIC Codes and Output Used in the Turnover Package for Individual Dose Calculations	1.9
2.1	Environmental Accumulation Model	2.5
2.2	Dynamic Compartments in the Environmental Accumulation Model	2.12
2.3	Example Schematic of Dairy Cow Feeding Regime Input Data	2.20
2.4	Individual Dose Model	2.25
4.1	Flow of Data Among HEDRIC Programs	4.2
4.2	PARSEL Pseudo-Code Listing	4.8
4.3	PARSEL Subprogram Calling Structure	4.9
4.4	DESCARTES Pseudo-Code Listing	4.14
4.5	DESCARTES Subprogram Calling Structure	4.16
4.6	CIDER Pseudo-Code Listing	4.18
4.7	CIDER Subprogram Calling Structure	4.19
4.8	REPGEN Pseudo-Code Listing	4.21
4.9	REPGEN Subprogram Calling Structure	4.21

1.0 INTRODUCTION

The objective of the Hanford Environmental Dose Reconstruction (HEDR) Project is to estimate the radiation dose that individuals could have received as a result of emissions of radioactive material from nuclear operations at the Hanford Site since the site began operating in 1944. This report describes the models and associated computer codes developed for estimating environmental accumulation and dose to individuals resulting from Hanford releases of radionuclides to the atmosphere. Estimates of doses resulting from Hanford releases of radionuclides to surface water and groundwater are being addressed in separate reports.^{(a)(b)}

1.1 BACKGROUND

In response to modeling criteria stated in the Technical Steering Panel's Research Directive 89-5, "Modeling Approach" (originally issued January 29, 1989, and modified February 14, 1991), the HEDR Project is being undertaken in a series of steps. The first step, Phase I, was the development of basic techniques and proof of principle. The objective of Phase I was to demonstrate through calculation that adequate models and support data existed or could be developed to allow estimation of realistic doses to individuals from releases of radionuclides to the environment that occurred as early as 1944. Within the constraints of that first phase, a computational methodology was developed (Napier 1991a) and the feasibility of the project was demonstrated (PNL 1991a, 1991b, 1991c). Preliminary dose estimates, at a temporal resolution of 1 month, were developed for a 10-county region surrounding the Hanford Site during the years 1944 through 1947.

A number of investigations were made using the Phase I dose estimates. In these investigations, the adequacy of the Phase I models was evaluated in terms of accuracy, precision, and reproducibility of the results. Potential

-
- (a) Walters, W. H., et al. 1992. Literature and Data Review for the Surface-Water Pathway: Columbia River and Adjacent Coastal Areas (Draft)
 - (b) Freshley, M.D., and P. D. Thorne. 1992. Ground-Water Contribution to Dose from Past Hanford Operations. PNL-7978 HEDR, Pacific Northwest Laboratory, Richland, Washington.

improvements were identified for the atmospheric pathways models in the areas of spatial and temporal resolution, model internal structure, and propagation of uncertainties. The major investigations are documented in two reports by Simpson (1991, 1992). The first of these reports discusses how initial assumptions made in coupling the various portions of the Phase I computer codes affected the dose estimates. The companion report deals with the effect of the internal computational structure of the codes on the estimates. Both reports indicated that assumptions made and numerical processes used in Phase I could be improved.

Based on the conclusions of these and other investigations, the next step in the model development process was begun: the design and development of restructured atmospheric pathways models and associated computer codes. This report describes the initial version of the restructured environmental accumulation and individual dose models and their corresponding codes.

The scales for spatial and temporal resolution for the restructured atmospheric pathways models (Shiplier 1991a) have been altered from those of the Phase I model. The general domain of the new set of related atmospheric pathway codes, HEDRIC (Hanford Environmental Dose Reconstruction Integrated Codes), is illustrated in Figure 1.1. The area includes 20 counties in eastern Washington State, 7 counties in north-central Oregon, 6 counties in northern Idaho, and the portions of the Nez Perce Indian Reservation that project beyond these counties. The area is bounded on the east by the Rocky Mountains, on the south by the Blue Mountains, and on the west by the Cascade Mountains. The area generally encompasses all of the Columbia Basin. Overlaid on this area is a detailed grid set on six-mile centers (Figure 1.2). Calculations can be performed over the entire grid area if input data are available. However, current plans call for calculations to be performed for a 19-county area around the Hanford Site (Shiplier 1991a). This area includes the Phase I study area of Benton, Franklin, Adams, Grant, Kittitas, Yakima, Klickitat, and Walla Walla counties in Washington and Morrow and Umatilla counties in Oregon. Added Washington counties are Douglas, Lincoln, Spokane, Whitman, Columbia, Garfield, and Asotin along with the Oregon counties of

FIGURE 1.1. Expanded Model Domain for HEDRIC Codes

Sherman and Gilliam. Individual calculations will be performed to the degree possible at this level of spatial resolution. Some information required by the models will be available at the individual grid level, and other information will be available only for larger geographic areas, such as census subdivisions or counties. Calculations will be performed at a temporal resolution of 1 day.

Model and code design specifications for the restructured environmental accumulation and individual dose programs were prepared by Napier et al. (1992). These specifications define the computations needed to incorporate all of the relevant data collected by the HEDR Project staff and to produce

FIGURE 1.2. Detailed Grid for Calculation of Air Concentration and Deposition, Environmental Concentrations and Individual Doses

radiation dose estimates. Napier et al. (1992) also address the requirements that the computer codes must meet and outline the methods selected for meeting those requirements. They also define the models and outline the structure and logic of the envisioned computer code.

This report describes the initial implementation of the design specifications in Napier et al. (1992). To aid the reader, the basic equations defined in the design specifications are repeated in this report, and deviations from

or revisions to the original specifications are described. Also included are a user's guide describing the operation of the programs and a programmer's guide describing the logical structure of the programs and their input and output files.

1.2 RELATIONSHIPS TO OTHER MODELS, CODES, AND DATA BASES

The goal of the project is to calculate radiation doses, with associated variability and uncertainty, for specific individuals and groups of demographically similar individuals. Each technical task of the HEDR Project is developing databases for use as input to the HEDRIC codes. Included in these databases is information on weather patterns and historical meteorological observations, environmental factors and characteristics, agricultural practices, and distribution of commercial feed and food products. As input for dose calculations, information is developed for individuals on the basis of geographic location, age, lifestyle, and dietary habits for specified time frames. With this quantity and complexity of input information, it is apparent that the number of variables and potential permutations is very large and the level of detail is very fine. A major undertaking is required to store and manage this much information. The codes and their associated input and output files have been designed to accomplish this goal.

The system of computer codes developed for atmospheric releases, called HEDRIC, is illustrated in Figure 1.3. It consists of four separate but inter-related models and their associated codes: source term, atmospheric transport, environmental accumulation, and individual dose. The source term and atmospheric transport models, documented elsewhere,^{(a)(b)} are briefly described in the following paragraphs.

The source term computer code^(a) provides estimates of historical hourly releases of radionuclides to the atmosphere from the Hanford Site. These release estimates are based on 1) operating histories of the facilities, and

(a) Heeb 1992 (Draft)

(b) Ramsdell 1992 (Draft)

FIGURE 1.3. Organization of the HEDRIC Atmospheric Pathways Codes

2) estimates of release fractions, which are based on the chemical processes and effluent control technologies employed. Uncertainties in the actual amounts released are addressed through use of multiple simulations, each of which represents an alternative release history that is consistent with existing knowledge. Together, these alternative release histories represent the range of releases that could have occurred. Approximately 100 separate realizations of the complete hourly release history will be prepared using this source term code. Thus, the uncertainty in the amount of each hourly release will be represented by a distribution of possible released amounts. The techniques and data used in the source term code are more fully described in the model documentation.^(a)

(a) Heeb 1992 (Draft)

The set of hourly release histories together with hourly meteorological data constitute the input to the atmospheric transport code.^(a) This code provides estimates of the concentration of radionuclides in the air and the deposition rate of the radionuclides onto the earth's surface for each of the grid cells (nodes) illustrated in Figure 1.2. These variables are integrated within the atmospheric transport program on a daily basis. Uncertainties in the source term and imprecision in the input meteorological records are addressed in a manner similar to and compatible with the source term program. Up to 100 separate realizations of the historical conditions by day will be prepared. Each of the realizations will be internally consistent and will represent a possible set of historical conditions consistent with available knowledge. The techniques and data used in the atmospheric transport code are described more fully elsewhere.^(a)

The outputs of the atmospheric transport code (daily air concentrations of radionuclides and daily deposition of radionuclides on surfaces) are used as input to the environmental accumulation code DESCARTES. Version 1.0 of DESCARTES, which is described in this report, calculates time-dependent estimates of the concentrations of radionuclides in soil, plants, and animal products. The code operates on a daily time step, allowing resolution of individual migration patterns throughout the project domain (Figure 1.1). The basic model estimates the quantities of radionuclides in significant environmental media and food products by day at each grid node (Figure 1.2). This generates a set of realizations of spatially and temporarily varying environmental radionuclide concentrations corresponding to the set of realizations of air concentration and deposition produced by the atmospheric transport model. A connected commercial foods submodel is planned to adjust for the fact that agricultural products are not all consumed by humans or fed to livestock at the location at which they are grown (Shipler 1991a). This commercial foods submodel simulates the process of transporting crops and animal products grown at one location to other locations for sale and consumption. This submodel,

(a) Ramsdell 1992 (Draft)

too, is run for each of the model realizations to account for the uncertainties in air concentration, deposition, and radionuclide concentrations in animal feeds and food products.

The codes associated with the source term, atmospheric transport, and environmental accumulation models generate large databases of possible environmental radionuclide concentrations in time and space. The fourth code, the individual dose code CIDER, is also described in this report and uses differing individual physical characteristics, living patterns, and food consumption habits in the estimation of individual doses. This code simulates individual people "living" in the environments defined by the earlier programs and being exposed to the temporally and spatially varying contamination levels. Any one simulation may consist of up to 100 realizations, one for each of the individual source term, atmospheric transport, and environmental accumulation combinations, to allow the construction of distributions of possible doses.

The complete project atmospheric dose estimation code set, HEDRIC, consists of all four codes depicted in Figure 1.3. After running the source term, atmospheric transport, and environmental accumulation codes, a very large database of possible contamination patterns will have been calculated and stored. The first three HEDRIC codes will not need to be run again. The stored information will represent the environmental contamination and associated statistical uncertainty for the entire area for the time period of the simulation.

Dose estimates for any individual or population group within the project domain can be made by accessing the multiple realizations of contamination levels stored in the database. It is anticipated that agencies outside the HEDR Project will desire this capability to estimate doses to actual individuals. Toward the Project's end, a turnover package, consisting of the entire database and the individual dose code, CIDER, will be prepared as illustrated in Figure 1.4. This structure will minimize the need for other groups to run entire simulations.

The structure described for the turnover package is adequate for calculation of doses to individuals. For storing information generated by the codes for large numbers of "representative" individuals and for investigating

FIGURE 1.4. HEDRIC Codes and Output Used in the Turnover Package for Individual Dose Calculations

spatially varying phenomena, the HEDR Project is using a commercial geographic information system (GIS). The GIS will be used to access selected subsets of the entire database, to order and display the selected information for analyses, and to prepare presentation materials. The GIS is independent of the system used to estimate the doses.

1.3 PROPOSED LIFE CYCLES FOR HEDRIC CODES

This document represents the second step in the development of a comprehensive environmental accumulation and dosimetry model for use by the HEDR Project. Computer programs typically go through a "life cycle" of development, testing, refinement, use, and retirement. The code design specification (Napier et al. 1992) is the starting point of the code development. This report documents the implementation of the specifications, with minor changes and enhancements found to be necessary during the coding. Now that Version 1.0 of HEDRIC is complete, it will enter an extensive period of testing, analysis, and refinement before it is ultimately used for production of dose estimates.

Computer codes and their parameters are inextricably intertwined. A major effort is under way to define the parameters that will be used in HEDRIC. The parameters used in the environmental accumulation and individual dose programs are listed in Appendix A. A HEDR report is being prepared on the initial set of values and distributions for dose factors and other model parameters (Shipler 1991a). These initial parameters will be used in the testing and analysis of HEDRIC prior to the planned HEDR sensitivity and uncertainty workshop (Shipler 1991a). Depending on the outcome of this workshop, the parameters may be revised. Values of parameters developed in earlier phases of the project (PNL 1991a, Napier 1991a) are currently being used in the code. The intent in using these parameters is to allow initial programming and testing of the code; the values of these parameters may change upon completion of either the parameter definition subtask or the sensitivity/uncertainty workshop.

The models described in this report compose a limited set of potential exposure pathways. The pathways selected are those that were most likely to have contributed a significant fraction of the total dose to individuals from releases in the early years of Hanford operations. Based on further research and within the framework of HEDRIC, additional pathways may be added. Similarly, of the radionuclides known to have been released from Hanford Site operations, only the dominant radionuclides (those that most likely contributed a significant fraction of the total dose to individuals) (Napier 1991b)

are currently being considered for inclusion in the model. Other pathways or radionuclides will be added only after consideration by the Project's Technical Steering Panel (Shipler 1991a).

In the course of the computer code life cycle, when major changes are made to a code, supplementary documentation is prepared. If accumulated changes warrant, complete new code documentation may be produced. This concept applies to DESCARTES and CIDER. Final code documentation is planned to be completed in FY 1994 (Shipler 1991b).

2.0 MODEL DESCRIPTION

In the HEDR Project, mathematical representations (models) of the physical processes affecting radionuclide transport are used to estimate dose to humans from atmospheric releases of radioactive materials. The processes modeled are 1) release to the atmosphere from fuel and facilities, 2) atmospheric transport and dispersion, 3) translocation and accumulation in environmental media and food products, and 4) exposure of humans and the resultant dose. These four processes correspond to the four major HEDRIC codes.

The models used to estimate environmental accumulation of radionuclides and resultant dose to humans have been changed from those used to produce the Phase I atmospheric pathways dose estimates (PNL 1991a, 1991c). For Phase I, a quasi-steady state model was used to calculate monthly average environmental concentrations. Monthly doses were then calculated from the monthly concentrations (Napier 1991a). In the revised environmental accumulation model, radionuclide concentrations in vegetation and soil are calculated dynamically on a daily basis by solving a set of simultaneous differential equations. The daily soil and vegetation concentrations can be used to calculate daily doses or dose commitments. Integration of these daily values over time can be performed to provide similar information to that which was produced using the quasi-steady state model (NCRP 1984, ICRP 1979).

Whicker and Schultz (1982) discuss dynamic modeling processes similar to the one used here. A major advantage of dynamic models is that they represent environmental radionuclide translocation processes as continuous functions of time and are thus able to yield more precise estimates of transient environmental concentrations and dose. Knowledge of these transients may increase insight into the dynamics of radionuclide translocation patterns and the specific short-term exposure situations that may result from acute releases (ICRP 1979). Because of this greater temporal detail, dynamic models are also better able to account for the effects of short-term movements of people within the study area. Dynamic modeling is therefore the more appropriate methodology for the types and durations of radionuclide releases under investigation in the HEDR Project. In particular, dynamic models use the

daily inputs of radionuclide air concentration and soil deposition calculated by the air transport model more effectively than the Phase I models did.^(a)

This chapter describes the revised models for environmental accumulation and individual dose as implemented in Version 1.0 of the computer codes DESCARTES and CIDER.

2.1 MODEL LIMITATIONS

The current versions of the environmental accumulation and individual dose models and their implementation in computer codes differ somewhat from the design specifications (Napier et al. 1992) as a result of additional insight gained during the code development process. Described below are the general limitations of the modeling approach used. Specific model changes from information presented in the design specifications are discussed in Section 2.2.2 and 2.3.1. Limitations and current status of the associated computer codes DESCARTES and CIDER are described in Section 3.2.

The environmental accumulation and individual dose models are simplified mathematical representations of the environmental and agricultural ecosystems in the regions around the Hanford Site. Because all of the transfer and accumulation processes of these ecosystems are not easily measured or well-known, simplifications are necessary to allow these systems to be modeled effectively. The variability that occurs naturally in each of these processes must also be addressed in a simplified manner.

Because of the unknowns in these natural processes and their natural variability, these models and their computer code implementations are not intended to provide a single estimate of the radiation dose received by a specific individual. When fully developed, these codes will provide a range of doses within which an individual's dose is estimated to fall and to which a quantifiable degree of confidence can be attached. However, because of the incomplete knowledge base and the natural variability in data and parameters, this dose range may extend over an order of magnitude or more.

(a) Ramsdell 1992 (Draft)

The scopes of the environmental accumulation and individual dose models are limited by the scopes of other HEDR models and work tasks providing input data. These tasks, in turn, are limited by practical considerations of what can be achieved within the time and budget constraints of the HEDR Project. Radionuclides originating from atmospheric emissions with subsequent transport through terrestrial environmental pathways are considered in these models. The models include exposure pathways that are generally accepted by the scientific community as being the major dose contributors in most cases. These pathways include submersion in contaminated air, inhalation of contaminated air, direct irradiation from contaminated soil and other surfaces, and ingestion of contaminated food products. Pathways that are considered minor dose contributors, such as contamination entering food products from irrigation water, are not included because expert opinion and/or the results of screening calculations indicate that they are unlikely to make significant contributions to total dose. Special exposure situations may be modeled by adjusting the parameters of the environmental accumulation submodels or the individual dose model. Where needed, additional screening calculations will be used to evaluate minor exposure pathways to assist the Technical Steering Panel of the HEDR Project in determining whether the dose potential is large enough to warrant additions or changes to the equations of the models and associated computer codes.

The detail to which specific differences in input data or parameters can be considered is limited in several ways in these models. For example, no distinction can be made between specific foods within any one of the general food categories (such as cabbage and spinach within the leafy vegetables food category). This is not specifically a model limitation, but rather a limitation of the data available to describe the parameters used in the model. Similarly, the ability to distinguish differences in individual characteristics is limited. Two individuals in the same age and sex class can be distinguished only by differences in food consumption rates and certain minor lifestyle considerations, such as general diet categories and the relative amount of time spent indoors or outdoors. Other specific differences such as race, weight, metabolism, or economic status are not modeled.

Finally, the results of these models are expressed as a measure of radiation dose received. The potential health effects of this dose are not modeled and are outside the scope of the HEDR Project.

2.2 ENVIRONMENTAL ACCUMULATION MODEL

The environmental accumulation model tracks and calculates the accumulation and transfer of radionuclides from initial atmospheric deposition and interception through various soil, vegetation, and animal products compartments. This model contains the set of four linked linear differential equations that give the models their dynamic nature, generating daily soil and vegetation concentrations. Other portions of the model use these daily concentration data and equilibrium-type equations to calculate time-dependent radionuclide concentrations in animal products. The model also performs ancillary calculations required by the core equations. Environmental concentration data needed by the individual dose model are stored in large binary files. Figure 2.1 illustrates the entire environmental accumulation model.

The model function may be visualized as a series of sequential operations. The biomass submodel generates daily biomass values for each plant type modeled. These values are then used in the soil and vegetation submodel to determine the daily concentrations of radionuclides in soil and vegetation. Results are calculated for every grid node, providing the concentration in vegetables, grains, and fruits directly consumed by people and in plants (grass, alfalfa, silage, grain) used for animal feed. Animal feed concentrations are then used to determine concentrations in animal products (beef, venison, poultry, eggs, milk), also on a grid basis. Finally, the radionuclide concentrations in commercially distributed milk are calculated. Values of radionuclide concentrations in soil, vegetation, and animal products are stored for later use by the individual dose model, described in Section 2.3, to calculate individual doses.

2.2.1 Model Development

Although there are a number of food chain transport models and codes documented in the scientific literature, most are steady state or quasi-steady

FIGURE 2.1. Environmental Accumulation Model

state. Few are dynamic in nature. The primary source used for development of the HEDR environmental accumulation model was PATHWAY, developed by Whicker and Kirchner (1987) for dose reconstruction studies around the Nevada Test Site. A number of other dynamic food chain models have also been developed, including TERMOD (Booth and Kaye 1971); RAGTIME (Pleasant et al. 1980); FOOD-MARC (Simmonds and Linsley 1981; Linsley et al. 1982); RADFOOD (Koch and Tadmor 1986); and ECO-SYS (Prohl et al. 1989). PATHWAY was selected as the basis for development of the HEDR environmental accumulation model because it was recently developed, well documented, and developed for use in a region with geographic and climatic characteristics similar to those of the area surrounding the Hanford Site.

Investigations into dynamic food chain models developed by these researchers led to the conclusion that the use of four state variables (the dependent variables, representing two vegetation and two soil compartments for each vegetation type) in the environmental accumulation model was the appropriate approach for this task. In contrast, the PATHWAY model has seven state variables. The additional variables are needed in PATHWAY because the associated code does a mass balance calculation incorporating deep soil sink and radiological sink compartments and contains an animal-specific meat compartment. In contrast, the HEDR environmental accumulation model does not contain mass balance equations because the additional calculations would be very time-consuming. Daily animal product concentrations are calculated using equilibrium equations, unlike PATHWAY which uses differential equations. This difference in calculating animal product concentrations is based on other work performed by Whicker (1985). His work showed that dynamic calculations for animal product concentrations were not warranted because differences in equilibrium calculations and time-integrated dynamic calculations were minimal.

The mechanisms for radionuclide transfer in the environmental accumulation model are generally standard processes that are well documented in the open literature (NCRP 1984, Peterson 1983, IAEA 1982). These processes and other processes that are not radionuclide dependent are described below.

2.2.2 Changes from the Design Specifications

Summarized below are differences between the equations, parameters, and parameter descriptions given here and those in the design specifications (Napier 1991a). Design specification equations are generally referred to first and have an associated equation number (e.g., 3.20). Current model equations are also referred to by number (e.g., 20).

Biomass

Biomass Equations (3.2) and (3.3) in the design specification have been changed from sine expressions to cosine expressions in Equations (1) and (2). The two sets of equations are mathematically equivalent, but the cosine equations are simpler and therefore easier to implement.

Interception Fraction

This equation has been moved forward in the model description text, where it is now Equation (4). It was Equation (3.12) in the design specifications. The basic form of the equation is unchanged, but the interception fraction symbol is now f_v rather than r , for consistency with the model, and the empirical constant symbol has been changed to α from μ , for consistency with common usage.

Soil and Vegetation Concentration

The differential equation describing the time-dependent quantity in the state variable Q_{ov} , Equation (3.7) in the design specifications, contained an error and has been changed to reflect correct compartment inflows and outflows. The corrected equation is shown as Equation (7) in this report.

Equations (3.10a) and (3.10b) of the design specifications, describing the transfer of material between compartments via senescence, have been changed. (Senescence is the process by which plants die and decay or move into winter dormancy.) The design specification formulation does not allow senescence to occur during the transition phase when growth is still occurring but senescence is beginning and increasing. The new formulations, Equations (10) and (11), allow senescence to occur during the transition period.

A related change was made in the equation describing root uptake (3.11). The design specification equation did not allow for any root uptake during the period when the overall growth rate was negative, that is, when senescence was the dominant process, during the late summer and fall. The equation was changed to Equation (12) in this report to allow root uptake to occur, decreasing over time, during the senescence period.

Animal Feeding and Feed Transport

The animal feeding equation in the model description, Equation (14), does not contain all of the terms of the design specifications Equation (3.14) because the data matrix that defines the transport of different types of animal feeds among the grid nodes is not yet available and hence has not been implemented in Version 1.0 of DESCARTES. The parameter symbols for fractional feed type and quantity of feed consumed have been changed slightly, from f to

f_{feed} , and from QF to Q_{feed} , respectively. The term representing the decay time for storage and production of food has been changed to allow modeling of consumption of feed that is 1) fresh, 2) harvested in the current year and stored, or 3) harvested the previous year and stored. Finally, the description of C , the radionuclide concentration in feed, has been changed to remove reference to the surface contamination removal process. This process is applicable only for two types of vegetation directly consumed by humans: leafy vegetables and fruits. Surface contamination removal is now addressed in the individual dose model.

Commercial Foods

In the commercial foods submodel, Equation (3.16b) of the design specifications is not included in the new description because food products other than milk are not included. The need for other food production and distribution information will be evaluated in FY 1992 (Shipler 1991a) and if included will be available in FY 1993 (Shipler 1991b). Equation (3.16a) is included as Equation (16) with minor changes in the parameter symbols: the fraction of a food product at grocery stores from a particular collection center has been changed from F to f_{groc} , and the fraction of a food product at a collection center (creamery) from a particular source has been changed from f to f_{cream} . Also, in the current implementation, the radioactive decay during storage and production is handled in the individual dose model, so the exponential decay term has been removed from this equation.

2.2.3 Submodels

The environmental accumulation model is described in terms of its various constituent parts, called submodels, as shown in Figure 2.1. Each of these submodels contain equations for modeling specific processes of radionuclide transfer. They address biomass, radionuclide concentrations in soil and vegetation, animal feeding and feed transport, concentrations in animal products, and commercial foods accumulation and distribution.

Biomass Submodel

The biomass submodel performs ancillary calculations to generate data required by two pathways for transfer of contaminants from the atmosphere and

soil to plants. Foliar interception of depositing atmospheric contamination depends on the above-ground mass of the vegetation; root uptake, on the other hand, is a function of the rate of growth of the plants. By calculating plant growth rate and integrating the growth to determine total biomass, the biomass submodel provides data to drive both processes.

A plant growth model adapted from that of Whicker and Kirchner (1987) is used for biomass calculations. Instead of using empirical growth rate modifiers for light and temperature, the plant growth rate (dB/dt) is calculated daily using a cosine function modified by a growth rate constant. This function, as modified, approximates the growth rate observed by Whicker and Kirchner (1987). The growth rate is estimated during the period January 1 through July 31 of any year as

$$\frac{dB}{dt} = \frac{k_g}{2} \left[1 - \cos\left(\frac{2\pi d}{365}\right) \right] B \left(\frac{B_{\max} - B}{B_{\max}} \right) \quad (1)$$

where dB/dt = the rate of change of biomass with time, $\text{kg(dry)}/\text{m}^2\text{-day}$, for each plant type

k_g = the growth rate constant, d^{-1}

B = the current daily biomass, $\text{kg(dry)}/\text{m}^2$

$B_{\max}$ = the maximum potential biomass for each plant type, $\text{kg(dry)}/\text{m}^2$

d = the current Julian day (number of days since the start of the year).

Senescence is assumed to affect plant biomass and radionuclide transport between August 1 and December 31 (Whicker and Kirchner 1987). Senescence is considered to be negligible during other months. The rate of change of biomass becomes

$$\frac{dB}{dt} = \frac{k_g}{2} \left[1 - \cos\left(\frac{2\pi d}{365}\right) \right] B \left(\frac{B_{\max}^* - B}{B_{\max}^*} \right) - k_s (B - B_{\min}) \quad (2)$$

where k_s = senescence rate constant (d^{-1})

B_{min} = minimum winter biomass for each plant type, $kg(dry)/m^2$

$$B_{max}^* = \frac{k_g (B_{max})^2}{B_{max} (k_g - k_s) + k_s B_{min}} \quad (3)$$

The term B_{max}^* is necessitated by a maximum realized biomass that would differ between Equations (1) and (2) if B_{max} were not adjusted due to the added senescence term. Use of B_{max}^* preserves the continuity of dB/dt during the transition from the equation governing the non-senescence time period to the equation for the senescence time period. Daily biomass values are currently calculated for nine plant categories. Three categories are modeled as being consumed exclusively by humans: leafy vegetables, other vegetables, and fruit.^(a) Animals are modeled as exclusively consuming four categories: pasture grass (irrigated), alfalfa, silage, and grass hay (Beck et al. 1992). Both humans and animals are assumed to consume grain or grain products. The final plant category, sagebrush, is consumed by neither. Sagebrush, which was the most commonly monitored vegetation type in the early years of Hanford operations, will be used as a "validation ecosystem" when testing and validating the model.

Harvest, which removes plant biomass and associated radionuclide concentration from the submodel, may occur on specific dates, continuously, or not at all, depending upon the plant category. Leafy vegetables, other vegetables, fruit, and pasture grass are assumed to be harvested gradually and continuously, representing the harvesting of fresh produce by humans or grazing by animals. There is currently no biomass loss term for harvesting in these plant categories, i.e., daily concentrations are calculated and stored without concomitant loss of biomass or radionuclide that would occur during actual grazing or harvesting. Grass hay, silage, and grain are modeled as

(a) Callaway, J. M., Jr. 1990. Estimation of Food Consumption. PNL-7620 HEDR (Draft), Pacific Northwest Laboratory, Richland, Washington.

being harvested once per yearly growing season; alfalfa is modeled as being harvested three times per season. Harvest dates are provided from a file of dates specific to each plant category (Beck et al. 1992). On the day of harvest, the biomass is reset to the minimum value, and the concentration per unit surface area of radionuclides in the vegetation is reduced by a proportional amount. The final plant category, sagebrush, is not harvested or ingested.

The daily biomass values and rates of change of biomass calculated are internal variables used by the soil and vegetation concentration submodel and are not stored for outside examination or for use by the individual dose model.

Soil and Vegetation Concentration Submodel

This submodel calculates the concentration of radionuclides in soil and vegetation on a daily basis. Daily vegetation concentrations are needed because plants are directly consumed by humans and by animals whose products, such as milk, meat, or eggs, are then consumed by humans. Soil concentrations are calculated because radionuclides deposited on soil can be transferred back to the air by resuspension or to plants through root uptake, rainsplash, or redeposition of resuspended particles. Contaminated soil directly irradiates exposed humans (grandshine dose) and is also directly ingested by animals during grazing. The different mechanisms for transfer between soil and vegetation compartments are shown schematically in Figure 2.2.

Conditions on a given day are likely to be different at every node on the atmospheric dispersion grid because of local meteorological differences and variations in the length of time required for released material to be dispersed and transported to the different nodes. Environmental concentrations in soil and vegetation depend on local depositions and therefore also vary from point to point and from day to day. The basic model illustrated in Figure 2.2 is implemented each day at each grid node. As shown in Figure 2.1, input comes from the biomass submodel and from the atmospheric transport model.

FIGURE 2.2. Dynamic Compartments in the Environmental Accumulation Model

Initial Deposition and Interception. The first step in the submodel is to segregate atmospheric deposition data provided by the atmospheric transport code^(a) between the outer vegetation and upper soil compartments. Deposition and foliar interception is the basic process for transfer of material from atmosphere to vegetation and soil and, thus, has been widely studied. The interception fraction calculated by the model is based on work by Chamberlain (1970) and observations by Pinder et al. (1988, 1989). The interception fraction is calculated as

$$f_v = 1 - e^{-\alpha B} \quad (4)$$

where f_v = the vegetation foliar interception fraction, dimensionless
 α = empirical foliar interception constant, m^2/kg (dry)
 B = current daily biomass, $kg(dry)/m^2$

The constant α and the variable B are usually defined in terms of dry biomass, B being a daily input from the biomass submodel. The interception fraction is a variable function of plant biomass, which is in turn a function of crop type and time of year. As described below, the soil and vegetation submodel uses nine different plant or crop types, each with a specific calculated interception fraction. The initial soil deposition fraction is calculated simply as the fraction of the total deposition available that is not intercepted by vegetation, or $1 - f_v$. For a more detailed discussion of deposition and foliar interception and selection for this model, see Napier et al. (1992).

Soil and Vegetation Transfer Processes. The same basic environmental transfer and accumulation submodel is used for each of the nine plant categories described in the biomass submodel, although some of the parameters may vary with plant category. In the soil and vegetation concentration submodel, these categories are called "vegetation ecosystems" because each of the plant categories is modeled together with its respective soil compartments. The model is composed of a series of four coupled differential equations, two each

(a) Ramsdell 1992 (Draft)

for soil and vegetation, and is based on work by Whicker and Kirchner (1987) as mentioned in Section 2.1.1. Each equation calculates the daily radionuclide concentration in one of the four compartments shown in Figure 2.2. The equations are

$$\begin{aligned} \frac{dQ_{usl}}{dt} = & I f_s - Q_{usl} (\lambda_{perc} + \lambda_{rad} + \lambda_{splash}) + Q_{ov} \lambda_{weath} \\ & - R_{resus} + R_{senc,iv} + R_{senc,ov} \end{aligned} \quad (5)$$

$$\frac{dQ_{rz}}{dt} = Q_{usl} \lambda_{perc} - Q_{rz} (\lambda_{leach} + \lambda_{rad}) - R_{root} \quad (6)$$

$$\begin{aligned} \frac{dQ_{ov}}{dt} = & I f_v - Q_{ov} (\lambda_{weath} + \lambda_{rad} + \lambda_{trans}) + Q_{usl} \lambda_{splash} \\ & - R_{senc,ov} + R_{resus} \end{aligned} \quad (7)$$

$$\frac{dQ_{iv}}{dt} = Q_{ov} \lambda_{trans} - Q_{iv} \lambda_{rad} + R_{root} - R_{senc,iv} \quad (8)$$

- where
- Q_{usl} = quantity (activity) in the upper soil compartment at time t , Ci/m^2
 - Q_{rz} = quantity (activity) in the root zone soil compartment at time t , Ci/m^2
 - Q_{ov} = quantity (activity) in the outer vegetation compartment at time t , Ci/m^2
 - Q_{iv} = quantity (activity) in the inner vegetation compartment at time t , Ci/m^2
 - I = input areal deposition rate from the output of the atmospheric transport model for this location and day t , $Ci/(m^2 \cdot d)$
 - f_v = vegetation foliar interception fraction, dimensionless
 - f_s = soil deposition fraction, equal to $1 - f_v$
 - λ_{perc} = percolation rate constant, d^{-1}
 - λ_{rad} = radiological decay constant, d^{-1}

- λ_{splash} = rainsplash rate constant, d^{-1}
 λ_{weath} = weathering rate constant, d^{-1}
 λ_{trans} = plant translocation rate constant, d^{-1}
 λ_{leach} = leaching to deeper soil rate constant, d^{-1}
 R_{resus} = rate of radionuclide redeposition on vegetation from resuspension of soil, $\text{Ci/m}^2 \cdot \text{d}$
 R_{senc} = rate of radionuclide transfer from plants to soil by senescence of vegetation, $\text{Ci/m}^2 \cdot \text{d}$
 R_{root} = rate of radionuclide uptake through roots, $\text{Ci/m}^2 \cdot \text{d}$.

Resuspension and Redeposition. Resuspension is a process where contamination is lifted from the soil surface and can then be either redeposited on soil or vegetation or be available for inhalation. This so-called "resuspension rate" is a function of the upper soil layer contamination, as

$$R_{\text{resus}} = Q_{\text{ubl}} \text{ML} V_d / \rho_1 \quad (9)$$

where ρ_1 = the soil surface areal density to a depth of 1 mm, kg/m^2

ML = the mass loading factor of local soil in air, kg/m^3

V_d = the deposition velocity of dust for local soil or vegetation, m/d .

Sehmel (1984) discusses deposition and resuspension in detail, and, in particular, addresses airborne dust loading around Hanford. Mass loading is also considered by the NCRP (1984). The mass loading factor represents the typical dust loading over an area or region and provides a better estimate of general area resuspension than does use of a resuspension factor, which is based on very localized resuspension. The deposition velocity is an empirically determined parameter based on the ratio of measured deposition per unit area to integrated air concentration.

Senescence of Vegetation. The process by which plants die and decay or move into winter dormancy provides a mechanism for transferring contamination from vegetation to the soil surface. This process, called senescence, is coupled with the biomass submodel. The model that used by Whicker and Kirchner (1987) and tracks the transfer of contamination within plant tissues and on plant surfaces (subscripts iv and ov, respectively) down to soil, as

$$R_{senc, iv} = 0 \text{ from January 1 to July 31}$$

$$R_{senc, iv} = \frac{Q_{iv}}{B} k_s (B - B_{min}) \text{ from August 1 to December 31} \quad (10)$$

$$R_{senc, ov} = 0 \text{ from January 1 to July 31}$$

$$R_{senc, ov} = \frac{Q_{ov}}{B} k_s (B - B_{min}) \text{ from August 1 to December 31} \quad (11)$$

Equations (10) and (11) allow senescence losses from vegetation compartments to be considered even when the biomass is increasing. Biomass is modeled to continue to increase during the first few weeks in August after the biomass submodel equation is changed on August 1 to include senescence effects. Senescence gradually increases and internal plant growth decreases until the daily growth rate becomes negative some time in late August.

Uptake Through Roots. The uptake of soil contamination by plant roots is driven by the growth rate of the biomass. Plant uptake via roots draws upon the inventory in the soil rooting zone compartment. The model used is a slight variant from that of Whicker and Kirchner (1987), which incorporates the commonly measured concentration ratio (CR), given as Ci/kg of plant (dry) per Ci/kg of soil, and the growth rate. However, Equation (12) uses soil areal density, which incorporates soil bulk density and root zone depth. The root uptake rate is modeled as

$$\begin{aligned}
 R_{root} &= Q_{rz} \frac{CR}{\rho_2} \left(\frac{dB}{dt} \right) \text{ from January 1 to July 31} \\
 R_{root} &= Q_{rz} \frac{CR}{\rho_2} \left(\frac{dB}{dt} + k_s(B - B_{min}) \right) \text{ August 1 to December 31}
 \end{aligned}
 \tag{12}$$

where ρ_2 is the soil areal density from 1 mm to 15 cm depth, kg/m².

Harvest and Tillage. Two types of events that are essentially external to the submodel function, harvest and tillage/planting, must be considered in the removal or transfer of radioactive material in the submodel. Both are currently modeled as discrete events. Harvest is the removal of a fraction of the inventory contained in the two vegetation compartments, with the fraction removed being the ratio of the existing pre-harvest biomass minus the minimum biomass to the existing biomass. Tillage results in the mixing of the inventories of the upper and lower soil compartments. The result is the total inventory of the area apportioned by mass between the two layers. The soil surface layer is assumed to be 0.1 cm in depth, and the root zone layer is from 0.1 cm to 15 cm in depth. Tillage and planting are assumed to take place once in the spring, and affect only four of the vegetation ecosystems: leafy vegetables, other vegetables, grain, and silage. Harvest and tillage dates are taken from the file of such dates (Beck et al. 1992). The planting date of annual crops (and hence the date at which the biomass begins to grow) is assumed to be identical to the tillage date.

Translocation, Weathering, Rainsplash, Percolation and Leaching. All of these transfer mechanisms are modeled as continuous processes in the core differential equations, after Whicker and Kirchner (1987). This is done using rate constants, which may be considered as indicators of the fractional change per unit time. The use of all of these processes for radionuclide modeling has previously been described (Simmonds and Linsley 1981; NCRP 1984; Peterson 1983; Koch 1986; Whicker and Kirchner 1987), although the representation of them as rate constants in dynamic modeling is somewhat new.

Translocation is the process whereby contamination on the outer surface of the plant is absorbed or transferred into the inner tissues of the plant (Hungate et al. 1963). Currently, no distinction is made between the translocation rates for the different types of plants; all use the same translocation rate constant. Weathering is the removal of contamination from the plant surface, typically by meteorological conditions, and transfer to the soil surface (Chadwick and Chamberlain 1970).

Rainsplash is a mechanism for transferring contamination from the soil surface back to the plant surface by the action of water falling on the soil (Anspaugh 1987). Whicker and Kirchner (1987) were the first to include this pathway. Currently, irrigated and unirrigated land are treated identically, as unirrigated (semi-arid) land. Only sprinkler-type irrigation is important to rainsplash, and this type of irrigation was not prevalent during the time period currently addressed.

Percolation and leaching are both soil transfer processes (Baes and Sharp 1981). Percolation is transfer from the upper soil layer to the root zone; leaching is transfer from the root zone to the deep soil sink where contamination becomes unavailable for further transfer. The deep soil sink is a modeling convenience representing loss of radionuclides from the system. It may be a soil depth to which plant roots do not penetrate or a soil matrix to which contamination is irreversibly bound and therefore no longer available to plants.

Radioactive Decay. Removal of radionuclides from the submodel by radioactive decay is included in each of the differential equations. Radioactive decay is described by a rate constant and reduces the total radionuclide inventory.

Daily Concentration Values. The state variables defined by the above equations represent the status of the environment on a given day. The information generated for each compartment (state variable) of each vegetation ecosystem for each day is stored and used as the input to all additional portions of the atmospheric pathways model. The computational algorithm uses total inventory (activity per unit area) rather than concentration. Vegetation data

are saved as units of concentrations (activity per unit mass) for input to the individual dose model. The conversion from inventory (Q) to concentration (C) (quantity per unit biomass) is

$$C = Q/B \quad (13)$$

where, as above, C is the concentration (Ci/kg), Q is the inventory per unit surface area in each compartment (Ci/m²), and B is the biomass per unit surface area (kg/m²). The term C represents the radionuclide concentration in both animal feed and plants consumed by humans; the same term is used in two different equations (inner and outer vegetation compartments) for each plant type.

Most, but not all, vegetation concentration data are saved by DESCARTES and stored with other environmental concentrations. For most animal feed vegetation categories (pasture grass, alfalfa hay, grass hay, silage), concentrations from inner and outer vegetation compartments are combined and saved because there is no distinction between the two compartments when the feed is ingested by animals. For leafy vegetables and fruits, inner and outer vegetation concentrations are saved separately because the outer vegetation portion is subject to contaminant removal by food preparation techniques. For the categories of other vegetables and grain, the outer vegetation concentrations are not saved because outer or exposed portions of the plants are discarded and are not ingested.

Animal Feeding and Feed Transport Submodel

Output from the soil and vegetation submodel provides the concentration of radionuclides in these environmental media at each grid node. Currently, animals are assumed to obtain feed and fodder from the grid area where they live, and no transport of feed between grid areas is considered. A feed transport model analogous to the milk transport model described as part of the commercial foods submodel will be added when feed transport data become available.

Dairy cows are assumed to have eaten several types of diets, with a diet defined as a sum of fractional feed-type intakes. Five types of feed are available: alfalfa hay, pasture grass, grass hay, silage, and grain. The amount of each feed type eaten by the various animals on a particular feeding regime may be graphically represented as shown in Figure 2.3 (for illustration only). There are currently four dairy cow feeding regimes for the HEDR study area (Beck et al. 1992). Two of these feeding regimes are characteristic of irrigated farms. The first regime consists of grain, stored alfalfa hay, fresh irrigated pasture, and silage. The second consists of grain, stored alfalfa hay, and fresh irrigated pasture. The other two feeding regimes are characteristic of non-irrigated farms. Regime three consists of alfalfa hay fed year-round, with a minor grain supplement. Regime four consists of grass hay fed year-round, with a minor grain supplement (Beck et al. 1992).

Beef cattle, deer, and poultry (producing both meat and eggs) are currently considered in the animal feeding submodel in addition to dairy cattle. Beef cattle are assumed to eat pasture grass only and are also assumed to eat different (smaller) total feed quantities than dairy cattle

FIGURE 2.3. Example Schematic of Dairy Cow Feeding Regime Input Data (Whicker and Kirchner 1987)

(Beck et al. 1992; Comar 1966). Deer are assumed to eat still smaller quantities of fresh forage only (represented by pasture grass) (Urness 1981), while the poultry diet includes grain and fresh forage (Ensminger et al. 1990). All animals are considered to ingest small amounts of soil during the normal feeding process, the quantity of soil varying with the different animals and diets (Darwin 1990).

The quantity of a radionuclide consumed by a "typical" animal at a given location on a given day can be expressed as

$$A_{\text{cons}}(d, l) = \sum_{t=1}^T f_{\text{feed}}(d, l, t) Q_{\text{feed}}(d, l) C_{\text{feed}}(j, l, t) e^{-\lambda_{\text{rad}} t_s} \quad (14)$$

where $A_{\text{cons}}(d, l)$ = radionuclide consumption rate of the animal at location l on day d , Ci/d

d = current day

l = location

t = feed type

T = total number of feed types

$f_{\text{feed}}(d, l, t)$ = fraction of the total quantity of feed type t eaten by the animal at location l during day d . The sum of these fractions over all feed types on day d must equal 1.0 (Beck et al. 1992).

$Q_{\text{feed}}(d, l)$ = total quantity of feed the animal eats during day d (kg/d dry weight) at location l (Beck et al. 1992).

$C(j, l, t)$ = radionuclide concentration (Ci/kg) in feed of type t harvested on day j at location l from soil and vegetation model [equal to $(Q_{iv} + Q_{ov})/B$ for the proper vegetation type]

λ_{rad} = radioactive decay constant, d^{-1}

t_s = decay correction time for stored feed; where t_s
 = $d - j_t$, if $d - j_t > 0$; for stored feed harvested and consumed during the current year
 = $365 - (d - j_t)$, if $d - j_t \leq 0$; for stored feed harvested during the previous year and consumed this year
 = 0 for feed harvested and consumed fresh.

where $(d-j_t)$ = decay correction time for feeds not consumed fresh, d

j_t = Julian day on which feed type t was most recently harvested.

The definitions of feed include the dates when feed types were harvested or when animals were put on pasture. (Thus, each of the three cuttings of alfalfa during the growing season is considered to be a different "type" of feed.)

Information produced by this submodel is used only by the animal products submodel to generate concentrations of radionuclides in animal products. Therefore, the information does not need to be stored for later use by the individual dose model.

Animal Product Concentration Submodel

There are currently five animal products included in the model: cow's milk, eggs, poultry, beef, and venison. The three animal product categories of milk, meat, and eggs are commonly recognized as being the potentially significant dose contributors (Peterson 1983; NCRP 1984). Preliminary knowledge of agricultural patterns in the study area indicated inclusion of poultry and beef because large quantities of these products were probably consumed in the study area (Callaway 1990). Venison was included to consider an additional potential exposure pathway for Native Americans.

The concentration of radionuclides in animal products is a function of the time-varying diets of the animals. The animal product submodel addresses animal product radionuclide concentrations in a simpler, quasi-steady-state manner, assuming equilibrium conditions and using feed-to-animal product transfer factors. With such a model, the concentration of a radionuclide in animal products on a given day is directly related to that day's intake of that radionuclide. This method provides the correct integrated concentrations of radionuclides in animal products and, thus, the correct estimate of dose from intake of such products, but loses some of the temporal (day-to-day) resolution. Use of this formulation in dynamic pathway codes was suggested by the work of Whicker (1985) and is discussed in greater detail in Napier et al. (1992).

The equation for estimation of radionuclide concentrations in animal products can be written using the intake rate developed in Equation (14) as simply

$$A_p(d, l) = \left(A_{cons}(d, l) + \frac{FS_a Q_{usl}(d, l)}{\rho_1} \right) TF_p \quad (15)$$

where $A_p(d, l)$ = the concentration in the animal product p on day d at location l , Ci/L (milk) or Ci/kg (meat, eggs, etc.)

TF_p = the transfer coefficient applicable to this animal product for the radionuclide under consideration, in d/L (milk) or d/kg (meat, eggs)

FS_a = animal soil ingestion rate, kg/d.

Because these data are used directly in the dose model, the daily concentration of each radionuclide in each animal product at each location is stored.

Commercial Foods Accumulation and Distribution Submodel

This submodel takes into account the commercial production and distribution of various types of food. Foods produced at various locations may be combined or pooled at collection centers, such as creameries and packing plants. These foods may then be redistributed to grocery stores throughout the region. Thus, foods eaten by people at one location may have come from one or more other locations. The only commercial food currently included in the model is cow's milk (Beck et al. 1992). An analogous submodel will be included for other commercially grown foods if other commercial foods are found to be potentially important dose contributors (Shieler 1991a).

The concentration of radionuclides in a particular type of commercially available food is given as

$$F_{pa}(d, l) = \sum_{c=1}^C f_{groc}(l, c) \sum_{l=1}^L f_{cream}(l, c) A_p(d, l) \quad (16)$$

where $F_{pa}(d, l)$ = the undecayed concentration in commercially available animal product p (milk) produced on day d and commercially available at location l, Ci/kg

$A_p(d, l)$ = radionuclide concentration (Ci/kg) in animal product produced at location l on day d

C = total number of collection centers

$f_{groc}(l, c)$ = fraction of food product at grocery stores at location l that came from collection center c

L = number of source locations

$f_{cream}(l, c)$ = fraction of food product from source location l at collection center c.

Because these data are used directly in the dose model, the daily concentration of each radionuclide in each food product is stored.

2.3 INDIVIDUAL DOSE MODEL

The source term, atmospheric transport, and environmental accumulation models establish the concentrations of radionuclides in environmental media and food products for all locations and times of interest. In the individual dose model, human receptors are introduced into the calculation. The dose model calculates dose via four exposure pathways: submersion in contaminated air, inhalation of contaminated air, irradiation from contaminated surfaces and soils, and ingestion of farm products and vegetation (Peterson 1983, NCRP 1984). The individual dose model is currently designed for the calculation of doses to reference individuals. The general structure for this model is shown in Figure 2.4. Inputs to this model are the outputs from both the environmental accumulation model and the atmospheric transport model, plus generic consumption and exposure rates for each category of individual type (age, sex, lifestyle).

FIGURE 2.4. Individual Dose Model

Currently, for a given individual, the dose model calculates the radiological dose to a single organ, the thyroid, from a single radionuclide, iodine-131, at a single location. For the time an individual resided at a given location, doses from external exposure and inhalation and most ingestion pathways are dependent solely on the local characteristics. The exception for ingestion is milk, where doses may result from consumption of either local or commercial (grocery store) milk.

2.3.1 Changes from the Design Specification

Minor changes have been made to the equation for calculating external dose from air submersion (3.17). The term χ (Chi) in Equation (17) is actually defined as the integrated daily concentration of radionuclides in air ($\text{Ci}\cdot\text{s}/\text{m}^3$), so the units for DS are rad, the integrated daily external dose from air submersion. For units consistency a time conversion factor was added to the equation. A similar change was made for Equation (3.19) used to calculate the internal dose from inhalation. In Equation (19), D_{INH} is the

integrated daily committed dose from inhalation, with units of rad. The equation includes integrated air concentration and local mass loading, so the mass loading is multiplied by the number of seconds in a day to provide a similar integrated term. To account for the amount of time spent outdoors and breathing the outside air concentration, a second term, f_{time} , is added. This term is unitless and is defined as the fractional daily exposure period for inhalation of contaminated outside air. This parameter is dependent upon the amount of time an individual spends outside exposed to contaminated air and resuspended contamination and is therefore a general lifestyle indicator.

Ingestion Dose from Local and Commercially Available Food

With the exception of milk, doses from ingestion of locally and commercially produced foods are considered together in CIDER because milk is the only commercially produced food product for which distribution data are currently available. The symbol for ingestion dose has been changed from D_{cf} in Equation (3.20) to D_{ing} in Equation (20). Also, as noted under the commercial foods submodel above, the decay for storage and production time is handled in this equation, so the exponential decay term has been added to this equation. This particular equation is applicable only to animal products and those human-edible plant categories for which only inner compartment concentrations are saved for dose calculations (other vegetables, grains). For leafy vegetables and fruits, a separate equation (21) has been added to account for the separate inner and outer vegetation compartments considered in the calculation, with a parameter (WFA) to account for the potential loss of surface contamination caused by the food preparation processes. New parameters are $F_{\text{p,iv}}$ and $F_{\text{p,ov}}$, the radionuclide concentrations in the inner and outer vegetation compartments, respectively.

2.3.2 Age-Dependent Dose Factors

Doses are calculated for people of various ages because an individual's dose response to a given intake amount changes with age (ICRP 1989). Dose factors are provided for seven age/sex groups (Table 2.1) (ICRP 1989). Each age group is also subdivided into male and female. Dosimetry for male and female children through about age 15 is essentially the same and is modeled as being identical; the only potential variable is the difference in food

TABLE 2.1. Age Groups Represented in Dose Model

<u>Age/sex Group</u>	<u>Age Basis for Dose Factor</u>	<u>Nominal Body Mass</u>
newborn	3 months	3.4 kg
1 - 2 yr	1 yr	9.8 kg
2 - 7 yr	5 yr	19 kg
7 - 12 yr	10 yr	32 kg
12 - 17 yr	15 yr	55 kg
adult female	17+ yr	58-kg
adult male	17+ yr	70-kg

consumption by the sexes. After age 15, the dosimetric calculations vary as well. The dose factors are linearly interpolated to each calendar year of an individual's life from the age-specific dose factors available. Currently, age-dependent dose factors for iodine-131 are taken from ICRP Publication 56 (ICRP 1989).

2.3.3 Location-Specific Doses

Doses from external exposure and inhalation are functions only of location and age. The equations presented below are commonly used in environmental dosimetry calculations; they have previously been integrated into a set of environmental dosimetry computer codes, as used here, by Napier et al. (1988).

The dose from submersion in contaminated air is calculated using a semi-infinite plume approximation because the equation is computationally simple and is a conservative method for estimating doses from this pathway. It has previously been determined that air submersion is a minor pathway with small dose contribution from radionuclides that have been investigated in the HEDR Project (Napier 1991b). Calculation of external air submersion dose is simply the integrated air concentration multiplied by an air submersion dose factor and a shielding factor (≤ 1.0) to account for building shielding while indoors (Napier et al. 1988). The equation is given for daily exposures but may be integrated to any period through simple summation over time:

$$DS(d, l) = \chi(d, l) DFS Sh1 CF \quad (17)$$

- where $DS(d, l)$ = the daily air submersion dose for day d at location l , rad
- $\chi(d, l)$ = the integrated daily concentration of radionuclide in air at location l on day d , $Ci \cdot s/m^3$
- DFS = the semi-infinite plume external dose rate factor (for each radionuclide), rad/d per Ci/m^3
- $Sh1$ = a shielding factor to account for time spent indoors or in other shielded locations, dimensionless.
- CF = time conversion factor, $1.16 \times 10^{-5} d \cdot s^{-1}$

A similar equation is used to calculate dose from external exposure to contamination deposited on the ground, called "groundshine." Dose is calculated using an infinite slab approximation and a shielding factor (Napier et al. 1988) (e.g., assuming contamination is uniformly spread over a large area surrounding the exposed individual):

$$DG(d, l) = (Q_{usl}(d, l) DFU + Q_{rz}(d, l) DFR) Sh2 \quad (18)$$

- where $DG(d, l)$ = the dose rate from groundshine at location l on day d , rad/d
- $Q_{usl}(d, l)$ = the concentration of the radionuclide in the upper soil layer from the environmental accumulation model, Ci/m^2
- $Q_{rz}(d, l)$ = the concentration of the radionuclide in the rooting zone from the environmental accumulation model, Ci/m^2
- DFU = the dose rate factor for surface plane deposition (for each radionuclide), rad/d per Ci/m^2
- DFR = the dose rate factor for radionuclides mixed in the soil (for each radionuclide), rad/d per Ci/m^2 (similar units for a different geometry)

Sh2 = shielding factor for groundshine plane sources to account for time spent indoors or in other shielded locations, dimensionless.

As before, integrated doses may be calculated by summing daily doses over the time period of interest.

Radioactive material may be inhaled directly from a passing cloud or from contaminated surface dust suspended back into the local air. The dose from inhalation (D_{INH}) is not received immediately; the material deposited in the body continues to irradiate the individual, but the dose received over time from a single day's inhalation is credited to that day using the concept of committed dose. This concept allows summation of inhalation exposures over a specified period and assignment of a committed dose to that period (ICRP 1977). For simplicity, internal committed dose is referred to as dose. The concept of committed dose allows internal and external daily doses to be added together even though, strictly speaking, the internal dose has not been fully received. Both internal dose and quantity of material inhaled are functions of age. Thus, the dose received is

$$D_{INH}(d, l) = f_{time} \left(\chi(d, l) + Q_{usl} \frac{ML}{\rho_1} S \right) BR \cdot DFI \quad (19)$$

where $D_{INH}(d, l)$ = dose from inhalation at location l on day d, rad

$\chi(d, l)$ = the integrated daily concentration of radionuclide in air at location l on day d, Ci·s/m³

ML = the mass loading factor of local soil in air, kg/m³

ρ_1 = the soil surface density to a depth of 1 mm, kg/m²

BR = the breathing rate appropriate for the age of the individual, m³/s

DFI = the inhalation dose factor appropriate for the age of the individual, rad/Ci inhaled

S = 86,400 seconds (in one day)

f_{time} = the fraction of a day spent exposed to contaminated air.

2.3.4 Ingestion of Local and Commercially Available Foods

Committed dose from ingestion of foodstuffs is calculated by multiplying the foodstuff radionuclide concentration by the amount consumed and the appropriate dose factor and summing over all foodstuffs and radionuclides (Soldat, Robinson, and Baker 1974). The levels of radionuclide contamination of local food crops and animal products is available in the database prepared using the environmental accumulation model. Distribution of commercially available foods (milk) (Beck et al. 1992) is handled in the commercial foods submodel; the concentration in commercial milk at local grocery stores is also available from the environmental accumulation database. The model assumes that all commercial milk products consumed are purchased locally. The ingestion dose equation also corrects for the decay of the radionuclides during handling and storage.

The following notation assumes an implicit summation over all radionuclides. Therefore, the dose resulting from eating foods contaminated with these radionuclides is

$$D_{ing}(d, l) = DFG \sum_{p=1}^P F_p(d - th_p, l) R_p e^{(-\lambda_{rad} th_p)} \quad (20)$$

where $D_{ing}(d, l)$ = the dose from local or commercial foods consumed at location l on day d , rad

DFG = the ingestion dose factor appropriate to the age of the individual, rad/Ci

R_p = the consumption rate of food product p , kg/d

P = the total number of food products ingested

$F_p(d - th_p, l)$ = the food product total concentration on day $(d - th_p)$ at location l from the environmental concentrations data files, Ci/kg

th_p = holdup time, from production or harvest to consumption, days.

Equation (20) is applicable to animal products and those plant categories where only the inner vegetation compartment concentrations are saved for dose

calculations (other vegetables, grains). For two types of plants ingested by humans, (leafy vegetables and fruits) inner and outer vegetation compartment concentrations are stored separately to allow the removal of surface contamination by food preparation activities. For these two plant categories,

$$D_{veg2}(d, l) = DFG \sum_{p=1}^P (F_{p,iv}(d-th_p, l) + F_{p,ov}(d-th_p, l) WFA) R_p e^{(-\lambda_r th_p)} \quad (21)$$

where $D_{veg2}(d, l)$ = the ingestion dose from the two plant types at location l on day d , rad

$F_{p,iv}(d-th_p, l)$ = the radionuclide concentration of the inner vegetation on day $(d-th_p)$ at location l , from the environmental concentrations Database, Ci/kg

$F_{p,ov}(d-th_p, l)$ = the radionuclide concentration of the outer vegetation on day $(d-th_p)$ at location l , from the environmental concentrations Database, Ci/kg

WFA = a surface washing fraction to account for food preparation methods that remove surface contamination, e.g., washing or peeling, unitless.

2.4 FUTURE MODELING ENHANCEMENTS

Investigation into model enhancements and refinements is scheduled to continue throughout FY92 and FY93 (Shipler 1991b) until the implementing codes are finalized and documented in FY94. Planned expansion and refinements to DESCARTES and CIDER are discussed in Section 3.8. In addition, several areas have been identified for investigation and possible incorporation into these codes.

One of the areas is the deposition of airborne radioactive material onto stored forage, subsequent ingestion by livestock, and transfer to milk. Black and Barth (1976) discuss this pathway for iodine-131 ingested by cows following atmospheric nuclear testing at the Nevada Test Site. These authors also discuss inhalation of airborne radioactive material by livestock and transfer to milk.

Also under investigation is the inclusion of a biomass and radionuclide loss function for vegetation that is grazed or harvested continuously (daily), including leafy vegetables, other vegetables, fruit, and pasture grass. Whicker and Kirchner (1987) include such a function in the PATHWAY code.

Investigation will continue into other environmental accumulation pathways and modeling functions that may contribute to individual dose from past Hanford atmospheric releases. In particular, the direction of the Technical Steering Panel of the HEDR Project will be sought in identifying possible model enhancements.

3.0 USER'S GUIDE

Users can run the atmospheric pathways codes to obtain dose estimates for individuals under specific conditions. This User's Guide provides an overview of the function and operation of the computer codes for environmental accumulation and individual dose; a discussion of current code status and limitations; and detailed instructions for using the codes.

3.1 OVERVIEW OF CODES

Calculations of the environmental accumulation of radionuclides and the subsequent dose to humans are performed by two computer codes: DESCARTES, the code that implements the HEDR environmental accumulation model, and CIDER, the code that implements the individual dose model. Two other programs are also required to fully implement the calculation process. PARSEL, the input parameter selection and generation program, selects and stores parameter values to be subsequently used by DESCARTES and CIDER for stochastic calculation of accumulation and dose. REPGEN is a post-processor and report-generator program that selects and reports output information from both DESCARTES and CIDER. This section provides instructions for running the current operating version (Version 1.0) of each of the four programs.

Version 1.0 of PARSEL, a precursor to the full parameter selection program, generates a single set of input parameter values by using constant values of deterministic parameters and best estimates of parameters that are modeled by probability distributions. The full implementation of PARSEL, scheduled for completion later in FY 1992, will create multiple input parameter files by generating multiple random samples from each parameter distribution, taking into account known or estimated correlations among parameters. Each parameter file represents a separate realization of parameter values to be input to an independent Monte Carlo simulation run.

The environmental accumulation code, DESCARTES, uses the parameter values in the PARSEL files and the air concentration and deposition values in the atmospheric transport model output files to calculate and store a database of radionuclide concentrations in environmental media and food products. This

database represents one or more possible realizations of concentration values over space and time. (Because the current version of PARSEL is capable of generating only one input parameter set for one model realization, DESCARTES is currently limited in practice to one simulation run, although there are no such inherent limitations in the DESCARTES code itself.)

Once DESCARTES has generated the database of concentrations in environmental media and food products, CIDER can be run to compute one or more realizations of the estimated dose received by an individual. (Because the current version of PARSEL is capable of generating only one input parameter set for one model realization, CIDER is also currently limited in practice to one simulation run, although there are no such inherent limitations in the CIDER code itself.) CIDER retrieves the portions of the environmental concentrations database that contain the location-specific environmental and food product concentrations to which the individual was assumed to be exposed during the time period of interest. CIDER then uses these concentrations and the appropriate values from the input parameter files to generate estimates of the dose the individual received via each exposure pathway. These dose estimates are then written to a set of output files. CIDER can be executed many times, accessing the same environmental concentrations database that was produced with one execution of the DESCARTES code to compute dose estimates for many individuals.

The current version of REPGEN, a precursor to the final report generator program, retrieves data from individual dose estimate files and/or environmental database files as necessary to report and summarize user-selected environmental media concentrations, food product concentrations, and dose estimates. The full implementation of REPGEN, scheduled for completion later in FY 1992, will permit the user to select the types of distributional statistics to be calculated and reported for each selected model output variable.

Section 3.2 discusses the status and limitations of the current version of the programs. Section 3.3 contains information on accessing the computer system on which the programs are currently implemented. Sections 3.4 through

3.7 describe the operation of the current version (Version 1.0) of each of the four programs. Section 3.8 provides information on planned expansion and refinement of the computer programs and underlying models.

3.2 CURRENT CODE STATUS AND LIMITATIONS

The initial (Version 1.0) code implementation has retained some of the scope and data limitations of the Phase I code (Napier 1991a). Many of these limitations will be addressed later in the Project, as discussed in Section 3.8. Iodine-131 is the only radionuclide for which environmental transfer and accumulation and radiation doses are currently calculated. The geographical area over which these calculations are made is limited to a 31 x 31 grid; the spatial variations of input data and parameters (with the exception of the atmospheric transport model results) are based on the 10-county Phase I census subdivisions. The time period for available data remains the three years from 1945 through 1947, plus the last six days of 1944 (to include the first known significant atmospheric releases from Hanford Site operations, beginning on December 26, 1944). Because no allowances have yet been made for individual food consumption patterns other than the source of dairy products, dose calculations are currently performed for a reference individual of a specified age and sex living at a specified location and consuming dairy products from a specified source.

As with the Phase I code, the exposure pathways addressed are submersion, inhalation, groundshine, ingestion of local vegetables and fruits, ingestion of locally and commercially produced milk, and ingestion of other animal products. These are the primary potential exposure pathways. Secondary exposure pathways, such as water ingestion and contribution from irrigation water, are not addressed. There is as yet no provision for customized (non-standard) feeding regimes. All feed for animals is assumed to have been grown at the same grid location where the animals were fed. All vegetation and animal products (except milk) consumed by an individual are assumed to have been produced at the local grid location where the individual lived. The only commercial food included is milk; the code allows for movement of milk from areas of production to collection centers (creameries) and then to

grocery stores. The code assumes that tillage and planting occur on the same date and only in the spring, with no allowance for fall planting of winter wheat or other winter grain crops. The pasture grass ecosystem currently forms the basis for calculating groundshine doses. A "residential vegetation ecosystem" is planned that would simulate a residential lawn and be used to calculate groundshine doses.

The thyroid is the only organ for which dose (in rad) is calculated. Dose to nursing infants or pre-natal doses are not yet calculated, and there is no capability yet to compute doses to population groups. The dose code is limited to computing dose for one location of residence during one specified time period. However, the code can be run multiple times, each covering a different time period and residence location for the same individual. The code is currently not capable of computing doses received at work locations that differ from residence locations.

3.3 COMPUTER ACCESS

The computer programs described in this report are currently implemented on the PNLZ VAX Cluster at the Pacific Northwest Laboratory. During mid-FY 1992 the code will be transferred to a SUN 4/490 workstation for dedicated HEDR use. To run the programs, you must have an active personal account on the PNLZ cluster and be authorized to access the HEDR Project account. To set your workspace area to the HEDR Project account, you must type the following line at the VMS \$ prompt or include it in your automatic log-in process:

SET PROJECT HEDR

You are then ready to begin running any of the programs described in this report.

3.4 INPUT PARAMETER SELECTION PROGRAM: PARSEL

PARSEL must be run before the environmental accumulation and dose calculation codes are run. Version 1.0 of PARSEL is currently limited to generating a single set of best-estimate input parameter values for a single

Monte Carlo model run. This implementation of PARSEL is sufficient to allow the initial versions of the computational codes (DESCARTES and CIDER) to be developed and tested.

PARSEL can be run interactively on the VAX from any directory within the HEDR Project account by typing

PARSEL

at the VMS \$ prompt. Sections 3.4.1 and 3.4.2 describe the interactive prompts and corresponding responses that must be entered by the user. Section 3.4.3 explains how to run PARSEL as a background (batch) process on the VAX.

3.4.1 Entering Program Control Options

The current version of PARSEL prompts the user to enter the length of time, in days, for which time-dependent parameter values are to be generated:

Enter the total number of days (beginning with 12/26/1944):

Enter a positive integer. This number must be less than or equal to the number of days contained in the atmospheric transport data files that will be used for the current simulation.

(NOTE: Each of the programs assumes that December 26, 1944, the date of the first known significant atmospheric release from Hanford operations, is the first day of the simulation time period. The last 6 days of 1944 are considered a part of the year 1945 for simulation purposes.)

PARSEL currently generates only one input parameter value set containing constant or best-estimate values for all parameters on a 31 x 31 spatial grid. When fully developed, PARSEL will prompt the user to enter the desired number of Monte Carlo runs (parameter value sets) and the dimensions of the spatial grid.

3.4.2 Selecting Parameter Values

Because the current version of PARSEL uses predetermined best-estimate values for all input parameters that are represented by probability distributions, there are no user-selectable distributional types, distribution parameters, or coefficients. When fully developed, PARSEL will allow the user to choose distribution input parameters for which there now exists considerable uncertainty in the form and/or quantification of the probability distribution. Coefficients of correlation between specific pairs of input parameters may also be user-selectable.

3.4.3 Running PARSEL as a Background (Batch) Process

PARSEL can be directed to run in the background as a batch process on the VAX. There is only one run-time input value to be placed in the input stream: the number of days (beginning with December 26, 1944) for which input parameter values are desired. To submit PARSEL as a batch process, type

BG_-PARSEL [number-of-days]

at the VMS \$ prompt, where [number-of-days] is replaced by an integer denoting the number of days (beginning with December 26, 1944) for which you wish to generate input parameter values.

3.5 ENVIRONMENTAL ACCUMULATION CODE: DESCARTES

In theory, DESCARTES needs to be run only once to generate the environmental media and food product concentrations database that may be accessed many times by the dose estimation code. However, practical considerations dictated that DESCARTES be made sufficiently flexible with regard to user input specification to facilitate testing and the potential need to regenerate selected portions of the environmental database.

DESCARTES can be run interactively on the VAX from any directory within the HEDR Project account by typing

DESCARTES

at the VMS \$ prompt. Sections 3.5.1 and 3.5.2 describe the options and values that must be supplied by the user. Section 3.5.3 explains how to run DESCARTES as a background (batch) process on the VAX.

3.5.1 Entering Program Control Options

Version 1.0 of DESCARTES prompts you to

Enter number of runs:

Enter the number of simulation runs you desire. This number should be less than or equal to the number of Monte Carlo simulation runs represented in the set of input parameter files and the atmospheric transport data files.

(NOTE: Because the current version of PARSEL is limited to generating parameters for one simulation run, you must enter 1 in response to this prompt. In readiness for a full parameter selection code that can generate multiple input parameter sets, DESCARTES has been programmed to accept and execute more than one simulation run.)

3.5.2 Selecting Output Options

For testing purposes, Version 1.0 of DESCARTES allows you to restrict the output to values at specific locations (grid cells), or you may choose to output concentration values for all grid cells that have a complete set of input data values. You will be asked:

Do you want output for all locations?

You must enter a string of one or more characters that begins with Y (yes) or N (no). Any other entry will cause the above prompt to reappear. If you enter Y, all grid cells that are marked for computation (i.e., for which a

complete input data set exists) will be selected for output and the program will proceed to execute with no further user interaction. If you enter N, the following prompt will appear:

Enter a location for which output is desired;

Press CTRL Z to end list:

Enter a grid cell number corresponding to the first location for which output is desired. The grid cells are numbered consecutively (beginning with 1) starting at the lower left (southwest) corner of the grid and proceeding from left to right (west to east) across the southern-most row of cells, then moving to the western-most cell of the next row of cells immediately above (north) of the previous row and repeating the process until all cells have been numbered. As an aid to locating the grid cell that coincides with a specific geographic location, the user should have access to a detailed paper map of the study area containing familiar geo-political features. The current grid, with the cell number printed inside each grid cell, should be imprinted on the map or attached to the map as a transparent overlay.

If the number entered is a valid grid cell number that has not already been entered and for which model input data exists, it will be added to the list of locations for output. Otherwise, a message will explain why the number you entered was not accepted. In either case, you will be prompted again to enter another location for output. You can enter any number of locations, one at a time, in this manner. When you have entered all desired grid cell numbers, press <CTRL Z>. The list of locations for output is then ended and the program executes with no further user interaction.

If you enter <CTRL Z> without selecting any locations, a message appears informing you that you have not selected any locations for output. You will then be prompted again to enter a location for which output is desired.

3.5.3 Running DESCARTES as a Background (Batch) Process

Because the run times for DESCARTES are typically quite long (on the order of hours), it is recommended that DESCARTES be submitted to run as a batch process. You can then terminate your interactive session and the

program will continue to run or will start at a specified time. To submit DESCARTES as a batch process, you must create a batch input file that contains the command to run DESCARTES, followed by lines containing the information you would have entered on the keyboard in response to prompts for user input if the program had been run interactively. Then submit the batch input file to one of the VAX batch queues using the VMS SUBMIT command at the \$ prompt:

SUBMIT /<options> filename

where "filename" is the name of the batch input file and "/<options>" denotes zero or more appropriate options to the SUBMIT command.

The following is an example of a DESCARTES batch input file:

```
$ RUN DESCARTES
1      ! Number of simulation runs
N      ! Do not select all locations for output
137    ! List of locations (cell numbers) for output
189
341
342    ! Last item in location list (end of file = CTRL Z).
```

3.6 INDIVIDUAL DOSE CODE: CIDER

CIDER calculates estimated dose to an individual as a result of exposure to and uptake of radionuclide concentrations in location-dependent and time-dependent environmental media and food product concentrations. These concentrations will have been stored in the database created by DESCARTES. To perform the calculations, DESCARTES requires certain information about the individual to estimate dose reception factors and to determine the sources and amounts of food products ingested and the locations and extent of the individual's exposure to environmental media. The required characterization of the individual includes birthdate, sex, source of daily products, and dates and location of residence within the study area during the time period of interest.

CIDER can be run interactively on the VAX from any directory within the HEDR Project account by typing

CIDER

at the VMS \$ prompt. Sections 3.6.1 and 3.6.2 describe the options and values that must be supplied by the user. Section 3.6.3 explains how to run CIDER as a background (batch) process on the VAX.

3.6.1 Entering Program Control Options

CIDER prompts for the following items of program control information:

Enter beginning date of dose period (mm/dd/yyyy):

Enter the date to begin calculating doses. The date must be entered in the indicated format: a two-digit month number, followed by a slash (/), followed by a two-digit day of the month, followed by another slash, followed by a four-digit year number. The date you enter must not be earlier than the first day represented in the input parameter files and the environmental database files.

Enter ending date of dose period (mm/dd/yyyy):

Enter a date, in the format shown, to indicate the last day for which you wish to calculate doses. The date you enter must not be later than the last day represented in the input parameter files and the environmental database files.

Enter number of runs:

Enter the number of simulation runs (Monte Carlo realizations) desired. This number must be less than or equal to the number of runs represented in the input parameter files and the environmental database files.

3.6.2 Entering Characteristics of Individuals

CIDER prompts you to enter information characterizing the individual for whom dose estimates are to be calculated. (The prompting messages address the person entering the information as if she or he were the individual of interest.) You are presented with the following prompts for information:

Enter the five-character id number assigned to you (or the individual):

Enter a unique identification number to identify the individual for whom the dose is being computed. This number will be incorporated into the dose estimate file names.

Enter the location where you lived:

Enter the grid cell number that best defines the location where the individual lived during the time period of interest (the beginning and ending dates entered previously).

(NOTE: The current version of CIDER is limited to only one principal residence location during the simulated time period. A future version will allow the entry of a sequence of residence locations and associated beginning and ending dates for each location.)

Enter your birthdate (mm/dd/yyyy):

Enter the individual's birthdate in the format shown.

Enter your sex (M/F):

Enter M for male or F for female. If your response is F, the following prompt will appear:

Were you a nursing mother?

Enter Y (yes) if the individual was a nursing mother at any time during the time period of interest; otherwise, enter N (no). If your response is Y, the following two prompts will appear:

Enter date when you began nursing your baby (mm/dd/yyyy):

Enter the beginning date in the format shown.

Enter date when you stopped nursing your baby (mm/dd/yyyy):

Enter the ending date in the format shown. (Estimate the ending date if not known precisely.)

Milk Source:

- 1 - Family cow*
- 2 - Local grocery stores*
- 3 - Creamery*

What was the principal source of milk that you drank (1 - 3)?

Enter 1, 2, or 3. If your response is 1, the following selection menu and prompt will appear:

Feed regimes:

- 1 - Summer pasture, fall alfalfa hay and silage, winter alfalfa hay, year-round grain supplement*
- 2 - Summer pasture, fall and winter alfalfa hay, year-round grain supplement*
- 3 - Year-round alfalfa hay with grain supplement*
- 4 - Year-round grass hay with grain supplement*

Which feed regime best describes how the cow was fed (1 - 4)?

Enter 1, 2, 3, or 4.

If your response to the milk source question was 3 (creamery), the following selection menu and prompt will appear:

Creameries:

1- Ritzville Dairy	2- Morning Sun, Moses Lake	3- Twin City, Tri-Cities
4- Fairview Guernsey, Prosser	5- Connell Dairy	6- Willow Point, Grand Coulee
7- Swanson's, Ephrata	8- Gregory, Ellensburg	9- Enfield, Ellensburg
10- Faust, Ellensburg	11- Kayner's, Roslyn	12- Del Ducco, Roslyn
13- Sawyer's, Cle Elum	14- Mayflower, Hermiston	15- Pendleton Dairy
16- Walla Walla Dairyman	17- Young's, Walla Walla	18- Shady Lawn, Walla Walla
19- Prescott Creamery	20- Duff's, Walla Walla	21- Selah Home Dairy
22- Cascade, Yakima	23- Yakima City Creamery	24- Darigold, Yakima
(+ Distr., Kennewick)
25- Carnation, Sunnyside
(+ Distr., Richland)	26- Local Producers from
Yakima County Census
Subdivision YA-9 (Sunnyside
Area)	

Which creamery did the milk come from (1 - 26)?

Enter a number between 1 and 26 to indicate the creamery that supplied the greatest amount of milk consumed by the individual during the time period in question.

3.6.3 Running CIDER as a Background (Batch) Process

To submit CIDER as a batch process, you must create a batch input file that contains the command to run CIDER, followed by lines that contain the information you would have entered on the keyboard in response to prompts for user input if the program had been run interactively. Then submit the batch input file to one of the VAX batch queues using the VMS **SUBMIT** command at the \$ prompt:

SUBMIT /<options> filename

where "filename" is the name of the batch input file and "/<options>" denotes zero or more appropriate options to the **SUBMIT** command.

The following is an example of a CIDER batch input file:

```
$ RUN CIDER
04/15/1945  ! Beginning date of time span of interest (mm/dd/yyyy)
12/31/1945  ! Ending date of time span of interest (mm/dd/yyyy)
1           ! Number of simulation runs
33129       ! 5-character id of individual
137         ! Principal residence location (cell no.) during time span
07/04/1930  ! Birthdate (mm/dd/yyyy)
F           ! Female
N           ! Not a nursing mother during time span
1           ! Milk source: family cow
3           ! Dairy cow feeding regime #3
```

3.7 POST-PROCESSOR AND REPORT GENERATOR PROGRAM: REPGEN

Version 1.0 of REPGEN is limited to reporting results for a single (best-estimate) model run. This implementation of REPGEN is sufficient to allow the reporting of best-estimate test results from the initial versions of the computational codes DESCARTES and CIDER.

REPGEN can be run interactively on the VAX from any directory within the HEDR Project account by typing

REPGEN

at the VMS \$ prompt. Sections 3.7.1 and 3.7.2 describe the interactive prompts and corresponding responses that the user must enter. Section 3.7.3 explains how to run REPGEN as a background (batch) process on the VAX.

3.7.1 Selecting Model Results for Reporting

REPGEN allows you to select the environmental and food product concentration variables and/or dose component variables to be included in the output report. You will first be asked to

Select the set of variables to be reported:

A - All variables (concentration and dose)

C - all Concentration variables

D - all Dose variables

E - Enter individual variables

If you select E, you will be shown a numbered menu of available variables:

CONCENTRATIONS:

1- Leafy Veg.	2- Other Veg.	3- Grain	4- Fruit
5- Alfalfa	6- Pasture grass	7- Grass/Upper Soil	8- Grass/Root Zone
9- Silage	10- Sagebrush	11- Grass Hay	12- Beef
13- Venison	14- Poultry	15- Eggs	16- Grocery Milk
17- Reg.1 Milk	18- Reg.2 Milk	19- Reg.3 Milk	20- Reg.4 Milk

DOSES:

21- Submersion	22- Groundshine	23- Non-Milk Ingest.	24- Milk Ingest.
25- Inhalation	26- Total		

You will then be asked to

Enter one or more numbers corresponding to the variables you wish to include in the report. Separate each number by a comma, by one or more spaces, or by a <RETURN>. End the list by entering <CTRL Z>.

Numbers that are outside the range of variable numbers in the menu will be ignored. If you enter <CTRL Z> without selecting any variables, the program will return to the first menu and prompt ("Select the set of variables to be reported"). When you have completed your selection, the variables you selected will be listed on the screen and you will be asked to confirm:

Is this the list of variables you wish to include in the report? (Y/N):

Enter Y to accept the list. Enter N to reject the list, and the program will re-prompt you to enter a new list of variables.

If you have selected concentration variables, you will be asked to:

Enter a location for reporting concentration results;

Press CTRL Z to end list:

Enter a grid cell number corresponding to the first location for which a report of the selected concentration variables is desired. If the program verifies that the number entered is a valid grid cell number that contains calculated model results and has not already been entered, the number will be added to the list of locations for reporting. Otherwise, a message will explain why the number you entered was not accepted. In either case you will be prompted again to enter another location for reporting. You can enter any number of locations, one at a time, in this manner. When you have entered all desired grid cell numbers, press <CTRL Z>. The list of reporting locations is then ended.

If you selected dose variables, you will be asked to

Enter the id number of the individual:

Enter the identification number of the individual for whom the dose report is desired. This should be the same identification number that was entered when the CIDER run was made for that individual.

You will next be asked to enter the beginning and ending dates of the time period to be covered by the report:

Enter the beginning date for reporting results (mm/dd/yyyy):

Enter the beginning date in the format shown. This date must not be earlier than the beginning date of data in the environmental concentration and/or dose estimate files selected.

Enter the ending date for reporting results (mm/dd/yyyy):

Enter the ending date in the format shown. This date must not be later than the ending date of data in the environmental concentration and dose estimate files selected.

3.7.2 Selecting Output Options

The current version of REPGEN allows you to select the level of temporal resolution at which selected variables will be reported. You will be asked to

Select one of the following levels of temporal resolution for reporting:

- D - Daily values with monthly and yearly aggregation*
- M - Monthly and yearly aggregated values*
- Q - Quarterly and yearly aggregated values*
- Y - Yearly aggregated values only*

Enter one of the above letters (D, M, Q, or Y). Any other entry will cause the above menu and prompt to be repeated. Aggregation of concentration variables is done by averaging. Aggregation of dose variables is done by summation (the code reports both the dose received within the time period and cumulative dose from the beginning of the simulation to the end of the time period).

Finally, you are prompted to

Enter the filename of the output report:

Enter a valid VAX/VMS filename for storing the report. The program will then proceed to execute without further user interaction. When the program completes, the report file can be printed or viewed on your terminal.

3.7.3 Running REPGEN as a Background (Batch) Process

To submit REPGEN as a batch process, you must create a batch input file that contains the command to run REPGEN, followed by lines containing the

information you would have entered on the keyboard in response to prompts for user input if the program had been run interactively. The following is an example of a REPGEN batch input file:

```
$ RUN REPGEN
E          ! Enter individual variables
1          !   Leafy veg.
6          !   Pasture grass
17         !   Milk from cows on Feed Regime 1
24         !   Milk ingestion dose
26         !   Total dose
\          ! CTRL Z - end of list of output variables
Y          ! Confirmation of correct list
137        ! First location in list of locations to be reported
341
342        ! Last item in location list
\          ! CTRL Z - end of location list
33129      ! 5-character id of individual for whom dose is reported
12/26/1944 ! Beginning date of reporting period
12/31/1945 ! Ending date of reporting period
M          ! Report monthly and yearly aggregated values
REPORT.OUT ! Name of output report file
```

3.8 PLANNED EXPANSION AND REFINEMENT

The following items are planned as future additions and improvements to the current (Version 1.0) computer codes. Some of these items (in particular, the full parameter selection program) will be completed during FY 1992. Others are dependent upon the availability of certain input data and will be scheduled contingent upon the availability of such data.

- The parameter selection program PARSEL will be fully implemented to include the capability to generate samples from parameter distributions, including possible joint correlated distributions. This capability will allow the generation and storage of multiple realizations of the input parameter set, for input to multiple Monte Carlo simulation runs.

- Grid-based data will be incorporated into the programs as they become available. PARSEL will be enhanced to enable the mapping and conversion onto the target grid cells of data that are based on other divisions of the spatial area (e.g., counties or townships) in addition to the Phase I census subdivisions currently being used.
- The capability to generate model input and output data on an arbitrarily spaced grid of arbitrary dimensions (up to some maximum number of cells) will be included. Current plans are to use a 41 x 51 grid, each cell of which is approximately 6 miles square, for the final model calculations (see Figure 1.2).
- The programs currently calculate concentrations and dose for only one radionuclide: iodine-131. The code and database will be expanded to include one or more additional radionuclides if screening calculations and other investigations determine that such is warranted.
- The commercial foods/feed submodel currently computes concentrations in commercially collected and distributed food products for only one food category: milk. The code implementation will be expanded to include commercial distribution of other human food products and animal feed.
- The computation of groundshine dose is currently based on the radionuclide concentration in the upper soil and root zone compartments of the pasture grass ecosystem. The environmental accumulation model and program will be expanded to include an additional ecosystem, residential lawns, for the purpose of computing a more realistic groundshine dose.
- The environmental accumulation model and program will be expanded to include an additional food/feed crop ecosystem: fall-planted winter grains.
- CIDER currently uses the default (reference) food consumption data for individuals of a specific age and sex in the determination of ingestion dose. The code will be modified to allow the input of individual historical diet information if that information is known or can be reasonably estimated.
- CIDER will be modified to allow the inclusion of dose exposure received at a work location other than the location of residence. It will also be expanded to perform dose calculations based on multiple residence and/or work locations during different periods of time (to accommodate an individual who may have moved from one location to another or changed job locations during the time of interest).

- CIDER will be expanded to include the ability to recalculate radionuclide concentration in milk using a customized milk cow feeding regime (feeding information supplied by an individual who drank milk from cows on a known non-standard feed regime). This expansion will entail the incorporation of a portion of the animal feeding submodel/code into CIDER.
- CIDER will be expanded to allow computation of collective doses to specific population groups.
- CIDER will be expanded to allow computation of pre-natal (fetal) doses and doses to breastfed infants.
- CIDER will be expanded to allow computation of effective dose equivalent (EDE) in addition to the current thyroid dose from iodine-131. The calculation of other organ-specific doses may be added as other radionuclides are included in the model.
- The report generator code REPGEN will be expanded to allow the computation and reporting of distributional statistics from the results of multiple Monte Carlo simulation runs.

4.0 PROGRAMMER'S GUIDE

This guide provides an overview of the programming aspects of the HEDRIC computer codes for estimating environmental concentrations and individual dose. It also provides detailed descriptions of the logical structure of the programs and their input and output files.

4.1 OVERVIEW

4.1.1 Program Organization and Relationships

Calculation of radionuclide transport from atmospheric deposition through environmental and food-chain pathways and the resulting individual doses to humans from the air pathway are implemented by two computer codes: DESCARTES, the code that implements the environmental accumulation model, and CIDER, the code that implements the individual dose model. Two other programs are required to fully implement the calculation process: PARSEL, an input parameter selection and generation program, and REPGEN, a post-processor and report generator program that selects and reports output information from either DESCARTES or CIDER. Figure 4.1 shows the flow of data among the four programs.

4.1.2 Language and Coding Standards

To enhance readability and portability, the HEDRIC codes were programmed in standard ANSI Fortran-77, with selected common extensions (e.g., INCLUDE and DO WHILE statements). Structured programming techniques were used insofar as possible within the limitations of Fortran-77 and the selected extensions. IMPLICIT NONE statements were included at the beginning of all program modules to force explicit declaration of all program variables. Multidimensional array accesses were optimized based on the VAX/VMS Fortran storage convention, in which the leftmost array dimension varies most frequently (i.e., consecutive increments in the leftmost array dimension map to consecutive internal memory locations). Comments are included throughout the programs to facilitate interpretation and increase readability.

FIGURE 4.1. Flow of Data Among HEDRIC Programs

Each of the program modules (main programs and subprograms) includes a documentation header section that follows a standard format and includes the following comment blocks:

- Definition block - includes information on the routine's source file name, purpose, limitations and prerequisites, argument definitions (if a subprogram), input and output files used, and subprograms called
- Implementation block - includes author name and other relevant development information, original implementation date, modification history, and algorithm descriptions (as appropriate).

The actual program statements are organized into two sections:

- Declaration block - includes parameter statements, constants, variables, arrays, common blocks, and data statements
- Execution block - includes all executable code statements.

4.1.3 Data Input and Output

This section describes in general terms the flow of data into, among, and out of the four main programs, as depicted in Figure 4.1.

ASCII (text) versions of the model input data files are listed in Appendix B, and the code listings are presented in Appendix C. Some of the larger input files were converted to unformatted binary files to facilitate rapid input by the model codes. For each unformatted binary input file, there exists a corresponding program that was used to create the binary file from the equivalent text file. All input files and all input data conversion programs are maintained in a HEDR Code Management System (CMS) Library on the VAX.

The input data files and the atmospheric transport model output files are read by PARSEL. The program first transposes and reformats the data from the atmospheric transport files into files that can be more efficiently input by the calculational codes. These files are named I-131_AIRyy.rrr, where yy is the last two digits of the year, and rrr is a three-digit simulation run number. PARSEL then outputs all input data and parameter values as unformatted binary parameter files. For each parameter value set (simulation run), PARSEL produces one file of time-invariant parameter values. This file is named PARAMS.rrr, where rrr is a three-digit simulation run number (including leading zeros). PARSEL also produces one or more yearly files (one for each year or partial year of the requested simulation time span) for each simulation run. Each yearly parameter file contains a complete set of time-varying model input parameter values for a single Monte Carlo simulation run for all or part of a single year. These files are named PARAMSyy.rrr, where yy is the last two digits of the year and rrr is a three-digit simulation run number. More specific information on the contents of each of the records in the parameter files can be found in the PARSEL code listing in Appendix C1.

DESCARTES reads the transposed atmospheric transport data files and the parameter files produced by PARSEL and generates a database of direct-access binary files that contain the calculated environmental media and food product concentration data. DESCARTES produces one output file for every combination of radionuclide, location, and year. These files are named I-131_xxxx.yy,

where I-131 is the five-character name of the radionuclide (iodine-131), xxxx is replaced by the location (grid cell) number, and yy is replaced by the last two digits of the year. These files store the concentration data in the form of compressed time series. A detailed description of the environmental database file structure is contained in Section 4.3.2.

CIDER accesses selected portions of the concentration files produced by DESCARTES, depending on the time period of interest and the residence location of the individual for whom dose exposure is being estimated. CIDER also reads age-specific dose reception factors, default food consumption data, and other required input parameters from the PARSEL parameter files. CIDER calculates estimates of dose to the individual from exposure by submersion, inhalation, groundshine, and ingestion. These dose estimates are written to direct-access binary files with a compressed time series structure identical to that of the concentration files produced by DESCARTES.

REPGEN, the post-processor and report generator program, can read both the environmental concentration files and the dose estimate files. REPGEN extracts model output data for the individuals, locations, time periods, and variables selected by the user and generates requested summarizations and tabular reports.

4.1.4 Computer System Requirements

One of the most demanding computer resource requirements of the DESCARTES and CIDER codes arises from the need to store the enormous amount of output data the code can produce. This large amount of information results from the fact that the set of concentration variables (currently 20) is calculated repeatedly over large space and time domains and is stored at comparatively high spatial resolution (6-mile-square grids) and temporal resolution (daily). This large number of space/time "cells" must be further multiplied by the number of Monte Carlo simulation runs that will eventually be needed to produce realistic measures of the distribution and uncertainty in the results. For a 3-year time span, a spatial grid of 31 x 31 cells, and 100 Monte Carlo realizations, a maximum of 8 gigabytes of storage may be required.

A second requirement is that the computer system be sufficiently fast (with respect to disk I/O access and throughput as well as processor execution speed) for the code runs to be completed in a "reasonable" period of time. The speed requirement is most critical for the individual dose program CIDER, because a quick turnaround (on the order of a few minutes) in calculating individual dose estimates is desired.

A final requirement is that the computer system provide adequate programming language and operating system support and a good software development environment, including useful program debugging and maintenance tools.

During FY 1991, HEDR staff studied computer systems and ran benchmark programs on existing computers at PNL to obtain basic timing/performance estimates for calculations similar to those that would be performed by the HEDRIC codes. Considering both performance and cost, it was concluded that a Sun 4/490 workstation/server configured with multiple large-capacity disks would be the best choice to meet the computer system requirements outlined above. HEDR staff procured the recommended system, which was delivered to PNL near the end of FY 1991.

The initial (Version 1.0) code development was done on the PNL VAX, however, because of several factors: 1) the HEDR Sun 4/490 arrived late; 2) existing code from the preliminary (Phase I) HEDR codes, some of which was directly transferable to the current codes, was resident on the VAX; 3) the VAX has a superior software development and debugging environment; 4) the code developers were more familiar with the VAX/VMS operating system environment and thus, the use of the VAX allowed the initial code development to proceed more efficiently; and 5) having complete codes on both the VAX and Sun facilitates verification of code implementation and conversion. The programs described in this report will be transferred to the HEDR Sun computer early in calendar year 1992.

4.1.5 System-Dependent Code Elements

Standard Fortran-77 coding elements, with common extensions that are available on most system compilers, were used with few exceptions. These exceptions occurred when there was a significant coding efficiency, an

4.2.2 Input and Output Files

PARSEL reads the input data files listed in Appendix B. PARSEL also reads the air concentration and deposition data from the atmospheric transport model output files. The current version of PARSEL reads monthly data files named JAN45.AIR, FEB45.AIR, ..., DEC45.AIR for the year 1945 only.

The current version of PARSEL creates three output files:

- I-131 AIR45.001, an unformatted binary file in which the atmospheric transport model data have been transposed so that air concentration and deposition data for all days of the year (1945) for a single location (grid cell) are contained in one record
- PARAMS.001, an unformatted binary file containing best-estimate values for all of the time-invariant input parameters
- PARAMS45.001, an unformatted binary file containing best-estimate values for all of the input parameters that vary with time (either daily, seasonally, or yearly). The first record in the file contains daily parameters that do not vary with location. This record is followed by repeating groups of three records, one group of records per location, that contain values of location-dependent parameters.

4.2.3 Code Logic and Structure

Pseudo-code was generated as a final design step prior to coding PARSEL. As changes were made in the logic and structure of the program during development and testing, corresponding revisions were made in the pseudo-code listing. The final pseudo-code listing, presented in Figure 4.2, reflects the actual implementation of Version 1.0 of PARSEL and is thus useful for describing and understanding the program. Figure 4.3 shows the subprogram calling structure of PARSEL. Appendix C1 contains a complete listing of PARSEL and its subprograms.

4.2.4 Algorithms and Special Routines

PARSEL's only special algorithm is a technique for taking input data items from the census subdivisions by which the data are currently stored and mapping them to the corresponding individual grid cells on the 31 x 31 spatial grid. The program first reads a file that identifies the census subdivision within which each cell in the 31 x 31 grid is contained or is associated by

```

Read air transport model results file, transpose, and write to output file.
Get user input:
  - Total NUMBER_OF_DAYS (beginning 12/26/44) for generation of parameters.
Set best-estimate values of parameters that were not set with DATA statements.
Read the node-to-census subdivision mapping file.
For each node:
  If node maps to a specific census subdivision:
    Set NODE_STATUS = 0 (normal status: node has complete input data)
  Else (node does not map to any HEDR Phase I census subdivision):
    Set NODE_STATUS = -2 (do not calculate: incomplete input data)
End Loop for each node
Build linked lists of nodes that correspond to each census subdivision.
Convert linked lists to sequential lists.
Read input data from input files:
  - Pasture season begin and end dates by census subdivision
  - Feed regime fractions by census subdivision
  - Milk source (census subdivisions) to creamery matrix
  - Milk creamery to grocery (census subdivisions) matrix
  - Seasonal food consumption data by age group
  - Alfalfa and grass hay cutting dates by census subdivision
For each census subdivision:
  For each node that maps to this subdivision (from sequential lists):
    Store subdivision-based input values in array positions for this node
  End Loop on nodes for this subdivision
End Loop on census subdivisions
Open the time-invariant output file (PARAMS.001)
Write Records A1-A7 (time-invariant parameters); Close the file.
For YEAR = 1 to NUM_YEARS:
  Copy daily food consumption values to output array from appropriate
    seasonal consumption array based on day of year.
  Create file name (PARAMSyy.001) using last two digits of current YEAR.
  Open yearly parameter file (PARAMSyy.001).
  Write Record B1 (location-invariant, time-varying parameters).
  For each location (NODE):
    If NODE_STATUS = 0 (active node with complete input data):
      For each plant-type:
        Determine and store harvest and tillage dates for this NODE and YEAR
      End Loop on plant-type
      For each day in current YEAR:
        For each plant-type:
          For each dairy feed regime:
            Determine and store dairy feed consumption from seasonal data
          End Loop on dairy feed regime
        End Loop on plant-type
      End Loop on day in year
      Write Records B2-B4 to file for this location
    End (If NODE_STATUS = 0)
  End Loop for each location
  Close yearly parameter file
End Loop on YEAR
End Program

```

FIGURE 4.2. PARSEL Pseudo-Code Listing

PARSEL Main Program	
GET_NODES_TO_CD	- Reads the node-to-census subdivision mapping file
TRANS_AIR	- Reads the atmospheric transport model files and writes the data in transposed form (daily within location) to a set of output files

FIGURE 4.3. PARSEL Subprogram Calling Structure

geographic proximity. Some of the grid cells lie outside the 10-county region within which the census subdivision-based data exist. PARSEL sets a flag (NODE_STATUS) for each such cell to indicate to the DESCARTES and CIDER programs that no calculations are to be performed for that cell.

Having created a one-to-one table of a census subdivision number (or "none") for each grid cell, PARSEL generates the inverse (one-to-many) mapping of zero or more cells for each census subdivision. This allows subdivision-based input data values to be directly deposited, as they are read, into the array elements that correspond to the cells associated with each subdivision. The algorithm that performs this mapping first creates a linked-list representation from the one-to-one table, then traverses this linked list to create a sequential list with first-element pointers for each census subdivision.

4.3 ENVIRONMENTAL ACCUMULATION CODE

4.3.1 Function

The environmental accumulation code, DESCARTES, performs the following functions:

- reads the transposed atmospheric transport files and model input parameter files that were generated by PARSEL
- uses the biomass growth equations to calculate biomass densities for each plant type as a function of location and time
- uses the programmed model equations together with the atmospheric transport data, the input parameter values, and the biomass densities to calculate concentrations of radionuclides (iodine-131) in soil, vegetation, and animal products for each relevant spatial location (grid cell) and time increment (day) during the time span of interest

- uses the commercial milk accumulation and distribution submodel to calculate concentration in commercial milk at each of the known major creameries (collection centers) and at grocery stores in each location
- writes the concentration values to a set of files (environmental concentrations database).

4.3.2 Input and Output Files

DESCARTES reads the transposed atmospheric transport data files and the model input parameter files and writes the calculated concentrations to a collection of direct-access binary files that form the environmental and food product concentrations database. This database provides the input to CIDER.

The following paragraphs describe the content and structure of the concentration files. The file design provides for compact (compressed) storage in the event of missing, non-applicable, and indirectly referenced data and also provides the ability to access data directly and quickly for specific variables and time periods. The grouping of data items in the new structure is closely aligned with the manner in which the data will be accessed by the dose calculation code.

Separate direct-access binary files are created for each relevant combination of radionuclide, location, and calendar year. The domain of each file can be identified by the file name. The two file types and a general description of their contents are listed below:

<code><nuc>_xxxx.yy</code>	- concentration data for nuclide <code><nuc></code> at location <code>xxxx</code> in year <code>19yy</code>
<code><nuc>_DAIRYzz.yy</code>	- commercial milk concentration data for nuclide <code><nuc></code> at creamery <code>zz</code> during year <code>19yy</code>

Within each file, values of each variable (radionuclide concentration in a specific environmental component or compartment) are potentially stored for each day of the year for each Monte Carlo simulation run. Sequences of data values that are missing or are not applicable for a particular time period within the year are not stored but are flagged as missing or not applicable in the file's header section. For some periods the values of a variable from a

different location (file) are to be used as the values for the current location. For these periods, no values are stored for the current location. Instead, the data values from the referenced location (file) will be retrieved by the subroutine that reads the environmental concentration data.

The detailed file organization, content, and record structure is as follows: The data values are organized by variable (environmental component), by sequential day within the year, and by Monte Carlo run number. The logical record length of each file is 32 longwords (128 bytes). Each file has an approximate maximum size of the number of variables by the number of Monte Carlo runs by 371 days per year (1945 includes the last six days of 1944) by 4 bytes per data value. For example, 20 variables and 100 Monte Carlo runs would produce a maximum file size of approximately 3.0 megabytes.

A "time period" is defined as a series of consecutive days within the year during which a particular variable's "access code" is constant. The access code of a variable for a specific time period, as stored in the "access codes" header records, is defined as follows:

- 2 ==> data values are missing (unknown, not computed)
- 1 ==> data values are less than a minimum threshold (currently 1.0×10^{-30}) or irrelevant (not applicable)
- 0 ==> use the data stored in this file for this time period
- $n > 0$ ==> use the data from location n for this time period

The first few records in a file are header records that contain information about the values stored in the data records. The header records are organized as follows:

First record in file: General file header record (all fields Integer*4)

<u> No. of </u> <u>variables </u>	<u>No. of MonteCarlo</u> <u>repetitions </u>	<u>No. of direct access</u> <u>records in file </u>
$n = \text{NVAR}$	NREP	NRECS

First Monte Carlo run:

"Access codes" header records: 1 record per variable, 2 fields per period,
all fields Integer*4

```
-- NAC1 = Record no. of first "Access codes" header record for this run

      Direct Access      Direct Access
      Record No. of      Record No. of
      1st data record    last data record
      for this var.      for this var.
      -----
-> 1st var.:  NF1         NL1         Begin day/code . . . Begin day/code
   2nd var.:  NF2         NL2         Begin day/code . . . Begin day/code
      :
      :
last var.(n): NFn         NLn         Begin day/code . . . Begin day/code
```

NOTE: A "begin day" of zero indicates there are no more time periods for this variable; the remainder of the record is unused. The "end day" of a period is the day before the begin day of the next period for the same variable, or the last day of the simulation for the current year if there are no more valid time periods.

Data records:

NF1 = Record no. of first data record for first variable, determined by

$$NF1 = NAC1 + NVAR$$

Real*4 values in the data records (32 Real*4 values per record) are stored in the following order:

```
[ First var.: First "code 0" (local data) period: First day of period ]
[ First var.: First "code 0" period:      Second day of period ]
      :
      :
[ First var.: First "code 0" period:      Last day of period ]
-----
[ First var.: Second "code 0" period: First day of period ]
[ First var.: Second "code 0" period: Second day of period ]
      :
      :
[ First var.: Second "code 0" period: Last day of period ]
-----
      :
      :
```

```

-----
[ First var.: Last "code 0" period: First day of period ]
[ First var.: Last "code 0" period: Second day of period ]
.
.
[ First var.: Last "code 0" period: Last day of period ]
=====
(***) Above sequence is repeated for each variable in turn. (***)
(***) A new record is started at beginning of data for each variable. (***)

Second Monte Carlo run: (start a new logical record)

(***) Repeat the sequence of the first MonteCarlo run:
    "Access codes" header records for second MonteCarlo run,
    followed by data records for second MonteCarlo run. (***)
.
.
.
Last Monte Carlo run: (start a new logical record)

(***) Repeat the sequence of the other MonteCarlo runs. (***)

```

The software routine that opens a database file and retrieves data from the file will use the information in the header records to compute the direct-access record numbers of the records containing the requested data values.

4.3.3 Code Logic and Structure

Pseudo-code was generated as a final design step prior to coding DESCARTES. As changes were made in the logic and structure of the code during development and testing, corresponding revisions were made in the pseudo-code listing. The final pseudo-code listing, presented in Figure 4.4, reflects the actual implementation of DESCARTES and is thus useful for describing and understanding the code. Figure 4.5 shows the subprogram calling structure of DESCARTES. Appendix C2 contains a complete listing of DESCARTES and its subprograms.

4.3.4 Algorithms and Special Routines

The dairy cow feeding submodel and the commercial milk accumulation and distribution submodel are implemented using in-line code to perform simple matrix multiplication.

```

Get user inputs:
- RUNMAX (number of MonteCarlo repetitions)
- List of LOCATIONS for which output is desired.
  (NOTE: nuclide list and first/last year are determined by param. files)
For RUN = 1 to RUNMAX: (for each MonteCarlo repetition)
  Open time-invariant input parameter file (PARAMS.rrr) for current RUN (rrr);
  Read time-invariant input parameters into internal variables/arrays; Close
  file
  For NUCLIDE = 1 to NUCMAX:
    For YEAR = FIRST_YEAR to LAST_YEAR:
      If NUCLIDE > 1 then
        * Open existing Biomass file (BIOMASSyy.rrr) for this RUN (rrr) and YEAR
          (yy)
      Else If NUCLIDE = 1 and NUCMAX > 1 then
        * Open new Biomass file for output for this RUN and YEAR
      Open transposed air model file (<nuc>_AIRyy.rrr) for this NUCLIDE, RUN,
      YEAR
      Open input param. file (PARAMSyy.rrr) for this RUN and YEAR
      Read location-invariant input params. from input param. file (Record B1)
      Set conc. in commercial milk at all creameries for all days in year to zero
      For LOCATION = 1 to LOCMAX:
        ---If NODE_STATUS = 0 (data available) for this LOCATION, then do the following:
          * If this location is in output list, then open output file (<nuc>_llll.yy).
            (If RUN=1, then open as 'NEW' and write 1st header record; else, open as
            OLD)
          * Read location-varying input params from input param. file (Records B2,B3,B4)
            for this LOCATION for all PLANT_TYPES, ANIMAL_PRODUCTS, NUCLIDES, and
            NUM_DAYS_IN_YEAR into appropriate arrays.
          * If NUCLIDE > 1, then read Biomass and Growth-rate for this LOCATION for
            all PLANT_TYPES and DAYS_WITHIN_YEAR from Biomass file into arrays:
              BIOMASS(1:PLANT_MAX, 1:NUM_DAYS_IN_YEAR),
              GROWTH(1:PLANT_MAX, 1:NUM_DAYS_IN_YEAR)
          * Read air conc. and deposition for this LOCATION and NUCLIDE for all
            DAYS_WITHIN_YEAR from transposed atmospheric transport files into arrays:
              AIR_CONC(1:NUM_DAYS_IN_YEAR), SURFACE_DEP(1:NUM_DAYS_IN_YEAR)
          * If YEAR is FIRST_YEAR, then set initial conditions of time-integrated
            concentration variables to zero for all PLANT_TYPES for this LOCATION;
            Else, Set initial conditions of time-integrated variables to values saved
            from last day of previous year for all PLANT_TYPES for this LOCATION.
          * For DAY_WITHIN_YEAR = 1 to NUM_DAYS_IN_YEAR:
            For PLANT_TYPE = 1 to PLANT_MAX:
              - If NUCLIDE = 1 then
                * If DAY_WITHIN_YEAR is a harvest date for PLANT_TYPE, then
                  Reset BIOMASS to BMIN and adjust plant conc. proportionally
                * Calc. Biomass Growth-rate for current PLANT_TYPE and
                  DAY_WITHIN_YEAR
                * Use EULER formula to integrate Biomass to current DAY_WITHIN_YEAR
                * Store current Growth-rate and Biomass in appropriate array
                  location for current PLANT_TYPE and DAY_WITHIN_YEAR.
              End [ If NUCLIDE = 1 ]
            - Calculate derivatives of time-integrated variables (upper soil,
              root-zone soil, outer vegetation, inner vegetation).

```

FIGURE 4.4. DESCARTES Pseudo-Code Listing (page 1 of 2)

```

- Use EULER formula to integrate variables to current DAY_WITHIN_YEAR
- Store values of integrated variables by PLANT_TYPE and
  DAY_WITHIN_YEAR
End Loop on PLANT_TYPE
For ANIMAL_PRODUCT=1 to ANIMAL_MAX (excluding milk and commercial
  foods):
- Calculate concentration in current ANIMAL_PRODUCT.
- Store concentration values by ANIMAL_PRODUCT and DAY_WITHIN_YEAR
End Loop on ANIMAL_PRODUCT
For DAIRY_FEED_REGIME = 1 to MAX_REGIMES:
- Calc. conc. in milk from local cows fed on current DAIRY_FEED_REGIME
- Store milk conc. values by DAIRY_FEED_REGIME and DAY_WITHIN_YEAR.
End loop on DAIRY_FEED_REGIME
Calc. MIXED_CONC(1,d) -- conc. in mixed-regime commercial milk sent to
  creameries from this LOCATION (1) on current DAY_WITHIN_YEAR (d).
Distribute (add) mixed commercial milk to each creamery (distribution
  center) that receives milk from this LOCATION (1):
COMM_MILK_CONC(j,d) = COMM_MILK_CONC(j,d) + SOURCE_FRAC(j,1)*MIXED_CONC(1,d)
where COMM_MILK_CONC(j,d) = conc. in commercial milk at creamery j on day d;
SOURCE_FRAC(j,1)= fraction of milk at creamery j that is from location 1;
MIXED_CONC(1,d)= conc. in commercial milk produced at location 1 on day d
End Loop on DAY_WITHIN_YEAR
* Save latest values (at last day of current YEAR) of all integrated
  variables by PLANT TYPE and LOCATION as initial conditions for next year.
* If this LOCATION is in output list, then form "access codes" header records and
  time period data records from all output arrays for current RUN, and write header
  and data records to output file for current LOCATION, NUCLIDE, and YEAR; Close
  output file.
* If NUCLIDE = 1 and NUCMAX > 1, then write Biomass and Growth-rate arrays for this
  LOCATION to Biomass file.
----END [ If NOOE_STATUS = 0 ]
End Loop on LOCATION
Close transposed air model file; Close yearly input param. file.
If NUCLIDE = 1 and NUCMAX > 1, then close Biomass file.
For CREAMERY = 1 to NUM_CREAMERIES:
Open output file for this CREAMERY, YEAR, and NUCLIDE.
  (If RUN = 1, then open as 'NEW', else open as 'OLD')
Write CREAMERY milk conc. for all DAYS_IN_YEAR for this RUN; Close file.
End Loop on CREAMERY
For LOCATION = 1 to LOC_MAX:
For DAY_WITHIN_YEAR = 1 to NUM_DAYS_IN_YEAR:
  For CREAMERY = 1 to NUM_CREAMERIES
    * Redistribute commercial milk to grocery stores at this LOCATION
      from this CREAMERY for this DAY.
    * Store conc. in grocery milk in output arrays
  End Loop on CREAMERY
End Loop on DAY_WITHIN_YEAR
Append grocery milk arrays for this RUN to output file for current
  LOCATION, NUCLIDE, and YEAR
End Loop on LOCATION
End Loop on YEAR
End Loop on NUCLIDE
Delete Biomass files for this RUN
End Loop on RUN
Close all files; End Program

```

FIGURE 4.4. DESCARTES Pseudo-Code Listing (page 2 of 2)

DESCARTES Main Program

READ_USER_INPUT	-	Gets number of runs and output locations from user
READ_TIME_INV	-	Reads time-invariant parameter values into common block variables
READ_LOC_INV	-	Reads time-dependent, location-invariant parameter values into common block variables
READ_LOC_VAR	-	Reads time-dependent, location-dependent parameter values into common block variables
READ_AIR_MODEL	-	Reads atmospheric transport model results into common block variables
BIOMASS_GROWTH_RATE	-	Calculates biomass and growth rate of a given plant type on a given day
CALC_PLANT_CONC	-	Calculates concentration in 2 soil and 2 plant compartments for a given plant type on a given day
PLANT_DERIV	-	Calculates time derivatives of soil and plant concentrations for a given plant type and day
PLANT_INTEG	-	Updates (integrates) soil and plant concentrations for a given plant type and day
ANIMAL_PRODUCTS_CONC	-	Calculates concentrations in animal products (excluding milk)
MILK_CONC	-	Calculates concentrations in milk for each of four feed regimes
WRITE_CONC_TO_FILE	-	Writes all local concentrations in soil, vegetation, animal products, and milk to a file for current nuclide, location, and year
WRITE_CREAMERY	-	Writes concentrations of commercial milk at all creameries to a file
WRITE_MILK	-	Writes concentrations of milk in grocery stores to output file for current nuclide, location, and year

FIGURE 4.5. DESCARTES Subprogram Calling Structure

DESCARTES contains a special routine to write the environmental and food product concentration values in a compressed time series format to unformatted binary files. (See Section 4.3.2 for a description of the file structure.)

4.4 INDIVIDUAL DOSE CODE (CIDER)

4.4.1 Function

CIDER performs the following functions:

- prompts the user to enter information characterizing the individual for whom doses are to be estimated
- reads selected data from the environmental concentration files created by DESCARTES
- reads selected values from the input parameter files created by PARSEL

- uses the input parameter values, the information entered about the individual, and the appropriate environmental concentration data to calculate doses to the individual by air submersion, by inhalation, by groundshine, by milk ingestion, and by ingestion of all other foods
- writes the calculated doses to compressed binary output files.

4.4.2 Input and Output Files

Based on the individual's location and time period of interest, CIDER selects and reads the appropriate environmental concentration files. These files are named I-131_xxxx.yy, where I-131 is the radionuclide, xxxx is the location (grid cell) number, and yy is the last two digits of the year. CIDER also reads the input parameter files PARAMS.001 and PARAMSyy.001.

CIDER creates output files to store the calculated daily dose components. These files are named I-131_ccccc.yy, where ccccc is a unique five-character identification number assigned to the individual for whom doses are being calculated, and yy is the last two digits of the year. The structure of these files is identical to the structure of the environmental accumulation files as described in Section 4.3.2.

4.4.3 Code Logic and Structure

A pseudo-code listing was generated as a final design step prior to coding CIDER. As changes were made in the logic and structure of the code during development and testing, corresponding revisions were made in the pseudo-code listing. The final pseudo-code listing, presented as Figure 4.6, reflects the actual implementation of CIDER and is thus useful for describing and understanding the code. Figure 4.7 shows the subprogram calling structure of CIDER. Appendix C3 contains a complete listing of CIDER and its subprograms.

4.4.4 Algorithms and Special Routines

For each day of the simulation, CIDER uses the individual's entered birthdate and the current date to determine the individual's age as of his/her most recent birthday. This age is then used to perform a simple linear interpolation of dose factors between representative ages of the age groups upon which the dose factors are based.

```

Get user inputs:
- FIRST_DATE (beginning date of dose calculation period)
- LAST_DATE (ending date of dose calculation period)
- RUNMAX (number of MonteCarlo repetitions)
- INDIVIDUAL_ID (5-character i.d. assigned to individual)
- RES_LOC (individual's residence location during time period specified)
- BIRTHDATE (individual's date of birth)
- SEX (individual's sex (M/F))
  [ If SEX=F: ]
  - NURSING_MOTHER (TRUE if individual was a nursing mother)
    [ If NURSING_MOTHER: ]
    - BEGIN_NURSING (begin date of nursing period)
    - END_NURSING (end date of nursing period)
  - MAIN_MILK_SOURCE (family cow, local grocery stores, or creamery)
    [ If FAMILY_COW: ]
    - FEED_REGIME_NUMBER (1-4)
    [ IF CREAMERY: ]
    - CREAMERY_NUMBER (1-26)
Determine year and Julian day-of-year for entered dates:
- FIRST_YEAR,      BEG_DAY (beginning of dose calculation period)
- END_YEAR,        END_DAY (end of dose calculation period)
- NURSING_START_YEAR, NURSING_START_DAY (beginning of nursing period)
- NURSING_END_YEAR, NURSING_END_DAY (end of nursing period)
Determine AGE = individual's age on FIRST_DATE of dose period
If NURSING: Determine if active nursing mother on FIRST_DATE of dose period
Determine individual's age/sex/lifestyle (ASL) group on FIRST_DATE
Determine (interpolate) FIRST_DATE dose factors based on ASL group and age
For RUN = 1 to RUNMAX:
  Read time-invariant parameter values from param. file for current RUN
  For YEAR = FIRST_YEAR to END_YEAR:
    Determine BIRTH_DY = day-of-year of individual's birthday for current YEAR
    Read time-varying parameter values from param. file for current RUN and YEAR
    for location RES_LOC
    Read in air concentrations at location RES_LOC for current RUN and YEAR
    Read environmental concentrations at loc. RES_LOC for current RUN and YEAR
    If YEAR=FIRST_YEAR then FIRST_DAY=BEG_DAY, else FIRST_DAY=1.
    If YEAR=LAST_YEAR then LAST_DAY=END_DAY, else LAST_DAY=DAYS_IN_YEAR(YEAR)
    For DAY = FIRST_DAY to LAST_DAY:
      If DAY=BIRTH_DAY or (DAY=NURSING_START_DAY and YEAR=NURSING_START_YEAR)
        or (DAY=NURSING_END_DAY and YEAR=NURSING_END_YEAR) then
        Re-calculate individual's age/sex/lifestyle (ASL) group
        Re-calculate (interpolate) ASL-specific dose factors
      End IF
      Calculate submersion dose for this RUN and DAY
      Calculate groundshine dose for this RUN and DAY
      Calculate inhalation dose for this RUN and DAY
      Calculate non-milk ingestion dose for this RUN and DAY
      Calculate milk ingestion dose for this RUN and DAY
      Write daily dose variables to output file
    End Loop on DAY
  End Loop on YEAR
End Loop on RUN
Close all files; End program

```

FIGURE 4.6. CIDER Pseudo-Code Listing

CIDER Main Program

VALID_DATE	-	Determines if user has entered a valid birthdate or other date in mm/dd/yyyy format
DETERMINE_DOY	-	Calculates Julian day of year from day/month/year
DETERMINE_ASL	-	Determines individual's Age/Sex/Lifestyle (ASL) group from information entered by user
INTERPOLATE_DOSE	-	Calculates (interpolates) dose factors based on individual's current age relative to ASL groups
READ_AIR_MODEL	-	Reads atmospheric transport model results into common block variables
READ_TIME_INV	-	Reads time-invariant parameter values into common block variables
READ_LOC_INV	-	Reads time-dependent, location-invariant parameter values into common block variables
READ_LOC_VAR	-	Reads time-dependent, location-dependent parameter values into common block variables
READ_ENVIR_OUTPUT	-	Reads concentrations from environmental database files into common block variables
WRITE_DOSE_TO_FILE	-	Writes daily cumulative doses to output file

FIGURE 4.7. CIDER Subprogram Calling Structure

CIDER contains a special routine to extract selected concentration information from the unformatted binary compressed time series files that make up the environmental concentrations database. (See Section 4.3.2 for a description of the compressed time series file structure.)

4.5 POST-PROCESSOR AND REPORT GENERATOR PROGRAM (REPGEN)

4.5.1 Function

REPGEN performs the following functions:

- prompts the user to specify the model output data (variables, individual identification number, locations, and time span of interest) to be accessed
- prompts the user to specify the regular time periods (monthly, quarterly, yearly) over which the results are to be aggregated and reported
- reads requested information from the appropriate compressed binary dose estimate files and/or environmental concentration files
- calculates and outputs a report containing the requested data items and summarizations.

4.5.2 Input and Output Files

Depending on the information requested by the user, the following two types of files may be read as input by REPGEN: the environmental media and food product concentrations database files that are generated by DESCARTES (see Section 4.3.2), and the dose estimate files created by CIDER (see Section 4.4.2). REPGEN produces output reports in the form of text files that can be printed, viewed on the screen, and saved for future use. Appendix D contains portions of an example REPGEN report.

4.5.3 Code Logic and Structure

A pseudo-code listing was generated as a final design step prior to coding REPGEN. As changes were made in the logic and structure of the code during development and testing, corresponding revisions were made in the pseudo-code listing. The final pseudo-code listing, presented in Figure 4.8, reflects the actual implementation of REPGEN and is thus useful for describing and understanding the code. Figure 4.9 shows the subprogram calling structure of REPGEN. Appendix C4 contains a complete listing of REPGEN and its subprograms.

4.5.4 Algorithms and Special Routines

REPGEN uses standard algorithms for calculating average concentrations and cumulative doses over monthly, quarterly, and yearly time periods. It uses a special routine for reading and interpreting the dose and concentration information contained in the compressed binary files generated by DESCARTES and CIDER. (See Section 4.3.2 for a description of the compressed binary file structure.)

```

Get user inputs:
- Set of variables to be reported (keep concentration variables and dose
  variables in separate lists)
- Beginning and ending dates for reporting results
- Levels of temporal aggregation for reporting results
  (Daily, monthly, quarterly, yearly)
- File name of output report
If any concentration variables were selected:
- Prompt user to enter one or more locations for reporting concentrations
If any dose variables were selected:
- Prompt user to enter identification number of individual
Determine FIRST_YEAR and LAST_YEAR included in requested reporting time span
Report concentration data:
  Read concentration variable descriptions from variable definition file
  Generate and write column headings for each concentration variable
  For NODE (location) = FIRST_NODE_IN_LIST to LAST_NODE_IN_LIST:
    For YEAR = FIRST_YEAR to LAST_YEAR:
      Open environmental database file for this NODE and YEAR
      Read data for selected concentration variables
      Aggregate data to temporal reporting periods requested by user
      Write output lines for each temporal reporting period in current YEAR
      Close environmental database file
    End of Loop on YEAR
  End of Loop on NODE
End of reporting concentration data
Report dose data:
  Read dose variable descriptions from variable definition file
  Generate and write column headings for each dose variable
  For YEAR = FIRST_YEAR to LAST_YEAR:
    Open dose estimation file for the selected individual (id number) and YEAR
    Read data for selected dose variables
    Aggregate data to temporal reporting periods requested by user
    Write output lines for each temporal reporting period in current YEAR
    Close dose estimation file
  End of Loop on YEAR
End of reporting dose data
End Program

```

FIGURE 4.8. REPGEN Pseudo-Code Listing

HAPPOST Model Program

VALID_DATE	-	Determines if user has entered a valid birthdate or other date in mm/dd/yyyy format
DETERMINE_DOY	-	Calculates Julian day of year from day/month/year
SELECT_MODEL_OUTPUT	-	Reads concentration data from environmental database files or dose data from dose estimation files
TEMPORAL_AGGREGATE	-	Aggregates daily dose or concentration data over time periods specified (monthly, quarterly, or yearly)

FIGURE 4.9. REPGEN Subprogram Calling Structure

5.0 QUALITY ASSURANCE

5.1 DATA QUALITY OBJECTIVES

As part of its Quality Assurance Plan, the HEDR Project establishes data quality objectives for each task and subtask within the project. The data quality objectives established for the development of the atmospheric pathways programs are as follows (Shipler 1991a):

- The final computer codes must conform to the code design specifications, as documented in Napier et al. (1992), with respect to accuracy, completeness, and representativeness. Peer reviews, test cases, Battelle software control procedures and associated records, and internal quality assurance reviews are to be used to measure the degree to which this objective is met.
- The final computer codes must provide results that are within a factor of two of available validation data. Direct comparison of results, validation studies, and peer reviews are to be used to verify that this objective is met.

As of the date of this report, the following steps have been taken to assure the achievement of data quality objectives for the computer codes that are documented in this report:

- An internal technical review of this report and the codes described herein was performed (see Section 5.2).
- Limited testing of the codes was performed using best-estimate parameter values for a single model run.
- Battelle software control procedures have been applied to the initial version of the codes.

5.2 CODE VERIFICATION PROCEDURES

An independent technical review of this report and the initial version of the codes described herein was performed to ensure that, with the exception of the differences noted in this report, the codes accurately implement the model and code design specifications. Members of the review committee were selected based on their experience in dosimetric modeling, familiarity with

the Fortran-77 language, and experience in structured software design and development techniques. The results of the review are documented in HEDR project records.

5.3 OTHER QUALITY ASSURANCE PROCEDURES

To ensure and document the quality of software developed within the HEDR Project, a set of standard software control procedures as defined in the Battelle Pacific Northwest Laboratories Quality Assurance Manual (PNL-MA-70) has been adopted and are incorporated within the HEDR Quality Assurance Plan. Included are procedures for performing an internal development review of software and documentation; software configuration management (version and update control); conversion testing; software application control; and transfer of software, data, and/or documentation into or out of the project. These procedures are normally initiated upon completion of the initial version of an internally developed software system or related set of computer programs. They have been initiated and are being applied to the HEDRIC codes.

6.0 REFERENCES

- Anspaugh, L. R. 1987. Retention by Vegetation of Radionuclides Deposited in Rainfall-- A Literature Summary. UCRL-53810, Lawrence Livermore National Laboratory, Livermore, California.
- Baes, C. F., and R. D. Sharp. 1981. Predicting Radionuclide Leaching From Root Zone Soil For Assessment Applications. CONF-801606-44, Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Beck, D. M., R. F. Darwin, A. R. Erickson, and R. L. Eckert. 1992. Milk Cow Feed Intake and Milk Production and Distribution Estimates for Phase I. PNL-7227 HEDR, Pacific Northwest Laboratory, Richland, Washington.
- Black, S. C., and D. S. Barth. 1976. Radioiodine Prediction Model for Nuclear Tests. EPA-600-/4-76-0276. United States Environmental Protection Agency, Las Vegas, Nevada.
- Booth, R. S., and S. V. Kaye. 1971. A Preliminary Systems Analysis Model of Radioactivity Transfer to Man from Deposition in a Terrestrial Environment. ORNL-TM-3135, Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Chadwick, R. C., and A. C. Chamberlain. 1970. "Field Loss of Radionuclides from Grass." Atmospheric Environment 4(1):51-56.
- Chamberlain, A. C. 1970. "Interception and Retention of Radioactive Aerosols by Vegetation." Atmospheric Environment 4(2):57-78.
- Comar, C. L. 1966. "Radioactive Materials in Animals--Entry and Metabolism." Russell, R. S., editor. Radioactivity and Human Diet Pergamon Press, Oxford.
- Darwin, R. F., Jr. 1990. Soil by Dairy Cattle. PNL-SA-17918 HEDR. Pacific Northwest Laboratory, Richland, Washington.
- Ensminger, M. E., J. E. Oldfield, and W. W. Heineman. 1990. Feeds and Nutrition. Ensminger Publishing Co., Cloves, California.
- Hungate, F. P., J. F. Cline, R. L. Uhler, and A. A. Selders. 1963. "Foliar Sorption of I^{131} by Plants." Health Physics 9(12):1159-1166.
- IAEA. 1982. Generic Models and Parameters for Assessing the Environmental Transfer of Radionuclides from Routine Releases: Exposures of Critical Groups. Safety Series No. 57, International Atomic Energy Agency, Vienna, Austria.
- ICRP. 1977. Recommendations of the International Commission on Radiological Protection. ICRP Publication 26, International Commission on Radiological Protection. Pergamon Press, Oxford.

ICRP. 1979. Radionuclide Release into the Environment: Assessment of Doses to Man. ICRP Publication 29, International Commission on Radiological Protection. Pergamon Press, Oxford.

ICRP. 1990. Age-dependent Doses to Members of the Public from Intake of Radionuclides: Part 1. ICRP Publication 56, International Commission on Radiological Protection. Pergamon Press, Oxford.

Koch, J., and J. Tadmor. 1986. "RADFOOD - A Dynamic Model for Radioactivity Transfer Through the Human Food Chain." Health Physics 50(6):721-737.

Linsley, G. S., J. R. Simmonds, and S. M. Haywood. 1982. FOOD-MARC: The Food Chain Transfer Module in the Methodology for Assessing the Radiological Consequences of Accidental Releases. NRPB-M-76, National Radiological Protection Board, United Kingdom.

Napier, B. A. 1991a. Computational Model Design Specification for Phase I of the Hanford Environmental Dose Reconstruction Project. PNL-7274 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Napier, B. A. 1991b. Selection of Dominant Radionuclides for Phase I of the Hanford Environmental Dose Reconstruction Project. PNL-7231 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Napier, B. A., R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell. 1988. GENII - The Hanford Environmental Radiation Dosimetry Software System. PNL-6584 Vol. 1-3, Pacific Northwest Laboratory, Richland, Washington.

Napier, B. A., J. C. Simpson, T. A. Ikenberry, R. A. Burnett, and T. B. Miley. 1992. Final Design Specification for the Environmental Pathways and Dose Model. PNL-7844 HEDR, Pacific Northwest Laboratory, Richland, Washington.

NCRP. 1984. Radiological Assessment: Predicting the Transport, Bioaccumulation, and Uptake by Man of Radionuclides Released to the Environment. NCRP Report No. 76, National Council on Radiation Protection and Measurements, Bethesda, Maryland.

Pacific Northwest Laboratory. 1991a. Summary Report: Phase I of the Hanford Environmental Dose Reconstruction Project. PNL-7410 HEDR Rev. 1, Richland, Washington.

Pacific Northwest Laboratory. 1991b. Columbia River Pathway Report: Phase I of the Environmental Dose Reconstruction Project. PNL-7411 HEDR Rev. 1, Richland, Washington.

Pacific Northwest Laboratory. 1991c. Air Pathway Report: Phase I of the Hanford Environmental Dose Reconstruction Project. PNL-7412 HEDR, Rev. 1, Richland, Washington.

Peterson, H. T. Jr. 1983. "Terrestrial and Aquatic Food Chain Pathways." J. E. Till, and H. R. Meyer, eds. Radiological Assessment: A Textbook on Environmental Dose Analysis. NUREG/CR-3332, U.S. Nuclear Regulatory Commission, Washington, D.C.

Pinder, J. E. III, T. G. Ciravolo, and J. W. Bowling. 1988. "The Interrelationships Among Plant Biomass, Plant Surface Area and the Interception of Particulate Deposition by Grasses." Health Physics 55(1):51-58.

Pinder, J. E. III, K. W. McLeod, and D. C. Adriano. 1989. "The Accuracy of Some Simple Models for Predicting Particulate Interception and Retention in Agriculture Systems." Health Physics 56(4):441-450.

Pleasant, J. C., L. M. McDowell-Boyer, and G. G. Killough. 1980. RAGTIME: A Fortran IV Implementation of a Time-Dependent Model for Radionuclides in Agricultural Systems. NUREG/CR-1196, U.S. Nuclear Regulatory Commission, Washington, D.C.

Proehl, G., H. Muller, P. Jacob, and H. G. Paretzke. 1988. The Dynamic Radioecological Model ECOSYS - A Tool for the Management of Nuclear Accidents' Consequences. In Proc. 4th Symp. Int. Mtg. Impact of Accidents of Nuclear Origin on the Environment, Cadarache, France, March, 14-18, p. B43, Commissariat a l'Energie Atomique.

Sehmel, G. A. 1984. "Deposition and Resuspension." Atmospheric Science and Power Production, D. Randerson, ed. DOE/TIC-27601. U.S. Department of Energy, Washington, D.C.

Shipler, D. B. 1991a. FY 1992 Task Plans for the Hanford Environmental Dose Reconstruction Project. PNL-7757 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Shipler, D. B. 1991b. FY 1993 Task Plans for the Hanford Environmental Dose Reconstruction Project. PNL-7760 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Simmonds, J. R. and G. S. Linsley. 1981. "A Dynamic Modeling System for the Transfer of Radioactivity in Terrestrial Food Chains." Nuclear Safety 22(6):766-777.

Simpson, J. C. 1992. Dose Estimate Variability Caused by Air Model Uncertainties. PNL-7737 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Simpson, J. C. 1991. Effects of the Loss of Correlation Structure on Phase I Dose Estimates. PNL-7638 HEDR, Pacific Northwest Laboratory, Richland, Washington.

Soldat, J. K., N. M. Robinson, and D. A. Baker. 1974. Models and Computer Codes for Evaluating Radiation Doses. BNWL-1754, Pacific Northwest Laboratory, Richland, Washington.

Urness, P. J. 1981. "Food Habits and Nutrition." Mule and Blacktail Deer of North America, O. C. Wallmo, ed. University of Nebraska Press, Lincoln.

Whicker, F. W. and T. B. Kirchner. 1987. "PATHWAY: A Dynamic Food-Chain Model to Predict Radionuclide Ingestion After Fallout Deposition." Health Physics 52(6):717-737.

Whicker, F. W., and V. Schultz. 1982. Radioecology: Nuclear Energy and the Environment. Volumes I and II. CRC Press, Boca Raton, Florida.

Whicker, F. W. 1985. Addendum to the CSU Progress Report Dated January 17, 1985. Colorado State University, Fort Collins, Colorado.

APPENDIX A

LIST OF VARIABLE AND PARAMETER NAMES

APPENDIX A

LIST OF VARIABLE AND PARAMETER NAMES

State Variables

- Q_{usl} = quantity (activity) in the upper soil compartment, Ci/m^2
 Q_{rz} = quantity (activity) in the rooting zone soil compartment, Ci/m^2
 Q_{ov} = quantity (activity) in the outer vegetation compartment, Ci/m^2
 Q_{iv} = quantity (activity) in the inner vegetation compartment, Ci/m^2

Soil and Vegetation Rate Constants

- λ_{perc} = percolation rate constant, d^{-1}
 λ_{splash} = rainsplash (irrigation) rate constant, d^{-1}
 λ_{weath} = weathering rate constant, d^{-1}
 λ_{trans} = plant translocation rate constant, d^{-1}
 λ_{leach} = leaching to deeper soil rate constant, d^{-1}

Physical constants

- λ_{rad} = radiological decay constant, d^{-1}

Soil and Vegetation Variables

- R_{resus} = rate of radionuclide redeposition on vegetation from resuspension of soil, $Ci/m^2 \cdot d$
 R_{senc} = rate of radionuclide transfer from plants to soil by senescence of vegetation, $Ci/m^2 \cdot d$
 R_{root} = rate of radionuclide uptake through roots, $Ci/m^2 \cdot d$
 C = vegetation radionuclide concentration, Ci/kg

Biomass

- B = the current daily biomass, $kg(dry)/m^2$
 B_{max} = the maximum potential biomass, $kg(dry)/m^2$

- $B_{\min}$ = minimum winter biomass, kg(dry)/m²
 k_g = the growth rate constant, d⁻¹
 k_s = senescence rate constant, d⁻¹

Interception and Deposition

- I = areal deposition rate (input from atmospheric transport model), Ci/(m²·d)
 f_v = vegetation foliar interception fraction, dimensionless
 α = empirical foliar interception constant, m²/kg
 f_s = soil deposition fraction, equal to 1 - f_v

Resuspension Parameters

- ML = the mass loading factor of local soil in air, kg/m³
 V_d = the local deposition velocity of dust for soil or vegetation, m/d
 ρ_1 = the soil surface areal density to a depth of 1 mm, kg/m²

Root Uptake Parameters

- CR = plant-to-soil concentration ratio, unitless
 ρ_2 = upper soil layer areal density from 1 mm to 15 cm depth, kg/m²

Animal Feeding Variable and Parameters

- A_{cons} = animal radionuclide consumption rate, Ci/d
 f_{feed} = fraction of the total animal feed quantity that is feed type t
 Q_{feed} = total quantity of feed that the animal eats per day, kg/d (dry)
 C_{feed} = animal feed radionuclide concentration, Ci/kg
 t_s = production and storage time, days
 j_t = Julian day on which feed type t was most recently harvested

Animal Product Variables and Parameters

- A_p = animal product concentration, Ci/L (milk) or Ci/kg (beef, venison, poultry, eggs)

TF_p	=	animal product transfer coefficient, d/L (milk) or d/kg (beef, venison, poultry, eggs)
FS_a	=	animal soil ingestion rate, kg/d
F_{pa}	=	radionuclide concentration in commercially available milk, Ci/kg
f_{groc}	=	fraction of milk at a grocery store from collection center c
f_{cream}	=	fraction of milk at collection center c that came from location l

Individual Dose Calculations

DS	=	air submersion dose, rad
χ	=	integrated daily radionuclide air concentration, Ci·s/m ³ (input from atmospheric transport model)
DFS	=	the semi-infinite plume external dose rate factor, rad/d per Ci/m ³
Sh1	=	shielding factor for semi-infinite clouds, dimensionless
DG	=	the dose from groundshine, rad/d
DFU	=	the dose rate factor for surface plane deposition, rad/d per Ci/m ²
DFR	=	the dose rate factor for radionuclides in the soil, rad/d per Ci/m ²
Sh2	=	shielding factor for groundshine plane sources, unitless
D_{inh}	=	dose from inhaling contaminated air, rad
BR	=	age-dependent breathing rate, m ³ /s
DFI	=	age-dependent inhalation dose factor, rad/Ci inhaled
f_{time}	=	fraction of a day exposed to contaminated air, seconds
D_{ing}	=	ingestion dose from local or commercial foods, rem
DFG	=	age-dependent ingestion dose factor, rad/Ci
R_p	=	food product consumption rate, kg/d
F_p	=	the food product total concentration, from the environmental concentrations database, Ci/kg
th_p	=	holdup time, from production or harvest to consumption, days
D_{veg2}	=	age-dependent ingestion dose from leafy vegetables and fruits, rad

- $F_{p, iv}$ = radionuclide concentration of the inner vegetation compartment for leafy vegetables and fruits, Ci/kg
- $F_{p, ov}$ = radionuclide concentration of the outer vegetation compartment for leafy vegetables and fruits, Ci/kg
- WFA = surface contamination washing fraction, unitless

APPENDIX B

INPUT DATA FILE LISTINGS

[HEDR, LWDATA,MILK]CENSUS_TRACT_INDEX.ASC

1 AD01 WA Adams
2 AD02
3 AD03
4 AD04
5 AD05
6 AD06
7 AD07
8 AD08
9 AD09
10 AD10
11 BE01 WA Benton
12 BE02
13 BE03
14 BE04
15 BE05
16 BE06
17 BE07
18 BE08
19 BE09
20 FR01 WA Franklin
21 FR02
22 FR03
23 FR04
24 FR05
25 GR01 WA Grant
26 GR02
27 GR03
28 GR04
29 GR05
30 GR06
31 GR07
32 GR08
33 GR09
34 GR10
35 GR11
36 GR12
37 GR13
38 GR14
39 KI01 WA Kittitas
40 KI02
41 KI03
42 KI04
43 KI05
44 KI06
45 KI07
46 KI08
47 KI09
48 KI10
49 KI11
50 KL01 WA Klickitat
51 KL02
52 KL03
53 KL04
54 KL05
55 KL06
56 KL07
57 KL08
58 KL09
59 KL10
60 MO01 OR Morrow
61 MO02
62 MO03
63 UM01 OR Umatilla
64 UM02

B.1

66 UM04
67 UM05
68 UM06
69 UM07
70 UM08
71 UM09
72 UM10
73 UM11
74 UM12
75 UM13
76 UM14
77 WA01 WA Walla Walla
78 WA02
79 WA03
80 WA04
81 WA05
82 WA06
83 WA07
84 WA08
85 WA09
86 WA10
87 WA11
88 YA01 WA Yakima
89 YA02
90 YA03
91 YA04
92 YA05
93 YA06
94 YA07
95 YA08
96 YA09
97 YA10
98 YA11
99 YA12

[1 .RAWDATA.MILK]MILK_PROCESSOR_INDEX.ASC

- 1 Ritzville Dairy, Ritzville (AD2)
- 2 Morning Sun, Moses Lake (GR7)
- 3 Twin City (TriCities/Benton County)
- 4 Fairview Guernsey, Prosser (BE6)
- 5 Connell Dairy (FR1)
- 6 Willow Point, Grand Coulee (GR1)
- 7 Swanson's Dairy, Ephrata (GR5)
- 8 Gregory, Ellensburg (KI11)
- 9 Enfield, Ellensburg (?) (KI11)
- 10 Faust, Ellensburg (KI11)
- 11 Kayner's, Roslyn (KI5)
- 12 Del Ducco, Roslyn (KI5)
- 13 Sawyer's, Cle Elum (KI5)
- 14 Mayflower, Hermiston (UM3)
- 15 ??? Dairy, Pendleton (UM6)
- 16 Walla Walla Dairyman's (WA11)
- 17 Young's, Walla Walla (WA11)
- 18 Shady Lawn, Walla Walla (WA11)
- 19 Prescott Creamery (WA1)
- 20 Duff's, Walla Walla (WA11)
- 21 Selah Home Dairy (YA2)
- 22 Cascade, Yakima (YA5)
- 23 Yakima City Creamery (YA5)
- 24 Darigold, Yakima (YA5) + Distr., Kennewick (BE3)
- 25 Carnation, Sunnyside (YA9) + Distr., Richland (BE7)
- 26 Local Producers from YA9

Milk transport from SOURCES to processors/distributors - HEDR Phase I (1944-47)

26 processors/distributors (p/d) - see Milk P/D Index for identification by number

99 census divisions (c.d.) + "OTHR" (outside the study area)

a(i,j) = % of milk received by p/d j that came from c.d. i -- Column headings are p/d number

3 rows per c.d.: mean/min/max -- Sum of means for each column should equal 100%

p/d#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
AD01	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
AD02	1.0	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
	1.0																									
AD03	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
	1.0																									
AD04	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
AD05	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
AD06	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
AD07	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
AD08	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
AD09	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
AD10	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
BE01	0	0	.16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			.11																							
BE02	0	0	.21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			0																							
BE03	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
BE04	0	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
			.05																							
BE05	0	0	.15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.04	0	0	0
			.37	1.0																		0				
			.32	1.0																		0				
BE06	0	0	.42	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.09	0	0	0	0
			0	0																		0				
BE07	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
BE08	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
BE09	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
FR01	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[illegible]

KI05	0	0	0	0	0	0	0	0	0	0	1.0	1.0	.50	0	0	0	0	0	0	0	0	0	0	0	0	0
											1.0	1.0	.25													
											1.0	1.0	.75													
KI06	0	0	0	0	0	0	0	0	0	0	0	0	.50	0	0	0	0	0	0	0	0	0	0	0	0	0
													.25													
													.75													
KI07	0	.03	0	0	0	0	0	0	.78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.02	0	0
		0							.73														0			
		.08							.83														.07			
KI08	0	.10	0	0	0	0	0	0	.22	0	0	0	0	0	0	0	0	0	0	0	0	0	.07	0	0	0
		.05							.17														.02			
		.15							.27														.12			
KI09	0	0	0	0	0	0	0	.50	0	.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								.25		.25																
								.75		.75																
KI10	0	.07	0	0	0	0	0	.50	0	.50	0	0	0	0	0	0	0	0	0	0	0	0	.04	0	0	0
		.02						.25		.25													0			
		.12						.75		.75													.09			
KI11	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
KL01	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
KL02	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
KL03	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
KL04	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
KL05	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
KL06	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
KL07	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
KL08	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
KL09	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
KL10	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
MO01	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0
													1.0													
													1.0													
MO02	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
MO03	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
UM01	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
UM02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05	.05	.05	0	.05	0	0	0	0	0	0	0

[illegible]

[illegible]

[illegible]

B.11

[illegible]

B.12

[illegible]

WA11	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	
YA01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05	.10	.55	.30	0	0	0	0	0
																					0	.05	.40	.20					
																					.10	.15	.60	.35					
YA02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.30	.10	.10	.10	0	0	.40	0	0
																					.20	0	0	0			.30		
																					.40	.20	.20	.20			.50		
YA03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.10	.10	0	0	.40	.40	0
																					0	0	0	0			.30	.30	
																					.20	.20				.50	.50		
YA04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.30	.30	0	0	.40	0	0
																					.20	.20				.30			
																					.40	.40				.50			
YA05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05	.10	.30	.15	0	0	.40	0	0
																					0	.05	.20	.05			.30		
																					.10	.15	.40	.25			.50		
YA06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.10	.10	.10	0	0	.40	.30	0
																					0	0	0	0			.30	.20	
																					.20	.20	.20				.50	.40	
YA07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05	.05	.05	0	.75	.10	0
																					0	0	0	0			.60	0	
																					.10	.10	.10				.85	.20	
YA08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.25	.50	.25	0	0	0	0	0
																					.15	.40	.15						
																					.35	.60	.35						
YA09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.15	.10	.75	0	0	
																					.10	0	.60						
																					.20	.20	.85						
YA10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.15	.10	.75	0	0	
																					.10	0	.60						
																					.20	.20	.85						
YA11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.25	.50	.25	0	0	0	0	0
																					.15	.40	.15						
																					.35	.60	.35						
YA12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.25	.50	.25	0	0	0	0	0
																					.15	.40	.15						
																					.35	.60	.35						

Milk transport from processors/distributors to URBAN consumers - HEDR Phase I (1944-47)

26 processors/distributors (p/d) by p/d index no., plus "Family cow", "Local p/d", "Other" (outside the study area)

99 census divisions (c.d.)

a(i,j) = % of milk consumed by c.d. i that came from p/d j -- Column headings are p/d number

3 rows per c.d.: mean/min/max. Sum of means for each row should = 100% (0% if no urban areas in c.d.)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	FAM COW	LOC P/D	OTHER
AD01	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
AD02	.25	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	.75	
	0																											.50	
	.50																											1.0	
AD03	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
AD04	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
AD05	.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	.90	
	0																											.75	
	.25																											1.0	
AD06	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
AD07	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
AD08	.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	.90	
	0																											.75	
	.25																											1.0	
AD09	.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	.90	
	0																											.75	
	.25																											1.0	
AD10	0	.50	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	.50	
		.25																										.25	
		.75																										.75	
BE01	0	0	.40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.60	0	0	0	0
			.30																						.50				
			.50																						.70				
BE02	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
BE03	0	0	.60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.20	.20	0	0	0	0
			.40																					.10	.10				
			.70																					.30	.30				
BE04	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
BE05	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
BE06	0	0	0	.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.25	.25	0	0	0	0
			.40																				0	0					
			.60																				.50	.50					
BE07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.95	0	.05	0	0
																								0	.90	0			
																								1.0	.10	0			
BE08	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
BE09	0	0	.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.70	0	0	0	0
			.20																						.60				
			.40																						.80				
FR01	0	0	.30	0	.60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.05	.05	0	0	0	0

B.14

[illegible]

[illegible]

[illegible]

[HEL...RAWDATA.FEED]PASTURE_SEASONS.ASC

Pasture season dates -- HEDR Phase IA (1944-47)

2 dates (columns): 1=beginning of "Spring"; 2=beginning of "Late fall"

99 census divisions (c.d.); 365 --> no fresh forage for cows in this c.d.

Data values are sequential day number in year (beginning with Jan. 1 = Day 1)

3 rows per c.d.: mean (best estimate) / min / max (triangular distr.)

	1	2
AD01	365	365
	365	365
	365	365
AD02	365	365
	365	365
	365	365
AD03	365	365
	365	365
	365	365
AD04	365	365
	365	365
	365	365
AD05	365	365
	365	365
	365	365
AD06	365	365
	365	365
	365	365
AD07	365	365
	365	365
	365	365
AD08	365	365
	365	365
	365	365
AD09	365	365
	365	365
	365	365
AD10	365	365
	365	365
	365	365
BE01	110	280
	100	260
	120	300
BE02	105	280
	95	260
	115	300
BE03	105	280
	95	260
	115	300
BE04	105	280
	95	260
	115	300
BE05	110	280
	100	260
	120	300
BE06	110	280
	100	260
	120	300
BE07	105	280
	95	260
	115	300
BE08	105	280
	95	260
	115	300
BE09	105	280
	95	260
	115	300
FR01	105	280

	95 260		
	115 300		
FR02	365 365		
	365 365		
	365 365		
FR03	105 280		
	95 260		
	115 300		
FR04	105 280		
	95 260		
	115 300		
FR05	105 280		
	95 260		
	115 300		
GR01	130 270	! Estimated - Jan. 91 RAB	
	120 250		
	140 290		
GR02	365 365		
	365 365		
	365 365		
GR03	365 365		
	365 365		
	365 365		
GR04	365 365		
	365 365		
	365 365		
GR05	365 365		
	365 365		
	365 365		
GR06	365 365		
	365 365		
	365 365		
GR07	130 270	! Estimated - Oct. 91 RAB	(non-zero Reg. 1 & 2 fractions)
	120 250		
	140 290		
GR08	130 270	! Estimated - Oct. 91 RAB	(non-zero Reg. 1 & 2 fractions)
	120 250		
	140 290		
GR09	365 365		
	365 365		
	365 365		
GR10	365 365		
	365 365		
	365 365		
GR11	365 365		
	365 365		
	365 365		
GR12	365 365		
	365 365		
	365 365		
GR13	365 365		
	365 365		
	365 365		
GR14	365 365		
	365 365		
	365 365		
KI01	130 270		
	120 250		
	140 290		
KI02	365 365		
	365 365		
	365 365		
KI03	365 365		
	365 365		
	365 365		
KI04	365 365		

	65	365
	365	365
KI05	365	365
	365	365
	365	365
KI06	140	260
	130	240
	150	280
KI07	130	260
	120	240
	140	280
KI08	130	270
	120	250
	140	290
KI09	130	260
	120	240
	140	280
KI10	130	260
	120	240
	140	280
KI11	130	260
	120	240
	140	280
KL01	365	365
	365	365
	365	365
KL02	365	365
	365	365
	365	365
KL03	365	365
	365	365
	365	365
KL04	365	365
	365	365
	365	365
KL05	130	270
	120	250
	140	290
KL06	130	270
	120	250
	140	290
KL07	365	365
	365	365
	365	365
KL08	365	365
	365	365
	365	365
KL09	365	365
	365	365
	365	365
KL10	365	365
	365	365
	365	365
MO01	105	280
	95	260
	115	300
MO02	115	275
	105	255
	125	295
MO03	120	280
	110	260
	130	300
UM01	100	285
	90	265
	110	305
UM02	100	285

	90	265
	110	305
UM03	120	280
	110	260
	130	300
UM04	365	365
	365	365
	365	365
UM05	365	365
	365	365
	365	365
UM06	365	365
	365	365
	365	365
UM07	120	275
	110	255
	130	295
UM08	365	365
	365	365
	365	365
UM09	120	280
	110	260
	130	300
UM10	365	365
	365	365
	365	365
UM11	100	285
	90	265
	110	305
UM12	105	280
	95	260
	115	300
UM13	365	365
	365	365
	365	365
UM14	365	365
	365	365
	365	365
WA01	365	365
	365	365
	365	365
WA02	365	365
	365	365
	365	365
WA03	365	365
	365	365
	365	365
WA04	110	280
	100	260
	120	300
WA05	365	365
	365	365
	365	365
WA06	365	365
	365	365
	365	365
WA07	365	365
	365	365
	365	365
WA08	105	280
	95	260
	115	300
WA09	105	280
	95	260
	115	300
WA10	105	280

	95	260
	115	300
WA11	105	280
	95	260
	115	300
YA01	365	365
	365	365
	365	365
YA02	120	270
	110	250
	130	290
YA03	120	270
	110	250
	130	290
YA04	120	270
	110	250
	130	290
YA05	115	270
	105	250
	125	290
YA06	115	270
	105	250
	125	290
YA07	115	270
	105	250
	125	290
YA08	365	365
	365	365
	365	365
YA09	110	280
	100	260
	120	300
YA10	110	280
	100	260
	120	300
YA11	365	365
	365	365
	365	365
YA12	365	365
	365	365
	365	365

Milk cow feeding regime practices in each study area census division - HEDR Phase I (1944-47)

4 feeding regimes -- 1:G,AH,P,S; 2:G,AH,P; 3:G,AH; 4:G,GH

99 census divisions (c.d.)

a(i,j) = % of milk produced in c.d. i from cows fed using feeding regime j

3 rows per c.d.: mean/min/max -- Sum of means for each row should = 100% (0 if no milk prod. in c.d.)

	1	2	3	4
AD01	0	0	5	95
			0	90
			10	100
AD02	0	0	5	95
			0	90
			10	100
AD03	0	0	5	95
			0	90
			10	100
AD04	0	0	5	95
			0	90
			10	100
AD05	0	0	5	95
			0	90
			10	100
AD06	0	0	5	95
			0	90
			10	100
AD07	0	0	5	95
			0	90
			10	100
AD08	0	0	5	95
			0	90
			10	100
AD09	0	0	5	95
			0	90
			10	100
AD10	0	0	5	95
			0	90
			10	100
BE01	45	45	10	0
	20	20	0	
	65	65	15	
BE02	45	50	5	0
	20	20	0	
	65	65	15	
BE03	45	50	5	0
	20	20	0	
	65	65	15	
BE04	45	50	5	0
	20	20	0	
	65	65	15	
BE05	45	50	5	0
	20	20	0	
	65	65	15	
BE06	0	0	95	5
			90	0
			100	10
BE07	45	50	5	0
	20	20	0	
	65	65	15	
BE08	0	5	0	95
		0		90
		10		100
BE09	0	5	0	95
		0		90
		10		100
FR01	45	45	0	10

	20	20		0
	65	65		15
FR02	0	0	5	95
			0	90
			10	100
FR03	45	45	10	0
	20	20	0	
	65	65	15	
FR04	45	45	0	10
	20	20		0
	65	65		15
FR05	45	45	10	0
	20	20	0	
	65	65	15	
GR01	0	0	5	95
			0	90
			10	100
GR02	0	0	5	95
			0	90
			10	100
GR03	0	0	5	95
			0	90
			10	100
GR04	0	0	5	95
			0	90
			10	100
GR05	0	0	5	95
			0	90
			10	100
GR06	0	0	5	95
			0	90
			10	100
GR07	45	50	5	0
	20	25	0	
	65	70	10	
GR08	45	50	5	0
	20	25	0	
	65	70	10	
GR09	0	0	5	95
			0	90
			10	100
GR10	0	0	5	95
			0	90
			10	100
GR11	0	0	5	95
			0	90
			10	100
GR12	0	0	5	95
			0	90
			10	100
GR13	0	0	5	95
			0	90
			10	100
GR14	0	0	5	95
			0	90
			10	100
KI01	45	45	10	0
	20	20	0	
	65	65	15	
KI02	0	0	5	95
			0	90
			10	100
KI03	0	0	5	95
			0	90
			10	100
KI04	0	0	5	95

			0	90
			10	100
KI05	0	0	5	95
			0	90
			10	100
KI06	45	45	10	0
	20	20	0	
	65	65	15	
KI07	45	45	10	0
	20	20	0	
	65	65	15	
KI08	45	45	10	0
	20	20	0	
	65	65	15	
KI09	45	45	10	0
	20	20	0	
	65	65	15	
KI10	45	45	10	0
	20	20	0	
	65	65	15	
KI11	45	45	10	0
	20	20	0	
	65	65	15	
KL01	0	0	5	95
			0	90
			10	100
KL02	0	0	5	95
			0	90
			10	100
KL03	0	0	5	95
			0	90
			10	100
KL04	0	0	5	95
			0	90
			10	100
KL05	0	90	10	0
	85	0		
	100	15		
KL06	0	90	10	0
	85	0		
	100	15		
KL07	0	0	5	95
			0	90
			10	100
KL08	0	0	5	95
			0	90
			10	100
KL09	0	0	5	95
			0	90
			10	100
KL10	0	0	5	95
			0	90
			10	100
M001	45	45	10	0
	20	20	0	
	65	65	15	
M002	45	45	10	0
	20	20	0	
	65	65	15	
M003	45	45	10	0
	20	20	0	
	65	65	15	
UM01	0	0	5	95
			0	90
			10	100
UM02	45	45	10	0

	20	20	0	
	65	65	15	
UM03	45	45	10	0
	20	20	0	
	65	65	15	
UM04	0	0	5	95
			0	90
			10	100
UM05	0	0	5	95
			0	90
			10	100
UM06	0	0	5	95
			0	90
			10	100
UM07	45	45	10	0
	20	20	0	
	65	65	15	
UM08	0	0	5	95
			0	90
			10	100
UM09	45	45	10	0
	20	20	0	
	65	65	15	
UM10	0	0	5	95
			0	90
			10	100
UM11	45	45	10	0
	20	20	0	
	65	65	15	
UM12	45	45	10	0
	20	20	0	
	65	65	15	
UM13	0	0	5	95
			0	90
			10	100
UM14	0	0	5	95
			0	90
			10	100
WA01	0	0	5	95
			0	90
			10	100
WA02	0	0	5	95
			0	90
			10	100
WA03	0	0	5	95
			0	90
			10	100
WA04	45	45	10	0
	20	20	0	
	65	65	15	
WA05	0	0	5	95
			0	90
			10	100
WA06	0	0	5	95
			0	90
			10	100
WA07	0	0	5	95
			0	90
			10	100
WA08	45	45	10	0
	20	20	0	
	65	65	15	
WA09	45	45	10	0
	20	20	0	
	65	65	15	
WA10	45	45	10	0

	20	20	0	
	65	65	15	
WA11	45	45	10	0
	20	20	0	
	65	65	15	
YA01	0	0	5	95
			0	90
			10	100
YA02	45	45	10	0
	20	20	0	
	65	65	15	
YA03	45	45	10	0
	20	20	0	
	65	65	15	
YA04	45	45	10	0
	20	20	0	
	65	65	15	
YA05	45	45	10	0
	20	20	0	
	65	65	15	
YA06	45	45	10	0
	20	20	0	
	65	65	15	
YA07	45	45	10	0
	20	20	0	
	65	65	15	
YA08	0	0	5	95
			0	90
			10	100
YA09	45	45	10	0
	20	20	0	
	65	65	15	
YA10	45	45	10	0
	20	20	0	
	65	65	15	
YA11	0	0	5	95
			0	90
			10	100
YA12	0	0	5	95
			0	90
			10	100

[HEL...RAWDATA.FEED] CUTTING.DAT

1 166 211 256 200
2 166 211 256 200
3 166 211 256 200
4 166 211 256 200
5 166 211 256 200
6 166 211 256 200
7 176 221 256 210
8 176 221 256 210
9 166 211 256 200
10 166 211 256 200
11 156 201 246 190
12 151 196 241 185
13 151 196 241 185
14 151 196 241 185
15 156 201 246 190
16 156 201 246 190
17 151 196 241 185
18 151 196 241 185
19 151 196 241 185
20 151 196 241 185
21 151 196 241 185
22 151 196 241 185
23 151 196 241 185
24 151 196 241 185
25 156 201 246 190
26 161 206 251 195
27 166 211 256 200
28 166 211 256 200
29 156 201 246 190
30 156 201 246 190
31 161 206 251 195
32 166 211 256 200
33 166 211 256 200
34 166 211 256 200
35 161 206 251 195
36 161 206 251 195
37 156 201 246 190
38 151 196 241 185
39 176 221 266 210
40 186 231 276 220
41 186 231 276 220
42 186 231 276 220
43 176 221 266 210
44 186 231 276 220
45 176 221 266 210
46 176 221 266 210
47 176 221 266 210
48 176 221 266 210
49 176 221 266 210
50 176 221 266 210
51 166 211 256 200
52 171 216 261 205
53 176 221 266 210
54 176 221 266 210
55 176 221 266 210
56 171 216 261 205
57 176 221 266 210
58 166 211 256 200
59 166 211 256 210
60 151 196 241 185
61 161 206 251 195
62 166 211 256 200
63 146 191 236 180
64 146 191 236 180

65	166	211	256	200
66	156	201	246	190
67	146	191	236	180
68	166	211	256	200
69	166	211	256	200
70	166	211	256	200
71	166	211	256	200
72	166	211	256	200
73	146	191	236	180
74	151	196	241	185
75	166	211	256	200
76	166	211	256	200
77	161	206	251	195
78	161	206	251	195
79	156	201	246	190
80	156	201	246	190
81	156	201	246	190
82	156	201	246	190
83	161	206	251	195
84	151	196	241	185
85	151	196	241	185
86	151	196	241	185
87	151	196	241	185
88	166	211	256	200
89	166	211	256	200
90	166	211	256	200
91	166	211	256	200
92	161	206	251	195
93	161	206	251	195
94	161	206	251	195
95	166	211	256	200
96	156	201	246	190
97	156	201	246	190
98	156	201	246	190
99	176	221	266	210

[HL RAWDATA.FOOD]FALL15.ASC

m1

28

21.25 7.543 0 110.7 6.355 82.91 93.28 0 1007 1.205 6.956
33.23 16.18 0 109.2 14.74 90.39 106.9 0 581.1 4.444 5.826
1104 262.8 0 -558.8 123.9 -330.4 -71.01 0 5425 -3.544 0
262.8 261.6 0 362.1 51.28 -232.5 76.05 0 1645 -4.96 0
0 0 0 0 0 0 0 0 0 0 0
-558.8 362.1 0 11930 40.25 -1254 7305 0 16370 -36.89 0
123.9 51.28 0 40.25 217.4 338.5 -121.3 0 762.5 -4.391 0
-330.4 -232.5 0 -1254 338.5 8171 1149 0 -5095 80.35 0
-71.01 76.05 0 7305 -121.3 1149 11420 0 18940 -25.65 0
0 0 0 0 0 0 0 0 0 0 0
5425 1645 0 16370 762.5 -5095 18940 0 3.376E+005 69.73 0
-3.544 -4.96 0 -36.89 -4.391 80.35 -25.65 0 69.73 19.75 0
0 0 0 0 0 0 0 0 0 0 33.94

m2

164

70.93 8.478 13.25 132.4 29.08 233.9 82.5 0.05535 739.2 3.787 6.956
55.73 11.76 30.11 108.3 32.7 128.3 119.1 0.5535 385.1 6.851 5.826
3106 -47.2 -105.7 1412 140.3 -364.6 -33.2 -1.357 2984 -9.163 0
-47.2 138.4 -24.79 122.2 -8.847 -138.8 -169.6 0.4688 308.1 -2.256 0
-105.7 -24.79 906.5 241.2 33.48 583.2 -143.5 3.316 -139.2 0.6188 0
1412 122.2 241.2 11720 -513.3 222.2 77.39 -1.738 6796 80.09 0
140.3 -8.847 33.48 -513.3 1069 -557.9 -354.6 2.317 -327.5 -31.36 0
-364.6 -138.8 583.2 222.2 -557.9 16460 932.3 0.639 -2668 -12.3 0
-33.2 -169.6 -143.5 77.39 -354.6 932.3 14200 -4.683 1055 170.6 0
-1.357 0.4688 3.316 -1.738 2.317 0.639 -4.683 0.3064 30.06 0.1289 0
2984 308.1 -139.2 6796 -327.5 -2668 1055 30.06 1.483E+005 -36.94 0
-9.163 -2.256 0.6188 80.09 -31.36 -12.3 170.6 0.1289 -36.94 46.93 0
0 0 0 0 0 0 0 0 0 0 33.94

m3

217

103 10.57 31.86 207.4 21.66 307.2 101.7 0.6435 873 3.999 6.956
66.14 13.51 47.31 142.5 32.34 144 114.5 5.591 388.1 7.284 5.826
4375 -99.21 450.6 2319 263.1 1086 818 -37.35 1643 44.8 0
-99.21 182.6 30.68 65.48 18.35 -223.8 -168.7 -1.435 62.93 -3.838 0
450.6 30.68 2238 1261 151.4 1151 926.4 32.27 1175 5.858 0
2319 65.48 1261 20310 18.43 2155 1306 -24.7 9900 119.4 0
263.1 18.35 151.4 18.43 1046 153.7 218.5 13.2 1393 11.78 0
1086 -223.8 1151 2155 153.7 20740 2177 -53.14 9837 -61.89 0
818 -168.7 926.4 1306 218.5 2177 13110 223.4 9066 15.85 0
-37.35 -1.435 32.27 -24.7 13.2 -53.14 223.4 31.25 -56.41 -2.199 0
1643 62.93 1175 9900 1393 9837 9066 -56.41 1.506E+005 -59.36 0
44.8 -3.838 5.858 119.4 11.78 -61.89 15.85 -2.199 -59.36 53.05 0
0 0 0 0 0 0 0 0 0 0 33.94

m4

267

130.7 13.25 41.26 259.4 26.98 365.5 98.71 0.429 1002 4.011 6.956
74.1 17.31 58.06 178.8 35.9 184 119.5 3.93 467.5 7.733 5.826
5490 -197.5 4.302 2278 61.18 -354.4 -460.4 -35.82 1663 -76.21 0
-197.5 299.6 29.14 476.5 -21.75 95.53 -18.61 -0.2721 1659 -19.41 0
4.302 29.14 3371 1920 406.4 2168 291.3 3.651 2470 -1.347 0
2278 476.5 1920 31960 706 1513 320.4 -54.1 26160 -62.21 0
61.18 -21.75 406.4 706 1289 640.7 -158.7 -6.349 822.4 13.6 0
-354.4 95.53 2168 1513 640.7 33850 1627 -52.06 12300 61.18 0
-460.4 -18.61 291.3 320.4 -158.7 1627 14280 31.93 2681 40.65 0
-35.82 -0.2721 3.651 -54.1 -6.349 -52.06 31.93 15.44 45.93 -0.699 0
1663 1659 2470 26160 822.4 12300 2681 45.93 2.186E+005 -404.9 0
-76.21 -19.41 -1.347 -62.21 13.6 61.18 40.65 -0.699 -404.9 59.8 0
0 0 0 0 0 0 0 0 0 0 33.94

B.31

m5
189
170.7 13.84 54.83 284.7 41.75 406.7 96.62 1.162 1044 4.849 6.956
98.8 22.32 72.23 191.6 58.11 238.5 133.4 6.199 738.3 7.459 5.826
9761 -518.3 121.3 2897 -68.89 2475 316.3 -13.47 3720 18.42 0
-518.3 498.3 120.9 770 146.6 708.2 277 5.121 4796 9.991 0
121.3 120.9 5217 2478 348.3 2257 915.3 5.373 4053 15.62 0
2897 770 2478 36720 2929 3611 4832 -2.376 23200 183 0
-68.89 146.6 348.3 2929 3377 568.3 -93.02 -25.94 4033 104.5 0
2475 708.2 2257 3611 568.3 56900 8274 -63.59 52840 204.6 0
316.3 277 915.3 4832 -93.02 8274 17800 -20.68 16840 -16.92 0
-13.47 5.121 5.373 -2.376 -25.94 -63.59 -20.68 38.43 -260.9 -2.497 0
3720 4796 4053 23200 4033 52840 16840 -260.9 5.451E+005 468.1 0
18.42 9.991 15.62 183 104.5 204.6 -16.92 -2.497 468.1 55.64 0
0 0 0 0 0 0 0 0 0 0 33.94

m6
680
166.7 15.02 78.38 291.7 49.13 310.7 96.42 2.096 477.2 8.779 6.956
105.6 23.04 91.69 214.3 59.76 208.9 141.5 9.583 469.8 20.16 5.826
11150 -413.6 -164.6 4434 466.9 1477 -149.4 -43.5 297.6 95.76 0
-413.6 531 150 740.7 34.43 -130 140.2 -10.16 -75.78 -16.26 0
-164.6 150 8408 1709 224.7 -555.4 1077 18.45 1637 68.38 0
4434 740.7 1709 45920 1306 4834 3760 -63.66 10070 -17.28 0
466.9 34.43 224.7 1306 3572 -149.8 -329.8 -6.872 1151 60.23 0
1477 -130 -555.4 4834 -149.8 43620 2098 -130.1 17070 294.7 0
-149.4 140.2 1077 3760 -329.8 2098 20020 47.46 7844 -28.52 0
-43.5 -10.16 18.45 -63.66 -6.872 -130.1 47.46 91.83 -211.6 -0.1931 0
297.6 -75.78 1637 10070 1151 17070 7844 -211.6 2.207E+005 284.1 0
95.76 -16.26 68.38 -17.28 60.23 294.7 -28.52 -0.1931 284.1 406.2 0
0 0 0 0 0 0 0 0 0 0 33.94

8.32

m7
34
21.88 7.608 7.97 102.8 7.632 74.2 92.24 0 907.4 0 6.956
41.61 18.29 27.58 160 23.11 107.5 143.4 0 556.8 0 5.826
1731 303.6 293.7 2190 -51.91 3149 2548 0 5931 0 0
303.6 334.4 -24.71 260.4 60 344 508.7 0 254.5 0 0
293.7 -24.71 760.5 838.4 -62.44 -68.19 1523 0 2073 0 0
2190 260.4 838.4 25610 -573.1 5694 5806 0 32340 0 0
-51.91 60 -62.44 -573.1 533.9 320.4 -725.6 0 -334.5 0 0
3149 344 -68.19 5694 320.4 11560 3607 0 22270 0 0
2548 508.7 1523 5806 -725.6 3607 20560 0 -718.6 0 0
0 0 0 0 0 0 0 0 0 0 0
5931 254.5 2073 32340 -334.5 22270 -718.6 0 3.1E+005 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 33.94

m8
144
62.59 7.833 13.28 148.4 34.72 211.9 69.23 0.08303 798.1 3.268 6.956
43.45 10.06 23.41 102.9 41.99 123.2 88.13 0.8372 430 6.54 5.826
1888 -33.2 -75.27 764.6 350.3 791.9 283.3 2.764 1625 -43.47 0
-33.2 101.1 23.31 61.05 24.07 30.96 95.71 -0.6398 237.8 -3.025 0
-75.27 23.31 548.2 253 -41.29 360.4 198.1 1.098 -432.9 24.35 0
764.6 61.05 253 10590 -34.57 1179 1696 -9.65 4381 -38.36 0
350.3 24.07 -41.29 -34.57 1763 -806.7 240.5 2.285 1065 0 0
791.9 30.96 360.4 1179 -806.7 15170 868.3 2.268 18110 -88.6 0
283.3 95.71 198.1 1696 240.5 868.3 7767 1.328 7428 2.306 0
2.764 -0.6398 1.098 -9.65 2.285 2.268 1.328 0.7009 7.561 0.2683 0
1625 237.8 -432.9 4381 1065 18110 7428 7.561 1.849E+005 -222.2 0
-43.47 -3.025 24.35 -38.36 0 -88.6 2.306 0.2683 -222.2 42.77 0
0 0 0 0 0 0 0 0 0 0 33.94

m9
224

87 11.12 35.39 194.2 26.7 288.6 90.92 0.3114 819.9 2.994 6
 55.36 16.08 51.7 135.1 32.04 152.99 2.567 376.2 4.967 5.826
 3065 -181.6 100.2 1609 106.4 -808 21.92 8.527 1312 -14.3 0
 -181.6 258.5 -38.24 -267.3 18.55 -87.99 -195.8 3.384 -217.7 -0.8785 0
 100.2 -38.24 2673 1964 101.1 1525 573.3 0.5309 175.1 -8.732 0
 1609 -267.3 1964 18260 -77.95 1623 -521.8 -18.04 13270 -54.38 0
 106.4 18.55 101.1 -77.95 1027 -34.1 260.1 -0.658 -494.2 47.11 0
 -808 -87.99 1525 1623 -34.1 23110 1339 23.02 8808 49.09 0
 21.92 -195.8 573.3 -521.8 260.1 1339 9800 -12.96 7039 12.79 0
 8.527 3.384 0.5309 -18.04 -0.658 23.02 -12.96 6.589 18.35 -0.5738 0
 1312 -217.7 175.1 13270 -494.2 8808 7039 18.35 1.415E+005 -136.4 0
 -14.3 -0.8785 -8.732 -54.38 47.11 49.09 12.79 -0.5738 -136.4 24.67 0
 0 0 0 0 0 0 0 0 0 0 33.94

m10

269
 113.1 9.288 43.86 214.8 25.11 303.9 97.8 0.4843 807.2 3.73 6.956
 61.08 13.14 59.22 165.35 163.6 127.4 2.996 379.7 6.977 5.826
 3731 -141.3 770.5 3236 284.3 949.2 754.6 16.1 3084 -18.33 0
 -141.3 172.7 37.36 97.6 -27.14 -202.1 71.99 -0.5513 -44.91 2.017 0
 770.5 37.36 3507 2404 246.7 823.5 399.8 1.242 1417 15.7 0
 3236 97.6 2404 27230 231 2025 1829 -42.02 14470 -27.63 0
 284.3 -27.14 246.7 231 1225 -561.1 392.3 -6.502 -239.2 24.18 0
 949.2 -202.1 823.5 2025 -561.1 26760 -333.4 64.2 9937 -6.848 0
 754.6 71.99 399.8 1829 392.3 -333.4 16220 -22.13 1644 36.44 0
 16.1 -0.5513 1.242 -42.02 -6.502 64.2 -22.13 8.976 35.26 -0.6062 0
 3084 -44.91 1417 14470 -239.2 9937 1644 35.26 1.441E+005 188.1 0
 -18.33 2.017 15.7 -27.63 24.18 -6.848 36.44 -0.6062 188.1 48.68 0
 0 0 0 0 0 0 0 0 0 0 33.94

m11

213
 117.9 11.81 43.8 205.4 25.69 260.2 70.72 0.4636 583.6 4.587 6.956
 75.44 16.53 61.31 167.2 40.21 162.5 106.4 4.311 398.3 8.288 5.826
 5691 -234.4 952.7 3571 60.66 1005 1132 -28.94 2855 -24.39 0
 -234.4 273.1 -21.28 27.64 89.04 -273.9 117.8 -4.773 -467.4 -13.56 0
 952.7 -21.28 3758 1118 184.9 -149.4 123.9 12.68 1319 87.91 0
 3571 27.64 1118 27970 894.4 2962 1406 1.442 6262 -115.1 0
 60.66 89.04 184.9 894.4 1617 431.1 -166.9 -11.27 1650 17.66 0
 1005 -273.9 -149.4 2962 431.1 26390 -881.6 -45.52 9254 -2.693 0
 1132 117.8 123.9 1406 -166.9 -881.6 11320 -26.6 6908 66.15 0
 -28.94 -4.773 12.68 1.442 -11.27 -45.52 -26.6 18.58 -99.58 0.1072 0
 2855 -467.4 1319 6262 1650 9254 6908 -99.58 1.587E+005 59.43 0
 -24.39 -13.56 87.91 -115.1 17.66 -2.693 66.15 0.1072 59.43 68.7 0
 0 0 0 0 0 0 0 0 0 0 33.94

m12

1144
 105.2 13.45 71.71 217.3 32.56 216.9 85.62 2.214 337.1 7.214 6.956
 74.94 20.18 85.2 154.6 44.29 155 127.4 11.98 322.1 12.95 5.826
 5616 -314.6 140.5 985 199.2 290.3 -601.6 -44.01 -748.4 -19.41 0
 -314.6 407.3 55.02 6.241 -2.681 -203.3 110.6 -4.595 -253.5 -14.37 0
 140.5 55.02 7259 1370 358.5 -250.9 1628 28.59 109.8 35.31 0
 985 6.241 1370 23910 897.2 1270 1320 -5.559 4583 80.09 0
 199.2 -2.681 358.5 897.2 1962 212.8 -197.5 -11.68 371 6.309 0
 290.3 -203.3 -250.9 1270 212.8 24020 296.2 -105.9 8637 100.3 0
 -601.6 110.6 1628 1320 -197.5 296.2 16240 -79.4 2463 339.9 0
 -44.01 -4.595 28.59 -5.559 -11.68 -105.9 -79.4 143.6 -84.93 5.121 0
 -748.4 -253.5 109.8 4583 371 8637 2463 -84.93 1.038E+005 337.9 0
 -19.41 -14.37 35.31 80.09 6.309 100.3 339.9 5.121 337.9 167.7 0
 0 0 0 0 0 0 0 0 0 0 33.94

m13

9
 91.01 11.97 66.28 251.3 35.65 328.7 87.33 1.315 713.1 5.376 6.956
 55.19 12.81 69.07 182.9 57.48 307.2 125.2 3.944 465.6 8.864 5.826

3046 -32.53 -964.5 5995 -574.2 4950 311.1 -105.4 14700 -181.5 0
 -32.53 164.1 525.6 -30.45 -123.7 -2125 -497.4 17.53 1121 37.02 0
 -964.5 525.6 4771 480 1993 -10500 -5580 226.1 -8940 -38.57 0
 5995 -30.45 480 33440 1682 -7639 -8314 -279.1 -14390 -1.621 0
 -574.2 -123.7 1993 1682 3304 1289 -2462 124.4 -15150 -172.2 0
 4950 -2125 -10500 -7639 1289 94340 10620 -265.3 19160 -893.1 0
 311.1 -497.4 -5580 -8314 -2462 10620 15690 -128.9 21980 -144.3 0
 -105.4 17.53 226.1 -279.1 124.4 -265.3 -128.9 15.55 -594.9 -3.671 0
 14700 1121 -8940 -14390 -15150 19160 21980 -594.9 2.167E+005 -1849 0
 -181.5 37.02 -38.57 -1.621 -172.2 -893.1 -144.3 -3.671 -1849 78.58 0
 0 0 0 0 0 0 0 0 0 0 33.94

m14

14
 18.36 6.055 2.967 107.9 12.7 57.14 121.8 0 873.9 1.785 6.956
 33.08 10.06 11.08 142.2 27.3 60.06 141.6 0 597.1 4.91 5.826
 1094 102.2 34.44 668 -90.3 1798 2060 0 -19.75 32.16 0
 102.2 101.3 8.584 34.36 -51.38 256.9 233.7 0 2584 -11.61 0
 34.44 8.584 122.7 315.1 108.6 -58.54 58.02 0 -416.7 -5.711 0
 668 34.36 315.1 20230 3351 1930 5799 0 15030 602.7 0
 -90.3 -51.38 108.6 3351 745.3 -132.8 1291 0 2087 95.04 0
 1798 256.9 -58.54 1930 -132.8 3607 4956 0 1219 77.55 0
 2060 233.7 58.02 5799 1291 4956 20040 0 12760 116.1 0
 0 0 0 0 0 0 0 0 0 0 0
 -19.75 2584 -416.7 15030 2087 1219 12760 0 3.565E+005 -93.82 0
 32.16 -11.61 -5.711 602.7 95.04 77.55 116.1 0 -93.82 24.11 0
 0 0 0 0 0 0 0 0 0 0 33.94

m15

77
 73.58 4.88 7.302 161.3 21.25 216.2 84.11 0.2283 734.7 3.701 6.956
 48.95 8.703 15.18 191.8 25.29 99.12 115.6 1.571 377.6 6.895 5.826
 2397 -75.84 -19.32 1502 -111.4 -63.08 -1098 0.5382 591.6 -0.6751 0
 -75.84 75.74 17.44 -128.5 -13.21 -199.3 -86.55 0.7108 151.2 3.001 0
 -19.32 17.44 230.4 576.3 23.42 326.5 698.6 -0.02384 -429.9 4.187 0
 1502 -128.5 576.3 36770 1324 3897 110.9 -1.506 4417 -142.8 0
 -111.4 -13.21 23.42 1324 639.8 5.014 -544.1 5.323 1824 11.16 0
 -63.08 -199.3 326.5 3897 5.014 9825 3038 7.473 561.5 -97.74 0
 -1098 -86.55 698.6 110.9 -544.1 3038 13370 -10.35 -10570 70.97 0
 0.5382 0.7108 -0.02384 -1.506 5.323 7.473 -10.35 2.467 -137.6 2.675 0
 591.6 151.2 -429.9 4417 1824 561.5 -10570 -137.6 1.426E+005 -119.8 0
 -0.6751 3.001 4.187 -142.8 11.16 -97.74 70.97 2.675 -119.8 47.55 0
 0 0 0 0 0 0 0 0 0 0 33.94

m16

89
 96.48 11.05 18.47 222.9 25.13 325.7 76.14 0.06227 970.1 4.199 6.956
 57.78 14.7 29.43 155.2 33.99 135.9 94.68 0.5881 404.4 6.945 5.826
 3339 -22.08 197.2 1148 -255.3 1664 -175.1 -1.189 584.2 22.47 0
 -22.08 216 78.27 892.1 -17.49 71.89 -37.57 -0.7001 1849 16.13 0
 197.2 78.27 865.8 1041 -25.01 643.7 487.5 -0.2596 2035 28.61 0
 1148 892.1 1041 24100 -10.55 2552 925.9 -4.565 19970 132.6 0
 -255.3 -17.49 -25.01 -10.55 1155 -115.5 -9.655 -1.579 412.4 46.27 0
 1664 71.89 643.7 2552 -115.5 18460 849 -9.829 11320 -62.28 0
 -175.1 -37.57 487.5 925.9 -9.655 849 8964 -4.287 5016 91.39 0
 -1.189 -0.7001 -0.2596 -4.565 -1.579 -9.829 -4.287 0.3459 48.05 1.033 0
 584.2 1849 2035 19970 412.4 11320 5016 48.05 1.636E+005 1039 0
 22.47 16.13 28.61 132.6 46.27 -62.28 91.39 1.033 1039 48.23 0
 0 0 0 0 0 0 0 0 0 0 33.94

m17

113
 125.4 16.77 29.15 303.3 25.73 380.4 117.2 0.6919 1032 3.297 6.956
 72.74 22.3 39.94 243.1 38.12 198.4 145.1 5.293 556.2 5.409 5.826
 5292 -53.53 -296.4 3944 -535.1 -1385 2237 26.95 -1092 -6.295 0
 -53.53 497.3 -27.61 471.7 -160.7 -26.54 501.4 -3.069 3113 20.99 0

-296.1 -27.61 1595 1806 94.4 1696 567.8 -10.36 5376 17.93 0
 3944 471.7 1806 59100 -1195 10420 1446 -28.31 27990 -15.78 0
 -535.1 -160.7 94.4 -1195 1453 -423.4 -425.8 -17.96 -4282 -17.32 0
 -1385 -26.54 1696 10420 -423.4 39350 1669 -88.19 15440 151.3 0
 2237 501.4 567.8 1446 -425.8 1669 21040 -28.41 12990 80.03 0
 26.95 -3.069 -10.36 -28.31 -17.96 -88.19 -28.41 28.02 288.5 -1.231 0
 -1092 3113 5376 27990 -4282 15440 12990 288.5 3.093E+005 523.4 0
 -6.295 20.99 17.93 -15.78 -17.32 151.3 80.03 -1.231 523.4 29.25 0
 0 0 0 0 0 0 0 0 0 33.94

m18

74
 170.5 15.97 30.57 357.7 33.39 426.1 115 0.2352 1099 4.861 6.956
 112.7 31.01 50.6 262.6 42.79 230.6 204.9 2.013 724.1 9.142 5.826
 12710 -402 -154 2161 -183.3 77.98 -1987 10.21 -3021 39.17 0
 -402 961.5 40.79 -1083 -18.57 -114.4 -292.3 1.686 -3413 2.551 0
 -154 40.79 2560 -172.8 -75.78 1575 72.58 1.223 -476.3 18.04 0
 2161 -1083 -172.8 68970 -22.47 -7508 12590 9.518 31760 391.4 0
 -183.3 -18.57 -75.78 -22.47 1831 680.7 -403.3 -7.926 2262 35.21 0
 77.98 -114.4 1575 -7508 680.7 53160 1228 -9.284 15030 364.7 0
 -1987 -292.3 72.58 12590 -403.3 1228 41990 40.85 21220 16.86 0
 10.21 1.686 1.223 9.518 -7.926 -9.284 40.85 4.054 42.28 7.878 0
 -3021 -3413 -476.3 31760 2262 15030 21220 42.28 5.244E+005 370.7 0
 39.17 2.551 18.04 391.4 35.21 364.7 16.86 7.878 370.7 83.58 0
 0 0 0 0 0 0 0 0 0 33.94

m19

263
 172.3 15.28 55.32 307.6 46.76 332.9 109.6 1.813 578.1 8.967 6.956
 122.7 26.79 75.83 223.6 62.58 206.6 159.6 9.507 484.9 15.95 5.826
 15050 -397.7 632.6 3841 606.6 3472 -1586 -90.98 2975 -215.2 0
 -397.7 717.6 46.72 -53.92 -124.1 193.7 -55.57 36.42 662.5 -20.93 0
 632.6 46.72 5750 1492 -526.8 -1097 931.8 38.21 625.1 67.72 0
 3841 -53.92 1492 50020 -433.9 1432 5853 -48.9 3796 128.4 0
 606.6 -124.1 -526.8 -433.9 3917 801.5 -39.95 -47.6 -2701 -21.96 0
 3472 193.7 -1097 1432 801.5 42670 4648 -198.3 21340 230.6 0
 -1586 -55.57 931.8 5853 -39.95 4648 25470 -78.89 -5649 475.9 0
 -90.98 36.42 38.21 -48.9 -47.6 -198.3 -78.89 90.38 -230.5 -5.761 0
 2975 662.5 625.1 3796 -2701 21340 -5649 -230.5 2.351E+005 -355.7 0
 -215.2 -20.93 67.72 128.4 -21.96 230.6 475.9 -5.761 -355.7 254.3 0
 0 0 0 0 0 0 0 0 0 33.94

m20

20
 14.39 11.81 0 81.32 2.708 83.46 79.06 0 899.3 1.25 6.956
 33.97 29.68 0 121.7 10.51 124.2 106.3 0 510.1 4.154 5.826
 1154 793.5 0 835.3 196.3 1739 2297 0 849 1.411 0
 793.5 881.1 0 -65.05 221.1 1649 1079 0 -605.7 -15.54 0
 0 0 0 0 0 0 0 0 0 0
 835.3 -65.05 0 14820 166.3 6807 4297 0 16830 390.4 0
 196.3 221.1 0 166.3 110.4 -32.64 136.3 0 -407.4 -3.58 0
 1739 1649 0 6807 -32.64 15440 3474 0 15080 250.9 0
 2297 1079 0 4297 136.3 3474 11300 0 21210 -9.276 0
 0 0 0 0 0 0 0 0 0 0
 849 -605.7 0 16830 -407.4 15080 21210 0 2.602E+005 235.2 0
 1.411 -15.54 0 390.4 -3.58 250.9 -9.276 0 235.2 17.26 0
 0 0 0 0 0 0 0 0 0 33.94

m21

55
 64.71 6.19 6.447 129.8 21.45 199.9 78.11 0.08303 644.4 2.349 6.956
 57.5 9.124 15.6 101.8 29.29 111.8 85.62 0.4428 363.5 4.306 5.826
 3307 -122.8 27.8 2452 358.8 -687.7 172.3 -4.431 -1108 -51.5 0
 -122.8 83.25 24.19 40.85 -5.078 -163.2 33.59 0.2465 -852.4 -3.261 0
 27.8 24.19 243.3 -7.936 25.13 -106.3 -48.08 -0.5457 -720.1 -6.111 0
 2452 40.85 -7.936 10360 -140.1 796.2 -784.2 -7.21 -2515 -84.12 0

358.8 -5.078 25.13 -140.1 858.1 -127.7 117.9 0.1686 532.4 10.09 0
 -687.7 -163.2 -106.3 796.2 -127.7 12490 1617 -4.355 2478 51.97 0
 172.3 33.59 -48.08 -784.2 117.9 1617 7331 -0.7962 -155.6 -30.23 0
 -4.431 0.2465 -0.5457 -7.21 0.1686 -4.355 -0.7962 0.1961 -19.48 -0.00572 0
 -1108 -852.4 -720.1 -2515 532.4 2478 -155.6 -19.48 1.321E+005 -40.69 0
 -51.5 -3.261 -6.111 -84.12 10.09 51.97 -30.23 -0.00572 -40.69 18.54 0
 0 0 0 0 0 0 0 0 0 0 33.94

m22

87
 93.31 8.095 23.69 222.9 17 296.8 123.8 0.3875 966.6 5.102 6.956
 68.92 14.22 31.49 145.6 27.4 141.1 134.7 2.435 424.7 9.273 5.826
 4750 -64.7 -134.6 3371 232.3 116.7 1402 -5.875 6322 -23.01 0
 -64.7 202.3 -30.91 439 -44.82 -6.022 -180.1 -2.979 924.2 -12.93 0
 -134.6 -30.91 991.7 -449.3 115.6 -551.1 517.4 -1.457 -535 54.9 0
 3371 439 -449.3 21190 442.7 3349 3039 0.3545 7851 6.749 0
 232.3 -44.82 115.6 442.7 750.7 576.1 -250.9 3.403 -942.5 18.29 0
 116.7 -6.022 -551.1 3349 576.1 19920 722.3 -34.72 7372 9.161 0
 1402 -180.1 517.4 3039 -250.9 722.3 18140 -42.64 -343.2 134.9 0
 -5.875 -2.979 -1.457 0.3545 3.403 -34.72 -42.64 5.932 -72.4 6.911 0
 6322 924.2 -535 7851 -942.5 7372 -343.2 -72.4 1.804E+005 59.07 0
 -23.01 -12.93 54.9 6.749 18.29 9.161 134.9 6.911 59.07 85.99 0
 0 0 0 0 0 0 0 0 0 0 33.94

m23

87
 111 10.7 25.28 255.2 18.27 303.5 114.1 0.602 879.9 2.892 6.956
 65.2 18.53 42.82 170.1 31.5 162.7 123.9 5.611 495.7 4.563 5.826
 4251 -20.54 36.29 987.1 -119.1 -986.5 48.46 -2.927 6141 29.16 0
 -20.54 343.3 83.3 857.3 -118.5 150.7 130.8 -6.55 2581 9.638 0
 36.29 83.3 1833 3299 303.4 1184 991.8 -15.38 2292 -22.08 0
 987.1 857.3 3299 28940 -573.2 -525.8 2760 89.72 25550 37.26 0
 -119.1 -118.5 303.4 -573.2 991.9 174.2 507.2 -11.13 -1358 -17.25 0
 -986.5 150.7 1184 -525.8 174.2 26470 4272 9.129 12340 -25.24 0
 48.46 130.8 991.8 2760 507.2 4272 15340 23.63 16820 -14.13 0
 -2.927 -6.55 -15.38 89.72 -11.13 9.129 23.63 31.49 -41.72 6.145 0
 6141 2581 2292 25550 -1358 12340 16820 -41.72 2.457E+005 411.6 0
 29.16 9.638 -22.08 37.26 -17.25 -25.24 -14.13 6.145 411.6 20.82 0
 0 0 0 0 0 0 0 0 0 0 33.94

m24

108
 104.8 8.366 33.74 225.8 21.1 239.9 57.66 1.695 776.2 4.244 6.956
 80.7 15.05 51.53 188.9 33.63 147.6 94.84 12.52 494.3 8.636 5.826
 6512 -126.3 -432.5 2424 455.9 714.7 -199 0 11170 -112.2 0
 -126.3 226.4 49.63 -264.3 -47.57 -288.7 78.49 -14.32 -275.2 28.46 0
 -432.5 49.63 2656 1957 357 -342.3 542.5 3.872 1325 75.65 0
 2424 -264.3 1957 35680 971.8 -1171 3081 579.5 -280.1 86.45 0
 455.9 -47.57 357 971.8 1131 431.8 92.49 43.38 4587 -3.194 0
 714.7 -288.7 -342.3 -1171 431.8 21790 1778 208.9 17870 15.3 0
 -199 78.49 542.5 3081 92.49 1778 8995 -98.58 7454 13.1 0
 0 -14.32 3.872 579.5 43.38 208.9 -98.58 156.8 -786.1 9.733 0
 11170 -275.2 1325 -280.1 4587 17870 7454 -786.1 2.443E+005 187.8 0
 -112.2 28.46 75.65 86.45 -3.194 15.3 13.1 9.733 187.8 74.58 0
 0 0 0 0 0 0 0 0 0 0 33.94

m25

461
 101.4 11.02 54 217.2 25.67 231 98.12 0.8995 356.6 7.092 6.956
 84.64 16.98 70.44 158.2 38.66 150.7 135 6.525 387.7 12.91 5.826
 7164 -228.5 107.3 1326 -75.27 -1058 -936.7 -19.33 -328.2 -12.02 0
 -228.5 288.2 -15.54 179.9 -78.1 -43.48 11.46 -3.655 -276.4 4.821 0
 107.3 -15.54 4962 2061 92.6 -594.3 1388 25.28 -27.31 149.1 0
 1326 179.9 2061 25020 330.3 2145 2541 24.77 3926 118.4 0
 -75.27 -78.1 92.6 330.3 1495 262.1 370.5 10.09 1874 17.97 0
 -1058 -43.48 -594.3 2145 262.1 22700 2176 -63.9 13030 -21.39 0

```

-95      11.46 1380 2541 370.5 2176 18220 -61.64 4605 245.6 0
-19.33 -3.655 25.28 24.77 10.09 -63.9 -61.64 42.57 -161.9 0.2527 0
-328.2 -276.4 -27.31 3926 1874 13030 4605 -161.9 1.503E+005 -30.03 0
-12.02 4.821 149.1 118.4 17.97 -21.39 245.6 0.2527 -30.03 166.6 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m26

```

4
111.9 7.837 59.88 465 49.45 292 108.6 0 671.1 11.77 6.956
88.1 9.292 38.51 241.1 43.27 177 155.7 0 325.9 18.35 5.826
7762 470.7 2511 -19310 -1460 -13800 -12060 0 -13750 -1350 0
470.7 86.35 19.33 -1813 -303.5 -501.6 -680.2 0 -2271 -89.35 0
2511 19.33 1483 -5813 -516.6 -3905 -5435 0 -5233 -611.3 0
-19310 -1813 -5813 58130 7563 26790 33460 0 61910 3991 0
-1460 -303.5 -516.6 7563 1872 -620.2 4164 0 14000 566.1 0
-13800 -501.6 -3905 26790 -620.2 31320 16590 0 1153 1653 0
-12060 -680.2 -5435 33460 4164 16590 24260 0 35980 2835 0
0 0 0 0 0 0 0 0 0 0 0
-13750 -2271 -5233 61910 14000 1153 35980 0 1.062E+005 4736 0
-1350 -89.35 -611.3 3991 566.1 1653 2835 0 4736 336.7 0
0 0 0 0 0 0 0 0 0 0 33.94

```

ss

m1

34
21.92 9.082 0.00 175 7.127 92.95 90.57 0 955.7 0.6822 6.956
40.14 14.17 0.00 203.6 24.83 144.8 129.5 0 730.9 3.321 5.826
1611 271.9 -38.79 -457.7 -40.86 145.3 961.5 0 6249 -8.531 0
271.9 200.9 15.27 718.7 63.35 192.9 444.1 0 2517 -0.706 0
-38.79 15.27 101.4 -102.5 15.5 -138.5 -161.6 0 706.4 -1.204 0
-457.7 718.7 -102.5 41470 1755 12710 7963 0 -27540 314.5 0
-40.86 63.35 15.5 1755 616.6 2125 -93.24 0 1343 6.185 0
145.3 192.9 -138.5 12710 2125 20970 693.7 0 21170 -16.35 0
961.5 444.1 -161.6 7963 -93.24 693.7 16770 0 13630 -38.27 0
0 0 0 0 0 0 0 0 0 0
6249 2517 706.4 -27540 1343 21170 13630 0 5.342E+005 -446.6 0
-8.531 -0.706 -1.204 314.5 6.185 -16.35 -38.27 0 -446.6 11.03 0
0 0 0 0 0 0 0 0 0 0 33.94

m2

171
61.75 8.904 0.00 149.4 26.81 239.6 63.68 0 728.6 2.974 6.956
45.65 12.39 0.00 137.9 35.9 156.9 99.91 0 418.1 6.042 5.826
2084 -3.96 104.4 459.4 147.5 1310 -59.29 0 362.6 -24.82 0
-3.96 153.6 -5.334 88.87 71.64 -235.3 105.3 0 -134.7 -1.423 0
104.4 -5.334 513.1 43.73 88.64 -131.5 43 0 -672.5 -0.5474 0
459.4 88.87 43.73 19010 -108.9 1947 385.7 0 9743 0 0
147.5 71.64 88.64 -108.9 1289 -568.8 111.2 0 -1141 -2.386 0
1310 -235.3 -131.5 1947 -568.8 24610 313.5 0 -787.2 -45.5 0
-59.29 105.3 43 385.7 111.2 313.5 9982 0 -292.4 272.2 0
0 0 0 0 0 0 0 0 0 0
362.6 -134.7 -672.5 9743 -1141 -787.2 -292.4 0 1.748E+005 50.53 0
-24.82 -1.423 -0.5474 0 -2.386 -45.5 272.2 0 50.53 36.5 0
0 0 0 0 0 0 0 0 0 0 33.94

m3

209
92.21 10.58 0.00 203.7 22.47 329.9 87.73 0.4359 941.6 3.427 6.956
53.3 15.09 0.00 130.3 32.35 169.5 108.6 2.65 471.5 6.638 5.826
2840 -26.55 417.9 1521 120.7 1463 -393.7 -1.977 5252 -0.7076 0
-26.55 227.8 -40.32 45.25 47.37 -271.2 101.7 -0.68 -227.8 -10.12 0
417.9 -40.32 1856 320 243.9 1541 608.4 4.567 1422 11.73 0
1521 45.25 320 16990 282.5 2209 2534 15.2 13150 56.24 0
120.7 47.37 243.9 282.5 1047 120.6 629.1 6.001 1464 1.718 0
1463 -271.2 1541 2209 120.6 28720 1086 44.91 16460 51.75 0
-393.7 101.7 608.4 2534 629.1 1086 11800 68.51 5378 60.57 0
-1.977 -0.68 4.567 15.2 6.001 44.91 68.51 7.022 -143.7 -1.442 0
5252 -227.8 1422 13150 1464 16460 5378 -143.7 2.223E+005 -9.39 0
-0.7076 -10.12 11.73 56.24 1.718 51.75 60.57 -1.442 -9.39 44.06 0
0 0 0 0 0 0 0 0 0 0 33.94

m4

194
116.8 12.83 0.00 246.3 22.47 378 81.15 0.2422 1056 4.04 6.956
66.51 18.2 0.00 161.1 36.64 203.9 110.4 2.207 589.1 7.141 5.826
4424 -300.2 1082 1254 31.68 -81.38 -770.7 0.1468 -1881 -59.37 0
-300.2 331.3 97.86 255.1 70.03 55.67 405.8 -1.286 793.4 15.21 0
1082 97.86 4752 1399 174.3 2600 395.6 10.95 6010 0 0
1254 255.1 1399 25960 295.2 -1741 88.91 -12.09 28760 111.6 0
31.68 70.03 174.3 295.2 1343 -1151 -64.7 0.3235 431.7 10.2 0
-81.38 55.67 2600 -1741 -1151 41580 1530 -22.95 29310 141.2 0
-770.7 405.8 395.6 88.91 -64.7 1530 12180 1.218 10210 20.49 0
0.1468 -1.286 10.95 -12.09 0.3235 -22.95 1.218 4.872 -107.9 -0.7407 0
-1881 793.4 6010 28760 431.7 29310 10210 -107.9 3.47E+005 168.3 0
-59.37 15.21 0 111.6 10.2 141.2 20.49 -0.7407 168.3 50.99 0
0 0 0 0 0 0 0 0 0 0 33.94

m5
212
164.8 15.28 0.00 268.9 34.79 376.2 70.37 1.391 1019 4.559 6.956
108 21.89 0.00 186.3 50.83 216.3 104.2 9.652 650.9 7.092 5.826
11670 -439.9 106.4 3905 -186.7 654.1 -1452 -164.7 9000 -0.7661 0
-439.9 479.3 -17.25 85.66 -67.88 108.9 171 34.66 -484.5 -0.4658 0
106.4 -17.25 4313 1419 253.7 -28.4 643 -32.33 4574 49.36 0
3905 85.66 1419 34710 700.7 684.9 892.7 122.3 15890 112.3 0
-186.7 -67.88 253.7 700.7 2583 -835.4 -397.1 -23.55 -5028 20.18 0
654.1 108.9 -28.4 684.9 -835.4 46770 -540.6 -18.79 40260 -9.202 0
-1452 171 643 892.7 -397.1 -540.6 10850 -56.3 14640 61.31 0
-164.7 34.66 -32.33 122.3 -23.55 -18.79 -56.3 93.16 -433.5 -6.023 0
9000 -484.5 4574 15890 -5028 40260 14640 -433.5 4.236E+005 447.7 0
-0.7661 -0.4658 49.36 112.3 20.18 -9.202 61.31 -6.023 447.7 50.29 0
0 0 0 0 0 0 0 0 0 33.94

m6
777
173.7 17.09 0.00 281 49.38 312.7 77.28 2.311 455.5 7.676 6.956
118 26.59 0.00 197.3 59.76 206.4 126.7 12.17 466 13.17 5.826
13920 -593 -144.4 2235 472.4 194.9 -1285 -112 7093 -43.5 0
-593 707.1 102.3 257.1 77.87 -527 266.1 15.86 -570 13.65 0
-144.4 102.3 7642 1121 -261.2 0 1495 10.64 692.6 51.79 0
2235 257.1 1121 38920 1085 1955 949.6 93.64 10850 129.9 0
472.4 77.87 -261.2 1085 3572 653.9 -249.8 -4.364 1560 45.64 0
194.9 -527 0 1955 653.9 42620 1569 -108 15200 119.6 0
-1285 266.1 1495 949.6 -249.8 1569 16040 -21.58 2361 155.1 0
-112 15.86 10.64 93.64 -4.364 -108 -21.58 148.1 -521.8 2.564 0
7093 -570 692.6 10850 1560 15200 2361 -521.8 2.172E+005 300.7 0
-43.5 13.65 51.79 129.9 45.64 119.6 155.1 2.564 300.7 173.3 0
0 0 0 0 0 0 0 0 0 33.94

m7
48
22.24 8.198 0.00 160.8 8.459 74.66 134.1 0 972 1.23 6.956
30.59 13.56 0.00 148.1 18.6 93.42 143.4 0 528.1 5.229 5.826
935.8 129 47.18 1173 79.67 214.3 43.87 0 1745 -12.8 0
129 183.9 -5.339 590.4 -28 221.7 560 0 1948 -6.594 0
47.18 -5.339 269.2 473.8 70.8 441.4 331.7 0 -658.4 21.02 0
1173 590.4 473.8 21930 427 1923 12810 0 4770 72.01 0
79.67 -28 70.8 427 346.1 408.4 773.6 0 -1768 -10.02 0
214.3 221.7 441.4 1923 408.4 8727 5426 0 986.7 70.34 0
43.87 560 331.7 12810 773.6 5426 20560 0 -8103 34.49 0
0 0 0 0 0 0 0 0 0 0
1745 1948 -658.4 4770 -1768 986.7 -8103 0 2.789E+005 -171.2 0
-12.8 -6.594 21.02 72.01 -10.02 70.34 34.49 0 -171.2 27.34 0
0 0 0 0 0 0 0 0 0 33.94

m8
143
60.82 9.756 0.00 142.5 25.88 213.7 85.21 0.04151 780.8 3.444 6.956
39.73 13.38 0.00 109.2 31.55 122.4 111 0.3252 409.9 6.111 5.826
1579 -31.89 40.65 486 -132.9 267.5 150 -1.253 3111 -26.23 0
-31.89 179 9.58 94.95 35.03 -137.6 74.25 0.2828 43.87 -13.08 0
40.65 9.58 418.6 37.98 10.33 -137.8 320.3 -0.2528 -972.9 -1.25 0
486 94.95 37.98 11930 413.5 574.9 860.7 3.977 4611 46.05 0
-132.9 35.03 10.33 413.5 995.5 -378.5 -98.07 -1.006 -814.8 26.22 0
267.5 -137.6 -137.8 574.9 -378.5 14990 883.3 -2.269 4015 -33.67 0
150 74.25 320.3 860.7 -98.07 883.3 12320 -3.321 773.6 -9.497 0
-1.253 0.2828 -0.2528 3.977 -1.006 -2.269 -3.321 0.1058 7.865 -0.06757 0
3111 43.87 -972.9 4611 -814.8 4015 773.6 7.865 1.68E+005 102.7 0
-26.23 -13.08 -1.25 46.05 26.22 -33.67 -9.497 -0.06757 102.7 37.35 0
0 0 0 0 0 0 0 0 0 33.94

m9
219

87.51 12.88 0.00 182.7 19.5 297.4 101 0.2837 910.3 2.958 6.956
 59.02 17.83 0.00 118.9 33.35 141.2 110.3 3.826 386.9 5.037 5.826
 3484 -197.8 205.5 2267 23.62 -8.336 768.2 -12.65 2124 -2.378 0
 -197.8 317.8 18.29 -76.3 -43.99 193.9 -76.68 0.7503 -420.8 -3.143 0
 205.5 18.29 1343 605.7 -1.222 -51.76 606.3 5.749 3559 -6.092 0
 2267 -76.3 605.7 14140 -3.965 100.8 773.7 -0.4549 2162 -25.75 0
 23.62 -43.99 -1.222 -3.965 1112 -32.97 404.6 34.58 1200 -3.863 0
 -8.336 193.9 -51.76 100.8 -32.97 19950 -327.1 -22.7 4590 27.03 0
 768.2 -76.68 606.3 773.7 404.6 -327.1 12160 24.48 4993 37.22 0
 -12.65 0.7503 5.749 -0.4549 34.58 -22.7 24.48 14.64 109.6 -0.7131 0
 2124 -420.8 3559 2162 1200 4590 4993 109.6 1.497E+005 115 0
 -2.378 -3.143 -6.092 -25.75 -3.863 27.03 37.22 -0.7131 115 25.37 0
 0 0 0 0 0 0 0 0 0 0 33.94

m10

223
 101.9 13.03 0.00 210 19.49 302.4 75.66 0.7749 800.4 4.203 6.956
 64.36 16.36 0.00 136.1 31.26 167.1 94.31 4.511 375 7.01 5.826
 4143 -108.5 467.9 665.8 -28.16 -1140 -315.7 -29.91 -627.5 -9.926 0
 -108.5 267.7 -98.45 118 23.53 -142.2 117.3 -7.75 18.41 -7.455 0
 467.9 -98.45 3928 1382 207.6 774.8 691.6 47.22 1316 7.468 0
 665.8 118 1382 18530 1042 90.96 2182 -38.68 5410 83.96 0
 -28.16 23.53 207.6 1042 977 -245.4 100.2 -8.037 164.1 11.61 0
 -1140 -142.2 774.8 90.96 -245.4 27910 393.9 -37.68 4574 66.76 0
 -315.7 117.3 691.6 2182 100.2 393.9 8895 -15.74 4704 33.72 0
 -29.91 -7.75 47.22 -38.68 -8.037 -37.68 -15.74 20.35 113.3 3.447 0
 -627.5 18.41 1316 5410 164.1 4574 4704 113.3 1.406E+005 -65.72 0
 -9.926 -7.455 7.468 83.96 11.61 66.76 33.72 3.447 -65.72 49.14 0
 0 0 0 0 0 0 0 0 0 0 33.94

m11

205
 98.4 15.13 0.00 214.3 26.44 251.4 63.71 1.204 663.3 5.278 6.956
 62.11 19.09 0.00 170.4 40.82 156.4 88.73 7.417 462.2 8.615 5.826
 3858 -257.2 -153.9 1757 238.4 398.4 -308.6 36.4 1062 -50.84 0
 -257.2 364.3 14.19 380.5 -41.3 -376.2 130.4 -13.87 -599.9 -7.564 0
 -153.9 14.19 6824 2970 141.6 -646.2 2756 -15.93 2711 -9.252 0
 1757 380.5 2970 29040 285.2 3252 2177 22.75 23550 -132.1 0
 238.4 -41.3 141.6 285.2 1667 734.5 94.18 -13.32 18.87 20.75 0
 398.4 -376.2 -646.2 3252 734.5 24480 1735 -23.21 12000 -91.65 0
 -308.6 130.4 2756 2177 94.18 1735 7873 -47.39 4101 -5.351 0
 36.4 -13.87 -15.93 22.75 -13.32 -23.21 -47.39 55.01 -281.1 1.022 0
 1062 -599.9 2711 23550 18.87 12000 4101 -281.1 2.136E+005 159.3 0
 -50.84 -7.564 -9.252 -132.1 20.75 -91.65 -5.351 1.022 159.3 74.22 0
 0 0 0 0 0 0 0 0 0 0 33.94

m12

1108
 104.5 14.99 0.00 213.5 33.07 217.7 70.26 2.013 318.1 8.207 6.956
 79.03 22.13 0.00 156.2 43.62 153.1 111.1 11.49 314.2 13.9 5.826
 6246 -272.9 410.8 963.1 110.3 -108.9 -447.9 -37.22 -496.7 -59.33 0
 -272.9 489.8 89.26 210.9 -33.79 -138.9 -44.27 -11.69 -13.91 -6.153 0
 410.8 89.26 8033 28 156.4 -713.5 1394 51.47 -1098 82.23 0
 963.1 210.9 28 24410 204.4 1818 1458 -75.36 2258 126 0
 110.3 -33.79 156.4 204.4 1902 213.7 24.23 -2.505 849.8 15.76 0
 -108.9 -138.9 -713.5 1818 213.7 23440 -442.3 -22.86 7120 -25.54 0
 -447.9 -44.27 1394 1458 24.23 -442.3 12350 -35.74 2060 225.5 0
 -37.22 -11.69 51.47 -75.36 -2.505 -22.86 -35.74 131.9 -191.3 -0.1597 0
 -496.7 -13.91 -1098 2258 849.8 7120 2060 -191.3 98750 -91.74 0
 -59.33 -6.153 82.23 126 15.76 -25.54 225.5 -0.1597 -91.74 193.2 0
 0 0 0 0 0 0 0 0 0 0 33.94

m13

12
 111.2 13.67 0.00 202.6 33.82 255.8 72.75 0 566.5 8.333 6.956
 83.63 13.72 0.00 114.6 35.11 213.5 71.55 0 390 11.01 5.826

699. 5.8 -890.6 939.5 878.1 -6661 -1861 0 -3360 -409 0
 355.8 188.3 122 -124.3 327.7 -512.8 428.1 0 321.2 -22.52 0
 -890.6 122 1533 -426.3 453.7 3193 137.3 0 -1863 -142.7 0
 939.5 -124.3 -426.3 13140 -1010 1052 1427 0 -8986 537.8 0
 878.1 327.7 453.7 -1010 1233 -419.9 658.3 0 -397.2 -116 0
 -6661 -512.8 3193 1052 -419.9 45600 -1681 0 22900 216.4 0
 -1861 428.1 137.3 1427 658.3 -1681 5120 0 2428 415.3 0
 0 0 0 0 0 0 0 0 0 0
 -3360 321.2 -1863 -8986 -397.2 22900 2428 0 1.521E+005 -429.5 0
 -409 -22.52 -142.7 537.8 -116 216.4 415.3 0 -429.5 121.3 0
 0 0 0 0 0 0 0 0 0 0 33.94

m14

15
 30.36 14.29 0.00 197.1 9.61 142.2 143.8 0 929.7 0.06536 6.956
 32.64 19.39 0.00 154.5 16.02 132 189.5 0 534.1 0.2451 5.826
 1065 83.56 147.6 201.7 125.5 504 68.02 0 3783 1.248 0
 83.56 376.1 -15.79 1474 57.48 880.5 551.2 0 2724 2.738 0
 147.6 -15.79 38.04 -105.8 25.4 -64.3 -268.8 0 369 -0.1179 0
 201.7 1474 -105.8 23860 -930.5 265 11240 0 2145 -6.02 0
 125.5 57.48 25.4 -930.5 256.7 606.8 -1111 0 299.5 2.584 0
 504 880.5 -64.3 265 606.8 17420 6501 0 -25590 14.75 0
 68.02 551.2 -268.8 11240 -1111 6501 35890 0 -39260 -4.736 0
 0 0 0 0 0 0 0 0 0 0
 3783 2724 369 2145 299.5 -25590 -39260 0 2.853E+005 -1.309 0
 1.248 2.738 -0.1179 -6.02 2.584 14.75 -4.736 0 -1.309 0.06007 0
 0 0 0 0 0 0 0 0 0 0 33.94

m15

62
 55.45 6.658 0.00 153.6 23.67 211.6 107.8 0 741.2 3.162 6.956
 51.83 15.65 0.00 131.9 35.14 106 149.5 0 487 5.613 5.826
 2686 -3.244 162.5 -437.3 -293.2 982.9 2603 0 -807.7 -48 0
 -3.244 244.8 25.41 -113.5 -63.78 -8.289 -189.5 0 -144.8 9.221 0
 162.5 25.41 156 248.7 53.11 107.2 22.41 0 304.2 1.052 0
 -437.3 -113.5 248.7 17390 495.8 3283 -197.1 0 -642.2 -51.8 0
 -293.2 -63.78 53.11 495.8 1235 458 136.6 0 -3509 21.11 0
 982.9 -8.289 107.2 3283 458 11230 4229 0 -9082 92.18 0
 2603 -189.5 22.41 -197.1 136.6 4229 22350 0 0 62.09 0
 0 0 0 0 0 0 0 0 0 0
 -807.7 -144.8 304.2 -642.2 -3509 -9082 0 0 2.372E+005 -358.1 0
 -48 9.221 1.052 -51.8 21.11 92.18 62.09 0 -358.1 31.5 0
 0 0 0 0 0 0 0 0 0 0 33.94

m16

76
 95.71 9.662 0.00 220.9 14.16 322.8 106 0.1315 1039 3.092 6.956
 56.59 16.66 0.00 162.5 27.33 188.4 125.6 1.128 449.2 6.471 5.826
 3202 -163.1 -1.981 1886 247.4 1034 1941 -7.786 7880 7.689 0
 -163.1 277.5 -71.13 -154.3 -5.007 -442.4 -251.1 -1.259 -823.1 -5.497 0
 -1.981 -71.13 1225 341.4 311.8 263.7 -136.3 2.921 5613 92.63 0
 1886 -154.3 341.4 26420 -235.4 2327 -3819 -5.499 14160 -122 0
 247.4 -5.007 311.8 -235.4 746.8 -5.148 -391.4 -1.849 3818 63.84 0
 1034 -442.4 263.7 2327 -5.148 35490 4308 -25.49 28940 113.4 0
 1941 -251.1 -136.3 -3819 -391.4 4308 15790 -13.89 10610 -112.2 0
 -7.786 -1.259 2.921 -5.499 -1.849 -25.49 -13.89 1.272 10.13 -0.4087 0
 7880 -823.1 5613 14160 3818 28940 10610 10.13 2.018E+005 674.4 0
 7.689 -5.497 92.63 -122 63.84 113.4 -112.2 -0.4087 674.4 41.87 0
 0 0 0 0 0 0 0 0 0 0 33.94

m17

111
 120.6 13.89 0.00 296.5 27.61 429.7 72.7 0.173 1085 3.317 6.956
 74.65 20.23 0.00 187.2 38.58 326.6 87.78 1.834 574.9 5.976 5.826
 5573 -303.6 25.02 -1300 -288 1487 308 -4.654 -4463 59.78 0
 -303.6 409.3 -17.63 572 -22.64 -1335 -330.3 -2.448 965.3 -0.9673 0

23.02 -17.63 1123 -489.4 23.27 -1259 588.3 6.697 3564 27.24 0
-1300 572 -489.4 35050 -216.7 -6970 1972 19.91 430.5 -110.8 0
-288 -22.64 23.27 -216.7 1488 -642.6 -84.67 6.508 465.7 25.13 0
1487 -1335 -1259 -6970 -642.6 1.067E+005 3383 4.192 26660 150.3 0
308 -330.3 588.3 1972 -84.67 3383 7706 5.955 4390 75.55 0
-4.654 -2.448 6.697 19.91 6.508 4.192 5.955 3.362 -51.65 -0.5808 0
-4463 965.3 3564 430.5 465.7 26660 4390 -51.65 3.305E+005 374.5 0
59.78 -0.9673 27.24 -110.8 25.13 150.3 75.55 -0.5808 374.5 35.72 0
0 0 0 0 0 0 0 0 0 0 33.94

m18

103

169.9 10.56 0.00 322.3 38.69 408.4 59.57 0.8026 1064 5.11 6.956
94.23 25.8 0.00 202.7 57.63 228.5 83.55 4.186 645.2 9.232 5.826
8880 -14.58 321.7 2847 429 4306 -62.98 -14.2 304 149.6 0
-14.58 665.4 15.22 -177.8 -34.19 341.8 -112.1 9.286 1248 33.1 0
321.7 15.22 1777 2239 206.5 789.7 -140.9 22.05 924.6 25.29 0
2847 -177.8 2239 41110 1846 5651 2490 37.34 12300 41.18 0
429 -34.19 206.5 1846 3321 -368.7 765.6 42.22 -2528 0 0
4306 341.8 789.7 5651 -368.7 52200 1794 113.8 30810 293.2 0
-62.98 -112.1 -140.9 2490 765.6 1794 6980 4.896 6738 47.82 0
-14.2 9.286 22.05 37.34 42.22 113.8 4.896 17.52 -124.2 -4.135 0
304 1248 924.6 12300 -2528 30810 6738 -124.2 4.163E+005 446.7 0
149.6 33.1 25.29 41.18 0 293.2 47.82 -4.135 446.7 85.23 0
0 0 0 0 0 0 0 0 0 0 33.94

m19

317

165.4 14.36 0.00 305.2 45.4 331.8 80.06 1.211 498.4 7.9 6.956
101.7 24.01 0.00 191.5 53.52 215 131.2 9.216 466.3 13.27 5.826
10340 -297.8 257.3 1324 146.9 1946 -427.1 -16.87 2703 43.17 0
-297.8 576.4 159 436.8 97.65 -562.6 0 11.95 -414.3 -0.3185 0
257.3 159 5540 1867 330.6 1216 976.9 61.05 2013 90.86 0
1324 436.8 1867 36670 399.7 6299 5730 15.88 4197 190.6 0
146.9 97.65 330.6 399.7 2864 -414.2 98.34 -15.78 -2246 -10.65 0
1946 -562.6 1216 6299 -414.2 46220 4148 -75.29 16640 516.3 0
-427.1 0 976.9 5730 98.34 4148 17220 -74.99 4958 355.2 0
-16.87 11.95 61.05 15.88 -15.78 -75.29 -74.99 84.94 47.28 2.446 0
2703 -414.3 2013 4197 -2246 16640 4958 47.28 2.175E+005 792 0
43.17 -0.3185 90.86 190.6 -10.65 516.3 355.2 2.446 792 176 0
0 0 0 0 0 0 0 0 0 0 33.94

m20

23

17.26 10.27 0.00 65.79 5.485 29.98 95.28 0.946.5 0.527 6.956
28.46 20.3 0.00 92.63 14.25 52.6 126.4 0 778.9 2.382 5.826
809.8 380.6 67.97 1165 21.09 501.5 2255 0 443.3 -5.49 0
380.6 412 -22.41 1075 13.02 55.52 2065 0 1075 -3.722 0
67.97 -22.41 100.7 -31.61 53.5 369.6 -208.1 0 1735 0.2629 0
1165 1075 -31.61 8580 -128.1 -29.23 6990 0 5628 -21.18 0
21.09 13.02 53.5 -128.1 203.2 596.8 490.1 0 899.3 28.99 0
501.5 55.52 369.6 -29.23 596.8 2767 1310 0 4016 78.68 0
2255 2065 -208.1 6990 490.1 1310 15980 0 9059 58.1 0
0 0 0 0 0 0 0 0 0 0 0
443.3 1075 1735 5628 899.3 4016 9059 0 6.067E+005 -191.1 0
-5.49 -3.722 0.2629 -21.18 28.99 78.68 58.1 0 -191.1 5.672 0
0 0 0 0 0 0 0 0 0 0 33.94

m21

55

65.18 7.238 0.00 142 15.7 175.1 64.04 1.764 727.3 4.914 6.956
52.33 12.46 0.00 109.4 19.46 99.62 92.85 8.282 407.5 6.936 5.826
2738 -78.9 89.08 -45.8 123.2 -823.7 -855.2 32.51 170.6 42.11 0
-78.9 155.3 -4.741 -8.179 -49.22 -70.76 281.1 -3.818 -126.9 4.408 0
89.08 -4.741 401 595.9 -91.57 -109.7 7.438 -20.07 873.2 6.39 0
-45.8 -8.179 595.9 11970 -227.8 1123 30.47 -135.9 7089 3.035 0

123 -49.22 -91.57 -227.8 378.6 436.2 -265.6 -9.669 7.93 14.0 0
 -823.7 -70.76 -109.7 1123 436.2 9925 1656 100.7 9257 77.39 0
 -855.2 281.1 7.438 30.47 -265.6 1656 8621 -54.6 454.1 48.95 0
 32.51 -3.818 -20.07 -135.9 -9.669 100.7 -54.6 68.59 178.9 -3.447 0
 170.6 -126.9 873.2 7089 7.93 9257 454.1 178.9 1.661E+005 955.4 0
 42.11 4.408 6.39 3.035 14.71 77.39 48.95 -3.447 955.4 48.11 0
 0 0 0 0 0 0 0 0 0 0 33.94

m22

90
 83.73 9.307 0.00 210.8 17.31 276.9 82.5 0.08995 898.7 4.114 6.956
 55.26 14.49 0.00 133.6 32.72 143.8 104.7 0.6089 425.4 7.373 5.826
 3054 -132.1 178.4 -502.2 59.66 508.6 352.9 -4.273 -352.6 31.78 0
 -132.1 209.8 71.5 333 51.18 -172.9 -87.96 -0.5998 55.47 -7.691 0
 178.4 71.5 963.5 1012 74.13 -111.6 175.5 0.8316 -79.23 -7.095 0
 -502.2 333 1012 17860 542.1 3344 209.9 -2.848 5231 -87.7 0
 59.66 51.18 74.13 542.1 1070 -334 -71.93 -0.5577 167 -3.618 0
 508.6 -172.9 -111.6 3344 -334 20680 195.7 6.829 2753 -232.2 0
 352.9 -87.96 175.5 209.9 -71.93 195.7 10960 0.8924 4810 -57.9 0
 -4.273 -0.5998 0.8316 -2.848 -0.5577 6.829 0.8924 0.3707 -28.49 -0.1751 0
 -352.6 55.47 -79.23 5231 167 2753 4810 -28.49 1.81E+005 517.6 0
 31.78 -7.691 -7.095 -87.7 -3.618 -232.2 -57.9 -0.1751 517.6 54.37 0
 0 0 0 0 0 0 0 0 0 0 33.94

m23

100
 100.2 9.363 0.00 200 18.43 300.8 59.67 0.07611 827.4 3.476 6.956
 56.58 12.64 0.00 145.2 31.72 142 84.73 0.7818 406.7 6.63 5.826
 3201 -26.46 583.9 427.1 73.58 297.2 19.17 6.503 -3774 43.89 0
 -26.46 159.7 -8.012 489.8 -38.08 -303.2 -133.8 -0.7411 395.7 -3.184 0
 583.9 -8.012 1240 828.3 90.49 110 170.1 -2.203 -300.8 40.16 0
 427.1 489.8 828.3 21070 -32.23 1834 1353 -14.64 1358 -36.57 0
 73.58 -38.08 90.49 -32.23 1006 -279.2 129 -1.463 -838.5 26.29 0
 297.2 -303.2 110 1834 -279.2 20150 1876 -15.43 16450 239.1 0
 19.17 -133.8 170.1 1353 129 1876 7179 -1.325 2136 87.63 0
 6.503 -0.7411 -2.203 -14.64 -1.463 -15.43 -1.325 0.6113 -49.6 -0.2747 0
 -3774 395.7 -300.8 1358 -838.5 16450 2136 -49.6 1.654E+005 -175.3 0
 43.89 -3.184 40.16 -36.57 26.29 239.1 87.63 -0.2747 -175.3 43.96 0
 0 0 0 0 0 0 0 0 0 0 33.94

m24

93
 106 10.36 0.00 214.6 12.46 305.5 52.21 1.744 628.9 3.803 6.956
 75.64 16.27 0.00 184.9 23.68 185 78.76 10 487.4 9.771 5.826
 5721 -200.6 109.6 1203 263.3 2239 363.4 -36.32 1364 -10.35 0
 -200.6 264.8 -7.857 -102.3 -55.88 -3.01 -101.3 6.675 705.9 -4.452 0
 109.6 -7.857 2878 684.5 47 -466.4 215.5 109.5 2641 -18.35 0
 1203 -102.3 684.5 34190 -372.2 4549 2024 53.65 13070 -189.7 0
 263.3 -55.88 47 -372.2 560.8 -573.9 -210.8 0.4738 -450.2 -0.6942 0
 2239 -3.01 -466.4 4549 -573.9 34220 2273 -111 25700 -52.42 0
 363.4 -101.3 215.5 2024 -210.8 2273 6203 36.25 5912 -37.71 0
 -36.32 6.675 109.5 53.65 0.4738 -111 36.25 100.1 604.7 0.6843 0
 1364 705.9 2641 13070 -450.2 25700 5912 604.7 2.376E+005 138.1 0
 -10.35 -4.452 -18.35 -189.7 -0.6942 -52.42 -37.71 0.6843 138.1 95.48 0
 0 0 0 0 0 0 0 0 0 0 33.94

m25

373
 104.6 11.73 0.00 213.1 28.37 222.8 60.21 1.169 362.7 5.944 6.956
 69.61 19.75 0.00 150.5 41.83 140.2 94.49 6.67 324 9.788 5.826
 4846 -258.4 195.8 1519 26.21 175.6 256.5 -2.786 1218 -12.95 0
 -258.4 389.9 187.4 -59.42 -48.74 94.1 126.9 -1.185 -281.5 -26.28 0
 195.8 187.4 7724 529 198.5 -1035 1611 -2.345 598 19.78 0
 1519 -59.42 529 22640 -107 1729 2033 -38.14 1073 -86.89 0
 26.21 -48.74 198.5 -107 1750 129 340 8.371 1206 6.961 0
 175.6 94.1 -1035 1729 129 19650 781.5 -31.79 5722 27.44 0

-2.786 -1.185 -2.345 -38.14 8.371 -31.79 -24.58 44.49 -19.45 -3.46 0
 1218 -281.5 598 1073 1206 5722 30.62 -19.45 1.05E+005 190.3 0
 -12.95 -26.28 19.78 -86.89 6.961 27.44 23.12 -3.46 190.3 95.8 0
 0 0 0 0 0 0 0 0 0 0 33.94

m26

3
 69.45 9.512 0.00 209.7 118.2 177.6 71.98 0 651.8 8.913 6.956
 56.6 16.48 0.00 62.5 106.2 114 124.7 0 84.69 5.801 5.826
 3203 -811.4 -1896 -1404 5569 3838 -6140 0 1232 213.7 0
 -811.4 271.6 634.4 -111.2 -1086 -227.2 2055 0 353.1 -18.35 0
 -1896 634.4 1482 -259.9 -2538 -530.9 4801 0 824.9 -42.88 0
 -1404 -111.2 -259.9 3906 -4731 -6938 -841.7 0 -5235 -346.2 0
 5569 -1086 -2538 -4731 11270 10320 -8221 0 5413 547.4 0
 3838 -227.2 -530.9 -6938 10320 12990 -1720 0 8977 659.1 0
 -6140 2055 4801 -841.7 -8221 -1720 15550 0 2672 -138.9 0
 0 0 0 0 0 0 0 0 0 0 0
 1232 353.1 824.9 -5235 5413 8977 2672 0 7173 442.6 0
 213.7 -18.35 -42.88 -346.2 547.4 659.1 -138.9 0 442.6 33.65 0
 0 0 0 0 0 0 0 0 0 0 33.94

?

m1

19

```

23.98 14.13 4.832 163.5 8.81 59.11 104 0 796.1 0 6.956
44.24 21.01 15.1 178 21.63 66.1 95.57 0 643.4 0 5.826
1957 453.6 40.08 5582 303.3 643.3 -575 0 -6831 0 0
453.6 441.6 32.68 2024 169.1 1095 737.1 0 -2271 0 0
40.08 32.68 228 88.7 77.1 401.3 -194.8 0 1205 0 0
5582 2024 88.7 31680 635.3 4989 5665 0 -13630 0 0
303.3 169.1 77.1 635.3 468 344.6 204.7 0 2547 0 0
643.3 1095 401.3 4989 344.6 4369 2521 0 6422 0 0
-575 737.1 -194.8 5665 204.7 2521 9135 0 1414 0 0
0 0 0 0 0 0 0 0 0 0
-6831 -2271 1205 -13630 2547 6422 1414 0 4.14E+005 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 33.94

```

m2

115

```

68.96 9.18 18.8 157.5 35.13 216.7 85.39 0 742.1 4.081 6.956
46.99 13.6 30.2 102.9 38.97 115.2 114.8 0 444.7 6.377 5.826
2208 -11.5 -29.8 570.8 -54.94 -389.8 -528.7 0 1003 -35.96 0
-11.5 185 -32.04 -298.2 37.11 -208.4 213.9 0 266.1 -7.893 0
-29.8 -32.04 912.2 236.3 61.21 191.4 398.8 0 -1370 3.274 0
570.8 -298.2 236.3 10600 248.7 3012 1572 0 -6592 -40.04 0
-54.94 37.11 61.21 248.7 1519 507.3 250.6 0 -121.3 -9.692 0
-389.8 -208.4 191.4 3012 507.3 13270 -912.6 0 973.3 -110.9 0
-528.7 213.9 398.8 1572 250.6 -912.6 13180 0 714.8 92.98 0
0 0 0 0 0 0 0 0 0 0
1003 266.1 -1370 -6592 -121.3 973.3 714.8 0 1.977E+005 385.6 0
-35.96 -7.893 3.274 -40.04 -9.692 -110.9 92.98 0 385.6 40.66 0
0 0 0 0 0 0 0 0 0 33.94

```

m3

143

```

98.68 14.03 32.89 168.1 23.01 285 84.71 0.1937 820.8 3.946 6.956
85.91 23.75 55.73 120.6 33.6 147.8 108.3 2.013 384.7 7.043 5.826
7380 -469.3 38.3 -621.8 -127 584.1 -158.2 -3.114 -1355 192.4 0
-469.3 564.1 -27.79 -103.1 -31.92 -526.6 213.5 -2.774 -593.9 -16.22 0
38.3 -27.79 3106 2077 118 856.6 211.3 -5.835 1672 38.85 0
-621.8 -103.1 2077 14550 206.7 1070 1019 -14.33 5290 0 0
-127 -31.92 118 206.7 1129 417.2 818.9 -3.383 465.3 -4.259 0
584.1 -526.6 856.6 1070 417.2 21850 416.3 32.44 5402 167.6 0
-158.2 213.5 211.3 1019 818.9 416.3 11730 -16.57 3500 39.67 0
-3.114 -2.774 -5.835 -14.33 -3.383 32.44 -16.57 4.054 -85.2 -0.7799 0
-1355 -593.9 1672 5290 465.3 5402 3500 -85.2 1.48E+005 -24.38 0
192.4 -16.22 38.85 0 -4.259 167.6 39.67 -0.7799 -24.38 49.6 0
0 0 0 0 0 0 0 0 0 33.94

```

m4

157

```

123.9 14.12 42.37 237.7 32.87 352.1 102.6 0.6365 925.5 3.574 6.956
74.04 22.61 61.82 152 51.43 189.1 141.9 3.411 567.4 7.586 5.826
5483 -530.7 975 630.2 -3.808 -2604 788 30.06 3235 32.02 0
-530.7 511.2 39.13 -41.23 234.9 606.9 160.4 -5.707 1796 -22.3 0
975 39.13 3821 2255 -60.41 2034 263.1 17.92 14030 147.7 0
630.2 -41.23 2255 23100 1610 4396 2890 -10.89 16210 208.7 0
-3.808 234.9 -60.41 1610 2645 1838 218.9 25.09 2364 -29.26 0
-2604 606.9 2034 4396 1838 35740 -2119 -3.224 29070 -116.2 0
788 160.4 263.1 2890 218.9 -2119 20130 27.1 -3060 -48.44 0
30.06 -5.707 17.92 -10.89 25.09 -3.224 27.1 11.64 180 -2.277 0
3235 1796 14030 16210 2364 29070 -3060 180 3.22E+005 163.6 0
32.02 -22.3 147.7 208.7 -29.26 -116.2 -48.44 -2.277 163.6 57.55 0
0 0 0 0 0 0 0 0 0 33.94

```

170
 192 16.95 63.81 288.6 42.69 380 79.11 1.121 1074 6.418 6.956
 108.7 25.52 91.06 197.4 59.24 235.3 111.4 9.161 698.3 11.41 5.826
 11820 -202.5 -792 6072 244.8 3224 -678 -40.84 10480 -54.6 0
 -202.5 651.2 608.8 820.9 362.9 -324.3 676.3 -16.83 -784.1 -3.204 0
 -792 608.8 8291 6505 663.5 -428.6 1450 -46.71 -3243 169.4 0
 6072 820.9 6505 38950 900.3 2369 4637 30.73 20810 45.05 0
 244.8 362.9 663.5 900.3 3510 -446.2 1062 34.19 1779 56.8 0
 3224 -324.3 -428.6 2369 -446.2 55380 3014 -79.77 56860 8.058 0
 -678 676.3 1450 4637 1062 3014 12400 -22.44 8087 3.813 0
 -40.84 -16.83 -46.71 30.73 34.19 -79.77 -22.44 83.92 -601.3 -5.332 0
 10480 -784.1 -3243 20810 1779 56860 8087 -601.3 4.876E+005 -263 0
 -54.6 -3.204 169.4 45.05 56.8 8.058 3.813 -5.332 -263 130.3 0
 0 0 0 0 0 0 0 0 0 0 33.94

m6
 1014
 168.9 17.74 85.68 302.1 47.68 281.2 99.93 3.051 448.6 8.37 6.956
 113.4 27.51 102.4 224.5 59.24 217.9 174.4 14.29 462.9 14.93 5.826
 12870 -387 92.9 3641 134.4 -420.1 -1365 6.486 367.5 -72.8 0
 -387 756.9 188.7 407.6 -9.78 -119.9 326.3 -9.439 -547.6 -17.25 0
 92.9 188.7 10480 2229 163.8 1138 982.3 -2.927 -1327 93.22 0
 3641 407.6 2229 50400 1410 4304 3603 -176.5 13300 261.4 0
 134.4 -9.78 163.8 1410 3510 103.3 651.1 8.469 -575.9 56.6 0
 -420.1 -119.9 1138 4304 103.3 47460 1140 -137 18250 211.4 0
 -1365 326.3 982.3 3603 651.1 1140 30430 -37.4 6056 414 0
 6.486 -9.439 -2.927 -176.5 8.469 -137 -37.4 204.3 -410.2 0 0
 367.5 -547.6 -1327 13300 -575.9 18250 6056 -410.2 2.143E+005 428.4 0
 -72.8 -17.25 93.22 261.4 56.6 211.4 414 0 428.4 222.8 0
 0 0 0 0 0 0 0 0 0 0 33.94

m7
 18
 18.99 7.14 0.04661 133.1 4.167 80.13 167 0 750.1 3.002 6.956
 45.08 12.76 0.202 172 10.91 85.9 205.1 0 543.1 12.27 5.826
 2032 299.2 -0.3641 868.2 -39.85 1212 6426 0 -758.9 -59.17 0
 299.2 162.9 -0.3609 994.4 3.344 36.18 1469 0 34.66 -21.61 0
 -0.3641 -0.3609 0.04079 -3.856 0.3329 12.75 -6.339 0 -1.865 -0.1511 0
 868.2 994.4 -3.856 29580 -281.6 -251.1 -1799 0 17370 -82.28 0
 -39.85 3.344 0.3329 -281.6 119.1 156.6 -11.19 0 -907 -8.167 0
 1212 36.18 12.75 -251.1 156.6 7378 1956 0 3452 -187.6 0
 6426 1469 -6.339 -1799 -11.19 1956 42080 0 -1114 135.9 0
 0 0 0 0 0 0 0 0 0 0 0
 -758.9 34.66 -1.865 17370 -907 3452 -1114 0 2.95E+005 -2332 0
 -59.17 -21.61 -0.1511 -82.28 -8.167 -187.6 135.9 0 -2332 150.5 0
 0 0 0 0 0 0 0 0 0 0 33.94

m8
 85
 60.13 6.345 12.21 125.6 33.87 178.4 89.78 0 584.8 5.053 6.956
 41.31 12.82 19.84 95.93 34.85 115.1 118.2 0 342.3 6.859 5.826
 1707 -137.1 141.8 -47.56 28.79 1526 1719 0 -579.9 42.79 0
 -137.1 164.2 -4.577 -40.57 -49.57 -118 56.07 0 -210.6 -2.285 0
 141.8 -4.577 393.6 451.1 64.99 586.6 30.49 0 -339.6 4.626 0
 -47.56 -40.57 451.1 9203 942.8 2793 -499.1 0 4565 12.5 0
 28.79 -49.57 64.99 942.8 1214 -116.3 -597.4 0 811.2 49.95 0
 1526 -118 586.6 2793 -116.3 13240 3428 0 2324 14.2 0
 1719 56.07 30.49 -499.1 -597.4 3428 13980 0 2307 107 0
 0 0 0 0 0 0 0 0 0 0 0
 -579.9 -210.6 -339.6 4565 811.2 2324 2307 0 1.172E+005 443.8 0
 42.79 -2.285 4.626 12.5 49.95 14.2 107 0 443.8 47.04 0
 0 0 0 0 0 0 0 0 0 0 33.94

m9
 132

90 12.1 34.3 194.7 25 269.4 100.3 0.429 783 4.355 6.956
 69.3 20.47 47.9 143 35.69 164.8 139.9 4.041 392.3 7.292 5.826
 4763 -114.4 641.3 661.4 145.3 864.5 -202.8 -17.01 -2112 79.51 0
 -114.4 418.9 92.15 -43.91 19.72 -337.3 171.8 4.135 -353.3 -14.33 0
 641.3 92.15 2294 1905 343.6 1460 985 -4.064 3213 58.67 0
 661.4 -43.91 1905 20460 -142.9 7804 2021 -11.56 13520 43.81 0
 145.3 19.72 343.6 -142.9 1274 194.1 768.9 -1.586 1624 -24.98 0
 864.5 -337.3 1460 7804 194.1 27170 2721 -52.61 23670 133.4 0
 -202.8 171.8 985 2021 768.9 2721 19570 -35.05 -1976 137.7 0
 -17.01 4.135 -4.064 -11.56 -1.586 -52.61 -35.05 16.33 -107.8 -1.856 0
 -2112 -353.3 3213 13520 1624 23670 -1976 -107.8 1.539E+005 -20.02 0
 79.51 -14.33 58.67 43.81 -24.98 133.4 137.7 -1.856 -20.02 53.17 0
 0 0 0 0 0 0 0 0 0 0 33.94

m10

179
 104.2 11.16 39.82 205.2 27.17 300.8 93.03 0.9271 794.7 4.375 6.956
 62.3 15.77 63.18 146.3 45.22 167.8 131.3 6.449 460.5 8.035 5.826
 3881 23.58 814.8 1796 462 -1767 98.14 -39.77 -2611 26.53 0
 23.58 248.6 199.3 161.5 37.07 -142.9 236 -6.711 508.3 -3.421 0
 814.8 199.3 3992 1932 1034 1930 1501 -19.96 5237 120.8 0
 1796 161.5 1932 21410 1475 908.4 3688 -66.05 10440 32.92 0
 462 37.07 1034 1475 2044 -151.7 973.5 38.78 1645 33.79 0
 -1767 -142.9 1930 908.4 -151.7 28150 -2159 60.59 15610 209 0
 98.14 236 1501 3688 973.5 -2159 17230 -25.4 6227 -35.86 0
 -39.77 -6.711 -19.96 -66.05 38.78 60.59 -25.4 41.58 -181.1 -1.71 0
 -2611 508.3 5237 10440 1645 15610 6227 -181.1 2.121E+005 99.91 0
 26.53 -3.421 120.8 32.92 33.79 209 -35.86 -1.71 99.91 64.56 0
 0 0 0 0 0 0 0 0 0 0 33.94

m11

165
 107.1 15.85 51.28 197.5 27.4 253.9 79.46 1.183 558.8 5.18 6.956
 67.86 22.01 69.1 163.7 38.34 188.4 168.5 6.061 451.6 8.905 5.826
 4604 -209 726.8 2466 -41.63 -2045 3522 -26.73 2635 -69.49 0
 -209 484.2 -16.73 381.9 32.9 178.2 -29.67 -6.936 775.1 -1.96 0
 726.8 -16.73 4775 1674 227.9 1276 535.7 -10.89 8145 175.4 0
 2466 381.9 1674 26800 790.9 4318 -855.4 -100.2 10650 -2.916 0
 -41.63 32.9 227.9 790.9 1470 1408 84.01 -12.78 571.4 61.8 0
 -2045 178.2 1276 4318 1408 35490 -1746 -17.13 19570 90.59 0
 3522 -29.67 535.7 -855.4 84.01 -1746 28410 -50.05 10120 -106.6 0
 -26.73 -6.936 -10.89 -100.2 -12.78 -17.13 -50.05 36.74 -84.85 -3.616 0
 2635 775.1 8145 10650 571.4 19570 10120 -84.85 2.039E+005 0 0
 -69.49 -1.96 175.4 -2.916 61.8 90.59 -106.6 -3.616 0 79.3 0
 0 0 0 0 0 0 0 0 0 0 33.94

m12

1281
 102.7 14.21 82.95 216 32.22 194.9 85.91 2.532 335.1 8.031 6.956
 74.91 24.44 91.58 155 44.74 150.4 136.9 13.82 338.4 13.07 5.826
 5611 -314.9 -144.1 1439 167.6 -202.8 -635.9 -24.84 709.8 -47 0
 -314.9 597.2 96.24 90.89 -3.28 -158 93.68 17.9 -165.4 -23.96 0
 -144.1 96.24 8388 1306 69.66 55.09 1442 91.11 -155 124.5 0
 1439 90.89 1306 24010 485.3 2027 3925 -42.82 5925 28.36 0
 167.6 -3.28 69.66 485.3 2002 -201.8 49 22.25 -151.4 -21.05 0
 -202.8 -158 55.09 2027 -201.8 22620 555.9 -110.1 8550 21.62 0
 -635.9 93.68 1442 3925 49 555.9 18740 7.567 3845 298.9 0
 -24.84 17.9 91.11 -42.82 22.25 -110.1 7.567 190.9 -369.4 -2.167 0
 709.8 -165.4 -155 5925 -151.4 8550 3845 -369.4 1.145E+005 327.3 0
 -47 -23.96 124.5 28.36 -21.05 21.62 298.9 -2.167 327.3 170.9 0
 0 0 0 0 0 0 0 0 0 0 33.94

m13

12
 164.4 12.8 63.5 149.9 54.63 317.6 106.6 0 807.1 11.27 6.956
 151.2 14.5 45.99 97.4 55.61 161.1 124.8 0 603.1 8.979 5.826

-760.5 210.3 428.1 549.4 129.8 -170.5 973.7 0 3883 -35.93 0
 -1932 428.1 2115 241.9 554.9 2689 5120 0 1609 -68.96 0
 -7699 549.4 241.9 9486 1554 -3531 1386 0 704.9 -9.619 0
 -1782 129.8 554.9 1554 3093 -17.92 999.5 0 11000 24.97 0
 12350 -170.5 2689 -3531 -17.92 25960 3982 0 -1263 668.3 0
 -6716 973.7 5120 1386 999.5 3982 15580 0 -2710 -14.57 0
 0 0 0 0 0 0 0 0 0 0
 -1276 3883 1609 704.9 11000 -1263 -2710 0 3.638E+005 -422.4 0
 901.2 -35.93 -68.96 -9.619 24.97 668.3 -14.57 0 -422.4 80.62 0
 0 0 0 0 0 0 0 0 0 33.94

m14

28.21 6.078 0 152.5 5.373 51.05 59.86 0 766.6 0.9518 6.956
 52.25 17.2 0 170.5 15.21 57.84 66.87 0 631.6 2.696 5.826
 2731 573.3 0 8268 533.2 1451 -1929 0 -99.02 94.53 0
 573.3 295.7 0 1580 -37.39 724.1 -416.2 0 3801 -6.63 0
 0 0 0 0 0 0 0 0 0 0
 8268 1580 0 29070 1748 3521 -7285 0 7430 309.8 0
 533.2 -37.39 0 1748 231.3 -73.89 -368.1 0 -3275 41 0
 1451 724.1 0 3521 -73.89 3346 158.6 0 23750 -13.1 0
 -1929 -416.2 0 -7285 -368.1 158.6 4471 0 4519 -65.26 0
 0 0 0 0 0 0 0 0 0 0
 -99.02 3801 0 7430 -3275 23750 4519 0 3.99E+005 -580.7 0
 94.53 -6.63 0 309.8 41 -13.1 -65.26 0 -580.7 7.269 0
 0 0 0 0 0 0 0 0 0 33.94

m15

52
 64.11 9.48 13.84 125.7 18.48 220.5 52.76 0 597.9 4.334 6.956
 47.22 17.02 25.46 89.05 32.25 131.2 70.9 0 330.1 7.847 5.826
 2230 -171.2 416.1 -727.5 -150.8 532.8 -87.06 0 -2120 -19.27 0
 -171.2 289.6 -103.6 227.3 -62.02 -180.8 196.7 0 -28.08 -1.87 0
 416.1 -103.6 648.4 -156.5 -52.56 -514.4 294.3 0 353 -36.97 0
 -727.5 227.3 -156.5 7930 864.5 -794.4 88.4 0 -1587 136.3 0
 -150.8 -62.02 -52.56 864.5 1040 -194.6 -352.2 0 489.7 20.5 0
 532.8 -180.8 -514.4 -794.4 -194.6 17210 -1516 0 2468 6.176 0
 -87.06 196.7 294.3 88.4 -352.2 -1516 5027 0 1428 101.8 0
 0 0 0 0 0 0 0 0 0 0
 -2120 -28.08 353 -1587 489.7 2468 1428 0 1.089E+005 352.2 0
 -19.27 -1.87 -36.97 136.3 20.5 6.176 101.8 0 352.2 61.58 0
 0 0 0 0 0 0 0 0 0 33.94

m16

72
 84.78 10.68 22.7 208.7 21.65 286.6 95.52 0.6504 895.3 3.787 6.956
 49.87 14.91 34.55 122.5 29.14 108 104.5 4.705 414.6 5.155 5.826
 2487 21.56 346.3 2224 -15.98 312.2 2073 24.87 -682.4 18.51 0
 21.56 222.2 40.18 352.5 -74.71 191.5 -14.02 -5.05 599.6 -4.995 0
 346.3 40.18 1194 110.1 200.4 -794.5 747.1 -15.12 -1490 61.28 0
 2224 352.5 110.1 15010 549.8 1349 2944 -88.19 16000 61.9 0
 -15.98 -74.71 200.4 549.8 849 84.93 -51.75 8.637 1244 40.41 0
 312.2 191.5 -794.5 1349 84.93 11650 327 -115.3 14320 -43.96 0
 2073 -14.02 747.1 2944 -51.75 327 10910 58 10010 169.1 0
 24.87 -5.05 -15.12 -88.19 8.637 -115.3 58 22.14 -122.9 -0.4123 0
 -682.4 599.6 -1490 16000 1244 14320 10010 -122.9 1.719E+005 -463.9 0
 18.51 -4.995 61.28 61.9 40.41 -43.96 169.1 -0.4123 -463.9 26.58 0
 0 0 0 0 0 0 0 0 0 33.94

m17

99
 135.4 10.38 27.7 240.4 22.92 351.4 83.07 0.4497 957.5 4.203 6.956
 87.03 17.69 44.09 212.5 32.65 206 122.9 2.553 512.8 8.742 5.826
 7574 -95.43 153.5 2108 93.75 806.5 1861 -47.77 4998 -93.58 0
 -95.43 312.8 -37.43 221.8 28.29 -320.5 -210.9 0.04516 462.6 2.628 0

153. -37.43 1944 1387 108 1326 1658 -6.079 4138 20.81 0
 2108 221.8 1387 45160 1478 -8185 2952 -31.47 8827 -241.5 0
 93.75 28.29 108 1478 1066 -1311 220.7 -2.417 -251.1 -34.82 0
 806.5 -320.5 1326 -8185 -1311 42420 -3013 -11.57 29360 460.9 0
 1861 -210.9 1658 2952 220.7 -3013 15110 -19.77 7501 -2.149 0
 -47.77 0.04516 -6.079 -31.47 -2.417 -11.57 -19.77 6.518 61.54 -1.897 0
 4998 462.6 4138 8827 -251.1 29360 7501 61.54 2.63E+005 416.9 0
 -93.58 2.628 20.81 -241.5 -34.82 460.9 -2.149 -1.897 416.9 76.42 0
 0 0 0 0 0 0 0 0 0 0 33.94

m18

84
 182.1 20.09 49.47 277.6 35.83 344.4 70.36 2.09 997 6.785 6.956
 98.34 30.34 60.31 174.5 60.91 220.6 124.6 8.42 744.9 15.46 5.826
 9670 -134.3 990.4 3399 2306 -1757 196.1 64.59 10770 -229.5 0
 -134.3 920.7 237.9 1255 -201.5 1372 -155 51.36 -6035 -49.72 0
 990.4 237.9 3637 3348 -18.37 1543 593.7 -31.49 2920 -74.58 0
 3399 1255 3348 30470 2701 962.7 2110 13.23 7151 -240.1 0
 2306 -201.5 -18.37 2701 3711 255.3 1070 32.31 6035 0 0
 -1757 1372 1543 962.7 255.3 48670 -82.48 46.44 24650 92.08 0
 196.1 -155 593.7 2110 1070 -82.48 15530 -87.09 21440 543.2 0
 64.59 51.36 -31.49 13.23 32.31 46.44 -87.09 70.9 -395.2 -8.851 0
 10770 -6035 2920 7151 6035 24650 21440 -395.2 5.549E+005 -518.1 0
 -229.5 -49.72 -74.58 -240.1 0 92.08 543.2 -8.851 -518.1 238.9 0
 0 0 0 0 0 0 0 0 0 0 33.94

m19

534
 172.5 15 72.4 319.4 50.62 302.2 85.41 2.463 498.3 6.434 6.956
 108 23.62 93.42 233.2 59.26 197.3 146.4 14.16 489.6 11.02 5.826
 11660 -387.7 -211.9 3753 492.8 -916.3 -47.43 -22.94 475.9 -122.6 0
 -387.7 557.9 -30.89 16.53 44.79 -18.64 214.4 -21.08 705.4 -14.84 0
 -211.9 -30.89 8727 1198 -5.536 2433 943.6 -13.23 -548.9 66.92 0
 3753 16.53 1198 54400 1064 -1381 3517 -29.73 456.8 331.6 0
 492.8 44.79 -5.536 1064 3512 654.7 763.4 31.05 2118 -51.6 0
 -916.3 -18.64 2433 -1381 654.7 38920 2888 -122.9 11880 -58.71 0
 -47.43 214.4 943.6 3517 763.4 2888 21430 47.69 645.1 8.067 0
 -22.94 -21.08 -13.23 -29.73 31.05 -122.9 47.69 200.6 159.5 15.61 0
 475.9 705.4 -548.9 456.8 2118 11880 645.1 159.5 2.397E+005 -286 0
 -122.6 -14.84 66.92 331.6 -51.6 -58.71 8.067 15.61 -286 121.5 0
 0 0 0 0 0 0 0 0 0 0 33.94

m20

14
 13.93 6.125 0 77.84 0.5752 31.92 68.61 0 727.4 0 6.956
 26.86 13.13 0 104.9 2.132 42.98 78.86 0 657.9 0 5.826
 721.6 302.6 0 1054 23.08 329.1 1218 0 -3039 0 0
 302.6 172.4 0 793.5 20.66 210.5 625.4 0 -630.5 0 0
 0 0 0 0 0 0 0 0 0 0 0
 1054 793.5 0 11010 139.2 2593 5859 0 18290 0 0
 23.08 20.66 0 139.2 4.547 1.925 75.84 0 56.11 0 0
 329.1 210.5 0 2593 1.925 1848 1214 0 14760 0 0
 1218 625.4 0 5859 75.84 1214 6219 0 -2853 0 0
 0 0 0 0 0 0 0 0 0 0 0
 -3039 -630.5 0 18290 56.11 14760 -2853 0 4.328E+005 0 0
 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 33.94

m21

44
 63.52 9.676 8.172 145.5 23.04 201.6 60.36 0.1453 725.5 4.24 6.956
 40.88 14.13 21.47 123.1 34.58 117.7 91.57 0.9825 360.6 5.813 5.826
 1672 70.48 96.56 -35.23 253.1 -264.8 614 7.11 -44.22 -32.8 0
 70.48 199.7 -28.82 467.9 41.54 216.3 112.6 -1.472 840.6 -32.77 0
 96.56 -28.82 461 370 26.73 -409.5 507.3 -1.245 -1680 11.36 0
 -35.23 467.9 370 15160 -902.5 3218 4520 -15.6 -4305 -80.86 0

255.1 11.03 20.73 -302.3 11.30 -830.6 -481.3 0.727 -2145 -21.11 0
 -264.8 216.3 -409.5 3218 -830.6 13860 -1089 -20.01 7132 -113.6 0
 614 112.6 507.3 4520 -481.3 -1089 8385 33.65 -2377 -22.89 0
 7.11 -1.472 -1.245 -15.6 6.727 -20.01 33.65 0.9653 -16.65 0.514 0
 -44.22 840.6 -1680 -4305 -2145 7132 -2377 -16.65 1.3E+005 -81.74 0
 -32.8 -32.77 11.36 -80.86 -21.11 -113.6 -22.89 0.514 -81.74 33.79 0
 0 0 0 0 0 0 0 0 0 0 33.94

m22

71
 92.69 7.945 27.72 214.1 18.64 260.2 99.1 0.2076 838.3 3.117 6.956
 59.78 13.29 31.55 141.1 25.03 107.6 128 1.564 397.9 5.997 5.826
 3573 -17.48 56.59 3282 -161.6 -186.6 1003 -10.1 -3734 -6.453 0
 -17.48 176.6 -18.87 738.9 -18.96 201.7 -149.7 -0.04156 655.7 -3.825 0
 56.59 -18.87 995.6 231.6 -22.11 1277 12.12 5.329 2411 -11.16 0
 3282 738.9 231.6 19920 -56.51 -577.3 3577 -15.89 7356 -77.86 0
 -161.6 -18.96 -22.11 -56.51 626.4 -105.1 -173 7.788 -1982 21.46 0
 -186.6 201.7 1277 -577.3 -105.1 11590 509.9 -7.575 1199 127.8 0
 1003 -149.7 12.12 3577 -173 509.9 16390 -18.42 -1732 17.66 0
 -10.1 -0.04156 5.329 -15.89 7.788 -7.575 -18.42 2.445 4.978 0.1594 0
 -3734 655.7 2411 7356 -1982 1199 -1732 4.978 1.583E+005 2.386 0
 -6.453 -3.825 -11.16 -77.86 21.46 127.8 17.66 0.1594 2.386 35.96 0
 0 0 0 0 0 0 0 0 0 0 33.94

m23

79
 100.3 10.97 34.04 205.5 17.93 253.8 91.26 0.2975 776.9 1.92 6.956
 74.32 15.65 49.2 141.8 27.26 123 126.6 1.591 433.5 4.367 5.826
 5524 -252.3 -157.2 2381 -133.7 27.41 -781.2 7.57 7925 -47.06 0
 -252.3 244.8 -31.56 141.9 6.397 69.26 -63.41 -1.519 -915.6 0.205 0
 -157.2 -31.56 2421 1172 -89.86 -78.64 984.5 14.09 1791 4.942 0
 2381 141.9 1172 20090 738 557.7 -448.8 0.6767 13520 -31.57 0
 -133.7 6.397 -89.86 738 743 -338.5 -448.8 1.171 -1441 -5.237 0
 27.41 69.26 -78.64 557.7 -338.5 15120 -389.3 -21.91 14390 49.93 0
 -781.2 -63.41 984.5 -448.8 -448.8 -389.3 16040 -15.52 4282 -91.25 0
 7.57 -1.519 14.09 0.6767 1.171 -21.91 -15.52 2.532 -35.18 -0.03475 0
 7925 -915.6 1791 13520 -1441 14390 4282 -35.18 1.879E+005 71.93 0
 -47.06 0.205 4.942 -31.57 -5.237 49.93 -91.25 -0.03475 71.93 19.07 0
 0 0 0 0 0 0 0 0 0 0 33.94

m24

88
 108.2 8.113 44.71 200.9 14.9 204.2 70.85 0.429 691.2 5.306 6.956
 80.61 15.82 68.45 161.4 30.23 136.7 129.1 2.871 533.4 13.3 5.826
 6498 -52.27 347.6 3317 104.8 540.1 1145 -5.324 11010 56.83 0
 -52.27 250.1 -37.89 393 -44.47 216.2 -36.75 -1.544 -1383 -16.2 0
 347.6 -37.89 4686 1734 701.5 -954.7 4251 2.359 4673 -45.52 0
 3317 393 1734 26040 1107 -286.9 4562 -19.92 16870 -251.1 0
 104.8 -44.47 701.5 1107 914 -504.3 1421 -0.9549 -483.8 9.651 0
 540.1 216.2 -954.7 -286.9 -504.3 18700 -2136 22.38 9481 120 0
 1145 -36.75 4251 4562 1421 -2136 16670 -30.77 14390 -53.23 0
 -5.324 -1.544 2.359 -19.92 -0.9549 22.38 -30.77 8.245 75.05 -2.291 0
 11010 -1383 4673 16870 -483.8 9481 14390 75.05 2.845E+005 -1199 0
 56.83 -16.2 -45.52 -251.1 9.651 120 -53.23 -2.291 -1199 176.9 0
 0 0 0 0 0 0 0 0 0 0 33.94

m25

608
 106.1 12.79 77.94 218.2 28.58 200.4 80.31 1.605 344.7 6.789 6.956
 78.45 22.67 97.33 156.1 38.5 136.6 120.3 9.389 339.8 11.14 5.826
 6155 -252.5 -244.4 1163 63.42 235.8 -66.04 -28.73 639.9 -61.15 0
 -252.5 513.7 -44.12 38.91 21.81 34.05 141.7 -10.43 -61.62 -5.805 0
 -244.4 -44.12 9474 1474 -101.2 106.4 2505 -23.76 1158 84.54 0
 1163 38.91 1474 24360 90.12 1066 1971 32.24 5516 109.5 0
 63.42 21.81 -101.2 90.12 1482 184 162 11.93 -680.3 -9.431 0
 235.8 34.05 106.4 1066 184 18660 65.71 21.8 8588 -10.65 0

```

-61.15 141.7 2505 1971 162 65.71 14460 -44.04 2370 96.42 0
-28.73 -10.43 -23.76 32.24 11.93 21.8 -44.04 88.15 -15.95 -0.6273 0
639.9 -61.62 1158 5516 -680.3 8588 2370 -15.95 1.155E+005 189.2 0
-61.15 -5.805 84.54 109.5 -9.431 -10.65 96.42 -0.6273 189.2 124 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m26

```

8
82.78 11.38 55.88 173.5 56 210.2 74.46 11.37 853.6 6.373 6.956
59.22 17.24 73.16 125.2 58.54 117 92.05 28.62 507.3 6.038 5.826
3506 -456.3 2396 370.7 -714.1 -4426 -1434 486.3 -4506 -129.1 0
-456.3 297.1 -546 -1099 234.1 1545 1241 -148 -227.4 -41.21 0
2396 -546 5352 -1319 -299.8 -4938 -2788 -600.9 -26280 -9.276 0
370.7 -1099 -1319 15680 -2324 -4687 -4702 -566.1 14160 341.7 0
-714.1 234.1 -299.8 -2324 3427 2465 -1002 303.2 -1247 -127.2 0
-4426 1545 -4938 -4687 2465 13680 5503 -652.8 17860 -327 0
-1434 1241 -2788 -4702 -1002 5503 8474 -142.3 13730 -107.8 0
486.3 -148 -600.9 -566.1 303.2 -652.8 -142.3 818.9 9146 -19.87 0
-4506 -227.4 -26280 14160 -1247 17860 13730 9146 2.573E+005 -193 0
-129.1 -41.21 -9.276 341.7 -127.2 -327 -107.8 -19.87 -193 36.45 0
0 0 0 0 0 0 0 0 0 0 33.94

```

ss

m1

```

41
27.06 11.57 1.973 153.2 11.95 86.44 94.36 0 1025 0.3023 6.956
47.24 17.6 10.52 173 29.22 110.2 144.3 0 608.8 1.332 5.826
2232 262.8 184.3 1896 160.1 1119 -633.8 0 7708 0.692 0
262.8 309.8 42.4 947.1 61.21 -77.56 83.8 0 2068 -3.024 0
184.3 42.4 110.6 194.7 -24.28 144.8 -19.73 0 83.24 -0.6163 0
1896 947.1 194.7 29930 -202.2 6956 6789 0 20120 -13.82 0
160.1 61.21 -24.28 -202.2 853.9 38.63 -413.2 0 3629 7.822 0
1119 -77.56 144.8 6956 38.63 12140 7390 0 3286 -3.668 0
-633.8 83.8 -19.73 6789 -413.2 7390 20810 0 -1230 -18.06 0
0 0 0 0 0 0 0 0 0 0 0
7708 2068 83.24 20120 3629 3286 -1230 0 3.706E+005 -66.48 0
0.692 -3.024 -0.6163 -13.82 7.822 -3.668 -18.06 0 -66.48 1.773 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m2

```

122
69.83 7.851 13.16 147.8 26.67 214.8 77.22 0.1868 724.1 3.681 6.956
49.11 13.07 26.38 107.4 30.37 123.1 129.3 1.335 401.5 5.556 5.826
2411 -25.67 237.1 10.54 123.8 -30.23 -108 -3.41 59.14 -9.821 0
-25.67 170.8 32.06 -120.7 34.93 -131.9 59.15 -1.187 -236.1 -11.76 0
237.1 32.06 695.9 422 6.41 415.7 -37.52 -1.233 1133 -18.47 0
10.54 -120.7 422 11520 -430.4 568.3 1499 -2.294 -4741 32.21 0
123.8 34.93 6.41 -430.4 922.5 228.1 -137.5 2.92 256.1 11.14 0
-30.23 -131.9 415.7 568.3 228.1 15160 -1799 -20.06 494.2 30.09 0
-108 59.15 -37.52 1499 -137.5 -1799 16720 25.56 -3063 -11.49 0
-3.41 -1.187 -1.233 -2.294 2.92 -20.06 25.56 1.783 10.72 0.1855 0
59.14 -236.1 1133 -4741 256.1 494.2 -3063 10.72 1.612E+005 243.1 0
-9.821 -11.76 -18.47 32.21 11.14 30.09 -11.49 0.1855 243.1 30.86 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m3

```

179
89.5 8.127 27.27 188.8 27.72 274.5 92.72 0.5812 765.5 4.085 6.956
59.42 13.54 59.83 132.2 43.48 160.7 149.7 6.123 440.8 8.922 5.826
3530 -78.04 383.9 1359 204.1 -582.3 391.3 10.55 4740 -68.38 0
-78.04 183.4 100.5 141.4 32.97 13.05 -40.53 -0.9121 -340.2 -2.899 0
383.9 100.5 3580 893.5 70.23 615.2 1182 -1.465 1345 66.19 0
1359 141.4 893.5 17470 948.1 637 -1029 110.9 7224 116.7 0
204.1 32.97 70.23 948.1 1890 -69.85 -416.4 70.28 0 31.42 0
-582.3 13.05 615.2 637 -69.85 25810 -529 -49.19 1700 335.4 0
391.3 -40.53 1182 -1029 -416.4 -529 22400 49.49 -2309 -24.03 0
10.55 -0.9121 -1.465 110.9 70.28 -49.19 49.49 37.49 -13.5 0 0
4740 -340.2 1345 7224 0 1700 -2309 -13.5 1.943E+005 -114 0
-68.38 -2.899 66.19 116.7 31.42 335.4 -24.03 0 -114 79.6 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m4

```

192
116.4 14.78 33.45 253 28.51 351.2 123.1 1.052 915.7 3.419 6.956
66.62 23.49 48.19 171.8 40.73 233.3 210.9 10.23 539.6 6.618 5.826
4438 -241 208.7 1488 -89.54 -1849 1658 2.045 3775 20.28 0
-241 551.7 48.68 238 58.35 -32.88 -381.5 7.211 -2117 -14.3 0
208.7 48.68 2323 1482 337.6 1866 1077 -29.59 2470 20.41 0
1488 238 1482 29500 454.7 2204 5326 7.031 4356 -180.7 0
-89.54 58.35 337.6 454.7 1659 2480 764.5 -10.84 -549.4 -11.32 0
-1849 -32.88 1866 2204 2480 54420 738.1 -150.4 16490 33.96 0
1658 -381.5 1077 5326 764.5 738.1 44490 -129.5 32320 9.771 0
2.045 7.211 -29.59 7.031 -10.84 -150.4 -129.5 104.7 38.65 -3.251 0
3775 -2117 2470 4356 -549.4 16490 32320 38.65 2.912E+005 178.5 0
20.28 -14.3 20.41 -180.7 -11.32 33.96 9.771 -3.251 178.5 43.79 0
0 0 0 0 0 0 0 0 0 0 33.94

```

m5
160
174.1 15.65 44.34 296.8 36.5 373.5 98.81 1.031 988.6 6.07 6.956
95.42 23.63 64.35 217.6 50.97 228.3 172.2 6.663 710.5 10.84 5.826
9104 -448.7 558.7 3903 744 1503 -1397 10.17 4000 -26.9 0
-448.7 558.3 124.7 -262.2 -115.6 -178 -65.11 -12.12 -1058 27.15 0
558.7 124.7 4141 406 -127.9 1543 2216 -38.16 1006 72.56 0
3903 -262.2 406 47340 2517 3924 8693 -207.3 42210 292.5 0
744 -115.6 -127.9 2517 2598 -395.6 553 -35.66 -3513 -29.29 0
1503 -178 1543 3924 -395.6 52120 3538 -25.86 46720 529.7 0
-1397 -65.11 2216 8693 553 3538 29660 -89.5 39770 224 0
10.17 -12.12 -38.16 -207.3 -35.66 -25.86 -89.5 44.4 -426.1 -3.973 0
4000 -1058 1006 42210 -3513 46720 39770 -426.1 5.048E+005 1587 0
-26.9 27.15 72.56 292.5 -29.29 529.7 224 -3.973 1587 117.5 0
0 0 0 0 0 0 0 0 0 33.94

m6
656
171.5 14.89 69.87 311.8 47.71 285.6 130.2 2.975 439.5 8.309 6.956
125.3 26.31 87.92 223.6 59.15 203.4 221.3 15.34 477.4 14.97 5.826
15710 -758.3 165.3 2579 415.1 153 -1165 -76.9 59.84 28.15 0
-758.3 692 198.9 -47.06 48.23 5.351 -209.6 -1.614 -866.5 -44.5 0
165.3 198.9 7730 373.6 322.4 1323 330.8 89.01 797.5 60.55 0
2579 -47.06 373.6 50010 383.6 1729 3365 -96.05 9609 -224.3 0
415.1 48.23 322.4 383.6 3498 288.8 575.9 -28.13 -2090 -5.313 0
153 5.351 1323 1729 288.8 41380 3646 -131.1 14180 18.27 0
-1165 -209.6 330.8 3365 575.9 3646 48970 -207.1 6128 745.5 0
-76.9 -1.614 89.01 -96.05 -28.13 -131.1 -207.1 235.3 -183.1 6.43 0
59.84 -866.5 797.5 9609 -2090 14180 6128 -183.1 2.279E+005 421.7 0
28.15 -44.5 60.55 -224.3 -5.313 18.27 745.5 6.43 421.7 224.1 0
0 0 0 0 0 0 0 0 0 33.94

B.53
m7
25
30.24 7.369 1.056 68.96 5.64 84.98 110.9 0 1139 0.8742 6.956
54.37 15.85 5.313 90.57 13.96 93.84 115.1 0 646.2 3.141 5.826
2956 2.585 1.444 4.925 196.6 750 3636 0 -4497 -3.587 0
2.585 251.2 -8.168 14.35 -43.36 -117.5 410.5 0 -2222 -3.236 0
1.444 -8.168 28.23 214.2 -6.23 75.79 -83.79 0 -370.8 16.07 0
4.925 14.35 214.2 8204 361.6 3145 62.56 0 -19140 136.6 0
196.6 -43.36 -6.23 361.6 194.8 267.2 110.9 0 -72.16 2.368 0
750 -117.5 75.79 3145 267.2 8805 1804 0 6488 42.15 0
3636 410.5 -83.79 62.56 110.9 1804 13250 0 -23130 -41.58 0
0 0 0 0 0 0 0 0 0 0 0
-4497 -2222 -370.8 -19140 -72.16 6488 -23130 0 4.176E+005 -146.2 0
-3.587 -3.236 16.07 136.6 2.368 42.15 -41.58 0 -146.2 9.868 0
0 0 0 0 0 0 0 0 0 33.94

m8
122
61.04 8.399 14.6 127.8 27.43 193.4 88.96 0.04843 776.7 2.913 6.956
42 12.13 27.79 94.86 36.56 104.7 123.6 0.2768 403.2 6.164 5.826
1764 -38.71 216 952.3 -21.5 228.7 -534.6 -0.4069 -677.4 -48.68 0
-38.71 147.1 29.33 -41.42 52.32 116.8 101.9 0.2383 -273.9 -10.39 0
216 29.33 772.5 76.46 -68.08 -20.37 -137.4 -0.3615 -560.4 -29.64 0
952.3 -41.42 76.46 8999 270.5 427 -1032 -3.256 -2027 -2.924 0
-21.5 52.32 -68.08 270.5 1337 -367.4 -352.4 0.1518 -884.5 -12.85 0
228.7 116.8 -20.37 427 -367.4 10960 414 0.1449 -126.6 9.035 0
-534.6 101.9 -137.4 -1032 -352.4 414 15270 -1.026 1644 -25.9 0
-0.4069 0.2383 -0.3615 -3.256 0.1518 0.1449 -1.026 0.0766 -7.923 0.08871 0
-677.4 -273.9 -560.4 -2027 -884.5 -126.6 1644 -7.923 1.626E+005 208.8 0
-48.68 -10.39 -29.64 -2.924 -12.85 9.035 -25.9 0.08871 208.8 38 0
0 0 0 0 0 0 0 0 0 33.94

m9
154

62.73 15.27 56.02 155.6 35.59 143.3 108.9 6.421 407.4 6.879 5.826
 3934 2.873 625.5 126.9 -428.6 -494.5 -218.7 55.18 2070 -87.16 0
 2.873 233.1 167.6 -9.501 57.06 98.47 -88.16 2.941 -590.8 -15.12 0
 625.5 167.6 3139 78.44 -67.79 -208.8 738.5 -8.273 2739 58.19 0
 126.9 -9.501 78.44 24200 326.7 -2988 -2271 48.95 380.3 -10.7 0
 -428.6 57.06 -67.79 326.7 1267 -561.1 128 -10.74 -3654 33.3 0
 -494.5 98.47 -208.8 -2988 -561.1 20540 -1046 -38.65 11150 176.5 0
 -218.7 -88.16 738.5 -2271 128 -1046 11870 -48.97 7101 31.48 0
 55.18 2.941 -8.273 48.95 -10.74 -38.65 -48.97 41.23 227.6 -0.7067 0
 2070 -590.8 2739 380.3 -3654 11150 7101 227.6 1.659E+005 -311.1 0
 -87.16 -15.12 58.19 -10.7 33.3 176.5 31.48 -0.7067 -311.1 47.32 0
 0 0 0 0 0 0 0 0 0 0 33.94

m10

210
 92.91 12.29 35.19 206.3 23.81 273.1 121.8 1.965 753.1 3.742 6.956
 62.8 18.28 51.47 147.3 37.36 155.4 154.5 12.03 446.8 7.945 5.826
 3944 -286.9 310.3 1064 -154.8 -956.5 164.9 -48.36 5359 -8.982 0
 -286.9 334 73.37 137.3 48.48 207.3 -163.7 0.4398 -783.9 -16.7 0
 310.3 73.37 2649 219.9 330.7 919.9 477 -38.4 2162 -15.95 0
 1064 137.3 219.9 21700 919 320.5 773.6 109.9 1974 -78.41 0
 -154.8 48.48 330.7 919 1396 -261.3 -507.8 -8.99 -2554 -19.29 0
 -956.5 207.3 919.9 320.5 -261.3 24150 984.3 -160.8 12990 -116.1 0
 164.9 -163.7 477 773.6 -507.8 984.3 23860 105.9 13180 154.6 0
 -48.36 0.4398 -38.4 109.9 -8.99 -160.8 105.9 144.8 -424.7 -5.64 0
 5359 -783.9 2162 1974 -2554 12990 13180 -424.7 1.996E+005 7.1 0
 -8.982 -16.7 -15.95 -78.41 -19.29 -116.1 154.6 -5.64 7.1 63.13 0
 0 0 0 0 0 0 0 0 0 0 33.94

m11

194
 112.3 10.43 48.11 209 23.98 236.3 89.29 2.013 586.4 6.164 6.956
 74.77 16.89 69.68 148.3 36.59 153.4 169.4 11.91 463.6 12.3 5.826
 5590 -311.9 854.4 2561 -202.4 -1388 1773 -74.78 -1074 -6.435 0
 -311.9 285.2 -11.77 293 41.4 323.8 143 12.27 1276 -0.2076 0
 854.4 -11.77 4855 1767 102 -406.2 2018 -48.12 0 -4.284 0
 2561 293 1767 21990 162.8 2298 2838 30.02 9554 200.5 0
 -202.4 41.4 102 162.8 1339 140.3 216.9 20.48 -203.5 30.14 0
 -1388 323.8 -406.2 2298 140.3 23540 -1247 -237.5 9601 -133.9 0
 1773 143 2018 2838 216.9 -1247 28680 -88.73 11850 83.29 0
 -74.78 12.27 -48.12 30.02 20.48 -237.5 -88.73 141.8 -5.52 3.514 0
 -1074 1276 0 9554 -203.5 9601 11850 -5.52 2.149E+005 775.2 0
 -6.435 -0.2076 -4.284 200.5 30.14 -133.9 83.29 3.514 775.2 151.2 0
 0 0 0 0 0 0 0 0 0 0 33.94

m12

923
 105.9 13.87 74.91 259.8 31.57 190.3 144.6 2.934 342.1 7.259 6.956
 80.59 21.75 91.71 194.4 42.97 149.8 213.8 16.06 339.1 13.35 5.826
 6494 -264.6 251.3 595.2 162.8 -265.6 -1602 -51.77 -1230 -92.49 0
 -264.6 473 85.76 143.7 9.345 -104.3 125.5 -8.731 22.12 -14.22 0
 251.3 85.76 8411 766.5 260.1 -549.6 2000 -25.04 -497.6 61.2 0
 595.2 143.7 766.5 37780 33.41 262.1 4322 -124.9 4614 57.07 0
 162.8 9.345 260.1 33.41 1846 225.3 73.49 -18.63 786.9 -33.26 0
 -265.6 -104.3 -549.6 262.1 225.3 22450 -2178 -86.62 6351 35.99 0
 -1602 125.5 2000 4322 73.49 -2178 45710 -37.77 1450 422.3 0
 -51.77 -8.731 -25.04 -124.9 -18.63 -86.62 -37.77 257.9 -277.7 -2.572 0
 -1230 22.12 -497.6 4614 786.9 6351 1450 -277.7 1.15E+005 683.4 0
 -92.49 -14.22 61.2 57.07 -33.26 35.99 422.3 -2.572 683.4 178.1 0
 0 0 0 0 0 0 0 0 0 0 33.94

m13

10
 96 8.048 109.5 238.1 44.15 283.3 152.2 0.5881 592.5 3.954 6.956
 88.22 11.21 163.8 179.4 47.84 217.7 174 1.868 521.5 7.235 5.826

77 283.7 -3254 1029 734.3 -9680 9059 -62.96 9018 55.53 0
 -283.7 125.6 -675.3 321.6 -258.9 -329.4 -922.5 -5.276 -1181 -18.57 0
 -3294 -675.3 26820 14630 5656 15580 11940 267.7 -6406 73.46 0
 1029 321.6 14630 32180 1879 -5819 7149 134.7 -10480 -75.27 0
 734.3 -258.9 5656 1879 2289 1208 3830 59.43 -3767 -96.91 0
 -9680 -329.4 15580 -5819 1208 47400 -6252 200.9 41900 417.4 0
 9059 -922.5 11940 7149 3830 -6252 30290 0.3251 21510 399.1 0
 -62.96 -5.276 267.7 134.7 59.43 200.9 0.3251 3.49 -289.4 -2.595 0
 9018 -1181 -6406 -10480 -3767 41900 21510 -289.4 2.72E+005 1181 0
 55.53 -18.57 73.46 -75.27 -96.91 417.4 399.1 -2.595 1181 52.34 0
 0 0 0 0 0 0 0 0 0 0 33.94

m14

17
 8.825 6.63 0.87 115 8.768 92.63 133.4 0 720.3 0.4575 6.956
 26.79 15.94 3.558 138.1 27.76 140.3 210.3 0 668.8 1.895 5.826
 718 -42.7 35.18 -751.2 -40.92 2018 4665 0 -2760 18.74 0
 -42.7 254 0 875.9 367.2 809.3 -83.78 0 -3325 0 0
 35.18 0 12.66 -3.439 -8.001 -18.47 19.45 0 -516.3 6.743 0
 -751.2 875.9 -3.439 19070 502.3 6974 6186 0 -1293 -1.832 0
 -40.92 367.2 -8.001 502.3 770.8 412.8 -151.8 0 -4475 -4.263 0
 2018 809.3 -18.47 6974 412.8 19680 18640 0 -6005 -9.838 0
 4665 -83.78 19.45 6186 -151.8 18640 44220 0 -12240 10.36 0
 0 0 0 0 0 0 0 0 0 0
 -2760 -3325 -516.3 -1293 -4475 -6005 -12240 0 4.473E+005 -275.1 0
 18.74 0 6.743 -1.832 -4.263 -9.838 10.36 0 -275.1 3.593 0
 0 0 0 0 0 0 0 0 0 0 33.94

m15

54
 78.01 8.525 9.896 164.3 25.53 185.1 82.23 0 670.3 3.239 6.956
 46.21 14.2 23.68 103.9 31.99 126.7 133.3 0 373.2 6.14 5.826
 2135 -149.6 105 1714 204 257.5 -357.3 0 17.25 -19.01 0
 -149.6 201.7 -49.43 -323.1 -1.363 118.7 503.5 0 -439.9 -3.313 0
 105 -49.43 560.6 199.3 172.7 365.9 1114 0 1246 39.54 0
 1714 -323.1 199.3 10800 438.7 973.9 1898 0 6554 27.43 0
 204 -1.363 172.7 438.7 1023 1329 430.6 0 -3259 75.81 0
 257.5 118.7 365.9 973.9 1329 16040 303.9 0 10450 244.2 0
 -357.3 503.5 1114 1898 430.6 303.9 17770 0 4378 40.92 0
 0 0 0 0 0 0 0 0 0 0
 17.25 -439.9 1246 6554 -3259 10450 4378 0 1.393E+005 100.8 0
 -19.01 -3.313 39.54 27.43 75.81 244.2 40.92 0 100.8 37.7 0
 0 0 0 0 0 0 0 0 0 0 33.94

m16

71
 93.16 8.352 25.88 219.5 18.35 290.9 108.2 0.08995 834.2 5.38 6.956
 56.6 13.09 41.57 147.2 27.82 139.6 167.1 0.768 472.7 10.68 5.826
 3203 78.54 200 1341 61.4 387 586.3 2.651 3184 -146.3 0
 78.54 171.4 126.8 161.8 -81.22 29.23 24.06 -0.362 340.4 -25.59 0
 200 126.8 1728 666.9 69.39 1456 3723 -1.437 -1140 96.82 0
 1341 161.8 666.9 21660 -556.8 -1951 1180 0.7912 -12520 -334.9 0
 61.4 -81.22 69.39 -556.8 773.9 368.8 269.6 -1.688 -2538 50.82 0
 387 29.23 1456 -1951 368.8 19470 2472 -10.82 4486 228.1 0
 586.3 24.06 3723 1180 269.6 2472 27920 -10.01 1896 519.4 0
 2.651 -0.362 -1.437 0.7912 -1.688 -10.82 -10.01 0.5898 37.4 -0.5004 0
 3184 340.4 -1140 -12520 -2538 4486 1896 37.4 2.235E+005 95.95 0
 -146.3 -25.59 96.82 -334.9 50.82 228.1 519.4 -0.5004 95.95 114.1 0
 0 0 0 0 0 0 0 0 0 0 33.94

m17

110
 137.4 10.8 29.01 331.1 21 334.8 161.2 0.01384 1033 2.786 6.956
 70.3 17.14 41.2 231.4 29.97 157.8 233.8 0.173 519.8 4.91 5.826
 4942 -7.229 -72.41 2636 438.2 1343 -98.62 -0.9728 5554 62.13 0
 -7.229 293.8 125.7 563.2 78.07 202.9 709.3 0.2698 320.7 -7.153 0

2636 563.2 1611 53560 -55.48 3214 12230 -1.201 8661 -20.45 0
 438.2 78.07 372.9 -55.48 898 723.7 1422 0.01037 -1386 -10.59 0
 1343 202.9 1073 3214 723.7 24910 9891 3.44 10670 68.2 0
 -98.62 709.3 3304 12230 1422 9891 54670 1.132 27470 -173.4 0
 -0.9728 0.2698 0.449 -1.201 0.01037 3.44 1.132 0.02992 4.496 -0.04671 0
 5554 320.7 -1949 8661 -1386 10670 27470 4.496 2.702E+005 66.36 0
 62.13 -7.153 -8.497 -20.45 -10.59 68.2 -173.4 -0.04671 66.36 24.11 0
 0 0 0 0 0 0 0 0 0 0 33.94

m18

66
 203.7 13.48 54 306.7 31.83 405.8 137.5 0 1011 4.967 6.956
 104.1 19.45 73.17 230.7 47.97 227.6 197.3 0 779.3 8.962 5.826
 10850 556.9 -396.3 2788 1474 -6162 4910 0 8846 85.88 0
 556.9 378.1 350 399.3 113.8 -716.9 -560 0 -4061 -25.79 0
 -396.3 350 5355 152 888 882.6 1111 0 -1026 -87.22 0
 2788 399.3 152 53240 1782 787.7 819.3 0 4675 -95.13 0
 1474 113.8 888 1782 2301 -775 312.2 0 -9307 -39.55 0
 -6162 -716.9 882.6 787.7 -775 51790 852.9 0 55510 -287.6 0
 4910 -560 1111 819.3 312.2 852.9 38910 0 67180 -194.5 0
 0 0 0 0 0 0 0 0 0 0 0
 8846 -4061 -1026 4675 -9307 55510 67180 0 6.073E+005 391.1 0
 85.88 -25.79 -87.22 -95.13 -39.55 -287.6 -194.5 0 391.1 80.33 0
 0 0 0 0 0 0 0 0 0 0 33.94

m19

254
 179.8 15.91 64.49 377.1 41.81 329.1 143.4 2.429 489.1 7.324 6.956
 111.4 26.19 86.55 298.9 54.91 215.2 252.5 14.98 488.3 12.67 5.826
 12410 -411.4 154.3 2331 -477.1 -335.6 -2531 -245.3 -2121 67.73 0
 -411.4 686.1 335.5 1488 50.34 -281.9 -291 15.3 -818.6 -16.92 0
 154.3 335.5 7491 1733 275.6 -2198 3059 28.52 -2367 86.62 0
 2331 1488 1733 89350 1379 7012 1283 -447.8 12410 140.1 0
 -477.1 50.34 275.6 1379 3015 295.4 1262 -0.8225 1984 23.65 0
 -335.6 -281.9 -2198 7012 295.4 46320 -2065 19.34 22910 35.44 0
 -2531 -291 3059 1283 1262 -2065 63750 -56.73 1479 367.8 0
 -245.3 15.3 28.52 -447.8 -0.8225 19.34 -56.73 224.4 -563.2 -15.56 0
 -2121 -818.6 -2367 12410 1984 22910 1479 -563.2 2.384E+005 -80.41 0
 67.73 -16.92 86.62 140.1 23.65 35.44 367.8 -15.56 -80.41 160.5 0
 0 0 0 0 0 0 0 0 0 0 33.94

m20

16
 37.95 15.96 4.412 143.7 11.27 78.63 114.1 0 650.2 2.864 6.956
 68.5 33.16 17.65 147.5 24.34 90.59 170.7 0 597.7 6.438 5.826
 4693 1756 -178.9 676.8 588.6 -1533 8899 0 12320 131 0
 1756 1100 -74.91 -68.45 -150.1 -354.5 4245 0 1784 116.3 0
 -178.9 -74.91 311.5 -184.8 116.4 793 147.6 0 1192 -13.52 0
 676.8 -68.45 -184.8 21740 32.3 1469 -2165 0 54640 -323.7 0
 588.6 -150.1 116.4 32.3 592.5 533.6 373.9 0 4364 -30.4 0
 -1533 -354.5 793 1469 533.6 8207 -1469 0 920.5 -168.6 0
 8899 4245 147.6 -2165 373.9 -1469 29140 0 408.1 417.6 0
 0 0 0 0 0 0 0 0 0 0 0
 12320 1784 1192 54640 4364 920.5 408.1 0 3.572E+005 -1354 0
 131 116.3 -13.52 -323.7 -30.4 -168.6 417.6 0 -1354 41.45 0
 0 0 0 0 0 0 0 0 0 0 33.94

m21

69
 68.63 10.87 8.638 147.9 27.13 182.7 84.49 0 747.1 3.419 6.956
 51.32 19.54 23.09 141.1 41.15 120 128.1 0 518.8 7.304 5.826
 2634 -32.1 -215.6 -1543 -209.1 295.6 -1098 0 1757 -39.73 0
 -32.1 382 19.4 -104.8 92.48 -480.8 67.6 0 1724 -19.13 0
 -215.6 19.4 533 -397.5 -68.4 238.2 139 0 -1437 21.25 0
 -1543 -104.8 -397.5 19920 -243.9 -1016 4068 0 -1098 -21.65 0

-20. . 92.48 -68.4 -243.9 1693 -380.2 -26.36 0 -1686 -36.37
 295.6 -480.8 238.2 -1016 -380.2 14400 -2675 0 -9400 1.753 0
 -1098 67.6 139 4068 -26.36 -2675 16410 0 731 160 0
 0 0 0 0 0 0 0 0 0 0 0
 1757 1724 -1437 -1098 -1686 -9400 731 0 2.691E+005 -1019 0
 -39.73 -19.13 21.25 -21.65 -36.37 1.753 160 0 -1019 53.35 0
 0 0 0 0 0 0 0 0 0 0 33.94

m22

98
 102.1 8.127 15.61 227.1 12.58 266.4 121.1 0.06919 777.7 4.714 6.956
 61.4 13.18 27.93 146.1 19.75 156.1 211.9 0.6504 412 9.069 5.826
 3770 -43.69 209.2 1166 -25.47 -2713 -2693 -1.038 -758.8 55.12 0
 -43.69 173.6 20.61 331 28.37 -325 78.18 1.808 722 -16.49 0
 209.2 20.61 780.3 322.3 17.66 780.6 367 0.8175 -402.8 -22.04 0
 1166 331 322.3 21330 204.8 91.21 2971 7.979 11370 -222.5 0
 -25.47 28.37 17.66 204.8 390.2 -83.26 242.8 -0.8351 -349.9 -6.449 0
 -2713 -325 780.6 91.21 -83.26 24370 6418 -5.077 -2251 -25.48 0
 -2693 78.18 367 2971 242.8 6418 44900 -7.993 26280 -69.18 0
 -1.038 1.808 0.8175 7.979 -0.8351 -5.077 -7.993 0.423 23.85 -0.3126 0
 -758.8 722 -402.8 11370 -349.9 -2251 26280 23.85 1.697E+005 470.7 0
 55.12 -16.49 -22.04 -222.5 -6.449 -25.48 -69.18 -0.3126 470.7 82.24 0
 0 0 0 0 0 0 0 0 0 0 33.94

m23

111
 118.6 9.307 31.88 268 20.26 299.8 148.6 3.39 767.6 3.685 6.956
 158.4 16.48 46.27 192.2 36.92 202.7 221.6 27.56 455.7 8.37 5.826
 25100 -331.6 322.5 1918 1591 -3725 -3018 52.39 8446 80.88 0
 -331.6 271.6 -22.11 408.6 51.72 557.8 660.8 -31.34 548.2 -9.38 0
 322.5 -22.11 2141 1619 27.33 243.8 1650 -104.6 1455 53.44 0
 1918 408.6 1619 36950 -262.6 740.2 6686 -434.4 10340 -57.92 0
 1591 51.72 27.33 -262.6 1363 -232 -564.5 381.6 3567 47.29 0
 -3725 557.8 243.8 740.2 -232 41080 -2649 -407.7 3879 -142.5 0
 -3018 660.8 1650 6686 -564.5 -2649 49080 -470.1 16560 192.9 0
 52.39 -31.34 -104.6 -434.4 381.6 -407.7 -470.1 759.5 841.4 92.5 0
 8446 548.2 1455 10340 3567 3879 16560 841.4 2.077E+005 -64.84 0
 80.88 -9.38 53.44 -57.92 47.29 -142.5 192.9 92.5 -64.84 70.06 0
 0 0 0 0 0 0 0 0 0 0 33.94

m24

99
 102.9 10.91 33.74 228.6 15.61 256.4 107.3 1.667 526.9 4.735 6.956
 73.36 18.24 49.96 184.1 29.1 189.8 226.5 7.798 418.2 9.371 5.826
 5381 -14.72 -212.6 2984 -42.69 -737.9 2060 -28.03 6135 65.99 0
 -14.72 332.7 -119.4 110.8 88.62 190.4 702.3 22.9 -305.1 5.127 0
 -212.6 -119.4 2496 -1398 -85.77 3148 -509.3 19.87 -4116 84.75 0
 2984 110.8 -1398 33880 942.5 1327 7587 -175.1 6158 239.7 0
 -42.69 88.62 -85.77 942.5 846.6 767.6 6.59 14.75 -194.7 67.89 0
 -737.9 190.4 3148 1327 767.6 36020 3095 -68.08 4762 106.7 0
 2060 702.3 -509.3 7587 6.59 3095 51300 -121.9 15250 154.9 0
 -28.03 22.9 19.87 -175.1 14.75 -68.08 -121.9 60.8 -280.4 -2.923 0
 6135 -305.1 -4116 6158 -194.7 4762 15250 -280.4 1.749E+005 -364.4 0
 65.99 5.127 84.75 239.7 67.89 106.7 154.9 -2.923 -364.4 87.82 0
 0 0 0 0 0 0 0 0 0 0 33.94

m25

399
 109.5 11.85 63.73 294.2 25.35 198.8 133.1 1.716 326.4 7.484 6.956
 79.98 18.11 83.49 206.6 36.14 136.5 207 10.82 306.9 13.19 5.826
 6397 -269.5 447.4 1157 219.7 284 33.11 -57.99 662.8 -27.44 0
 -269.5 328.1 176.9 228.3 0 -173.1 -11.25 -9.604 -5.559 0.239 0
 447.4 176.9 6971 1587 515.9 342 4563 50.59 -461.2 167.4 0
 1157 228.3 1587 42700 627.3 1608 4791 547.9 6088 100.9 0
 219.7 0 515.9 627.3 1306 0 209.5 51.23 -33.27 16.21 0
 284 -173.1 342 1608 0 18650 2092 -93.09 4736 36.03 0

55.11 11.15 4583 4791 29.5 2992 42850 -8.96 4193 409.7 0
 -57.99 -9.604 50.59 547.9 51.23 -93.09 -8.96 117.1 -119.6 0 0
 662.8 -5.559 -461.2 6088 -33.27 4736 4193 -119.6 94190 457.6 0
 -27.44 0.239 167.4 100.9 16.21 36.03 409.7 0 457.6 174.1 0
 0 0 0 0 0 0 0 0 0 0 33.94

m26

7
 131.7 13.8 61.96 289.9 68.04 290.6 195.1 0 537 15.91 6.956
 61.96 21.07 99.07 160.4 50.86 155.5 317.7 0 456.5 21.54 5.826
 3839 -151.4 -1197 3478 -1834 -3016 -7716 0 -707.1 -222.9 0
 -151.4 443.9 -720.2 -1010 -615 3.276 -2195 0 -3587 -124.4 0
 -1197 -720.2 9815 10200 3134 -3790 29360 0 17680 1426 0
 3478 -1010 10200 25730 1909 -10720 33680 0 12590 2657 0
 -1834 -615 3134 1909 2586 2689 11990 0 394.6 713.1 0
 -3016 3.276 -3790 -10720 2689 24180 -10180 0 -55010 -566.1 0
 -7716 -2195 29360 33680 11990 -10180 1.009E+005 0 47270 5638 0
 0 0 0 0 0 0 0 0 0 0 0
 -707.1 -3587 17680 12590 394.6 -55010 47270 0 2.084E+005 98.32 0
 -222.9 -124.4 1426 2657 713.1 -566.1 5638 0 98.32 464 0
 0 0 0 0 0 0 0 0 0 0 33.94

??

APPENDIX C

CODE LISTINGS

APPENDIX C1

PARSEL CODE LISTINGS

```

C.*. * PARSEL.FOR *****
C
C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: PARSEL
C
C DESCRIPTION: PARSEL is the main program of the parameter selection
C               pre-processor code for the HEDRIC environmental/dose codes.
C
C Version 1.0 generates parameter values representing one simulation run (001),
C one radionuclide (I-131), and the year 1945 using best estimates of all input
C parameters.
C
C LANGUAGE: F77
C
C FILE I/O:
C
C - Input files:
C   PARSEL reads all HEDRIC model input files. See Appendix B of the
C   HEDRIC dose code report for a complete listing of these files.
C
C - Output files:
C   Three unformatted binary files are created:
C
C   * I-131 AIR45.001 - Transposed air transport model results. Each record
C   holds daily air concentration and surface deposition values for one node.
C   See subroutine TRANS_AIR for detailed file structure.
C
C   * PARAMS.001 - Contains all time-invariant input parameter values (those
C   parameters that do not vary by year or day within_year).
C   - Records A1 thru A7 are written once to this file.
C
C   * PARAMS45.001 - Time-varying input parameters for year 1945, run no. 001:
C   - Record B1 (time-varying, location-invariant parameters) is written
C   once at the beginning of this file, followed by:
C   - Records B2 thru B4 (params. that vary with both time and location) are
C   written repeatedly as a group, once for each location (node) that
C   has a node_status of 0 (nodes for which we have data and will do the
C   calculations).
C   The unformatted binary parameter file PARAMS45.001 thus looks like this:
C
C       Record B2, first node with NODE_STATUS = 0
C       Record B3, first node with NODE_STATUS = 0
C       Record B4, first node with NODE_STATUS = 0
C
C       Record B2, second node with NODE_STATUS = 0
C       Record B3, second node with NODE_STATUS = 0
C       Record B4, second node with NODE_STATUS = 0
C
C       .
C       .
C       .
C
C       Record B2, last node with NODE_STATUS = 0
C       Record B3, last node with NODE_STATUS = 0
C       Record B4, last node with NODE_STATUS = 0
C
C   In the future, multiple yearly parameter files generated from correlated
C   distributions will be named 'PARAMSyy.rrr', where yy is the year (19yy),
C   and rrr is the MonteCarlo run number.
C
C FUNCTIONS/SUBROUTINES CALLED:
C   Get_Nodes_to_CD
C   Trans_Air
C
C-----BEGIN IMPLEMENTATION BLOCK-----

```

```

C
C  AUTHOR: Robert A. Burnett
C          Battelle, Pacific Northwest Laboratory
C
C  DATE: 12/31/91
C
C  MODIFICATION HISTORY:
C
C-----
C ##### BEGIN DECLARATION BLOCK #####
C-----
C
C  IMPLICIT NONE
C  INTEGER NUM_NUCS, NUM_NODES, NUM_YEARS, NUM_PLANT, NUM_ANIMAL
C  INTEGER NUM_DAIRY_REGIMES, NUM_ASL, NUM_FOODS, NUM_CREAMERIES
C  REAL*4 LAMBDA I131
C  INTEGER MAX_DAYS, PYN, YN, PYNMAXH, PNN, PDASL, APD
C  INTEGER MAX_HARVESTS, NUM_CD
C  PARAMETER (NUM_NUCS = 1)      ! I131
C  PARAMETER (NUM_NODES = 961)   ! 31 X 31 grid
C  PARAMETER (NUM_YEARS = 1)     ! 1945 (later add 1946, 1947)
C  PARAMETER (NUM_PLANT = 9)     ! original 8 from Phase 1 plus "grass hay"
C  PARAMETER (NUM_ANIMAL = 4)    ! NON-MILK animals / animal products:
C    ! Cattle (beef), Deer (venison), fowl (poultry), Laying hens (eggs)
C  PARAMETER (NUM_DAIRY_REGIMES = 4) ! Number of dairy cow feeding regimes
C  PARAMETER (NUM_ASL = 26)      ! Number of Age-Sex-Lifestyle population groups
C  PARAMETER (NUM_FOODS = 11)    ! Number of food groups consumed by humans
C  PARAMETER (NUM_CREAMERIES = 26) ! Number of dairy processors/distributors
C  PARAMETER (LAMBDA I131 = 0.69314718 / 8.05) ! ln(2) / I131_half_life
C  PARAMETER (MAX_DAYS = 33)    ! Maximum no. of days in a year
C ***** Later change the above to 371 *****
C *** MAX_DAYS = 371 allows for last 6 days of 1944 to be included in 1945 ***
C  PARAMETER (MAX_HARVESTS = 3) ! Max. no. of harvests per year
C    ! over all plant types
C  PARAMETER (PYN = NUM_PLANT * NUM_YEARS * NUM_NODES)
C  PARAMETER (PYNMAXH = MAX_HARVESTS * PYN)
C  PARAMETER (PNN = NUM_PLANT * NUM_NODES * NUM_NUCS)
C  PARAMETER (YN = NUM_YEARS * NUM_NODES)
C  PARAMETER (PDASL = NUM_PLANT * NUM_ASL * MAX_DAYS)
C  PARAMETER (APD = NUM_ANIMAL * NUM_PLANT * MAX_DAYS)
C  PARAMETER (NUM_CD = 99)      ! Number of Phase 1 Census divisions (CD's)
C
C ##### Records A1 thru A7 get written to the PARAMS.001 file #####
C (These records contain variables that do not vary by year or day-within-year)
C
C-- Record A1 - Dimensional parameters
C
C  integer KNUM_NUCS / NUM_NUCS /
C  integer KNUM_NODES / NUM_NODES /
C  integer KNUM_YEARS / NUM_YEARS /
C  integer KNUM_PLANT / NUM_PLANT /
C  integer KNUM_ANIMAL / NUM_ANIMAL /
C  integer KNUM_DAIRY_REGIMES / NUM_DAIRY_REGIMES /
C  integer KNUM_ASL / NUM_ASL /
C  integer KNUM_FOODS / NUM_FOODS /
C  integer KNUM_CREAMERIES / NUM_CREAMERIES /
C  integer FIRST_YEAR / 1945 /
C-- First year (1945) includes last 6 days of 1944 (beginning 12/26/44).
C  integer DAYS_IN_YEAR(NUM_YEARS) / 33 / ! Test data covers 33 days
C  integer NUM_HARVESTS(NUM_PLANT) / 0, 0, 1, 0, 3, 0, 1, 0, 1 /
C  character*5 NUC_LIST(NUM_NUCS) / '1-131' /
C
C-- Record A2 - Parameters that do not vary with time, space, plant type,
C  or Age/Sex/Lifestyle (ASL) population group
C
C  real*4 XS / 0.001 / ! Depth of soil surface layer, meters
C  real*4 XR / 0.250 / ! Depth of root zone soil layer    ters

```



```

      real*4 LAMBDA_DECAY(NUM_NUCS) ! Radiological decay const.
      data LAMBDA_DECAY(1) / LAMBDA_I131 / ! Iodine-131
C
C--Record A3 - Params. varying with plant or animal type, NOT with time or space
C
      real*4 LAMBDA_SPLASH(NUM_PLANT) / NUM_PLANT * 8.6E-4 /
      real*4 KSEN(NUM_PLANT) / 4 * 0.05, 0.15, 4 * 0.05 /
      real*4 KINT(NUM_PLANT) / 2.9, 3.6, 2.9, 3.6, 5 * 2.9 /
      real*4 KG(NUM_PLANT) / 4 * 0.12, 0.27, 4 * 0.12 /
      real*4 LAMBDA_TRANS(NUM_PLANT, NUM_NUCS) / NUM_PLANT * 8.5E-3 /
      real*4 TF_MILK(NUM_NUCS) / 0.012 /
      real*4 TF_ANIMAL(NUM_ANIMAL, NUM_NUCS)
      data TF_ANIMAL / 0.002, 0.002, 0.018, 2.8 /
C
C-- Record A4 - Parameters that vary with space, NOT with plant type or time
C
C- NODE_STATUS(i) = 0 : calculate values for node i (input data are present)
C- NODE_STATUS(i) = -2 : do not compute node i (some input data are missing)
      integer NODE_STATUS (NUM_NODES)
      real*4 LAMBDA_PERC(NUM_NODES) / NUM_NODES * 0.3 /
      real*4 PS(NUM_NODES) ! Surface layer soil density, kg/m**3
      data PS / NUM_NODES * 1500. /
      real*4 PR(NUM_NODES) ! Root zone soil density, kg/m**3
      data PR / NUM_NODES * 1500. /
      real*4 LAMBDA_LEACH(NUM_NODES, NUM_NUCS) / NUM_NODES * 0.002 /
      real*4 REGIME_FRACTIONS (NUM_DAIRY_REGIMES, NUM_NODES)
C
C-- Record A5 - Parameters that vary with space and plant type, not time
C
      real*4 BMAX(NUM_PLANT, NUM_NODES)
      data BMAX
CCC $ /NUM_NODES * (.2, .5, .144, .54, .2, .3, .3, .04, .3)/
      real*4 BMIN(NUM_PLANT, NUM_NODES)
      data BMIN
CCC $ /NUM_NODES * (.01, .01, .01, .27, .01, .03, .01, .01, .03)/
      real*4 LAMBDA_WEATHER (NUM_PLANT, NUM_NODES, NUM_NUCS)
      data LAMBDA_WEATHER / PNN * 0.05 /
      real*4 CR (NUM_PLANT, NUM_NODES, NUM_NUCS) / PNN * 0.1 /
C
C-- Record A6 - Parameters governing milk accumulation and distribution
C
      real*4 MILK_SOURCE_FRACTION (NUM_NODES, NUM_CREAMERIES)
      real*4 MILK_DISTR_FRACTION (NUM_CREAMERIES, NUM_NODES)
C
C-- Record A7 - Age/Sex/Lifestyle-specific dose factors and shielding factors
      real*4 SUBMERSION_SHIELD_FACTOR (NUM_ASL)
      data SUBMERSION_SHIELD_FACTOR / NUM_ASL * 0.9 /

      real*4 GROUNDSHINE_SHIELD_FACTOR (NUM_ASL)
      data GROUNDSHINE_SHIELD_FACTOR / NUM_ASL * 0.9 /

      real*4 SUBMERSION_DOSE_FACTOR (NUM_ASL, NUM_NUCS)
      data SUBMERSION_DOSE_FACTOR / NUM_ASL * 1.6E+05 /

      real*4 GROUNDSHINE_DOSE_FACTOR (NUM_ASL, NUM_NUCS) ! Upper soil layer
      data GROUNDSHINE_DOSE_FACTOR / NUM_ASL * 3.6E+03 /

      real*4 ROOTZONE_DOSE_FACTOR (NUM_ASL, NUM_NUCS) ! Root zone groundshine
      data ROOTZONE_DOSE_FACTOR / NUM_ASL * 3.6E+03 /

      real*4 INGESTION_DOSE_FACTOR (NUM_ASL, NUM_NUCS)
      data INGESTION_DOSE_FACTOR /
$ 1.5E+07, 9.8E+06, 6.4E+06, 4.2E+06, 2.75E+06, 1.8E+06,
$ 1.5E+07, 9.8E+06, 6.4E+06, 4.2E+06, 2.75E+06, 1.8E+06,
$ 1.8E+06,
$ 1.5E+07, 9.8E+06, 6.4E+06, 4.2E+06, 2.75E+06, 1.8E+06,

```

```

$      1.5E+07, 9.8E+06, 6.4E+06, 4.2E+06, 2.75E+06, 1.8E+06,
$      1.8E+06 /

real*4 INHALATION_DOSE_FACTOR(NUM_ASL, NUM_NUCS)
data INHALATION_DOSE_FACTOR /
$      7.7E+06, 5.22E+06, 3.54E+06, 2.4E+06, 1.62E+06, 1.1E+06,
$      7.7E+06, 5.22E+06, 3.54E+06, 2.4E+06, 1.62E+06, 1.1E+06,
$      1.1E+06,
$      7.7E+06, 5.22E+06, 3.54E+06, 2.4E+06, 1.62E+06, 1.1E+06,
$      7.7E+06, 5.22E+06, 3.54E+06, 2.4E+06, 1.62E+06, 1.1E+06,
$      1.1E+06 /

C
C- End of records that are written to the time-invariant param. file PARAMS.001
C #####
C
C
C #####
C ### Record B1 and repeating groups of Records B2, B3, and B4 are written ###
C ### to each yearly parameter file (in this case, file PARAMS45.001) #####
C ### These records contain variables that vary by year and/or by day of year #
C
C-- Record B1 - Daily varying parameters that do not vary with location --
C (This record gets written once at the beginning of the file)
C
real*4 T_FRESH_MILK_FC(MAX_DAYS) ! Family cow: fresh milk holdup, days
data T_FRESH_MILK_FC / MAX_DAYS * 1.0 /
real*4 T_FRESH_MILK_CR(MAX_DAYS) ! Creamery: fresh milk holdup, days
data T_FRESH_MILK_CR / MAX_DAYS * 4.0 /
real*4 T_FRESH_MILK_GRO(MAX_DAYS) ! Grocery: fresh milk holdup, days
data T_FRESH_MILK_GRO / MAX_DAYS * 4.0 /
real*4 T_STORED_MILK_GRO(MAX_DAYS) ! Grocery: stored milk holdup, days
data T_STORED_MILK_GRO / MAX_DAYS * 30.0 /
real*4 T_ANIMAL(NUM_ANIMAL, MAX_DAYS) ! Animal product holdup time, days
data T_ANIMAL / MAX_DAYS * (7.5, 7.5, 1.75, 1.75) /
real*4 BREATHING_RATE(NUM_ASL, MAX_DAYS)
data BREATHING_RATE / 3.8, ! Group 1
CCC $ 6., 10., 14., 18.,
CCC $ 23., 3.8, ! Groups 6 and 7
CCC $ 6., 10., 14., 18.,
CCC $ 20., 20., 3.8, ! Groups 12, 13, 14
CCC $ 6., 10., 14., 18.,
CCC $ 23., 3.8, ! Groups 19 and 20
CCC $ 6., 10., 14., 18.,
CCC $ 20., 20. / ! Groups 25 and 26

C--Daily food consumption rates for "reference" individuals from each ASL group
real*4 REF_FOOD_CONSUMPTION (NUM_FOODS, NUM_ASL, MAX_DAYS)

C--Fraction of contamination removed from food plants by washing and processing
real*4 WASHING_FRACTION (NUM_PLANT, NUM_ASL, MAX_DAYS)
data WASHING_FRACTION / PDASL * 0.0 / ! Code will set non-zero values
C
C *****
C--Records B2, B3, and B4 contain location-dependent variables, and are thus
C written repeatedly as a group, once for each location (node).
C ** NOTE **: Since each parameter in these records is set to its value for
C the first node and then written to the file for that node, then repeated
C for the other nodes, the arrays do not have a dimension for NUM_NODES.
C
C-Record B2 - Dates (day of year) of discrete events for current year, location
C
integer CURRENT_PASTURE_BEGIN
integer CURRENT_PASTURE_END
integer CURRENT_HARVEST_DATES (MAX_HARVESTS, NUM_PLANT)
integer CURRENT_TILLAGE_DATES (NUM_PLANT)

```

```

C-- Record B3 - Location-dependent variables that also vary by day of year --
C      (values are for current location)
      real*4 ML (MAX_DAYS) ! Mass loading of soil in air, kg/m**3
      data ML / MAX_DAYS * 0.015 /

      real*4 VD (MAX_DAYS) ! Deposition velocity, m/day
      data VD / MAX_DAYS * 173. /

C
C-- Record B4 - Daily parameters associated with animal products and feeding
C      (values are for current location)
      real*4 DAIRY_SOIL_INGESTION (NUM DAIRY REGIMES, MAX_DAYS)
CCC     data DAIRY_SOIL_INGESTION / MAX_DAYS * (1.0, 1.0, 1.5, 1.5) /
      real*4 ANIMAL_SOIL_INGESTION (NUM ANIMAL, MAX_DAYS)
CCC     data ANIMAL_SOIL_INGESTION / MAX_DAYS * (0.5, 0.1, 0.01, 0.01) /
      real*4 DAIRY_FEED_CONSUMPTION (NUM DAIRY REGIMES, NUM_PLANT,
      $                               MAX_DAYS)
      real*4 ANIMAL_FEED_CONSUMPTION (NUM ANIMAL, NUM_PLANT,
      $                               MAX_DAYS)
      data ANIMAL_FEED_CONSUMPTION /APD * 0./ ! Code will set non-zero values

C
C- End of repeating group (records B2, B3, B4) - Repeat for each location (node)
C *****
C
C-- End of records written to yearly parameter file (PARAMS45.001) --
C #####
C
C -----
C ##### Local variables #####
C
C-- Census division numbers associated with each node ( 0 ==> none )
      INTEGER NCD(NUM_NODES)

      INTEGER AIR_DAYS ! No. days of data in this year's air transport file(a)
      INTEGER LLIST(NUM_NODES) / NUM_NODES * 0 / ! Linked lista
      INTEGER LHEAD(NUM_CD) / NUM_CD * 0 / ! Linked list heads
      INTEGER CDNODES(NUM_NODES) ! Sequential lists of nodes in each CD
      INTEGER CDSTART(NUM_CD+1) ! Index into CDNODES for 1st node of each CD
      INTEGER ISTATUS, I, J, K, M, JX, NN, TEMP_DAYS
      INTEGER NNTOT, IDAY, INODE, IYEAR, CURRENT_YEAR
      INTEGER ALF_CUT1, ALF_CUT2, ALF_CUT3, GRASS_CUT
      real*4 DISTR_SUM
      character*1 dumchar
      integer*2 PS_DATES (2, NUM_CD)
      integer idum1, idum2, asl_idx
      character*120 odum(6)
      character*60 PFILE, filespec
      character*3 block_ident
      character*3 block_check( NUM_ASL ) /
      . 'm1', 'm2', 'm3', 'm4', 'm5', 'm6', 'm7', 'm8', 'm9', 'm10',
      . 'm11', 'm12', 'm13', 'm14', 'm15', 'm16', 'm17', 'm18', 'm19', 'm20',
      . 'm21', 'm22', 'm23', 'm24', 'm25', 'm26' /

C-- Temporary storage for pasture begin/end dates, harvest and tillage dates,
C by location (node)
      integer PASTURE_BEGIN (NUM_YEARS, NUM_NODES)
      integer PASTURE_END (NUM_YEARS, NUM_NODES)
      integer HARVEST_DATES (MAX_HARVESTS, NUM_PLANT, NUM_YEARS, NUM_NODES)
      data HARVEST_DATES / PYNMAXH * 0 / ! Code fills in non-zero dates
      integer TILLAGE_DATES (NUM_PLANT, NUM_YEARS, NUM_NODES)

      REAL*4 CD_REG_FRACS (NUM DAIRY REGIMES, NUM_CD)
      REAL*4 CD_SOURCE_FRAC (NUM CREAMERIES, NUM_CD)
      REAL*4 CD_DISTR_FRAC (NUM CREAMERIES, NUM_CD)

      real*4 WINTER_FOOD_CONSUMPTION (NUM_FOODS, NUM_ASL)
      real*4 SPRING_FOOD_CONSUMPTION (NUM_FOODS, NUM_ASL)

```

```

real*4 SUMMER FOOD CONSUMPTION (NUM_FOODS, NUM_ASL)
real*4 FALL FOOD CONSUMPTION (NUM_FOODS, NUM_ASL)

```

```

REAL*4 WINTER FEED (NUM DAIRY REGIMES, NUM_PLANT)

```

```

DATA WINTER_FEED / 0.0, 0.0, 0.0, 0.0, ! 1 - Leafy veg.
$                  0.0, 0.0, 0.0, 0.0, ! 2 - Other veg.
$                  1.5, 1.5, 1.0, 1.0, ! 3 - Grain
$                  0.0, 0.0, 0.0, 0.0, ! 4 - Fruit
$                  9.5, 9.5, 9.0, 0.0, ! 5 - Alfalfa
$                  0.0, 0.0, 0.0, 0.0, ! 6 - Pasture grass
$                  0.0, 0.0, 0.0, 0.0, ! 7 - Silage
$                  0.0, 0.0, 0.0, 0.0, ! 8 - Sagebrush
$                  0.0, 0.0, 0.0, 9.0 / ! 9 - Grass hay

```

```

REAL*4 SPRING FEED (NUM DAIRY REGIMES, NUM_PLANT)

```

```

DATA SPRING_FEED / 0.0, 0.0, 0.0, 0.0, ! 1 - Leafy veg.
$                  0.0, 0.0, 0.0, 0.0, ! 2 - Other veg.
$                  1.5, 1.5, 1.0, 1.0, ! 3 - Grain
$                  0.0, 0.0, 0.0, 0.0, ! 4 - Fruit
$                  5.25, 5.25, 9.0, 0.0, ! 5 - Alfalfa
$                  4.25, 4.25, 0.0, 0.0, ! 6 - Pasture grass
$                  0.0, 0.0, 0.0, 0.0, ! 7 - Silage
$                  0.0, 0.0, 0.0, 0.0, ! 8 - Sagebrush
$                  0.0, 0.0, 0.0, 9.0 / ! 9 - Grass hay

```

```

REAL*4 SUMMER FEED (NUM DAIRY REGIMES, NUM_PLANT)

```

```

DATA SUMMER_FEED / 0.0, 0.0, 0.0, 0.0, ! 1 - Leafy veg.
$                  0.0, 0.0, 0.0, 0.0, ! 2 - Other veg.
$                  1.5, 1.5, 1.0, 1.0, ! 3 - Grain
$                  0.0, 0.0, 0.0, 0.0, ! 4 - Fruit
$                  1.0, 1.0, 9.0, 0.0, ! 5 - Alfalfa
$                  8.5, 8.5, 0.0, 0.0, ! 6 - Pasture grass
$                  0.0, 0.0, 0.0, 0.0, ! 7 - Silage
$                  0.0, 0.0, 0.0, 0.0, ! 8 - Sagebrush
$                  0.0, 0.0, 0.0, 9.0 / ! 9 - Grass hay

```

```

REAL*4 EARLY FALL FEED (NUM DAIRY REGIMES, NUM_PLANT)

```

```

DATA EARLY_FALL_FEED / 0.0, 0.0, 0.0, 0.0, ! 1 - Leafy veg.
$                  0.0, 0.0, 0.0, 0.0, ! 2 - Other veg.
$                  1.5, 1.5, 1.0, 1.0, ! 3 - Grain
$                  0.0, 0.0, 0.0, 0.0, ! 4 - Fruit
$                  5.25, 5.25, 9.0, 0.0, ! 5 - Alfalfa
$                  4.25, 4.25, 0.0, 0.0, ! 6 - Pasture grass
$                  0.0, 0.0, 0.0, 0.0, ! 7 - Silage
$                  0.0, 0.0, 0.0, 0.0, ! 8 - Sagebrush
$                  0.0, 0.0, 0.0, 9.0 / ! 9 - Grass hay

```

```

REAL*4 LATE FALL FEED (NUM DAIRY REGIMES, NUM_PLANT)

```

```

DATA LATE_FALL_FEED / 0.0, 0.0, 0.0, 0.0, ! 1 - Leafy veg.
$                  0.0, 0.0, 0.0, 0.0, ! 2 - Other veg.
$                  1.5, 1.5, 1.0, 1.0, ! 3 - Grain
$                  0.0, 0.0, 0.0, 0.0, ! 4 - Fruit
$                  5.25, 9.5, 9.0, 0.0, ! 5 - Alfalfa
$                  0.0, 0.0, 0.0, 0.0, ! 6 - Pasture grass
$                  4.25, 0.0, 0.0, 0.0, ! 7 - Silage
$                  0.0, 0.0, 0.0, 0.0, ! 8 - Sagebrush
$                  0.0, 0.0, 0.0, 9.0 / ! 9 - Grass hay

```

```

C
C #####
C ##### Begin Execution Block #####
C #####
C

```

```

C-- Read air transport results, transpose, and write to file I-131_AIR45.001 --
CALL TRANS_AIR ('[HEDR.PHASE2.DATA]JAN45.AIR',
$              'I-131_AIR45.001', AIR_DAYS)

```

```

C-- Ask user for total number of simulation days for which parameter values
C are desired --
type *,

```

```

$ 'Enter the total number of days (beginning with 12/26/1944):'
read (*,*) TEMP_DAYS

```

```

IF (TEMP_DAYS .gt. AIR_DAYS) THEN
  TYPE *, ' There are only ', AIR_DAYS,
  $      ' days of data in the air transport files --'
  TYPE *, '-- Resetting number of simulation days to ', AIR_DAYS
  TEMP_DAYS = AIR_DAYS
ENDIF
DAYS_IN_YEAR(1) = TEMP_DAYS

```

C
C-- Set values in arrays that couldn't be easily set with DATA statements

```

DO K = 1, NUM_NODES
  BMAX(1,K) = 0.2
  BMAX(2,K) = 0.5
  BMAX(3,K) = 0.144
  BMAX(4,K) = 0.54
  BMAX(5,K) = 0.27
  BMAX(6,K) = 0.27
  BMAX(7,K) = 0.3
  BMAX(8,K) = 0.04
  BMAX(9,K) = 0.27

  BMIN(1,K) = 0.01
  BMIN(2,K) = 0.01
  BMIN(3,K) = 0.01
  BMIN(4,K) = 0.27
  BMIN(5,K) = 0.01
  BMIN(6,K) = 0.03
  BMIN(7,K) = 0.01
  BMIN(8,K) = 0.01
  BMIN(9,K) = 0.03

  DO J = 1, NUM_YEARS
    TILLAGE_DATES(1,J,K) = 21 ! Jan. 15 (to test the model with only
    TILLAGE_DATES(2,J,K) = 21 !      33 days of data)
    TILLAGE_DATES(3,J,K) = 21
    TILLAGE_DATES(4,J,K) = 0
    TILLAGE_DATES(5,J,K) = 0
    TILLAGE_DATES(6,J,K) = 0
    TILLAGE_DATES(7,J,K) = 21
    TILLAGE_DATES(8,J,K) = 0
    TILLAGE_DATES(9,J,K) = 0
  ENDDO
ENDDO ! K = 1, NUM_NODES

DO K = 1, MAX_DAYS
  T_ANIMAL(1,K) = 7.5
  T_ANIMAL(2,K) = 7.5
  T_ANIMAL(3,K) = 1.75
  T_ANIMAL(4,K) = 1.75

  BREATHING_RATE(1,K) = 3.8
  BREATHING_RATE(2,K) = 6.0
  BREATHING_RATE(3,K) = 10.0
  BREATHING_RATE(4,K) = 14.0
  BREATHING_RATE(5,K) = 18.0
  BREATHING_RATE(6,K) = 23.0
  BREATHING_RATE(7,K) = 3.8
  BREATHING_RATE(8,K) = 6.0
  BREATHING_RATE(9,K) = 10.0
  BREATHING_RATE(10,K) = 14.0
  BREATHING_RATE(11,K) = 18.0
  BREATHING_RATE(12,K) = 20.0
  BREATHING_RATE(13,K) = 20.0
  BREATHING_RATE(14,K) = 3.8
  BREATHING_RATE(15,K) = 6.0
  BREATHING_RATE(16,K) = 10.0
  BREATHING_RATE(17,K) = 14.0

```

```

BREATHING_RATE(18,K)= 18.0
BREATHING_RATE(19,K)= 23.0
BREATHING_RATE(20,K)= 3.8
BREATHING_RATE(21,K)= 6.0
BREATHING_RATE(22,K)= 10.0
BREATHING_RATE(23,K)= 14.0
BREATHING_RATE(24,K)= 18.0
BREATHING_RATE(25,K)= 20.0
BREATHING_RATE(26,K)= 20.0

```

```

DO M = 1, NUM_ASL
  WASHING_FRACTION(1,M,K) = 0.4 ! Leafy vega.
  WASHING_FRACTION(4,M,K) = 0.2 ! Fruit
ENDDO

```

```

ANIMAL_FEED_CONSUMPTION(1,6,K) = 9.2 ! Beef cows - fresh forage
ANIMAL_FEED_CONSUMPTION(2,6,K) = 1.5 ! Deer - fresh forage
ANIMAL_FEED_CONSUMPTION(3,6,K) = 0.005 ! Poultry - fresh forage
ANIMAL_FEED_CONSUMPTION(4,6,K) = 0.005 ! Laying hens - fresh forage
ANIMAL_FEED_CONSUMPTION(3,3,K) = 0.090 ! Poultry - grain
ANIMAL_FEED_CONSUMPTION(4,3,K) = 0.090 ! Laying hens - grain

```

```

DAIRY_SOIL_INGESTION(1,K) = 1.0
DAIRY_SOIL_INGESTION(2,K) = 1.0
DAIRY_SOIL_INGESTION(3,K) = 1.5
DAIRY_SOIL_INGESTION(4,K) = 1.5

```

```

ANIMAL_SOIL_INGESTION(1,K) = 0.5
ANIMAL_SOIL_INGESTION(2,K) = 0.1
ANIMAL_SOIL_INGESTION(3,K) = 0.01
ANIMAL_SOIL_INGESTION(4,K) = 0.01

```

```

ENDDO ! K = 1, MAX_DAYS

```

C1.8

```

C
C-- Read the node-to-cd file, store cd's in NCD, status in NODE_STATUS
call get_nodes_to_cd(13, NCD, NUM_NODES, ISTATUS)
if (ISTATUS .ne. 0) then
  TYPE *, ' ERROR accessing node-to-census-division file --'
  TYPE *, ' Program terminating ... '
  STOP
endif

```

```

C - Set node statuses and build linked lists
do J = 1, NUM_NODES
  K = NCD(J)
  if (K .eq. 0) then
    NODE_STATUS(J) = -2
  else
    NODE_STATUS(J) = 0
    LLIST(J) = LHEAD(K)
    LHEAD(K) = J
  endif
enddo

```

```

C
C-- Convert linked lists (LLIST) to sequential lists (CDNODES)
M = 1
DO 8765 K = 1, NUM_CD
  I = LHEAD(K)
  CDSTART(K) = M
  IF (I .GT. 0) THEN
    CDNODES(M) = I ! There is at least 1 node in this CD
    M = M + 1
    I = LLIST(I)
    IF (I .EQ. 0) GO TO 8764 ! Branch if end of linked list
    CDNODES(M) = I
    M = M + 1
  ENDIF
8762
8764

```

```

      GO TO 8762
8764      CONTINUE
      ENDIF
8765      CONTINUE
C-- Store next available sequential list (CDNODE) position in CDSTART(NUM_CD+1)
C    so that we will know how many nodes are in the last CD.
      CDSTART (NUM_CD + 1) = M

C-- *** TEST SECTION - print the linked list array and the sequential array ***
CCC      open(unit=14,file='test_list.out',status='new')
CCC      WRITE(14,800)
CC800    FORMAT('      J  BEG  LIST  J')
CCC      do 123 j = 1, num_nodes
CCC        if (j .le. num_cd+1) then
CCC          write(14,802) j, cdstart(j), cdnodes(j), j
CC802    format(1X, 4I5)
CCC        else
CCC          write(14,803) cdnodes(j), j
CC803    format(11X, 2I5)
CCC        endif
CC123    CONTINUE
CCC      close(unit=14)
C-- ***** END OF TEST SECTION *****

C-- Read pasture season begin and end dates
      open( unit=12,
           .   file='[hdr.rawdata.feed]pasture_seasons.bin',
           .   form = 'unformatted',
           .   shared, status = 'old', readonly )

      read( 12 ) dumchar ! skip over 1-record file header
      read( 12 ) PS DATES
      close ( unit = 12 )

C-- Read feed regime fractions for each CD into CD_REG_FRACS
      open( unit = 12,
           .   file = '[HDR.RAWDATA.FEED]CENSUS_FEED_REGIMES.BIN',
           .   form = 'unformatted',
           .   shared,
           .   status = 'old', readonly)

      read( 12 ) ldum1,idum2,(cdum(j),j=1,6)
      read( 12 ) CD REG FRACS
      close( unit = 12 )

C-- Read milk source-to-creamery matrix
      open(unit=12,
           $   file='[hdr.rawdata.milk]milk_sources.bin',
           $   form='unformatted', shared, status='old', readonly)
      read(12) idum1, idum2
      read(12) CD SOURCE_FRAC
      close(unit=12)

C-- Read milk creamery-to-grocery matrix
      open(unit=12,
           $   file='[hdr.rawdata.milk]milk_rural_pre47.bin',
           $   form='unformatted', shared, status='old', readonly)
      read(12) idum1, idum2
      read(12) cd distr_frac
      close(unit=12)

C-- Read the winter food consumption files
      filespec = '[hdr.rawdata.food]winter45.asc'
      OPEN ( unit=12,
           $   file=filespec,
           $   status= 'OLD', shared, READONLY )

```

```

Do asl_idx = 1, NUM ASL
  read( 12, 120 ) block_ident
120   format(a3)
C-- Make sure we got the block we wanted
  if( block_ident .ne. block_check( asl_idx ) ) then
    write(*,133 ) filespec,
    $       block_check( asl_idx ), block_ident
133   format(' ERROR Reading ',a/
    $       ' Expected block-id ',a,' but got ',a )
    STOP
  endif
  read(12,*) ! Skip line containing no. of survey cases
C-- Read the mean food consumption values for 11 food categories for this ASL
  read( 12,* )
  $       (WINTER FOOD CONSUMPTION (m, asl_idx), m = 1, NUM FOODS)
C-- Skip one line containing std.deviation, 11 lines containing correlation
C matrix, and one blank line following
  do J = 1,13
    read( 12,* )
  enddo
Enddo
Close( unit=12 )

C-- Read the spring food consumption files
filespec = '[hedr.rawdata.food]spring45.asc'
OPEN ( unit=12,
  $     file=filespec,
  $     status= 'OLD', shared, READONLY )
Do asl_idx = 1, NUM ASL
  read( 12, 120 ) block_ident
C-- Make sure we got the block we wanted
  if( block_ident .ne. block_check( asl_idx ) ) then
    write(*,133 ) filespec,
    $       block_check( asl_idx ), block_ident
    STOP
  endif
  read(12,*) ! Skip line containing no. of survey cases
C-- Read the mean food consumption values for 11 food categories for this ASL
  read( 12,* )
  $       (SPRING FOOD CONSUMPTION (m, asl_idx), m = 1, NUM FOODS)
C-- Skip one line containing std.deviation, 11 lines containing correlation
C matrix, and one blank line following
  do J = 1,13
    read( 12,* )
  enddo
Enddo
Close( unit=12 )

C-- Read the summer food consumption files
filespec = '[hedr.rawdata.food]summer45.asc'
OPEN ( unit=12,
  $     file=filespec,
  $     status= 'OLD', shared, READONLY )
Do asl_idx = 1, NUM ASL
  read( 12, 120 ) block_ident
C-- Make sure we got the block we wanted
  if( block_ident .ne. block_check( asl_idx ) ) then
    write(*,133 ) filespec,
    $       block_check( asl_idx ), block_ident
    STOP
  endif
  read(12,*) ! Skip line containing no. of survey cases
C-- Read the mean food consumption values for 11 food categories for this ASL
  read( 12,* )
  $       (SUMMER FOOD CONSUMPTION (m, asl_idx), m = 1, NUM FOODS)
C-- Skip one line containing std.deviation, 11 lines containing correlation

```



```

C matrix, and one blank line following
      do J = 1,13
        read( 12,*)
      enddo
    Enddo
    Close( unit=12 )

C-- Read the fall food consumption files
    filespec = '[hedr.rawdata.food]fall45.asc'
    OPEN ( unit=12,
      $     file=filespec,
      $     status= 'OLD', shared, READONLY )
    Do asl_idx = 1, NUM ASL
      read( 12, 120 ) block_ident
C-- Make sure we got the block we wanted
      if( block_ident .ne. block_check( asl_idx )) then
        write(*,133 ) filespec,
      $     block_check( asl_idx ), block_ident
        STOP
      endif
      read(12,*) ! Skip line containing no. of survey cases
C-- Read the mean food consumption values for 11 food categories for this ASL
      read( 12,*)
      $ (FALL FOOD CONSUMPTION (m, asl_idx), m = 1, NUM FOODS)
C-- Skip one line containing std.devations, 11 lines containing correlation
C matrix, and one blank line following
      do J = 1,13
        read( 12,*)
      enddo
    Enddo
    Close( unit=12 )

C1.11 C-- Read mean alfalfa cutting dates (3 per year) and mean grass hay cut date
      open( unit=12,
        $ file='[HEDR.RAWDATA.FEED]cutting.dat',
        $ shared, status = 'old', readonly )

      DO K = 1, NUM CD
C--Calc. sum of milk Distribution fractions to this cd from all creameries.
C-- (They will not always add to 1.0 due to non-zero fam. cow and/or local p/d)
      DISTR_SUM = 0.0
      DO J = 1, NUM CREAMERIES
        DISTR_SUM = DISTR_SUM + CD_DISTR_FRAC(J,K)
      ENDDO
      read(12,117) ALF_CUT1, ALF_CUT2, ALF_CUT3, GRASS_CUT
      format( T3, 414 )
117 IF (CDSTART(K+1) .GT. CDSTART(K)) THEN
      NNTOT = CDSTART(K+1) - CDSTART(K) ! No. of nodes in this CD
      DO M = CDSTART(K), CDSTART(K+1) - 1 ! For all nodes in this CD
        NN = CDNODES(M)
        IF (NN .LE. 0 .or. NN .gt. NUM NODES) THEN
          TYPE *, ' Bad node value: ', NN, ' for CD # ', K
          type *, ' Node index position = ', M
          STOP
        ENDIF
        DO J = 1, NUM DAIRY REGIMES
          REGIME_FRACTIONS(J,NN) = CD_REG_FRACS(J,K)
        ENDDO
        DO J = 1, NUM CREAMERIES
          MILK_SOURCE_FRACTION(NN,J) = CD_SOURCE_FRAC(J,K) / NNTOT
        IF (DISTR_SUM .GT. 0.) THEN
          MILK_DISTR_FRACTION (J,NN) = CD_DISTR_FRAC (J,K) / DISTR_SUM
        ELSE
          MILK_DISTR_FRACTION (J,NN) = 0.
        ENDIF
      ENDDO
    ENDDO

```

```

      DO J = 1, NUM_YEARS
        PASTURE_BEGIN(J,NN) = PS_DATES(1,K)
        PASTURE_END(J,NN) = PS_DATES(2,K)
C-- Alfalfa harvests (3 cutting dates)
        HARVEST_DATES(1,5,J,NN) = ALF_CUT1
        HARVEST_DATES(2,5,J,NN) = ALF_CUT2
        HARVEST_DATES(3,5,J,NN) = ALF_CUT3
C-- Grass hay harvest
        HARVEST_DATES(1,9,J,NN) = GRASS_CUT
C-- Grain harvest date
        HARVEST_DATES(1,3,J,NN) = GRASS_CUT + 14
C-- Silage harvest date
        HARVEST_DATES(1,7,J,NN) = PS_DATES(2,K) - 14
      ENDDO
    ENDDO
  ENDIF
  ENDDO
  close ( unit = 12 )
C
C***** Open the time-invariant output file (params.001) *****
  open(unit=14,file='PARAMS.001',form='unformatted',
    $      status='new')
C
C**** Write Records A1 thru A7 (parameters that do not vary with time) ****
C-- Record A1
  write(14) KNUM_NUCS, KNUM_NODES, KNUM_YEARS, KNUM_PLANT,
    $      KNUM_ANIMAL, KNUM_DAIRY_REGIMES, KNUM_ASL,
    $      KNUM_FOODS, KNUM_CREAMERIES, FIRST_YEAR, DAYS_IN_YEAR,
    $      NUM_HARVESTS, NUC_LIST
C-- Record A2
  write(14) XS, XR, LAMBDA_DECAY
C-- Record A3
  write(14) LAMBDA_SPLASH, KSEN, KINT, KG, LAMBDA_TRANS,
    $      TF_MILK, TF_ANIMAL
C-- Record A4
  write(14) NODE_STATUS, LAMBDA_PERC, PS, PR, LAMBDA_LEACH,
    $      REGIME_FRACTIONS
C-- Record A5
  write(14) BMAX, BMIN, LAMBDA_WEATHER, CR
C-- Record A6
  write(14) MILK_SOURCE_FRACTION, MILK_DISTR_FRACTION
C-- Record A7
  write(14) SUBMERSION_SHIELD_FACTOR, GROUNDSHINE_SHIELD_FACTOR,
    $      SUBMERSION_DOSE_FACTOR, GROUNDSHINE_DOSE_FACTOR,
    $      ROOTZONE_DOSE_FACTOR,
    $      INGESTION_DOSE_FACTOR, INHALATION_DOSE_FACTOR
  CLOSE(unit=14) ! Close the time-invariant input param. file (PARAMS.001)
C
C--Now prepare to write each yearly param. file (in this case, only one - 1945)
C
  DO IYEAR = 1, NUM_YEARS
C-- Set up the reference food consumption array to contain daily values
C  according to the season of the year
    DO IDAY = 1, DAYS_IN_YEAR(IYEAR)
C-- NOTE: The end-of-season comparison dates in the following if-then-else
C  block have 6 days added to them to allow for the fact that the
C  year "1945" has the last 6 days of 1944 added to it.
    IF (IDAY.GT. 6 .AND. IDAY.LE. 96) THEN ! winter (Jan/Feb/Mar)
      DO J = 1, NUM_ASL
        DO I = 1, NUM_FOODS
          REF_FOOD_CONSUMPTION(I,J,IDAY) = WINTER_FOOD_CONSUMPTION(I,J)
        ENDDO
      ENDDO
    ELSE IF (IDAY.GT.96 .AND. IDAY.LE.187) THEN ! spring (Apr/May/June)
      DO J = 1, NUM_ASL
        DO I = 1, NUM_FOODS

```

```

      REF FOOD_CONSUMPTION(I,J,IDAY)=SPRING_FOOD_CONSUMPTION(I,J)
    ENDDO
  ENDDO
ELSE IF (IDAY.GT.187 .AND. IDAY.LE.279) THEN ! summer (July/Aug/Sept)
  DO J = 1, NUM ASL
    DO I = 1, NUM FOODS
      REF FOOD_CONSUMPTION(I,J,IDAY)=SUMMER_FOOD_CONSUMPTION(I,J)
    ENDDO
  ENDDO
ELSE ! fall (Oct/Nov/Dec)
  DO J = 1, NUM ASL
    DO I = 1, NUM FOODS
      REF FOOD_CONSUMPTION(I,J,IDAY)=FALL_FOOD_CONSUMPTION(I,J)
    ENDDO
  ENDDO
ENDIF
ENDDO

```

```

C
C-- Create file name for param. file for current year, then open it
CURRENT_YEAR = FIRST_YEAR + IYEAR - 1901
WRITE (FFILE,9001) CURRENT_YEAR
9001 FORMAT('PARAMS',I2,'.001')
OPEN (UNIT=14, FILE=FFILE, FORM='UNFORMATTED', STATUS='NEW')
C-- Write record B1 to output file
write(14) T_FRESH_MILK_FC, T_FRESH_MILK_CR, T_FRESH_MILK_GRO,
$         T_STORED_MILK_GRO, T_ANIMAL, BREATHING_RATE,
$         REF_FOOD_CONSUMPTION, WASHING_FRACTION
DO INODE = 1, NUM_NODES
C-- Do only for nodes with status of zero ---
IF (NODE_STATUS(INODE) .EQ. 0) THEN
  CURRENT_PASTURE_BEGIN = PASTURE_BEGIN (IYEAR, INODE)
  CURRENT_PASTURE_END = PASTURE_END (IYEAR, INODE)
  DO J = 1, NUM_PLANT
    CURRENT_TILLAGE_DATES(J) = TILLAGE_DATES (J, IYEAR, INODE)
    DO I = 1, MAX_HARVESTS
      CURRENT_HARVEST_DATES(I,J)=HARVEST_DATES(I,J,IYEAR,INODE)
    ENDDO
  ENDDO

  DO IDAY = 1, DAYS_IN_YEAR(IYEAR)
    IF (IDAY .LE. 6 .OR. IDAY .GE. CURRENT_PASTURE_END) THEN
      DO K = 1, NUM_PLANT
        DO J = 1, NUM_DAIRY_REGIMES
          DAIRY_FEED_CONSUMPTION(J,K,IDAY)=LATE_FALL_FEED(J,K)
        ENDDO
      ENDDO
    ELSE IF (IDAY .LT. CURRENT_PASTURE_BEGIN) THEN
      DO K = 1, NUM_PLANT
        DO J = 1, NUM_DAIRY_REGIMES
          DAIRY_FEED_CONSUMPTION(J,K,IDAY) = WINTER_FEED(J,K)
        ENDDO
      ENDDO
    ELSE IF (IDAY .LT. CURRENT_PASTURE_BEGIN + 14) THEN
      DO K = 1, NUM_PLANT
        DO J = 1, NUM_DAIRY_REGIMES
          DAIRY_FEED_CONSUMPTION(J,K,IDAY) = SPRING_FEED(J,K)
        ENDDO
      ENDDO
    ELSE IF (IDAY .LT. CURRENT_PASTURE_END - 14) THEN
      DO K = 1, NUM_PLANT
        DO J = 1, NUM_DAIRY_REGIMES
          DAIRY_FEED_CONSUMPTION(J,K,IDAY) = SUMMER_FEED(J,K)
        ENDDO
      ENDDO
    ELSE
      DO K = 1, NUM_PLANT

```

```

        DO J = 1, NUM DAIRY REGIMES
          DAIRY_FEED_CONSUMPTION(J,K,IDAY)=EARLY_FALL_FEED(J,K)
        ENDDO
      ENDDO
    ENDIF
  ENDDO ! for each day in year
C ***** Write out records B2, B3, and B4 for the current location ***
C-- Record B2
      write(14) CURRENT_PASTURE_BEGIN, CURRENT_PASTURE_END,
      $          CURRENT_HARVEST_DATES, CURRENT_TILLAGE_DATES
C-- Record B3
      write(14) ML, VD
C-- Record B4
      write(14) DAIRY_SOIL_INGESTION, ANIMAL_SOIL_INGESTION,
      $          DAIRY_FEED_CONSUMPTION, ANIMAL_FEED_CONSUMPTION
      ENDIF ! if node status is zero
    ENDDO ! for each location (node)
C***** close the param. file for the current year *****
      CLOSE(14)
    ENDDO ! for each year
  stop
end

```

```

C * ' GET_NODES_TO_CD.FOR *****
C
C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: GET_NODES_TO_CD
C
C DESCRIPTION: GET_NODES_TO_CD is a subroutine that reads a file containing
C a mapping (translation) of the Phase I Census Subdivision
C number associated with each of the new model's grid node
C number (on a 31 x 31 spatial grid).
C The routine stores, in an array NCD(1:NUM_NODES), the
C HEDR Phase I census division (c.d.) no. that is associated
C with each node. If the c.d. number is blank (missing),
C a zero is stored in that array position.
C
C LANGUAGE: F77
C
C SUBROUTINE ARGUMENTS:
C
C -- INPUT ARGUMENTS --
C LUNIT is the logical unit number to use to read the node-to-cd file
C NUM_NODES is the dimension (number of nodes) in NCD
C
C -- Output argument:
C NCD(1:NUM_NODES) will contain, for each node, its associated primary
C census division number.
C ISTATUS = 0 if success, -1 if error reading or interpreting input file.
C
C FILE I/O:
C Input Node-to-Census Subdivision file:
C [HEDR.PHASE2.DATA]NODE_TO_CD.ASC
C
C FUNCTIONS/SUBROUTINES CALLED: None.
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Robert A. Burnett
C Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
C
C-----BEGIN DECLARATION BLOCK-----
C
C subroutine get_nodes_to_cd (LUNIT, NCD, NUM_NODES, ISTATUS)
C
C implicit none
C integer LUNIT
C integer NUM_NODES
C integer NCD(NUM_NODES)
C integer ISTATUS
C
C integer nnode, knode, kcd, nch, j
C character*120 string
C
C-----BEGIN EXECUTION BLOCK-----
C
C ISTATUS = 0
C OPEN(UNIT=LUNIT, FILE=' [HEDR.PHASE2.DATA]NODE_TO_CD.ASC',
C $ STATUS='OLD', READONLY, ERR=955)
C DO NNODE = 1, NUM_NODES
C READ(LUNIT, 789, ERR=966, END=977) NCH, (STRING(J:J), J=1, NCH)
C FORMAT(Q, <NCH>A1)

```

```

791      IF (NCH .LT. 28) THEN
          READ (STRING(1:18),791,err=988) KNODE
          FORMAT(10X,18)
          IF (KNODE .NE. NNODE) THEN
              type *, ' Node number out of sequence:'
              type *, '   Line no.: ', NNODE, ' ;      Node no.: ', KNODE
              GO TO 999
          ELSE
              NCD(KNODE) = 0
          ENDIF
      ELSE
          READ (STRING(1:28),793,err=988) KNODE, KCD
          FORMAT(10X,18,7X,13)
          IF (KNODE .NE. NNODE) THEN
              type *, ' Node number out of sequence:'
              type *, '   Line no.: ', NNODE, ' ;      Node no.: ', KNODE
              GO TO 999
          ELSE
              NCD(KNODE) = KCD
          ENDIF
      ENDIF
  ENDDO
C-- Normal (successful) return
  CLOSE(UNIT=LUNIT)
  ISTATUS = 0
  RETURN

C
955      CONTINUE
C-- Error opening file NODE_TO_CD.ASC
  type *, ' Error opening NODE_TO_CD.ASC'
  go to 999

C
966      CONTINUE
C-- Error reading file NODE_TO_CD.ASC
  type *, ' Error reading NODE_TO_CD.ASC - line no.', NNODE
  go to 999

C
977      continue
C-- Error: end-of-file reached on NODE_TO_CD.ASC
  type *,
  $ ' Premature end-of-file reached reading NODE_TO_CD.ASC -'
  type *, ' Attempting to read line no.', NNODE
  go to 999

C
988      continue
C-- Error decoding record from NODE_TO_CD.ASC (re-reading from char. string)
  type *,
  $ ' Error decoding (interpreting) NODE_TO_CD.ASC, line no.', NNODE
C-----
999      CONTINUE
C-- Common error return
  ISTATUS = -1
  CLOSE(UNIT=LUNIT)
  return
end

```

```

C ***** TRANS_AIR.FOR *****
C
C-----BEGIN DEFINITION BLOCK-----
C
C  PROGRAM_NAME:  TRANS_AIR
C
C  DESCRIPTION:  TRANS_AIR is a subroutine that reads a HEDRIC air model
C                results file, transposes it (rearranges the order of the data
C                items) and writes the transposed data to a file that can be
C                more easily and efficiently processed by the HEDRIC
C                environmental accumulation and dose codes.
C
C  LANGUAGE:  F77
C
C  SUBROUTINE ARGUMENTS:
C
C  FILE I/O:
C
C      Input (air model results) file name:  Passed in argument AIR_IN.
C      Input air transport data file contains all available
C      data for one month for all locations.
C
C      Transposed Air Model Output file name:  Passed in argument AIR_OUT.
C      This routine creates an unformatted binary output file; the
C      contents of this created file are as follows:
C
C      First record:  number_of_nodes,  number_of_days
C
C      Second record:  (air_conc(i), i=1,number_of_days),
C                     (surf_dep(i), i=1,number_of_days)    for node 1
C
C      Third record:  (air_conc(i), i=1,number_of_days),
C                     (surf_dep(i), i=1,number_of_days)    for node 2
C
C      .
C      .
C      .
C      Last record:  (air_conc(i), i=1,number_of_days),
C                     (surf_dep(i), i=1,number_of_days)    for last node
C
C  FUNCTIONS/SUBROUTINES CALLED:  None.
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C  AUTHOR:  Robert A. Burnett
C           Battelle, Pacific Northwest Laboratory
C  DATE:  7/15/91
C
C  MODIFICATION HISTORY:
C
C-----BEGIN DECLARATION BLOCK-----
C
C      subroutine TRANS_AIR ( AIR_IN, AIR_OUT, AIR_DAYS )
C      implicit none
C
C-- INPUT ARGUMENTS: --
C      character*(*) AIR_IN    ! Name of input air transport file
C      character*(*) AIR_OUT    ! Name of output (transposed) air transport file
C-- OUTPUT ARGUMENTS: --
C      integer*4 AIR_DAYS      ! Number of days of data in the input air file.
C
C      integer iday, i, j, k, num_x, num_y, num_nodes
C      integer max_days
C      parameter (max_days = 33)
C      parameter (num_x = 31)
C      parameter (num_y = 31)

```

```

parameter (num_nodes = num_x * num_y)
real tici(num_x, num_y), sci(num_x, num_y)
real xtici(max_days, num_x, num_y), xscl(max_days, num_x, num_y)
character*70 outstr1, outstr2

```

```

C
C----- BEGIN EXECUTION BLOCK -----
C
      open (unit=15, file=AIR IN, status='old',
$         form='unformatted', readonly)
C- Read general file header record
      read(15) outstr1
      iday = 0
1      continue
      read(15,end=99) outstr1
      read(15) tici
      read(15) outstr2
      read(15) sci
      if (iday .eq. max_days) go to 98
      iday = iday + 1
      do j = 1, num_y
        do i = 1, num_x
          xtici(iday,i,j) = tici(i,j)
          xscl (iday,i,j) = sci (i,j)
        enddo
      enddo
      go to 1
C
98      continue
      type *,
$      ' Reached max. no. of days: ',max_days,' before reaching EOF'
      go to 100
C
99      continue
      type *, ' End of air transport file reached after ',iday,
$      ' days of data'
100     continue
      close(unit=15)
      open(unit=16, file=AIR OUT, status='new',
$         form='unformatted')
C-- Write first record: no. of locations, no. of days
      write(16) num_nodes, iday
C-- Now write the data records on a node-by-node basis, each record containing
C   data for all days of the year for a single node (location).
      do j = 1, num_y
        do i = 1, num_x
C-- Write air concentration and surface deposition
          write(16) (xtici(k,i,j), k = 1,iday),
$                 (xscl (k,i,j), k = 1,iday)
        enddo
      enddo
      close(unit=16)
      AIR_DAYS = iday
      return
      end

```


APPENDIX C2

DESCARTES CODE LISTINGS

C2.1

```

0001 C-----BEGIN DEFINITION BLOCK-----
0002 C
0003 C PROGRAM_NAME: DESCARTES
0004 C
0005 C
0006 C DESCRIPTION: DESCARTES is the main program of the environmental pathways
0007 C model.
0008 C
0009 C LANGUAGE: F77
0010 C
0011 C FILE I/O:
0012 C Input Parameter file : PARAMS.XXX (XXX is run number)
0013 C Input Parameters that vary with year : PARAMSYY.XXX
0014 C (YY is year,XXX is run number)
0015 C Air Model Output file :I-131 AIRYY.XXX (YY is year,XXX is run number)
0016 C Biomass Output file : BIOMASSYY.XXX (YY is year,XXX is run number)
0017 C Environmen Concentrations Output file : I-131 LLLL.YY (LLLL is output
0018 C location, YY is year)
0019 C
0020 C FUNCTIONS/SUBROUTINES CALLED:
0021 C
0022 C Read_user_input
0023 C Read_time_inv
0024 C Read_loc_inc
0025 C Read_loc_var
0026 C Read_air_model
0027 C Biomass_growth_rate
0028 C Calc_plant_conc
0029 C Animal_products_conc
0030 C Milk_conc
0031 C Write_conc_to_file
0032 C Write_creamery
0033 C Write_milk
0034 C
0035 C-----BEGIN IMPLEMENTATION BLOCK-----
0036 C AUTHOR: Nils Reitz
0037 C Battelle, Pacific Northwest Laboratory
0038 C DATE: 7/15/91
0039 C
0040 C MODIFICATION HISTORY:
0041 C
0042 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0043 C
0044 C PROGRAM DESCARTES
0045 C IMPLICIT NONE
0046 C
0047 C-----BEGIN DECLARATION BLOCK-----
0048 C INCLUDE 'ENVIR.CMN'
0169 C INCLUDE 'USER_INPUT.CMN'
0180 C INCLUDE 'CALC_CONC.CMN'
0193 C
0194 C INTEGER YEAR,RUN,CREAMERY
0195 C INTEGER NUCLIDE,DAYS_IN_CURRENT_YEAR, LAST_JULY_DAY
0196 C INTEGER LOCATION,DAY
0197 C CHARACTER*3 CRUN,FILENAME*45
0198 C INTEGER AUN,BUN,CUN,DUN,EUN,LIB$GET_LUN,LIB$FREE_LUN,ISTAT
0199 C REAL TIAR,SC

```

```

0200 CHARACTER*4 CYEAR,CLOC
0201 INTEGER TYEAR,I,J,K,PLANT
0202 INTEGER A,B,CHECK
0203
0204 C-----BEGIN EXECUTABLE CODE-----
0205 C
0206 C CALCULATE DATA FOR ONLY THOSE NODES WHICH FEED NODE 137 DATA
0207 C DO I=1,TLOCMA
0208 C MILK_LIST(I)=.FALSE.
0209 C ENDDO
0210 C MILK_LIST(69)=.TRUE.
0211 C MILK_LIST(70)=.TRUE.
0212 C MILK_LIST(71)=.TRUE.
0213 C MILK_LIST(72)=.TRUE.
0214 C MILK_LIST(100)=.TRUE.
0215 C MILK_LIST(101)=.TRUE.
0216 C MILK_LIST(102)=.TRUE.
0217 C MILK_LIST(103)=.TRUE.
0218 C MILK_LIST(104)=.TRUE.
0219 C MILK_LIST(132)=.TRUE.
0220 C MILK_LIST(133)=.TRUE.
0221 C MILK_LIST(134)=.TRUE.
0222 C MILK_LIST(135)=.TRUE.
0223 C MILK_LIST(137)=.TRUE.
0224 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0225
0226 CALL READ_USER_INPUT
0227 DO RUN=1,RUNMAX
0228 WRITE(CRUN,100) RUN
0229 100 FORMAT(I3.3)
0230 FILENAME=' [HEDR,PHASE2,PARAMS]PARAMS.'//CRUN
0231
0232 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0233 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0234 ISTAT = LIB$GET_LUN (AUN)
0235 OPEN(UNIT=AUN,FILE=FILENAME,FORM='UNFORMATTED',STATUS='OLD')
0236 CALL READ_TIME_INV(AUN)
0237 CLOSE(UNIT=AUN)
0238 ISTAT = LIB$FREE_LUN (AUN)
0239 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0240 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0241
0242 DO NUCLIDE=1,NUCMAX
0243 DO YEAR=1,NUM_YEARS
0244 TYEAR=YEAR+1944
0245 WRITE(CYEAR,90) TYEAR
0246 90 FORMAT(I4)
0247 FILENAME='BIOMASS'//CYEAR(3:4)//'.'//CRUN
0248 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0249 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0250 ISTAT = LIB$GET_LUN (BUN)
0251 IF (NUCLIDE.GT.1) THEN
0252 OPEN(UNIT=BUN,FILE=filename,form='unformatted',
0253 * STATUS='OLD')
0254 ELSEIF (NUCLIDE.EQ.1.AND.NUCMAX.GT.1) THEN
0255 OPEN(UNIT=BUN,FILE=filename,form='unformatted',
0256 * STATUS='NEW')

```

```

0257      ENDIF
0258      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0259      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0260
0261      FILENAME=' [HEDR.PHASE2.DATA]' //
0262      *      NUCLIST(NUCLIDE) //' _AIR' //CYEAR(3:4) //
0263      *      ' .' //CRUN
0264
0265      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0266      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0267      ISTAT=LIB$GET_LUN(CUN)      ! AIR MODEL RESULTS
0268      OPEN(UNIT=CUN,FILE=FILENAME,STATUS='OLD',
0269      *      FORM='UNFORMATTED')
0270      FILENAME=' [HEDR.PHASE2.PARAMS]PARAMS'
0271      *      //CYEAR(3:4) //' _' //CRUN
0272      ISTAT = LIB$GET_LUN (DUN)
0273      OPEN(UNIT=DUN,FILE=FILENAME,FORM='UNFORMATTED',
0274      *      STATUS='OLD')
0275      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0276      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0277
0278      CALL READ_LOC_INV(DUN)
0279
0280      C      SET MILK CONC IN CREAMERIES TO ZERO
0281      DO I=1,NUM_CREAMERIES
0282      DO J=1,DAYS_IN_CURRENT_YEAR
0283      CREAMERY_CONC(I,J)=0
0284      ENDDO
0285      ENDDO
0286
0287      C      LAST JULY DAY=DAYS_IN_YEAR(YEAR) - 153
0288      DAYS_IN_CURRENT_YEAR=DAYS_IN_YEAR(YEAR)
0289      CHECK=0
0290      DO LOCATION=1,LOC_MAX
0291      IF (NODE_STATUS(LOCATION).EQ.0) THEN
0292      IF (LOC_LIST(LOCATION)) THEN
0293
0294      WRITE(CLOC,101) LOCATION
0295      101      FORMAT(I3.3)
0296      C      IF (LOCATION.LT.10) THEN
0297      *      FILENAME=NUCLIST(NUCLIDE) //' _' //CLOC//
0298      *      ' .' //CYEAR(3:4)
0299
0300      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0301      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0302      ISTAT=LIB$GET_LUN(EUN)
0303      IF (RUN.EQ.1) THEN
0304      OPEN(UNIT=EUN,FILE=filename,access='direct',form='unformatted',
0305      $      RECL=128,STATUS='new')
0306      ELSE
0307      OPEN(UNIT=EUN,FILE=filename,access='direct',form='unformatted',
0308      $      RECL=128,STATUS='OLD')
0309      ENDIF
0310      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0311      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0312
0313      ENDIF ! IN LOCATION LIST

```

C2.4

```

0314      DO A=1,TNUM HARVESTS
0315      DO B=1,TPLANTMAX
0316      PREV_HARVEST(A,B)=HARVEST_DATES(A,B)
0317      ENDDO
0318      ENDDO
0319      CALL READ_LOC_VAR(DUN)
0320      IF (NUCLIDE.GT.1) THEN
0321      DO I=1,PLANTMAX
0322      DO J=1,DAYS_IN_YEAR(YEAR)
0323      READ(BUN) BIOMASS(I,J), GROWTH(I,J)
0324      ENDDO
0325      ENDDO
0326      ENDIF
0327      CHECK=CHECK+1
0328      CALL READ_AIR_MODEL(CUN,CHECK)
0329      IF (YEAR.EQ.1) THEN
0330      C      ???SET INITIAL CONDITIONS FOR TIME_INTEGRATED VARIABLES FOR
0331      C      CURRENT_NUCLIDE?????
0332      ENDIF
0333      C      IF (MILK_LIST(LOCATION)) THEN
0334      DO DAY=1,DAYS_IN_CURRENT_YEAR
0335      DO PLANT=1,PLANTMAX
0336      IF (NUCLIDE.EQ.1) THEN
0337      CALL BIOMASS_GROWTH_RATE(YEAR,
0338      *      LOCATION,DAY,PLANT)
0339      ENDIF
0340      C      IF (DAY.GE.tillage_dates(plant,year,location)) THEN
0341      CALL CALC_PLANT_CONC(PLANT,YEAR,LOCATION,DAY,NUCLIDE)
0342      C      ENDIF
0343      ENDDO ! PLANT
0344      CALL ANIMAL_PRODUCTS_CONC(NUCLIDE,YEAR,
0345      *      LOCATION,DAY)
0346      CALL MILK_CONC(NUCLIDE,YEAR,LOCATION,DAY)
0347
0348      C      USING MILK ACCUM MATRIX OF PRODUCTION LOCATION VS CREAMERIES
0349      C      ETC.
0350
0351      DO I=1,TNUM_CREAMERIES
0352      CREAMERY_CONC(I,DAY)=CREAMERY_CONC(I,DAY)+
0353      $      MIXED_MILK_CONC(DAY,LOCATION)*
0354      $      MILK_SOURCE_FRACTION(LOCATION,I)
0355      ENDDO
0356
0357
0358      C      type *,'day=',day
0359      ENDDO ! DAY OF YEAR
0360
0361      DO I=1,TPLANTMAX
0362      PREV_INTEG(I,LOCATION,1)=BIOMASS(I,DAYS_IN_YEAR(YEAR))
0363      DO J=2,5
0364      PREV_INTEG(I,LOCATION,J)=PLANT_CONC(I,
0365      $      DAYS_IN_YEAR(YEAR),(J-1))
0366      ENDDO
0367      ENDDO
0368      DO I=1,TPLANTMAX
0369      DO J=1,NUM_HARVESTS(I)
0370      IF (DAY.GE.HARVEST_DATES(J,I)) THEN

```

```

0371       IF (EATEN(I)) THEN
0372             HARV_CONC(I,J)-PLANT_CONC(I,HARVEST_DATES(J,I),3)
0373             $                               +PLANT_CONC(I,HARVEST_DATES(J,I),4)
0374       ELSE
0375             HARV_CONC(I,J)-PLANT_CONC(I,HARVEST_DATES(J,I),3)
0376       ENDIF
0377   ENDIF
0378       ENDDO
0379   ENDDO
0380
0381   C????????????????????????????????????????????????????????????????????????????????????
0382       IF (LOCLIST(LOCATION)) THEN
0383             CALL WRITE_CONC_TO_FILE(YEAR, LOCATION, EUN)
0384
0385   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0386   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0387       CLOSE (UNIT=EUN)
0388       ISTAT=LIB$FREE LUN(EUN)
0389   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0390   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0391
0392       ENDIF
0393
0394       IF (NUCLIDE.EQ.1.AND.NUCMAX.GT.1) THEN
0395             DO I=1, PLANTMAX
0396                   DO J=1, DAYS_IN_YEAR(YEAR)
0397                         WRITE(BUN) BIOMASS(I,J), GROWTH(I,J)
0398                   ENDDO
0399             ENDDO
0400       ENDIF
0401
0402   C       ENDIF !IN milk list
0403       endif ! node status
0404       ENDDO ! LOCATION
0405
0406   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0407   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0408       CLOSE (UNIT=CUN)
0409       ISTAT=LIB$FREE LUN(CUN)
0410       CLOSE (UNIT=DUN)
0411       ISTAT=LIB$FREE LUN(DUN)
0412   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0413   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0414
0415       IF (NUCLIDE.EQ.1.AND.NUCMAX.GT.1) THEN
0416   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0417   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0418       CLOSE (UNIT=BUN)
0419       ISTAT=LIB$FREE LUN(BUN)
0420   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0421   C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0422       ENDIF
0423
0424       DO CREAMERY=1, NUM_CREAMERIES
0425             CALL WRITE_CREAMERY(RUN, YEAR, NUCLIDE, CREAMERY)
0426       ENDDO
0427

```

```
0428
0429      DO LOCATION=1,TLOCMA
0430      DO DAY=1,DAYS IN CURRENT YEAR
0431      DO CREAMERY=1,TNUM CREAMERIES
0432      RURAL_CONC(LOCATION, DAY)=CREAMERY CONC(CREAMERY, DAY) *
0433      $      MILK_DISTR_FRACTION(CREAMERY, LOCATION)
0434      ENDDO
0435      ENDDO
0436      IF (LOCLIST(LOCATION)) THEN
0437      *      FILENAME=NUCLIST(NUCLIDE) //'_'//CLOC//
0438      *      '._'//CYEAR(3:4)
0439
0440      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0441      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0442      ISTAT=LIB$GET_LUN(EUN)
0443      OPEN(UNIT=EUN, FILE=filename, access='direct', form='unformatted',
0444      $      RECL=128, STATUS='OLD')
0445
0446      CALL WRITE_MILK(EUN, YEAR, LOCATION)
0447      CLOSE(UNIT=EUN)
0448      ISTAT=LIB$FREE_LUN(EUN)
0449      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0450      C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
0451      ENDF  ! IN LOCATION LIST
0452
0453      ENDDO ! LOCATION
0454      ENDDO ! YEAR
0455      ENDDO ! NUCLIDE
0456      C      DELETE BIOMASS FILES FOR THIS RUN
0457      ENDDO ! RUN
0458      C CLOSE ALL FILES
0459      END
```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: READ_USER_INPUT
C
C DESCRIPTION: Prompts user for number of runs and output locations.
C               model.
C
C LANGUAGE: F77
C
C FILE I/O: NONE
C
C FUNCTIONS/SUBROUTINES CALLED: NONE
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C         Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Subroutine read_user_input
implicit none

C-----BEGIN DECLARATION BLOCK-----
include 'envir.cmn'
include 'user_input.cmn'

integer out
character*3 answer
integer lcount,i

C-----BEGIN EXECUTABLE CODE-----
      lcount=0
      type *, ' Enter number of runs: '
      read (*,*) runmax
      if (runmax.lt.1) goto 9
      answer= ' '
10     type *, ' Do you want output for all locations ? '
      read(*,101,err=10,end=10) answer
101    format(a3)
      if (answer(1:1).ne.'y'.and.answer(1:1).ne.'Y'.and.
         answer(1:1).ne.'n'.and.answer(1:1).ne.'N') goto 10
      if (answer(1:1).eq.'y'.or.answer(1:1).eq.'Y') then
         do i=1,tlocmax
            loclist(i)=.true.
         enddo
         lcount=tlocmax
      else
11     type *, ' Enter at least one output location : '
         type *, ' Press ctrl-Z to end list: '
         read (*,*,end=12) out
         if ((out.ge.1).and.(out.le.tlocmax)) then
            loclist(out)=.true.
            lcount=lcount+1
         else
            type *, 'Invalid location ',out
            endif
         goto 11
         endif
12     continue
         if (lcount.eq.0) goto 11

      Return

```



```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_time_inv
C
C DESCRIPTION: Read input parameters from the input parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C   lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file Params.XXX (XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      Subroutine read_time_inv(lun)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

```

      Include 'envir.cmn'

```

```

      Integer    lun          !logical unit assigned input file
      Integer    i

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

      read( lun) nucmax,locmax,num_years,plantmax,apmax,num_dr,
      .          num_asl,num_foods,num_creamerles,first_year,
      .          days_in_year,num_harvests,nuc1ist

```

```

      num_days=0
      Do i=1,num_years
        num_days=num_days + days_in_year(i)
      Enddo

```

```

      read( lun) xs,xr,ldecay
      read( lun) lsplash,ksen,kint,kg,ltrans,tfmilk,tf_animal
      read( lun) node_status,lperc, ps,pr,lleach,regime_fractions
      read( lun) bmax,bmin,lweather,cr
      read( lun) milk_source_fraction,milk_distr_fraction
      read( lun) sshield_factor,gshield_factor,
      .          submersion_dose_factor,groundshine_dose_factor,
      .          rootzone_dose_factor,
      .          ingestion_dose_factor,inhalation_dose_factor
      return
      end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_loc_inv
C
C DESCRIPTION: Read location invariant parameters from the input
C               parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C   lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file ParamsYY.XXX (YY is year, XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C         Bettelle, Pacific Northwest Laboratory
C
C DATE:12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Subroutine read_loc_inv(lun)
      implicit none

C-----BEGIN DECLARATION BLOCK-----
      Include 'envlr.cmn'

      Integer      lun          !logical unit assigned input file

C-----BEGIN EXECUTABLE CODE-----
      read(lun) tfresh_milk_fc, tfresh_milk_cr,
      .         tfresh_milk_gro, tstored_milk_gro, t_animal,
      .         breathing_rate, ref_food_consumption, fwash

      return
      end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_loc_var
C
C DESCRIPTION: Read location variant parameters from the input parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C   lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file ParamsYY.XXX (YY is year, XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      Subroutine read_loc_var(lun)
      Implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----
      Include 'envir.cmn'.

```

```

      Integer      lun      !logical unit assigned input file

```

```

C-----BEGIN EXECUTABLE CODE-----
      read(lun) pasture_begin,pasture_end,harvest_dates,
      .      tillage_dates
      read(lun) ml,vd
      read(lun) dairy_soil_ingestion,animal_soil_ingestion,
      .      dairy_feed_consumption,animal_feed_consumption
      return
      end

```

C2.10

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_air_model
C
C DESCRIPTION: Reads input parameters from the air model.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   aun - logical unit number of file I/O
C
C FILE I/O: Reads binary file I-131_airYY.XXX (YY is year, XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   Subroutine read_air_model(aun,check)
C   implicit none
C
C-----BEGIN DECLARATION BLOCK-----
C   include 'envir.cmn'
C
C   integer aun,loc
C   integer nnodes,nday
C   integer check
C
C-----BEGIN EXECUTABLE CODE-----
C
C   if (check.eq.1) then
C     read(aun) nnodes,nday
C   endif
C   read(aun) air_conc,surface_conc
C
C   return
C   end

```

C2.11

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: biomass_growth_rate
C
C DESCRIPTION: Calculate the biomass and growth rate for each
C               plant type, loc and day of a given year
C
C LANGUAGE: F77
C
C PARAMETERS:
C
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C   day - current day
C   plant - current plant type
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C         Battelle, Pacific Northwest Laboratory
C
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      Subroutine biomass_growth_rate(year,loc,day,plant)
      implicit none

```

C2.12

```

C-----BEGIN DECLARATION BLOCK-----
      include 'envir.cmn'
      include 'user_input.cmn'
      include 'calc_conc.cmn'

      integer year,plant,loc,day
      real tbiomasa,h,tgrate
      real pi
      real bmax(tpiantmax)
      integer last_july_day
      data pi/3.14159/
      integer i,harday
      integer days_in_current_year,tday
      logical hdate,growing

C-----BEGIN EXECUTABLE CODE-----
      last_july_day=days_in_year(year) - 153
      days_in_current_year=days_in_year(year)
      if (node_status(loc).eq.0) then
C         if (milk_list(loc)) then

          hdate=.false.
          growing=.false.
          h=1.0
          tday=day-1

          harday=harvest_dates(1,plant)
          do i=1,3
            if (harvest_dates(i,plant).eq.day) then
              hdate=.true.
            endif
          enddo

          f (replicated(plant).and.(day.lt.tillage_dates(plant)))

```

```

        tbiomass=bmin(plant, loc)
        tgrate=0
        growing=.false.
    elseif (replanted(plant).and.(harday.ne.0).and.
$      (day.ge.harday)) then
        tbiomass=bmin(plant, loc)
        tgrate=0
        growing=.false.
    elseif (.not.replanted(plant).and.hdate) then
        tbiomass=bmin(plant, loc)
        growing=.true.
    elseif (day.eq.1.and.year.eq.1) then
        tbiomass=bmin(plant, loc)
        growing=.true.
    elseif (day.eq.1.and.year.ne.1) then
        tbiomass=prev_integ(plant, loc, 1)
        growing=.true.
    else
        tbiomass=biomass(plant, tday)
        growing=.true.
    endif

    if (growing) then
        if (day.le.last_july_day) then
            tgrate=kg(plant)/2*(1-cos(2*pi*day/days_in_current_year))
$          *tbiomass * ((bmax(plant, loc)-tbiomass)
$          /bmax(plant, loc))
            else
                bsmax(plant)=(kg(plant)*bmax(plant, loc)*
$                  bmax(plant, loc))/
$                  (bmax(plant, loc)*
$                  (kg(plant)-ksen(plant))+ksen(plant)*
$                  bmin(plant, loc))
                tgrate=kg(plant)/2*(1-cos(2*pi*day/days_in_current_year))
$              * tbiomass *
$              ((bsmax(plant)-tbiomass)/bsmax(plant))-
$              ksen(plant)*(tbiomass-bmin(plant, loc))
            endif
        endif
        Biomass(plant, day)=tbiomass+h*tgrate
        Growth(plant, day)=tgrate

    endif          ! DATA AVAILABLE FOR THAT LOC

return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Calc_Plant_conc
C
C DESCRIPTION: Calculate Plant concentrations of nuclide
C
C LANGUAGE: F77
C
C PARAMETERS:
C   plant - current plant type
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C   day - current day
C   nuclide- current nuclide (assumed I_131)
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      SUBROUTINE CALC_PLANT_CONC(PLANT,YEAR,LOC,DAY,NUCLIDE)
      implicit none

```

```

C2.14 C-----BEGIN DECLARATION BLOCK-----
      include 'envir.cmn'
      include 'user_input.cmn'
      include 'calc_conc.cmn'

      integer plant,NUCLIDE
      integer year,loc,day
      real Y(4),dydt(4),yout(4)
      integer tday

      integer days_in_current_year
      integer valid_rec
      integer planting_date,harday
      integer h,i,t,nt,n,jplant,jloc,jday
      logical calc_conc,hdate

      data h/1/          ! one day time step
      data n/4/          ! solve four area conc of plant

```

```

C-----BEGIN EXECUTABLE CODE-----
      days_in_current_year=days_in_year(year)
c      IF (TILLAGE_DATES(PLANT).GT.DAYS_IN_CURRENT_YEAR)
c      $      GOTO 15

```

```

C Call plant_deriv to get dydt for nt=0
      tday=day-1
      harday=harvest_dates(1,plant)
      do i=1,3
         if (harvest_dates(i,plant).eq.day) then
            hdate=.true.
         endif
      enddo

```

```

if (replanted(plant).and.(day.lt.tillage_dates(plant))) then
  calc conc=.false.
  if (day.eq.1.and.year.eq.1) then
    do i=1,2
      y(i)=0
    enddo
  elseif (day.eq.1.and.year.ne.1) then
    do i=1,2
      y(i)=prev_integ(plant,loc,i)
    enddo
  else
    do i=1,2
      y(i)=plant_conc(plant,tday,1)
    enddo
  endif
  do i=3,4
    y(i)=0
  enddo
elseif (replanted(plant).and.(harday.ne.0).and.
$ (day.ge.harday)) then
  calc conc=.false.
  do i=1,2
    y(i)=plant_conc(plant,tday,1)
  enddo
  do i=3,4
    y(i)=0
  enddo
elseif (.not.replanted(plant).and.hdate) then
  calc conc=.false.
  do i=1,2
    y(i)=plant_conc(plant,tday,1)
  enddo
  do i=3,4
    y(i)=(plant_conc(plant,tday,i) * bmin(plant,loc))/
$ biomass(plant,tday)
  enddo
else ! plant is growing
  calc conc=.true.
  if (day.eq.1.and.year.eq.1) then
    do i=1,4
      y(i)=0
    enddo
  elseif (day.eq.1.and.year.ne.1) then
    do i=1,4
      y(i)=prev_integ(plant,loc,i)
    enddo
  else
    do i=1,4
      y(i)=plant_conc(plant,tday,i)
    enddo
  endif
endif

if (day.eq.tillage_dates(plant)) then
  y(1)=(xs*plant_conc(plant,day,1) +
$ xr*plant_conc(plant,day,2))/
$ (xs+xr)
  y(2)=y(1)
endif

```

- C Calculate deposition and interception fractions
 $fd(PLANT) = EXP(-k_{int}(plant) * biomass(plant, day))$
 $fi(PLANT) = 1 - fd(PLANT)$
 $t = t + h$
- C Call Yderiv for dYdt values at current time


```

                                nuclide,calc_conc)
C  Call subroutine plant_integ to solve the B, Q differential equations
    CALL plant_integ(Y,dYdt,n,h,Yout,calc_conc)
C  Move new values of variables to array Y
    DO 20 I = 1, n
        Y(I) = Yout(I)
20    CONTINUE
C  Write new values to output file
    Do i=1,4
        plant_conc(plant,day,I)=Y(I)
    Enddo

15    RETURN
    END

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Plant_derlv
C
C DESCRIPTION: Calculates the concentration rate in the upper soil, root zone,
C               surface vegetation, and tissue vegetation of a plant.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   plant - current plant type
C   loc - current location
C   day - current day
C   Y Vector of functions solutions at previous time step
C   dYdt Vector of function derivatives calculated by Yderiv
C   year - current year (ie. year=1 implies 1945)
C   nuc_num - current nuclide (assumed I-131)
C   calc_conc - logical defining whether concentration in upper soil
C               or root zone is calculated for current plant
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C         Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

C2.17

```

SUBROUTINE Plant_Deriv(plant,loc,day,Y,dYdt,
-                       year,nuc_num,CALC_CONC)
  Implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----
C INCLUDE 'envir.cmn'
C INTEGER plant, loc, day_count, year
C Integer I, day, nuc_num
C REAL Y(5), dYdt(5), tcr, t
C Real rsenci, rsenco, rsesus, sc, rroot
C Integer tyear, last_july_day
C LOGICAL CALC_CONC

```

```

C-----BEGIN EXECUTABLE CODE-----
C IF (growth(plant,day).LT.0) THEN
C   tcr = 0.0
C ELSE
C   tcr = cr(plant,loc,nuc_num)
C ENDIF

```

```

C ML(MASSLOADING,VD,P1 And pr(loc) ARE FROM INPUT PARM FILE)
C RSESUS=Y(2)*ML(day)*VD(day)/PS(LOC)
C last_july_day=days in year(year) - 153
C if (day.le.last_july_day) then
C   rsenci=0
C   rsenco=0
C   rroot=Y(2)*(tcr/pr(loc))*growth(plant,day)
C else
C   rsenci=(Y(4)/biomass(plant,day))
C   - *ksen(plant)*(biomass(plant,day)- bmin(plant,loc))
C   rsenco=(Y(3)/biomass(plant,day))
C   - *ksen(plant)*(biomass(plant,day)- bmin(plant,loc))
C   rroot=Y(2)*(tcr/pr(loc))*(growth(plant,day)+ksen(plant)*

```

```

endif
C Calculate dQus(p,t)=dYdt(1); Qus(p,t)=Y(1)
  IF (CALC_CONC) THEN
    dYdt(1) = surface_conc(day)*fd(plant)
    - Y(1)*(lperc(loc)+ldecay(nuc_num)+lsplash(plant))
    - Y(3)*(lweather(plant,loc,nuc_num))-rsesus+rsenco
    - rsenci
C Calculate dQrz(p,t)=dYdt(2); Qrz(p,t)=Y(2)
    dYdt(2) = Y(1)*lperc(loc)
    - Y(2)*(ldecay(nuc_num)+lleach(loc,nuc_num))
    - rroot
    ENDIF
C Calculate dQov(p,t)=dYdt(3); Qov(p,t)=Y(3)
    dYdt(3) = surface_conc(day)*fi(plant)
    - Y(3)*(lweather(plant,loc,nuc_num)
    - +ldecay(nuc_num)+ltrans(plant,nuc_num))
    - + Y(1)*(lsplash(plant))+Rsesus
    - rsenco
C Calculate dQiv(p,t)=dYdt(4); Qiv(p,t)=Y(4)
    dYdt(4) = Y(3)*ltrans(plant,nuc_num)
    - Y(4)*ldecay(nuc_num)
    - + rroot - Rsenci
  RETURN
END

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C  PROGRAM_NAME: Plant_integ
C
C  DESCRIPTION: The subroutine EULER uses a simple Euler method to advance the
C                solution over an interval h and return the incremented variable as Yout
C
C  LANGUAGE:  F77
C
C  PARAMETERS:
C    Y      Vector of functions solutions at previous time step
C    dydt   Vector of function derivatives calculated by Yderiv
C    n      Number of functions to be solved
C    h      Time step = 1 day
C    Yout   Vector of function solutions at the new time
C    calc_conc logical defining whether concentration in upper soil
C              or root zone is calculated for current plant
C
C  FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C  AUTHOR: Nila Reitz
C          Battelle, Pacific Northwest Laboratory
C  DATE: 12/31/91
C
C  MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE Plant_Integ(Y,dydt,n,h,Yout,CALC_CONC)
IMPLICIT NONE

C-----BEGIN DECLARATION BLOCK-----
C
C    INTEGER n,H,I
C    REAL Y(n), dydt(n), Yout(n)
C    LOGICAL calc_conc
C
C-----BEGIN EXECUTABLE CODE-----
C
C    DO 14 I=1,n    !biomass already solved
C      Yout(I) = Y(I) + h*dydt(I)
14  CONTINUE
C    IF (CALC_CONC) THEN
C      YOUT(I)=Y(1)
C      YOUT(2)=Y(2)
C    ENDIF
C    RETURN
C    END

```

```

C=====BEGIN DEFINITION BLOCK=====
C
C PROGRAM_NAME: Animal_products_conc
C
C DESCRIPTION: Calculate Animal Product concentrations of I-131
C
C LANGUAGE: F77
C
C PARAMETERS:
C   nucnumb - current nuclide (assumed I-131)
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C   day - current day
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

SUBROUTINE animal_products_conc(nucnumb,year,loc,day)
Implicit none
include 'envir.cmn'
include 'calc_conc.cmn'
include 'user_input.cmn'

```

```

Real ap_quantity(tapmax,371)
Real ap_conc(tapmax,371)
Real total_consum(tapmax)
Integer h,i,j,day,num_days,year
Integer days_of_decay(3),loc
Integer nyear,nucnumb,plant,m
Real tplant_conc(3)

```

```

do i=1,apmax
  do j=1,tnum_days
    tap_conc(i,j)=0
  enddo
enddo

```

```

do i=1,apmax
  total_consum(i)=0
enddo

```

```

do i=1,apmax
  do j=1,plantmax
    total_consum(i)=total_consum(i)+
$      animal_feed_consumption(i,j,day)
  enddo
enddo

```

```

days_of_decay(1)=0
nyear=year-1
do h=1,apmax
  do plant=1,plantmax
    do m=1,3
      tplant_conc(m)=0
      days_of_decay(m)=0
    enddo
  enddo
enddo

```

```

if (plant.eq.1.or.plant.eq.2.or.plant.eq.4..or.plant.eq.8) then
  goto 10
elseif (plant.eq.3) then
  if (day.lt.harvest_dates(1,plant)) then
    if (nyear.eq.0) then
      tplant_conc(1)=0
    else
      tplant_conc(1)= harv_conc(plant,1)
      days_of_decay(1)=(days_in_year(nyear)-
        prev_harvest(1,plant)) +
$      day
$    endif
  else
    tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3) +
$    plant_conc(plant,harvest_dates(1,plant),4)
    days_of_decay(1)=day-harvest_dates(1,plant)
  endif
elseif (plant.eq.5) then
  if (day.lt.harvest_dates(1,plant)) then
    if (nyear.ne.0) then

      tplant_conc(1)=(1/3)*plant_conc(plant,prev_harvest(1,plant),3)
      tplant_conc(2)=(1/3)*plant_conc(plant,prev_harvest(2,plant),3)
      tplant_conc(3)=(1/3)*plant_conc(plant,prev_harvest(3,plant),3)
      days_of_decay(1)=day-prev_harvest(1,plant)
      days_of_decay(2)=day-prev_harvest(2,plant)
      days_of_decay(3)=day-prev_harvest(3,plant)

    endif
    elseif (day.ge.harvest_dates(1,plant).and.
$    day.lt.harvest_dates(2,plant)) then
      tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3)
      days_of_decay(1)=day-harvest_dates(2,plant)
    elseif (day.ge.harvest_dates(2,plant).and.
$    day.lt.harvest_dates(3,plant)) then
      tplant_conc(1)=(.5)*plant_conc(plant,harvest_dates(1,plant),3)
      tplant_conc(2)=(.5)*plant_conc(plant,harvest_dates(2,plant),3)
      days_of_decay(1)=day-harvest_dates(1,plant)
      days_of_decay(2)=day-harvest_dates(2,plant)
    elseif (day.ge.harvest_dates(3,plant)) then
      tplant_conc(1)=(1/3)*plant_conc(plant,harvest_dates(1,plant),3)
      tplant_conc(2)=(1/3)*plant_conc(plant,harvest_dates(2,plant),3)
      tplant_conc(3)=(1/3)*plant_conc(plant,harvest_dates(3,plant),3)
      days_of_decay(1)=day-harvest_dates(1,plant)
      days_of_decay(2)=day-harvest_dates(2,plant)
      days_of_decay(3)=day-harvest_dates(3,plant)
    endif
  elseif (plant.eq.6) then
    tplant_conc(1)=plant_conc(plant,day,3)
  elseif (plant.eq.7.or.plant.eq.9) then
    if (day.lt.harvest_dates(1,plant)) then
      if (nyear.eq.0) then
        tplant_conc(1)=0
      else
        tplant_conc(1)= harv_conc(plant,1)
        days_of_decay(1)=(days_in_year(nyear)-
$        prev_harvest(1,plant)) +
$        day
      endif
    else
      tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3)
      days_of_decay(1)=day-harvest_dates(1,plant)
    endif
  endif
endif

```

```

do m=1,3
  if (tplant_conc(m).ne.0) then
    ap_conc(h,day)=ap_conc(h,day)+
$       animal_feed_consumption(h,plant,day)
$       /total_consum(h)*
$       animal_feed_consumption(h,plant,day) *
$       tplant_conc(m)* exp(-ldecay(nucnumb)*
$       days_of_decay(m))
  endif
enddo
10  enddo ! planttype

  tap_conc(h,day)=(tap_conc(h,day)+
$       animal_soil_ingestion(h,day))*
$       tf_animal(h,nucnumb)

c    tap_decayed(h,day)=tap_conc(h,day-t_animal(h))* exp((-1)
c    $                    * ldecay(nucnumb)
c    $                    * t_animal(h))

  enddo ! animal product k

```

Return
END

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: milk_conc
C
C DESCRIPTION: Calculate Milk concentrations(types 1-5) of I-131
C
C LANGUAGE: F77
C
C PARAMETERS:
C   nuc_num - current nuclide (assumed I-131)
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C   day - current day
CC
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   Subroutine milk_conc(nuc_num,year,loc,day)
C   implicit none
C
C-----BEGIN DECLARATION BLOCK-----
C
C   include 'envir.cmn'
C   include 'user_input.cmn'
C   include 'calc_conc.cmn'
C
C   integer days_of_decay(3) , nuc_num
C   integer i,j,h,day,loc,tyear,year
C   real total_consum(4)
C   real milk_conc_matrix(4)
C   integer plant,nyear,m
C   real tplant_conc(3)
C   real fraction(3)
C
C-----BEGIN EXECUTABLE CODE-----
C
C   tyear=year-1
C
C   do i=1,tnum_dr
C     total_consum(i)=0
C   enddo
C
C   do i=1,tnum_dr
C     do j=1,tpplantmax
C       total_consum(i)=total_consum(i)+
C $         dairy_feed_consumption(i,j,day)
C     enddo
C   enddo
C
C   do i=1,tnum_dr
C     milk_conc_matrix(i)=0
C   enddo
C
C   do h=1,tnum_dr
C     do plant=1,plantmax
C       do m=1,3
C         tplant_conc(m)=0

```



```

      days_of_decay(m)=0
    enddo
  if (plant.eq.1.or.plant.eq.2.or.plant.eq.4..or.plant.eq.8) then
    goto 10
  elseif (plant.eq.3) then
    if (day.lt.harvest_dates(1,plant)) then
      if (nyear.eq.0) then
        tplant_conc(1)=0
      else
        tplant_conc(1)= harv_conc(plant,1)
        days_of_decay(1)=(days_in_year(nyear)-
          $      prev_harvest(1,plant)) +
          $      day
      endif
    else
      $      tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3) +
      $      plant_conc(plant,harvest_dates(1,plant),4)
      days_of_decay(1)=day-harvest_dates(1,plant)
    endif
  elseif (plant.eq.5) then
    if (day.lt.harvest_dates(1,plant)) then
      if (nyear.ne.0) then

        tplant_conc(1)=(1/3)*plant_conc(plant,prev_harvest(1,plant),3)
        tplant_conc(2)=(1/3)*plant_conc(plant,prev_harvest(2,plant),3)
        tplant_conc(3)=(1/3)*plant_conc(plant,prev_harvest(3,plant),3)
        days_of_decay(1)=day-prev_harvest(1,plant)
        days_of_decay(2)=day-prev_harvest(2,plant)
        days_of_decay(3)=day-prev_harvest(3,plant)
        fraction(1)=1/3
        fraction(2)=1/3
        fraction(3)=1/3
      endif
    elseif (day.ge.harvest_dates(1,plant).and.
      $      day.lt.harvest_dates(2,plant)) then
      tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3)
      days_of_decay(1)=day-harvest_dates(2,plant)
      fraction(1)=1
    elseif (day.ge.harvest_dates(2,plant).and.
      $      day.lt.harvest_dates(3,plant)) then
      tplant_conc(1)=(.5)*plant_conc(plant,harvest_dates(1,plant),3)
      tplant_conc(2)=(.5)*plant_conc(plant,harvest_dates(2,plant),3)
      days_of_decay(1)=day-harvest_dates(1,plant)
      days_of_decay(2)=day-harvest_dates(2,plant)
      fraction(1)=1/2
      fraction(2)=1/2
    elseif (day.ge.harvest_dates(3,plant)) then
      tplant_conc(1)=(1/3)*plant_conc(plant,harvest_dates(1,plant),3)
      tplant_conc(2)=(1/3)*plant_conc(plant,harvest_dates(2,plant),3)
      tplant_conc(3)=(1/3)*plant_conc(plant,harvest_dates(3,plant),3)
      days_of_decay(1)=day-harvest_dates(1,plant)
      days_of_decay(2)=day-harvest_dates(2,plant)
      days_of_decay(3)=day-harvest_dates(3,plant)
      fraction(1)=1/3
      fraction(2)=1/3
      fraction(3)=1/3
    endif
  elseif (plant.eq.6) then
    tplant_conc(1)=plant_conc(plant,day,3)
  elseif (plant.eq.7.or.plant.eq.9) then
    if (day.lt.harvest_dates(1,plant)) then
      if (nyear.eq.0) then
        tplant_conc(1)=0
      else
        tplant_conc(1)= harv_conc(plant,1)
        days_of_decay(1)=(days_in_year(nyear)-

```

```

      prev_harvest(1,plant)) +
$      day
    endif
  else
    tplant_conc(1)=plant_conc(plant,harvest_dates(1,plant),3)
    days_of_decay(1)=day-harvest_dates(1,plant)
  endif
endif

c  milk_conc_matrix(h)=sum of dairy feed consumption conc for this regime
  if (plant.eq.5) then
    do m=1,3
      if (tplant_conc(m).ne.0) then
        milk_conc_matrix(h)=milk_conc_matrix(h) +
$          fraction(m)*dairy_feed_consumption(h,plant,day)
$          /total_consum(h)*
$          fraction(m)*dairy_feed_consumption(h,plant,day)
$          *tplant_conc(m) * exp(-ldecay(nuc_num)
$          *days_of_decay(m))
      endif
    enddo
  else
    milk_conc_matrix(h)=milk_conc_matrix(h) +
$    dairy_feed_consumption(h,plant,day)
$    /total_consum(h)*
$    dairy_feed_consumption(h,plant,day) *
$    tplant_conc(1) * exp(-ldecay(nuc_num)
$    *days_of_decay(1))
  endif
10  enddo !plant

  tmilk_conc_matrix(h,day)=(milk_conc_matrix(h)+
$    dairy_soil_ingestion(h,day))*
$    tfmilk(nuc_num)

  Do I=1,tnum_dr
    mixed_milk_conc(day,loc)=mixed_milk_conc(day,loc)
$    + regime_fractions(I,loc)*
$    milk_conc_matrix(I)
  Enddo

c  ap_decayed(h,day)=tmilk_conc_matrix(h,day)* exp((-1)* ldecay(nuc_num)
c  $    *t_      (h,num_day))

  enddo
  return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: WRITE_CONC_TO_FILE
C
C DESCRIPTION: Writes the concentration for each variable to environmental
C               output file.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C   eun - logical unit number of file I/O
C
C FILE I/O: Output file <nuc>_cc.yy (creamery xx, year 19yy)
C
C FUNCTIONS/SUBROUTINES CALLED: none

```

```

C-----BEGIN IMPLEMENTATION BLOCK-----

```

```

C AUTHOR: Nila Reitz
C         Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

SUBROUTINE WRITE_CONC_TO_FILE(YEAR,LOC,EUN)
IMPLICIT NONE

```

C2.26

```

C-----BEGIN DECLARATION BLOCK-----

```

```

INCLUDE 'ENVIR.CMN'
INCLUDE 'USER_INPUT.CMN'
INCLUDE 'CALC_CONC.CMN'

```

```

INTEGER YEAR,TYEAR,NVAR
CHARACTER*40 FILENAME
CHARACTER* 4 CLOC,CYEAR
REAL*4 OLD_CONC,NEW_CONC
INTEGER BEGIN DAY(15), CODE(15)
INTEGER LOC,DAY,I,NREC,NREP
C   INTEGER ISTAT,LIB$GET_LUN,LIB$FREE_LUN,LUN
INTEGER MILKTYPES,PERIOD_COUNT
INTEGER K,L,EUN,REC_COUNT,VALUE_COUNT
INTEGER HEADER_COUNT,MT_PERIOD,MILK_TYPES,M,N
INTEGER FVCOUNT(22),LVCOUNT(22),VEG_CONC
REAL*4 TCONC(32)
INTEGER FIRST_TIME
REAL*4 MINCONC
DATA MT_PERIOD/15/
DATA VEG_CONC/13/

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

MINCONC=1.0e-30
HEADER_COUNT=1
MILKTYPES=5 ! REGIMES,RURAL (CREAMERY CONC SAVED FOR DOSE)
NVAR=VEG_CONC + APMAX + MILKTYPES
NREP=RUNMAX

```

```

C WRITE FIRST HEADER RECORD
C WRITE(UNIT=EUN,REC=HEADER_COUNT) NVAR,NREP,NREC
C WRITE(*,100) NVAR,NREP,NREC
C   HEADER_COUNT=HEADER_COUNT+1
100 FORMAT(3I)

```

```

C   HEADER RECORDS
C   ACCESS CODE N IS:
C   -2 ==> DATA VALUES ARE MISSING (UNKNOWN, NOT COMPUTED)
C   -1 ==> DATA VALUES ARE ZERO OR IRRELEVANT (N/A)
C   0 ==> USE DATA STORED IN THIS FILE FOR THIS TIME PERIOD
C   >0 ==> USE THE DATA FROM LOCATION N FOR THIS TIME PERIOD

```

```
REC_COUNT=NVAR+1
```

```
DO I=1, (NVAR-1)
```

```
DO M=1, 32
```

```
TCONC(M)=0
```

```
ENDDO
```

```
OLD_CONC=0
```

```
NEW_CONC=0
```

```
PERIOD_COUNT=0
```

```
FVCOUNT(I)=0
```

```
DO N=1, MT PERIOD
```

```
BEGIN_DAY(N)=0
```

```
CODE(N)=-1
```

```
ENDDO
```

```
FIRST_TIME=0
```

```
VALUE_COUNT=0
```

```
DO DAY=1, DAYS IN YEAR (YEAR)
```

```
IF (I.EQ.1) THEN
```

```
NEW_CONC=PLANT_CONC(1,DAY,4)/BIOMASS(1,DAY) !LEAFY VEG INNER
```

```
ELSEIF (I.EQ.2) THEN
```

```
NEW_CONC=PLANT_CONC(1,DAY,3)/BIOMASS(1,DAY) !LEAFY VEG OUTER
```

```
ELSEIF (I.EQ.3) THEN
```

```
NEW_CONC=PLANT_CONC(2,DAY,4)/BIOMASS(2,DAY) !OTH VEG INNER
```

```
ELSEIF (I.EQ.4) THEN
```

```
NEW_CONC=PLANT_CONC(3,DAY,4)/BIOMASS(3,DAY) !GRAIN INNER
```

```
ELSEIF (I.EQ.5) THEN
```

```
NEW_CONC=PLANT_CONC(4,DAY,4)/BIOMASS(4,DAY) !FRUITS INNER
```

```
ELSEIF (I.EQ.6) THEN
```

```
NEW_CONC=PLANT_CONC(4,DAY,3)/BIOMASS(4,DAY) !FRUITS OUTER
```

```
ELSEIF (I.EQ.7) THEN
```

```
NEW_CONC=(PLANT_CONC(5,DAY,3)+PLANT_CONC(5,DAY,4))
```

```
$ /BIOMASS(5,DAY)
```

```
ELSEIF (I.EQ.8) THEN
```

```
NEW_CONC=(PLANT_CONC(6,DAY,3)+PLANT_CONC(6,DAY,4))
```

```
$ /BIOMASS(6,DAY)
```

```
ELSEIF (I.EQ.9) THEN
```

```
NEW_CONC=PLANT_CONC(6,DAY,1)
```

```
ELSEIF (I.EQ.10) THEN
```

```
NEW_CONC=PLANT_CONC(6,DAY,2)
```

```
ELSEIF (I.EQ.11) THEN
```

```
NEW_CONC=(PLANT_CONC(7,DAY,3)+PLANT_CONC(7,DAY,4))
```

```
$ /BIOMASS(7,DAY)
```

```
ELSEIF (I.EQ.12) THEN
```

```
NEW_CONC=(PLANT_CONC(8,DAY,3)+PLANT_CONC(8,DAY,4))
```

```
$ /BIOMASS(8,DAY)
```

```
ELSEIF (I.EQ.13) THEN
```

```
NEW_CONC=(PLANT_CONC(9,DAY,3)+PLANT_CONC(9,DAY,4))
```

```
$ /BIOMASS(9,DAY)
```

```
ELSEIF ((I.GT.13).AND.(I.LE.17)) THEN
```

```
NEW_CONC=TAP_CONC((I-13),DAY)
```

```
ELSEIF ((I.GT.17).AND.(I.LE.21)) THEN
```

```
NEW_CONC=MILK_CONC_MATRIX((I-17),DAY)
```

```
ELSEIF (I.EQ.22) THEN
```

```
NEW_CONC=MIXED_MILK_CONC(DAY,LOC)
```

```
c????????????????????
```

```
ENDIF
```

```
IF (DAY.EQ.1) AND (NEW_CONC.GT.MINCONC) THEN
```

C2.27

```

        PERIOD COUNT=PERIOD COUNT+1
        BEGIN DAY(PERIOD COUNT)=DAY
        CODE(PERIOD COUNT)--1
    ELSEIF ((OLD CONC.LT.MINCONC).AND.(NEW CONC.GT.MINCONC)) THEN
        PERIOD COUNT=PERIOD COUNT+1
        BEGIN DAY(PERIOD COUNT)=DAY
        CODE(PERIOD COUNT)=0
    ELSEIF ((OLD CONC.GT.MINCONC).AND.(NEW CONC.LT.MINCONC)) THEN
        PERIOD COUNT=PERIOD COUNT+1
        BEGIN DAY(PERIOD COUNT)=DAY
        CODE(PERIOD COUNT)--1
    ENDIF

    IF (NEW CONC.GT.MINCONC) THEN
        FIRST TIME=FIRST TIME+1
        FVCOUNT(I)=REC COUNT+1
        VALUE COUNT=VALUE COUNT+1
        TCONC(VALUE COUNT)=NEW CONC
        IF (VALUE COUNT.EQ.32) THEN
            REC COUNT=REC COUNT+1
            WRITE(UNIT=EUN,REC=REC COUNT)
            * (TCONC(M),M=1,32)
C      80      WRITE(*,80) REC COUNT,TCONC(1)
101      FORMAT(1X,I2,1PE15.5)
            FORMAT(1PE15.5)

            DO M=1,32
                TCONC(M)=0
            ENDDO
            VALUE COUNT=0
        ENDIF
    ENDIF
C 2.28 C WRITE CONC TO FILE
        OLD CONC=NEW CONC
        ENDDO
        IF (VALUE COUNT.NE.0) THEN
            REC COUNT=REC COUNT+1
            WRITE(UNIT=EUN,REC=REC COUNT)
            * (TCONC(M),M=1,32)
        ENDIF

C WRITE ACCESS HEADER RECORD
        HEADER COUNT=I+1
        IF (FVCOUNT(I).EQ.0) THEN
            LVCOUNT(I)=0
        ELSE
            LVCOUNT(I)=REC COUNT
        ENDIF
        WRITE(UNIT=EUN,REC=HEADER COUNT)
        $ FVCOUNT(I),LVCOUNT(I),(BEGIN DAY(K),CODE(K),K=1,MT PERIOD)

        ENDDO
102      FORMAT(I2,2(16I2))
C REWRITE RECORD 1 TO UPDATE NUMBER OF RECORDS IN FILE
        NREC=REC COUNT
        WRITE(UNIT=EUN,REC=1) NVAR,NREP,NREC

        RETURN
        END

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Write_creamery
C
C DESCRIPTION: Writes concentrations at creameries to output creamery file.
C
C LANGUAGE: F77
C   run - current run
C   year - current year (ie. year=1 implies 1945)
C   nuclide - current nuclide (assumed I-131)
C   creamery - current creamery
C
C FILE I/O: Reads binary file Params.XXX (XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

SUBROUTINE WRITE_CREAMERY(RUN, YEAR, NUCLIDE, CREAMERY)
IMPLICIT NONE

```

```

C-----BEGIN DECLARATION BLOCK-----
C
C INCLUDE 'ENVIR.CMN'
C INCLUDE 'CALC_CONC.CMN'
C INCLUDE 'USER_INPUT.CMN'
C
C INTEGER RUN, YEAR, NUCLIDE, CREAMERY, TYEAR
C CHARACTER*2 CCREAM
C CHARACTER*4 CYEAR
C INTEGER REC_COUNT, LUN, LIB$GET_LUN, LIB$FREE_LUN, ISTAT
C INTEGER DAY
C CHARACTER*40 FILENAME

```

```

C-----BEGIN EXECUTABLE CODE-----
C
C   REC_COUNT=1
C   TYEAR=YEAR+1944
C   WRITE(CYEAR,100) TYEAR
100  FORMAT(I4)
C   DO CREAMERY=1, NUM_CREAMERIES
C     WRITE(CCREAM,101) CREAMERY
101  FORMAT(I2.2)
C
C     FILENAME=NUCLIST(NUCLIDE) //' ' //CCREAM//
C     *      ' ' //CYEAR(3:4)
C     ISTAT=LIB$GET_LUN(LUN)
C     IF (RUN.EQ.1) THEN
C       OPEN(UNIT=LUN, FILE=filename,
C $         access='direct', form='formatted',
C $         recl=128, STATUS='new')
C       OPEN(UNIT=LUN, FILE=FILENAME, STATUS='NEW',
C *       FORM='UNFORMATTED')
C     ELSE
C       OPEN(UNIT=LUN, FILE=filename,
C $         access='direct', form='formatted',

```

```

C      OPEN (UNIT=LUN, FILE=FILENAME, STATUS='OLD',
C      *      FORM='UNFORMATTED')
      ENDIF
      DO DAY=1, DAYS_IN_YEAR(YEAR)
      WRITE (UNIT=LUN, REC=REC_COUNT, FMT=102)
102 $      CREAMERY_CONC (CREAMERY, DAY)
      FORMAT (F10.5)
      REC_COUNT=REC_COUNT+1
      ENDDO
      CLOSE (UNIT=LUN)
      ISTAT=LIB$FREE_LUN (LUN)

C      ENDDO
      RETURN
      END

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: WRITE_MILK
C
C DESCRIPTION: Writes milk concentrations at end of environmental
C               concentrations output file.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   eun - logical unit number of file I/O
C   year - current year (ie. year=1 implies 1945)
C   loc - current location
C
C FILE I/O: Output file <nuc>_xxxx.yy (location xxxx, year 19yy)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reltz
C         Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE WRITE_MILK(EUN, YEAR, LOC)
C IMPLICIT NONE
C
C-----BEGIN DECLARATION BLOCK-----
C
C INCLUDE 'ENVIR.CMN'
C INCLUDE 'USER_INPUT.CMN'
C INCLUDE 'CALC_CONC.CMN'
C
C INTEGER YEAR, TYEAR, NVAR, EUN
C CHARACTER*11 FILENAME
C CHARACTER* 4 CLOC, CYEAR
C REAL*4 OLD CONC, NEW CONC
C INTEGER BEGIN_DAY(16), CODE(16)
C INTEGER LOC, DAY, I, NREC, NREP
C INTEGER ISTAT, LIB$GET_LUN, LIB$FREE_LUN
C INTEGER PERIOD_COUNT, HEADER_COUNT, VALUE_COUNT
C INTEGER L, REC_COUNT, CUR_COUNT, FVCOUNT(1), LVCOUNT(1), M, K
C REAL*4 TCONC(32)
C REAL*4 MINCONC
C INTEGER MT_PERIOD, FIRST_TIME
C
C DATA MT_PERIOD/15/
C
C-----BEGIN EXECUTABLE CODE-----
C
C HEADER RECORDS
C ACCESS CODE N IS:
C   -2 ==> DATA VALUES ARE MISSING (UNKNOWN, NOT COMPUTED)
C   -1 ==> DATA VALUES ARE ZERO OR IRRELEVANT (N/A)
C   0 ==> USE DATA STORED IN THIS FILE FOR THIS TIME PERIOD
C   >0 ==> USE THE DATA FROM LOCATION N FOR THIS TIME PERIOD
C
C MINCONC=1.0e-30
C I=1
C
C READ(UNIT=EUN, REC=1) NVAR, NREP, NREC
C format(3i)
C REC_COUNT=0
C CUR_COUNT=NREC+1

```



```

FVCOUNT(I)=NREC+1
PERIOD COUNT=0
VALUE COUNT=0
FIRST TIME=0

```

```

DO DAY=1,DAYS IN YEAR(YEAR)
  NEW CONC=RURAL CONC(LOC, DAY)
  IF ((DAY.EQ.I).AND.(NEW CONC.GT.MINCONC)) THEN
    PERIOD COUNT=PERIOD COUNT+1
    BEGIN DAY(PERIOD COUNT)=DAY
    CODE(PERIOD COUNT)--1
  ELSEIF ((OLD CONC.LT.MINCONC).AND.(NEW CONC.GT.MINCONC)) THEN
    PERIOD COUNT=PERIOD COUNT+1
    BEGIN DAY(PERIOD COUNT)=DAY
    CODE(PERIOD COUNT)=0
  ELSEIF ((OLD CONC.GT.MINCONC).AND.(NEW CONC.LT.MINCONC)) THEN
    PERIOD COUNT=PERIOD COUNT+1
    BEGIN DAY(PERIOD COUNT)=DAY
    CODE(PERIOD COUNT)--1
  ENDIF

```

```

  IF (NEW CONC.GT.MINCONC) THEN
    FIRST TIME=FIRST TIME+1
    FVCOUNT(I)=REC COUNT+1
    VALUE COUNT=VALUE COUNT+1
    TCONC(VALUE COUNT)=NEW CONC
    IF (VALUE COUNT.EQ.32) THEN
      REC COUNT=REC COUNT+1
      WRITE(UNIT=EUN, REC=REC COUNT)
      (TCONC(M), M=1, 32)
      WRITE(*, 80) REC COUNT, TCONC(1)
      FORMAT(1X, I2, 1PE15.5)
      FORMAT(1PE15.5)
      VALUE COUNT=0
      CUR COUNT=CUR COUNT+1
    ENDIF
  ENDIF
  OLD CONC=NEW CONC
ENDDO
IF (VALUE COUNT.NE.0) THEN
  REC COUNT=REC COUNT+1
  WRITE(UNIT=EUN, REC=CUR COUNT)
  (TCONC(M), M=1, 32)
ENDIF

```

```

C WRITE ACCESS HEADER RECORD
HEADER COUNT=NVAR+1 1 LAST VARIABLE ACCESS RECORD
IF (FVCOUNT(I).EQ.0) THEN
  LVCOUNT(I)=0
ELSE
  LVCOUNT(I)=REC COUNT
ENDIF
WRITE(UNIT=EUN, REC=HEADER COUNT)
$ FVCOUNT(I), LVCOUNT(I), (BEGIN DAY(K), CODE(K), K=1, MT PERIOD)

```

```

102 FORMAT(I2, 16I2)
C REWRITE RECORD 1 TO UPDATE NUMBER OF RECORDS IN FILE
NREC=NREC+ REC COUNT
WRITE(UNIT=EUN, REC=1) NVAR, NREC, NREC
103 FORMAT(3I2)

```

```

RETURN
END

```

C2.32

[illegible]

```

endif
CCCCCCCCCCCCCCCCCCCC
filename='[hdr.phase2.data]'//
*      nuclist(nuc_numb)//'_air'//cyear(3:4)//
*      './'//crun

C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  istat=lib$get lun(cun)      ! air model results
  open(unit=cun,file=filename,status='old',
    form='unformatted')
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

  check=check+1
  do i=1,plocation
    call read_air_model(cun,check)
    check=check+1
  enddo

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  READ MASS LOADING FOR THIS LOCATION (RECORD B3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

  filename='[hdr.phase2.params]params'
  *      //cyear(3:4)//'_.'//crun

C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  istat = lib$get lun (dun)
  *      open(unit=dun,file=filename,form='unformatted',
    status='old')
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

  call read_loc_inv(dun)
  do i=1,plocation
    call read_loc_var(dun)
  enddo

C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  close(unit=dun)
  istat = lib$free lun (dun)
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

  call read_envir_output(nuc_numb,tyear,plocation)
  if (tyear.eq.byear) then
    start_day=begin_doy
  else
    start_day=1
  endif
  if (tyear.eq.eyear) then
    end_day=end_doy
  else
    end_day=days_in_year(year)      ! ???input data start with fyear???
  endif
  do day = start_day,end_day
    if (day.eq.birth_doy) then
      age=age+1
      if (age.lt.22.and.(asl_index.ne.13
        .or.asl_index.ne.26)) then
S      call determine_asl(age,sex,lifestyle,asl_index,
$      boy,eoy,nyear,eyear)

```

```

        call interpolate_dose(age,asl_index,nuc_numb)
      endif
    elseif ((day.eq.boy).and.(year.eq.nyear)) then
      if (lifestyle.eq.'rural'.or.lifestyle.eq.'RURAL') then
        asl_index=13
      else
        asl_index=26
      endif
      call interpolate_dose(age,asl_index,nuc_numb)
    elseif ((day.eq.eoy).and.(year.eq.eyear)) then
      if (lifestyle.eq.'rural'.or.lifestyle.eq.'RURAL') then
        asl_index=12
      else
        asl_index=25
      endif
      call interpolate_dose(age,asl_index,nuc_numb)
    endif
  endif

```

```

$ dlh(day,plocation)=(air_conc(day)+(plant_conc(6,day,1)
$      * ml(day)/pr(plocation)*ftime))
$      * breathing_rate(asl_index,day)
$      * tdose_factor(1)

```

```

$ DS(day,plocation)=air_conc(day)*tdose_factor(2)*
$      sshield_factor(asl_index)
$ d_groundshine(day,plocation)=(plant_conc(6,day,1)
$      * tdose_factor(3)
$      + plant_conc(6,day,2)
$      * tdose_factor(4))*gshield_factor(asl_index)

```

```

c milk ingestion due to var 14-16 properly decayedtdose_factor(5)
  do i=9,10
    ref_consump=ref_food_consumption(i,asl_index,day)
    if (i.eq.10) then
      holdup=tatored_milk_gro(day)
    else
      if (ms.eq.1) then
        holdup=tfresh_milk_fc(day)
        conc=tmilk_conc_matrix(fr,day)
      elseif (ms.eq.2) then
        holdup=tfresh_milk_gro(day)
        conc=mixed_milk_conc(day,plocation)
      else
        holdup=tfresh_milk_cr(day)
        conc=creamery_conc(sc,day)
      endif
    endif
    if (i.eq.9.or.1.eq.10) then
      mtemp_decayed(i,day)=conc* exp((-1)
$      * ldecay(nuc_numb)
$      * holdup)
      dconc=mtemp_decayed(i,day)*ref_consump
      decayed_conc(day,plocation)=decayed_conc(day,plocation)
$      +dconc
    endif
  enddo
  migestion(day,plocation)=tdose_factor(5)*
$      decayed_conc(day,plocation)

```

```

c ingestion due to var 1-4,10-13 properly decayed tdose_factor(6)
  do i=1,num_foods
    if (i.eq.1) then
      ref_consump=ref_food_consumption(1,asl_index,day)
      conc=tap_conc(1,day)
    elseif (i.eq.2) then

```

```

      conc=tap_conc(3,day)
    elseif (i.eq.3) then
      ref_consump=ref_food_consumption(3,asl_index,day)
      conc=((1-fwash(1,asl_index,day))*plant_conc(1,day,3)
        +plant_conc(1,day,4))
$
    elseif (i.eq.4) then
      ref_consump=ref_food_consumption(4,asl_index,day)
      conc=plant_conc(2,day,4)
    elseif (i.eq.5) then
      ref_consump=ref_food_consumption(i,asl_index,day)
      conc=tap_conc(4,day)
    elseif (i.eq.6) then
      ref_consump=ref_food_consumption(i,asl_index,day)
      conc=plant_conc(3,day,4)
    elseif (i.eq.7) then
      ref_consump=ref_food_consumption(i,asl_index,day)
      conc=((1-fwash(4,asl_index,day))*plant_conc(4,day,3)+
$      plant_conc(4,day,4))
    endif
    if (i.eq.1) then
      holdup=t_animal(1,day)
    elseif (i.eq.2) then
      holdup=t_animal(3,day)
    elseif (i.eq.5) then
      holdup=t_animal(4,day)
    else
      holdup=0
    endif
    if (i.le.7) then
      otemp_decayed(I,day)=conc* exp((-1)
$      * ldecay(nuc_numb)
$      * holdup)
      dconc=otemp_decayed(i,day)*ref_consump
      decayed_conc(day,plocation)=decayed_conc(day,plocation)+dconc
    endif
  enddo
  olngestion(day,plocation)=tdose_factor(6)*
$  decayed_conc(day,plocation)
enddo ! days in year
c  do i=1,tlocmax
c  if (loclist(i)) then
c  call write_dose_to_file(year,plocation,eun)
c  endif
c  enddo
enddo !years
enddo !number of runs

C #####
C #####
  close(unit=eun)
  istat=lib$free lun(eun)
C #####
C #####
c900 enddo ! nuclist
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Valid_date
C
C DESCRIPTION: Determines if a date is valid.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   date - user entered date
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      function valid_date(date)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

C3.9

```

      integer day,month,year
      integer dim(12,2)
      integer quo,rem
      logical century,cdiv,ydiv
      character*10 date
      logical vyear,vmonth,vday
      logical leap_year,valid_date

      data dim/31,28,31,30,31,30,31,31,30,31,30,31,
-          31,29,31,30,31,30,31,31,30,31,30,31/

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

      read(date(1:2),100) month
      read(date(4:5),100) day
      read(date(7:10),101) year
100 format(i2)
101 format(i4)

```

```

      leap_year = .false.
      vday = .false.
      vmonth = .false.
      vyear = .false.
      quo=year/100
      rem=year-(quo*100)
      if (rem.eq.0) then
        quo=year/400
        rem=year-(quo*400)
        if (rem.eq.0) then
          leap_year = .true.
        endif
      else ! not a century
        quo=year/4
        rem=year-(quo*4)
        if (rem.eq.0) then

```

```

        leap_year = .true.
    endif
endif

if (year .ge. 1800 .and. year .le. 1991) then
    vyear=.true.
    if ((month.gt.0).and.(month.lt.13)) then
        vmonth=.true.
        if (leap_year) then
            if (day.le.dim(month,2) .and. day.gt.0) vday=.true.
        else
            if (day.le.dim(month,1) .and. day.gt.0) vday=.true.
        endif
    endif
endif
if (vyear.and.vmonth.and.vday) then
    valid_date=.true.
else
    valid_date=.false.
endif
return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Determine_doy
C
C DESCRIPTION: Determine the day of year of a date.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   date - user entered date
C   doy - day of year of date
C   year - year of current date
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

subroutine determine_doy(date,doy,year)
implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

```

character*10 date
integer doy, YEAR

integer day, month
integer dim(12,2)
integer quo, rem, i
logical leap_year

data dim/31,28,31,30,31,30,31,31,30,31,30,31,
-      31,29,31,30,31,30,31,31,30,31,30,31/

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

100 read(date(1:2),100) month
101 read(date(4:5),100) day
102 read(date(7:10),101) year
103 format(i2)
104 format(i4)

```

```

leap_year = .false.
quo=year/100
rem=year-(quo*100)
if (rem.eq.0) then
  quo=year/400
  rem=year-(quo*400)
  if (rem.eq.0) then
    leap_year = .true.
  endif
else ! not a century
  quo=year/4
  rem=year-(quo*4)
  if (rem.eq.0) then
    leap_year = .true.
  endif
endif

```



```

endif

doy = day
if (leap_year) then
  if (month .gt. 1) then
    do i=1,month-1
      doy=doy+dim(i,2)
    enddo
  endif
else ! Not a leap year; check for 02/29 (birthday?)
  if (month .eq. 2 .and. day .eq. 29) then
    doy = 60 ! Feb. 29 birthday on non-leap year: use March 1
  else if (month .gt. 1) then
    do i=1,month-1
      doy=doy+dim(i,1)
    enddo
  endif
endif
endif

```

C-- If year is 1945, then add 6 days (last 6 days of 1944) to the day-of-year.

```

  if (year .eq. 1945) then
    doy = doy + 6
  
```

C-- If date is between 12/26/44 and 12/31/44, consider it to be part of 1945.

```

  else if (year.eq.1944 .and. month.eq.12 .and. day .ge. 26) then
    year = 1945
    doy = day - 25
  endif

```

```

return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Determine_asl
C
C DESCRIPTION: Determines user's asl_index based on age,sex,lifestyle, and
C               whether user was a nursing mother.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   age - the user's age
C   sex - the user's sex
C   lifestyle - rural or urban
C   asl_index - user's age index
C   nursing - signifies whether user was nursing a child
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C         Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

C3.13

```

subroutine determine_asl(age,sex,lifestyle,asl_index,nursing)
implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----
C
C include 'envir.cmn'
C
C integer asl_index
C character*1 sex, lifestyle*5
C logical rural, male
C logical nursing
C real*4 age

```

```

C-----BEGIN EXECUTABLE CODE-----
C
C asl_index = 0
C if (sex(1:1).eq.'m'.or.sex(1:1).eq.'M') then
C   male = .true.
C else
C   male = .false.
C endif
C if (lifestyle.eq.'rural'.or.lifestyle.eq.'RURAL') THEN
C   rural = .true.
C else
C   rural = .false.
C endif
C
C if (nursing) then
C   if (rural) then
C     asl_index = 26
C   else
C     asl_index = 13
C   endif
C elseif (age.lt.1 .and. .not.rural .and. male) then

```

```

elseif (age.ge.1 .and. age.lt.2 .and. .not.rural
$      .and. male) then
    asl_index=2
elseif (age.ge.2 .and. age.lt.7 .and. .not.rural
$      .and. male) then
    asl_index=3
elseif (age.ge.7 .and. age.lt.12 .and. .not.rural
$      .and. male) then
    asl_index=4
elseif (age.ge.12 .and. age.lt.17 .and. .not.rural
$      .and. male) then
    asl_index=5
elseif (age.ge.17 .and. .not.rural .and. male) then
    asl_index=6
elseif (age.lt.1 .and. .not.rural .and. .not.male) then
    asl_index=7
elseif (age.ge.1 .and. age.lt.2 .and. .not.rural
$      .and. .not.male) then
    asl_index=8
elseif (age.ge.2 .and. age.lt.7 .and. .not.rural
$      .and. .not.male) then
    asl_index=9
elseif (age.ge.7 .and. age.lt.12 .and. .not.rural
$      .and. .not.male) then
    asl_index=10
elseif (age.ge.12 .and. age.lt.17 .and. .not.rural
$      .and. .not.male) then
    asl_index=11
elseif (age.ge.17 .and. .not.rural .and. .not.male) then
    asl_index=12

elseif (age.lt.1 .and. rural .and. male) then
    asl_index=14
elseif (age.ge.1 .and. age.lt.2 .and. rural
$      .and. male) then
    asl_index=15
elseif (age.ge.2 .and. age.lt.7 .and. rural
$      .and. male) then
    asl_index=16
elseif (age.ge.7 .and. age.lt.12 .and. rural
$      .and. male) then
    asl_index=17
elseif (age.ge.12 .and. age.lt.17 .and. rural
$      .and. male) then
    asl_index=18
elseif (age.ge.17 .and. rural .and. male) then
    asl_index=19
elseif (age.lt.1 .and. rural .and. .not.male) then
    asl_index=20
elseif (age.ge.1 .and. age.lt.2 .and. rural
$      .and. .not.male) then
    asl_index=21
elseif (age.ge.2 .and. age.lt.7 .and. rural
$      .and. .not.male) then
    asl_index=22
elseif (age.ge.7 .and. age.lt.12 .and. rural
$      .and. .not.male) then
    asl_index=23
elseif (age.ge.12 .and. age.lt.17 .and. rural
$      .and. .not.male) then
    asl_index=24
elseif (age.ge.17 .and. rural .and. .not.male) then
    asl_index=25
endif
return

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Interpolate_dose
C
C DESCRIPTION: Calculates the dose factors for the user.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   age - user's age
C   asl_index - age group index of user
C   nuc_num - current nuclide (assumed I-131)
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      subroutine interpolate_dose(age,asl_index,nuc_num)
      implicit none

```

```

C3.15 C-----BEGIN DECLARATION BLOCK-----
      include 'envir.cmn'
      include 'dose.cmn'

      integer asl_index,nuc_num
      integer i
      real*4 age,y1(6),y2(6),x1,x2,m(6)

```

```

C-----BEGIN EXECUTABLE CODE-----
      if (age.lt.1.or.age.ge.18.or.asl_index.eq.13.or.
$      asl_index.eq.26.or.age.eq.masl(asl_index)) then
        tdose_factor(1)=inhalation_dose_factor(asl_index,nuc_num)
        tdose_factor(2)=submersion_dose_factor(asl_index,nuc_num)
        tdose_factor(3)=groundshine_dose_factor(asl_index,nuc_num)
        tdose_factor(4)=rootzone_dose_factor(asl_index,nuc_num)
        tdose_factor(5)=ingestion_dose_factor(asl_index,nuc_num)
        tdose_factor(6)=ingestion_dose_factor(asl_index,nuc_num)
      elseif (age.gt.masl(asl_index)) then
        x1=masl(asl_index)
        x2=masl(asl_index+1)
        y1(1)=inhalation_dose_factor(asl_index,nuc_num)
        y1(2)=submersion_dose_factor(asl_index,nuc_num)
        y1(3)=groundshine_dose_factor(asl_index,nuc_num)
        y1(4)=rootzone_dose_factor(asl_index,nuc_num)
        y1(5)=ingestion_dose_factor(asl_index,nuc_num)
        y1(6)=ingestion_dose_factor(asl_index,nuc_num)
        y2(1)=inhalation_dose_factor((asl_index+1),nuc_num)
        y2(2)=submersion_dose_factor((asl_index+1),nuc_num)
        y2(3)=groundshine_dose_factor((asl_index+1),nuc_num)
        y2(4)=rootzone_dose_factor((asl_index+1),nuc_num)
        y2(5)=ingestion_dose_factor((asl_index+1),nuc_num)
        y2(6)=ingestion_dose_factor((asl_index+1),nuc_num)

```

```

        m(i)=(y2(i)-y1(i))/(x2-x1)
        tdose_factor(i)=m(i)* (age-x1)+y1(i)
    enddo
elseif (age.lt.masl(asl_index)) then
    x1=masl(asl_index)
    x2=masl(asl_index-1)
    y1(1)=inhalation_dose_factor(asl_index,nuc_numb)
    y1(2)=submersion_dose_factor(asl_index,nuc_numb)
    y1(3)=groundshine_dose_factor(asl_index,nuc_numb)
    y1(4)=rootzone_dose_factor(asl_index,nuc_numb)
    y1(5)=ingestion_dose_factor(asl_index,nuc_numb)
    y1(6)=ingestion_dose_factor(asl_index,nuc_numb)
    y2(1)=inhalation_dose_factor((asl_index-1),nuc_numb)
    y2(2)=submersion_dose_factor((asl_index-1),nuc_numb)
    y2(3)=groundshine_dose_factor((asl_index-1),nuc_numb)
    y2(4)=rootzone_dose_factor((asl_index-1),nuc_numb)
    y2(5)=ingestion_dose_factor((asl_index-1),nuc_numb)
    y2(6)=ingestion_dose_factor((asl_index-1),nuc_numb)
    do i=1,6
        m(i)=(y2(i)-y1(i))/(x2-x1)
        tdose_factor(i)=m(i)* (age-x1)+y1(i)
    enddo

    endif !interpolate the dose
return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_air_model
C
C DESCRIPTION: Reads input parameters from the air model.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   aun - logical unit number of file I/O
C
C FILE I/O: Reads binary file I-131_airYY.XXX (YY is year, XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nils Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Subroutine read_air_model(aun,check)
      implicit none

C-----BEGIN DECLARATION BLOCK-----
      include 'envir.cmn'

      integer aun,loc
      integer nnodes,nday
      integer check

C-----BEGIN EXECUTABLE CODE-----
      if (check.eq.1) then
        read(aun) nnodes,nday
      endif
      read(aun) air_conc,surface_conc

      return
      end

```

C3.17

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_time_inv
C
C DESCRIPTION: Read input parameters from the input parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C   lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file Params.XXX (XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      Subroutine read_time_inv(lun)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

```

      Include 'envir.cmn'

```

```

      Integer    lun          !logical unit assigned input file
      Integer    i

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

      read( lun) nucmax, locmax, num_years, plantmax, apmax, num_dr,
      .          num_asl, num_foods, num_creameries, first_year,
      .          days_in_year, num_harvests, nucliat

      num_days=0
      Do I=1, num_years
         num_days=num_days + days_in_year(i)
      Enddo

      read( lun) xs, xr, ldecay
      read( lun) lapiash, ksen, kint, kg, ltrana, tfmilk, tf_animal
      read( lun) node_status, lperc, ps, pr, ileach, regime_fractions
      read( lun) bmax, bmin, lweather, cr
      read( lun) milk_source_fraction, milk_distr_fraction
      read( lun) sshield_factor, gshield_factor,
      .          submersion_dose_factor, groundahine_dose_factor,
      .          rootzone_dose_factor,
      .          ingestion_dose_factor, inhalation_dose_factor

      return
      end

```

C-----BEGIN DEFINITION BLOCK-----

C
C PROGRAM_NAME: Read_loc_inv
C
C DESCRIPTION: Read location invariant parameters from the input
C parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file ParamsYY.XXX (YY is year, XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C

C-----BEGIN IMPLEMENTATION BLOCK-----

C AUTHOR: Nila Reitz
C Battelle, Pacific Northwest Laboratory
C DATE:12/31/91
C
C MODIFICATION HISTORY:
C

CC

Subroutine read_loc_inv(lun)
implicit none

C-----BEGIN DECLARATION BLOCK-----

Include 'envir.cmn'

Integer lun !logical unit assigned input file

C-----BEGIN EXECUTABLE CODE-----

read(lun) tfresh_milk_fc, tfresh_milk_cr,
. tfresh_milk_gro, tstored_milk_gro, t_animal,
. breathing_rate, ref_food_consumption, fwash

return
end


```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Read_loc_var
C
C DESCRIPTION: Read location variant parameters from the input parameter file
C
C LANGUAGE: F77
C
C PARAMETERS:
C   lun - logical unit number of file I/O
C
C FILE I/O: Reads binary file ParamsYY.XXX (YY is year; XXX is run number)
C
C FUNCTIONS/SUBROUTINES CALLED: none
C

```

```

C-----BEGIN IMPLEMENTATION BLOCK-----
C

```

```

C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      Subroutine read_loc_var(lun)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----
      Include 'envir.cmn'

```

```

      Integer      lun          !logical unit assigned input file

```

```

C-----BEGIN EXECUTABLE CODE-----
      read(lun) pasture_begin,pasture_end,harvest_dates,
      .          tillage_dates
      read(lun) ml,vd
      read(lun) dairy_soil_ingestion,animal_soil_ingestion,
      .          dairy_feed_consumption,animal_feed_consumption
      return
      end

```

C3.20

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   COMMON BLOCK Dose
C
C   Date: 12/31/1991
C
C   Modification History:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      real*4 tdose_factor(6)
      real*4 ds(max_days,tlocmax),d_groundshine(max_days,tlocmax),
-      dinh(max_days,tlocmax),mingestion(max_days,tlocmax),
-      oingestion(max_days,tlocmax),masl(tnum_asl)
      real*4 basl(tnum_asl),easl(tnum_asl)
      data basl/0,1,5,10,15,20,0,1,5,10,15,20,21,
-      0,1,5,10,15,20,0,1,5,10,15,20,21/
      data easl/1,5,10,15,20,99,1,5,10,15,20,99,50,
-      1,5,10,15,20,99,1,5,10,15,20,99,50/
      data masl/.25,1,5,10,15,17,.25,1,5,10,15,17,21,
-      .25,1,5,10,15,17,.25,1,5,10,15,17,21/
      common/dose/tdose_factor,basl,easl,masl,ds,d_groundshine,dinh,
-      mingestion,oingestion

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   COMMON BLOCK Envir
C   This common block contains data used in the HEDR Phase II
C   environmental concentration model.
C   This data is read from the input parameter file and air model
C   output file into this common block.
C
C   Date: 12/31/1991
C
C   Modification History:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C   TIME INVARIANT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
integer num_days,num_nuc

```

```

Integer tnucmax,tlocmax,tnum_years,tplantmax,tapmax,
-       tnum_dr,tnum_harvests,tnum_asl,tnum_days,
-       tnum_foods,tnum_creameries,max_days

```

```

Parameter (tnum_years=1,tlocmax=961 ,tplantmax=9)
Parameter (tnucmax=1,tapmax=4,tnum_dr=4,tnum_harvests=3)
Parameter (tnum_asl=26,tnum_days=33,tnum_foods=11)
Parameter (tnum_creameries=26)

```

```

C
Parameter (max_days=33)
Real fd(tplantmax),fi(tplantmax)

```

```

Integer nucmax, locmax, num_years, plantmax, apmax,num_dr,
-       num_asl, num_foods, num_creameries, first_year,
-       days_in_year(tnum_years),num_harvests(tplantmax)
Character*5 nuclist(tnucmax)

```

```

Real*4 xs, xr, ldecay(tnucmax)

```

```

Real*4 lsplash(tplantmax), keen(tplantmax), kint(tplantmax),
-       kg(tplantmax), ltrans(tplantmax,tnucmax),tfmilk(tnucmax),
-       tf_animal(tapmax,tnucmax)

```

```

Integer node_status(tlocmax)
Real* 4 lperc(tlocmax),ps(tlocmax),pr(tlocmax),
-       lleach(tlocmax,tnucmax),
-       regime_fractions(tnum_dr,tlocmax)

```

```

Real *4 Bmax(tplantmax,tlocmax), Bmin(tplantmax,tlocmax),
-       lweather(tplantmax,tlocmax,tnucmax),
-       cr(tplantmax,tlocmax,tnucmax)

```

```

Real*4 milk_source_fraction(tlocmax,tnum_creameries),
-       milk_distr_fraction(tnum_creameries,tlocmax)

```

```

Real*4 sshield_factor(tnum_asl),gshield_factor(tnum_asl),
-       submersion_dose_factor(tnum_asl,tnucmax),
-       groundshine_dose_factor(tnum_asl,tnucmax),
-       rootzone_dose_factor(tnum_asl,tnucmax),
-       ingestion_dose_factor(tnum_asl,tnucmax),
-       inhalation_dose_factor(tnum_asl,tnucmax)

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C      Repeat records
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      LOCATION INVARIANT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Real*4 tfresh_milk_fc(max_days), tfresh_milk_cr(max_days),
      -      tfresh_milk_gro(max_days),
      -      tstored_milk_gro(max_days), t_animal(tapmax,max_days),
      -      breathing_rate(tnum_asl,max_days),
      -      ref_food_consumption(tnum_foods,tnum_asl,max_days),
      -      fwash(tplantmax,tnum_asl,max_days)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      LOCATION VARIANT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      Integer pasture_begin,
      -      pasture_end,
      -      harvest_dates(tnum_harvests,tplantmax),
      -      tillage_dates(tplantmax)

      Real* 4 ml(max_days),vd(max_days)

      Real*4 dairy_soil_ingestion(tnum_dr,max_days),
      -      animal_soil_ingestion(tapmax,max_days),
      -      dairy_feed_consumption(tnum_dr,tplantmax,max_days),
      -      animal_feed_consumption(tnum_dr,tplantmax,max_days)

C*****
C      air model
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Real*4 air_conc(tnum_days), surface_conc(tnum_days)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      Real*4 biomass(tplantmax,tnum_days),growth(tplantmax,tnum_days)
      Integer prev_harvest(tnum_harvests,tplantmax)

      COMMON/envir/fd,fi,num_days,
      -      nucmax, locmax, num_years, plantmax, apmax,num_dr,
      -      num_asl, num_foods, num_creameries, first_year,
      -      days_in_year, num_harvests,nuclist,node_status,
      -      pasture_begin, pasture_end,
      -      harvest_dates, tillage_dates, xs, xr, ldecay,
      -      lplash, ksen, kint, kg, ltrans,
      -      lperc, ps,pr,lleach,
      -      Bmax, Bmin, lweather, cr,
      -      ml,vd,
      -      tfresh_milk_fc, tfresh_milk_cr, tfresh_milk_gro,
      -      tstored_milk_gro, t_animal,dairy_soil_ingestion,
      -      animal_soil_ingestion, dairy_feed_consumption,
      -      animal_feed_consumption, regime_fractions,tfmilk, tf_animal,
      -      milk_source_fraction, milk_distr_fraction,
      -      breathing_rate, ref_food_consumption,fwash,
      -      sshield_factor,gshield_factor,
      -      submersion_dose_factor,groundshine_dose_factor,
      -      rootzone_dose_factor,
      -      ingestion_dose_factor,inhalation_dose_factor,
      -      air_conc, surface_conc,biomass,growth,
      -      prev_harvest

```


APPENDIX C4

REPGEN CODE LISTINGS

```

C ***** REPGEN.FOR *****
C
C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: REPGEN
C
C DESCRIPTION: REPGEN is the main program of the HEDRIC environmental/dose
C               report generator code.
C
C LANGUAGE: F77
C
C FILE I/O:
C   Input Concentration Files : I-131 LLLL.YY
C   (LLLL is location (node) number; YY is last 2 digits of year)
C   Input Dose Estimate Files: I-131 DDDDD.YY
C   (DDDDD is individual's ID number; YY is last 2 digits of year)
C   Input files containing concentration and dose variable descriptors:
C   CONC_HEADERS.ASC (concentration variables)
C   DOSE_HEADERS.ASC (dose variables)
C
C   Output report file name is entered by user.
C
C FUNCTIONS/SUBROUTINES CALLED:
C
C Valid_date
C Determine_doy
C Select_model_output
C Temporal_aggregate
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Robert A. Burnett
C         Battelle, Pacific Northwest Laboratory
C
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
C-----BEGIN DECLARATION BLOCK-----
C
C implicit none
C integer*4 MAX_CONC, MAX_DOSE, MAX_TOT, MAX_VAR, MAX_LOC
C integer*4 MAX_COLS, MAX_DAYS
C parameter (MAX_CONC = 20) ! Total number of available conc. variables
C parameter (MAX_DOSE = 6) ! Total number of available dose variables
C parameter (MAX_TOT = MAX_CONC + MAX_DOSE)
C parameter (MAX_VAR = 200)
C parameter (MAX_LOC = 961)
C parameter (MAX_COLS = 10) ! Max. columns of variables across page
C parameter (MAX_DAYS = 371) ! Max. no. of days per year (1945 + 6 days)
C-- Function subprogram declarations --
C   integer*4 temporal_aggregate
C-- Local variables --
C   real*4 VY(MAX_COLS) ! Yearly averages or sums
C   real*4 VYC(MAX_COLS) ! Yearly cumulatives
C   real*4 VQ(4,MAX_COLS) ! Quarterly averages or sums
C   real*4 VQC(4,MAX_COLS) ! Quarterly cumulatives
C   real*4 VM(12,MAX_COLS) ! Monthly averages or sums
C   real*4 VMC(12,MAX_COLS) ! Monthly cumulatives
C   real*4 VD(MAX_DAYS, MAX_COLS) ! Daily averages or sums
C   real*4 VDC(MAX_DAYS, MAX_COLS) ! Daily cumulatives
C   real*4 VIC(MAX_COLS) ! Initial cumul. values (at begin. of year)
C   real*4 VTEMP(MAX_DAYS) ! Temporary storage of file variable values
C   integer*4 NC, isat, DAYS_IN_YEAR, NMONTH, NQUARTER, YTYPE
C   integer*4 FBEGIN, FEND
C   character*1 answer, resp, temp_agg
C   character*10 REP_BEGIN_DATE, REP_END_DATE

```

```

CHARACTER*48 REPORT FILE, RESULTS FILE
character*12 HEAD1(MAX COLS), HEAD2(MAX COLS), HEAD3(MAX COLS)
character*12 CHEAD1(MAX CONC), CHEAD2(MAX CONC), CHEAD3(MAX CONC)
character*12 DHEAD1(MAX DOSE), DHEAD2(MAX DOSE), DHEAD3(MAX DOSE)
integer*4 begin_doy, end_doy, begin_year, end_year
character*5 PERSON ID
character*9 MNAME(I2)
integer*4 J, K, M, MQ, N BEGIN, N END, N COUNT, var(MAX VAR)
integer*4 num_conc, num_dose, num_locs, indx, LOC, YEAR
integer*4 conc_list(MAX CONC), dose_list(MAX DOSE)
integer*4 conc_begin(MAX CONC), dose_begin(MAX DOSE)
integer*4 conc_end(MAX CONC), dose_end(MAX DOSE)
integer*4 MEND(12,3) ! Julian date of last day of month for 3 year types
integer*4 loc_list(MAX LOC)
logical VALID_DATE, DAILY OUT, MONTHLY OUT, QUARTERLY OUT
data conc_begin /1,3,4,5,7,8,9,10,11,12,13,14,15,16,17,22,
$              18, 19, 20, 21 /
data conc_end   /2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,22,
$              18, 19, 20, 21 /
data dose_begin / 2, 3, 5, 4, 1, 1 /
data dose_end   / 2, 3, 5, 4, 1, 5 /
C- Julian date of last day of month for 3 year types: non-leap, leap, "1945+"
data MEND / 31,59,90,120,151,181,212,243,273,304,334,365,
$          31,60,91,121,152,182,213,244,274,305,335,366,
$          37,65,96,126,157,187,218,249,279,310,340,371 /
C-- Names of the months
data MNAME / 'January ', 'February ', 'March ', 'April ',
$           'May ', 'June ', 'July ', 'August ',
$           'September ', 'October ', 'November ', 'December ' /

```

```

C----- BEGIN EXECUTION BLOCK -----
C

```

```

C4.2 C-- Get user inputs:
11  WRITE (*,899)
899  FORMAT(/, ' Select the set of variables to be reported:',/,
$      '      A - All variables (concentration and dose)',/,
$      '      C - all Concentration variables',/,
$      '      D - all Dose variables',/,
$      '      E - Enter individual variables from a menu')
WRITE(*,900)
900  FORMAT($, ' >> ')
READ (*,902,end=9999) answer
902  format(A)
IF (answer .eq. 'a' .or. answer .eq. 'A') THEN
    DO K = 1, MAX CONC
        conc_list(K) = K
    ENDDO
    DO K = 1, MAX DOSE
        dose_list(K) = K
    ENDDO
    num_conc = MAX CONC
    num_dose = MAX DOSE
ELSE IF (answer .eq. 'c' .or. answer .eq. 'C') THEN
    DO K = 1, MAX CONC
        conc_list(K) = K
    ENDDO
    num_conc = MAX CONC
    num_dose = 0
ELSE IF (answer .eq. 'd' .or. answer .eq. 'D') THEN
    DO K = 1, MAX DOSE
        dose_list(K) = K
    ENDDO
    num_conc = 0
    num_dose = MAX DOSE
ELSE IF (answer .eq. 'e' .or. answer .eq. 'E') THEN
    DO K = 1, MAX TOT

```



```

        var(K) = 0
        ENDDO
        write (*,903)
903    format(/,' CONCENTRATIONS:',/,
$ ' 1- Leafy Veg. 2- Other Veg. 3- Grain ',
$ ' 4- Fruit',/,
$ ' 5- Alfalfa 6- Pasture grass 7- Grass/Upper Soil',
$ ' 8- Grass/Root Zone',/,
$ ' 9- Silage 10- Sagebrush 11- Grass Hay ',
$ ' 12- Beef',/,
$ ' 13- Venison 14- Poultry 15- Eggs ',
$ ' 16- Grocery Milk',/,
$ ' 17- Reg.1 Milk 18- Reg.2 Milk 19- Reg.3 Milk ',
$ ' 20- Reg.4 Milk'/)
        write(*,904)
904    format(' DOSES:',/,
$ ' 21- Submersion 22- Groundshine 23- Non-Milk Ingest.',
$ ' 24- Milk Ingest.',/,
$ ' 25- Inhalation 26- Total',/,/,7X,
$ 'Enter one or more numbers corresponding to the variables',
$ ' you wish to',/,7X,'include in the report. Separate each',
$ ' number by a comma, by one or more',/,7X,'spaces, or by a',
$ ' <RETURN>. End the list by entering <CTRL-Z>.')
        write(*,900)
        read(*,*,end=21) (var(K),K=1,MAX_VAR)
21    CONTINUE
        num_conc = 0
        num_dose = 0
        DO J = 1, MAX_VAR
            IF (var(J) .gt. 0 .and. var(J) .le. MAX_CONC) THEN
                num_conc = num_conc + 1
                conc_list(num_conc) = var(J)
            ELSE IF (var(J) .gt. MAX_CONC .and. var(J) .le. MAX_TOT) THEN
                num_dose = num_dose + 1
                dose_list(num_dose) = var(J) - MAX_CONC
            ENDIF
        ENDDO
        ELSE
            write (*,909)
909    format(' Please enter one of the letters shown on the menu.',/)
            go to 11
        ENDIF

C
IF(num_conc .le. 0 .and. num_dose .le. 0) go to 11

C
IF (num_conc .gt. 0) THEN
    write (*,*) ' The concentration variables selected are:'
    N_BEGIN = 1
C
23    CONTINUE
    N_END = min(num_conc, N_BEGIN + 9)
    N_COUNT = N_END - N_BEGIN + 1
    write (*,911) (conc_list(K), K = N_BEGIN, N_END)
911    format(1x,<N_COUNT>I7)
    N_BEGIN = N_END + 1
    IF(N_BEGIN .le. num_conc) go to 23
C
ENDIF
IF (num_dose .gt. 0) THEN
    write (*,*) ' The dose variables selected are:'
    N_BEGIN = 1
C
26    CONTINUE
    N_END = min(num_dose, N_BEGIN + 9)
    N_COUNT = N_END - N_BEGIN + 1
    write (*,911) (dose_list(K) + MAX_CONC, K = N_BEGIN, N_END)

```

```

      N_BEGIN = N_END + 1
      IF (N_BEGIN .le. num_dose) go to 26
C
      ENDIF
      IF (answer .eq. 'e' .or. answer .eq. 'E') then
31      write (*,*) ' Is this the list of variables ',
      $      ' you wish to include in the report? (Y/N)'
      write (*,900)
      read (*,902) resp
      if (resp .eq. 'N' .or. resp .eq. 'n') then
          GOTO 16
      else if (resp .ne. 'y' .and. resp .ne. 'Y') then
          WRITE (*,*) ' Please answer Y (y) or N (n)'
          GOTO 31
      ENDIF
C-- Response was Y or y
      ENDIF
C
C-- If any concentration variables were chosen, ask for location
      indx = 0
      IF (num_conc .gt. 0) THEN
33      indx = indx + 1
34      write(*,*)
      $      ' Enter a location for reporting concentration results;',
      $      ' Press <CTRL-Z> to end list:'
      write(*,900)
      read (*,*,end=35) loc_list(indx)
      IF (loc_list(indx).le.0 .or. loc_list(indx).gt.MAX_LOC) THEN
915      write(*,915) loc_list(indx)
          format(' Location',I7,' is not a valid grid cell number.')
          go to 34
      ELSE
          go to 33
      ENDIF
C-- Come here on <CTRL-Z>
35      CONTINUE
      IF (indx .eq. 1) THEN      ! If no locations were entered
          write(*,*) ' Please enter at least one location.'
          go to 34
      ENDIF
      num_locs = indx - 1
      ENDIF
C
C-- If any dose variables were selected get ID number of individual
      IF (num_dose .gt. 0) THEN
          write(*,*)
      $      ' Enter the 5-character ID number of the individual:'
          write(*,900)
          read (*,902) PERSON_ID
      ENDIF
C
C-- Get beginning and ending dates for the report --
41      write(*,*)
      $      ' Enter the beginning date for reporting results (mm/dd/yyyy):'
      write(*,900)
      read (*,917) REP_BEGIN_DATE
917      format(A10)
      IF (.not.valid_date(REP_BEGIN_DATE)) THEN
          write(*,*) ' Not a valid date.'
          go to 41
      ENDIF
46      write(*,*)
      $      ' Enter the ending date for reporting results (mm/dd/yyyy):'
      write(*,900)
      read (*,917) REP_END_DATE
      IF (.not.valid_date(REP_END_DATE)) THEN

```

```

        write(*,*) ' Not a valid date.'
        go to 46
    ENDIF
    call determine_doy (REP_BEGIN_DATE, begin_doy, begin_year)
    call determine_doy (REP_END_DATE, end_doy, end_year)
    begin_year = begin_year - 1900
    IF (begin_year .eq. 44) then
        begin_year = 45
        if (begin_doy .gt. 360) begin_doy = begin_doy - 360
    ENDIF
    end_year = end_year - 1900
CCC    read(REP_BEGIN_DATE(9:10),919) begin_year    ! Last 2 digits of 1st year
CCC    read(REP_END_DATE(9:10),919) end_year    ! Last 2 digits of last year
919    format(I2)
C
C-- Select output options (temporal resolution and output file name)
51    write (*,921)
921    format(/,
$ ' Select one of the following levels of temporal resolution',
$ ' for reporting:',/,
$ '      D - Daily values with monthly and yearly aggregation'/
$ '      M - Monthly and yearly aggregated values'/
$ '      Q - Quarterly and yearly aggregated values',/
$ '      Y - Yearly aggregated values only')
    write (*,900)
    read (*,902) temp_agg
    DAILY_OUT = .FALSE.
    MONTHLY_OUT = .FALSE.
    QUARTERLY_OUT = .FALSE.
    IF (temp_agg .eq. 'd' .or. temp_agg .eq. 'D') THEN
        DAILY_OUT = .TRUE.
        MONTHLY_OUT = .TRUE.
    ELSE IF (temp_agg .eq. 'm' .or. temp_agg .eq. 'M') THEN
        MONTHLY_OUT = .TRUE.
    ELSE IF (temp_agg .eq. 'q' .or. temp_agg .eq. 'Q') THEN
        QUARTERLY_OUT = .TRUE.
    ELSE IF (temp_agg .ne. 'y' .and. temp_agg .ne. 'Y') THEN
        Write (*,909)
        go to 51
    ENDIF
C
C-- Prompt for output file name
    write (*,*) ' Enter the file name of the output report:'
    write (*,900)
    read (*,902) REPORT_FILE
C-- Open report file
    OPEN(unit=14, file=REPORT_FILE, status='NEW')
C
    IF (NUM_CONC .gt. 0) THEN
C *****
C ***** Report Concentration data *****
C *****
C
C-- Read conc. variable descriptions from file
    OPEN (unit=11,file='[HEDR.PHASE2.DATA]CONC_HEADERS.ASC',
$        status='OLD', READONLY)
    DO J = 1, MAX_CONC
        READ (11,923) CHEAD1(J),CHEAD2(J),CHEAD3(J)
923    format(A12/A12/A12)
    ENDDO
    CLOSE(unit=11)
    DO J = 1, NUM_CONC
        HEAD1(J) = CHEAD1(CONC_LIST(J))
        HEAD2(J) = CHEAD2(CONC_LIST(J))
        HEAD3(J) = CHEAD3(CONC_LIST(J))
    ENDDO

```

```

DO LOC = 1, NUM_LOCS
  RESULTS_FILE(I:6) = 'I-131'
  write(RESULTS_FILE(7:10), '(I4.4)') LOC_LIST(LOC)
  RESULTS_FILE(I:11) = '.'
  DO YEAR = BEGIN_YEAR, END_YEAR
    IF (YEAR .eq. 45) THEN
      DAYS_IN_YEAR = 371
      YTYPE = 3 ! year type 3 (1945: all of 1945 + last 6 days of 1944)
    ELSE IF (mod(YEAR,4) .eq. 0) THEN
      DAYS_IN_YEAR = 366
      YTYPE = 2 ! year type 2 (leap year)
    ELSE
      DAYS_IN_YEAR = 365
      YTYPE = 1 ! year type 1 (non-leap, not 1945)
    ENDIF
    C-- Complete the file name by adding last 2 digits of year
    write(RESULTS_FILE(12:13), '(I2)') YEAR
    C ##### DEBUG #####
    CCC write(*,*) RESULTS_FILE
    C ##### DEBUG #####
    OPEN(unit=13, file=RESULTS_FILE, status='old',
      $ form='unformatted', access='direct', READONLY)
    NC = min(NUM_CONC, MAX_COLS)
    DO J = 1, NC
      DO K = 1, 371 ! Zero out the daily values array
        VD(K,J) = 0.
      ENDDO
      VIC(J) = 0.
    C-- Get file variable number(s) that correspond to this reported conc. result
    FBEGIN = conc_begin(J)
    FEND = conc_end(J)
    C-- Read conc. data for entire year
    call select_model_output(13, FBEGIN, VD(1,J),
      $ 1, DAYS_IN_YEAR, YEAR)
    IF(FEND.GT.FBEGIN) THEN
    C-- Add other file variables to this reported variable
      DO K = FBEGIN+1, FEND
        call select_model_output(13, K, VTEMP,
          $ 1, DAYS_IN_YEAR, YEAR)
        DO M = 1, DAYS_IN_YEAR
          VD(M,J) = VD(M,J) + VTEMP(M)
        ENDDO ! for each day of year
      ENDDO ! for each added file variable
    ENDIF ! if there are other file variables to add
    C-- Aggregate this report variable
    istat=temporal_aggregate('A',DAYS_IN_YEAR,VD(1,J),VIC(J),
      $ VM(1,J),VQ(1,J),VY(J),VDC(1,J),VMC(I,J),VQC(1,J),VYC(J))
    ENDDO ! for each reported conc. variable
  C
  C-- Write results to report file
  C-- Write general information at beginning of report
  IF (temp_agg .eq. 'D' .or. temp_agg .eq. 'd') THEN
    write(14,990) LOC, YEAR
  ELSE IF (temp_agg .eq. 'M' .or. temp_agg .eq. 'm') THEN
    write(14,991) LOC, YEAR
  ELSE IF (temp_agg .eq. 'Q' .or. temp_agg .eq. 'q') THEN
    write(14,992) LOC, YEAR
  ELSE
    write(14,993) LOC, YEAR
  ENDIF
  990 FORMAT(
    $ ' *** Estimated Environmental/Food Product Concentrations ***',
    $ '//, ' Location:',I6,/,
    $ ' DAILY, MONTHLY, AND YEARLY AVERAGES FOR ',I4/)
  991 FORMAT(
    $ ' *** Estimated Environmental/Food Product Concentrations ***',

```

```

      $ //,' Location:',I6,/,
$ ' MONTHLY AND YEARLY AVERAGES FOR ',I4/)
992   FORMAT(
$ ' *** Estimated Environmental/Food Product Concentrations ***',
$ //,' Location:',I6,/,
$ ' QUARTERLY AND YEARLY AVERAGES FOR ',I4/)
993   FORMAT(
$ ' *** Estimated Environmental/Food Product Concentrations ***',
$ //,' Location:',I6,/,
$ ' YEARLY AVERAGES FOR ',I4/)
C * * * * *
C-- Write column headers
      write (14,1001) (HEAD1(M),M=1,NC)
1001   format(10X,<NC>A12)
      write (14,1002) (HEAD2(M),M=1,NC)
1002   format('      Time ',<NC>A12)
      write (14,1003) (HEAD3(M),M=1,NC)
1003   format('      Period ',<NC>A12)
C-- Write results depending on aggregation levels desired
      NMONTH = 1
      NQUARTER = 1
      DO K = 1, DAYS IN YEAR
        IF (DAILY OUT) THEN
          WRITE (14,1005) K, (VD(K,M),M=1,NC)
1005   FORMAT(' Day =',I4,<NC>PE12.4)
        ENDIF
        IF (MONTHLY OUT) THEN
          IF (K.EQ. MEND(NMONTH,YTYPE)) THEN
            WRITE (14,1006) MNAME(NMONTH), (VM(NMONTH,M),M=1,NC)
1006   FORMAT(1X,A9,<NC>PE12.4)
            NMONTH = NMONTH + 1
          ENDIF
        ENDIF
        IF (QUARTERLY OUT) THEN
          MQ = 3*NQUARTER
          IF (K.EQ. MEND(MQ,YTYPE)) THEN
            WRITE (14,1007) NQUARTER, (VQ(NQUARTER,M),M=1,NC)
1007   FORMAT(' Quarter ',I1,<NC>PE12.4)
            NQUARTER = NQUARTER + 1
          ENDIF
        ENDIF
      ENDDO ! on days in year
C-- Always report the yearly averages
      write (14,1008) YEAR, (VY(M),M=1,NC)
1008   format(
$ ' -----',
$ //,' YEAR 19',I2,<NC>PE12.4)
      CLOSE(unit=13) ! Close conc. database file for this year and loc.
      ENDDO ! on YEAR
      ENDDO ! on LOC
      ENDIF ! End of reporting concentration data
C=====
      IF (NUM DOSE .gt. 0) THEN
C *****
C ***** Report Dose data *****
C *****
C
C-- Read dose variable descriptions from file
      OPEN (unit=11,file='(HEDR.PHASE2.DATA)DOSE_HEADERS.ASC',
$        status='OLD', READONLY)
      DO J = 1, MAX DOSE
        READ (11,923) DHEAD1(J),DHEAD2(J),DHEAD3(J)
      ENDDO
      CLOSE(unit=11)
      DO J = 1, NUM DOSE
        HEAD1(J) = DHEAD1(DOSE LIST(J))

```

```

HEAD2(J) = DHEAD2(DOSE_LIST(J))
HEAD3(J) = DHEAD3(DOSE_LIST(J))
ENDDO
RESULTS_FILE(1:6) = 'I-131'
RESULTS_FILE(7:11) = PERSON_ID
RESULTS_FILE(12:12) = '.'
DO YEAR = BEGIN_YEAR, END_YEAR
  IF (YEAR .eq. 45) THEN
    DAYS_IN_YEAR = 371
    YTYPE = 3 ! year type 3 (1945: all of 1945 + last 6 days of 1944)
  ELSE IF (mod(YEAR,4) .eq. 0) THEN
    DAYS_IN_YEAR = 366
    YTYPE = 2 ! year type 2 (leap year)
  ELSE
    DAYS_IN_YEAR = 365
    YTYPE = 1 ! year type 1 (non-leap, not 1945)
  ENDIF
C-- Complete the file name by adding last 2 digits of year
  write (RESULTS_FILE(13:14), '(I2)') YEAR
C ##### DEBUG #####
CCC write(*,*) RESULTS_FILE
C ##### DEBUG #####
  OPEN(unit=13, file=RESULTS_FILE, status='old',
    $ form='unformatted', access='direct', READONLY)
  NC = min (NUM_DOSE, MAX_COLS)
  DO J = 1, NC
    DO K = 1, 371 ! Zero out the daily values array
      VD(K,J) = 0.
    ENDDO
    VIC(J) = 0.
C-- Get file variable number(s) that correspond to this reported dose result
    FBEGIN = dose_begin(J)
    FEND = dose_end(J)
C-- Read dose data for entire year
    call select_model_output(13, FBEGIN, VD(1,J),
      $ 1, DAYS_IN_YEAR, YEAR)
    IF (FEND.GT.FBEGIN) THEN
C-- Add other file variables to this reported variable
      DO K = FBEGIN+1, FEND
        call select_model_output(13, K, VTEMP,
          $ 1, DAYS_IN_YEAR, YEAR)
        DO M = 1, DAYS_IN_YEAR
          VD(M,J) = VD(M,J) + VTEMP(M)
        ENDDO ! for each day of year
      ENDDO ! for each added file variable
    ENDIF ! if there are other file variables to add
C-- Aggregate this report variable
    istat=temporal_aggregate('S', DAYS_IN_YEAR, VD(1,J), VIC(J),
      $ VM(1,J), VQ(1,J), VY(J), VDC(1,J), VMC(1,J), VQC(1,J), VYC(J))
    ENDDO ! for each reported dose variable
C
C-- Write results to report file
C-- Write general information at beginning of report
  IF (temp_agg .eq. 'D' .or. temp_agg .eq. 'd') THEN
    write (14, 1990) LOC, YEAR
  ELSE IF (temp_agg .eq. 'M' .or. temp_agg .eq. 'm') THEN
    write (14, 1991) LOC, YEAR
  ELSE IF (temp_agg .eq. 'Q' .or. temp_agg .eq. 'q') THEN
    write (14, 1992) LOC, YEAR
  ELSE
    write (14, 1993) LOC, YEAR
  ENDIF
1990 FORMAT(
  $ '1', ' *** Estimated Individual Doses ***',
  $ '//', ' Individual ID: ', A5, '//',
  $ ' DAILY, MONTHLY AND YEARLY TOTALS FOR ', I4/)

```

```

1991.      FORMAT(
$ '1',' *** Estimated Individual Doses ***',
$ //,' Individual ID: ',A5,//,
$ ' MONTHLY AND YEARLY TOTALS FOR ',I4/)
1992      FORMAT(
$ '1',' *** Estimated Individual Doses ***',
$ //,' Individual ID: ',A5,//,
$ ' QUARTERLY AND YEARLY TOTALS FOR ',I4/)
1993      FORMAT(
$ '1',' *** Estimated Individual Doses ***',
$ //,' Individual ID: ',A5,//,
$ ' YEARLY TOTALS FOR ',I4/)
C * * * * *
C-- Write column headers
      write (14,1001) (HEAD1(M),M=1,NC)
      write (14,1002) (HEAD2(M),M=1,NC)
      write (14,1003) (HEAD3(M),M=1,NC)
C-- Write results depending on aggregation levels desired
      NMONTH = 1
      NQUARTER = 1
      DO K = 1, DAYS IN YEAR
        IF (DAILY OUT) THEN
          WRITE (14,1005) K, (VD(K,M),M=1,NC)
        ENDIF
        IF (MONTHLY OUT) THEN
          IF (K .EQ. MEND(NMONTH,YTYPE)) THEN
            WRITE (14,1006) MNAME(NMONTH), (VM(NMONTH,M),M=1,NC)
            NMONTH = NMONTH + 1
          ENDIF
        ENDIF
        IF (QUARTERLY OUT) THEN
          MQ = 3*NQUARTER
          IF (K .EQ. MEND(MQ,YTYPE)) THEN
            WRITE (14,1007) NQUARTER, (VQ(NQUARTER,M),M=1,NC)
            NQUARTER = NQUARTER + 1
          ENDIF
        ENDIF
      ENDDO ! on days in year
C-- Always report the yearly totals
      write (14,1008) YEAR, (VY(M),M=1,NC)
      CLOSE(unit=13) ! Close database file for this year and loc.
      ENDDO ! on YEAR
      ENDIF ! End of reporting dose data
C
9999      STOP
          END

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Valid_date
C
C DESCRIPTION: Determines if a date is valid.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   date - user entered date
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      function valid_date(date)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

```

      integer day,month,year
      integer dim(12,2)
      integer quo,rem
      logical century,cdlv,ydiv
      character*10 date
      logical vyear,vmonth,vday
      logical leap_year,valid_date

      data dim/31,28,31,30,31,30,31,31,30,31,30,31,
      - 31,29,31,30,31,30,31,31,30,31,30,31/

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

      read(date(1:2),100) month
      read(date(4:5),100) day
      read(date(7:10),101) year
100  format(i2)
101  format(i4)

```

```

      leap_year = .false.
      vday = .false.
      vmonth = .false.
      vyear = .false.
      quo=year/100
      rem=year-(quo*100)
      if (rem.eq.0) then
        quo=year/400
        rem=year-(quo*400)
        if (rem.eq.0) then
          leap_year = .true.
        endif
      else ! not a century
        quo=year/4
        rem=year-(quo*4)
        if (rem.eq.0) then

```



```

        leap_year = .true.
    endif
endif

if (year .ge. 1800 .and. year .le. 1991) then
    vyear=.true.
    if ((month.gt.0).and.(month.lt.13)) then
        vmonth=.true.
        if (leap_year) then
            if (day.le.dim(month,2) .and. day.gt.0) vday=.true.
        else
            if (day.le.dim(month,1) .and. day.gt.0) vday=.true.
        endif
    endif
endif
endif
if (vyear.and.vmonth.and.vday) then
    valid_date=.true.
else
    valid_date=.false.
endif
return
end

```

```

C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: Determine_doy
C
C DESCRIPTION: Determine the day of year of a date.
C
C LANGUAGE: F77
C
C PARAMETERS:
C   date - user entered date
C   doy - day of year of date
C   year - year of current date
C
C FILE I/O: none
C
C FUNCTIONS/SUBROUTINES CALLED: none
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C AUTHOR: Nila Reitz
C   Battelle, Pacific Northwest Laboratory
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

      subroutine determine_doy(date,doy,year)
      implicit none

```

```

C-----BEGIN DECLARATION BLOCK-----

```

```

      character*10 date
      integer doy, YEAR

      integer day, month
      integer dim(12,2)
      integer quo, rem, i
      logical leap_year

      data dim/31,28,31,30,31,30,31,31,30,31,30,31,
-          31,29,31,30,31,30,31,31,30,31,30,31/

```

```

C-----BEGIN EXECUTABLE CODE-----

```

```

      read(date(1:2),100) month
      read(date(4:5),100) day
      read(date(7:10),101) year
100 format(i2)
101 format(i4)

      leap_year = .false.
      quo=year/100
      rem=year-(quo*100)
      if (rem.eq.0) then
        quo=year/400
        rem=year-(quo*400)
        if (rem.eq.0) then
          leap_year = .true.
        endif
      else ! not a century
        quo=year/4
        rem=year-(quo*4)
        if (rem.eq.0) then
          leap_year = .true.
        endif

```

```

endif
doy = day
if (leap_year) then
  if (month .gt. 1) then
    do i=1,month-1
      doy=doy+dim(i,2)
    enddo
  endif
else ! Not a leap year; check for 02/29 (birthday?)
  if (month .eq. 2 .and. day .eq. 29) then
    doy = 60 ! Feb. 29 birthday on non-leap year: use March 1
  else if (month .gt. 1) then
    do i=1,month-1
      doy=doy+dim(i,1)
    enddo
  endif
endif
endif

```

C-- If year is 1945, then add 6 days (last 6 days of 1944) to the day-of-year.

```

  if (year .eq. 1945) then
    doy = doy + 6
  
```

C-- If date is between 12/26/44 and 12/31/44, consider it to be part of 1945.

```

  else if (year.eq.1944 .and. month.eq.12 .and. day .ge. 26) then
    year = 1945
    doy = day - 25
  endif

```

```

return
end

```

```

C ***** SELECT_MODEL_OUTPUT.FOR *****
C
C-----BEGIN DEFINITION BLOCK-----
C
C PROGRAM_NAME: SELECT_MODEL_OUTPUT
C
C DESCRIPTION: This subroutine retrieves specified concentration values from
C               an environmental accumulation database file, or dose values from
C               a dose estimate file. It retrieves values for specified
C               file variables and time periods.
C
C LANGUAGE: F77
C
C SUBROUTINE ARGUMENTS: (See INPUT ARGUMENTS and OUTPUT ARGUMENTS in
C                       DECLARATION block.)
C
C FILE I/O:
C   Input Model Results file (Concentration or Dose) -- the file must
C   be opened by the calling program on logical unit "eun".
C
C FUNCTIONS/SUBROUTINES CALLED: None.
C
C-----BEGIN IMPLEMENTATION BLOCK-----
C
C AUTHOR: Robert A. Burnett
C         Battelle, Pacific Northwest Laboratory
C
C DATE: 12/31/91
C
C MODIFICATION HISTORY:
C
C-----BEGIN DECLARATION BLOCK-----
C
C   subroutine select_model_output (eun,envir_var,evalues,
C   *                               bday,eday,year)
C   implicit none
C
C   include 'envir.cmn'
C   include 'calc_conc.cmn'
C
C-- INPUT ARGUMENTS --
C   integer eun           ! Logical unit number of input file to be read
C   integer envir_var     ! File variable number for which to retrieve data
C   integer bday          ! First Julian day within year to retrieve values
C   integer eday          ! Last Julian day within year to retrieve values
C   integer year          ! Year - 1900 (last 2 digits of year)
C
C-- OUTPUT ARGUMENTS --
C   real*4 evalues(371)  ! Array to receive the daily conc. or dose values
C
C-- Local Variables
C   integer i,j,mt_period
C   integer start_value
C   real*4 tconc(32)
C   integer begin_day(15),code(15)
C   integer data_rec1,data_rec2,rec_count
C   integer envir_var,bday,eday
C   integer start_rec,endv,start,current_day
C   integer bperiod,eperiod
C   integer new_count,old_count,k,l
C   integer lperiod
C
C   data mt_period/15/

```

C--- BEGIN EXECUTION BLOCK -----

```

C
  read(unit=eun,rec=(envir var+1)) data_rec1,data_rec2,
  *   (begin_day(1),code(1),1=1,mt_period)

  if (data_rec1.eq.0) then
    write(*,105) envir var
105    format('There is no data for variable ',i3)
    goto 900
  endif

  do i=1,371
    evalues(i)=0
  enddo
  do i=1,32
    tconc(i)=0
  enddo

  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  C Define which periods bday and eday are in (1,mt_period).
  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

  bperiod=0
  eperiod=0
  lperiod=0
  do j=1,mt_period
    if (begin_day(j).eq.0) then
      lperiod=j-1
      goto 100
    endif
  enddo

100  if (bday.gt.days_in_year(year)) then
    bperiod=lperiod+1
    eperiod=lperiod+1
  else
    do i=1,lperiod
      if (bday.ge.begin_day(i)) then
c      $      (bday.le.begin_day(i+1))) then
        bperiod=i
      endif
      if ((eday.ge.begin_day(i)).and.
        $      (eday.le.days_in_year(year))) then
        eperiod=i
      endif
    enddo
  endif
  if (eperiod.eq.0) then
    eperiod=lperiod+1
  endif

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C Find 1st data value in toonc that is used
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  start_value=1
  start=1
  endv=bperiod-1
  do j=start,endv
    if (code(j).eq.0) then
      start_value=start_value+(begin_day(j+1)-begin_day(j))
    endif
  enddo

```

```

start_rec=start_value/32+1
rec_count=data_rec1
do i=data_rec1,(data_rec1+start_rec-1)
  read(unit=eun,rec=1) (tconc(i),
$      i=1,32)
  rec_count=rec_count+1
enddo
old_count=(start_value-(start_value/32*32))
new_count=1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Fill in values
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
current_day=bday
do j=bperiod,eperiod
  if (j.gt.lperiod) then
    endv=eday
    do k=current_day,endv
      evalues(new_count)--2
      new_count=new_count+1
    enddo
    current_day=eday+1
  else
    if (j.eq.eperiod) then
      endv=eday
    elseif (begin_day(j+1).eq.0) then
      endv=days_in_year(year)
    else
      endv=begin_day(j+1)-1
    endif
    do k=current_day,endv
      evalues(new_count)=tconc(old_count)
      new_count=new_count+1
      old_count=old_count+1
      if (old_count.gt.32) then
        read(unit=eun,rec=rec_count) (tconc(1),
$          i=1,32)
        rec_count=rec_count+1
        old_count=1
      endif
    enddo
    current_day=endv+1
  endif
enddo
900 return
end

```



```

C-----BEGIN EXECUTION BLOCK-----
C
C-- Check year type: (1) non-leap, (2) leap, or: (3) 1945 + 6 days of 1944
IF (DAYS IN YEAR .EQ. 365) THEN
  YEAR_TYPE = 1
ELSE IF (DAYS IN YEAR .EQ. 366) THEN
  YEAR_TYPE = 2
ELSE IF (DAYS IN YEAR .EQ. 371) THEN
  YEAR_TYPE = 3
ELSE
  WRITE(*,*)
  $ ' TEMPORAL AGGREGATE error - improper number of days in year:',
  $   DAYS IN YEAR
  TEMPORAL_AGGREGATE = -1
  RETURN
ENDIF
C-- Calculate daily cumulatives from daily values
C-- (we will use cumulatives in calculating either averages or sums)
VDC(1) = VIC + VD(1)
DO J = 2, DAYS IN YEAR
  VDC(J) = VDC(J-1) + VD(J)
END DO
C-- Check aggregation method --
IF (AVE_OR_SUM .eq. 'S' .or. AVE_OR_SUM .eq. 's') THEN
C----- Aggregate by sums and calculate cumulatives -----
C- First calculate monthly totals and cumulatives
  J = EOM(1, YEAR_TYPE)
  VMC(1) = VDC(J)
  VM(1) = VDC(J) - VIC
  DO MONTH = 2, 12
    J = EOM(MONTH, YEAR_TYPE)
    VMC(MONTH) = VDC(J)
    VM(MONTH) = VMC(MONTH) - VMC(MONTH-1)
  ENDDO
C-- Now calculate quarterly totals and cumulatives
  J = EOM(3, YEAR_TYPE)
  VQC(1) = VDC(J)
  VQ(1) = VDC(J) - VIC
  DO QEND = 2, 4
    MONTH = 3 * QEND
    J = EOM(MONTH, YEAR_TYPE)
    VQC(QEND) = VDC(J)
    VQ(QEND) = VQC(QEND) - VQC(QEND-1)
  ENDDO
C-- Now calculate yearly total and cumulative
  J = EOM(12, YEAR_TYPE)
  VYC = VDC(J)
  VY = VYC - VIC
C
  ELSE IF (AVE_OR_SUM .eq. 'A' .or. AVE_OR_SUM .eq. 'a') THEN
C----- Aggregate by averages -----
C- First, calculate monthly averages
  J = EOM(1, YEAR_TYPE)
  VMC(1) = VDC(J)
  VM(1) = (VDC(J) - VIC) / J
  K = J
  DO MONTH = 2, 12
    J = EOM(MONTH, YEAR_TYPE)
    VMC(MONTH) = VDC(J)
    VM(MONTH) = (VMC(MONTH) - VMC(MONTH-1)) / (J - K)
    K = J
  ENDDO
C-- Now calculate quarterly averages
  J = EOM(3, YEAR_TYPE)
  VQC(1) = VDC(J)
  VQ(1) = (VDC(J) - VIC) / J

```



```

      K = J
      DO QEND = 2, 4
        MONTH = 3 * QEND
        J = EOM (MONTH, YEAR_TYPE)
        VQC(QEND) = VDC(J)
        VQ(QEND) = (VQC(QEND) - VQC(QEND-1)) / (J - K)
        K = J
      ENDDO
C-- Now calculate yearly average
      J = EOM(12, YEAR_TYPE)
      VYC = VDC(J)
      VY = (VYC - VIC) / J
C
      ELSE
        write(*,*)
$      ' TEMPORAL_AGGREGATE error - Unrecognized aggregation type:',
$      AVE OR SUM
      TEMPORAL_AGGREGATE = -1
      RETURN
    ENDIF
    TEMPORAL_AGGREGATE = 0 | "Successful completion" status
    RETURN
  END

```


APPENDIX D

EXAMPLE HEDRIC-POST REPORT

Location: 258

MONTHLY AND YEARLY AVERAGES FOR 1945

	Leafy Veg. Ci/Kg	Grass: Upper Soil Ci/m2	Beef Ci/kg	Regime-1 Milk Ci/liter
January	4.1280E-11	7.2322E-08	4.9697E-15	6.0675E-11
February	2.8908E-13	4.7012E-09	4.1138E-15	4.8525E-11
March	5.0054E-14	7.0895E-11	8.1928E-14	3.1173E-11
April	8.7711E-14	9.7663E-11	3.3834E-10	2.9209E-05
May	3.1292E-15	3.0284E-12	5.1827E-12	3.6347E-06
June	5.1511E-15	4.1047E-12	7.4039E-13	9.9301E-08
July	1.1847E-12	6.6009E-10	3.7072E-11	4.4484E-07
August	8.9921E-13	4.4828E-11	2.2991E-12	9.8935E-08
September	6.9446E-14	9.3380E-13	4.1266E-14	5.2144E-08
October	5.5432E-15	2.2377E-13	9.7412E-15	6.4745E-10
November	3.3136E-15	1.0947E-14	4.1193E-15	9.9432E-11
December	9.2339E-16	6.9173E-15	2.2847E-15	6.7540E-11
Year 1945	3.6566E-12	6.4916E-09	3.1981E-11	2.7950E-06

D.1