Assessment of Effectiveness of
Geologic Isolation Systems

**Multicomponent Mass
Transport Model:
A Model for Simulating
Migration of Radionuclides
in Ground Water**

J. F. Washburn
F. E. Kaszeta
C. S. Simmons
C. R. Cole

July 1980

Prepared for the
Office of Nuclear Waste Isolation
Under its Contract with the
U.S. Department of Energy

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

The views, opinions and conclusions contained in this report are those of the contractor and do not necessarily represent those of the United States Government or the United States Department of Energy.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
Under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$ _____*: Microfiche \$3.00

*Pages	NTIS Selling Price
001-025	\$4.00
026-050	\$4.50
051-075	\$5.25
076-100	\$6.00
101-125	\$6.50
126-150	\$7.25
151-175	\$8.00
176-200	\$9.00
201-225	\$9.25
226-250	\$9.50
251-275	\$10.75
276-300	\$11.00

3 3679 00053 7912

Assessment of Effectiveness of
Geologic Isolation Systems

MULTICOMPONENT MASS TRANSPORT MODEL:
A MODEL FOR SIMULATING MIGRATION
OF RADIONUCLIDES IN GROUND WATER

J. F. Washburn
F. E. Kaszeta
C. S. Simmons
C. R. Cole

July 1980

Prepared for the
Office of Nuclear Waste Isolation
under its Contract with the
U.S. Department of Energy

Pacific Northwest Laboratory
Richland, Washington 99352

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

10/10/10

FOREWORD

The Assessment of Effectiveness of Geologic Isolation Systems (AEGIS) Program is both developing and applying the methods of assessing the far-field, long-term, post-closure safety of deep geologic repositories for nuclear waste. These methods require the application of computer codes that simulate the hydrologic systems, model the transport of released radionuclides through the hydrologic systems to the biosphere and, where applicable, assess the radiological dose to humans.

One task within AEGIS is to develop the methodology to analyze the consequences of water pathway release from loss of repository containment as defined by specific release scenarios. The modeling technology involves: 1) hydrologic codes to define the ground-water flow field and indicate water flow paths and travel times, 2) transport codes to describe the movement and concentrations of the radionuclides in the flow field, and 3) dose codes to determine the resultant radiation doses to individuals and/or populations. Figure i is a schematic flow diagram for the release consequence analysis.

If the quantity of data is immense, it is compiled in a data system. The data is prepared by input subroutines for use by the hydrologic and transport codes. The hydrologic codes provide ground-water flow field directions, rates, and velocities for use in the transport codes. Results of the transport codes include graphs of radionuclide concentration in the ground water versus travel time. After considering the dilution of radionuclide concentrations in the receiving surface-water body (e.g., lake, river, bay), the transport code results are used as the input source terms for the dose codes to calculate radiation dose to individuals and populations, when dose assessments are required.

The hydrologic and transport codes developed include several levels of complexity. Code selection and use are determined by the quantity and quality of input data. There are hydrologic codes at three levels, transport codes at two levels, and several dose codes at one level. The codes and data systems are as follows:

FIGURE i. Schematic Diagram of Consequence Analysis

- HYDROLOGIC CODES:

PNL-3162 PATHS Ground-water Hydrologic Code - PATHS is a first-level (simplest), idealized, hybrid analytical/numerical code for two-dimensional, saturated ground-water flow and single component transport. The geology of the modeled region must be homogeneous.

PNL-3160 VTT (Variable Thickness Transient) Ground-water Hydrologic Code - VTT is a second-level (intermediate complexity) two-dimensional saturated ground-water flow code, using the Boussinesq approximation, finite difference approach. It has two-dimensional (quasi three-dimensional) multiaquifer capability. Heterogeneous geology is considered.

PNL-2939 FE3DGW (Finite-Element, Three-Dimensional Ground-water) Hydrologic Code - FE3DGW is a third-level (high complexity) three-dimensional, finite-element approach (Galerkin formulation) to solve for saturated ground-water flow. Heterogeneous geology is considered.

- TRANSPORT CODES:

PNL-2970 GETOUT Transport Code - GETOUT is a first-level, one-dimensional, analytical solution considering radioactive chain decay. GETOUT considers only simple release and hydrologic functions; single speciation, constant flow rate, dispersion and sorption three-member, linear decay chains.

PNL-3179 MMT (Multicomponent Mass Transport) Code - MMT is a second-level, one-dimensional numerical, discrete parcel random walk (DPRW) algorithm. MMT considers chain decay, single speciation, equilibrium sorption, time-variant leach rate and dispersion, multi-membered linear or branched decay chains.

- DOSE CODES:

PNL-3180 ARRRG and FOOD - ARRRG determines dose from drinking water and exposure to water, consumption of aquatic food, and shorelines. FOOD determines dose from terrestrial food.

PNL-3209 PABLM - is a combination of ARRRG and FOOD with additional features related to chronic releases.

BNWL-B-264 KRONIC - determines chronic external dose from air pathways.

BNWL-B-351 SUBDOSA - determines acute external dose from air pathways.

BNWL-B-389 DACRIN - determines chronic or acute inhalation dose from air pathways.

- DATA SYSTEMS:

PNL-3139 SIRS (Sorption Information Retrieval System) - SIRS is a storage and retrieval system for experimental data on sorption/desorption analyses for a wide variety of radionuclides, ground-water compositions, and rocks and minerals.

PNL-3161 CIRMIS (Comprehensive Information Retrieval and Model Input Sequence) Data System - CIRMIS is a storage and retrieval system for code input and output data, including graphical interpretation and display.

This report is the MMT Document, PNL-3179. Return the form on the last page of this report to remain on the distribution list of future revisions of the code.

CONTENTS

FOREWORD	iii
FIGURES	ix
ACKNOWLEDGMENT	xi
SUMMARY	xiii
DEVELOPMENT AND DESCRIPTION OF THE MULTICOMPONENT MASS TRANSPORT (MMT1D) MODEL	1
INTRODUCTION	1
MODEL-EQUATION APPROACH	2
Simplifying Assumptions	3
DIRECT-SIMULATION APPROACH	9
Particles of Mass	10
Convective Transport	10
Dispersive Transport	11
Total Particle Movement	11
Concentration Distribution	12
Source/Sink Terms	12
Boundary Conditions	13
ADVANTAGES AND DISADVANTAGES OF THE DIRECT- SIMULATION APPROACH	13
Advantages	13
Disadvantages	15
NUMERICAL IMPLEMENTATION OF THE MULTICOMPONENT MASS TRANSPORT (MMT1D) MODEL	17
Convection Component	18

Dispersive Component	19
Conversion to Intensive Values.	20
Source/Sink Contribution.	21
Radioactive Decay Losses of Contributions	21
Sorption	23
COMPARISON WITH AN ANALYTICAL SOLUTION	25
INTRODUCTION	25
DISCUSSION	25
CONCLUSIONS	34
REFERENCES	37
APPENDIX A - MMT1D USER'S MANUAL	A-1
APPENDIX B - CODE LISTINGS	B-1
APPENDIX C - ANALYTICAL SOLUTION FOR COMPARISONS	C-1

FIGURES

i	Schematic Diagram of Consequence Analysis	iv
1	Results of Comparison Between MMT1D and an Analytical Solution for ^{129}I	27
2	^{129}I Comparison with Parcel Density Doubled	27
3	^{129}I Comparison with Parcel Density Increased Four-Fold	28
4	^{129}I Comparison with Parcel Density Increased Eight-Fold	28
5	Results of Comparison Between MMT1D and an Analytical Solution for ^{248}Cm	29
6	^{248}Cm Comparison with Parcel Density Doubled	29
7	^{248}Cm Comparison with Parcel Density Increased Four-Fold	30
8	^{238}Cm Comparison with Parcel Density Increased Eight-Fold	30
9	Results of Comparison Between MMT1D and an Analytical Solution for ^{244}Pu	31
10	^{244}Pu Comparison with Parcel Density Doubled	31
11	^{244}Pu Comparison with Parcel Density Increased Four-Fold	32
12	^{244}Pu Comparison with Parcel Density Increased Eight-Fold	32

ACKNOWLEDGMENT

This research was supported by the Waste Isolation Safety Assessment Program conducted by Pacific Northwest Laboratory. This program was sponsored by the Office of Nuclear Waste Isolation, which is managed by Battelle Memorial Institute under Contract DE-AC06-76RLO-1830 with the Department of Energy. On 1 October 1979, WISAP became the Assessment of Effectiveness of Geologic Isolation Systems (AEGIS) Program and the Waste/Rock Interaction Technology (WRIT) Program. This report was issued by AEGIS.

SUMMARY

This report presents the results of the development of a one-dimensional radionuclide transport code, MMT1D (Multicomponent Mass Transport), for the AEGIS Program. Multicomponent Mass Transport is a numerical solution technique that uses the discrete-parcel-random-walk (DPRW) method to directly simulate the migration of radionuclides. MMT1D accounts for:

- convection
- dispersion
- sorption-desorption
- first-order radioactive decay
- n-membered radioactive decay chains.

Comparisons between MMT1D and an analytical solution for a similar problem show that:

- MMT1D agrees very closely with the analytical solution.
- MMT1D has no cumulative numerical dispersion like that associated with solution techniques such as finite differences and finite elements.
- for current AEGIS applications, relatively few parcels are required to produce adequate results.
- the power of MMT1D is the flexibility of the code in being able to handle complex problems for which analytical solutions cannot be obtained.

Multicomponent Mass Transport (MMT1D) codes were developed at Pacific Northwest Laboratory to predict the movement of radiocontaminants in the saturated and unsaturated sediments of the Hanford Site. All MMT models require ground-water flow patterns that have been previously generated by a hydrologic model. This report documents the computer code and operating procedures of a third generation of the MMT series: the MMT differs from previous versions by simulating the mass transport processes in systems with radionuclide decay chains. Although MMT is a one-dimensional code, the user is referred to the documentation of the theoretical and numerical procedures of the three-dimensional MMT-DPRW code for discussion of expediency, verification, and error-sensitivity analysis (Ahlstrom 1977).

1. Introduction

2. Methodology

3. Results

4. Discussion

5. Conclusion

6. References

7. Appendix

8. Acknowledgments

9. Author Biographies

10. Contact Information

11. Declaration of Interest

12. Funding Source

13. Data Availability

14. Ethics Statement

15. Supplementary Materials

16. Correspondence

17. Copyright

18. Publisher's Note

19. Publisher's Disclaimer

20. Publisher's Terms and Conditions

DEVELOPMENT AND DESCRIPTION OF THE MULTICOMPONENT MASS TRANSPORT (MMT1D) MODEL

INTRODUCTION

Mass transport modeling of a given physico-chemical region to describe the movement and concentrations of the radionuclides in the flow field has been approached in two fundamental ways. The first approach will be termed the "model-equation" method. The modeled region is described with an appropriate mathematical equation, and then the equation is solved using initial and boundary conditions that characterize the region. The second method is a "direct-simulation" approach that attempts to define numerical structures that represent specific constituents or physical structures of the region and allows these numerical representations to interact with the system creating a more direct simulation of the region.

The model-equation method usually leads to a complex set of partial-differential equations that are typically solved numerically by finite-difference or finite-element techniques and, more recently, by integrated-finite-difference techniques (Narasimhan 1976). The direct simulation approach requires an efficient bookkeeping structure to control the numerical representations so that the initial conditions, boundary conditions, and other physical or chemical constraints are satisfied.

For the Assessment of Effectiveness of Geologic Isolation Systems (AEGIS) program, the first step was to review several ground-water mass transport models and weigh their advantages and disadvantages for consequence analysis of deep geologic repositories (Raymond 1977). It was found that the results from the model-equation approach using finite-difference or finite-element techniques suffered from numerical dispersion, which could be exaggerated by the required long simulation times ($\sim 10^6$ years). Another disadvantage was a greater development time, since all codes would have required modification to include decay chains. The direct-simulation code developed by Pinder and Cooper (1970) using the method of characteristics was also investigated. Its results suffered less from numerical dispersion; however, it had some stability and mass-conservation problems.

A direct-simulation technique was developed similar to that of Pinder and Cooper, except that the particles had mass instead of concentration associated with them, and the dispersion process was handled as a random process rather than by solving finite-difference equations representing the diffusion equation. This method was adapted from a direct-simulation heat transport analog developed by Eliason and Foote (1972). The primary advantages of the developed code are the following:

- It is always mass conservative.
- There is no cumulative numerical dispersion.
- There is inherent numerical stability.
- It facilitates handling of multicomponent systems.

The following is the conceptual development of both the model-equation and direct-simulation approaches. The direct-simulation approach is actually derived from the differential equations. This will provide a basis for comparing both methods.

MODEL-EQUATION APPROACH

The equations for mass transfer are extensions of the continuity equation for a specific chemical species. Stated in non-mathematical terms for a given control volume:

Net rate of mass efflux of species A from control volume	+	Net rate of accumu- lation of A within control volume	-	Rate of chemical pro- duction of A within the control volume	= 0
--	---	---	---	--	-----

A general continuity equation developed by Gray (1975) for multiphase transport of species α is written for porous media as:

$$\begin{aligned}
 & \epsilon \frac{\partial \langle \rho_\alpha \rangle}{\partial t} + \langle \rho_\alpha \rangle \frac{\partial \epsilon}{\partial t} + \nabla \cdot (\langle V \rangle \langle \rho_\alpha \rangle) + \nabla \cdot \langle \hat{\rho}_\alpha \hat{V} \rangle \\
 & + \frac{1}{V} \int_{A_{fs}(t)} \rho_\alpha (V_\alpha - W_\alpha) \cdot n_\alpha dA = - \nabla \cdot \langle J_\alpha \rangle - \\
 & \frac{1}{V} \int_{A_{fs}(t)} J_\alpha \cdot n_\alpha dA + \epsilon \langle R_\alpha \rangle^f
 \end{aligned} \tag{1}$$

where

ρ_α = Mass concentration of α in the fluid

V = Fluid pore velocity

W = Velocity of the interface

ϵ = Porosity

J_α = Diffusive flux of α with respect to the fluid pore velocity

R_α = Rate of production of α in the fluid phase

n_α = Unit vector normal to the fluid-solid interface

$\langle \rangle$ = Indicates the average of the quantity enclosed

$\wedge$ = The local fluctuations of the quantity.

The first term in the general continuity equation is the accumulation term. The next term accounts for concentration changes due to changes in porosity, which is useful for unsaturated conditions. The third term accounts for convection of species by movement of the fluid. The following term is the divergence of the average of the product of the local fluctuations of the mass concentration and pore velocity. This represents hydrodynamic dispersion as defined by Bear (1972). The last term on the lefthand side and the first term on the right in Equation (1) relate to the effect of geometry on diffusive flux (tortuosity) and the diffusive flux, respectively. The last two terms relate to interphase mass transport and chemical production, respectively.

Simplifying Assumptions

Equation (1) represents a completely general transport equation; however, not all the terms in Equation (1) are easily quantified given present technology. Several simplifying assumptions are necessary in order to obtain a usable formulation. Other justifications for simplifying Equation (1) include the following:

- The level of complexity of a model used to simulate the actual release consequences is not justified beyond the level suitable for the complexity and quality of the input data.
- Computation time and money constraints limit the level of complexity that can be considered by a model.

The simplifying assumptions will be denoted with italics where they appear in the following text.

It is assumed that the effect of changing atmospheric pressure is negligible. This assumption should be valid for most situations that are to be considered.

A velocity field is required for input to the model. This data can be obtained from a hydrologic model or from analysis of data produced from a field study program. In either case, the flow field is calculated before computing the mass transport simulation. Inherent in this procedure is the assumption that *the flow patterns are independent of the chemical composition or temperature of the ground-water solution.* This assumption is sound if the system is nearly isothermal and all solutes are at relatively low concentrations. The concentrations of the solutes must be low enough to not significantly affect the viscosity or density of the solution. For many cases this assumption will be valid, but for instances concerning salt water intrusion or dissolution of salts from salt domes or bedded salt formations, this assumption is invalid.

The result of the above assumption is to allow decoupling the mass transport equations from the flow equations and heat transport equations, thus enabling simulation of each phenomenon separately rather than simultaneously. In near surface aquifers where diurnal temperature fluctuations may be important or in cases where heat sources (such as a nuclear waste repository) are available, this assumption may not be valid.

Resolution of the flow patterns below a certain scale in a regional aquifer is usually not feasible, if not impossible. This small scale uncertainty in the flow field results in the term called mechanical or hydrodynamic dispersion (Bear 1972).

Schwartz (1977) subdivided hydrodynamic dispersion by scale and called it microscopic, macroscopic, or megascopic dispersion. Each term represents the scale of heterogeneity in the ground-water system. At the microscopic scale, dispersion occurs because the complex network of interconnected passages that comprise the soil structure causes continuous division and re-division of the

flowing fluid. On the macro-scale local variations in lithologic units are responsible for dispersion. Regional variations in lithology account for megascopic dispersion. In any case, the variation in hydraulic conductivity accounts for mechanical dispersion.

Harleman, Melhorn, and Rumer (1963) correlated the dimensionless ratio of longitudinal dispersion to the kinematic viscosity with a grain-size Reynolds number and a permeability Reynolds number. The grain-size Reynolds number was calculated with d_{50} , the 50% grain size, and the permeability Reynolds number was calculated with the square root of the intrinsic permeability. For sand they found:

$$\frac{D_L}{\nu} = 0.90 R_{d_{50}}^{1.2}$$

and

$$\frac{D_L}{\nu} = 88 R_k^{1.2}$$

where D_L/ν is the inverse Schmidt number for longitudinal dispersion. Unfortunately, dispersion has been quantified primarily in laboratory column experiments, and a paucity of field dispersion experiments exists because of measurement difficulties.

The two larger scale dispersions are made of microscopic dispersion and have results similar to molecular diffusion. In fact, Bear (1972) concluded that after a sufficiently long period of time hydrodynamic dispersion could be modeled as molecular diffusion. *Therefore, for the MMT1D model, it was assumed that hydrodynamic dispersion processes can be included with molecular diffusion in the relative flux term. Furthermore, it is assumed that the relative mass flux can be adequately described by expressions of the form of Fick's First Law. Thus,*

$$J_{\alpha} = -\rho \bar{D}_{\alpha} \nabla W_{\alpha}$$

where

ρ = total mass density of the solution

D_α = hydrodynamic dispersion tensor

W_α = mass fraction of species α (ρ_α/ρ)

Incorporating this assumption into Equation (1) results in:

$$\epsilon \frac{\partial \langle \rho_\alpha \rangle}{\partial t} + \langle \rho_\alpha \rangle \frac{\partial \epsilon}{\partial t} + \nabla \cdot (\langle V \rangle \langle \rho_\alpha \rangle) = \nabla \cdot (\rho \bar{D}_\alpha \Delta W_\alpha) + \epsilon (R_\alpha)^f \quad (2)$$

Incorporated into Fick's First Law are the assumptions that the mixture is an ideal solution and that only binary interactions at the molecular level are significant.

The dispersivity tensor, $\bar{D}_\alpha$, is a function of both space and time. If dispersion is assumed to be isotropic in the directions longitudinal and transverse to the direction of flow, then the components of $\bar{D}_\alpha$ can be represented by:

$$\begin{aligned} D_{xx}^\alpha &= D_L^\alpha \cos^2 \theta \cos^2 \phi + D_T^\alpha (\sin^2 \phi + \sin^2 \theta \cos^2 \phi) \\ D_{xy}^\alpha &= (D_L^\alpha - D_T^\alpha) \sin \phi \cos \phi \cos^2 \theta \\ D_{xz}^\alpha &= (D_T^\alpha - D_L^\alpha) \sin \theta \cos \theta \cos \phi \\ D_{yx}^\alpha &= D_{xy}^\alpha \\ D_{yy}^\alpha &= D_L^\alpha \cos^2 \theta \sin^2 \phi + D_T^\alpha (\cos^2 \phi + \sin^2 \theta \sin^2 \phi) \\ D_{yz}^\alpha &= (D_T^\alpha - D_L^\alpha) \cos \theta \sin \theta \sin \phi \\ D_{zx}^\alpha &= D_{xz}^\alpha \\ D_{zy}^\alpha &= D_{yz}^\alpha \\ D_{zz}^\alpha &= D_L^\alpha \sin^2 \theta + D_T^\alpha \cos^2 \theta \end{aligned}$$

where

ϕ = angle between the direction of flow and the (X,Y,Z) coordinate system in the X-Y plane measured from the X axis.

θ = vertical angle between the direction of flow and the X-Y plane, measured from the X-Y plane.

D_L^α = dispersion coefficient in the direction of flow L^2/T

D_T^α = dispersion coefficient in the direction transverse to flow L^2/T

D_L^α and D_T^α are commonly (Bear 1972) assumed to be represented by:

$$\begin{aligned} D_L &= a_L |\bar{v}| \\ D_T &= a_T |\bar{v}| \end{aligned} \quad (3)$$

where a_L and a_T are longitudinal and transverse dispersivities as defined by Bear (1972), which are functions of only a given porous medium. The implication is that $\bar{D}$ is a function only of soil type, not of species α . The assumption made in Equation (3) is that the contribution of molecular diffusion is negligible or constant for all species.

The convective term can be expanded giving:

$$\nabla \cdot \rho_\alpha V = \rho_\alpha (\nabla \cdot V) + (V \cdot \nabla \rho_\alpha) \quad (4)$$

The pore velocity, V , can be expressed as q/ϕ where q is the seepage (or Darcy) velocity and ϕ is the volumetric moisture content or porosity for saturated flow. The assumption here is that *Darcy's Law holds for description of saturated ground-water flow*. This should be a valid assumption for cases where pore sizes are greater than 20 μm . However, as Boersma, Lindstrom, and Saxina (1973) point out for pore sizes less than 20 μm , there is a significant deviation from Darcy's law.

Expanding Equation (4) results in:

$$\nabla \cdot \rho_{\alpha} V = \rho_{\alpha} \frac{q}{\phi^2} + \frac{\rho_{\alpha}}{\phi} (\nabla \cdot q) + V \cdot \nabla \rho_{\alpha} \quad (5)$$

If the transporting medium is assumed incompressible, which is valid except near the boiling point of water, then:

$$\nabla \cdot q = 0 \quad \text{for saturated systems,} \quad (6)$$

and

$$\nabla \cdot q = - \frac{\partial \phi}{\partial t} \quad \text{for unsaturated systems.} \quad (7)$$

For AEGIS applications the only region that would invalidate the above assumption would be near a nuclear waste repository and then for only a relatively short time after emplacement of the waste.

Incorporating the preceding assumptions and assuming that the total mass density of the mixture is relatively constant, Equation (1) becomes:

$$\frac{\partial \rho_{\alpha}}{\partial t} + V \cdot \nabla \rho_{\alpha} = \nabla \cdot \bar{D} \nabla \rho_{\alpha} + r_{\alpha} \quad (8)$$

for saturated systems.

In ground-water systems the production term, r_{α} , is usually a complex function of several variables including soil or rock type, temperature, pressure, and concentration of all species found at any point in the system. This function is normally represented by a complex rate expression. However, for most problems of practical interest these expressions are not known; and due to very difficult experimental problems, these expressions are not likely to be obtained before a waste repository is put into operation. Fortunately, most ground water moves slowly and reaction kinetics are rapid enough to allow

the assumption that most ground-water systems are at or near chemical equilibrium at all times. Thus, at the end of each numerical time step, all species are in equilibrium with each other.

Ground water consists of many constituents. Some of the species exist at concentrations high enough to affect the chemistry of other species in solution. These are called macroions. It is assumed that the radionuclides leached from a repository or naturally occurring exist only in minute quantities compared to macroions and do not appreciably affect the chemistry of other constituents in solution. These are referred to as microions. This assumption will be important in a later section dealing with how MMT handles the production term.

The equations that describe the current computerized versions of MMT1D are:

$$\frac{\partial \rho_{\alpha}}{\partial t} + V \frac{\partial \rho_{\alpha}}{\partial x} = D \frac{\partial^2 \rho_{\alpha}}{\partial x^2} + r_{\alpha} \quad (9)$$

for the one-dimensional model, and the vertically-averaged equation

$$\frac{\partial \rho_{\alpha}}{\partial t} + \left(V - \frac{\bar{D}}{H} \nabla H \right) \cdot \nabla \rho_{\alpha} = \nabla \cdot \bar{D} \nabla \rho_{\alpha} + r_{\alpha} \quad (10)$$

for the two-dimensional version.

DIRECT-SIMULATION APPROACH

The intent of the direct-simulation approach is to create a numerical analog that directly represents the physical behavior of a system. The model must satisfy the principle of mass conservation, and it must account for all important physical constraints and driving forces present in the system. A direct simulation approach applicable to AEGIS ground-water transport is outlined in the following text. All assumptions will again be indicated by *italics*.

Particles of Mass

The first step in creating an analog of a mass transport system is to define a numerical construct that represents the chemical species of interest. It has proven useful to view material systems as being comprised of a number of discrete particles. Pinder and Cooper (1970), Bredehoeft and Pinder (1973), and Konikow and Bredehoeft (1974) chose to have these particles represent concentration. In Ahlstrom et al. (1977), these particles represent mass or radioactivity. The latter was chosen for the MMT1D models. Carried to the atomic or molecular scale, this concept has been established as a reliable description of the nature of matter.

It is not computationally feasible to consider molecular-scale particles; however, the same concept can be used to create a satisfactory analog. The following assumptions apply to MMT1D:

- Material that is in solution with ground water can be represented by an ensemble of a finite number of discrete particles of mass.
- Water that is carrying the material is assumed to be a continuum, subject only to laminar flow.
- All particles are assumed to move with the continuum and at its velocity.
- All particles are independent of one another.
- Particle motion is governed by Newtonian rather than relativistic mechanics.

Because of computational restrictions, the number of particles is usually restricted to something on the order of 10^4 to 10^5 . The fourth assumption is valid only for dilute solutions, but is true for all except highly concentrated solutions. By definition the particles occupy zero volume.

Convective Transport

Convection of the particles is controlled by the motion of the host medium. *It is assumed that the number or type of particles does not significantly alter the flow properties of the host medium.* This implies that the

momentum and mass transport equations are decoupled and that the motion of the particles is governed by the physical properties of the fluid and the geometry of the system.

The flow of water in a ground-water system is usually represented by a matrix of velocity components. This matrix must represent a mass conservative flow field in order to correctly simulate mass transport in subsurface flow systems. The flow field can be obtained by several methods, but most often it is obtained from a ground-water hydrologic model. In the AEGIS suite of codes these flow simulations can be made by PATHS, VTT, or FAST. Each particle is allowed to move for a time interval, Δt , as determined by its location in the flow field. The time step is restricted so that the maximum distance a particle can travel is no larger than one grid spacing of the grid upon which the velocity field is defined.

Dispersive Transport

This transport process, usually called diffusion or dispersion, results in a net flux relative to the ambient convective transport. A qualitative discussion of dispersion was previously presented. Here again this effect has been handled by different methods. Pinder and Cooper (1970), using the method of characteristics, obtained intermediate concentrations after convecting the particles and then solved an isotropic diffusion equation by finite differences using those concentrations. Ahlstrom et al. (1977) *assumed the particles of mass are subject to various dispersive mechanisms which cause statistically random displacements*. Justification for this approach is presented in Ahlstrom et al. (1977) in Section II.

Total Particle Movement

To summarize particle movement, a particle of mass is defined that is assumed to be subject to displacements resulting from both convective and dispersive mechanisms during a given time step. If a large number of particles are released at a point in a uniform flow field after several time steps, an ellipsoidal cloud will result with a center point moving with the average flow velocity and the major semi-axis coincident with the direction of flow.

Another significant point to note is that based upon the assumptions presented above, the motion of a particle is dependent only on the nature of the flow system and not on the type of species being transported. This suggests that each particle can be tagged with more than one mass quantity, each representing a different species. By computing the movement of one set of particles, the transport of several species can be simulated simultaneously with considerable savings in computer time. However, for the AEGIS versions, in order to minimize run time and to handle decay chains, it is more convenient to allow each species to have its own set of particles.

Concentration Distribution

At the end of any desired time step, the solution can be halted and the amount of mass residing within any defined volume can be tabulated yielding an average concentration value for the volume. The solution can then continue transporting each particle from where it was halted, stopping again to compute another concentration distribution when desired. This procedure is completely mass conservative as opposed to some earlier Lagrangian solution techniques such as the particle in cell (PIC) method (Pinder 1970), which tags each particle with a concentration rather than a mass. Averaging a set of concentrations to calculate an overall cell concentration can often lead to serious mass conservation problems.

Source/Sink Terms

Injection or withdrawal of contaminants from a system is easily simulated by adding or removing particles at appropriate locations. Other types of source/sink mechanisms, such as radioactive decay or chemical reaction, require that the mass quantities with each particle be adjusted or redistributed. This applies to AEGIS applications except when reversible sorption and decay are the only source/sink terms. It is then possible to alter the velocity field by a retardation coefficient. This will be explained later in the text. The current soil-waste reaction chemical model used in the two-dimensional MMT is explained by Routson and Serne (1972).

Boundary Conditions

The boundary conditions for the direct-simulation model can be specified quite easily. Two distinct types can be identified:

- Free-flow boundary - any particle transported out of the system across this type of boundary is assumed to have exited from the system. New particles with appropriate mass are created at inflow boundaries.
- Reflecting or no-flow boundary - any particle encountering this type of boundary is reflected back into the system.

ADVANTAGES AND DISADVANTAGES OF THE DIRECT-SIMULATION APPROACH

Advantages

The direct approach to model development has the following significant advantages over the model-equation approach:

- Always Mass Conservative - The fundamental approach of the algorithm is inherently mass conservative in contrast to the somewhat similar PIC technique.
- Inherent Stability - The response of this numerical analog is inherently stable for time step size and other model parameters such as the dispersion coefficient and the magnitude of the velocity (see Trent 1975).
- No Cumulative Numerical Dispersion - Most of the numerical smearing problems often found in other numerical schemes are eliminated (see Trent 1975). The main reason for this is the Lagrangian approach to advection computation. Many of the stability and numerical dispersion problems of Eulerian methods arise from the approximations made to the advective term. This property was one of the primary motivating factors for the development of the direct approach. The only numerical dispersion in this approach occurs when concentrations are computed. This results from calculating an average concentration for each grid cell. However, this numerical dispersion is not carried forward in time since particle positions and associated masses are not affected by this averaging process.

- Ease of Control of Solution Accuracy - The accuracy of the solution can be easily controlled by specifying the number of particles to be used in the simulation. This allows a rough preliminary debugging solution to be computed for the full length of a simulation using only a few particles at a substantial reduction in cost compared to a more accurate run using a large number particles (see Ahlstrom et al. 1977).
- Solution Stacking - If nonlinear source/sink terms are not present in a given simulation (based upon the previously stated assumption, this is the only way nonlinearity can enter the problem), the results of one solution for a given problem may be averaged with subsequent solutions of the same case to give more accurate results. For instance, suppose that after computing a simulation it was apparent that too few particles had been used, the entire simulation need not be re-computed using more particles. Instead, an additional run of the same simulation can be computed (making sure the pseudo random number generator continues where it left off) and its results averaged with the previous run. This process effectively increases the particle density in each cell and can be repeated as many times as is necessary to achieve the desired solution accuracy.
- Adaptability to Small, Economical Computer System - The fact that each particle is independent of any other makes this solution technique particularly easy to program for small computers with limited addressable memory but with fairly large mass storage (disk) resources. Only a few sets of particle data need to be in memory at any one time, with the remainder of the data residing on disk. The smaller machines, although somewhat slower than most larger systems, are usually much more cost effective and allow real-time user interaction. The independence of the particle trajectory calculations also makes this model attractive for use on computers designed with a high degree of parallelism.
- Ease to Coupling New Source/Sink Models - The capability of being able to simply re-distribute particle mass in order to respond to an externally defined reaction scheme and the simplicity of adding or removing particles to simulate injection or withdrawal both allow flexibility.

- Complicated Mathematical Structures Are Avoided - The direct, discrete-particle solution approach is basically simple. The entire scheme can be described with a few algebraic expressions. The main task in implementing this algorithm is concerned with creating an efficient bookkeeping structure to keep track of particles. This type of code is much easier for nonprofessional computer technicians to understand than are the complex numerical solution schemes necessary for solving the differential equations upon which models are usually based. This allows more rapid program development and debugging and makes the resulting code easier to maintain and modify.
- Easy to Use for Three-Dimensional Applications - Assuming the accuracy is to remain constant, computation time increases linearly with the number of vertical levels used. This is a much less rapid increase in computation time than is the case with other methods. Also, problems that often arise from very high aspect ratios (the ratio of the horizontal size of a system to its vertical extent) are nonexistent.
- Complex Boundaries Are Easily Handled - All that is necessary to account for boundaries is a knowledge of the coordinates of the boundaries and the location of each particle. When a boundary is encountered by a particle, appropriate action, as discussed above, is taken based on the boundary type.

Disadvantages

In addition to these advantages, three primary disadvantages of the discrete particles method have been noted:

- Computation Speed - The primary drawback of the particle-based method outlined above is the computational speed. For circumstances where model parameters have values that are within the stability range of a finite-difference or finite-element solution schemes, these types of algorithms are usually faster for a given degree of accuracy, except perhaps for some three-dimensional cases. This problem is most acute when high accuracy solutions are required and nonlinear terms are present. The convergence of the discrete particle scheme improves only as the square root of computation time (e.g., four times as many particles must be used to

double the accuracy). When the problem is nonlinear, the entire solution set of particles must be brought forward in time before the next time step can be computed. This eliminates many of the advantages gained from the assumed independence of each particle.

The relatively large computation times are partially offset by the reliability of the algorithm for any combination of values of model parameters and by the compatibility of this method with small economical computer systems. However, this approach very likely is not the best method to use for problems requiring a solution with a very high degree of accuracy. Fortunately, most environmental simulation problems are not of this type. For most large-scale environmental simulations, the required input data are usually not known within an accuracy of better than a few percent. Consequently, a simulation using a relatively small number of particles is usually acceptable.

- Presence of Random Noise - A certain amount of statistical random noise is always present in solutions computed with the direct simulation scheme. As more particles are added, the amplitude of the noise decreases; but because of the statistical nature of the random walk analog, the random noise is always present to some degree. The random noise portion of the solution can be reduced, quite markedly in some instances, by postprocessing the results with various smoothing or filtering methods (see the section on Error and Sensitivity Analysis, in Ahlstrom et al. 1977).
- Number of Particles - A general criterion has yet to be developed for selecting the number of particles needed to obtain acceptable solution accuracy. Although the solution stacking procedure can be helpful in this regard, much depends upon the experience of the user.

Although the discrete particle approach has been shown to have several advantages and commendable properties, it should not be viewed as the best method for all problems. The possible applications of this scheme are quite broad, but it probably should not be used when the computed solution accuracy must be within 5% of the true value. However, for environmental simulations, where predictions that might be in error by a few percent are totally acceptable, this method holds great promise.

NUMERICAL IMPLEMENTATION OF THE MULTICOMPONENT MASS TRANSPORT (MMT1D) MODEL

The numerical scheme resulting from the direct simulation approach is termed the Discrete-Parcel-Random-Walk (DPRW) method. The basic device or numerical tool employed by this procedure is a hypothetical entity referred to as a computational parcel. The term "parcel" was chosen to distinguish the numerical tool from the more general but slightly vague "particles" referred to in the previous section. The continuum of dissolved or suspended matter to be modeled is represented as consisting of a finite ensemble of these parcels. The parcels have, by definition, zero size, but each has associated with it a set of Cartesian spatial coordinates $(x_p, y_p, z_p)_k$ and a set of discrete quantities of mass $\epsilon^{k,n}$, where:

p = the parcel index ($p = 1, 2, 3 \dots N_p$) where N_p is the total number of parcels used to represent a given quantity of matter.

k = the transported species index ($k = 1, 2, 3 \dots K$) where K is the total number of constituents present in the system.

n = the time level index ($n = 1, 2, 3 \dots N_t$) where N_t is the number of time increments to be computed.

For example, the location of parcel 3 after five time steps is (x_3^5, y_3^5, z_3^5) . If the problem is concerned with five constituents, this process would be repeated for all the parcels associated with each constituent.

During a given time step a new location for each parcel is computed as determined by advective and dispersive mechanisms, and the weights associated with each parcel are adjusted to account for any source/sink processes. The sequence of these computations is as follows:

1. First, the new location as determined by convective motion is calculated.
2. This new location is then modified by a random step to simulate dispersion.
3. Parcel weights are adjusted to account for first-order source/sink mechanisms such as decay.

4. The masses of all the parcels in each cell are added to obtain the total mass of each constituent in each cell.
5. Source/sink mechanisms such as radionuclide decay chains are then taken into account.
6. In the case of decay chains, parcels of daughters are then created.

Each of these steps are more fully explained in the following subsections.

Convection Component

The MMT1D-DPRW models require as input a mass conservative velocity field. At present the one-dimensional version requires a single constant velocity as its input, while a two-dimensional version allows for spatially and time variant velocity fields. The components must be arranged on a regularly-spaced grid that remains stationary with time. One or two velocity components as required by the dimensionality of the problem are defined at each nodal point of the flow field. In the case of a spatially variant problem the velocity field is linearly interpolated at points away from the node locations.

As defined in Pinder and Cooper (1970), for two dimensions the convective characteristic equations of Equation (7) are:

$$\frac{dx}{dt} = u \text{ and } \frac{dy}{dt} = v \quad (11)$$

where

u = velocity component in the x-direction

v = velocity component in the y-direction

The convective transport contribution is therefore calculated by:

$$x_p^* = x_p^n + \Delta t^n u^n \quad (12)$$

$$y_p^* = y_p^n + \Delta t^n v^n \quad (13)$$

where

Δt = the computational time step

x_p^n = the x location of parcel, p, at the end of the nth time step

* = indicates an intermediate value of the new parcel location.

Dispersive Component

A dispersive transport component is then calculated for each parcel by assuming that the ensemble of parcels is subject to Brownian-like random motion resulting from the tortuous path that the host fluid takes through a complex medium such as soil. The equation that represents dispersion along the direction of flow is:

$$x_p' = \sqrt{24D_L\Delta t} (0.5 - [R]_0') \quad (14)$$

This equation is sufficient for one-dimensional simulations, while dispersion in the direction normal to the direction of flow is represented by:

$$y_p' = \sqrt{24D_T\Delta t} (0.5 - [R]_0') \quad (15)$$

where

x_p' = the location of parcel, p, along the X' -axis oriented in the direction of flow

y_p' = the location of parcel, p, along the y' -axis oriented normal to the direction of flow

D_L = longitudinal dispersion coefficient

D_T = transverse dispersion coefficient

$[R]$: = Random number between 0 and 1 from a uniform distribution.

These equations are derived in Ahlstrom et al. (1977).

The remaining computation is to transform the dispersive components from the (x', y') coordinate system to the (x, y) coordinate system and add the dispersive contribution to the convective contribution to obtain the location of the parcel at the end of the current time step. The new location is calculated by:

$$x_p^{n+1} = x_p^* + x_p' \cos \phi - y_p' \sin \phi \quad (16)$$

$$y_p^{n+1} = y_p^* + x_p' \sin \phi + y_p' \cos \phi \quad (17)$$

where ϕ is the angle between the (x', y') coordinate system and the (x, y) coordinate system. Equation (16) is sufficient for one-dimensional systems where $\phi = 0$.

Conversion to Intensive Values

When the advective and dispersive computations have been completed for every parcel in the system, a grid network can be superimposed upon the spatially-distributed ensemble of parcels. The nodal points of the grid are labeled with i, j indices, as appropriate to the problem, where

$i = 1, 2, 3 \dots I$ $I = \text{number of nodal points in } x\text{-direction}$

$j = 1, 2, 3 \dots J$ $J = \text{number of nodal points in } y\text{-direction}$

The nodal points form the vertices for $(I-1) \times (J-1)$ rectangles which are referred to as cells. The dimensions of cell (i, j) are Δx_i by Δy_j

where:

$$\Delta x_i = x_{i+1} - x_i \quad (18)$$

$$\Delta y_j = y_{j+1} - y_j \quad (19)$$

Parcel "p" is said to lie within cell (i, j) if:

$$x_i \leq x^{k, n+1} < x_{i+1} \quad (20)$$

$$y_j \leq y^{k, n+1} < y_{j+1} \quad (21)$$

$$z_z \leq z^{k, n+1} < z_{z+1} \quad (22)$$

The total amount of mass of each species within a cell (as defined above) is computed by summing the mass quantities associated with each parcel within that cell. If concentration is needed for calculation of a source/sink term,

the total mass within each cell is divided by the volume of water within each cell. Thus, concentration within each cell for species α is calculated by

$$C_{ij}^* = \frac{G \epsilon_{ij}}{V_{ij} \phi_{ij}} \quad (23)$$

where:

G = a conversion factor from model distance units to those desired by the user (e.g., converting from Ci/ft^3 to pCi/ml)

V_{ij} = volume of cell i,j

ϕ_{ij} = porosity or volumetric water content of cell i,j

ϵ_{ij}^* = the intermediate concentration of species α in cell i,j

This intermediate concentration is now used for source/sink calculations requiring concentration (e.g., cases where concentration differences provide the driving force).

Source/Sink Contribution

The source/sink phenomena accounted for by AEGIS versions of MMT are first order decay, radionuclide decay chains, and sorption.

Radioactive Decay Losses of Contributions

First-order decay is handled easily by adjusting each parcel's associated mass by

$$\epsilon_{\alpha, \rho}^{n+1} = \epsilon_{\alpha, \rho}^n e^{-\lambda_{\alpha} \Delta t} \quad (24)$$

where λ_{α} is the first-order decay constant of species α , and Δt is the time step. In the case of decay chains all current parcel weights are decayed by the above equation. In the case of chains, concentration is not required. The mass sums are then adjusted by analytically solving the appropriate set of simultaneous ordinary differential equations that describe the decay losses and additions for the chain of interest over the last time step. Since the current parcel weights have been first-order decayed, the amount of a daughter to be created is the difference between the first-order decayed weight sum and

the weight sum adjusted by the analytical solution describing the decay chain. The operation is described by:

$$\Delta \epsilon_{\alpha} = \epsilon_{\alpha, \rho}^{n+1} - \epsilon_{\alpha, \rho}^{*, n+1} e^{-\lambda_{\alpha} \Delta t} \quad (25)$$

where, $\Delta \epsilon_{\alpha}$ is the amount of α to be created, and "*" indicates the intermediate weight sum before adjustment by decay chain mechanism. $\Delta \epsilon_{\alpha}$ is then divided by the number of parcels the user desires to create to obtain the weight of the new parcels of α created by parents of α .

The remaining problem is to locate the created parcels in the system relative to the parents creating the parcel. This is done as a random time event relative to the center of mass within a cell of the parent nuclide. While summing the weights in a given cell, a center of mass for the parents is calculated by

$$x_{\alpha, c} = \frac{\sum_{m=1}^m x_{\alpha} \epsilon_{\alpha, m}^n}{\sum_{m=1}^m \epsilon_{\alpha, m}^n} \quad (26)$$

where x_{α} is the x-location of the parcel and $x_{\alpha, c}$ is the x-location of the center of mass in cell i,j. During the time step, the center of mass of the parcels has moved a distance Δx_p , while a postulated center of mass of daughter parcels starting from the same location would have moved a distance Δx_d . Each created parcel is assumed to be created at some random time $[R_0^1]$ during the time step, and is convected and dispersed during the remaining fraction of the time step after its creation. Thus, the location of a given parcel is calculated by

$$x_d^{n+1} = x_{p, c}^n + r \Delta x_p + \frac{v}{R_d} (1-r) \Delta t + \sqrt{24 D_L (1-r) \Delta t} (0.5 - [R_0^1]) \quad (27)$$

where

x_d^{n+1} = the x-location of the created daughter parcel after current time step

$x_{p,c}$ = the X-location of center of mass of the parent from the previous time step

$r = [R]_0^1$ which is the fraction of the current time step when the daughter parcel was created

R_d = retardation coefficient of the daughter

$[R]_0^1$ = random walk dispersion movement random number

Thus the new parcels are created and located as above. This operation is repeated for each cell containing parcels and for every time step. When a parcel decays to a zero weight (zero within the accuracy of the machine), it is eliminated from further consideration.

Sorption

When considering sorption of the radionuclides by the porous media through which the radionuclides and water are flowing, Equation (6) becomes for one-dimension and first-order decay:

$$\frac{\partial \rho_\alpha}{\partial t} + \frac{B_d}{\phi} \frac{\partial S_\alpha}{\partial t} = D \frac{\partial^2 \rho_\alpha}{\partial x^2} - V \frac{\partial \rho_\alpha}{\partial x} - \lambda_\alpha \left(\rho_\alpha + \frac{B_d}{\phi} S \right) \quad (28)$$

where

B_d = bulk density of the media

S = mass concentration of α sorbed to the porous media

λ_α = decay constant

Since the concentration of radionuclides is assumed to be small, the sorption equilibrium curve should be nearly linear and is described by

$$S_\alpha = K_d \rho_\alpha \quad (29)$$

where, K_d is the distribution coefficient. Thus, by substituting Equation (29), the last term in equation (Eq-28) becomes $\lambda_\alpha \left(1 + \frac{B_d}{\phi} S \right) \rho_\alpha$. Since sorption is assumed to occur rapidly with respect to ground-water movement, Equation (29) becomes:

$$\frac{\partial S_{\alpha}}{\partial t} = K_d \frac{\partial \rho_{\alpha}}{\partial t} \quad (30)$$

when differentiated with respect to time. It must be stressed that this is only true when sorption occurs rapidly relative to the movement of the ground water or when the chemistry of the ground water is at or near equilibrium at all times. Thus, Equation (28) becomes

$$R_{\alpha} \frac{\partial \rho_{\alpha}}{\partial t} = D \frac{\partial^2 \rho_{\alpha}}{\partial x^2} - V \frac{\partial \rho_{\alpha}}{\partial x} - R_{\alpha} \lambda_{\alpha} \rho_{\alpha} \quad (31)$$

after substituting Equation (30); where R_{α} is the retardation coefficient equal to $1 + \frac{B_d}{\phi} K_d$. For decay chain migration Equation (31) becomes

$$\begin{aligned} R_1 \frac{\partial N_1}{\partial t} &= D \frac{\partial^2 N_1}{\partial x^2} - V \frac{\partial N_1}{\partial x} - \lambda_1 R_1 N_1 \\ R_2 \frac{\partial N_2}{\partial t} &= D \frac{\partial^2 N_2}{\partial x^2} - V \frac{\partial N_2}{\partial x} - \lambda_2 R_2 N_2 + \lambda_2 R_1 N_1 \\ &\vdots \\ R_n \frac{\partial N_n}{\partial t} &= D \frac{\partial^2 N_n}{\partial x^2} - V \frac{\partial N_n}{\partial x} - \lambda_n R_n N_n + \lambda_n R_{n-1} N_{n-1} \end{aligned} \quad (32)$$

for linear chains, where N indicates activity concentration usually expressed in curies. Thus, the characteristic convection equations are

$$\frac{dx}{dt} = \frac{V}{R_{\alpha}} \quad \text{and} \quad \frac{dy}{dt} = \frac{V}{R_{\alpha}} \quad (33)$$

The retardation of nuclide migration is accounted for by dividing the velocity and dispersion coefficient by the retardation coefficient. This gives an accurate result at the output of the ground water to the biosphere.

COMPARISON WITH AN ANALYTICAL SOLUTION

INTRODUCTION

The Multicomponent Mass Transport model was compared to an analytical solution for two different cases. One case compared results for a single nuclide, ^{129}I , and the other compared results of a two-member decay chain comprised of ^{248}Cm and ^{244}Pu . The analytical solutions were derived by Lester et al. (1975). The hydrologic parameters are the same for both cases, and the distribution coefficients are the same as those found in Burkholder et al. (1979). Table 1 summarizes the input data for both cases. The analytical solutions are given in Appendix C.

Results of the comparisons appear in Figures 1 through 12. The results for ^{129}I appear in Figures 1 through 4; ^{248}Cm , in Figures 5 through 8, and ^{244}Pu , in Figures 9 through 12. Each figure for each nuclide shows the results of the comparison between the analytical solution and the MMT1D solutions obtained for different numbers of parcels. In each case the results were smoothed by a moving average filter method.

DISCUSSION

Table 1 shows the input data used for the comparison of the MMT1D solution and the analytical solution. The dispersivity used was 0.01 to obtain the dispersion coefficient of $1.0 \text{ m}^2/\text{yr}$. This dispersion coefficient was used because the computer limits the size of the Peclet number that can be used in obtaining the analytical solution. However, this is a good test of MMT1D because in the case of ^{129}I either end of the curve is very sharp and any problems with numerical dispersion in the smoothing algorithm would be pointed out.

In all cases, the solution obtained by MMT1D compared very well with the analytical solution. In general, more parcels produced a better result, as expected. However, eight times more parcels in Figures 4, 8, and 12, than in

TABLE 1. Input Data Used for Comparisons Between MMT1D and the Analytical Solution Obtained by Lester et al. (1975)

<u>Nuclide Data</u>			
<u>Nuclide</u>	<u>Inventory, Ci</u>	<u>Half-Life, yr</u>	<u>Distribution Coefficient, kd, ml/g</u>
^{129}I	1.0	1.72×10^7	0.0
^{248}Cm	0.5	4.7×10^5	333
^{244}Pu	5.36×10^{-5}	7.6×10^7	1,000
<u>Hydrologic Data</u>			
Velocity of Ground-Water Flow			100 m/yr
Length of Flow Path			10,000 m
Soil-to-Solution Ratio			5.0 g/ml
Dispersion Coefficient			$1.0 \text{ m}^2/\text{yr}$
Leach Time			1,000 yr

Figures 1, 5, and 9 did not produce a comparable increase in the accuracy of the comparison. The random noise present in the former was, however, roughly half the random noise present in the latter.

Figures 1 to 4 represent the results of using progressively more parcels for the solution. The solution presented in Figure 1 used 6,500 parcels, Figure 2 used twice that, Figure 3 used four times as many, and Figure 4 used eight times more parcels than used in Figure 1. The other two series of plots represent the same increase in numbers of parcels with Figures 5 and 9 using 3,250 and 7,500, respectively.

In the case of ^{129}I , the results get progressively better as the number of parcels increases. This is because as the number of parcels increases it is possible to resolve the curve much better. As the number of parcels increases the time increment into which the exiting parcels can be summed decreases in proportion; the statistical sample increases. The reader will

FIGURE 1. Results of Comparison Between MMT1D and an Analytical Solution for ^{129}I

FIGURE 2. ^{129}I Comparison with Parcel Density Doubled

FIGURE 3. ^{129}I Comparison with Parcel Density Increased Four-Fold

FIGURE 4. ^{129}I Comparison with Parcel Density Increased Eight-Fold

FIGURE 5. Results of Comparison Between MMT1D and an Analytical Solution for ^{248}Cm

FIGURE 6. ^{248}Cm Comparison with Parcel Density doubled

FIGURE 7. ^{248}Cm Comparison with Parcel Density Increased Four-Fold

FIGURE 8. ^{248}Cm Comparison with Parcel Density Increased Eight-Fold

FIGURE 9. Results of Comparison Between MMT1D and an Analytical Solution for ^{244}Pu

FIGURE 10. ^{244}Pu Comparison with Parcel Density Doubled

FIGURE 11. ^{244}Pu Comparison with Parcel Density Increased Four-Fold

FIGURE 12. ^{244}Pu Comparison with Parcel Density Increased Eight-Fold

notice that, for the ^{129}I curves, discrepancies occur at the corners of the curves. This is not due to the model, but to the smoothing of the raw results. The corners are so sharp that the complete change in slope occurs within one time increment used to accumulate the results. A window of one increment is larger than the time in which the curve changes slope, and, therefore, the corners are rounded. In each case for ^{129}I the analytical solution appeared to give the best fit of the raw results, while the smoothed curves was rounded at the corners.

For ^{248}Cm , MMT1D appears to produce the best solution for the cases shown in Figures 6 and 8. However, the average deviation between the two curves is least in Figure 8. In fact, the average deviation decreases from Figure 5 to 8. In all cases for ^{248}Cm , MMT1D compares very well with the analytical solution. However, there are deviations in the predicted peak values for Figures 5 and 7. These are due both to smoothing and to the statistical nature of MMT1D.

Figures 9 through 12 show a comparison of results for the daughters of ^{248}Cm and ^{244}Pu . For the most part, the curves compare very well with the analytical solution, especially at the first arrival of ^{244}Pu and for the region preceding the peak. However, in the region of the peak some discrepancies occur. Again, smoothing is partially responsible for the differences, but another problem is the number of parcels used to represent the peak. In order to use enough parcels to resolve the earlier portion of the curve, it was necessary to sacrifice parcels for the peak area. Also, another problem is the number of time increments used to represent the peak. By increasing the number of time increments that represent the peak, a better fit was obtained.

A slight but significant error may be caused by differences in end boundary conditions; this is especially noticeable in the case of ^{244}Pu . The peak discharge rate obtained by MMT1D arrives earlier than the peak obtained by the analytical solution. At present, MMT1D is programmed for the case of a finite length flow path, while the analytical solution was derived for an infinite flow path.

Gradients close to the discharges are larger for finite lengths solution than for an analytical solution, thus resulting in a higher outflow ratio. This difference would not be noticeable in the middle of the curve; however, it may be at either end. This may explain the difference in peak arrival time. It would be useful to compare the results with an analytical solution derived for a soil column of finite length. However, at the time of this writing, one was not available that also accounted for decay chains.

CONCLUSIONS

In all cases for these comparisons, the results obtained compared very well with the analytical solution. For the small amount of dispersion used, the ^{129}I cases show discrepancies at the corners of the curves, even for the larger numbers of parcels. However, the results for cases with larger numbers of parcels still compare better. For most applications, including for AEGIS, the results achieved by using the least number of parcels are completely satisfactory.

Smoothing raw results to the extent of producing a smooth, eye-pleasing curve may not be appropriate. In most cases the analytical solution appeared to be the best fit of the raw data. However, producing smooth curves is probably not realistic anyway, because seldom does mother nature produce smooth data, especially in ground-water mass transport.

The strength of the MMT1D-DPRW approach used in MMT1D is not in being able to closely reproduce analytical solutions, but in being able to handle complex problems as well as simple ones.

The subjects that can be analyzed include:

- nuclide-dependent leach rates
- multiple ionic species of the same radionuclide
- spatially and/or temporally variant distribution coefficients
- spatially and/or temporally varying ground-water flow velocities
- more complex ground-water chemistry.

The one-dimensional version of MMT1D can also be used to simulate multidimensional transport by using a network of streamtubes. The advantage of the MMT1D-DPRW approach used in MMT1D is that it has flexibility at computer costs comparable to less flexible models.

REFERENCES

- Ahlstrom, S. W., et al. 1977. Multicomponent Mass Transport Model: Theory and Numerical Implementation (Discrete-Parcel-Random-Walk Version). BNWL-2127, Pacific Northwest Laboratory, Richland, Washington.
- Bear, J. 1972. Dynamics of Fluids in Porous Media. American Elsevier Publishing Co., New York.
- Boersma, L., F. T. Lindstrom and S. K. Saxina. 1973. "Limitations of Darcy's Law in Glass Bead Porous Media." SSSAP. 37:333-336.
- Bredehoeft, J. D. and G. F. Pinder. 1973. "Mass Transport in Flowing Ground Water." Water Resour. Res. 9(1):194-210.
- Burkholder, H. C., M. O. Cloninger, D. A. Baker and G. Jansen. 1975. Incentives for Partitioning High-Level Waste. BNWL-1927, Pacific Northwest Laboratory, Richland, Washington.
- Eliason, J. R., and H. P. Foote. 1972. Long Beach Generating Station Thermal Transport Modeling Study. Prepared for the Southern California Edison Company, 2120133F, Pacific Northwest Laboratory, Richland, Washington.
- Gray, W. G. 1975. "A Derivation of the Equations of Multiphase Transport." Chem. Eng. Sci. 30:229-233.
- Harleman, D. R. F., P. F. Melhorn and R. R. Rumer. 1963. "Dispersion-Permeability Correlation in Porous Media." Journal of the Hydraulics Division, ASCE. 89(H-12):67-85. Proc. Paper 3459.
- Konikow, L. F., and J. D. Bredehoeft. 1974. "Modeling Flow and Chemical Quality Changes in an Irrigated Stream-Aquifer System." Water Resour. Res. 10(3):546-562.
- Lester, D. H., G. Jansen and H. C. Burkholder. 1975. "Migration of Radionuclide Chains Through an Adsorbing Medium." Chemical Engineering Progress Symposium Series.
- Narasimhan, T. N., and P. A. Witherspoon. 1976. "An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media." Water Resour. Res. 12(1):57-64.
- Pinder, G. R., and H. H. Cooper. 1970. "A Numerical Technique for Calculating the Transient Position of the Saltwater Front." Water Resour. Res. 6(3):875-882.

Raymond, J. R. (chairman). 1977. Workshop on Transport Modeling for Nuclear Waste Repositories. PNL-SA-7468, Pacific Northwest Laboratory, Richland, Washington.

Routson, R. C., and R. J. Serne. 1972. One-Dimensional Model of the Movement of Trace Radioactive Solute Through Soil Columns. BNWL-1718, Pacific Northwest Laboratory, Richland, Washington.

Schwartz, F. W. 1977. "Macroscopic Dispersion in Porous Media: The Controlling Factors." Water Resour. Res. August 1977, 743-752.

Trent, D. W. 1975. Mathematical Modeling of Transport Processes in Aquatic Systems. BNWL-SA-5379, Pacific Northwest Laboratory, Richland, Washington.

APPENDIX A

MMT1D USER'S MANUAL

APPENDIX A

MMT1D USER'S MANUAL

INTRODUCTION

The Multicomponent Mass Transport (MMT) codes were developed to predict the movement of radioactive contaminants in both the saturated and unsaturated sediments of the Hanford site. The MMT code requires ground-water flow field to be previously generated by a hydrologic code. Although MMT1D is a one-dimensional code, the user is referred to the documentation of the theoretical and numerical procedures of the three-dimensional MMT-DPRW code for discussion of expediency, verification, and error-sensitivity analysis (Ahlstrom 1977).

The programming language used in the listings is FLECS. FLECS is a language one level higher than FORTRAN. The code listings in the appendixes are converted to FORTRAN IV before compilation. FLECS is more easily read and understood than FORTRAN IV.

MMTSET INITIALIZING CODE

The MMTSET initializing program is designed to precede an initial MMT1D simulation. MMTSET is employed to access input data files, perform initial inventory computations, and save these initial results in requested output files for use in the MMT1D code. Because the MMT1D code saves intermediate results at requested time intervals it may be restarted without reaccessing MMTSET. This appendix describes input and output files and general flow of the MMTSET.

Input and Leach Rate Files

MMTSET requires two file-Q input data files. The first file containing the MMT1D input data references a second file containing the leach rate data.

The MMTSET input data file contains the contaminant data, hydrologic simulation control parameters data, and additional file access names. Variable names and descriptions are given in Table A.1. An example file listing is given in Figure A.1.

The leach rate file is normally generated by a hydrodynamic flow model. The leach rate data file is described in Table A.2, and an example listing is shown in Figure A.2.

Parcel Location and Result File Initialization

MMTSET initializes the two file-Q data files required by the MMT1D model. Both output files will be initialized with an identical header. The parcel location file (i.e., MMTRUN1) is designed for computational and restart purposes and will be updated periodically during execution and at the end of each time step. The result file, however, saves requested time plane results for further analysis. The file is appended at requested time intervals or as requested by the operator. The result file contains 18,000 blocks. For storage purposes, the result file can be compressed (i.e., COMPRS1) using the supportive routine GETMMT and expanded again using PUTMMT.

Program Execution Flow

Besides entering input and leach rate data, MMTSET performs initializing computations. Decay constants, retardation factors, time factors, and save frequencies are calculated. The original inventory is also initially decayed, discharge rates are determined for each contaminant, and the output variables are initialized.

Five procedures are unique to MMTSET. Four procedures are common to both MMTSET and MMT1D. Eight are used only in MMT1D. Their usage is shown in Table A.1. MMTSET program flow, including procedures and variables calculated, is summarized as follows:

1. Enter the run number identification number
2. Initialize constants (RNPARN, PAWORD, PINC, RINC, RNPARC)
3. Enter input file information (IUN, FNAME)
4. Open, read and close input file
5. Open, read and close leach rate file

TABLE A.1. Procedure Usage and Description Location

	MMTSET	MMTID	Description Location
CALCULATE-TIME-PLANE-SAVES	X		A-8a
CALCULATE-DECAY-CONSTANTS-AND-TIME-FACTORS	X		A-8a
INITIALIZE-FOR-CALCULATION	X		A-8a
ORDER-RETARDATION-FACTORS	X		A-8a
CONTROL-PARCELS	X		A-8a
DECAY-INVENTORY	X	X	A-15
DETERMINE-DISCHARGE-RATE	X	X	A-13
DO-THE-INTERPOLATION	X	X	A-13
GET-FRACTION-REMAINING	X	X	A-12
CALCULATE-TIME-STEP		X	A-8
RELEASE-PARCELS		X	A-14
MOVE-PARCELS		X	A-17
PERFORM-DECAY-CALCULATIONS-FOR-CREATION		X	A-18
CREATE-PARCELS		X	A-19
OUTPUT-TIME-PLANE		X	A-19
COMPRESS-PARCELS		X	A-19
COPY-FROM-TO		X	A-19

DATA FILE FOR MMT DOCUMENTATION FOR 2-MEMBER CHAINS

2

T

T

CM248 .5,5.7E5,333.2.35.31

PU244 5.36E=05,7.6E7,999.9,35.31

100.,1.,10.,10000.,1.,1.,0.3

1.5E6,1.5E6,0.,0.,100,8196,16,16

256,256,

50,10

1.,0.001,0.01

CHNANLCM1

IODLCH.DAT

FIGURE A.1. Example MMTSET Input Data File

TABLE A.2. Input File Description

LUN = 2 Read Statement	Name	Description	Explanation	Column	Format
1		Title		1-8	20ALI
2	NCON	Number of contaminants		1-5	I5
3	DECHAN	Decay chain? (T or F)		1	L1
4	LINEAR	Linear chain (T or F)	Entered only if DECHAN = 'T'	1	L1
5	NAME(I,J)	Contaminant description		1-9	9A1
	AINV(I)	Initial inventory in curies	Contaminant data. Entered for each contaminant (I = 1, NCON)	10-19	F10.0
	HL(I)	Half life in days		20-29	F10.0
	AKD(I)	Kd value in inverse in Beta units		30-39	F10.0
	CONFAC(I)	Concentration C.F.		40-49	F10.0
5a	FACTOR(I,J)	Input factor file of relationships between contaminants	Entered only if DECHAN = 'T' and LINEAR = 'F'	1 NCON	<NCON> F10.0
6	VEL	Average velocity (ft/yr)	Hydrologic Data	1-10	F10.0
	DLEN	Dispersion length (ft))		11-20	F10.0
	BETA	Soil-to-solution ratio (bulk density/porosity)		21-30	F10.0
	COLEN	Column length		31-40	F10.0
	WIDTH	Column width		41-50	F10.0
	HEIGHT	Column weight		51-60	F10.0
	POROS	Porosity		61-70	F10.0
7	SIMLEN	Simulation length (years)	Simulation parameters	1-10	F10.0
	SAVERQ	Save frequency (years)		11-20	F10.0
	TRELMX	Maximum time step during release		21-30	F10.0
	DTMINM	Minimum time step during transport		31-40	F10.0
	NSETZ	Maximum number of parcels per contaminant		41-50	I10
	NPLMX	No. parcels exited from system before compression		51-60	I10
	NDGPTS	No. of daughter parcels created per time step during transport		41-50	I10
	NDGPRT	Number of daughter parcels created per time step during release		71-80	I10
8	RELCEN	Center of release position		1-10	F10.0
	RELEN	Length of release position		11-20	F10.0
	FRACMX	Maximum fraction of inventory to be released per time step		21-30	F10.0
9	PARFNM	Parcel location file name	File-Q Files	1-9	9A1
10	FNAME	Leach rate file name		1-9	9A1

TABLE A.3. Leach Rate File Description

LUN = 2 Read Statement		Name	Description	Explanation	Column	Format
1		NFRPTS	No. of points in fraction remaining versus time function		1-5	I5
		RELTIM	Time of complete leaching		6-15	F10.0
		RESTIM	Repository residence time before leaching begins		16-25	F10.0
2		TREM(I)	Array of time points in fraction remaining versus time function	Enter one record for each point remaining (I=1, NFRPTS)	1-10	F10.0
		FREM(I)	Fraction remaining		11-20	F10.0

```

21,18850.1,50.,          SPENT FUEL LEACH RATE FUNCTION
0.,1.
50.001,1.
1050.,.761
2050.,.634
3050.,.529
4050.,.440
5050.,.364
6050.,.297
7050.,.240
8050.,.191
9050.,.150
10050.,.115
11050.,.086
12050.,.063
13050.,.044
14050.,.030
15050.,.019
16050.,.011
17050.,.0056
18050.,.0024
18850.,0.

```

FIGURE A.2. Example Leach Rate Input Data File

6. Perform the CALCULATE-TIME-PLANE-SAVES (DNMTPS, SAVFRQ) procedure.
7. Define the CALCULATE-DECAY-CONSTANTS-AND-TIME-FACTORS (AK(I), DC,(I) NPARCM) procedure
8. Perform the INITIALIZE-FOR-CALCULATION
9. Calculate initial inventory decay with DECAY-INVENTORY (C, AINV)
10. Perform the ORDER-RETARDATION-FACTORS procedure
11. Perform the CONTROL-PARCELS procedure
12. Calculate fraction remaining with GET-FRACTION-REMAINING (FREL)
13. Calculate fraction of remaining inventory to be released with DETERMINE-DISCHARGE-RATE (RATE, TOTFRL)
14. Open and initialize the file-Q parcel location files (PARFNM)
15. Open and initialize the MMT1D file (MMT1D)
16. End.

The MMT1D execution flow is more simply depicted by use of a flow chart, Figure A.3.

MMT1D CODE

MMT1D is a one-dimensional transport code for radionuclide decay chains. Although the model is designed to be restarted using saved intermediate time-step data, initial implementation requires the use of the MMTSET initializing program previously discussed. This section is an explanation of the MMT1D code input and output.

MMT1D Operating Instructions

Prior to executing MMT1D, the operator must sign on and mount the required File-Q disk. After execution is initiated, the operator will be requested to enter the File-Q disk unit number for access to the MMT1D file, result file, and parcel location file. After the MMT1D and result files are read, the operator will be asked if the simulation will be an initial (I) or restart (R) run. If restarted, the parcel location file, containing the data from the previous run, will be read. Execution will be automatic. An example of MMT1D code input is given in Figure A.4.

FIGURE A.3. MMT1D Flow Chart

Run Time Control Switches

To allow the operator some control while the model is executing, three of the console sense switches are monitored. The special RSX-11 Fortran Subroutine SSWTCH(I,J) determines whether a specific console switch, I, is up (J = 1) or down (J = 2).

```
SCI>> RUN MMT1DC
11:15:42
ENTER FILEQ DISK UNIT AND DRIVE NO.(DB1,DB2?)>DB1
ENTER THE MMT RUN NUMBER
98
INITIAL OR RESTART (I OR R)
I
IS PARCEL RESOLUTION CONTROL REQUIRED? 0=NO 1=YES
ANSWER YES TO ENTER DAUGHTER MULTIPLIERS
0

SCI>> RUN MMT1DC
11:12:32
ENTER FILEQ DISK UNIT AND DRIVE NO.(DB1,DB2?)>DB1
ENTER THE MMT RUN NUMBER
98
INITIAL OR RESTART (I OR R)
I
IS PARCEL RESOLUTION CONTROL REQUIRED? 0=NO 1=YES
ANSWER YES TO ENTER DAUGHTER MULTIPLIERS
1
ENTER A DAUGHTER MULTIPLIER FOR      2 DAUGHTERS
1,2
ENTER A FRACTIONAL DECAY LIMIT TO EXIT
.001
USE DAUGHTER PARCEL DENSITY CONTROL? 0=NO 1=YES
1
RESET DAUGHTER PARCEL SEPARATIONS? 0=NO 1=YES
1
ENTER PARCEL SEPARATIONS (FT) FOR      2
1.,1.
```

FIGURE A.4. Examples of Interaction Input for MMT1D

Switch 15 allows the operator to terminate execution. If switch 15 is up (JSW = 1), the model terminates at the end of the time step. Switch 11 provides additional control over parcel compression. If switch 11 is up (JSW = 1), the parcel location file will be reordered at the completion of the time step. Switch 8 allows the operator to monitor progress of the executing model. If switch 8 is up (JSWCH = 1), a status report is outputted to determine for each time step (Figure A.5).

```

DATA FILE FOR MMT DOCUMENTATION FOR 2-MEMBER CHAINS
TIME STEP= 7.000000 TIME PLANE SAVE= 1
RUN =98 RESULT FILE =CHNANLCH1 PARCEL FILE =
DPRWID SIMULATION IS 5.0666668E-03 % COMPLETE
LAST TIME STEP TOOK 40444.57
SIMULATION TIME = 76.00000 YEARS
PARCELS: GENERATED AND LOST
2048.000 2384.000
0.0000000 5.0000000
EXITED PARCELS: TOTAL AND COMPRESSED
0.0000000 0.0000000
0.0000000 0.0000000
FRACTIONAL SPREAD LIMITS:
0.00000 0.00000
0.00089 0.00564
TIME STEP IS 9.000000
CONTROLLING NUCLIDE IS 1
FRACTION RELEASED= 9.0000033E-03
TOTAL FRACTION RELEASED = 8.4999979E-02
RELEASE-PARCELS 2304.000 0.2700270 256
RELEASE-PARCELS 2640.000 9.0000004E-02 256

```

FIGURE A.5. Example of Status Report for MMT1D

PROCEDURE DESCRIPTIONS

In order for the operator to obtain a basic understanding of the MMTSET AND MMT1D codes, the program flow and use of each procedure should be understood. Refer to Figure A.5 for the important steps, decisions, and entry into each procedure of the MMT1D code. A description of each procedure executed in codes follows.

CALCULATE-TIME-PLANE- SAVES

The number of time plane saves (NUMPTS) is determined from the total simulation length (SIMLEN) and the time-plane-save frequency (SAVFRQ).

$$\text{NUMPTS} = \text{SIMLEN} / \text{SAVFRQ}$$

$$\text{IF}(\text{MOD}(\text{SIMLEN}, \text{SAVFIRZ}).\text{GT}.0) \text{ NUMPTS} = \text{NUMPTS} + 1$$

This procedure is only called from MMTSET.

CALCULATE-DECAY-CONSTANTS-AND-TIME-FACTORS

Unique to MMTSET, this procedure calculates the retardation factor (AK(I)) from the beta (BETA) and the Kd, (AKD(I)) for each contaminant.

$$\text{AK}(\text{I}) = 1.0 + \text{BETA} + \text{ADK}(\text{I})$$

It also determines the decay factors,

$$\text{DC}(\text{I}) = -\text{ALOG}(.5) / \text{HL}(\text{I})$$

INITIALIZE-FOR-CALCULATION

This procedure is unique to MMTSET. It defines and initializes the following variables:

- minimum half-life (HLMIN)
- dispersion factor (DUMDC)
- time plane-save time, number, and step (TSAV, NTSAB, TSNO)
- delta X (XLOC)
- minimum contaminant number (MNSTRT)
- random number generator (DINIT)
- minimum release distance (ALOW)
- parcel and contaminant arrays (XMXL, XMNL)
- lost parcel arrays (RNPS, RNPL)
- elimination array (ELIM)
- time to residence time (TIME)
- total fraction released (TOTFRL)
- parcel file to and from pointers (TSFROM, TSTO)

ORDER-RETARDATION FACTORS

This procedure places the retardation factors (AK(I)) in ascending order and saves this order in array IBEGIN(I). It also determines the minimum retardation factor (AKMIN).

CONTROL-PARCELS

This procedure is used to set up the number of parcels released and the number of daughters generated. It is unique to MMTSET. If the maximum release time step (TRELMX) is given as zero in the input file, it is calculated as follows:

$$\text{TRELMX} = (\text{RELTIM} - \text{RESTIM}) * \text{FRACMX}$$

The maximum time step during release (TST1) is also calculated as in the CALCULATE-TIME-STEP procedure. The total number of time steps required to empty the repository (ISTEPS), and the average release time step (AVGTST) are calculated. Their values are outputted to the terminal to aid in modification of the number of parcels released.

The maximum number of time steps required for daughter parcels to exit (NSTEPS) is also determined and used to modify the time step taken when a contaminant exits the system.

This procedure controls the terminal input and output during execution of MMTSET. A further understanding of CONTROL-PARCELS is obtained from execution of MMTSET.

CALCULATE-TIME-STEP

This procedure determines the maximum possible time step (TSTEP) as a function of velocity, save frequency, release time, and maximum fraction of remaining inventory released. It is the first procedure called.

A time step based on velocity (TST1) is determined from the minimum retardation factor (AKMIN), the maximum velocity (VELMAX), and the maximum dispersion factor (DMAX), when,

$$AKMIN = AK(I)$$

$$VELMAX = VEL/AKMIN$$

and $DMAX = DUMDC/SQRT(4*AKMIN).$

The quadratic formula is employed to solve the following parabolic equation for TST1:

$$VELMAX (TST1)^2 + DMAX (\sqrt{TST1}) - XLOC = 0$$

which restricts the maximum displacement to an adjacent cell.

If release has been reached, TST1 is compared to the minimum time step during transport (DTMINM) and the smaller value is chosen

$$TST1 = MAX(TST1, DTMINM)$$

Secondly, a time step based on the time-plane-save frequency is calculated. The time remaining until the next time-plane-save (TST2) is calculated using the time-plane-save time (TSAVE) and the simulation time (TIME):

$$TST2 = TSAVE-TIME$$

The simulation time remaining is calculated for comparison. The minimum of the two calculated times is chosen.

$$TST2 = MIN(TST2, SIMLEN-TIME)$$

If release time has been reached, the maximum possible time step (TSTEP) is the minimum of TST1 and TST2. Otherwise a third time step, the time until release time, is calculated:

$$TST3 = RELTIM-T1M$$

The maximum allowable fraction released (FRACMX) is also taken into consideration at any time before release time. The time step (TSTED) is calculated using GET-FRACTION-REMAINING, and the minimum of TST1, TST2, TST3, and the maximum time step allowed (TRELMX). Until the calculated fraction of the remaining inventory to be released (FREL) is less than the maximum allowed (FRACMX), the time step is reduced by one-half, and GET-FRACTION-REMAINING calculates a new FREL value.

CALCULATE-TIME-STEP additionally resets the time-plane-save logical variable (SAVE). A set of intermediate results is saved whenever TSTEP = TST2.

Upon returning to the main program the time variable is immediately incremented by the time step calculated:

$$\text{TIME} = \text{TIME} + \text{TSTEP}$$

GET-FRACTION-REMAINING

This procedure calculates the fraction of the remaining inventory that is to be released during the current time step (FREL). This procedure is called from both the CALCULATE-TIME-STEP and DETERMINE-DISCHARGE-RATE procedures. The fraction of inventory remaining is determined for both the beginning (TREM0 = TIME-TSTEP) and end (TREM0 = TIME) of the time step using the DO-THE-INTERPOLATION procedure for each calculation. The fraction released is the difference between these two values:

$$\text{FREL} = \text{FREM3} - \text{FREM0}.$$

DO-THE-INTERPOLATION

This procedure determines the fraction of the inventory remaining (FREM0) at any point in time (TREM0) through linear interpolation of the leach rate input data array (TREM(I),FREM(I)). It is called from within the GET-FRACTION-REMAINING procedure.

If the time used for interpolation falls outside the range of the time array, the corresponding fraction of remaining inventory value is interpolated from either the maximum or minimum time values entered, according to the specific circumstance. Otherwise, a time step selector (IFREM) and two logical variables (FREM1, FREM2) are used to select the appropriate times for interpolation from the leach rate time array.

The following interpolation is used to determine the fraction of the inventory remaining:

$$\begin{aligned} \text{FREM0} = & \text{FREM}(\text{IFREM}) + (\text{TREM0} - \text{TREM}(\text{IFREM})) * \\ & (\text{FREM}(\text{IFREM} + 1) - \text{FREM}(\text{FREM})) / \\ & (\text{TREM}(\text{IFREM} + 1) - \text{TREM}(\text{FREM})) \end{aligned}$$

DETERMINE-DISCHARGE-RATE

This procedure determines the rate of parcel discharge from the source for each radionuclide contaminant under consideration prior to release time. GET-FRACTION-REMAINING and DO-THE-INTERPOLATION procedures are called to determine the fraction of the remaining inventory to be released (FREL) for the time step considered. A discharge rate (RATE(I)) is calculated for each contaminant.

$$\text{RATE(I)} = \text{AINV(I)} * \text{FREL for } I = 1, \text{NCON}$$

where AINV(I) is the present inventory. The total fraction of inventory released is also updated for each time step.

$$\text{TOTFRL} = \text{TOTFRL} + \text{FREL}$$

The discharge rate calculations are displayed on the terminal screen when console switch 8 is in the up position.

RELEASE-PARCELS

After the discharge rate has been determined, this procedure releases the parcels from the source for times between residence and release time. The procedure is repeated for each contaminant considered ($I = 1, \text{NCON}$). The average distance moved (DSCDX) is a function of the average velocity (VEL), the time step (TSTEP), and the retardation factor (AK(I)),

$$\text{DSCDX} = \text{VEL} * \text{TSTEP} / \text{AK(I)}$$

Whenever a parcel's weight is zero, the discharge rate (RATE(I)) is also zero, and the parcels are not released.

The File-Q disk is organized into parcel sets with 1024 parcels per set. Therefore, an intricate paging algorithm is needed to locate the parcel positions in the parcel location file and read in parcels set by set. Additional checks are made to determine which segment of the parcel set is to be used in the release parcel computations. The number of parcels released (NRELS) is incremented by the number of parcels in the set read in.

The distance each particular parcel moves (XL(J)) is then calculated.
(RANDMI)

The spread over the release distance is a function of the minimum release distance (ALOW), the release length (RELEN) and a random number (RAN(F1,F2)). The distribution of the parcels over the release time is a function of the number of parcels released (HREL2), the number of parcels released per time step (NPRPTS) and the average distance moved (DSCDX).

$$XL(J) = ALOW + *RAN(I1,I2) - DSCDX + NRELZ * DSCDX/NPRPTS$$

where J is the specific parcel locator. The maximum and minimum distances moved per contaminant arrays (XMX(I), XMN(I)) are updated using the calculated distance (XL(J)). The associated parcel weight (WT(J)) is also updated,

$$XWT(J) = RATE(I)/FLOAT(NPRPTS)$$

The new parcel locations and weights are then re-entered in the parcel location file. A check is made by computing the calculated number of parcels released (NRELSO) and the total number released per time step (NPRPTS). The preceding parcel release routine is repeated while reading in additional sets of parcels until all the parcels have been released.

The remaining parcel array (RNPS(I)) is then incremented by the total number of parcels released (NPRPTS) and, if console switch 8 is up, a parcel release report is displayed at the terminal. The entire release parcel routine is repeated for each of the contaminants.

DECAY-INVENTORY

This procedure accounts for radioactive decay of the inventory for each contaminant (I = 1, NCON). Decay must be considered at times prior to release time, before parcels are moved.

For single and non-chained isotopes, inventory decay is an exponential function of the inventory on hand AINV(I), decay constant DC(I), and time-step (TSTEP).

$$AINV(I) = AINV(I) * EXP ((-DC(I) * TSTEP))$$

The process is more complex for decay chains, however. A calculation array (C(I,J), J = 1, NCON, J = 1, NCON) and the factor input file (FACTOR(I,J)) are used to account for all possible decay relationships. The

calculational array is initialized with zeros and the current inventory values are set along the diagonal. For example, C(2,2) is assigned the current inventory of contaminant two (AINV(2)). With linear decay chains the original contaminant data must decay from the first contaminant to the second and so forth. The factor input file depicts this decay process for branched chains.

Each isotope can then be decayed, as a function of the decay constant (DC(I)), into a daughter isotope

$$C(D,J) = C(I-1,J) * DC(I)/(DC(I)-DC(J))$$

where I represents the isotope and J, the daughter. The original inventory is then decreased by the amount decayed,

$$C(I,I) = C(I,I) - C(I,J)$$

The new inventory (XNEW) is the summation of the total exponential decay for each possible isotope-daughter of the calculational array.

$$DO(J = 1,K)$$

$$XNEW = XNEW + C(K,J) * EXP (-DC(J) * TSTEP)$$

where J represents the isotope and K, the daughter. The new sum then replaces the previous value,

$$AINV(K) = XNEW$$

Whenever branched decay chains are considered, the relationship between the isotopes is defined through a two-dimensional array (FACTOR(DAUGHTER, ISOTOPE)). Consider the example factor input file (FACTOR (9,9)) shown in Figure A.6.

The diagonal has negative 1 entries because isotopes do not decay into an identical species. The 1 entered in the column below each negative value indicates which daughter isotope is created. This two-dimensional array can describe the most complex decay chains possible.

MOVE-PARCELS

Along with the calculations involved with moving each parcel (including a dispersion step) this procedure also sums parcel weights, calculates center of mass for parcel creation, and accounts for weight exiting the system.

FIGURE A.6. Factor Input File

The procedure begins by initializing the logical array (ELEM(I)) used for dropping isotopes as they exit the system.

An intricate paging algorithm is used to locate the parcel sets to be moved in the File-Q parcel location file. Distances to be moved are also calculated. The maximum velocity step (DX) is a function of the average velocity (VEL), time step (TSTEP), and contaminant retardation factor AK(I)

$$DX = VEL / TSTEP / AK(I)$$

The maximum dispersion step for the contaminant also takes into account the dispersion factor (DISPC):

$$D2X = DUMDC / \text{SQRT} (AK(I))$$

The parcels are then read, disk page by disk page, and moved. For each active parcel, the current location (XD) is updated according to both the velocity step (DX), and the dispersion step.

$$XD = XD + DX + D2X * (0.5 - \text{RAN}(I1, I2))$$

where RAN(I1, I2) is a random-number generation.

Weights are calculated for the beginning of the time step. If the parcel has exited the system, it is decayed to its exit weight. Otherwise, decay occurs for the entire time step. The parcels exiting the system and the parcels lost through decay are accounted for in the low parcel array (RNPL(I)). The update data is then entered in the parcel location file.

PERFORM-DECAY-CALCULATIONS-FOR-CREATION

This procedure, executed only for decay chains, performs the decay calculations on the weight sums determined in MOVE-PARCELS and then calls CREATE-PARCELS for creation of the new daughter parcels. The decay process is identical to that previously used in DECAY-INVENTORY but is limited to the region where weight sums exist.

CREATE-PARCELS

This procedure is called from within the PERFORM-DECAY-CALCULATIONS-FOR-CREATION procedure. The first order decay weight sum (DUM1) is calculated for the weight sum (WTSUM(L,K)) of each isotope (K) and X node location (L),

$$DUM1 = WTSUM(K,L) * EXP(-DC(K) * TSTEP)$$

The new weight sum to be created is the difference between the chain decay weight sum (XNEW) and the first-order decay weight sum,

$$CONC(L,K) = XNEW - DUM1$$

The center of mass for the parcel is located by an isotope index (KK) calculated outside the procedure.

$$CENTMP(L,K) = CEN(L, KK)$$

OUTPUT-TIME-PLANE

This procedure updates the result file (RESFNM) at the selected time intervals. Weight sums per cell are calculated using an index location pointer (INDEX),

$$CONCEN(INDEX) = CONCEN(INDEX) + WT(J)$$

where J is the number of parcels read in. A concentration is then calculated for each cell (K),

$$CONCEN(K) = CONCEN(K) * CONFAC(I) / (DELD * WIDTH * HEIGHT * POROS)$$

where DELD is the calculated length of the contaminant spread and CONFAC(I) is the concentration factor for contaminant I.

The result file is then appended with the calculated data and a short list of concentrations is outputted to the line printer. Before returning to the main program, the time-plane-save time (TSAV) and number (NTSAV) are incremented.

COMPRESS-PARCELS

This procedure reorganizes the parcel location file by deleting those parcels that have exited the system. It is executed whenever the number of parcels exited is greater than the maximum number that must exit for compression (NPLMX) or whenever console switch 11 is up. Again a paging algorithm is used to read from the parcel location file and to compress the data by re-entering it without including the inactive parcels.

COPY-FROM-TO

This procedure is called at program termination to determine the file-Q address for the first page of parcels and to update the parcel location file.

MMT1D MODEL OUTPUT PROGRAM - PLTCVT

This section describes the use of the program, PLTCVT, to process the raw results obtained from MMT1D and to produce the final model output either on the line printer or with one of several graphics devices available on the PNL Water and Land Resources computing system.

As previously stated, a certain amount of random noise is present in solutions that are obtained with the DPRW modeling technique. It is, therefore, necessary to smooth and/or filter the raw model output to reduce the random noise and produce the final output.

PLTCVT uses a variable window-moving average filter algorithm to smooth MMT1D raw results. Options of the program include smoothing three different sections of a curve in varying degrees and/or smoothing the raw results in any given time span.

OPERATION OF PLTCVT

PLTCVT is initiated by:

RUN PLTCVT (ALT)

The program answers by asking whether the user wants results as rate versus time (0) or concentration versus time (1) as the following:

ENTER 0 FOR RATE VERSUS TIME3

ENTER 1 FOR CONCENTRATION VERSUS TIME

It continues by asking upon which disk unit the results are stored and which MMT1D run number the user wants to analyze as in the following:

ENTER THE FILE-Q DISK UNIT 1 = DB1, 2 = DB2

ENTER THE MMT1D RUN NUMBER

The run number can range from 1 to 98.

Next, the program reports which contaminants reside in the particular MMT1D run number file and information about each contaminant. At this point, the program also provides the user with the same information as the report on the terminal as well as a histogram of the raw results in the form of a line printer listing. These reports are necessary to the remainder of the program.

The program then continues by asking for which contaminant the user wants to analyze results, as in the following:

ENTER THE CONTAMINANT #

After determining which contaminant to analyze the program asks:

WHICH ENTRY MODE 1 = DELTA T, 2 = LOW, HIGH

Entry mode refers to how the user wants to specify the summing of the dead parcels. If the user specifies, DELTA T, then the program asks:

ENTER TIME LOW, DELTA TIME, NCELLS

In this mode, the user specifies the time increment in which to sum the weights of the exited parcels, and the time span over which the sums are created in the product of NCELLS and DELTA TIME. If the user specifies, LOW, HIGH, then the program asks:

ENTER TIME LOW, TIME HIGH, # CELLS

In this mode the user specifies the time span over which he wants to analyze results of the number of cells into which he wants accumulate the concentration. This is the mode used most often, since either method can be used to produce the same results. The number of cells is limited to 500.

At this point the user is told:

ENTER A 1 FOR LP:OUTPUT

Answering '1' produces a line printer histogram over the time span and with the number of cells determined from the previous step. The user wants the final results plotted as in the following:

The program then asks which device the user wants the final results plotted as in the following:

ENTER PLOT DEVICE - (1) DBO, (2)XY, (3) TG

"XY" refers to the Calcomp plotter, and "TG" refers to the TEKTRONIX memory scope. If the user answers "1" the program asks:

ENTER WHICH DISK UNIT 0, 1, OR 2

ENTER 3 FOR MTO:, 4 FOR MT1.

"MT" refers to one of the magtape units.

PLTCVT also provides for creation of a File-Q file that will contain information to be input to dose mdels. After the above, the program asks the user to enter the name of the dose file as in the following:

ENTER NAME OF DOSE FILE

The File-Q file name may be no longer than nine characters.

The user is then asked for a title for the plots for this session of PLTCVT:

ENTER A TITLE FOR THE GRAPHS

This title may be up to 80 characters long.

Control is now passed to subroutine PLT1T for smoothing and plotting the final output. At the beginning of the smoothing processes the program tells the user to:

ENTER FRACTIONAL ACCURACY LIMIT

ENTER ZERO FOR NO LIMIT

Most often the answer is "0". However, special circumstances arise in which the answer is nonzero. The most important case is in eliminating concentrations or rates below those that are physically not possible to measure (unrealistic examples would be like a rate of 10^{-32} curies/yr or concentrations like 10^{-32} curies/ml as reported in some nuclear-waste-management-related reports).

The program now plots the raw data on the Tektronix memory scope. After completion of the plot the program then asks:

INCLUDE SMOOTH ENDS? 0 = NO, 1 = YES

The answer to the above question depends upon the character of the plot on the memory scope. As previously stated the plot may be subdivided into, at most, three regions. Each region can have its own "window" (number of points to average together). If the plot has relatively sharp corners, with the exception of the peak, then answer "no". However, if the trend in the plot is relatively regular (i.e., no sharp changes in slope), then answer "yes". Answering "yes" to the previous question produces the following:

ENTER INTEGER WINDOW FACTOR AND REPEAT COUNT

Again, the answer depends on the character of the curve. In general, the window factor should be no larger than 10% of the number of points used to generate the raw curve (e.g., NCELLS = 300 THEN WINDOW FACTOR 30). The repeat count further determines the extent of smoothing. In some cases one or

two times is sufficient, while in others ten times may be required. Experience pays off in choosing the appropriate parameters. If the user is not satisfied with the results, then he can always try it again without restarting the program, as will be seen. Before smoothing begins the program asks the user to:

SELECT A DERIVATIVE METHOD (0) HALF (1) FULL

At this point the user decides if the smoothed results adequately fits the raw data. If not, the smoothing process is restarted.

If the user answers "NO" to "INCLUDE SMOOTH ENDS," then the programs require the user to:

ENTER (2) TIME DOMAIN DIVIDING POINTS

The two dividing points are placed near sharp changes in the curve. Experience teaches the user how close to sharp points to place the dividing points. The user can also divide to the region in two at this point by setting either dividing point at the maximum or minimum of the time span. The user is then told to:

ENTER (3) INTEGER WINDOWS AND REPEAT COUNT

A window factor is entered for each region along with the repeat count. Usually the two outside regions consist of higher frequency components than does the interior region, and thus require smaller windows than the interior region. In other words, the higher the frequency of the trend of the raw data, the smaller the smoothing window.

After the previous question is answered, the user is asked whether or not he wants to:

CORRELATE LEFT END? 0 = NO 1 = YES

For most applications the answer is "NO". This is primarily an experimental switch. (There is a situation where this is of use; however, it will not be discussed here.) Then the user will be asked to choose a derivative method to which the usual answer is "FULL". The user is then asked:

IS THE TG: GRAPH OK? (Y = 1, N = 0)

This was explained previously.

If the smoothed curve is satisfactory, then a few yes-no questions are asked. They are:

DO YOU WANT RAW DATA PLOTTED? 1 = YES, 0 = NO

WRITE THE DOSE FILE FOR (NUCLIDE) 1 = YES 0 = NO

ENTER A-1 FOR STOP, 0 TO CONTINUE, 2 FOR NEW

The first two are self-explanatory. The last question gives the user the option to terminate the program (0), to analyze another nuclide in the same result file (1), or to get a new set of nuclides to analyze (2).

The following is an example of a PLTCVT run. Figures are used to show the results plotted on the screen.

DESCRIPTION OF PLTCVT

PLTCVT takes full advantage of the interactive computing capabilities available on the PNL-Water and Land Resources computing system. The program uses the following devices:

- teletype or CRT terminal for program startup and user interaction
- TEKTRONIX memory scope for plotting raw results, for previewing plots of final results, and for user interaction for comparing the smoothed result with the raw solution
- Calcomp plotter for hardcopy output of plots
- Magtape and disk for storing plots, raw results, and final digital results
- line printer for hardcopy output of final digital results.

Figure A.7 shows a flow of control diagram for PLTCVT.

FIGURE A.7. Linkage and Flow of Control Diagram for PLTCVT.

A description of each of these routines in PLTCVT follows:

PLTCVT is the main program which controls flow to GETOUT and PLTIT, and provides initial interactive selection of various I/O options. GETCVT reads dead parcel information for the chosen result file, interactively determines the range of time over which to smooth the data, and can provide a histogram of rate or concentration versus the chosen time span. PLTIT controls smoothing, plotting and listing of the final results. This is the main user-computer interactive module. PLTIT places the user's eye in a feedback loop to determine the "best" fit of the smoothed curve to the raw results. REGRES provides a simple linear regression PLTIT, and is used for the end of the smoothed curve. GRAPH plots raw results and/or smooth results to the chosen plotting device. LISTEM provides a line printer listing of smoothing parameters, nuclide information, and smoothed results.

LET THE USER BEWARE

The MMT1D model and its support programs can be quite powerful when used properly. To the new user, obtaining satisfactory solutions may seem to be more art than science. Therefore, the user should:

- completely familiarize himself with the DPRW method (several references have been given)
- be experienced in the simulation of radionuclide migration in ground-water systems
- have a complete understanding of ground-water-transport phenomena
- have an understanding of ground-water hydrology

Even after the user has armed himself with the above, there is still no replacement for experience obtained by using the model.

APPENDIX B

CODE LISTINGS

MMT1D LISTING

```

-----
00001 C ***** MMT1D.FLX *****
00002 C ***** 7/29/79 *****
00003 C
00004 C ONE DIMENSIONAL VERSION OF THE MMT TRANSPORT CODE FOR
00005 C RADIONUCLIDE DECAY CHAINS
00006 C VERSION INCLUDING CONTROL OF PARCEL DENSITY
00007 C
00008 C LOGICAL UNITS ARE:
00009 C 1=FILE-G DISK DRIVE
00010 C 2=TERMINAL INPUT
00011 C 3=LINE PRINTER
00012 C
00013 C MMT1D TRANSPORT PROGRAM HEADER BLOCK
00014 C
00015 C PARAMETER NCNUM=9,NPARCS=1024,NSETM=50
00016 C PARAMETER MNDPC=50,NCONCN=500,MAXTPS=100
00017 C PARAMETER WRDHDR=761.,PARSIZ=18500.,AHDR0=18497.,AHDR1=18494.
00018 C PARAMETER ADRTPS=14403.,ADRCOM=17400.
00019 C NCNUM=MAX. NO OF CONTAMINANTS
00020 C NPARCS=MAX. NO. OF PARCELS CORE CAN HOLD (STORED ON ONE DISK PAGE)
00021 C MUST BE MULTIPLE OF 256
00022 C NSETM=MAXIMUM NUMBER OF NPARCS SETS OF PARCELS (NORMAL)
00023 C NSETZ=MAXIMUM NUMBER OF NPARCS SETS OF PARCELS (ACTUAL THIS RUN)
00024 C MNDPC=MAXIMUM NUMBER OF DAUGHTER PARCELS THAT CAN BE CREATED
00025 C PER TIME STEP
00026 C NCONCN=MAXIMUM RESOLUTION OF SPATIAL CONCENTRATION
00027 C MAXTPS=MAXIMUM NUMBER OF TIME PLANE SAVES
00028 C WRDHDR=NUMBER OF WORDS IN THE HEADER BLOCK
00029 C PARSIZ=BLOCKS IN THE PARCEL LOCATION FILE
00030 C AHDR0=ADDRESS FOR THE INPUT HEADER BLOCK
00031 C AHDR1=ADDRESS FOR THE WORKING HEADER BLOCK
00032 C ADRTPS=OFFSET ADDRESS FOR THE TIME PLANE SAVES
00033 C ADRCOM=OFFSET ADDRESS FOR THE COMPRESS WORK SPACE
00034 C
00035 C BYTE FNAME(28),IDRV(2),RSTR
00036 C LOGICAL FREH1,FREH2,SAVE
00037 C BYTE MMTRUN(10)
00038 C HEADER BLOCK DATA(WRDHDR WORDS)
00039 C BYTE NAME,PARFNM
00040 C LOGICAL*1 DECHAN,LINEAR,ELIM
00041 C COMMON /HDR/
00042 C 1NCON,AINV(NCNUM),AKD(NCNUM),HL(NCNUM),DISPC,BETA,RESTIM,
00043 C 2NAME(9,20),CONFAC(NCNUM),VEL,COLEN,WIDTH,HEIGHT,POROS,SIHLEN,
00044 C 3TSNO,TIME,TSTEP,TSAY,SAVFRQ,NTBAV,TRELMX,DTMINH,RELCEN,RELEN,
00045 C 4ALOW,RELTIM,TOTFRL,FRACMX,FREL,TREM(25),FREM(25),NFRPTS,
00046 C 5FACTOR(NCNUM,NCNUM),XMX(NCNUM),XMN(NCNUM),AK(NCNUM),DC(NCNUM),
00047 C 6DUMDC,MNSTRT,MNS,MNSL,I1,I2,ISFROM,ISTO,IHALF,NSETZP,RNPARM,
00048 C 7PAWORD,PINC,RINC,NOGPTS,NOGPTR,NPARCM,NPLMX,IRUN,RNPARC,
00049 C 8RNPS(NCNUM),RNPL(NCNUM),PLOST(NCNUM),NLOST(NCNUM),NSETZ,
00050 C 9TITLE(20),NONODE(NCNUM),NPAREL(NCNUM),PARFNM(11),DECHAN,
00051 C 1LINEAR,ELIM(NCNUM),IORDER(NCNUM),IBEGIN(NCNUM),
00052 C 2XSEP(NCNUM),RDECAY(NCNUM),IFIRST,PLIMIT
00053 C

```

```

00054      DIMENSION IMDAUG(NCNUM)
00055      C
00056      DIMENSION CONCEN(NCONCH),IQ(NPARCS*4),XCMP(NPARCS),WCMP(NPARCS)
00057      1,RATE(NCNUM),ISSNUM(NCNUM),SIZMIN(NCNUM)
00058      C
00059      COMMON/CREATE/ CENTMP(MNDPC,NCNUM),CONC(MNDPC,NCNUM),
00060      1      WTSUM(MNDPC,NCNUM),CEN(MNDPC,NCNUM),
00061      2      C(NCNUM,NCNUM)
00062      C
00063      COMMON /PARCEL/ XL(NPARCS),WT(NPARCS)
00064      C
00065      COMMON/COMPRS/XLCOM(NPARCS),WTCOM(NPARCS)
00066      C
00067      C
00068      EQUIVALENCE (NCELLS,XLCOM(1)),(XLOW,XLCOM(2)),(XHIGH,XLCOM(3)),
00069      1      (STIME,XLCOM(4)),(CONCEN(1),XLCOM(5)),
00070      2      (TCURY,XLCOM(505))
00071      EQUIVALENCE (IQ(1),CENTMP(1,1)),
00072      1      (XCMP(1),IQ(1)),
00073      1      (WCMP(1),IQ(NPARCS*2+1))
00074      C
00075      C DETERMINE WHICH FILE & DISK DRIVE
00076      WRITE(2,153)
00077      153  FORMAT(3,' ENTER FILE& DISK UNIT AND DRIVE NO.(DB1,DB2?)>')
00078      READ(2,154) IDR, IUN
00079      154  FORMAT(2A1,I1)
00080      CALL ASNLUN(1,IDR, IUN, ISTAT)
00081      C DETERMINE THE HMT RUN NUMBER FOR LOCATING THE PROPER FILES
00082      WRITE(2,*) 'ENTER THE HMT RUN NUMBER'
00083      READ(2,*) IRUN
00084      WHEN(IRUN.LT.10)
00085      . ENCODE(7,111,MMTRUN) IRUN
00086      111  . FORMAT('MMTRUN',I1)
00087      . MMTRUN(8)=0
00088      ...FIN
00089      ELSE
00090      . ENCODE(8,112,MMTRUN) IRUN
00091      112  . FORMAT('MMTRUN',I2)
00092      . MMTRUN(9)=0
00093      ...FIN
00094      WRITE(2,*) 'INITIAL OR RESTART (I OR R)'
00095      READ(2,1) RSTR
00096      1  FORMAT(A1)
00097      CALL DPFILE(1,MMTRUN,1.,IPR2)
00098      CALL DPR(IPR2,0.,PARFNM,10.)
00099      CALL DPFILE(1,PARFNM,PARSIZ,IPR0)
00100      CALL DPR(IPR0,AHDR0,NCON,WRDHDR)
00101      C IF RESTART CASE GET THE RESTART DATA
00102      IF(RSTR.EQ.'R')
00103      . CALL DPR(IPR0,AHDR1,NCON,WRDHDR)
00104      . WRITE(2,*) ' DO YOU WISH TO EXTEND SIMULATION TIME (0 = NO)'
00105      . READ(2,*)IXTEND
00106      . IF(IXTEND.GT.0)
00107      . . WRITE(2,*) ' ENTER NEW SIMULATION TIME'
00108      . . READ(2,*)SIMLEN
00109      . ...FIN

```

```

00110      ...FIN
00111      WRITE(2,*) 'IS PARCEL RESOLUTION CONTROL REQUIRED? 0=NO 1=YES'
00112      WRITE(2,*) 'ANSWER YES TO ENTER DAUGHTER MULTIPLIERS'
00113      READ(2,*) ICNTRL
00114      DO(I=1,NCON) IMDAUG(I)=1
00115      IF(ICNTRL.EQ.1)
00116      .   WRITE(2,*) 'ENTER A DAUGHTER MULTIPLIER FOR',NCON,'DAUGHTERS'
00117      .   READ(2,*) (IMDAUG(I),I=1,NCON)
00118      .   WRITE(2,*) 'ENTER A FRACTIONAL DECAY LIMIT TO EXIT'
00119      .   READ(2,*) FRAC
00120      .   WRITE(2,*) 'USE DAUGHTER PARCEL DENSITY CONTROL? 0=NO 1=YES'
00121      .   READ(2,*) ICNTRL
00122      .   IF(ICNTRL.EQ.1)
00123      .   .   WRITE(2,*) 'RESET DAUGHTER PARCEL SEPARATIONS? 0=NO 1=YES'
00124      .   .   READ(2,*) IRESET
00125      .   .   IF(IRESET.EQ.1)
00126      .   .   .   WRITE(2,*) 'ENTER PARCEL SEPARATIONS (FT) FOR',NCON
00127      .   .   .   READ(2,*) (XSEP(I),I=1,NCON)
00128      .   .   ...FIN
00129      .   ...FIN
00130      ...FIN
00131      CLOSE(UNIT=2)
00132      IF(FRAC.LE.0.)FRAC=0.
00133      C      CALCULATE DISPERSION LENGTH FROM DISPERSION(DIFFUSION).
00134      IF(VEL.GT.0.)DLEN=DISPC/VEL
00135      DO(I=1,NCNUM) ISSNUM(I)=1
00136      C      DECAY LIMIT SET AT 10 HALF LIVES OVER 100,000. PARCELS
00137      DO(I=1,NCON) SIZMIN(I)=AINV(I)*EXP(-.693*10)/1.E5
00138      C      BEGIN TIME LOOP
00139      C
00140      TIM=SECNDS(0.)
00141      C      ***** MAIN SIMULATION LOOP *****
00142      C
00143      C      JSWITCH IS THE CONSOLE CONTROL FOR INTERMEDIATE
00144      C      TERMINAL PRINTOUT
00145      CALL SSWTCH(8,JSWCH)
00146      C
00147      REPEAT UNTIL ( TIME .GE. SIMLEN )
00148      .   ISTOZZ=(ISTO-1)*IHALF
00149      .   ISFRMZ=(ISFROM-1)*IHALF
00150      .   WHEN(TIME,LT.RELTIM)
00151      .   .   NDAUG=NDGPTR
00152      C   .   INITIALIZES TIME STEP CONTROL
00153      .   .   IFIRST=IBEGIN(MNSTRT)
00154      .   .   I=IORDER(IFIRST)
00155      .   .   PLIMIT=RNPS(I)+RDECAY(I)-1.
00156      .   ...FIN
00157      .   ELSE
00158      .   .   NDAUG=NDGPTS
00159      .   .   I=IORDER(IFIRST)
00160      .   .   IF(RNPL(I)+RDECAY(I).GE.PLIMIT)
00161      .   .   .   IFIRST=IFIRST+1
00162      .   .   .   IF(IFIRST.GT.NCON)IFIRST=NCON
00163      .   .   .   I=IORDER(IFIRST)
00164      .   .   .   PLIMIT=RNPS(I)+RDECAY(I)-1.
00165      .   .   ...FIN

```

```

00166      . ...FIN
00167      . CALCULATE-TIME-STEP
00168      . TIME=TIME+TSTEP
00169      . IF(TIME,LE,RELTIM)
00170      . . DETERMINE-DISCHARGE-RATE
00171      . . RELEASE-PARCELS
00172      . . DECAY-INVENTORY
00173      . ...FIN
00174      . MOVE-PARCELS
00175      . IF(DECHAN,AND,MNSTRT,LT,NCON)
00176      . . PERFORM-DECAY-CALCULATIONS-FOR-CREATION
00177      . ...FIN
00178      . IF( SAVE )OUTPUT-TIME=PLANE
00179      . KDONE=0
00180      . DO(I=1,NCON)
00181      . . IF(RNPS(I)-RNPL(I),LE,0) KDONE=KDONE+1
00182      . ...FIN
00183      . FREM2=(KDONE,EQ,NCON),AND,(TIME,GE,RELTIM)
00184      . JSW IS THE CONSOLE CONTROL SIGNAL FOR COMPRESSING PARCELS
00185      . CALL SSWTCH(11,JSW)
00186      . MNS=MNSTRT
00187      . IF(MNSL,NE,MNSTRT)
00188      . . MNS=MNSL
00189      . . JSW=1
00190      . ...FIN
00191      . MNSL=MNSTRT
00192      . IF(TIME,GE,SIMLEN,OR,FREM2)
00193      . . MNS=1
00194      . . JSW=1
00195      . ...FIN
00196      . FREM1=,FALSE,
00197      . DO(I=MNS,NCON)
00198      . . RLST=RNPL(I)-PLOST(I)
00199      . . IF((RLST+.1,GE, FLOAT(NPLMX),OR,JSW,EQ,1),AND,RLST,GT,0)
00200      . . . COMPRESS-PARCELS
00201      . . . FREM1=,TRUE,
00202      . . ...FIN
00203      . ...FIN
00204      . CALL SSWTCH(8,JSW)
00205      . IF(JSW, EQ, 1)
00206      . . TMS=SECND(SIMS)
00207      . . PERCNT=TIME/SIMLEN*100.
00208      . . WRITE(2,6001) (TITLE(II),II=1,16)
00209      . . FORMAT(/1X,16A4)
00210      . . WRITE(2,*) 'TIME STEP=',TSNO,' TIME PLANE SAVE=',NTSAV
00211      . . WRITE(2,5000) IRUN,RESFNM,PARFNM
00212      . . WRITE(2,*) 'DPRWD SIMULATION IS',PERCNT,'% COMPLETE'
00213      . . WRITE(2,*) 'LAST TIME STEP TOOK ',TMS
00214      . . WRITE(2,*) 'SIMULATION TIME =',TIME,' YEARS'
00215      . . WRITE(2,*) 'PARCELS: GENERATED AND LOST'
00216      . . WRITE(2,*) (RNPS(I)+RDECAY(I),I=1,NCON)
00217      . . WRITE(2,*) (RNPL(I)+RDECAY(I),I=1,NCON)
00218      . . WRITE(2,*) 'EXITED PARCELS: TOTAL AND COMPRESSED'
00219      . . WRITE(2,*) (PLOST(I)+NLOST(I),I=1,NCON)
00220      . . WRITE(2,*) (PLOST(I),I=1,NCON)
00221      . . WRITE(2,*) 'FRACTIONAL SPREAD LIMITS:'

```

```

00222      . . WRITE(2,6002) (XMN(I)/COLEN,I=1,NCON)
00223      . . WRITE(2,6002) (XMX(I)/COLEN,I=1,NCON)
00224      6002 . . FORMAT(1X,9(1X,F7.5))
00225      . . TIMS=SECNDS(0.)
00226      . . CLOSE(UNIT=2)
00227      . . .FIN
00228      C . WRITE TO THE RESTART FILE
00229      . I=ISFROM
00230      . ISFROM=ISTO
00231      . ISTO=I
00232      . ISTOZZ=(ISTO-1)*IHALF
00233      . ISFRMZ=(ISFROM-1)*IHALF
00234      . TSNO=TSNO+1.
00235      . CALL DPW(IPRO,AHDR1,NCON,WRDHDR)
00236      . COPY-FROM-TO
00237      . IF(FREM2) GO TO 5556
00238      C . SWITCH 15 IS THE CONSOLE CONTROL FOR ABORTING EXECUTION
00239      . CALL SWITCH(15,J)
00240      . IF(J.EQ. 1)
00241      . . FREM1=.TRUE.
00242      . . MNS=1
00243      . . COPY-FROM-TO
00244      . . STOP
00245      . . .FIN
00246      . IF(NTSAV.GT.100)
00247      . . WRITE(2,*)'HMT TERMINATED FOR TOO MANY TIME PLANE SAVES',NTSAV
00248      . . FREM1=.TRUE.
00249      . . MNS=1
00250      . . COPY-FROM-TO
00251      . . STOP
00252      . . .FIN
00253      . . .FIN
00254      C
00255      C ***** END OF MAIN SIMULATION LOOP *****
00256      C
00257      C
00258      5556 TIM=SECNDS(TIM)
00259      FREM1=.TRUE.
00260      MNS=1
00261      COPY-FROM-TO
00262      CALL DPW(IPRO,AHDR1,NCON,WRDHDR)
00263      TIMM=TIM/60.
00264      WRITE(3,*)'EXECUTION TIME = ',TIM,TIMM
00265      STOP
00266      C
00267      C
-----
00268      TO COPY-FROM-TO
00269      . IF(FREM1)
00270      . . DO(I=MNS,NCON)
00271      . . . NSETS=(RNPS(I)+RNPARM+.1)/RNPARC
00272      C . . . CALCULATE ADDRESS FOR FIRST PAGE OF PARCELS FOR THIS CONTAMINANT
00273      . . . PBEG=ISFRMZ+(I-1)*NSETZP
00274      . . . PBEG1=ISTOZZ+(I-1)*NSETZP

```



```

00275      . . . DO(K=ISSNUM(I),NSETS)
00276 C      . . . . READ THIS PAGE OF PARCEL DATA
00277      . . . . CALL DPR(IPRO,PBEG,XL,PAWORD)
00278      . . . . CALL DPW(IPRO,PBEG1,XL,PAWORD)
00279      . . . . PBEG=PBEG+PINC
00280      . . . . PBEG1=PBEG1+PINC
00281      . . . . FIN
00282      . . . . FIN
00283      . . . . FIN
00284      . . . . FIN
00285 C
00286 C

```

```

-----
00287      TO CALCULATE-TIME-STEP
00288 C      . CALCULATE THE MAXIMUM TIME STEP THAT CAN BE TAKEN
00289 C      . BASED ON THE VELOCITY-OR-TIME PLANE SAVE
00290 C      . -OR-RELEASE TIME REQUIREMENT
00291 C      . -OR-MAX. FRACTION RELEASE REQUIREMENT
00292 C
00293 C*****NOTE EVENTUALLY T(1/2) MIN SHOULD ALSO CONTROL*****
00294 C*****THE TIME STEP TAKEN*****
00295 C
00296 C      . USES THE MINIMUM RETARDATION FACTOR TO CALCULATE THE MAX.
00297 C      . TIME STEP BASED ON VELOCITY OF DAUGHTERS REMAINING AFTER
00298 C      . RELEASED INVENTORY OF PARENTS HAS EXITED.
00299 C
00300 C      . I IS THE ORDER INDEX
00301      . NODES=NONODE(I)
00302      . XLOC=COLEN/NODES
00303      . AKMIN=AK(I)
00304      . VELMAX=VEL/AKMIN
00305      . DMAX=OUMDC/SQRT(4.*AKMIN)
00306 C
00307 C      . CALCULATE THE MAXIMUM TIME STEP BASED ON VELOCITY
00308 C      . AND DISPERSION COEFFICIENT
00309      . WHEN(VEL.GT.0.)
00310      . . TST1=((-DMAX+SQRT(DMAX**2+4.*VELMAX*XLOC))/VELMAX/2.)*2
00311      . . .FIN
00312      . ELSE TST1=(XLOC/DMAX)**2
00313 C
00314 C      . CALCULATE THE MAX. TIME STEP BASED ON TIME PLANE SAVE FREQUENCY
00315      . TST2=TSAP-TIME
00316      . TST2=MIN(TST2,SIMLEN-TIME)
00317 C
00318 C      . WHEN TIME IS .LT. RELEASE TIME CONSIDER MAX. TIME STEP BASED ON
00319 C      . RELEASE TIME
00320      . WHEN (TIME.LT.RELTIM)
00321      . . TST3=RELTIM-TIME
00322      . . TSTEP=MIN(TST1,TST2,TST3,TRELMX)
00323      . . TIME=TIME+TSTEP
00324      . . GET-FRACTION-REMAINING
00325      . . UNTIL(FREL.LE.FRACHX)
00326      . . TSTEP=TSTEP*.9
00327

```

```

00328      . . . TIME=TIMELL+TSTEP
00329      . . . GET=FRACTION-REMAINING
00330      . . . FIN
00331      . . . TIME=TIMELL
00332      . . . FIN
00333      . ELSE
00334      . . IF(TST1,LT,DTMINM) TST1=DTMINM
00335      . . TSTEP=MIN(TST1,TST2)
00336      . . . FIN
00337      . IF(JSWTCN.EQ. 1)
00338      . . WRITE(2,*) 'TIME STEP IS ',TSTEP
00339      . . WRITE(2,*) 'CONTROLLING NUCLIDE IS',I
00340      . . CLOSE(UNIT=2)
00341      . . . FIN
00342 C      . SET THE TIME PLANE SAVE LOGICAL AS NEEDED
00343      . SAVE = .FALSE.
00344      . IF(TSTEP.EQ. TST2)SAVE = .TRUE.
00345      . . . FIN
00346 C

```

```

-----
00347      TO COMPRESS-PARCELS
00348 D      . WRITE(3,*) 'COMPRESSION STATISTICS FOR CONTAMINANT I=',I
00349 D      . WRITE(3,*) 'COMPRESSION STATISTICS INITIAL'
00350 D      . WRITE(3,*) RNPS(I),RNPL(I),PLOST(I),NLOST(I)
00351 C      NUMBER LOST TO DISCHARGE SINCE THE LAST COMPRESSION
00352      . RLOST=NLOST(I)
00353 C      CURRENT NUMBER OF COMPRESSED PARCELS LOST TO DISCHARGE
00354      . CCLOST=PLOST(I)
00355 C      CURRENT NUMBER OF COMPRESSED LOST + NEWLY LOST + NEW DECAYED LOST
00356      . TLOST=RNPL(I)
00357      . WHEN(RLOST,EQ,0)
00358      . . NSETS=(RNPS(I)+RNPARM+.1)/RNPARC
00359      . . NSSET=(PLOST(I)+.1)/RNPARC+1
00360      . . ISSNUM(I)=NSSET
00361      . . IPLOST=PLOST(I)+1.1-(NSSET-1)*RNPARC
00362      . . PBEG=ISTOZZ+(I-1)*NSETZP+PINC*(NSSET-1)
00363      . . PBEG1=PBEG
00364      . . K=NSSET-1
00365      . . M=1
00366      . . RNPDP=RNPS(I)
00367      . . REPEAT UNTIL(K.EQ.NSETS)
00368      . . . K=K+1
00369      . . . CALL DPR(IPRO,PBEG,XL,PAWORD)
00370      . . . PBEG=PBEG+PINC
00371      . . . REND=RNPDP-(K-1)*RNPARC
00372      . . . WHEN(REND,LE,RNPARC) IEND=REND+.1
00373      . . . ELSE IEND=NPARCS
00374      . . . JSTRT=1
00375      . . . IF(K.EQ.NSSET) JSTRT=IPLOST
00376      . . . DO(J=JSTRT,IEND)
00377      . . . . WHEN(XL(J),GT,0.OR,WT(J),GT,0)
00378      . . . . . WTCOM(M)=WT(J)
00379      . . . . . XLCOM(M)=XL(J)
00380      . . . . M=M+1

```

```

00381      . . . . . IF(M,GE,NPARCS+1)
00382      . . . . . . CALL DPW (IPRO,PBEG1,XLCOM,PAWORD)
00383      . . . . . . PBEG1=PBEG1+PINC
00384      . . . . . . M=M+1
00385      . . . . . . ...FIN
00386      . . . . . ...FIN
00387      . . . . . ELSE
00388      . . . . . . RNPS(I)=RNPS(I)-1
00389      . . . . . . RNPL(I)=RNPL(I)-1
00390      . . . . . . RDECAY(I)=RDECAY(I)+1
00391      . . . . . ...FIN
00392      . . . . . ...FIN
00393      . . . . . ...FIN
00394      . . . . . IF(M,GT, 1)
00395      . . . . . . CALL DPW (IPRO,PBEG1,XLCOM,PAWORD)
00396      . . . . . ...FIN
00397      . . . . . ...FIN
00398      . . . . . ELSE
00399 C      CURRENT TOTAL NUMBER OF PARCELS ACTIVE + LOST OF ANY KIND
00400 C      DOES NOT INCLUDE ANY PREVIOUSLY DISCARDED PARCELS DUE TO
00401 C      DECAY LOSSES
00402      . . . TPRC=RNPS(I)
00403 D      . . . WRITE(3,*) 'A POS'
00404 D      . . . WRITE(3,*) RLOST,CCLOST,ILOST,TPRC
00405 C      GET ADDRESS AND INDICES FOR SETTING UP THE AREA REQUIRED FOR
00406 C      ACCOMODATING THE NEWLY DISCHARGED PARCELS
00407      . . . NSSC=INT((CCLOST+1,)/RNPARC)
00408      . . . IF(CCLOST+1,-NSSC*RNPARC,GT,0,) NSSC=NSSC+1
00409      . . . ISCPPOS=CCLOST+1,-(NSSC-1)*RNPARC
00410      . . . NESC=INT((CCLOST+RLOST)/RNPARC)
00411      . . . IF(CCLOST+RLOST-NESC*RNPARC,GT,0) NESC=NESC+1
00412      . . . IECPOS=CCLOST+RLOST-(NESC-1)*RNPARC
00413      . . . PADRCO=ISTOZZ+(I-1)*NSETZP+PINC*(NSSC-1)
00414 D      . . . WRITE(3,*) 'POS B'
00415 D      . . . WRITE(3,*) I,ISTO,NSETM,NPARCS,RNPARC,PINC,NCNUM
00416 D      . . . WRITE(3,*) NSSC,NESC,ISCPPOS,IECPOS,PADRCO
00417 C      COPY THE PARCEL DATA THAT WILL BE WRITTEN OVER BY THE NEWLY
00418 C      DISCHARGED PARCEL DATA TO THE END OF THE PARCEL DATA FILE
00419      . . . PBEG=PADRCO
00420      . . . IESN=INT((TPRC+1,)/RNPARC)
00421      . . . IF(TPRC+1,-IESN*RNPARC,GT,0,) IESN=IESN+1
00422      . . . ISPOS=TPRC+1,-(IESN-1)*RNPARC
00423      . . . PEND=ISTOZZ+(I-1)*NSETZP+PINC*(IESN-1)
00424      . . . CALL DPR(IPRO,PEND,XL,PAWORD)
00425      . . . NTOT=0
00426 D      . . . WRITE(3,*) 'POS C'
00427 D      . . . WRITE(3,*) PBEG,IESN,ISPOS,PEND,NTOT
00428      . . . DO(K=NSSC,NESC)
00429      . . . . CALL DPR(IPRO,PBEG,XLCOM,PAWORD)
00430      . . . . KS=1
00431      . . . . KE=1024
00432      . . . . PBEG=PBEG+PINC
00433      . . . . IF(K,EQ,NSSC) KS=ISCPPOS
00434      . . . . IF(K,EQ,NESC) KE=IECPOS
00435 D      . . . . WRITE(3,*) 'POS D',K,KS,KE
00436      . . . . DO(K1=KS,KE)

```

```

00437      . . . . XL(ISPOS)=XLCOM(K1)
00438      . . . . WT(ISPOS)=WTCOM(K1)
00439      . . . . ISPOS=ISPOS+1
00440      . . . . NTOT=NTOT+1
00441      . . . . IF(ISPOS.GT.1024)
00442      . . . . . ISPOS=1
00443      . . . . . CALL DPW(IPRO,PEND,XL,PAWORD)
00444      . . . . . PEND=PEND+PINC
00445      . . . . . DO(KK=1,1024)
00446      . . . . . . XL(KK)=0.
00447      . . . . . . WT(KK)=0.
00448      . . . . . . . . FIN
00449      . . . . . . . . FIN
00450      . . . . . . . . FIN
00451      . . . . . . . . FIN
00452      D . . . . WRITE(3,*) 'COMPRESS CONTAMINANT I',I
00453      D . . . . WRITE(3,*) 'NTOT,RLOST',NTOT,RLOST
00454      . . . . IF(ISPOS.NE.1) CALL DPW(IPRO,PEND,XL,PAWORD)
00455      . . . . TPRCA=TPRC+RLOST
00456      . . . . PBEGD=ADRCOM
00457      . . . . ISSTRT=INT((CCLOST+RLOST+1.)/RNPARC)
00458      . . . . IF(CCLOST+RLOST+1.-ISSTRT*RNPARC.GT.0.) ISSTRT=ISSTRT+1
00459      . . . . ISPOSP=CCLOST+RLOST+1.-(ISSTRT-1)*RNPARC
00460      . . . . ISEND=INT(TPRCA/RNPARC)
00461      . . . . IF(TPRCA=ISEND*RNPARC.GT.0.) ISEND=ISEND+1
00462      . . . . IEPOSP=TPRCA-(ISEND-1)*RNPARC
00463      D . . . . RDECY=0.
00464      D . . . . RACT=0.
00465      D . . . . RLST=0.
00466      . . . . PBEG=ISTOZZ+(I-1)*NSETZP+PINC*(ISSTRT-1)
00467      . . . . PBEG1=PBEG
00468      . . . . PBEG0=PBEG
00469      . . . . IM=ISPOSP
00470      . . . . CALL DPR(IPRO,PADRC0,XCMP,PAWORD)
00471      . . . . ICOMPS=ISCP0S
00472      D . . . . WRITE(3,*) 'POS E'
00473      D . . . . WRITE(3,*) TPRC,TPRCA,ISSTRT,ISEND,ISPOSP,IEPOSP
00474      D . . . . WRITE(3,*) PBEG,PBEGD,IM,ICOMPS
00475      . . . . DO(K=ISSTRT,ISEND)
00476      . . . . . CALL DPR(IPRO,PBEG,XL,PAWORD)
00477      . . . . . PBEG=PBEG+PINC
00478      . . . . . KS=1
00479      . . . . . KE=1024
00480      . . . . . IF(K.EQ.ISSTRT) KS=ISPOSP
00481      . . . . . IF(K.EQ.ISEND) KE=IEPOSP
00482      D . . . . . WRITE(3,*) 'POS F',K,PBEG,KS,KE
00483      . . . . . DO(KK=KS,KE)
00484      . . . . . . . . CONDITIONAL
00485      . . . . . . . . (WT(KK).GT.0.)
00486      . . . . . . . . . WTCOM(IM)=WT(KK)
00487      . . . . . . . . . XLCOM(IM)=XL(KK)
00488      D . . . . . . . . . RACT=RACT+1
00489      . . . . . . . . . IM=IM+1
00490      . . . . . . . . IF(IM.GE.NPARCS+1)
00491      . . . . . . . . . CALL DPW(IPRO,PBEG1,XLCOM,PAWORD)
00492      . . . . . . . . . PBEG1=PBEG1+PINC

```

```

00493      . . . . . IM=1
00494      . . . . . ...FIN
00495      . . . . . ...FIN
00496      . . . . . (XL(KK).GT.0.AND.WT(KK).LT.0.)
00497      . . . . . WCMP(ICOMPS)=WT(KK)
00498      . . . . . XCMP(ICOMPS)=XL(KK)
00499      . . . . . ICOMPS=ICOMPS+1
00500      . . . . . PLOST(I)=PLOST(I)+1
00501      . . . . . NLOST(I)=NLOST(I)-1
00502 D      . . . . . RLST=RLST+1.
00503      . . . . . IF(ICOMPS.GE.NPARCS+1)
00504      . . . . . . CALL DPW(IPRO,PBEGD,XCMP,PAWORD)
00505      . . . . . . PBEGD=PBEGD+PINC
00506      . . . . . . ICOMPS=1
00507      . . . . . ...FIN
00508      . . . . . ...FIN
00509      . . . . . (OTHERWISE)
00510      . . . . . . RNPL(I)=RNPL(I)-1
00511      . . . . . . RNPS(I)=RNPS(I)-1
00512      . . . . . . RDECAY(I)=RDECAY(I)+1
00513 D      . . . . . . RDECY=RDECY+1.
00514      . . . . . ...FIN
00515      . . . . . ...FIN
00516      . . . . . ...FIN
00517 D      . . . . . WRITE(3,*) 'POS G',RNPL(I),RNPS(I),NLOST(I),PLOST(I),
00518 D      1. . . . . ,RLST,RDECY,RACT,PBEG,PBEGD,PBEG1,IM,ICOMPS
00519      . . . . . ...FIN
00520      . . . . . WHEN(ICOMPS.GT.1)
00521      . . . . . . CALL DPW(IPRO,PBEGD,XCMP,PAWORD)
00522      . . . . . ...FIN
00523      . . . . . ELSE PBEGD=PBEGD-PINC
00524      . . . . . IF(IM.GT.1) CALL DPW(IPRO,PBEG1,XLCOM,PAWORD)
00525      . . . . . WHEN(IECPOS.EQ.1024)
00526      . . . . . . CALL DPR(IPRO,PBEGD,XL,PAWORD)
00527      . . . . . . PBEGD=PBEGD-PINC
00528      . . . . . ...FIN
00529      . . . . . ELSE
00530      . . . . . . CALL DPR(IPRO,PBEGD,XL,PAWORD)
00531      . . . . . . CALL DPR(IPRO,PBEGD,XCMP,PAWORD)
00532      . . . . . . DO(K=1,IECPOS)
00533      . . . . . . . XL(K)=XCMP(K)
00534      . . . . . . . WT(K)=WCMP(K)
00535      . . . . . ...FIN
00536      . . . . . ...FIN
00537 D      . . . . . WRITE(3,*) 'POS H',NSSC,NESC,PBEG,PBEGD,IM,ICOMPS
00538      . . . . . DO(K=NSSC,NESC)
00539      . . . . . . CALL DPW(IPRO,PBEGD,XL,PAWORD)
00540      . . . . . . IF(K.LT.NESC)
00541      . . . . . . . PBEGD=PBEGD-PINC
00542      . . . . . . . PBEGD=PBEGD-PINC
00543      . . . . . . . CALL DPR(IPRO,PBEGD,XL,PAWORD)
00544      . . . . . ...FIN
00545      . . . . . ...FIN
00546      . . . . . ...FIN
00547 D      . . . . . WRITE(3,*) 'END'
00548 D      . . . . . WRITE(3,*) 'COMPRESSION STATISTICS FOR CONTAMINANT I=',I

```

```

00549 D . WRITE(3,*) RLOST,CCLOST,TLOST,TPRC,TPRCA
00550 D . WRITE(3,*) ROECY,RLST,RACT,RNPL(I),RNPS(I),PLOST(I),NLOST(I)
00551 D . CLOSE(UNIT=3)
00552 ...FIN
00553 C

```

```

-----
00554 TO CREATE-PARCELS
00555 C . K=ISOTOPE
00556 C . L=X NODE LOCATION
00557 C . KK=ISOTOPE INDEX TO BE USED FOR DETERMINING THE CENTER OF MASS
00558 C .
00559 C . THE FIRST ORDER DECAYED WEIGHT SUM
00560 C . DUM1=WTSUM(L,K)*EXP(-DC(K)*TSTEP)
00561 C . XNEW CHAIN DECAYED WEIGHT SUM
00562 C . SO XNEW=DUM1 IS THE WEIGHT TO BE CREATED
00563 C .
00564 C . CONDITIONAL
00565 C . . (DUM1 .LT. XNEW)
00566 C . . . CONC(L,K)=XNEW-DUM1
00567 C . . ...FIN
00568 C . . (DUM1 .GE. XNEW)
00569 C . . . CONC(L,K)=0.
00570 C . . ...FIN
00571 C . ...FIN
00572 C . SET UP THE CENTER OF MASS FOR THIS PARCEL
00573 C . CENTMP(L,K)=CEN(L,KK)
00574 C ...FIN
00575 C

```

```

-----
00576 TO DETERMINE-DISCHARGE-RATE
00577 C . PROCEDURE FOR DETERMINING DISCHARGE RATES
00578 C .
00579 C . DETERMINE THE FRACTION OF THE REMAINING INVENTORY THAT IS TO BE
00580 C . RELEASED
00581 C . GET-FRACTION-REMAINING
00582 C . DO(I=1,NCON)
00583 C . . RATE(I)=AINV(I)*FREL
00584 C . ...FIN
00585 C . TOTFRL=TOTFRL+FREL
00586 C . IF(JSWCH .EQ. 1)
00587 C . . WRITE(2,*) 'FRACTION RELEASED=',FREL
00588 C . . WRITE(2,*) 'TOTAL FRACTION RELEASED =',TOTFRL
00589 C . . CLOSE(UNIT=2)
00590 C . ...FIN
00591 C ...FIN
00592 C

```

```

-----
00593 TO DECAY-INVENTORY
00594 C PROCEDURE FOR DECAYING THE INVENTORY
00595 C . WHEN ( DECHAN )

```

```

00596 C . . THIS LOOP IS ONLY FOR DECAY CHAINS
00597 . . WHEN (LINEAR)
00598 C . . . THIS LOOP IS ONLY FOR LINEAR DECAY CHAINS
00599 . . . DO(I=1,NCON)
00600 . . . . DO(J=1,NCON)
00601 . . . . . C(J,I)=0.
00602 . . . . . ...FIN
00603 . . . . . ...FIN
00604 . . . . C(1,1)=AINV(1)
00605 . . . . DO(I=2,NCON)
00606 . . . . . C(I,I)=AINV(I)
00607 . . . . . DO(J=1,I-1)
00608 . . . . . . C(I,J)=C(I-1,J)*DC(I)/(DC(I)-DC(J))
00609 . . . . . . C(I,I)=C(I,I)-C(I,J)
00610 . . . . . ...FIN
00611 . . . . . ...FIN
00612 . . . . DO(K=1,NCON)
00613 . . . . . XNEW=0.
00614 . . . . . DO(J=1,K)
00615 . . . . . . XNEW=XNEW+C(K,J)*EXP(-DC(J)*TSTEP)
00616 . . . . . ...FIN
00617 . . . . . AINV(K)=XNEW
00618 . . . . . ...FIN
00619 . . . . ...FIN
00620 C . . ===== WHEN BRANCHED CHAINS ARE NECESSARY =====
00621 . . ELSE
00622 . . . DO(I=1,NCON)
00623 . . . . DO(J=1,NCON)
00624 . . . . . C(J,I)=0.
00625 . . . . . ...FIN
00626 . . . . . ...FIN
00627 . . . . C(1,1)=AINV(1)
00628 . . . . DO(I=2,NCON)
00629 . . . . . C(I,I)=AINV(I)
00630 . . . . . DO(J=1,I-1)
00631 . . . . . . WHEN(FACTOR(I,J).EQ.0.)
00632 . . . . . . . C(I,J)=0.
00633 . . . . . . ...FIN
00634 . . . . . . ELSE
00635 . . . . . . . K=I
00636 . . . . . . . REPEAT UNTIL(FACTOR(K,J).NE.0. .OR. K.EQ.J)
00637 . . . . . . . K=K-1
00638 . . . . . . . ...FIN
00639 . . . . . . . C(I,J)=C(K,J)*DC(I)/(DC(I)-DC(J))
00640 . . . . . . . C(I,I)=C(I,I)-C(I,J)
00641 . . . . . . ...FIN
00642 . . . . . . ...FIN
00643 . . . . . . ...FIN
00644 . . . . DO(K=1,NCON)
00645 . . . . . XNEW=0.
00646 . . . . . DO(J=1,K)
00647 . . . . . . XNEW=XNEW+C(K,J)*EXP(-DC(J)*TSTEP)
00648 . . . . . ...FIN
00649 . . . . . AINV(K)=XNEW
00650 . . . . . ...FIN
00651 . . . . ...FIN

```

```

00652      . ...FIN
00653      . ELSE
00654      C      . . DECAY THE INVENTORY
00655      C      . . FOR SINGLE AND MULTIPLE NON-CHAINED ISOTOPES ONLY
00656      . . DO (I=1,NCON)
00657      C      . . . DECAY THE INVENTORY
00658      . . . AINV(I)=AINV(I)*EXP(-DC(I)*TSTEP)
00659      . . ...FIN
00660      . ...FIN
00661      ...FIN
00662      C

```

```

-----
00663      TO DO-THE-INTERPOLATION
00664      C      PROCEDURE FOR INTERPOLATING THE FRACTION REMAINING LIST
00665      C      .
00666      . IFREM=1
00667      . CONDITIONAL
00668      . . (TREM0.LE,TREM(1)) FREM0=FREM(1)
00669      . . (TREM0.GE,TREM(NFRPTS)) FREM0=FREM(NFRPTS)
00670      . . (OTHERWISE)
00671      . . . FREM1=IFREM.EQ,NFRPTS-1
00672      . . . FREM2=TREM0.GT,TREM(IFREM).AND,TREM0.LE,TREM(IFREM+1)
00673      . . . UNTIL(FREM1.OR,FREM2)
00674      . . . . IFREM=IFREM+1
00675      . . . . FREM1=IFREM.EQ,NFRPTS-1
00676      . . . . FREM2=TREM0.GT,TREM(IFREM).AND,TREM0.LE,TREM(IFREM+1)
00677      . . . ...FIN
00678      . . . FREM0=FREM(IFREM)+(TREM0-TREM(IFREM))*
00679      1. . . . (FREM(IFREM+1)-FREM(IFREM))/
00680      2. . . . (TREM(IFREM+1)-TREM(IFREM))
00681      . . ...FIN
00682      . ...FIN
00683      ...FIN
00684      C

```

```

-----
00685      TO GET-FRACTION-REMAINING
00686      C      FREM(25)=FRACTION REMAINING VALUES
00687      C      TREM(25)=TIME FOR EACH REMAINING VALUE
00688      C      NFRPTS=NUMBER OF POINT PAIRS
00689      C      TIME=T+DT
00690      C      TIME-TSTEP=T VALUE
00691      . TREM0=TIME-TSTEP
00692      . DO-THE-INTERPOLATION
00693      . FREM3=FREM0
00694      . TREM0=TIME
00695      . DO-THE-INTERPOLATION
00696      . FREL=FREM3-FREM0
00697      . IF(FREL.LT,0) FREL=0.
00698      ...FIN
00699      C
00700      C

```



```

-----
00701      TO MOVE=PARCELS
00702      C      .      PROCEDURE FOR MOVING PARCELS
00703      C      .
00704      C      .      ALSO THE
00705      C      .      PROCEDURE TO SUM PARCEL WEIGHTS FOR DECAY CALCULATIONS AND
00706      C      .      FOR OUTPUT OF CONCENTRATION VERSUS DISTANCE TIME PLANE SAVES
00707      C      .
00708      C      .      INITIALIZE THE LOGICAL ARRAY USED FOR DROPPING OUT CHAIN DECAY
00709      C      .      ISOTOPES AS THEY EXIT THE SYSTEM NEVER TO RETURN
00710      C      .      DO(I=1,NCNUM) ELIM(I) = .FALSE.
00711      C      .      NUMBER OF DAUGHTER PARCEL DECAY CELLS IS (NDAUG)
00712      C      .      RND AUG=FLOAT(NDAUG)+.1
00713      C      .      INITIALIZE OVERALL SPREAD LIMITS
00714      C      .      XMIN=XMN(MNSTRT)
00715      C      .      XMAX=XXM(MNSTRT)
00716      C      .      DO(I=1,NCON)
00717      C      .      .      INITIALIZE THE INDIVIDUAL CONTAMINANT SPREAD ARRAYS
00718      C      .      .      IF(I,GE,MNSTRT,AND,I,LT,NCON)
00719      C      .      .      .      XMAX=MAX(XMAX,XXM(I))
00720      C      .      .      .      XMIN=MIN(XMIN,XMN(I))
00721      C      .      .      .      ...FIN
00722      C      .      .      .      XXM(I)=0.
00723      C      .      .      .      XMN(I)=COLEN
00724      C      .      .      .      ...FIN
00725      C      .      .      CALCULATE DECAY CELL SIZE
00726      C      .      .      IF(DECHAN)TDMAX=(XMAX-XMIN)/NDAUG
00727      C      .      .      ZERO OUT THE WEIGHT SUMS AND OTHER ARRAYS FOR EACH CONTAMINANT
00728      C      .      .      AND DECAY CELL
00729      C      .      .      IF(DECHAN,AND,MNSTRT,LT,NCON)
00730      C      .      .      .      DO(I=1,NCON)
00731      C      .      .      .      .      DO(J=1,NDAUG)
00732      C      .      .      .      .      .      WTSUM(J,I)=0.
00733      C      .      .      .      .      .      CEN(J,I)=0.
00734      C      .      .      .      .      .      CONC(J,I)=0.
00735      C      .      .      .      .      .      CENTMP(J,I)=0.
00736      C      .      .      .      .      .      ...FIN
00737      C      .      .      .      .      .      ...FIN
00738      C      .      .      .      .      .      ...FIN
00739      C      .      .      NOW MOVE ALL THE PARCELS FOR ALL THE CONTAMINANTS
00740      C      .      .      WHICH STILL EXIST IN THE SYSTEM
00741      C      .
00742      C      .      CALCULATE THE WEIGHT SUMS FOR EACH CONTAMINANT IN EACH DECAY CELL
00743      C      .      .      DO(I=MNSTRT,NCON)
00744      C      .      .      .      CALCULATE THE DECAY FACTOR
00745      C      .      .      .      .      DECFAC=EXP(-DC(I)*TSTEP)
00746      C      .      .      .      .      CALCULATE THE NUMBER OF DISK PAGES OF PARCELS FOR THIS CONTAMINANT
00747      C      .      .      .      .      .      NSETS=(RNPS(I)+RNPARC+.1)/RNPARC
00748      C      .      .      .      .      .      NSSET=(PLOST(I)+.1)/RNPARC+1
00749      C      .      .      .      .      .      IPLOST=PLOST(I)+1.1-(NSSET-1)*RNPARC
00750      C      .      .      .      .      CALCULATE ADDRESS FOR FIRST PAGE OF PARCELS FOR THIS CONTAMINANT
00751      C      .      .      .      .      .      PBEG=ISFRMZ+(I-1)*NSETZP+(NSSET-1)*PINC
00752      C      .      .      .      .      .      PBEG1=ISTOZZ+(I-1)*NSETZP+(NSSET-1)*PINC
00753      C      .      .      .      .      GET THE RETARDATION FACTOR FOR THIS CONTAMINANT
00754      C      .      .      .      .      .      AKTMP=AK(I)

```

```

00755 C . . CALCULATE THE DISTANCE TO BE MOVED
00756 . . . A0=VEL/AKMP
00757 . . . DX=A0*ISTEP
00758 C . . CALCULATE THE MAXIMUM DISPERSION STEP FOR THIS CONTAMINANT
00759 . . . B01=DUMDC/SQRT(AKMP)
00760 . . . D2X=B01*SQRT(ISTEP)
00761 C . . SET RNPDI EQUAL TO THE TOTAL NUMBER OF ACTIVE PARCELS FOR THIS
00762 C . . CONTAMINANT THAT HAVE BEEN RELEASED
00763 . . . RNPDI=RNPDI(I)
00764 C . . RETRIEVE THE PARCELS FOR THIS CONTAMINANT DISK PAGE BY DISK PAGE
00765 C . . AND MOVE THEM
00766 . . . DO(K=NSSET,NSETS)
00767 C . . . READ THIS PAGE OF PARCEL DATA
00768 . . . CALL DPR(IPRO,PBEG,XL,PAWORD)
00769 C . . . CALCULATE THE ENDING PARCEL OFFSET INDEX ACCOUNTING FOR THE
00770 C . . . PAGES K THAT HAVE BEEN MOVED
00771 . . . REND=RNPDI-(K-1)*RNPDI
00772 . . . WHEN(REND,LE,RNPDI) IEND=REND+.1
00773 . . . ELSE IEND=RNPDI
00774 . . . JSTRT=1
00775 . . . IF(K,EQ,NSSET) JSTRT=IPLOST
00776 . . . DO(J=JSTRT,IEND)
00777 C . . . FOR THIS PAGE K OF PARCELS FOR CONTAMINANT I MOVE PARCEL J
00778 . . . IF(WT(J),GT,0.)
00779 C . . . . THIS MOVE OF PARCELS IS ONLY DONE FOR ACTIVE PARCELS
00780 . . . . XD=XL(J)
00781 . . . . XOLD=XD
00782 C . . . . XD IS THE CURRENT LOCATION SO UPDATE IT ACCORDING TO THE
00783 C . . . . VELOCITY STEP DX AND THE DISPERSION STEP DSTEP
00784 . . . . RANDM=0.5-RAN(I1,I2)
00785 . . . . DSTEP=D2X*RANDM
00786 . . . . XD=XD+DX+DSTEP
00787 C . . . . CALCULATE THE DECAY CELL INDEX
00788 C . . . . CENTER OF MASS IS CALCULATED FOR BEGINNING OF TIME STEP
00789 . . . . WTDJ=WT(J)
00790 . . . . IF(DECHAN,AND,MNSTRT,LT,NCON)
00791 . . . . . WHEN(TDMAX,EQ,0.)
00792 . . . . . INDEX=NDAUG
00793 . . . . . ...FIN
00794 . . . . . ELSE
00795 . . . . . . RINDEX=(XOLD-XMIN)/TDMAX+1.
00796 . . . . . . IF(RINDEX,LT,1.)RINDEX=1.
00797 . . . . . . IF(RINDEX,GT,RNDAUG)RINDEX=RNDAUG
00798 . . . . . . INDEX=RINDEX
00799 . . . . . ...FIN
00800 C . . . . . ACCUMULATE WEIGHT OF PARCELS BEFORE DECAY
00801 C . . . . . CENTER OF MASS RESTRICTED TO SYSTEM
00802 . . . . . WTSUM(INDEX,I)=WTSUM(INDEX,I)+WTDJ
00803 . . . . . IF(XOLD,GE,0.)CEN(INDEX,I)=CEN(INDEX,I)+WTDJ*XOLD
00804 . . . . . ...FIN
00805 C . . . . . DETERMINE IF THE PARCEL HAS EXITED THE SYSTEM
00806 . . . . . WHEN(XD,GE,COLEN)
00807 C . . . . . . PARCEL EXIT LOOP
00808 C . . . . . . MARK THE PARCEL AS INACTIVE
00809 . . . . . . CO=COLEN-XOLD
00810 . . . . . . B0=B01*RANDM

```

```

00811      . . . . . WHEN(VEL.GT.0.)
00812      . . . . . . TBNDSR=(-B0+SQRT(B0*B0+4.*A0*C0))/(2.*A0)
00813      . . . . . . TBND=TBNDSR*TBNDSR
00814      . . . . . . ...FIN
00815      . . . . . . ELSE TBND=TSTEP*C0/DSTEP
00816      . . . . . . XL(J)=TIME-TSTEP+TBND
00817      C . . . . . . DECAY PARCEL TO ITS EXIT WEIGHT
00818      . . . . . . WTDJ=WTDJ*EXP(-DC(I)*TBND)
00819      . . . . . . WT(J)=-1.*WTDJ
00820      . . . . . . IF(WT(J).GE.0.) WT(J)=-1.E-35
00821      . . . . . . NLOST(I)=NLOST(I)+1
00822      C . . . . . . INCREMENT THE PARCEL LOST COUNTER FOR THIS CONTAMINANT
00823      . . . . . . RNPL(I)=RNPL(I)+1
00824      . . . . . . ...FIN
00825      . . . . . . ELSE
00826      C . . . . . . NOW UPDATE THE PARCEL LOCATION AND DECAY WEIGHT
00827      . . . . . . XL(J)=XD
00828      . . . . . . WTDJ=WTDJ*DECFACTOR
00829      C . . . . . . UPDATE LOCATION FOR CALCULATING WEIGHT SUMS FOR PARCEL CREATION
00830      . . . . . . XMX(I)=MAX(XMX(I),XL(J))
00831      . . . . . . XMN(I)=MIN(XMN(I),XL(J))
00832      . . . . . . IF(XMN(I).LT.0.) XMN(I)=0.
00833      C . . . . . . MARK PARCELS LOST BY DECAY
00834      C . . . . . . (LOCATION TO REMOVE NEGLIGIBLE PARCELS IF REQUIRED)
00835      . . . . . . IF(WTDJ.LE.SIZMIN(I))
00836      . . . . . . . XL(J)=-1.
00837      . . . . . . . WTDJ=-1.
00838      . . . . . . . RNPL(I)=RNPL(I)+1
00839      . . . . . . ...FIN
00840      C . . . . . . SAVE WEIGHT OF PARCELS REMAINING IN SYSTEM
00841      . . . . . . WT(J)=WTDJ
00842      . . . . . . ...FIN
00843      . . . . . . ...FIN
00844      . . . . . . ...FIN
00845      . . . . . . CALL DPW(IPRO,PBEG1,XL,PAWORD)
00846      . . . . . . PBEG1=PBEG1+PINC
00847      . . . . . . PBEG=PBEG+PINC
00848      . . . . . . ...FIN
00849      C ***** CALCULATE CENTER OF MASS FOR PARCEL CREATION *****
00850      C . . . . .
00851      . . . . . IF(DECHAN.AND.MNSTRT.LT.NCON)
00852      . . . . . . DO(K=1,NDAUG)
00853      . . . . . . . IF(WTSUM(K,I).GT.0.)
00854      . . . . . . . . CEN(K,I)=CEN(K,I)/WTSUM(K,I)
00855      . . . . . . . . ELIM(I) = .TRUE.
00856      . . . . . . . ...FIN
00857      . . . . . . ...FIN
00858      . . . . . . ...FIN
00859      . . . . . . ...FIN
00860      . . . . . IF(DECHAN.AND.MNSTRT.LT.NCON)
00861      . . . . . . I=0
00862      . . . . . . REPEAT UNTIL ( ELIM(I).OR.I.EQ.NCON ) I=I+1.
00863      . . . . . . MNSTRT=I
00864      . . . . . . ...FIN
00865      . . . . . . ...FIN
00866      C

```

```

-----
00867      TO OUTPUT-TIME-PLANE
00868      C      .  PROCEDURE TO OUTPUT CONCENTRATION VS DISTANCE DATA
00869      C      .  AT THE SELECTED TIME INTERVALS
00870      C      .
00871      .  RBEG=(NTSAV-1)*36.+ADRTPS
00872      .  DO(I=1,NCON)
00873      .  .  TCURY=0.
00874      C      .  .  CALCULATE THE WEIGHT SUMS AND CONVERT TO CONCENTRATION FOR
00875      C      .  .  CONTAMINANT I
00876      .  .  RNPDI=RNPS(I)
00877      .  .  NSETS=(RNPS(I)+RNPARM+.1)/RNPARC
00878      .  .  NSSET=(PLOST(I)+.1)/RNPARC+1
00879      .  .  IPLOST=PLOST(I)+1.1-(NSSET-1)*RNPARC
00880      .  .  PBEG=ISTOZZ+(I-1)*NSETZP+PINC*(NSSET-1)
00881      .  .  DO(K=1,NCONCN) CONCEN(K)=0.
00882      .  .  NCELLS=NCONCN-2
00883      .  .  XLOW=XMN(I)
00884      .  .  XHIGH=XMX(I)
00885      .  .  STIME=TIME
00886      .  .  DELD=(XHIGH-XLOW)/NCELLS
00887      .  .  NCELLS=NCELLS+2
00888      .  .  XLOW=XLOW-DELD
00889      .  .  XHIGH=XHIGH+DELD
00890      .  .  WHEN(I.LT.MNSTRT.OR.RNPS(I).EQ.RNPL(I))
00891      .  .  .  XLOW=0.
00892      .  .  .  XHIGH=COLEN
00893      .  .  .  NCELLS=0
00894      .  .  .  CALL DPR(IPRO,RBEG,NCELLS,NCONCN*2.+10.)
00895      .  .  .  RBEG=RBEG+RINC
00896      .  .  .  ...FIN
00897      .  .  ELSE
00898      .  .  .  NCEL=0
00899      .  .  .  DO(K=NSSET,NSETS)
00900      .  .  .  .  CALL DPR(IPRO,PBEG,XL,PAWORD)
00901      .  .  .  .  PBEG=PBEG+PINC
00902      .  .  .  .  REND=RNPD-(K-1)*RNPARC
00903      .  .  .  .  WHEN(REND.LE.RNPARC) IEND=REND+.1
00904      .  .  .  .  ELSE IEND=RNPARC
00905      .  .  .  .  JSTRT=1
00906      .  .  .  .  IF(K.EQ.NSSET) JSTRT=IPLOST
00907      .  .  .  .  DO(J=JSTRT,IEND)
00908      .  .  .  .  .  IF(WT(J).GT.0.)
00909      .  .  .  .  .  .  WHEN(DELD.EQ.0.) INDEX=NCELLS
00910      .  .  .  .  .  .  ELSE
00911      .  .  .  .  .  .  RINDEX=(XL(J)-XLOW)/DELD+1.
00912      .  .  .  .  .  .  IF(RINDEX.LT.1.) RINDEX=1.
00913      .  .  .  .  .  .  IF(RINDEX.GT.NCELLS) RINDEX=NCELLS
00914      .  .  .  .  .  .  INDEX=RINDEX
00915      .  .  .  .  .  ...FIN
00916      .  .  .  .  CONCEN(INDEX)=CONCEN(INDEX)+WT(J)
00917      .  .  .  .  TCURY=TCURY+WT(J)
00918      .  .  .  .  NCEL=NCEL+1
00919      .  .  .  .  ...FIN

```

```

00920      . . . . .FIN
00921      . . . . .FIN
00922      . . . . .KK=0
00923      . . . . .WTMX=0.
00924      . . . . .CNFAC=CNFAC(I)/(DELD*WIDTH*HEIGHT*POROS)
00925      . . . . .DO(K=1,NCELLS)
00926      . . . . .  IF(MOD(K-1,5).EQ.0)
00927      . . . . .    . . . . .KK=KK+1
00928      . . . . .    . . . . .WT(KK)=0.
00929      . . . . .    . . . . .DO(KKK=K,K+5-1) WT(KK)=WT(KK)+CONCEN(KKK)
00930      . . . . .    . . . . .WTMX=MAX(WTMX,WT(KK))
00931      . . . . .    . . . . .FIN
00932      . . . . .    . . . . .CONCEN(K)=CONCEN(K)*CNFAC
00933      . . . . .    . . . . .FIN
00934      . . . . .    . . . . .CNFAC=CNFAC/5.
00935      . . . . .    . . . . .CNFAC1=CNFAC*WTMX/100.
00936      . . . . .    . . . . .CALL OPW(IPRO,RBEG,NCELLS,NCONCN*2.,+10.)
00937      . . . . .    . . . . .RBEG=RBEG+RINC
00938      C      . . . . .    . . . . .MAKE SHORT LIST OF CONCENTRATIONS ON THE LINE PRINTER
00939      . . . . .    . . . . .IF(I.NE.MNSTRT) WRITE(3,1553)
00940      1553  . . . . .    . . . . .FORMAT(1H1)
00941      . . . . .    . . . . .WRITE(3,6000) TITLE
00942      6000  . . . . .    . . . . .FORHAT(1X,20A4)
00943      . . . . .    . . . . .WRITE(3,5000) IRUN,RESFNM,PARFNM
00944      5000  . . . . .    . . . . .FORMAT(' RUN =',I2,' RESULT FILE =',10A1,
00945      1.    . . . . .    . . . . .' PARCEL FILE =',10A1)
00946      . . . . .    . . . . .WRITE(3,*) 'TIME PLANE SAVE #',NTSAV
00947      . . . . .    . . . . .WRITE(3,*) 'TOTAL CURIES=',TCURY
00948      . . . . .    . . . . .WRITE(3,2000) (NAME(KK,I),KK=1,9)
00949      2000  . . . . .    . . . . .FORMAT(' CONTAMINANT = ( ',9A1,')')
00950      . . . . .    . . . . .WRITE(3,*) 'TIME=',TIME,' TSTEP=',TSTEP
00951      . . . . .    . . . . .WRITE(3,*) 'NCELLS=',NCELLS,' DELTA X=',DELD
00952      . . . . .    . . . . .WRITE(3,*) 'XLOW=',XLOW,' XHIGH=',XHIGH
00953      . . . . .    . . . . .WRITE(3,*) 'MAXIMUM CURIES PER DELTA X SEGMENT=',WTMX
00954      . . . . .    . . . . .WRITE(3,*) 'FACTORS WT-CONC, X-CONC AND NO. PARCELS'
00955      . . . . .    . . . . .WRITE(3,*) CNFAC,CNFAC1,NCEL
00956      . . . . .    . . . . .DO(K=1,9) IQ(K)=K
00957      . . . . .    . . . . .WRITE(3,1154) (IQ(K),K=1,9)
00958      1154  . . . . .    . . . . .FORMAT(/T50,'PERCENT OF PEAK CURIES'
00959      1.    . . . . .    . . . . ./T13,100X,'1/' FRACTIONAL',T13,10X,9(I1,
00960      1.    . . . . .    . . . . ./' DISTANCE',T13,10('0++++++'),'0'))
00961      . . . . .    . . . . .DO(K=1,100)
00962      . . . . .    . . . . .  FRD=(K-1)*5*DELD+XLOW+2.5*DELD
00963      . . . . .    . . . . .  FRD=FRD/COLEN*100.
00964      . . . . .    . . . . .  IP=0
00965      . . . . .    . . . . .  IF(WTMX.GT.0.) IP=WT(K)/WTMX*100.
00966      . . . . .    . . . . .  IP=IP+1
00967      . . . . .    . . . . .  WRITE(3,1155) FRD
00968      1155  . . . . .    . . . . .  FORMAT(F11.3,T13,<IP>(1H*))
00969      . . . . .    . . . . .    . . . . .FIN
00970      . . . . .    . . . . .FIN
00971      . . . . .    . . . . .FIN
00972      C      . . . . .    . . . . .UPDATE THE TIME PLANE SAVE TIME AND THE TIME PLANE SAVE NUMBER
00973      . . . . .    . . . . .TSAV=TSAV+SAVFRQ
00974      . . . . .    . . . . .NTSAV=NTSAV+1
00975      . . . . .    . . . . .CLOSE(UNIT=3)

```

00976 ...FIN
00977 C

```

00978      TO PERFORM-DECAY-CALCULATIONS-FOR-CREATION
00979 C      . PROCEDURE TO PERFORM DECAY CALCULATIONS ON THE WEIGHT SUMS
00980 C      .
00981 C      .
00982 C      . OVER THE REGION WHERE WEIGHT SUMS EXIST DO THE DECAY CALCULATIONS
00983 C      . ON THE WEIGHT SUMS
00984 C      . MAXIMUM NUMBER OF DAUGHTER PARCELS CREATED EQUALS (NDAUG)
00985 C      . DECAY CELLS MULTIPLIED BY (IMDAUG)
00986 C      . DO(L=1,NDAUG)
00987 C      . . FIND OUT IF THERE ARE ANY WEIGHT SUMS IN THIS CELL
00988 C      . . AMIN=0.
00989 C      . . DO(KM=MNSTRT,NCON-1)
00990 C      . . . AMIN=MAX(WTSUM(L,KM),AMIN)
00991 C      . . . FIN
00992 C      . FOR CELLS THAT HAVE WEIGHT SUMS DO THE DECAY
00993 C      . IF(AMIN.GT.0.)
00994 C      . . WHEN (LINEAR)
00995 C      . . . DO(I=1,NCON)
00996 C      . . . . DO(J=1,NCON)
00997 C      . . . . . C(J,I)=0.
00998 C      . . . . . FIN
00999 C      . . . . . FIN
01000 C      . . . C(MNSTRT,MNSTRT)=WTSUM(L,MNSTRT)
01001 C      . . . DO(I=MNSTRT+1,NCON)
01002 C      . . . . C(I,I)=WTSUM(L,I)
01003 C      . . . . DO(J=MNSTRT,I-1)
01004 C      . *****
01005 C      . . . . . ===== WARNING ===== IN EVALUATING THE FOLLOWING CONSTANTS FOR
01006 C      . . . . . DECAY CHAINS USE DC(I=1) WHEN USING MASS UNITS, AND USE DC(I)
01007 C      . . . . . WHEN USING RADIOACTIVITY UNITS SUCH AS CURIES !!!!!
01008 C      . *****
01009 C      . . . . . C(I,J)=C(I-1,J)*DC(I)/(DC(I)-DC(J))
01010 C      . . . . . C(I,I)=C(I,I)-C(I,J)
01011 C      . . . . . FIN
01012 C      . . . . . FIN
01013 C      . . . . DO(K=MNSTRT+1,NCON)
01014 C      . . . . . XNEW=0.
01015 C      . . . . . DO(J=MNSTRT,K)
01016 C      . . . . . XNEW=XNEW+C(K,J)*EXP(-DC(J)*TSTEP)
01017 C      . . . . . FIN
01018 C      . . . . IF (XNEW.GT.0.)
01019 C      . FIND THE FIRST NON ZERO PARENT FOR DETERMINING CENTER OF MASS
01020 C      . FOR CREATION.
01021 C      . ?????????? SHOULD IT BE THE FIRST MAJOR PARENT ??????????????????
01022 C      . . . . . BMIN=0.
01023 C      . . . . . KK=K-1
01024 C      . . . . . FREM1=.FALSE.
01025 C      . . . . . REPEAT UNTIL(FREM1)
01026 C      . . . . . . WHEN(WTSUM(L,KK).NE.0.)
01027 C      . . . . . . FREM1=.TRUE.
01028 C      . . . . . BMIN=1.

```



```

01085      . . . . .FIN
01086      . . . . .FIN
01087      . . . . .FIN
01088      . . . . .FIN
01089      . . . . .FIN
01090      . DO(I=MNSTRT+1,NCON)
01091      . . RSTRT=RNPS(I)+1
01092      . . NPSTSC=(RNPS(I)+1)/RNPARC+1
01093      . . PBEG=ISTOZZ+(I-1)*NSETZP+(NPSTSC-1)*PINC
01094      . . CALL DPR(IPRO,PBEG,XL,PAWORD)
01095      . . RTOSS=RSTRT-FLOAT(NPSTSC-1)*RNPARC
01096      . . IOFF=RTOSS+.1
01097      . . J=IOFF
01098      . . IF(J.LE.0.OR.J.GT.1024)
01099      . . . WRITE(2,*) 'ERROR A',RNPS(I),NPSTSC,RNPARC,RTOSS
01100      . . . CLOSE(UNIT=2)
01101      . . . . .FIN
01102  C      . . COUNT NUMBER OF DECAY CELLS THAT WILL CREATE DAUGHTERS
01103      . . NFULL=0
01104      . . DO(K=1,NDAUG)
01105      . . . IF(CONC(K,I).GT.0.)NFULL=NFULL+1
01106      . . . . .FIN
01107  C      . . RELEASE (NRELS) NUMBERS OF EQUAL WEIGHT PARCELS
01108  C      . . ACCOUNT FOR RANDOM DECAY TIME AND RELATIVE MOTION
01109  C      . . OF PARENT AND DAUGHTER
01110      . . NRELS0=1
01111      . . IF(NFULL.NE.0) NRELS0=RNDAUG/NFULL+.1
01112      . . VDAUG=VEL/AK(I)
01113      . . VPT=VEL/AK(I-1)
01114      . . DELTAV=VDAUG-VPT
01115      . . DX0=VPT*TSTEP
01116      . . XLO=COLEN-DX0
01117      . . DPLACE=DUMDC/SQRT(AK(I))
01118  C      . . INITIALIZE PARCEL RESOLUTION CONTROL ON DECAY DENSITY
01119      . . IF(ICNTRL.EQ.1)
01120      . . . XSPACE=XSEP(I)*NRELS0
01121      . . . IMULT0=(TOMAX+TSTEP*ABS(DELTAV))/XSPACE
01122      . . . . .FIN
01123  C      . . DETERMINE LIMITING DISTANCE FOR PARCELS TO DECAY WITHOUT EXITING
01124      . . INDEX=0
01125      . . IF(FRAC.GT.0.)
01126      . . . XOLD=COLEN+ALOG(FRAC)*VDAUG/DC(I)
01127      . . . WHEN(TOMAX.EQ.0.)INDEX=NDAUG
01128      . . . ELSE
01129      . . . . RINDEX=(XOLD-XMIN)/TOMAX+1
01130      . . . . IF(RINDEX.LT.1.)RINDEX=1,
01131      . . . . IF(RINDEX.GT.RNDAUG)RINDEX=RNDAUG
01132      . . . . INDEX=RINDEX
01133      . . . . .FIN
01134      . . . IF(XMX(I-1)+DX0.GT.COLEN) INDEX=INDEX-1
01135      . . . . .FIN
01136  C      . . NOTE CENTMP IS CENTER OF MASS AT START OF PRESENT TIME STEP
01137      . . DO(K=1,NDAUG)
01138      . . . IF(CONC(K,I).GT.0.)
01139      . . . . X1=XMIN+(K-1)*TOMAX
01140      . . . . X2=X1+TOMAX

```



```

01141 C . . . . MULTIPLY THE NUMBER OF RELEASED DAUGHTERS IF THEY WILL EXIT
01142 . . . . NRELS=NRELS0
01143 . . . . IMULT=1
01144 . . . . IF(K,GT,INDEX)
01145 . . . . . IMULT=IMDAUG(I)
01146 C . . . . . CONTROL DECAY DENSITY
01147 . . . . . IF(ICNTRL,EQ,1)
01148 . . . . . . SIGMA=COLEN-X1
01149 . . . . . . IF(VEL,GT,0)SIGMA=-DLEN+SQRT(DLEN*DLEN+2.*DLEN*SIGMA)
01150 . . . . . . IMULT=IMULT0+2.*SIGMA/XSPACE
01151 . . . . . . IF(IMULT,LT,1)IMULT=1
01152 . . . . . . IF(IMULT,GT,IMDAUG(I))IMULT=IMDAUG(I)
01153 . . . . . . . . FIN
01154 . . . . . NRELS=NRELS*IMULT
01155 . . . . . . . . FIN
01156 C . . . . . MAKE NRELS AN ODD NUMBER OF PARCELS
01157 . . . . . NRELS=NRELS/2
01158 . . . . . NRELS=2*NRELS+1
01159 . . . . . NHALF=(NRELS-1)/2+1
01160 . . . . . WTPP=CONC(K,I)/NRELS
01161 C . . . . . DISTRIBUTE PARCELS BALANCED ABOUT CENTER OF MASS
01162 . . . . . IF(NRELS,GT,1)
01163 . . . . . . D1=(CENTMP(K,I)-X1)/NHALF
01164 . . . . . . D2=(X2-CENTMP(K,I))/NHALF
01165 . . . . . . WHEN(D1,GT,0,AND,D2,GT,0.)
01166 . . . . . . . W1=2*WTPP*D2/(D1+D2)
01167 . . . . . . . W2=2*WTPP-W1
01168 . . . . . . . . FIN
01169 . . . . . . ELSE
01170 . . . . . . . D1=0.
01171 . . . . . . . D2=0.
01172 . . . . . . . W1=WTPP
01173 . . . . . . . W2=WTPP
01174 . . . . . . . . FIN
01175 . . . . . . . . FIN
01176 . . . . . DD1=D1
01177 . . . . . DD2=D2
01178 . . . . . DO(N=1,NRELS)
01179 . . . . . . XL(J)=CENTMP(K,I)
01180 . . . . . . WT(J)=WTPP
01181 . . . . . . IF(N,LT,NHALF)
01182 . . . . . . . XL(J)=XL(J)-DD1
01183 . . . . . . . WT(J)=W1
01184 . . . . . . . DD1=DD1+D1
01185 . . . . . . . . FIN
01186 . . . . . . IF(N,GT,NHALF)
01187 . . . . . . . XL(J)=XL(J)+DD2
01188 . . . . . . . WT(J)=W2
01189 . . . . . . . DD2=DD2+D2
01190 . . . . . . . . FIN
01191 . . . . . . RANDT=TSTEP*RAN(I1,I2)
01192 . . . . . . IF(RANDT*VPT,GE,(XL(J)-XL0))
01193 . . . . . . . RANDM=.5-RAN(I1,I2)
01194 . . . . . . . DX=DX0+DELTAV*RANDT+DPLACE*RANDM*SQRT(RANDT)
01195 . . . . . . . XL(J)=XL(J)+DX
01196 . . . . . . . WHEN(XL(J),LT,COLEN)

```

```

01197 C . . . . . AGAIN UPDATE CONTAMINANT SPREAD LIMITS
01198 . . . . . XMX(I)=MAX(XMX(I),XL(J))
01199 . . . . . XMN(I)=MIN(XMN(I),XL(J))
01200 . . . . . IF(XMN(I).LT.0.)XMN(I)=0.
01201 . . . . . ...FIN
01202 . . . . . ELSE
01203 C . . . . . COUNT NEW PARCELS LOST DURING THE DECAY PERIOD
01204 . . . . . WT(J)=-1.*WT(J)
01205 . . . . . TBND=TIME+TSTEP*(COLEN-XL(J))/DX
01206 . . . . . XL(J)=TBND
01207 . . . . . RNPL(I)=RNPL(I)+1
01208 . . . . . NLOST(I)=NLOST(I)+1
01209 . . . . . ...FIN
01210 . . . . . IF(J.EQ.NPARCS)
01211 . . . . . CALL DPW(IPRO,PBEG,XL,PAWORD)
01212 . . . . . PBEG=PBEG+PINC
01213 . . . . . J=0
01214 . . . . . ...FIN
01215 . . . . . J=J+1
01216 C . . . . . COUNT ALL PARCELS CREATED DURING THE DECAY PERIOD
01217 C . . . . . BUT CREATED WITHIN THE SYSTEM
01218 . . . . . RNPS(I)=RNPS(I)+1
01219 . . . . . ...FIN
01220 . . . . . ...FIN
01221 . . . . . ...FIN
01222 . . . . . ...FIN
01223 . . . . . IF(J.NE.1) CALL DPW(IPRO,PBEG,XL,PAWORD)
01224 . . . . . ...FIN
01225 . . . . . ...FIN
01226 C
01227 C

```

```

-----
01228 . . . . . TO RELEASE-PARCELS
01229 C . . . . . PROCEDURE TO RELEASE THE PARCELS FROM THE SOURCE
01230 C . . . . . FOR TIMES BETWEEN THE RESIDENCE TIME AND RELEASE TIME
01231 C . . . . .
01232 C . . . . . RELEASE PARCELS FOR EACH CONTAMINANT
01233 . . . . . DO(I=1,NCON)
01234 . . . . . NPRPTS=NPAREL(I)
01235 C DETERMINE THE AVERAGE DISTANCE THE THE PARCELS RELEASED
01236 C DURING THIS TIME STEP WOULD MOVE
01237 . . . . . OSCDX=VEL*TSTEP/AK(I)
01238 C . . . . . ONLY RELEASE PARCELS THAT WIL HAVE A NON-ZERO WEIGHT
01239 . . . . . IF(RATE(I).GT.0..AND.NPRPTS.GT.0)
01240 C . . . . . INITIALIZE THE NUMBER ALREADY RELEASED THIS TIME STEP
01241 . . . . . NRELSO=0
01242 . . . . . NRELZ=0
01243 . . . . . UNTIL(NRELSO,EQ.NPRPTS)
01244 . . . . . RSTRT=RNPS(I)+NRELSO+1
01245 . . . . . RENQ=RNPS(I)+NPRPTS
01246 . . . . . NPSTSC=(RNPS(I)+NRELSO+.1)/RNPARG+1
01247 . . . . . PADRF=ISFRHZ+(I-1)*NSETZP+(NPSTSC-1)*PINC
01248 . . . . . CALL DPR(IPRO,PADRF,XL,PAWORD)
01249 . . . . . RTOSS=RSTRT-FLOAT(NPSTSC-1)*RNPARG

```

```

01250      . . . . IOFF=RTOSS+.1
01251      . . . . JJJ=IOFF
01252      . . . . IF(JJJ.LE.0.OR.JJJ.GT.1024)
01253      . . . . . WRITE(2,*) 'ERROR B',RNPS(I),NPSTSC,RNPARC,RTOSS,NRELSO
01254      . . . . . CLOSE(UNIT=2)
01255      . . . . . ...FIN
01256      . . . . . ROFFE=REND-FLOAT(NPSTSC-1)*RNPARC
01257      . . . . . WHEN(ROFFE.GT.RNPARC) IOFFE=NPARCS
01258      . . . . . ELSE IOFFE=ROFFE+.1
01259      . . . . . NRELSO=IOFFE-IOFF+1+NRELSO
01260      . . . . . DO(J=IOFF,IOFFE)
01261      . . . . . . NRELZ=NRELZ+1
01262      . . . . . . RANDM1=RAN(I1,I2)
01263      C ALLOW+RELEN*RANDM1 GIVES THE SPREAD OVER THE RELEASE INTERFACE.
01264      C (-DSCDX) ADJUSTS THE POSITION SINCE IT WILL BE MOVED THIS
01265      C DISTANCE BY THE MOVE PARCEL PROCEDURE.
01266      C NRELZ*DSCDX/NPRPTS DISTRIBUTES PARCELS UNIFORMLY OVER THE
01267      C RELEASE TIME.
01268      . . . . . XL(J)=ALLOW+RELEN*RANDM1-DSCDX+NRELZ*DSCDX/NPRPTS
01269      . . . . . XMN(I)=MIN(XMN(I),XL(J))
01270      . . . . . XMX(I)=MAX(XMX(I),XL(J))
01271      . . . . . IF(XMX(I).GT.COLEN) XMX(I)=COLEN
01272      . . . . . IF(XMN(I).LT.0.) XMN(I)=0.
01273      . . . . . WT(J)=RATE(I)/FLOAT(NPRPTS)
01274      . . . . . ...FIN
01275      . . . . . CALL DPW(IPRO,PADRF,XL,PAWORD)
01276      . . . . . ...FIN
01277      . . . . . RNPS(I)=RNPS(I)+NPRPTS
01278      . . . . . IF(JSHTCH.EQ.1)
01279      . . . . . . WRITE(2,*) 'RELEASE=PARCELS',RNPS(I),DSCDX,NRELZ
01280      . . . . . . CLOSE(UNIT=2)
01281      . . . . . ...FIN
01282      . . . . . ...FIN
01283      . . . . . ...FIN
01284      . . . . . ...FIN
01285      C
01286      C
01287      END

```

PROCEDURE CROSS-REFERENCE TABLE

```

00287 CALCULATE-TIME-STEP
00167

00347 COMPRESS-PARCELS
00200

00268 COPY-FROM-TO
00236 00243 00250 00261

00554 CREATE-PARCELS
01035 01084

00593 DECAY-INVENTORY

```

(FLECS VERSION 22.46) 19-MAY-80 14:26:30 PAGE 00025

00172

00576 DETERMINE-DISCHARGE-RATE
00170

00663 DO-THE-INTERPOLATION
00692 00695

00685 GET-FRACTION-REMAINING
00325 00329 00581

00701 MOVE-PARCELS
00174

00867 OUTPUT-TIME-PLANE
00175

00978 PERFORM-DECAY-CALCULATIONS-FOR-CREATION
00176

01228 RELEASE-PARCELS
00171

(FLECS VERSION 22.46)

PLTCVT LISTING

```

-----
00001 C *****PLTCVT,FLX*****
00002 C ***** 1/29/79 *****
00003 C
00004 C PROGRAM TO PLOT CONCENTRATION VS TIME PLOTS FROM ASSEMBLED
00005 C DEAD PARCEL DATA
00006 C
00007 C PARAMETER MAXTS=500
00008 C MMTID TRANSPORT PROGRAM HEADER BLOCK
00009 C
00010 C PARAMETER NCON=9,NPARCS=1024,NSETM=50
00011 C PARAMETER MNDPC=75,NCONCN=500,MAXTPS=100
00012 C PARAMETER WRDHDR=704,PARSIZ=18500,AHDR0=18497,AHDR1=18494,
00013 C PARAMETER ADRTPS=14403,ADRCOM=17400,
00014 C NCON=MAX, NO OF CONTAMINANTS
00015 C NPARCS=MAX, NO. OF PARCELS CORE CAN HOLD (STORED ON ONE DISK PAGE)
00016 C MUST BE MULTIPLE OF 256
00017 C NSETM=MAXIMUM NUMBER OF NPARCS SETS OF PARCELS (NORMAL)
00018 C NSETZ=MAXIMUM NUMBER OF NPARCS SETS OF PARCELS (ACTUAL THIS RUN)
00019 C MNDPC=MAXIMUM NUMBER OF DAUGHTER PARCELS THAT CAN BE CREATED
00020 C PER TIME STEP
00021 C NCONCN=MAXIMUM RESOLUTION OF SPATIAL CONCENTRATION
00022 C MAXTPS=MAXIMUM NUMBER OF TIME PLANE SAVES
00023 C WRDHDR=NUMBER OF WORDS IN THE HEADER BLOCK
00024 C PARSIZ=BLOCKS IN THE PARCEL LOCATION FILE
00025 C AHDR0=ADDRESS FOR THE INPUT HEADER BLOCK
00026 C AHDR1=ADDRESS FOR THE WORKING HEADER BLOCK
00027 C ADRTPS=OFFSET ADDRESS FOR THE TIME PLANE SAVES
00028 C ADRCOM=OFFSET ADDRESS FOR THE COMPRESS WORK SPACE
00029 C
00030 C BYTE FNAME(28),RSTR
00031 C LOGICAL FREM1,FREM2,SAVE
00032 C BYTE MMTRUN(10)
00033 C HEADER BLOCK DATA(WRDHDR WORDS)
00034 C BYTE NAME,PARFNM
00035 C LOGICAL*1 DECHAN,LINEAR,ELIM
00036 C COMMON /HOR1/
00037 C 1NCON,AINV(NCON),AKD(NCON),HL(NCON),DISPC,BETA,RESTIM,
00038 C 2NAME(9,20),CONFAC(NCON),VEL,COLEN,WIDTH,HEIGHT,POROS,SIMLEN,
00039 C 3TSNO,TIME,TSTEP,TSV,SAVFRQ,NTSAV,OTMAXM,OTHINM,RELCEN,RELEN,
00040 C 4ALOW,RELTIM,TOTFRL,FRACMX,FREL,TREM(25),FREM(25),NFRPTS,
00041 C 5FACTOR(NCON,NCON),XMX(NCON),XMN(NCON),AK(NCON),DC(NCON),
00042 C 6DUMDC,MNSTRT,MNS,MNSL,I1,I2,ISFROM,ISTO,IHALF,NSETZP,RNPARM,
00043 C 7PAWORD,PINC,RINC,NDGPTS,NDGPTR,NPARCM,NPLMX,IRUN,RNPARC,
00044 C 8RNPS(NCON),RNPL(NCON),PLOST(NCON),NLOST(NCON),NSETZ,
00045 C 9ITL4(20),NONODE(NCON),NPAREL(NCON),
00046 C 1PARFNM(11),DECHAN,LINEAR,ELIM(NCON)
00047 C
00048 C
00049 C COMMON /BLKING/TONS,IRENT,BTIML,BTIMH,IBCELL,BOTIM,
00050 C 1TMAXM,TMINM,TWGT,TOTNO,WTHAXM,WTHXT,RNOMX,RNOMXT,
00051 C 1WLOW,PARLOW,WITHI,PARHI,IDOIT,SINC,ISFAC,TCMAXM,
00052 C 1CMAXM,CRMAX,FLOWS,NPTS
00053 C COMMON /PTR/IPRO,IPR1,IPR2,ICMTR2

```

```

00054      COMMON /BLK1/TMIN,TMAX,CMAX,TCMAX,NVALL,CT(2,MAXTS),
00055      1TOTWT,ITOSIT
00056      C
00057      C
00058      COMMON /GRAPH/XLAB(20),NXCH,XLABSZ,IXIVRT,IXLOG,XHGT,
00059      1  YLAB(20),NYCH,YLABSZ,IYIVRT,IYLOG,YHGT,
00060      2  TITLE(20),NTITL,TITLSZ,TITLE1(20),NTITL1,TITSZ1,
00061      3  IASCL,FXMAX,FXMIN,FDX,FYMAX,FYMIN,FDY,
00062      4  IXLGMX,IXLGMN,IYLGMM,IYLGMM,
00063      5  IDEVIN,INTACT,OSFCT,IPENN,IFINPS,
00064      6  ISYM,SYMSZ,IOVER,MINDVX,MINDVY,XTOTAL,YTOTAL,
00065      7  TITLE2(20),NTITL2,TITSZ2,TITLEL(20),NTITLL,TITSZL,
00066      8  NDIG,NTIML,NMAJDV
00067      C
00068      COMMON /HDR/
00069      1NCO1,AIN1(NCNUM),AK1(NCNUM),H1(NCNUM),DISP1,BETA1,RESTI1,
00070      2TNAM1(9,20),CONFA1(NCNUM),VE1,COLE1,WIDT1,HEIGH1
00071      3,POR2,SIMLE1,TSN1,ITOS1,TITLR(20)
00072      C
00073      COMMON /BHEAD/NFTBS,NCONV(NCNUM),SVEL(NCNUM),
00074      1SCOL(NCNUM),CF(NCNUM),SW(NCNUM),SH(NCNUM),STSNO(NCNUM),TINV,
00075      1POR,TITL3(20)
00076      C
00077      BYTE FIL(28),FILN(28),TNAM1
00078      IFRST=0
00079      IFRST1=0
00080      REPEAT UNTIL(IDON.EQ.1)
00081      . IF(IFRST.EQ.0)
00082      . . WRITE(6,*) 'ENTER 0 FOR RATE VS TIME'
00083      . . WRITE(6,*) 'ENTER 1 FOR CONCENTRATION VS TIME'
00084      . . READ(6,*) IKINDP
00085      . . WRITE(6,*) 'ENTER THE METRIC TONS OF HEAVY METAL OR 1.'
00086      . . READ(6,*) TONS
00087      . . ...FIN
00088      . CALL GETCVT(IFRST,ICON,IKINDP)
00089      . IF(IFRST1.EQ.0)
00090      C SET UP LUN ASSIGNMENTS AND INITIALIZE FOR GRAPHING
00091      . . XTOTAL =0.0
00092      . . YTOTAL = 0.0
00093      . . START2=0.
00094      . . WRITE(6,*) 'ENTER PLOT DEVICE-(1) DB0,(2) XY, (3) TG'
00095      . . READ(6,*) IDEV
00096      . . SELECT (IDEV)
00097      . . . (1)
00098      . . . . WRITE(6,*) 'ENTER WHICH DISK UNIT 0,1 OR 2'
00099      . . . . WRITE(6,*) 'ENTER 3 FOR MT0,4 FOR MT1'
00100      . . . . READ(6,*) IUN
00101      . . . . SELECT (IUN)
00102      . . . . . (0) CALL ASNLUN(4,'DB',0)
00103      . . . . . (1) CALL ASNLUN(4,'DB',1)
00104      . . . . . (2) CALL ASNLUN(4,'DB',2)
00105      . . . . . (3) CALL ASNLUN(4,'MT',0)
00106      . . . . . (4) CALL ASNLUN(4,'MT',1)
00107      . . . . ...FIN
00108      . . . . ...FIN
00109      . . . . (2) CALL ASNLUN(4,'XY',0)

```

```

00110      . . . (3) CALL ASNLUN (4,'TG',0)
00111      . . . ...FIN
00112      . . . WHEN(IKINDP.EQ.0)
00113      . . . . OPEN(UNIT=2,NAME='PLTRAT.CRD',TYPE='OLD',FORM='FORMATTED'
00114      1. . . . ,READONLY)
00115      . . . ...FIN
00116      . . . ELSE
00117      . . . . OPEN(UNIT=2,NAME='PLTCVT.CRD',TYPE='OLD',FORM='FORMATTED'
00118      1. . . . ,READONLY)
00119      . . . ...FIN
00120      . . . READ(2,333) FIL
00121      333 . . . FORMAT(28A1)
00122      . . . READ(2,50) XLAB
00123      . . . READ(2,50) YLAB
00124      . . . READ(2,51) NXCH,NYCH,XLABSZ,YLABSZ
00125      . . . READ(2,52) XHGT,YHGT,IXIVRT,IYIVRT,IXLOG,IYLOG
00126      . . . READ(2,55) IASCL,IDEVIN,INTACT
00127      . . . READ(2,54) FXMAX,FXMIN,FDX,MINDVX
00128      . . . READ(2,54) FYMAX,FYMIN,FDY,MINDVY
00129      . . . READ(2,55) IXLGMX,IXLGMN,IYLGMM,IYLGMM
00130      . . . READ(2,52) OSFCT,SYMSZ,ISYM
00131      . . . READ(2,55) NDIG,NTIML,NMAJDOV
00132      50 . . . FORMAT(20A4)
00133      51 . . . FORMAT(2I5,2F6.2)
00134      52 . . . FORMAT(2F6.2,4I5)
00135      54 . . . FORMAT(3F6.2,I5)
00136      55 . . . FORMAT(4I5)
00137      . . . CLOSE(UNIT=2)
00138      . . . WRITE(6,*) 'ENTER NAME OF DOSE FILE'
00139      . . . READ(6,333) (FILN(I),I=1,9)
00140      . . . FILN(10)=' '
00141      . . . N=ICHR(FILN)
00142      . . . CALL DPFILE(1,FILN,400.,ICMTR2)
00143      C SETUP THE HEADER BLOCKS
00144      . . . WRITE(6,*) 'ENTER A TITLE FOR THE GRAPHS'
00145      . . . READ(6,56) TITL3
00146      56 . . . FORMAT(20A4)
00147      . . . DO(I=1,20) TITLR(I)=TITL3(I)
00148      . . . NFBSS=1
00149      . . . DO(I=1,NCNUM)
00150      . . . . SYEL(I)=VEL
00151      . . . . SCOL(I)=COLEN
00152      . . . . CF(I)=CONFAC(I)
00153      . . . . SH(I)=WIDTH
00154      . . . . SH(I)=HEIGHT
00155      . . . . STSNO(I)=1.
00156      . . . ...FIN
00157      . . . CALL DPR(IPRO,AHORO,NC01,184.)
00158      . . . POR2=POROS
00159      . . . POR=POROS
00160      . . . IFRST=1
00161      . . . ...FIN
00162      . . . CALL PLTIT(IFRST,ICON,IDEV,START2,IKINDP)
00163      . . . WRITE(6,*) 'ENTER A 1 FOR STOP 0 TO CONTINUE 2 FOR NEW'
00164      . . . READ(6,*) IDON
00165      . . . IF(IDON.EQ.2)

```


(FLECS VERSION 22.46) 19-MAY-80 15:06:55 PAGE 00004

```
00166      . . IFRST=0 .
00167      . . IFRST1=0
00168      . . CALL DPCL(IPR0)
00169      . . CALL DPCL(IPR2)
00170      . . CALL DPCL(ICMTR2)
00171      . . CLOSE(UNIT=4)
00172      . . CLOSE(UNIT=5)
00173      . . . . FIN
00174      . . . . FIN
00175      STOP
00176      END
```

(FLECS VERSION 22.46)

PLTIT AND ASSOCIATED SUBROUTINE LISTING

```

-----
00001      SUBROUTINE PLTIT(IFRST,ICON,IDEV,START2,IKINDP)
00002      C *****PLTREG.FLX*****
00003      C *****4/18/79*****
00004      C      COLE AND ASSOCIATES
00005      C
00006      C      PROGRAM SMOOTHS AND PLOTS CONCENTRATION VS TIME CURVE
00007      C
00008      C      PARAMETER NCNUM=9,MAXTS=500
00009      C      BYTE FILN(24),FIL(24),TNAME,TITLR1(80),GETOUT(10)
00010      C      DIMENSION TFLW(NCNUM),CC(101)
00011      C
00012      C      COMMON /BLKING/TONS,IBENT,BTIML,BTIMH,IBCELL,BOTIM,
00013      C      1TMAXH,TMINM,TWGHT,TOTNO,WTMAXM,WTMXT,RNOMX,RNOMXT,
00014      C      1WTLOW,PARLOW,WTMI,PARHI,IOOIT,SINC,ISFAC,TCHAXM,
00015      C      1CMAXM,CRMAX,FLOWS,NPTS
00016      C      COMMON /PTR/IPRO,IPR1,IPR2,ICMTR2
00017      C      COMMON /GRAPH/XLAB(20),NXCH,XLABSZ,IXIVRT,IXLOG,XHGT,
00018      C      1YLAB(20),NYCH,YLABSZ,IYIVRT,IYLOG,YHGT,
00019      C      2TITL(20),NTITL,TITLSZ,TITLE1(20),NTITL1,TITSZ1,
00020      C      3IASCL,FXMAX,FXMIN,FDX,FYMAX,FYMIN,FDY,
00021      C      4IXLGMX,IXLGMN,IYLGMX,IYLGMN,
00022      C      5IDEVIN,INTACT,OSFCT,IPENN,IFINPS,
00023      C      6ISYM,SYMSZ,IOVER,MINDVX,MINDVY,XTOTAL,YTOTAL,
00024      C      7TITL2(20),NTITL2,TITSZ2,TITLEL(20),NTITLL,TITSZL,
00025      C      8NDIG,NTIML,NMAJDV
00026      C
00027      C      COMMON /HDR/
00028      C      1NCON,AINV(NCNUM),AKD(NCNUM),HL(NCNUM),DISPC,BETA,RESTIN,
00029      C      2TNAME(9,20),CONFAC(NCNUM),VEL,COLEN,WIDTH,HEIGHT,POR2,SIMLEN,
00030      C      3TSNO,ITOS1,TITLR(20)
00031      C
00032      C      COMMON /BHEAD/NFTBS,NCONV(NCNUM),SVEL(NCNUM),
00033      C      1SCOL(NCNUM),CF(NCNUM),SW(NCNUM),SH(NCNUM),STSNO(NCNUM),TINV,
00034      C      1POR,TITL3(20)
00035      C
00036      C      COMMON /BLK1/TMIN,TMAX,CMAX,TCHAX,NVALL,CT(2,MAXTS),
00037      C      1TOTWT,ITOSIT
00038      C      COMMON /PARCEL/NNPTS,TFLOW,TN(MAXTS),CN(MAXTS),ITOS3,
00039      C      1T(MAXTS),C(MAXTS),IAAA(92)
00040      C
00041      C      EQUIVALENCE (TITLR(1),TITLR1(1))
00042      C      TOTAL FLOW RATE
00043      C      TFLOW= 0.
00044      C      DO(I=1,NFTBS)
00045      C      . TFLOW = TFLOW+SVEL(I)*SW(I)*SH(I)*POR
00046      C      . TFLW(I)=SVEL(I)*SW(I)*SH(I)*POR*28317./1.E+6
00047      C      ...FIN
00048      C      TFLOW1=TFLOW*28317./1.E+6
00049      C      TFLQW=TFLOW/365./192.5
00050      C
00051      C      IFT=1
00052      C      IFP=ICON
00053      C      WRITE OUT THE TITLE

```

```

00054 C
00055 WHEN(IKINOP,EQ,0)
00056 . ENCODE(57,607,TITLE)(TNAME(K,IFP),K=1,9),IFT
00057 607 . FORMAT('ISOTOPE',1X,9A1,8X,'FLOW TUBE',12,9X
00058 1. , 'RATE VS TIME')
00059 . NTITL=57
00060 ...FIN
00061 ELSE
00062 . ENCODE(66,60,TITLE)(TNAME(K,IFP),K=1,9),IFT
00063 60 . FORMAT('ISOTOPE',1X,9A1,8X,'FLOW TUBE',12,9X
00064 1. , 'CONCENTRATION VS TIME')
00065 . NTITL=66
00066 ...FIN
00067 I=81
00068 6500 I=I-1
00069 IF(I,EQ,10) GO TO 6501
00070 IF(TITLR1(I),EQ,' ') GO TO 6500
00071 6501 NTITLL=I
00072 DO(I=1,20) TITLEL(I)=TITLR(I)
00073 ENCODE(66,66,TITLE2)AKD(IFP),HL(IFP),BETA
00074 66 FORMAT('KD=',F8.2,1X,'ML/G',3X,'HALF-LIFE=',1PE10.4,1X,'YEARS'
00075 1,3X,'BETA=',0PF8.1,1X,'G/ML')
00076 NTITL2 = 66
00077 NTITL1=74
00078 TITSZ1 = 0.
00079 TITLZ = 0.
00080 TITSZ2=0.
00081 TITLZL=0.
00082 C
00083 C SET THE NUMBER OF POINTS IN CURVE
00084 589 NOPTS = NVALL
00085 NPTS = NOPTS
00086 C REASSIGN THE XY: TO TG:
00087 CLOSE(UNIT=4)
00088 CALL ASNLUN(4,'TG',0)
00089 XTOTAL = 0.
00090 YTOTAL = 0.
00091 IOVER=0
00092 IFINPS=1
00093 C PREPARE GRAPH POINTS
00094 C ASSUMES A CONSTANT TIME STEP
00095 DT=CT(1,2)-CT(1,1)
00096 TMIN=CT(1,1)
00097 TMAX=CT(1,NOPTS)
00098 SINC=DT
00099 ISFAC=1
00100 FLOWS=IFLW(IFT)
00101 CRMAX=0.
00102 TCMAXM=0.
00103 C
00104 C ADJUSTS THE ACCURACY FOR NEGLECT OF CONCENTRATION
00105 WRITE(6,*)'ENTER FRACTIONAL ACCURACY LIMIT'
00106 WRITE(6,*)'ENTER ZERO FOR NO LIMIT'
00107 READ(6,*) FRACT
00108 CMIN=0.
00109 CMAXM=0.

```

```

00110      TRATIO=0.
00111      DO (J=1,NOPTS)
00112      .   C(J)=CT(2,J)
00113      .   T(J)=CT(1,J)
00114      .   IF(C(J).GT.CMAXM)
00115      .     CMAXM=C(J)
00116      .     TCMAXM=T(J)
00117      .   ...FIN
00118      ...FIN
00119      IF(FRACT.GT.0.)
00120      .   CMIN=CMAXM*FRACT
00121      .   TRATIO=1.
00122      .   REPEAT UNTIL(TRATIO.LE.FRACT)
00123      .     CMINO=CMIN
00124      .     CINVO=0.
00125      .     CINV=0.
00126      .     DO (J=1,NOPTS)
00127      .       .   WHEN(C(J).GT.CMIN) CINVO=CINVO+C(J)
00128      .       .   ELSE CINV=CINV+C(J)
00129      .       .   ...FIN
00130      .     TRATIO=CINV/CINVO
00131      .     CMIN=CMIN/10.
00132      .     ...FIN
00133      .     CMIN=CMINO
00134      .     ...FIN
00135      C
00136      C      INTEGRATE CONCENTRATION OVER TIME
00137      CINVO=0.
00138      DO (J=1,NOPTS)
00139      .   CN(J)=0.
00140      .   IF(C(J).GT.CMIN)
00141      .     CINVO=CINVO+C(J)
00142      .     CN(J)=CINVO
00143      .     ...FIN
00144      .     ...FIN
00145      CINVO=CINVO*DT
00146      CRMAX=CMAXM
00147      CONOITIONAL
00148      .   (IKINDP.EQ.0) CMAXM=CMAXM/TFLW(IFT)
00149      .   (IKINDP.EQ.1)
00150      .     CINVO=CINVO*TFLW(IFT)
00151      .     CRMAX=CRMAX*TFLW(IFT)
00152      .     ...FIN
00153      .     ...FIN
00154      WRITE(6,*) 'C INVO=',CINVO,'TOTAL WT=',TOTWT
00155      WRITE(6,*) 'FRACTIONAL ERROR=',TRATIO
00156      AA=AINV(IPP)/NFTBS*TONS
00157      ENCODE(74,6502,TITLE1) AA,TOTWT
00158      6502  FORMAT('INITIAL INVENTORY=',1PE10.4,' CURIES.',
00159      1' PRESENT INVENTORY=',1PE10.4,' CURIES. ')
00160      C
00161      CALL GRAPH(T,C,NOPTS)
00162      C
00163      C      EXTRAPOLATE THE CONCENTRATION INTEGRAL
00164      ISTART=1
00165      10  DO (I=ISTART,NOPTS) IF(CN(I).EQ.0.) GO TO 1

```

```

00166      GO TO 20
00167  1      M=I-1
00168      IF(M.EQ.0) M=1
00169      L=0
00170      DO (J=1,NOPTS)
00171      .   IF(CN(J).GT.0.) GO TO 2
00172      .   CN(J)=CN(M)
00173      .   L=L+1
00174      ...FIN
00175      GO TO 20
00176  2      ISTART=J
00177      SLOPE=(CN(J)-CN(M))/(T(J)-T(M))
00178      DO (K=1,L)
00179      .   I=M+K
00180      .   CN(I)=SLOPE*(T(I)-T(M))+CN(M)
00181      ...FIN
00182      GO TO 10
00183  20      WRITE(6,*) 'SMOOTH(1), PLOT CURRENT CONCENTRATION INTEGRAL(2), '
00184      WRITE(6,*) 'OR GET NEXT FLOW TUBE(3)'
00185      READ(6,*) IQUES
00186      IF(IQUES.EQ.1) GO TO 602
00187      IFINPS=0
00188      CALL PLOTND
00189      IF(IQUES.EQ.2) GO TO 603
00190      IF(IQUES.EQ.3) RETURN
00191  602      WRITE(6,*) 'INCLUDE SMOOTH ENDS? 0=NO, 1=YES'
00192      READ(6,*) NEND
00193  C      CONTROLS SIZE OF END WINDOW
00194      WHEN(NEND.EQ.1)
00195      .   WRITE(6,*) 'ENTER INTEGER WINDOW FACTOR AND REPEAT COUNT'
00196      .   READ(6,*) NDT,NCOUNT
00197      ...FIN
00198      ELSE
00199      .   WRITE(6,*) 'ENTER (2) TIME DOMAIN DIVIDING POINTS'
00200      .   READ(6,*) T1,T2
00201      .   WRITE(6,*) 'ENTER (3) INTEGER WINDOWS AND REPEAT COUNT'
00202      .   READ(6,*) NDT1,NDT,NDT2,NCOUNT
00203      .   IF(NDT1.GT.50)NDT1=50
00204      .   IF(NDT2.GT.50)NDT2=50
00205  C      COMPUTE THE INTEGER DIVIDE POINTS FOR THE THREE REGIONS
00206      .   N1=(T1-TMIN)/DT+1.
00207      .   N2=(T2-TMIN)/DT+1.
00208      .   IF(N1.LT.2)N1=2
00209      .   IF(N2.GT.NOPTS-1)N2=NOPTS-1
00210      .   IF(NDT1.GT.(N1-1))NDT1=N1-1
00211      .   IF(NDT2.GT.(NOPTS-N2))NDT2=NOPTS-N2
00212      .   IF(NDT.LT.MAX(NDT1,NDT2))NDT=MAX(NDT1,NDT2)
00213      .   K1=N1+NDT-NDT1
00214      .   K2=N2-NDT+NDT2
00215      .   WRITE(6,*) 'CORRELATE LEFT END? 0=NO 1=YES'
00216      .   READ(6,*) NFIX
00217      ...FIN
00218  C      CONTROLS TYPE OF DERIVATIVE (SYMMETRIC OR RIGHT-HAND)
00219      WRITE(6,*) 'SELECT A DERIVATIVE METHOD(0)HALF (1)FULL'
00220      READ(6,*) NTYPE
00221  C

```

```

00222 C      SMOOTH THE INTEGRAL OF CONCENTRATION
00223          IRPEAT=0
00224          REPEAT UNTIL (IRPEAT,GE,NCOUNT)
00225          .   IRPEAT=IRPEAT+1
00226          .   WHEN(NEND,EG,1)
00227          .   .   IF(NDT,GE,1)
00228          .   .   .   DO (K=1,NDT)
00229          .   .   .   .   C(K)=CN(1)
00230          .   .   .   .   C(NOPTS+NDT+K)=CN(NOPTS)
00231          .   .   .   ...FIN
00232          .   .   ...FIN
00233          .   .   DO (K=1,NOPTS) C(NDT+K)=CN(K)
00234          .   .   ...FIN
00235          .   ELSE
00236          .   .   DO(K=1,NOPTS) C(K)=CN(K)
00237          .   .   ...FIN
00238          .   CO=0.
00239          .   CSAVE=0.
00240          .   DO (K=1,NOPTS)
00241          .   .   JOVER=1
00242          .   .   IFLAG=0
00243          .   .   WHEN(NEND,EG,1)
00244          .   .   .   JMN=K
00245          .   .   .   JMX=K+2*NDT
00246          .   .   .   ...FIN
00247          .   .   ELSE
00248          .   .   .   CONDITIONAL
00249          .   .   .   .   (K,LT,N1)
00250          .   .   .   .   .   JMN=MAX(1,K-NDT1)
00251          .   .   .   .   .   JMX=MIN(NOPTS,K+NDT1)
00252          .   .   .   .   .   IF(JMN,EG,1) IFLAG=1
00253          .   .   .   .   .   ...FIN
00254          .   .   .   .   (K,GT,N2)
00255          .   .   .   .   .   JMN=MAX(1,K-NDT2)
00256          .   .   .   .   .   JMX=MIN(NOPTS,K+NDT2)
00257          .   .   .   .   .   IF(JMX,EG,NOPTS) IFLAG=1
00258          .   .   .   .   .   ...FIN
00259          .   .   .   .   (OTHERWISE)
00260          .   .   .   .   .   KMN=MAX(N1-NDT1,K-NDT)
00261          .   .   .   .   .   KMX=MIN(N2+NDT2,K+NDT)
00262          .   .   .   .   .   KINC=MIN(K-KMN,KMX-K)
00263          .   .   .   .   .   JMN=K-KINC
00264          .   .   .   .   .   JMX=K+KINC
00265          .   .   .   .   .   ...FIN
00266          .   .   .   .   .   ...FIN
00267          .   .   .   .   IF(K,GE,N1,AND,K,LE,K1) JOVER=2
00268          .   .   .   .   IF(K,LE,N2,AND,K,GE,K2) JOVER=2
00269          .   .   .   ...FIN
00270 C      .   AVERAGE AND DIFFERENCE THE CONCENTRATION INTEGRAL
00271          .   .   DO(L=1,JOVER)
00272          .   .   .   CSAVE1=CN(K)
00273          .   .   .   CN(K)=0.
00274          .   .   .   NSIZE=0
00275          .   .   .   DO (J=JMN,JMX)
00276          .   .   .   .   NSIZE=NSIZE+1
00277          .   .   .   .   CN(K)=CN(K)+C(J)

```

```

00278      . . . . IF(IFLAG,EQ,1)
00279      . . . . . TN(NSIZE)=T(J)
00280      . . . . . CC(NSIZE)=C(J)
00281      . . . . . ...FIN
00282      . . . . . ...FIN
00283      . . . . CN(K)=CN(K)/NSIZE
00284      C      . . . . USES A REGRESSION VALUE AT ENDS--EQUIVALENT IN INTERIOR
00285      . . . . IF(IFLAG,EQ,1)
00286      . . . . . IF(NSIZE,GT,1)CALL REGRES(TN,CC,NSIZE,SLOPE,T(K),CN(K))
00287      . . . . . ...FIN
00288      . . . . JMN=JMN-1
00289      . . . . JMX=JMX-1
00290      . . . . . ...FIN
00291      . . . . CSAVE0=CN(K)
00292      C      . . . . CORRELATES END VALUES
00293      . . . . IF(NFIX,EQ,1)
00294      . . . . . WHEN(JOVER,EQ,1) C(K)=CSAVE0
00295      . . . . . ELSE C(K)=CSAVE1
00296      . . . . . ...FIN
00297      . . . . WHEN(JOVER,EQ,1)
00298      . . . . . IF(CSAVE0,LT,CSAVE)CSAVE0=CSAVE+C(K)-C0
00299      . . . . . CN(K)=CSAVE0-CSAVE
00300      . . . . . CSAVE=CSAVE0
00301      . . . . . ...FIN
00302      . . . . ELSE
00303      . . . . . CN(K)=CSAVE1-CSAVE0
00304      . . . . . CSAVE=CSAVE1
00305      . . . . . ...FIN
00306      . . . . C0=C(K)
00307      . . . . . ...FIN
00308      C      . . . . COMPUTES SYMMETRIC DERIVATIVE OF CONCENTRATION INTEGRAL
00309      . . . . IF(NTYPE,EQ,1)
00310      . . . . . DO(K=1,NOPTS-1) CN(K)=(CN(K)+CN(K+1))/2.
00311      . . . . . ...FIN
00312      . . . . IF(IRPEAT,LT,NCOUNT)
00313      . . . . . CINV=0.
00314      . . . . . DO(K=1,NOPTS)
00315      . . . . . . CINV=CINV+CN(K)
00316      . . . . . CN(K)=CINV
00317      . . . . . ...FIN
00318      . . . . . ...FIN
00319      . . . . . ...FIN
00320      C      . . . . SET THE INITIAL CONCENTRATION
00321      C      . . . . CN(1)=CN(2)
00322      . . . . ISFAC=NDT
00323      . . . . CINV=0.
00324      . . . . CRMAX=0.
00325      . . . . CMAXM=0.
00326      . . . . TCMAXM=0.
00327      . . . . INTEGRATE THE FINAL CONCENTRATION GRAPH
00328      C      . . . . DO (K=1,NOPTS)
00329      . . . . . CINV=CINV+CN(K)
00330      . . . . . IF(CMAXM,LT,CN(K))
00331      . . . . . . CMAXM=CN(K)
00332      . . . . . TCMAXM=T(K)
00333      . . . . .

```



```

00334      . ...FIN
00335      ...FIN
00336      CINV=CINV*DT
00337      CRMAX=CMAXM
00338      CONDITIONAL
00339      . (IKINDP.EQ.0) CMAXM=CMAXM/TFLW(IFT)
00340      . (IKINDP.EQ.1)
00341      . . CINV=CINV*TFLW(IFT)
00342      . . CRMAX=CRMAX*TFLW(IFT)
00343      . ...FIN
00344      ...FIN
00345      WRITE(6,*) 'C INV=',CINV,'C INV0=',CINV0,'TOTAL WT=',TOTWT
00346  C      PLOT OUT TO TG: TO CHECK PLOT
00347      IOVER=1
00348      IFINPS=0
00349      CALL GRAPH(T,CN,NOPTS)
00350      IOVER=0
00351      CALL PLOTND
00352      WRITE(6,*) 'IS THE TG: GRAPH OK? (Y= 1, N= 0)'
00353      READ(6,*) IOK
00354      IF (IOK.EQ.0) GO TO 589
00355      IF (IOK.EQ.1)
00356  603      . CONTINUE
00357      . CLOSE(UNIT=4)
00358      . SELECT (IDEV)
00359      . . (1)
00360      . . . SELECT(IUN)
00361      . . . . (0) CALL ASNLUN(4,'DB',0)
00362      . . . . (1) CALL ASNLUN(4,'DB',1)
00363      . . . . (2) CALL ASNLUN(4,'DB',2)
00364      . . . . (3) CALL ASNLUN(4,'MT',0)
00365      . . . . (4) CALL ASNLUN(4,'MT',1)
00366      . . . ...FIN
00367      . . ...FIN
00368      . . (2) CALL ASNLUN(4,'XY',0)
00369      . . (3) CALL ASNLUN(4,'TG',0)
00370      . ...FIN
00371  C      . PLOT OUT HYDROGRAPH
00372  C      .
00373      . IOVER = 0
00374      . XTOTAL = XTOT
00375      . YTOTAL = YTOT
00376      . WRITE(6,*) 'DO YOU WANT RAW DATA PLOTTED? 1=YES, 0=NO'
00377      . READ(6,*) IOVR
00378      . WRITE(6,*) 'DO YOU WANT GETOUT SOLUTION PLOTTED? 1=YES, 0=NO'
00379      . READ(6,*) IGTOUT
00380      . IF(IOVR.EQ.1)
00381      . . IFINPS=1
00382      . . NOPTS = NVALL
00383  C      . . PREPARE GRAPH POINTS
00384      . . DO (IJK=1,NOPTS)
00385      . . . C(IJK) = CT(2,IJK)
00386      . . . T(IJK) = CT(1,IJK)
00387      . . ...FIN
00388      . . CALL GRAPH(T,C,NOPTS)
00389      . . IOVER=1

```

```
00390      . . IFINPS=0
00391      . ...FIN
00392      . IF(IGTOUT.EQ.1)
00393      . . IFINPS=1
00394      . . WRITE(6,*) ' ENTER GETOUT FILE-Q FILENAME FOR COMPARISON>'
00395      . . READ(6,1021)GETOUT
00396 1021    . . FORMAT(9A1)
00397      . . CALL DPFILE(1,GETOUT,10,,IGET)
00398      . ...FIN
00399      . CALL LISTEM(T,CN,NOPTS,CINV,IFP,IKINDP)
00400      . CALL GRAPH(T,CN,NOPTS)
00401      . IF(IGTOUT.EQ.1)
00402      . . IFINPS=0
00403      . . IOVER=1
00404      . . CALL DPR(IGET,0.,T,1000.)
00405      . . CALL DPR(IGET,5.,C,1000.)
00406      . . CALL GRAPH(T,C,500)
00407      . ...FIN
00408      . IOVER=0
00409      . IFINPS=0
00410      . CALL PLOTND
00411      . XTOT =XTOTAL
00412      . YTOT =YTOTAL
00413      . ...FIN
00414      . WRITE(6,*) 'WRITE THE DOSE FILE FOR'
00415      . WRITE(6,1800)(TNAME(KK,IFP),KK=1,9)
00416 1800    . FORMAT(1X,9A1)
00417      . WRITE(6,*) '1 = YES  0 = NO'
00418      . READ(6,*) IWCH
00419      . IF(IWCH.EQ.1)
00420      . . NNPTS = NPTS
00421      . . CALL DPW(ICMTR2,START2,NNPTS,2004.)
00422      . . START2=START2+8.
00423      . ...FIN
00424      . RETURN
00425      . END
```

(FLECS VERSION 22.46)

APPENDIX C

ANALYTICAL SOLUTION FOR COMPARISONS

APPENDIX C

ANALYTICAL SOLUTION FOR COMPARISONS

DECAY STEP RELEASE EQUATIONS

$$N_{1,s} = F_1(1)$$

$$N_{2,s} = F_1(2)$$

$$\begin{aligned} & + F_2(1,2) F_3(1,2) F_4(1) \left[F_5^-(1,2) F_6^-(1,2) + F_5^+(1,2) F_6^+(1,2) \right] \\ & + F_2(1,2) \left[F_7(1,2) F_4(1) F_8(1) - F_4(2) F_8(2) \right. \\ & \quad \left. + F_7(1,2) F_4(1) F_9(1) - F_4(2) F_9(2) \right] \\ & + F_2(1,2) F_7(1,2) F_{10}(1,2) \left[F_{11}(1,2) \left(F_{12}^-(1,2,2,1) - F_{12}^-(1,2,1,2) \right) \right. \\ & \quad \left. + F_{11}^+(1,2) \left(F_{12}^+(1,2,2,1) - F_{12}^+(1,2,1,2) \right) \right] \end{aligned}$$

where:

$$F_1(1) = \frac{N_i^0}{2T} \exp(-R_i \Theta) \left[\operatorname{erfc} \left(\frac{n}{2} \sqrt{\frac{K_i P}{\Theta}} - \sqrt{\frac{p\Theta}{4K_i}} \right) + \exp(pn) \operatorname{erfc} \left(\frac{n}{2} \sqrt{\frac{K_i P}{\Theta}} + \sqrt{\frac{p\Theta}{4K_i}} \right) \right]$$

$$F_2(i,j) = \frac{N_i^0}{2T} \frac{R_i}{R_j - R_i}$$

$$F_3(i,j) = 1 - \frac{K_i}{K_j}$$

$$F_4(i) = \exp(-R_i \Theta)$$

$$F_5(i,j) = \exp \left[\frac{P}{2} \left(1 \pm \sqrt{1 + \frac{4K_j(R_j - R_i)}{P}} \right) \right]$$

$$F_6(i,j) = \operatorname{erfc} \left[\frac{\eta}{2} \sqrt{\frac{K_j P}{\Theta}} \pm \sqrt{\left(\frac{P}{4K_j} + R_j - R_i \right) \Theta} \right]$$

$$F_7(i,j) = \frac{K_i}{K_j}$$

$$F_8(i) = \operatorname{erfc} \left(\frac{\eta}{2} \sqrt{\frac{K_i P}{\Theta}} - \sqrt{\frac{P\Theta}{4K_i}} \right)$$

$$F_9(i) = \exp(P\eta) \operatorname{erfc} \left(\frac{\eta}{2} \sqrt{\frac{K_i P}{\Theta}} + \sqrt{\frac{P\Theta}{4K_i}} \right)$$

$$F_{10}(i,j) = \exp \left(- \frac{K_i R_i - K_j R_j}{K_i - K_j} \Theta \right)$$

$$F_{11}(i,j) = \exp \left[\frac{P\eta}{2} \left(1 \pm \sqrt{1 + \frac{4K_i K_j (R_j - R_i)}{P(K_i - K_j)}} \right) \right]$$

$$F_{12}(i,j,k,\ell) = \operatorname{erfc} \left[\frac{\eta}{2} \sqrt{\frac{K_k P}{\Theta}} \pm \sqrt{\left(\frac{P}{4K_k} + \frac{K_\ell (R_j - R_i)}{K_i - K_j} \right) \Theta} \right]$$

DECAYING BAND RELEASE EQUATIONS

$$N_{1,b} = N_{1,s}(\Theta) - N_{1,s} \left(\Theta - \frac{TV}{L} \right)$$

$$N_{2,b} = N_{2,s}(\Theta) - N_{2,s} \left(\Theta - \frac{TV}{L} \right)$$

TABLE C.1. Boundary Conditions for Solution of Equation (32)

Decaying Step	$N_1 = 0$	$N_1 = \frac{N_1^0}{T} e^{-\lambda_1 t}$	$N_1 = 0$
	$N_2 = 0$	$N_2 = \frac{N_2^0}{T} e^{-\lambda_2 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1} \frac{N_1^0}{T} (e^{-\lambda_1 t} + e^{-\lambda_2 t})$	$N_2 = 0 \quad N_2 = \text{finite}$
	$N_i = 0$	$N_i = \lambda_1 \lambda_2 \dots \lambda_{i-1} \frac{N_i^0}{T} \sum_{j=1}^i \frac{e^{-\lambda_j t}}{R \prod_{j=1}^i (\lambda_K - \lambda_j)}$ $+ \lambda_2 \lambda_3 \dots \lambda_{i-1} \frac{N_2^0}{T} \sum_{j=1}^i \frac{e^{-\lambda_j t}}{R \prod_{j=1}^i (\lambda_K - \lambda_j)}$ $+ \frac{N_i^0}{T} e^{-\lambda_i t}$	$N_i = 0 \quad N_i = \text{finite}$

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

UNITED STATES

A. A. Churm
DOE Chicago Patent Group
9800 South Cass Avenue
Argonne, IL 60439

27 DOE Technical Information
Center

2 Argonne National Laboratory
Reference Library
9800 South Cass Avenue
Argonne, IL 60439

20 Battelle Memorial Institute
Office of Nuclear Waste
Isolation
Attn: Beverly Rawles
505 King Avenue
Columbus, OH 43201

Albin Brandstetter
Office of Nuclear Waste
Isolation
505 King Avenue
Columbus, OH 43201

John Bredehoeft
U.S. Geological Survey
Reston, VA 22092

John Bird
Geology Department
Cornell University
Ithaca, NY 14853

2 Brookhaven National Laboratory
Reference Section
Information Division
Upton, Long Island, NY 11973

Wayne A. Carbiener
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Neville G. W. Cook
Dept of Materials Science and
Mineral Engineering
Hearst Mining Building
University of California
Berkeley, CA 94720

Carl R. Cooley
DOE Office of Nuclear Waste
Management
Washington, DC 20545

Mike Cullingford
Probabilistic Analysis Staff
Nuclear Regulatory Commission
Mail Stop 3103
Washington, DC 20555

Dan Egan
Environmental Protection Agency
Office of Radiation Programs
Technology Assessment Division
M/S AW559
Washington, DC 20460

Robert M. Garrells
Dept of Geologic Sciences
Northwestern University
Evanston, IL 60201

Mark A. Harwell
P.O. Box 667
Cannon Beach, OR 97110

Colin A. Heath
DOE Office of Nuclear Waste
Management
Washington, DC 20545

No. of
Copies

William M. Hewitt
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Peter L. Hofmann
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Muzaffer Kehnemuyi
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

John F. Kircher
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Ron B. Lantz
INTERA Environmental
Consultants, Inc.
11511 Katy Freeway, Suite 630
Houston, TX 77079

- 2 Lawrence Berkeley Laboratory
Reference Library
University of California
Berkeley, CA 94720

Stan E. Logan
Los Alamos Technical
Associates, Inc.
P.O. Box 410, 1650 Trinity Drive
Los Alamos, NM 87544

No. of
Copies

- 2 Lawrence Livermore Laboratory
Reference Library
P.O. Box 808
Livermore, CA 94550

- 2 Los Alamos Scientific Laboratory
Reference Library
P.O. Box 1663
Los Alamos, NM 87544

J. B. Martin
Asst. Director for Radioactive
Waste Management Branch
NRC Division of Materials and
Fuel Cycle Facility Licensing
Washington, DC 20555

John T. McGinnis
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

- 2 Barry N. Naft
NUS Corporation
4 Research Place
Rockville, MD 20805

- 2 J. O. Neff
Department of Energy
Columbus Program Office
505 King Avenue
Columbus, OH 43201

- 2 Neil A. Norman
Environmental Sciences
Department
Bechtel National Inc.
P.O. Box 3965
San Francisco, CA 94105

- 2 Oak Ridge National Laboratory
Central Research Library
Document Reference Section
Oak Ridge, TN 37830

No. of
Copies

Frank L. Parker
Dept. of Environmental Eng.
Vanderbilt University
Nashville, TN 37235

George F. Pinder
Dept. of Civil Engineering
Princeton University
Princeton, NJ 08540

H. L. Ragsdale
Department of Biology
Emory University
Atlanta, GA 30322

- 2 Savannah River Laboratory
Reference Library
Aiken, SC 29801

Arthur J. Soinski
Energy Resources Conservation
and Development Commission
1111 Howe Avenue
Sacramento, CA 95825

Howard P. Stephens
Sandia Laboratories
P.O. Box 5800
Albuquerque, NM 87115

David B. Stewart
U.S. Geological Survey
National Center 959
Reston, VA 22092

Robert Williams
Electric Power Research Inst.
3412 Hillview Avenue
P.O. Box 10412
Palo Alto, CA 94304

Paul A. Witherspoon
Earth Sciences Division
Lawrence Berkeley Laboratory
Building 90
Berkeley, CA 94720

No. of
Copies

FOREIGN

D'Allessandro Avogadro
Commission of European
Communities
Joint Research Centre
I-21020 Ispra (Varese)
ITALY

V. K. Barwell
Environmental Research Branch
Atomic Energy of Canada Limited
Chalk River, Ontario K0J1J0
CANADA

Center for Atomic Energy
Documentation (ZAED)
ATTN: Dr. Bell
Postfach 3640
D-7500 Karlsruhe
F.R. OF GERMANY

Ferruccio Gera
Radiation Protection and
Waste Management Division
Nuclear Energy Agency/OECD
38 boulevard Suchet
F-75016 Paris
FRANCE

K. H. Hubenthal
Bundesministerium fur Forschung
und Technologie
Stressemannstrasse 2
Postfach 200706
D-5300 Bonn
F.R. of GERMANY

- 2 INIS Clearinghouse
International Atomic Energy
Agency
Karntner Ring II
P.O. Box 100
1400 Vienna
AUSTRIA

No. of
Copies

Klaus Kuhn
Institut fur Tiefagerung
Wissenschaftliche Abteilung
Berliner Strasse 2
D-3392 Clausthal - Zellerfeld
F.R. OF GERMANY

Hans W. Levi
Hahn-Meitner-Institut fur
Kernforschung
Glienicker Strasse 100
D-1000 Berlin 39
F.R. OF GERMANY

Library
Studsvik Energiteknik AB
S-611 01 Nykoping
SWEDEN

Glenn Merrett
Storage and Disposal Branch
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba ROE ILQ
CANADA

Franz Peter Oesterle
Physikalisch-Chemische
Bundesanstalt
Bundesallee 100
D-3300 Braunschweig
F.R. OF GERMANY

Tonis Papp
Karnbranslesakerhet
Fack. 10240
Stockholm
SWEDEN

Egbert Schapermeier
Battelle-Institute e.V.
Am Romerhof 35
D-6000 Frankfurt am Main 90
F.R. of GERMANY

No. of
Copies

Tjalle Vandergraff
Atomic Energy of Canada Limited
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba ROE ILQ
CANADA

ONSITE

5 DOE Richland Operations Office

O. J. Elgert
R. B. Goranson
H. E. Ransom
J. J. Schreiber
F. R. Standerfer

8 Rockwell Hanford Operations

R. G. Baca
R. L. Bielefeld
D. J. Carrell
R. A. Deju
F. A. Deluca
R. E. Gephart
L. S. Leonhart
Rockwell Document Control

69 Pacific Northwest Laboratory

F. H. Dove (15)
G. L. Benson
F. W. Bond
D. L. Bradley
D. B. Cearlock
M. O. Cloninger
C. R. Cole (5)
W. J. Deutsch
F. H. Dove
S. K. Gupta
J. W. Lindberg
M. R. Kreiter
R. W. Nelson
G. M. Petrie

Pacific Northwest Laboratory (contd)

A. M. Platt
A. E. Reisenauer
R. J. Serne
D. J. Silveria
C. S. Simmons (5)
J. K. Soldat
J. F. Washburn (10)
Technical Information
Library (5)
Publishing Coordination (SH)(2)
Water and Land Resources Dept.
Library (10)