1

of

t

2

**DOE/NV/10630-63
UC-721**

**CASCADER: AN M-CHAIN GAS-PHASE RADIONUCLIDE
TRANSPORT AND FATE MODEL**

VOLUME 4 - USER'S GUIDE TO CASCADR9

BY

**DAVID E. CAWLFIELD
DUDLEY F. EMER
F. TOM LINDSTROM
GREGORY J. SHOTT**

**TECHNOLOGY SUPPORT SERVICE
WASTE OPERATIONS DEPARTMENT**

SEPTEMBER 1993

**WORK PERFORMED UNDER
CONTRACT NO. DE-AC08-89NV10630**

**PREPARED FOR
U.S. DEPARTMENT OF ENERGY
NEVADA OPERATIONS OFFICE**

**PREPARED BY
REYNOLDS ELECTRICAL & ENGINEERING CO., INC.
POST OFFICE BOX 98521
LAS VEGAS, NEVADA 89193-8521**

MASTER

JP

ABSTRACT

Chemicals and radionuclides move either in the gas-phase, liquid-phase, or both phases in soils. They may be acted upon by either biological or abiotic processes through advection and/or dispersion. Additionally during the transport of parent and daughter radionuclides in soil, radionuclide decay may occur.

This version of CASCADER called CASCADR9 starts with the concepts presented in volumes one and three of this series. For a proper understanding of how the model works, the reader should read volume one first. Also presented in this volume is a set of realistic scenarios for buried sources of radon gas, and the input and output file structure for CASCADER9.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
ABSTRACT	iii
FIGURES	v
TABLES	vii
ACKNOWLEDGEMENT	viii
1.0 INTRODUCTION	1
2.0 OVERVIEW OF CASCADR9	2
3.0 INPUT FILE STRUCTURE	5
3.1 THE BAROMETRIC PRESSURE DATA FILE	14
3.2 The Average Dispersion (DSOT) Data File	15
4.0 Output File Structure	17
5.0 EXAMPLE SCENARIOS	21
5.1 Scenario 1 -- Thorium-232/Thorium-230 Only	24
5.2 Scenario 2 -- Thorium-232/Thorium-230 and Natural Uranium Waste ...	32
5.3 Scenario 3 -- Both Manmade Sources Plus Natural Source With Extra 23 Foot Cap	33
6.0 INSTALLING THE PROGRAMS	39
APPENDIX A NOMENCLATURE AND FORTRAN SYMBOL LIST	43
APPENDIX B FORTRAN SOURCE CODE LISTING	47
APPENDIX C CURVATURE AND SMOOTH CUBIC SPLINE CONCEPTS USED IN CASCADR9	86
Introduction	86
Smooth Cubic Spline Method (Press et al., 1986)	86

TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
APPENDIX D	
CHAIN3: AN EXTENDED MULTI-PURPOSE BATEMAN CHAIN MODEL	94
Introduction	94
Balance Equations (Figure D.1)	94
Solution Method	96
APPENDIX E	
BATEMAN CASCADE CHAIN FOR SIX RADIONUCLIDES	109
APPENDIX F	
SUMMARY OF FORTRAN LIST-DIRECTED INPUT	116
REFERENCES	117
MATERIAL WORK IN PROCESS	118
DISTRIBUTION LIST	119

FIGURES

<u>Number</u>	<u>Page</u>
5.0.1 Verticle profile of stacking of thorium-232/thorium-230 waste in dugout end of trench 3	23
5.1.1 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 4 darcies (4.0×10^{-8} cm ²) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is 200×10^{-10} pCi/liter. The numerals on the plots themselves correspond to one, two, three, or four times respectively	26
5.1.2 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 40 darcies (4.0×10^{-7} cm ²) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is 200×10^{-10} pCi/liter. The numerals on the plots themselves correspond to one, two, three, or four times respectively	27
5.1.3 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 200 darcies (2.0×10^{-6} cm ²) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is 200×10^{-10} pCi/liter. The numerals on the plots themselves correspond to one, two, three, or four times respectively	28

FIGURES (Continued)

<u>Number</u>	<u>Page</u>
5.1.4 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 4 darcies (4.0×10^{-8} cm ²) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4)	29
5.1.5 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 40 darcies (4.0×10^{-7} cm ²) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4)	30
5.1.6 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 200 darcies (2.0×10^{-6} cm ²) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4)	31
5.2.1 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 4 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air	34
5.2.2 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 40 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air	35
5.2.3 Plots of the radon-222 surface flux in (pCi/cm ² -sec) using a Kappa of 200 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air	36
5.2.4 Plot of the radon-222 surface flux (top curve) in (pCi/cm ² -sec) using a Kappa of 4 darcies for the DU region and four tiers of mixed thorium waste underlying the DU waste. Surface flux (bottom curve) of radon-222 (pCi/cm ² -sec) using only the naturally occurring background radium-226 source uniformly distributed so that the soil air phase radon-222 concentration at 10 meters depth is 200 (pCi/liter)	37

5.3.1	Plot of the radon-222 surface flux in (pCi/cm ² -sec) using the same conditions as written for Figure 5.2.4 except now an additional 23-foot-thick layer of native soil is placed over the top of the existing 10-foot-thick closure cap	38
C.1	Raw barometric pressure data from the NOAA Weather Station META 5 in Area 5 of NTS	87
C.2	Splined barometric data - 1000 points generated over the first 4 days	88
C.3	Typical example of 40 days of splined real NOAA META 5 barometric pressure data (heavy line). Curvature function used in helping to choose local Δt values for accurate time stepping	89
D.1	General structure of multi-purpose Bateman chain model CHAIN3	95

TABLES

<u>Number</u>	<u>Page</u>
5.0.1 Numerical experiment number: radon-222 surface flux using CASCADR9, realistic barometric pressure $T = 20^{\circ}\text{C}$ (isothermal)	22
5.0.2 Gas phase permeabilities, porosity, and radon-222 diffusion coefficients . . .	24

ACKNOWLEDGEMENT

The authors wish to thank Kenneth B. Been, UNLV graduate student in Computer Science, for aiding in the running and debugging of the many computer runs which were done in developing this software. Michael V. Ekedahl, Senior Systems Analyst at the UNLV National Supercomputing Center for Energy and the Environment, for his assistance both at UNLV and at our site. Also, we say thank you to Mr. Austin Bradley, of the UNLV Computer Science Department, for his help in preparation of the user manual documentation in Sections 2, 3, and 4 of this volume.

Special thanks are given to Herbert Sequera and Juanita Kuhn of Reynolds Electrical & Engineering Co., Inc., Administrative Services Department, Information Products Section. Their tireless efforts in reading, proofing, and graphical arts have made this document possible.

1.0 INTRODUCTION

This report is a companion document to the earlier reports (Volumes 1, 2, and 3) describing the mathematical model CASCADER (Cawlfeld et al., 1993 and Lindstrom et al., 1993). As the title implies, this volume is the user's guide to running the software program (CASCADR9) that is based upon the model. Those readers wishing to understand the basic science upon which the model is built should read Volumes 1 and 3 (Lindstrom et al., 1992). This document will satisfy all demands from those readers only interested in learning how to set up data files and run the computer program. CASCADR9 begins with a brief overview on its mechanics, to include certain connections used in the naming of various input and output files.

2.0 OVERVIEW OF CASCADR9

Upon invocation, the CASCADR9 program will run through one simulation, taking input parameters from the input data file (see Chapter 3), produce several output files (see Chapter 4), and stop. When running this program on a Unix based system, shell scripts can easily be set up, generating several runs in succession. The batch files on a personal computer (PC) serve the same purpose.

The naming of all output files is based on the name of the input file. All files, including the input file, have a name in the form ROOT.EXTENSION. For input files, the EXTENSION must be "dat," and ROOT may be anything that will form a valid filename, not exceeding eight characters in length. Each output file has a unique EXTENSION, using the same ROOT as the input file. For example, if the input file is named "run01.dat," then the output files will be "run01.bin," "run01.020," etc. A word of caution: CASCADR9 will overwrite any pre-existing output files that have the same name. This means that each new input file should have a unique name, if the output files from previous runs are saved.

To invoke CASCADR9, simply enter the program name with one command line argument. That argument is the ROOT of the data file (and all output files). For example, if the data file is "run01.dat," then the command line which invokes the simulation with this data file input is the following:

```
cascadr9 run01
```

CASCADR9 uses an adaptive time stepping mechanism reducing the time step during periods of rapid change and increases during periods of stasis. However, the timestep, δt , is never increased beyond its initial value, which is specified in the data file. The initial step size of δt is taken as an upper limit. It is worthwhile to show some care in finding an appropriate initial δt . If the δt is too low, the program will take longer than necessary to execute, without gaining a significant amount of accuracy. If

the δt is too high, the program will exert much effort in readjusting the timestep, resulting in longer execution times. The δt in the data file should, therefore, be as large as possible, without causing the program to undergo extensive readjustments. This "magic" value of δt can be found simply by trial and error. The simulation can be started, and if extensive δt adjustments are applied, this information will be reported to the terminal and the program can be stopped with CTRL-C. If no adjustments are applied after a minute or two, the δt may be set too low. Occasional adjustments are not a problem; only when adjustments are continuous can they indicate a problem. The "magic" δt value depends primarily on the barometric pressure wave used; however, 60 seconds is probably a good starting value.

The amount of time required for the simulation depends on the average size of the δt used during the course of the run, which depends on the nature of the barometric pressure wave applied. We have found that a typical 40-day simulation takes about 25 minutes on a SUN SPARCstation 2. This value may vary widely, however, with different input parameters.

While the program is executing, several different types of information are sent to the terminal screen (standard output). First, some initial messages are given concerning the reading of the data file; these include a listing of the approximate print out times used for the output files. Once the actual simulation is started, a report is made every time the δt is adjusted up or down. Even when no δt adjustments are made, a periodic report of the current "simulation time" is produced. This is solely to reassure the human user that the simulation is proceeding without problems. The frequency of these time reports is not correlated with the frequency of data reports to the output files.

The remainder of this document is organized as follows. Chapters 3 and 4 discuss the input and output files, respectively, in detail. Chapter 5 describes several sample scenarios that demonstrate the use of the program. Chapter 6 explains program installation. Appendix A provides a cross reference listing of the mathematical

symbols used in the model (see Volume 1) and the FORTRAN variable names used in the program. Appendix B outlines the entire FORTRAN source code listing. Appendix C contains information on the curvature and smooth cubic spline concepts used in CASCADR9. Appendices D and E contain a description of an auxilliary program, CHAIN 3, which is an extended multi-purpose Bateman Chain Model. Appendix F, "Summary of FORTRAN List-Directed Input identifies the input data file format needed.

3.0 INPUT FILE STRUCTURE

This chapter describes in detail the data file required by the CASCADR9 program. The interface between the program and the input file is designed to provide flexibility in the format for data specification, as well as in the data itself. At the same time, the data file is (or should be) "self-documenting," so that users can easily and quickly modify data sets. This document considers a simple data file which illustrates these features.

If the name of the data file is "test.dat," then call "test" the root, and ".dat" the extension. The extension must be ".dat," but the root may be anything that will form a valid file name on the system being used. The root must be specified on the command line when the program is run; e.g.,

```
%cascadr9 test
```

The data file starts with a comment section which may include any useful information, such as the run number or a short description of the motivation behind the run that this data file is for.

```
RUN DATAFILE IS (C9226506.DAT)  TRANSPORT OF (RADON GAS) CHAIN.  
DIFFUSION,ADVECTION,DECAY CHAIN,1ST ORDER IRREV. BINDING., AIRPUMPING  
Realistic Spectrum of Fourier components from 2-yr to 15 min,  
NOAA, NTS Frenchman Flat data.
```

```
Burial depth tests.  Full cap in place.  
Package burial depth is 2 m below a 4 meter cap.  
This gives a package position of 06 thru 11 meters.
```

```
This run has both package and background operational.
```

```
    Note -- Comment lines now end with "$" in column 1  
$
```

This opening comment section can be arbitrarily long; its end is signified by a dollar sign, "\$," in the first position of its last line.

Comments are used extensively in these data files, and it is important to understand how they must be formatted. As it turns out, the opening comment section is not typical. The data file is organized into data fields, each of which contains a small

number of data values. Each field is separated by exactly one comment line. This single comment line will be skipped by the input routine of the program, without regard to the line's contents. Therefore, it is essential that this comment be there, and that it be only one line long. Normally the comment line will contain a brief description of the contents of the data field that follows.

Frequently, data file lists of numbers are required; the format required is standard FORTRAN input. Separators for data values are single commas, white space (spaces, tabs, and newlines), or both; no other separator, such as a semicolon or slash, will be accepted. See Appendix D for more details. The program uses double precision for all floating point numbers, so exponential and double precision notation are used; e.g., 5.0D0. Repeated data values may be specified by the shorthand "n*x," where n is a positive integer and x is the repeated value. This means input the value x, n times.

The first data field is a set of 10 flags be used to control various parameters in the program.

```
NFLAG (Set of Flags):  1  2  3  4  5  6  7  8  9 10
                      0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

The meaning of each flag is as follows.

- 1 - Data Preview Only. Do not run the program; just read in the data file and produce the output file which echoes the input data and shows, for example, the locations of the knots and the locations of the print depths (described below).
- 2 - List System Control Constants to Report. Writes various control parameters to the report file. The written parameters can be found in SUBROUTINE REPT1, in file C9REPT1.FOR. This is done once at the beginning of a run, if requested.

- 3 - List Initial Concentrations to Report. These are at each knot the velocity (VELOC), the concentration of each radionuclide (COLD), and the reversibly bound concentrations of nuclides (STIRO).
- 4 - List Initial Masses to Report. These are free, reversibly sorbed, and irreversibly bound masses.
- 5 - List Initial Velocity & Concentration to Screen.
- 6 - Debug by Listing Intermediate CSTR to Screen. These concentrations will be listed each time around the master computation loop.
- 7 - Read in Barometric Pressure from File. Do not use the Fourier coefficients listed later in the data file. When this option is used, a separate file for the barometric pressure data must be provided. The barometric pressure data file will be described below.
- 8 - Read in the Radionuclide Dispersion Coefficient Table from a File. When this option is selected, a separate file for the average boundary dispersion coefficients must be provided. This file will be described below.
- 9 - Include, as part of the time step determinant, a check on the curvature of the barometric pressure wave. This flag is only meaningful if the pressures are read from the baro.dat file, not when they are generated using the Fourier coefficients.
- 10 - Unused (unassigned).

After the NFLAGS the data file has the following:

LAYER	NODES	NUCS	NOUT
	518	4	400

NLAYER is the number of simulated soil region layers. NODES is the number of "knots" used in the simulation. These are the locations in one-dimensional space at which computations will be performed; more nodes means higher accuracy but longer execution time. NUCS is the number of radionuclides; the sample data assumes four. The program will periodically print the time to the screen, just so the user may monitor progress of the simulation; NOUT is the number of steps the program takes in between time reports.

The next two data fields are for the specification of the locations of the knots.

Spacing is 'A' for Absolute, 'R' for Relative
'R'

ZOFF, then NM1 spacings for DZZ

```

0.0D0
1.0D0 1.5D0 2.0D0 2.5D0 3.0D0 5.0D0 7.5D0 1.0D1 7.5D0 1.0D1
1.0D1 7.5D0 1.0D1 7.5D0 5.0D0 3.0D0 2.5D0 2.0D0 1.5D0 1.0D0
1.0D0 1.5D0 2.0D0 2.5D0 3.0D0 5.0D0 7.5D0 1.0D1 7.5D0 1.0D1
1.0D1 7.5D0 1.0D1 7.5D0 5.0D0 3.0D0 2.5D0 2.0D0 1.5D0 1.0D0
1.0D0 1.5D0 2.0D0 2.5D0 3.0D0 5.0D0 7.5D0 1.0D1 7.5D0 1.0D1
1.0D1 7.5D0 1.0D1 7.5D0 5.0D0 3.0D0 2.5D0 2.0D0 1.5D0 1.0D0
1.0D0 1.5D0 2.0D0 2.5D0 3.0D0 5.0D0 7.5D0 1.0D1 7.5D0 1.0D1
1.0D1 7.5D0 1.0D1 7.5D0 5.0D0 3.0D0 2.5D0 2.0D0 1.5D0 1.0D0
1.0D0 1.5D0 2.0D0 2.5D0 3.0D0 5.0D0 7.5D0 1.0D1
7.5D0 1.0D1 1.0D1 1.5D1 1.5D1 2.0D1
2.0D1
2.0D1
1.5D1 1.0D1 1.0D1 7.5D0 5.0D0 2.5D0 2.5D0 5.0D0
7.5D0 1.0D1 1.0D1 1.5D1 2.0D1
19*2.0D1
2.0D1 1.5D1 1.0D1 5.0D0 5.0D0
1.0D1 1.5D1 2.0D1
180*2.0D1
2.0D1 3.0D1 4.0D1 5.0D1 6.0D1 7.0D1 8.0D1 9.0D1 1.0D2
193*1.0D2

```

The knots can be specified in one of two ways. In "absolute" mode, the actual location of each knot will be specified. In "relative" mode, which is the one used in the example, each knot is specified by a displacement from the previous knot; the first number is the displacement of the first knot from 0.0. All numbers are given in centimeters. It is important to have the same number of nodes listed as expected,

according to the NODES variable discussed above, since this is how many the program will expect to read. Note the shorthand "150*2.0D1," which means "input the value 2.0D1, 150 times."

Spacing for layers: 'A' for absolute, 'R' for relative
'A'

Bottom depth for N-1 Layers; last one is depth of W.T., not listed.

100.0

200.0

300.0

400.0

The data field above specifies the soil region layers. In this example, five regions are specified. The ith number is the bottom depth of the ith region, starting from the surface (top). The bottom of the last layer is, by default, the depth of the water table (the last node). These depths are in centimeters; so the first layer is between the top and one meter down, the second is from one meter down to two meters down, etc. The last layer is from four meters to the last node at the bottom (245.8 meters).

CAUTION: The format for specification of layers is more restrictive than that for other data items in this program. The bottom depth of each layer must be listed on a separate line.

Many of the output reports generated by this program include values for measured quantities, such as radionuclide concentrations, at various depths in the column of nodes. The input file specifies which depths are sampled for output as follows.

NDEP; then 'N' for nodes or 'Z' for actual depths

09 'Z'

Print at these Nodal positions

0.0D0, 0.5D2, 1.5D2, 2.5D2, 3.5D2, 6.0D2, 11.0D2, 2.0D3, 2.44D4

The first number, NDEP, determines how many depths will be sampled. The "Z" means that the sample locations will be specified by supplying actual depths. Instead, the sample locations can be specified by giving node numbers; in that case, the list should contain integers, not real numbers.

```

Initial time: T0
0.0D0
Final time: TIME
3.456D+6
Number of print times: NPRT
240
DT
30.0D0

```

The initial and final times are specified in seconds. NPRT is the number of times the measured values will be printed to the output files. Since these values will likely be plotted against time, NPRT determines the granularity of the resulting graphs; 240 is a reasonable value. DT, in seconds, is the maximum, and default, step size that the simulation will take. In times when conditions are changing rapidly, DT may be reduced to preserve the accuracy of the results, but the step size will never go above the initial DT specified in the input file.

PARTDN	EPS	KAPPA	One row for each layer
2.6	0.37	4.00D-8	
2.6	0.35	4.00D-7	
2.6	0.30	4.00D-6	
2.0	0.137	4.00D-10	
2.6	0.37	4.00D-8	
EPSIN,	EPSOT		
0.37,	1.0		

PARTDN, EPS, and KAPPA are listed for each soil region layer. The *i*th line contains values for each parameter in the *i*th layer. PARTDN is the porous medium particle density, in units of (gm/cm³); EPS is the soil region porosity in (cm³/cm³); and KAPPA is the air permeability in (cm²). EPSIN is the porosity of the water table region, and EPSOT is the porosity of the boundary layer region.

DSO1,	DSO2,	DSO3,	DSOj	
0.10,	0.10,	0.10,	0.10	
DSP1	DSP2	DSP3	DSPj	One row for each layer
0.0225	0.10	0.10	0.10	
0.0207	0.10	0.10	0.10	
0.0164	0.10	0.10	0.10	
5.07D-3	0.10	0.10	0.10	
0.0225	0.10	0.10	0.10	
DSOT1,	DSOT2,	DSOT3,	DSOTj	
1.00,	1.00,	1.00,	1.00	

These data fields contain dispersion coefficients. A coefficient must be provided for each radionuclide; so DSO1 is for the first radionuclide, DSO2 for the second, etc. In

our sample data set, the number of nuclides (NUCS) is four, so that DSOj represents the fourth radionuclide. DSOj is the average dispersion coefficient of the jth radionuclide in the water table region. DSPj is the jth average dispersion coefficient in the various soil layer regions; each row contains the coefficients for each radionuclide, with the first row being the top layer. DSOTj is the jth dispersion coefficient in the boundary layer region. The units are all (cm²/sec). If NFLAG item 8 is set to one (1), these values are ignored but must still be present (section 3.2).

FREV1	FREV2	FREV3	FREVj	One row for each layer
0.10	0.5	0.5	0.5	
0.05	0.5	0.5	0.5	
0.01	0.5	0.5	0.5	
0.50	0.5	0.5	0.5	
0.10	0.5	0.5	0.5	
FIRR1	FIRR2	FIRR3	FIRRj	One row for each layer
0.10	0.5	0.5	0.5	
0.05	0.5	0.5	0.5	
0.01	0.5	0.5	0.5	
0.50	0.5	0.5	0.5	
0.10	0.5	0.5	0.5	

FREVj is the fraction of porous medium solids available for reversible binding of the jth radionuclide; FIRRj is the fraction available for irreversible binding. Again, the ith row gives the values for the ith layer, where the layers are numbered from the surface. These quantities are dimensionless.

KEQ1	KEQ2	KEQ3	KEQj	One row for each layer
0.00	0.0	0.0	0.0	
0.00	0.0	0.0	0.0	
0.00	0.0	0.0	0.0	
0.00	0.0	0.0	0.0	
0.00	0.0	0.0	0.0	

KEQj is the linear equilibrium reversible sorption coefficient for radionuclide j onto the solid particles in each soil region layer. Units are (cm³air/gm-particles).

```

THALF[0..MNUCS]
4.990D+10, 3.327D+5, 1.800D+2, 1.560D+3, 1.140D+3
AM[0..MNUCS]
226.D0, 222.D0, 218.D0, 214.D0, 214.D0
SA[0..MNUCS]
0.989D12, 1.5377D17, 2.768D20, 3.28D19, 4.41D19

```

THALF[j] is the half life for a stationary source of the jth radionuclide; its units are (sec). AM[j] is the atomic mass of the jth radionuclide in (gm); and SA[j] is the specific activity of the jth radionuclide, in (pCi/gm).

LTIRR1, LTIRR2, LTIRR3, LTIRRj; One row for each layer
 0.0, 1.0, 1.0, 1.0
 0.0, 1.0, 1.0, 1.0
 0.0, 1.0, 1.0, 1.0
 0.0, 1.0, 1.0, 1.0
 0.0, 1.0, 1.0, 1.0

LTIRRj is the rate coefficient for irreversible binding of jth radionuclide in each soil region layer. Units are (1/sec).

AREA
 1.0D+4

AREA refers to the total cross-sectional area at source region-soil interface. Units are (cm**2).

COT1, COT2, COT3, COTj
 0.0, 0.0, 0.0, 0.0
 CIN1, CIN2, CIN3, CINj
 1.D-28, 0.0D0, 0.0D0, 0.0D0

COTj and CINj refer to the jth radionuclide concentration in the atmosphere and at the water table, respectively. These are constants, and the units are (gm/cm³).

DXIN
 100.0D0
 DXOUT
 100.0D0

DXIN is the thickness of the water table coupling region in (cm), and DXOUT is the thickness of the boundary layer in (cm).

TAIR, RG, MASSA, GRAV, RHOWAT
 2.93D+2, 8.3143D+7, 29.0D0, 9.8D+2, 1.0D0

TAIR is the kelvin temperature of the air. RG is the gas constant in (erg/mole °C). MASSA is the molecular mass of air in (gm/mole). GRAV is gravitational acceleration

in (cm/sec²). RHOWAT is the density of water at 20 °C, units of (gm/cm³).

```
PAIRO and NPRES
917.0D0, 1
POC'S from 1 to 15
+0.0000D0, +0.0491D0, +0.0285D0, -0.0349D0, +0.0316D0
+0.0122D0, -0.0465D0, +0.0456D0, -0.1103D0, 0.0000D0
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0
POS'S from 1 to 15
-10.000D0, +0.0513D0, +0.0209D0, +0.0166D0, -0.0067D0
-0.0008D0, +0.0821D0, -0.0194D0, +0.1103D0, 0.0000D0
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0
Periods of each component in secs.
3.4560D+5, 1.5768D+7, 3.9420D+6, 8.6400D+5, 3.4560D+5
2.5920D+5, 8.6400D+4, 4.3200D+4, 7.2000D+3, 1.0000D+0
1.0000D+0, 1.0000D+0, 1.0000D+0, 1.0000D+0, 1.0000D+0
```

Atmospheric barometric pressure values for the simulation may be produced by a Fourier sum of sines and cosines containing as many as 15 components -- if NFLAG(7) is set to zero. NPRES is the number of components that will be used. Amplitudes for sine and cosine, as well as periods, must be given for 15 components even if fewer components are used, since the program will try to read 15 components. If fewer than 15 components are used, then the amplitudes and periods for the remaining components will be ignored, but some dummy values must be listed nonetheless. The units for the pressures are millibars (mbar), and the periods are in seconds. PAIRO refers to the "dc", or constant component of the atmospheric barometric pressure, in (mbar). In the example, the barometric pressure used for the simulation will be

```
pressure (t) == 917.0 - 10.0*sin(2*Pi*t/3.456D+5)
```

where t represents the time. If NFLAG item 7 is set to 1, these data items are ignored but must still be present (section 3.1).

```
Number of Source Regions; 'N' for nodes, 'Z' for depths.
01      'Z'
Initial Distribution Data
00      0.0D0 2.44D4 1.0000D-28, 0.0D0
01      6.0D2 11.00D2 1.0000D-28, 4.33017D-18
```

This final pair of data fields describes the locations of the radioactive sources. The first field contains the number of source regions ("01") and a code letter describing

how the locations of these source regions will be set. For code "Z," as in the example, the top and bottom of the source regions will be listed as an absolute depth, in (cm). The program can then internally ascertain which node numbers correspond most closely with these depths. For code "N," the actual node numbers must be given, and these will be the first and last nodes of the source region.

The first column under the comment "Initial Distribution Data" is the region index number. Region 0 must be first and it is the background region for the entire column. The other regions (in the example there is only one) are intended to be the radioactive source regions. The second and third columns are the beginning (top) and ending (bottom) depths for the region, in (cm). In the example the top of the background is 0.0 (i.e., the surface) and the bottom is 244 meters (the water table); in other words, the entire span of the simulation. The source region in the example is from 6 to 11 meters deep. The fourth column contains the initial concentration, in (gm/cm^3), of the first radionuclide in the appropriate source region. In the example the initial concentration in the source region is the same as in the background---the values are not summed. The final column in this example, the last field, is the buried waste source function for the first radionuclide in the region; its units are ($\text{gm radionuclide}/\text{cm}^3 \text{ sec}$).

3.1 THE BAROMETRIC PRESSURE DATA FILE

The Barometric Pressure Data File is required only if NFLAG item 7 is chosen. When NFLAG item 7 is zero, the barometric pressure will be computed by a Fourier series (using the data for the sine and cosine periods and amplitudes provided in the main data set). When NFLAG item 7 is a 1, then an external file of barometric pressures is required.

The file is a list of time and pressure pairs, one line per pair. The times should be in days, and the pressures in millibars. The data pairs are in FORTRAN free form (list-directed) input. Typical MET station data is taken at 15 minute intervals, but the data file does not require use at any specific interval. The barometric pressure data is read

in, and passed through a (natural) cubic splining procedure, so as to provide data to the program at any given time instance.

The time in days (da) given for the barometric pressure data is not necessarily keyed to the beginning of the simulation, except in the following way. Because a splining procedure is used, it is customary to provide a few points (5 to 20 points) at each end of the barometric data to free up the endpoints of the spline. Therefore, the splining procedure adjusts the data so that the first time, which is a whole number of days, is taken as the starting point for the simulation. A sample of a typical Barometric Pressure Data File is given below:

109.1250000	909.10
109.2500000	910.50
109.3750000	910.90
109.5000000	909.70
109.6250000	908.20
109.7500000	907.60
109.8750000	909.50
110.0000000	909.40
110.1250000	909.70
110.2500000	910.70
110.3750000	910.81
110.5000000	909.48
110.6250000	908.60
110.7500000	908.90
110.8750000	910.20
111.0000000	909.80
111.1250000	909.00
111.2500000	909.70
111.3750000	909.50
111.5000000	907.90

3.2 The Average Dispersion (DSOT) Data File

A separate file containing the average dispersion at the boundary layer is required only if NFLAG item 8 is set to one (1). If NFLAG item 8 is zero, the dispersion values DSOT1, DSOT2, etc., given in the main data set will be used. The DSOT values taken from the main data file, when used, are fixed once and for all time.

Often it is desirable to change the average dispersion values on a daily cycle. This is the purpose of the DSOT data file. It is simply a table of 24 values containing average dispersion values (the units are cm^2/sec). The table is arranged with one line

per hour, for twenty-four hours, starting at zero. That is, the table goes from 0 to 23 hours. Each line has a series of dispersion values, one value for each nuclide in the simulation. A typical file of DSOT dispersion values follows:

0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0
0.02D0	0.02D0	0.02D0	0.02D0
0.06D0	0.06D0	0.06D0	0.06D0
0.20D0	0.20D0	0.20D0	0.20D0
0.75D0	0.75D0	0.75D0	0.75D0
0.95D0	0.95D0	0.95D0	0.95D0
1.00D0	1.00D0	1.00D0	1.00D0
1.00D0	1.00D0	1.00D0	1.00D0
1.00D0	1.00D0	1.00D0	1.00D0
1.00D0	1.00D0	1.00D0	1.00D0
1.00D0	1.00D0	1.00D0	1.00D0
0.75D0	0.75D0	0.75D0	0.75D0
0.50D0	0.50D0	0.50D0	0.50D0
0.25D0	0.25D0	0.25D0	0.25D0
0.10D0	0.10D0	0.10D0	0.10D0
0.05D0	0.05D0	0.05D0	0.05D0
0.025D0	0.025D0	0.025D0	0.025D0
0.01D0	0.01D0	0.01D0	0.01D0
0.01D0	0.01D0	0.01D0	0.01D0

4.0 Output File Structure

Each output file generated by CASCADR9 is named with the same root as the input file, and with either the extension ".bin," or an extension of the form ".o2x," where x is 0, 1, 2, etc. For example, if the input file is called "test01.dat" then the output files will be "test01.bin," "test01.o20," "test01.o21," etc. The ".o2x" files are ascii files, while the ".bin" file is a binary file.

The ".o20" output file contains a description of the run. This includes, first, the input file name, date and time of the run, and the initial comment section from the input file. The second main part of the .o20 file is a listing of the approximate print out times, in (sec). Finally, the node depths are listed, along with the displacement between nodes, the node index numbers, and indicators of which nodes are at print depths. The print depth nodes are marked with a '>'. The units on all distances is (cm). This file is primarily useful when developing a data file. If the program is run with NFLAG(1) set to 1, this output file will be generated, and then the program will terminate itself. The output file can then be examined to see if the print out times, nodes, and print depths are in the desired positions.

The ".o21" output file reports the surface flux and cumulative flow of radionuclides at each print out time. The file is column oriented; the first column lists the time in (da), the second column has the surface flux in (pCi/cm²-sec), and the third has the cumulative flow in (pCi).

The remaining ".o2x" output files are each organized into (NDEPTS + 1) columns; the first column is time in days, and the remaining columns contain that file's particular measured quantity at a specific depth, or horizon. The horizons are listed in the output file in the same order in which they are listed in the input file. The barometric pressure (mbar) at the horizons are listed in the ".o22" file; bulk gas velocity (cm/sec) at horizons in the ".o23" file. The remaining ".o2x" files, starting with ".o24", contain the gas-phase concentrations, at horizons, for each radionuclide. The ".o24" file has

concentrations for the first radionuclide, ".o25" has the second, etc. Dimensions for concentrations are (pCi/cm³).

The binary (".bin") file contains all the information in the other output files, but in binary format. This makes it convenient for reading by plotting routines. It also contains some information that cannot be found in the other output files. The ".bin" file starts with an initial section containing some global information for the program. The bulk of the file then consists of periodic outputs of the current state of the simulation during execution (dumps). The number and timing of the dumps is determined by variables NPRT and TEND which are set in the input data file. These dumps correspond to those that generate the .o2x files, but they contain somewhat more information.

The initial section of the .bin file is formatted as follows:

Record	Contents
1	8 character version number
2	Input filename (12 char); date (9 char); time (8 char).
3	6 integers -- the number of spatial knots in the simulation (NODES) the number of radionuclides (NUCS) the number of nodes to be sampled on each dump (NCOL) the number of print times (NPRT) the number of pressures used in the barometric pressure wave generator (NPRES) the number of times the simulation will echo the current "simulated" time to the screen (NOUT)
4	3 double precision values -- the start time of the simulation (T0) the end time of the simulation, in secs (TEND) the default time step, in secs (DT0)
5	First title (comment) line from the data file (80 char)
6	Second title (comment) line from the data file (80 char)
7	Third title (comment) line from the data file (80 char)
8	(NODES - 1) double precision values -- the spacing between consecutive nodes, starting at the surface
9	NODES double precision values the absolute depth of each node

- 10 NCOL double precision values
 the absolute depth of each node to be sampled,
 that is, the output report horizons
- 11 NCOL integers
 the node index numbers of the print depths
- 12 An integer, value zero, followed by the starting time of
 the simulation, in days (double precision)

All distances are in centimeters. There will be NPRT groups of records forming the bulk of the .bin file. Each output group will consist of the following information:

- g1 Integer index number of this group (1, 2, ..., NPRT);
 The current simulation time, in days (DP)
- g2 NCOL double precision values --
 the barometric pressure, in mbar, at each print depth
- g3 NCOL double precision values --
 the average gas flow velocity (seepage velocity) at
 each print depth, in cm/sec.
- g4 NUCS double precision values --
 the surface flux for each radionuclide in
 (pCi/cm²-sec)
- g5 NUCS double precision values --
 the cumulative mass loss from the surface for each
 radionuclide, in (pCi)

For each radionuclide, the following information is then listed:

- g6 NCOL double precision values
 the concentration in specific activity units
 (pCi/gm) at each print depth
- g7 NCOL double precision values
 the irreversibly bound concentration (gm/gm) at each
 print depth
- g8 NCOL double precision values
 the mobile phase mass, in gm, at each print depth
- g9 NCOL double precision values
 the reversibly sorbed mass, in gm, at each print
 depth
- g10 NCOL double precision values
 the irreversibly sorbed mass (gm) at each print
 depth

Here, "for each radionuclide" means that the entire packet of values will be printed for the first radionuclide, then the entire packet for the second radionuclide, etc. That is, records g6 through g10 will be repeated, once for each nuclide.

If any of these output files already exist when the simulation is invoked, they will be overwritten without warning. Care should therefore be taken to ensure that each run of the simulation uses a data file with a unique root.

If one wishes to read the binary output file, the FORTRAN code listed in routines BRITE_INIT and C9BRITE may be used as a model.

5.0 EXAMPLE SCENARIOS

Thorium-232 waste, consist largely of thorium hydroxide and thorium oxides, has been approved for disposal in shallow trenches and pits at the Low-Level Waste (LLW) Radioactive Waste Management Site (RWMS) in Area 5 of the Nevada Test Site (NTS). A sophisticated gas transport model, CASCADR8 (Lindstrom et al., 1992a), was used to simulate the transport and fate of radon-220 from its source of origin nine feet below a closure cap of native soil, through the dry alluvial earth, to its point of release into the atmosphere.

Worst case radon-220 gas-phase concentrations, as well as surface fluxes, were estimated over 400 days (Lindstrom et al., 1992b). The maximum flux was then used in an exposure assessment model to estimate the total annual dose equivalent received by a person residing in a standard 2500-square-foot house with 10-foot walls. It was found that the predicted total annual effective dose equivalent (EDE) is 4×10^{-11} rem per year. The estimated EDE is many orders of magnitude less than the allowable annual limit of 0.025 rem per year for DOE waste disposal operations.

Recently it was learned that the thorium-232 waste stream estimates used by Lindstrom et al., (1992b) while correct chemically, were incorrect radiologically since a significant amount of thorium-230 waste would also be buried along with the thorium-232. A subsequent series of computer simulation runs, based upon the usual shallow land burial procedure, showed that inclusion of the anticipated thorium-230 waste gave radon-222 fluxes ($\text{pCi}/\text{cm}^2\text{-sec}$) in excess of the National Emission Standards for Hazardous Air borne Pollutants (NESHAP) level ($2 \times 10^{-3} \text{ pCi}/\text{cm}^2\text{-sec}$ [Lindstrom et al., 1993]). Five alternative depth of burial computer simulations using CASCADR9 were made. It was found that burying the thorium-232/thorium-230 waste mixture about 13 meters below the land surface brought the worst case radon-222 surface fluxes back under the NESHAP standard.

The Waste Operations Department (WOD) manager made a suggestion to dig out one end of trench 3 (old pit 6) to a depth of 51 feet below land surface and stack the

thorium-232/thorium-230 mixture in the usual "4-tier" stacking arrangement (mixed 55 gallon drums and 4' x 4' x 7' boxes) on the bottom of the "extra deep" portion of trench 3. Then add a ten foot backfill to the dugout portion of the trench to bring it back to the existing 25 foot trench level. Once that is accomplished WOD then wants to emplace their usual waste stream, 17 foot high via a "4-tier stacking" arrangement, of depleted uranium and natural uranium-238 waste (plus secular equilibrium daughters) in the trench, thereby, bringing the top of this "DU" layer to eight feet below grade. Thus, a backfill of eight feet of native alluvial soil would bring the trench back to grade. It was decided to place a 10 foot thick cap of native alluvial soil over the trench. Figure 5.0.1 shows a vertical profile sketch of the suggested burial geometry. From the bottom of the dugout trench to the water table is roughly 725 feet of "bone dry" alluvium (average water potential is 10 bars).

One of the design questions to be answered is the effect of the thorium-232/thorium-230 waste stacking height on the radon-222 surface flux (four possibilities). Another design question concerns the long term effective gas permeability of the 17 foot LLW layer shown in Figure 5.0.1. Three gas permeabilities, designed to bracket the long term effective gas permeability of the LLW layer, were chosen. Thus, a 4 x 3 factorial design of numerical simulations was setup. Table 5.0.1 shows the run number for each combination in the 4 x 3 factorial design.

Table 5.0.1 Numerical experiment number: radon-222 surface flux using CASCADR9, realistic barometric pressure $T = 20^{\circ}\text{C}$ (isothermal).

Stacking Height	$\kappa_{\text{air}}: 4.0 \times 10^{-8} \text{ cm}^2$	$4.0 \times 10^{-7} \text{ cm}^2$	$2.0 \times 10^{-6} \text{ cm}^2$
	$D_s: 0.022 \text{ cm}^2/\text{sec}$	$0.044 \text{ cm}^2/\text{sec}$	$0.055 \text{ cm}^2/\text{sec}$
	$\epsilon: 0.37 \text{ cm}^3/\text{cm}^3$	$0.56 \text{ cm}^3/\text{cm}^3$	$0.67 \text{ cm}^3/\text{cm}^3$
Single layer	1	5	9
Double layer	2	6	10
Triple layer	3	7	11
"4-tier"	4	8	12

Figure 5.0.1 Vertical profile of stacking thorium-232/thorium-230 waste in dugout end of trench 3.

Before computer simulations of the radon-222 transport and fate in the dry alluvial soil can be made, estimates of the porosity and the gas diffusion coefficient of radon-222 must be obtained. These estimates are obtained by using equations 3.1.2 and 3.1.3 on page 20 of Lindstrom et al., (1993b). Table 5.0.2 summarizes this data from which the estimates are obtained via linear interpolation.

Table 5.0.2 Gas phase permeabilities, porosity, and radon-222 diffusion coefficients.

ϵ (porosity, cm^3/cm^3)	κ_{air} (cm^2)	κ_{air} (darcies)	D_s/D_0	D_s (cm^2/sec) (for $D_0 = 0.1 \text{ cm}^2/\text{sec}$)
0.05	5.87×10^{-12}	0.000587	ND*	ND
0.1	1.05×10^{-10}	0.011	0.03162	0.003163
0.2	2.12×10^{-9}	0.212	0.08944	0.008944
0.3	1.40×10^{-8}	1.4	0.16432	0.016432
0.4	6.02×10^{-8}	6.02	0.25298	0.025298
0.5	2.12×10^{-7}	21.2	0.35355	0.035355
0.6	6.86×10^{-7}	68.6	0.46476	0.046476
0.7	2.26×10^{-6}	226.0	0.58566	0.058566
*ND - not determined				

5.1 Scenario 1 – Thorium-232/Thorium-230 Only

Lindstrom et al., (1992b) showed via simulation that the radon-222 which is derived from radium-224 in the thorium-232 chain poses no problem, radiologically, provided the source is at least eight feet below the "land surface." Here land surface means the top of the closure cap. Thus, as per the burial geometry shown in Figure 5.0.1, there is no radiological hazard from radon-220 emissions from the buried thorium-232. Lindstrom et al., (1993a) did show conclusively via simulation that even small amounts of thorium-230, viewed as an impurity in the thorium-232 waste stream, can cause surface fluxes of radon-222 in excess of the NESHAP standard when the top of a "4-tier" waste region is only eight feet below the land surface. Lindstrom et al., (1993a) shows how to estimate the constant in time radium-226 radon gas source (radon-222) for the thorium-230 impurity. Their source rate function based upon peak ingrowth radium-226 value is given as

$$S_{SP} = 5.048 \times 10^{-20} \frac{\text{gm}}{\text{cm}^3 - \text{sec}} \quad (5.1.1)$$

This value provides a conservative estimate since no credit is taken for waste packaging material, which for an unknown length of time, provides some measure of protection. Lindstrom et al., (1992a) provides an estimate of 200 pCi/liter or 0.2 pCi/cm³ for the steady background radon-222 gas concentration repeatedly measured at a 22 foot depth, inside the southeast corner boundary fence of RWMS. To reduce the background effect below the limit of detectability, this number was multiplied by 10⁻¹⁰ (ten order of magnitude reduction factor). That value was used in the 40-day simulation runs shown below as Figures 5.1.1 through 5.1.3. In each of the three figures, the curve labels for 1) is equivalent to a single tier, 2) is equivalent to two tiers, and so forth. In Figures 5.1.1 the LLW region has gas permeability, $\kappa_{air} = 4.0 \times 10^{-8} \text{ cm}^2$ (4 darcies). In Figure 5.1.2 the LLW region has gas permeability, $\kappa_{air} = 40.0 \times 10^{-8} \text{ cm}^2$ (40 darcies). In Figure 5.1.3 the LLW region has gas permeability, $\kappa_{air} = 200.0 \times 10^{-8} \text{ cm}^2$ (200 darcies).

In all 12 numerical experiments the surface flux of radon-222 due to the buried thorium-230 waste is roughly three orders of magnitude or better below NESHAP, even in the case of "4-tier" stacking and the LLW gas permeability set at 200 darcies.

While it is of interest to see the unconfounded radon-222 surface flux caused by the manmade source, we must include the background source of radon-222, which is assumed to be uniformly distributed throughout the entire alluvial column. Thus, another set of 12 numerical experiments was conducted, only this time the previously suppressed naturally occurring radium-226 source was set back to the value $1.3 \times 10^{-18} \text{ gm/cm}^3\text{-sec}$ for radium-226 from the previously used value of $1.3 \times 10^{-28} \text{ gm/cm}^3\text{-sec}$ (Lindstrom et al., 1992a). Figures 5.1.4 through 5.1.6 show the results of running the first 12 simulations over with the naturally occurring background radium-226 source present and operating. Even though there are four radon-222

Figure 5.1.1 Plots of the radon-222 surface flux in ($\text{pCi}/\text{cm}^2\text{-sec}$) using a Kappa of 4 darcies ($4.0 \times 10^{-8} \text{ cm}^2$) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is $200 \times 10^{-10} \text{ pCi}/\text{liter}$. The numerals on the plots themselves correspond to one, two, three, or four times respectively.

Figure 5.1.2 Plots of the radon-222 surface flux in ($\text{pCi}/\text{cm}^2\text{-sec}$) using a Kappa of 40 darcies ($4.0 \times 10^{-7} \text{ cm}^2$) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is $200 \times 10^{-10} \text{ pCi/liter}$. The numerals on the plots themselves correspond to one, two, three, or four times respectively.

Figure 5.1.3 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 200 darcies ($2.0 \times 10^{-6} \text{ cm}^2$) for the DU region and a background radium-226 source concentration set so low that the background soil air concentration of radon-222 is $200 \times 10^{-10} \text{ pCi/liter}$. The numerals on the plots themselves correspond to one, two, three, or four times respectively.

Figure 5.1.4 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 4 darcies (4.0×10^{-8} cm²) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4).

Figure 5.1.5 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 40 darcies (4.0×10^{-7} cm²) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4).

Figure 5.1.6 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 200 darcies ($2.0 \times 10^{-6} \text{ cm}^2$) for the DU region and a background radium-226 source concentration set at its realistic level so that the background soil air concentration of radon-222 is 200 pCi/liter. The plot is the same regardless of the number of thorium waste tiers (1,2,3, or 4).

surface flux curves shown on each of the Figures 5.1.4 through 5.1.6, the effect of adding in the naturally occurring radium-226 source, clearly "swamps out" the much smaller manmade effect due to the more deeply buried thorium-230 waste. Clearly, burying a "4-tier" stack of thorium-230 waste 45 feet below the land surface, results in a worst case surface radon-222 flux, which is about 2,000 times smaller than the NESHAP standard. The effect of the buried waste is lost in the presence of the naturally occurring radium-226.

5.2 Scenario 2 – Thorium-232/Thorium-230 and Natural Uranium Waste

This scenario presents the surface flux of radon-222 as a function of time for the case discussed above in Scenario 1; with the additional manmade source of radium-226 stemming from large amounts of depleted and natural uranium waste being buried in the LLW region shown in Figure 5.0.1. The LLW region containing depleted and natural uranium waste was characterized by Lindstrom et al., (1992c). Using the value for the source function in the LLW region, it was set at

$$S_{SP} = 1.4 \times 10^{-18} \frac{\text{gm}}{\text{cm}^3 - \text{sec}} \quad (5.2.1)$$

This number was derived based upon a charge of DU and natural uranium mix per 4' x 4' x 7' box having 1×10^6 gm of uranium-238 plus its first three daughters in secular equilibrium at time $t=0$. The Bateman chain model, CHAIN3, discussed in Appendix H was used to estimate the ingrowth of radium-226 overtime. The peak value of 0.34 grams of radium-226 for the 1×10^6 grams of uranium-238 and the first three daughters were used. The volume used in estimating the above source function was $112 \text{ ft}^3 (1.0 + 0.07) = 119.84 \text{ ft}^3$, which equates to 3.4 m^3 . Seven percent over the volume is a reasonable estimate of the excess volume, which occurs in stacking boxes of waste with 2 to 3 inches of native soil in between them. The computation and per box basis follows as

$$S_{SP} = \left(\frac{0.34 \text{ gm}}{3.4 \times 10^6 \text{ cm}^3} \right) \left(\frac{0.693}{5.05 \times 10^{10} \text{ sec}} \right) = 1.4 \times 10^{-18} \frac{\text{gm}}{\text{cm}^3 - \text{sec}}$$

Figures 5.2.1 through 5.2.3 show the results of running CASCADR9 for 40 days. The background naturally occurring radium-226 source has been included for realism (200 pCi/liter). Runs 1 through 4 are shown in Figure 5.2.1 as a single line. This is because the stacking of the thorium waste under the LLW was found to play an insignificant role relative to the naturally occurring radium-226 source of radon-222. The effect of having another radon gas source buried over the thorium is quite dramatic. The flux of radon-222 off the surface is clearly above NESHAP. Figures 5.2.2 and 5.2.3 show that the effect of changing the gas permeability in the LLW region is essentially negligible. Figure 5.2.4 shows plots of the surface radon-222 flux for runs 1 through 4 as per Figure 5.2.1 above and the thorium plus natural background radium-226 sources operating. The difference in fluxes is clearly significant.

5.3 Scenario 3 – Both Manmade Sources With Natural Source With Extra 23 Foot Cap

The cost of building deep trenches, which meet the current state safety criterion on wall slope, etc., can become prohibitive. The cost of constructing a berm ("mountain") out of native alluvial material over a filled trench may cost less than digging a deeper trench. However, from the gas flow physics' point of view in one space dimension, whether the trench is deeper or a small mountain is erected over the top it does not matter. Therefore, in the interest of furthering the knowledge of radon gas transport in our dry desert alluvial material, and so that more engineering design data is available, another set of simulations were used. In addition, a layer of native alluvial soil, with gas permeability 4.0 darcies, was added over the previous 10 foot closure cap to make the closure cap 33 feet thick over the trench. The results of runs 1 through 4 (runs 5 through 12 were not made based upon the arguments given in the previous scenario) are shown on Figure 5.3.1. Recall NESHAP requires radon-222 surface flux to be less than 2.0×10^{-3} (pCi/cm²-sec). Also, recall there is always erring on the conservative side by using the 4 million year ingrowth radium-226 value for the LLW region. Using the 10,000 year ingrowth radium-226 value in the LLW source region calculation, the radon-222 surface flux values are reduced by a factor of 15, which puts the maximum radon-222 surface flux well below NESHAP limit.

Figure 5.2.1 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 4 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air.

Figure 5.2.2 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 40 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air.

Figure 5.2.3 Plots of the radon-222 surface flux in (pCi/cm²-sec) using a Kappa of 200 darcies for the DU region and one through four tiers of the mixed thorium waste underlying the DU waste. The realistic background concentration of radon-222 is set at 200 pCi/liter of soil air.

Figure 5.2.4 Plot of the radon-222 surface flux (top curve) in (pCi/cm²-sec) using a Kappa of 4 darcies for the DU region and four tiers of mixed thorium waste underlying the DU waste. Surface flux (bottom curve) of radon-222 (pCi/cm²-sec) using only the naturally occurring background radium-226 source uniformly distributed so that the soil air phase radon-222 concentration at 10 meters depth is 200 (pCi/liter).

Figure 5.3.1 Plot of the radon-222 surface flux in (pCi/cm²-sec) using the same conditions as written for Figure 5.2.4 except now an additional 23-foot-thick layer of native soil is placed over the top of the existing 10-foot-thick closure cap.

6.0 INSTALLING THE PROGRAMS

CASCADR9 has been successfully compiled and run on the following machines and systems:

- IBM Compatible PC running MS-DOS and MS FORTRAN 5.10
- Sun SPARCstation-2 running SunOS 4.1.x and Sun FORTRAN 1.4
- VAX 8550 running VMS and VAX FORTRAN
- Cray Y-MP running UNICOS and cf77
- Intergraph C400 (IP-6450) Workstation using apf77

The source code and associated files for CASCADR9 will be distributed in two types of "archive" files. The IBM-PC version will be a self-extracting archive file named "CAS9ZIP.EXE." When "executed," this archive file will unpack itself to yield the source codes, a PC-386 executable file, sample data sets, and current implementation and installation notes.

For machines which use the Unix operating system, a compressed "tar" file will be provided. This file, named "cascadr9.tar.Z," will contain all of the above mentioned source code, data, etc., except that it will not normally contain an executable binary (as there are too many different systems to support in this fashion). The compressed tar file should first be decompressed using the Unix "uncompress" utility, and then further extracted using the Unix "tar" utility. The unpacked source files may then be compiled on the local host machine (examples are given below).

For those users with access to the Internet, both of the archival file types will be made available via "anonymous ftp." Our Internet access host is currently "nye.nscee.edu" (Domain Name IP Address: 131.216.39.3). The appropriate files may be gotten by

using the file transfer protocol (ftp) to access the host, using "anonymous" as a login name, and one's e-mail address as a password. The files will be stored in the "/pub/reeco" sub-directory. The file transfer mode must be binary.

Alternatively, upon request to the authors, the files may be sent directly on 3" floppy disk. The requested format may be either as a DOS diskette, or a Unix compatible diskette. In most cases, it will be possible to include a compiled binary for the specific machines (above) which we have access to. Contact the Technology Support Service, Waste Operations Department for advice or assistance in porting the codes to other machines.

The source codes required for the CASCADR9 program are:

<u>DOS & Vax</u>	<u>Unix</u>	<u>Include Files (all systems)</u>
CASCADR9.FOR	cascadr9.f	comasi.v09
C9READ1.FOR	c9read1.f	comcons.v09
C9REPT1.FOR	c9rept1.f	comctrl.v09
C9BRITE.FOR	c9brite.f	comgeom.v09
AIRPUMP9.FOR	airpump9.f	comlayr.v09
EVAL9.FOR	eval9.f	comname.v09
THOMAS9.FOR	thomas9.f	comnumr.v09
		compump.v09
		comrxp.v09
		comsize.v09
FUTIL.LIB (DOS only)		

For ease of compilation, all of the FORTRAN source and Include files should be placed in a single directory reserved for the CASCADR9 system. This will allow the use of "wild card" options when compiling on various systems (the "*"s used below). The program may then be compiled and linked, on various systems, as given below --

PC with MS-DOS and MS FORTRAN:

```
f1 /Ox /Fecascadr9 *.for futil.lib
```

Sun SPARCstation:

```
f77 -lv77 -O3 -native -o cascadr9 *.f
```

VAX running VMS:

```
fortran /opt /g_float cascadr9,c9read1,c9rept1
fortran /opt /g_float c9brite,airpump9,thomas9,eval9
link cascadr9,c9read1,c9rept1,c9brite, -
> airpump9,thomas9,eval9
```

Cray Y-MP:

```
cf77 -o cascadr9 -Wf"-dp -ev" *.f
```

Some minor variations may be required when porting between machines within this set. These modifications involve the processing of command line arguments by the program. Subroutine GET_ROOT, in source file "c9read1.f" (or "C9READ1.FOR"), contains a call to system routine GET_ARG. On the Sun and Cray, GET_ARG takes two arguments -- the index number of the command line argument desired, and the character string variable in which to store the argument. Under MS FORTRAN on a PC, GET_ARG requires an additional integer parameter, STATUS, which, for the purposes of this program, can be just a dummy variable -- its value is not used further. The GET_ARG routine is not supported on the VAX, so for this machine the statement "NARGC = IARGC()," in GET_ROOT, should be replaced with "NARGC = 0." The call to GET_ARG should then be commented out, or, alternatively, a dummy GET_ARG subroutine, which does nothing, may be added to the program. Thus modified, the VAX program will prompt for manual input of the ROOT name.

It is recommended that the data and output for CASCADR9 runs be made from a sub-directory other than where the source code is maintained. The reason for this is that the CASCADR9 program generates quite a few output files, which should probably be managed separately from the program codes. If the CASCADR9 program resides in one sub-directory, and the program is to be executed from an alternate (data) sub-directory, there are various ways to "point" to the program, which are system dependent. On most systems, one simply executes the program by giving the fully qualified path name, for example --

% /usr/myhome/casc9/cascadr9 mydata (Unix like)

C:> C:\CASC9\CASCADR9 MYDATA (DOS like)

\$ RUN [MYHOME.CASC9]CASCADR9 MYDATA (VAX like)

Alternatively, the use of the PATH variable on DOS systems, or substitution strings on DOS and VAX, may be of some use.

Questions may be addressed to the authors via electronic mail. Our current e-mail address is "cawlfldm@nye.nscee.edu."

APPENDIX A

NOMENCLATURE AND FORTRAN SYMBOL LIST

Symbol	Definition	Units	FORTTRAN Variable
A	total cross sectional area at source region-soil interface	(cm ²)	AREA
ϵ_{in}	porosity of water table region	(cm ³ /cm ³)	EPSIN
V_{in}	average flow velocity in the water table region	(cm/sec)	VELIN
D_{in}^j	average dispersion coefficient of jth radionuclide in the water table region	(cm ² /sec)	DSO(K)
ϵ	porosity of soil region (homogeneous)	(cm ³ /cm ³)	EPS
V_{air}	average gas flow velocity (seepage velocity) at (x,t) in the soil region	(cm/sec)	VELOC(J)
V_{out}	average gas flow velocity in the boundary layer region exterior to the soil-atmosphere boundary $X = 0$	(cm/sec)	VELOT
D_j	jth average dispersion coefficient in soil region	(cm ² /sec)	DSP(K)
Δx_{in}	thickness of water table coupling region	(cm)	DXIN
Δx_{out}	thickness of boundary layer	(cm)	DXOUT
Δx	spacing between nodal points	(cm)	DZ(J)

Symbol	Definition	Units	FORTTRAN Variable
D_{out}^j	average dispersion coefficient of jth radionuclide in the boundary layer region	(cm ² /sec)	DSOT(K)
C^j	concentration of jth radionuclide in the soil subregion	(gm/cm ³)	COLD(J), CNEW(J)
R^j	linear equilibrium retardation coefficient of jth radionuclide in the soil subregion	(dimensionless)	RET(K,J)
λ_o	decay constant for stationary source	(1/sec)	THALF(0) $\lambda_o = \frac{0.693}{THALF(0)}$
SA_j	specific activity of the jth radionuclide	(pico curies/gm)	SA(K)
M_j	atomic mass of jth radionuclide	(curies/gm)	AM(K)
λ_j	decay coefficient for jth radionuclide	(1/sec)	THALF(K) $\lambda_k = \frac{0.693}{THALF(J)}$
λ_{irr}^j	rate coefficient for irreversible binding of jth radionuclide	(1/sec)	LTIRR(K,J)
ρ_p	porous medium particle density	(gm/cm ³)	PARTDN(J)
f_{revj}	fraction of porous medium solids available for reversible binding of jth radionuclide	(dimensionless)	FREV(K,J)
f_{irj}	fraction of porous medium solids available for irreversible binding of jth radionuclide	(dimensionless)	FIRR(K,J)

Symbol	Definition	Units	FORTTRAN Variable
K_{eqj}	linear equilibrium reversible sorption coefficient for radionuclide j onto the solid particles	$\left(\frac{\text{cm}^3 \text{ air}}{\text{gm particles}} \right)$	KEQ(J)
$S_{SP} (x)$	buried waste source function for first radionuclide	$\left(\frac{\text{gm radionuclide}}{\text{cm}^3 \cdot \text{sec}} \right)$	SSP(I)
κ_{air}	air permeability	(cm^2)	KAPPA(J)
K_{air}	pneumatic conductivity	cm/sec	KAIR(J)
η_{air}	air viscosity	$\frac{\text{dyne} \cdot \text{sec}}{\text{cm}^2}$	ETAIR
ρ_{air}	air density	gm/cm ³	RAIRO
ρ_w	density of water at 20°C	gm/cm ³	RHOWAT
g	gravitational acceleration	(cm/sec ²)	GRAV
ΔP_{oc_i} & ΔP_{os_i}	amplitudes of ith barometric pressure component	cmHg	POC(I) & POS(I)
L_i	ith barometric component period	(sec)	LP(I)
P_{air}	barometric air pressure	dyne/cm ²	PRESSO(J), PRESSN(J)
P_A	average barometric pressure	mb	PAIRO
M_{air}	molecular mass of air	gm/mole	MASSA
R_g	gas constant	$\frac{\text{erg}}{\text{mole} \cdot ^\circ\text{K}}$	RG
T	kelvin temperature	(°K)	TAIR
D_{air}	pneumatic diffusivity	cm ² /sec	DAIR(J)

Symbol	Definition	Units	FORTTRAN Variable
S_1^j	irreversibly bound concentration of radionuclide j in the transport and fate region $[0, L_x]$	(gm/gm)	STIRO(J) & STIRN(J)
N_x	number of equal length subintervals into which $[0, L_x]$ is subdivided	(dimensionless)	NODES
Δt	time increment between adjacent time levels	(sec)	DT
$M_{Tfree\ k, n+1}^j$	mobile phase mass of radionuclide j at nodal point x_k in the transport and fate region at time level t_{n+1}	gm	MTF(J)
$M_{TR\ k, n+1}^j$	reversibly sorbed mass of radionuclide j at nodal point x_k in the transport and fate region at time level t_{n+1}	gm	MTS(J)
$M_{TIk, n+1}^j$	irreversible sorbed mass of radionuclide j at nodal point x_k in the transport and fate region at time level t_{n+1}	gm	MTIRR(J)
$C_{um, n+1}^j$	cumulative mass loss of the j th radionuclide from the surface $x = 0$	gm	CUMFO(J) & CUMFN(J)
ϵ_{out}	porosity of atmospheric boundary layer	cm^3/cm^3	EPSOT
q_{air}	specific discharge of air in the porous medium	$cm^3/cm^2 - sec$	QAIR(J)

APPENDIX B

FORTRAN SOURCE CODE LISTING

```

*COMSIZE
C
C*****
C      Euler-Lagrange time integrated, 4-chain radionuclide, irreversible
C      sorption, advection-dispersion equations - Bateman equations at
C      inlet boundary with mass conserving flux boundary conditions.
C
C      ! Version 9.0 !
C*****
C      COMMON BLOCKS FOR COMSIZE.V09
C*****
C      INTEGER          NZSET,      NPT,      MNUC,      MCOL
C
C      PARAMETER        (NZSET=389, NPT=0360, MNUC=1, MCOL = 16)
C
C      INTEGER          NODES, NM1, NM2, NUCS, NPRES, NFLAG,      NOUT
C      COMMON/DATA19/   NODES, NM1, NM2, NUCS, NPRES, NFLAG(10), NOUT
C
C      INTEGER          NDEPTS, NCOL
C      DOUBLE PRECISION ZDEPTS
C      COMMON/DATA20/   ZDEPTS(MCOL), NDEPTS(MCOL), NCOL
C
C      CHARACTER        DAYT*9, TYME*8, FINP*12
C      COMMON/DATA21/   DAYT, TYME, FINP
C
C      CHARACTER        VERSION*(*)
C      PARAMETER        (VERSION='CASCv9.0')
C

```

```

*COMASI
C
C*****
C      Last Revision:  March 17, 1992
C      Source File:    COMASI.V09
C*****
C      COMMON BLOCKS FOR COMASI.V09
C*****
C
C      DOUBLE PRECISION COT
C      COMMON/DATA9/    COT(MNUC)
C
C      DOUBLE PRECISION DSOT,      EPSOT, VELOT
C      COMMON/DATA29/   DSOT(MNUC), EPSOT, VELOT
C
C      DOUBLE PRECISION QFLXO,      QFLXN,
C      & CUMFO,      CUMFN
C      COMMON/DATA31/   QFLXO(MNUC), QFLXN(MNUC),
C      & CUMFO(MNUC), CUMFN(MNUC)
C

```



```

*COMCONS
C
C*****
C
C          COMMON BLOCKS FOR COMCONS.V09
C*****
C
C      DOUBLE PRECISION  CTSIR,          CTIR1,          CTIR2
COMMON/DATA42/          CTSIR(MNUC), CTIR1(MLAYER,MNUC), CTIR2(MNUC)
C
C      DOUBLE PRECISION  PI, OMEG,          PIO2, RHOWAT
COMMON/DATA47/          PI, OMEG(15), PIO2, RHOWAT
C
C      DOUBLE PRECISION  DAIR,          KAIR,          KADZ
COMMON/DATA47/          DAIR(NZSET), KAIR(NZSET), KADZ(NZSET)
C
C      DOUBLE PRECISION  RATIO
COMMON/DATA48/          RATIO(NZSET,MNUC)
C
C      DOUBLE PRECISION  CONLM
COMMON/DATA53/          CONLM(MLAYER,MNUC)
C

```

```

*COMCTRL
C
C*****
C
C          COMMON BLOCKS FOR COMCTRL.V09
C*****
C
C      INTEGER NPRT
DOUBLE PRECISION T0, DT, DT0, T, TEND, TNEXT, TOUT, TPRT
COMMON/DATA16/      T0, DT, DT0, T, TEND, TNEXT, TOUT, TPRT(NPT),
&      NPRT
C

```

```

*COMGEOM
C
C*****
C
C          COMMON BLOCKS FOR COMGEOM.V09
C*****
C
C      DOUBLE PRECISION  ZOFF, ZKNOT,          DZ
COMMON/DATA5/          ZOFF, ZKNOT(NZSET), DZ(NZSET-1)
C
C      DOUBLE PRECISION  ZSTR
COMMON/DATA17/          ZSTR(NZSET)
C
C      DOUBLE PRECISION  AREA, DXIN, DXOUT, DEPTH
COMMON/DATA18/          AREA, DXIN, DXOUT, DEPTH
C

```

```

*COMLAYR                                         Last Revision:  June 11, 1992
C                                         Source File:  COMSIZE.V09
*****
* contains global declarations necessary for keeping track of layers in
* the soil
* MLAYER: maximum number of layers allowed
* NLayer: actual # of layers needed in this run
* LAYER: array which stores the depth of the bottom border for each
* layer requested by the user
* LDIV: array for the node numbers (indices) of those nodes which are
* closest to the bottom border of each layer; start with zero,
* so that LDIV(0) is always 1 (the first node is the bottom of
* the 0th layer). Actually, these nodes have the properties of
* the layer below, so they can be seen as the top node of the
* next layer.
* JTOK: array which stores the layer number for each node
*****
      INTEGER MLAYER
      PARAMETER (MLAYER = 10)
C
      INTEGER NLayer
      COMMON/LAYRD1/ NLayer
C
      DOUBLE PRECISION LAYER
      COMMON/LAYRD2/ LAYER(0:MLAYER)
C
      INTEGER          LDIV,          JTOK
      COMMON/LAYRD3/ LDIV(0:MLAYER), JTOK(NZSET)
C

```

```

*COMNAME                                         Last Revision:  July 1, 1992
C                                         Source File:  COMNAME.V09
*****
C      COMMON BLOCKS FOR COMNAME.V09
C*****
C      CHARACTER*80  DUMMY, XNAME,      ROOT*63
      COMMON/DATA43/ DUMMY, XNAME(3), ROOT
C
      INTEGER          L12
      COMMON/DATA4A/ L12
C

```

```

*COMNUMR                                         Last Revision:  April 4, 1992
C                                         Source File:  COMNUMR.V09
*****
C      COMMON BLOCKS FOR COMNUMR.V09
C*****
C      DOUBLE PRECISION  CTEMP,          DTEMP,          XNOW
      COMMON/DATA30/  CTEMP(NZSET), DTEMP(NZSET), XNOW(NZSET)
C

```

*COMPUMP

Last Revision: July 29, 1992

Source File: COMPUMP.V09

COMMON BLOCKS FOR COMPUMP.V09 *

C

DOUBLE PRECISION PRESSO, PRESSN, VELOC
COMMON/DATA44/ PRESSO(NZSET), PRESSN(NZSET), VELOC(NZSET)

C

DOUBLE PRECISION KAPPA, TAIR, MASSA, RG, GRAV, ALPHA
COMMON/DATA45/ KAPPA(MLAYER), TAIR, MASSA, RG, GRAV, ALPHA

C

DOUBLE PRECISION PAIR0, RAIR0, POC, POS, LP
COMMON/DATA46/ PAIR0, RAIR0, POC(15), POS(15), LP(15)

C

DOUBLE PRECISION AA, BB, CC, DD
COMMON/DATA48/ AA(NZSET), BB(NZSET), CC(NZSET), DD(NZSET)

C

DOUBLE PRECISION ACON, BCON, ETAAIR
COMMON/DATA49/ ACON, BCON, ETAAIR

C..

DOUBLE PRECISION BWET, TWET, TOFF, DSOT0
COMMON/DATA50/ BWET, TWET, TOFF, DSOT0(0:23,MNUC)

C

*COMRNXP

Last Revision: June 15, 1992

Source File: COMRNXP.V09

COMMON BLOCKS FOR COMRNXP.V09 *

C

DOUBLE PRECISION AW, BW, CW, DW
COMMON/DATA1/ AW(NZSET), BW(NZSET), CW(NZSET), DW(NZSET)

C

DOUBLE PRECISION COLD
COMMON/DATA5/ COLD(NZSET,MNUC)

C

DOUBLE PRECISION CNEW
COMMON/DATA6/ CNEW(NZSET,MNUC)

C

DOUBLE PRECISION CIN
COMMON/DATA8/ CIN(MNUC)

C

DOUBLE PRECISION DSP, DSO, EPSIN, VELIN
COMMON/DATA10/ DSP(MLAYER,MNUC), DSO(MNUC), EPSIN, VELIN

C

DOUBLE PRECISION PARTDN, EPS, VEL
COMMON/DATA12/ PARTDN(MLAYER), EPS(MLAYER), VEL(NZSET)

C

DOUBLE PRECISION THALF, LAM
COMMON/DATA13/ THALF(0:MNUC), LAM(0:MNUC)

C

DOUBLE PRECISION AM, SA
COMMON /DATA14/ AM(0:MNUC), SA(0:MNUC)

C

DOUBLE PRECISION RET, FREV, FIRR, KEQ
COMMON/DATA15/ RET(MLAYER,MNUC), FREV(MLAYER,MNUC),
& FIRR(MLAYER,MNUC), KEQ(MLAYER,MNUC)

C

DOUBLE PRECISION ASOS, BSOS, CSOS, CSS
COMMON/DATA20/ ASOS(NZSET), BSOS(NZSET), CSOS(NZSET), CSS(NZSET)

C

DOUBLE PRECISION COEF

```

COMMON/DATA21/    COEF (NZSET)
C
DOUBLE PRECISION  YN,          YY,          DS1
COMMON/DATA22/    YN (NZSET), YY (NZSET), DS1 (NZSET)
C
DOUBLE PRECISION  CSTR
COMMON/DATA26/    CSTR (NZSET,MNUC)
C
DOUBLE PRECISION  LTIRR
COMMON/DATA33/    LTIRR (MLAYER,MNUC)
C
DOUBLE PRECISION  XLSPM
COMMON/DATA34/    XLSPM (MLAYER,MNUC)
C
DOUBLE PRECISION  MTF,          MTS,
&                 MTIRR
COMMON/DATA36/    MTF (NZSET-1,MNUC), MTS (NZSET-1,MNUC),
&                 MTIRR (NZSET-1,MNUC)
C
DOUBLE PRECISION  STIRO,          STIRN
COMMON/DATA39/    STIRO (NZSET,MNUC), STIRN (NZSET,MNUC)
C
DOUBLE PRECISION  SOURCE,          SSP
COMMON/DATA55/    SOURCE (NZSET), SSP (NZSET)
C

```

*CASCADR9

Last Revision: October 14, 1992
Source File: CASCADR9.FOR

```

C
PROGRAM CASCADR9
INCLUDE 'comsize.v09'
INCLUDE 'comlayr.v09'
INCLUDE 'comgeom.v09'
INCLUDE 'comname.v09'
INCLUDE 'comasi.v09'
INCLUDE 'compump.v09'
INCLUDE 'comrnxp.v09'
INCLUDE 'comctrl.v09'
INCLUDE 'comnumr.v09'
INCLUDE 'comcons.v09'
INTEGER           IPRINT, IREC, NU
INTEGER           I, J, KKK, LU, K
INTEGER           IADTC, IHDTc
DOUBLE PRECISION  YPO, YPO1
DOUBLE PRECISION  DENOM, CON1, TEST, DUMMYZ
DOUBLE PRECISION  DUMCON, COURANT, DUMMYP
LOGICAL           DONE, COURNTOK, COURFAIL, FROZEN
DOUBLE PRECISION  DZD, FDZ
DOUBLE PRECISION  TEMP, MU_T
DOUBLE PRECISION  QAIR
DOUBLE PRECISION  ECON
DOUBLE PRECISION  KVDZP, KVDZM
DOUBLE PRECISION  EDVDZ, EDVDZ1
CHARACTER          FNAME*12
*****
* Euler-Lagrange time integrated, N-chain radionuclide, irreversible
* sorption, advection-dispersion equations - Bateman equations at
* inlet boundary with mass accounting flux boundary conditions.
* Air pumping at the surface drives the advection terms.
*
*                               Version 9.0
*****

```

```

C*****
C This model is the joint effort of: F. T. Lindstrom, D. E. Cawlfeld,
C D. Daffern, M. E. Donahue, D. F. Emer, and Gregory J. Shott
C
C Special Projects Group
C Waste Management Division
C Reynolds Electrical and Engineering Co., Inc.
C P O Box 98521 M/S 501 Las Vegas, NV 89193-8521
C*****
C..
*+-----+
*| The revisions for cascadr9 allow the soil to consist of
*| layers, each having different properties. See the include file
*| comlayr.v09 for declarations needed to process the layers.
*|
*| The algorithm for processing each node sequentially, used
*| several times in this file, is as follows.
*|
*| process the first node;
*| for each layer do begin
*|   for each interior node in the layer do
*|     process the node;
*|   process the border node, if it's not the last one;
*| end;
*| process the last node;
*+-----+
DATA IHDTC/0/, FROZEN/.FALSE./
C..
C..The following is an Arithmetic Statement Function . . .
C..
      FDZ(K) = DZ(K-1) + DZ(K)
C
C CALCULATE CONSTANTS AND SYSTEM GROUP PARAMETERS
C
      WRITE(*,*)
      WRITE(*,*)'Reading initial data . . .'
      WRITE(*,*)
      CALL READ1
      CALL READ2
      CALL REPT2
      CLOSE(10)
C..
      IF (NFLAG(7) .NE. 0) THEN
        PRINT *, 'Using Barometric Data Table'
      END IF
C..
      IF (NFLAG(8) .NE. 0) THEN
        PRINT *, 'Using DSOT Table'
        FNAME = 'dsot.dat'
        OPEN(88, FILE=FNAME, STATUS='OLD')
        CALL READ3
        CLOSE(88)
        OPEN (88, FILE='dsot.chg', STATUS='UNKNOWN')
        DO NU = 1, NUCS
          DSOT(NU) = DSOT0(0,NU)
        END DO
      END IF
C..
      WRITE(*,*)
      WRITE(*,*)'CALCULATING CONSTANTS AND SYSTEM GROUP PARAMETERS'
      WRITE(*,*)
C
      PI = 3.141592654D0
C..
      TEMP = TAIR - 273.0D0

```

```

MU_T = 1.7080423D2 + 0.91243D0*TEMP
&      -1.6867D-2*TEMP**2 + 1.58D-4*TEMP**3
ETAIR = 1.0D-6*MU_T
C..
PAIR0 = PAIR0*1.0D+3
RAIR0 = MASSA*PAIR0 / (RG*TAIR)
C..
C...Compute DAIR and KAIR as a function of Z . . .
C..
ALPHA = MASSA*GRAV / (RG*TAIR)
DO I = 1, NODES
  K = JTOK(I)
  ECON = EXP(ALPHA*ZKNOT(I))
  DAIR(I) = (KAPPA(K)*PAIR0 / (EPS(K)*ETAIR)) * ECON
  KAIR(I) = (KAPPA(K)*RAIR0*GRAV / (EPS(K)*ETAIR)) * ECON
  KADZ(I) = ALPHA * KAIR(I)
END DO
C..
PRESSO(1) = 0.0D0
PRESSN(1) = PRESSO(1)
DO K = 1, NM1
  PRESSO(K+1) = 0.0D0
  PRESSN(K+1) = PRESSO(K+1)
END DO
DT = DT0
C
AA(1) = 0.0D0
BB(1) = 1.0D0
CC(1) = 0.0D0
DO K = 1, NLAYER
  DO I = LDIV(K-1)+1, LDIV(K)-1
    CON1 = 2.0D0*DAIR(I)*DT
    DZD = FDZ(I)
    AA(I) = -CON1 / (DZ(I-1)*DZD)
    BB(I) = 1.0D0 + CON1/(DZ(I-1)*DZ(I)) + DT*KADZ(I)
    CC(I) = -CON1 / (DZ(I)*DZD)
  END DO
  IF (K .LT. NLAYER) THEN
    I = LDIV(K)
    KVDZM = KAPPA(K) / DZ(I-1)
    KVDZP = KAPPA(K+1) / DZ(I)
    AA(I) = -KVDZM
    BB(I) = KVDZM + KVDZP + (KAPPA(K+1) - KAPPA(K))*ALPHA
    CC(I) = -KVDZP
  END IF
END DO
AA(NODES) = -1.0D0
BB(NODES) = 1.0D0 - ALPHA*DZ(NM1)
CC(NODES) = 0.0D0
C
DO I = 1, NPRES
  OMEG(I) = 2.0D0*PI/LP(I)
END DO
DO I = NPRES+1, 15
  OMEG(I) = 0.0D0
END DO
C
VELIN = 0.0D0
C..
CTSIR(1) = (1.0D0 - 0.5D0*DT*LAM(1)) /
&      (1.0D0 + 0.5D0*DT*LAM(1))
CTIR2(1) = 0.0D0
DO K = 1, NLAYER
  RET (K,1) = (1.0D0-EPS(K))*FREV(K,1)*PARTDN(K)*KEQ(K,1)/EPS(K)
  CTIR1(K,1) = (0.5D0*DT)*LTIRR(K,1)

```



```

&          + EPS(1)*VELOC(1)*COLD(1,NU)
&      ELSE
&          QFLXO(NU) = - EPSOT*DSOT(NU)*(COLD(1,NU)-COT(NU))/DXOUT +
&          VELOT*COT(NU)
&      ENDIF
&      QFLXO(NU) = QFLXO(NU) * SA(NU)
C
C      CUMFO(NU) = 0.0D0
C
C      DO J = 2, NODES
C          K = JTOK(J)
C          DZD = DZ(J-1)
C          MTF(J-1,NU) = EPS(K)*DZD*AREA*
&          0.5D0*(COLD(J,NU)+COLD(J-1,NU))
&          MTS(J-1,NU) = (1.0D0-EPS(K))*DZD*AREA*FREV(K,NU)*PARTDN(K)*
&          KEQ(K,NU)*0.5D0*(COLD(J,NU)+COLD(J-1,NU))
&          MTIRR(J-1,NU) = (1.0D0-EPS(K))*DZD*AREA*FIRR(K,NU)*
&          PARTDN(K)*0.5D0*(STIRO(J,NU) + STIRO(J-1,NU))
C      END DO
C      END DO
C
C      T = T0
C      TOUT = T
C
C      COMPUTE THE LU FACTORIZATION FOR THE THOMAS ALGORITHM
C      AS APPLIED TO THE LAGRANGE CUBIC SPLINE-ADVECTION APPROXIMATION.
C
C      ASOS(1) = 0.0D0
C      BSOS(1) = 2.0D0*DZ(1)
C      CSOS(1) = DZ(1)
C
C      DO J = 2, NM1
C          ASOS(J) = DZ(J-1)
C          BSOS(J) = 2.0D0*FDZ(J)
C          CSOS(J) = DZ(J)
C      END DO
C
C      ASOS(NODES) = DZ(NM1)
C      BSOS(NODES) = 2.0D0*DZ(NM1)
C      CSOS(NODES) = 0.0D0
C
C      CSS(1) = CSOS(1)/BSOS(1)
C      DO J = 2, NODES
C          DENOM = BSOS(J)-ASOS(J)*CSS(J-1)
C          CSS(J) = CSOS(J)/DENOM
C      END DO
C..
C      IF (NFLAG(5) .EQ. 1) THEN
C          DO I = 1, NODES
C              WRITE(*,30) ZKNOT(I),
&              (SA(NU)*COLD(I,NU), NU = 1, NUCS)
&          END DO
C      END IF
C..
C      IF (NFLAG(3) .EQ. 1) THEN
C          WRITE(20,*)
C          WRITE(20,4)' The initial velocity and concentration output '
C          WRITE(20,4)' has the following tabular form: '
C          WRITE(20,*)
C          WRITE(20,2024) 'Node,', 'zdepth,', 'Vel(z)',
&          ('COLD(z,', NU, '),', NU = 1, NUCS),
&          ('STIRO(z,', NU, '),', NU = 1, NUCS)
C          2024 FORMAT(1X, A6, A10, A12, 50(A9, I1, A2))
C          DO I = 1, NODES

```



```

        WRITE(20,32) I, ZKNOT(I), VELOC(I),
&          (SA(NU)*COLD(I,NU), NU = 1, NUCS),
&          (STIRO(I,NU), NU = 1, NUCS)
      END DO
    END IF
C..
    IF (NFLAG(4) .EQ. 1) THEN
      WRITE(20,*)
      WRITE(20,4)' The initial free, reversibly sorbed, and
      WRITE(20,4)' irreversibly bound masses have the following'
      WRITE(20,4)' tabular form
      WRITE(20,*)
      WRITE(20,2026) 'zdepth,',
&      ('Mtf(z,', NU, '),',
&      'Mts(z,', NU, '),',
&      'Mtirr(z,', NU, '),', NU = 1, NUCS)
2026  FORMAT(1X, A10, 50(A9, I1, A2))
      WRITE(20,*)
      DO I = 2, NODES
        DUMMYZ = (ZKNOT(I-1) + ZKNOT(I))*0.5D0
        WRITE(20,33) DUMMYZ,
&      (MTF(I-1,NU), MTS(I-1,NU), MTIRR(I-1,NU), NU = 1, NUCS)
      END DO
      WRITE(20,*)
    END IF
C..
C..Stop if initial data preview is requested
C..
    IF (NFLAG(1) .EQ. 1) STOP 'Initial Data Preview (only).'
```

C..

```

    IREC = 0
    CALL BRITE_INIT(IREC)
C
C BEGIN THE MASTER COMPUTATION LOOP.
C
    WRITE(*,*)
    WRITE(*,*)
    WRITE(*,27) 'Time = ', T
    IPRINT = 1
    DONE = .FALSE.
3000 CONTINUE
C..
C..Compute the local pressure field.
C..Compute the seepage velocities at each nodal point.
C..
    IADTC = 1
    COURFAIL = .FALSE.
    COURNTOK = .FALSE.
    DO WHILE (.NOT. COURNTOK)
      CALL AIRPUMP(COURFAIL, IHDT, FROZEN)
      CALL VELCTY(COURANT, COURNTOK, COURFAIL)
      IF (.NOT. COURNTOK) THEN
        DT = 0.5D0*DT
        IADTC = IADTC + IADTC
      END IF
    END DO
C..
C..IHDT is a counter which is used to hold the DT for a number of
C..sweeps after DT has been reduced ("I Hold DT, Courant").
C..AIRPUMP handles the details.
C..Probably should re-implement the stack method here.
C..
    IF (COURFAIL) THEN
      WRITE(*,2010) 'lowered', IADTC, DT, 'Courant #.'
      IF (IHDT .GT. 0) THEN
```

```

      IHDTC = IHDTC + IADTC**2
    ELSE
      IHDTC = IADTC**2
    ENDIF
  END IF
C..
  IF (NFLAG(8) .NE. 0) CALL SET_DSOT
  TNEXT = T + DT
  VELOC(NODES) = 0.0D0
  VEL(NODES) = 0.0D0
  VELOT = (EPS(1)*VELOC(1))/EPSOT
  VELIN = 0.0D0
C
C  FOR EACH NUCLIDE...
C
  DO NU = 1, NUCS
C
C  First go through and compute VEL's and ZSTR's ignoring interface
C  problems. Some of the values will be corrected afterwards.
C
    DO J = 1, NM1
      VEL(J) = VELOC(J) / (1.0D0+RET(JTOK(J),NU))
    END DO
    DO J = 2, NM1
      IF (VEL(J) .GT. 0.0D0) THEN
        BCON = (VEL(J)-VEL(J-1))/DZ(J-1)
        ACON = VEL(J)
        IF (BCON .EQ. 0.0D0) THEN
          ZSTR(J) = ZKNOT(J)-ACON*DT
        ELSE
          ZSTR(J) = ZKNOT(J) + (ACON/BCON)*(DEXP(-BCON*DT)-1)
        ENDIF
      ELSE
        BCON = (VEL(J+1)-VEL(J))/DZ(J)
        ACON = VEL(J)
        IF (BCON .EQ. 0.0D0) THEN
          ZSTR(J) = ZKNOT(J)-ACON*DT
        ELSE
          ZSTR(J) = ZKNOT(J)-ACON*(1.0D0-DEXP(-BCON*DT))/BCON
        ENDIF
      ENDIF
    ENDIF
  END DO
C
C  Now go through and make the necessary changes required by the interfaces.
C
  DO K = 2, NLAYER
    I = LDIV(K-1)
    QAIR = (-KAPPA(K-1)/ETAIR)*
    & ((PRESSN(I)-PRESSN(I-1))/DZ(I-1)) - ALPHA*PRESSN(I))
    IF (VEL(I+1) .GT. 0.0D0) THEN
      BCON = (VEL(I+1) -
      & (QAIR/(EPS(K)*(1.0D0+RET(K,NU))))) / DZ(I)
      ACON = VEL(I+1)
      IF (BCON .EQ. 0.0D0) THEN
        ZSTR(I+1) = ZKNOT(I+1) - ACON*DT
      ELSE
        ZSTR(I+1) = ZKNOT(I) + (ACON/BCON)*(DEXP(-BCON*DT)-1)
      ENDIF
    ENDIF
    IF (VEL(I-1) .LE. 0.0D0) THEN
      BCON = (QAIR/(EPS(K-1)*(1.0D0+RET((K-1),NU))) -
      & VEL(I-1)) / DZ(I-1)
      ACON = VEL(I-1)
      IF (BCON .EQ. 0.0D0) THEN
        ZSTR(I-1) = ZKNOT(I-1) - ACON*DT
      
```

```

                ELSE
                    ZSTR(I-1) = ZKNOT(I-1) -
&                    ACON*(1.0D0-DEXP(-BCON*DT))/BCON
                END IF
            END IF
        END DO

C
C
C    COMPUTE CUBIC SPLINE ESTIMATES OF CONCENTRATIONS AT THE PSTAR POINTS.
C
        DO J = 1, NM1
            YN(J) = (COLD(J+1,NU)-COLD(J,NU))/DZ(J)
        END DO

C
        YPO = - (2.0D0*DZ(1)+DZ(2))*DZ(2)*COLD(1,NU)
&            + (      DZ(1)+DZ(2)) * (DZ(1)+DZ(2)) *COLD(2,NU)
&            - DZ(1)*DZ(1)*COLD(3,NU)
        YPO = YPO / (DZ(1)*DZ(2)*(DZ(1)+DZ(2)))

C
        YY(1) = 6.0D0*(YN(1)-YPO)

C
        DO I = 2, NM1
            YY(I) = 6.0D0*(YN(I)-YN(I-1))
        END DO

C
        YPO1 = DZ(NM1)*DZ(NM1)*COLD(NM2,NU)
&            - (DZ(NM2)+DZ(NM1)) * (DZ(NM2)+DZ(NM1)) *COLD(NM1,NU)
&            + (2.0D0*DZ(NM1)+DZ(NM2)) *DZ(NM2)*COLD(NODES,NU)
        YPO1 = YPO1 / (DZ(NM2)*DZ(NM1)*(DZ(NM2)+DZ(NM1)))

C
        YY(NODES)=6.0D0*(YPO1-YN(NM1))

C
        DS1(1) = YY(1)/BSOS(1)

C
        DO J = 2, NODES
            DENOM = BSOS(J)-ASOS(J)*CSS(J-1)
            DS1(J) = (YY(J)-ASOS(J)*DS1(J-1))/DENOM
        END DO

C
        COEF(NODES) = DS1(NODES)

C
        DO J = NM1, 1, -1
            COEF(J) = DS1(J)-CSS(J)*COEF(J+1)
        END DO

C
        CALL EVAL(NODES, VEL, ZKNOT, ZSTR, DZ, COEF,
&            COLD(1,NU), CSTR(1,NU))

C
        END DO

C
        IF (NFLAG(6) .EQ. 1) THEN
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
            WRITE(20,*) 'TIME = ', T
            DO I = 2, NM1
                WRITE(20,802) I, ZKNOT(I), (CSTR(I,NU), NU = 1, NUCS)
            END DO
            802 FORMAT(1X,1P,I4,E13.5,4E13.5)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        END IF

C
C    DEFINE THE TRANSPORT-TRANSFER MATRIX COEFFICIENTS. THEN, SEQUENTIALLY CALL
C    THE THOMAS ALGORITHM FOR SOLVING THE FOUR COUPLED LINEAR TRIDIAGONAL
C    SYSTEMS.
C    FIRST RADIONUCLIDE
C
        CW(NODES) = 0.0D0

```

```

      BW(NODES) = EPS(NLAYER)*(DSP(NLAYER,1)/DZ(NM1)) +
&      EPSIN*DSO(1)/DXIN
      AW(NODES) = -EPS(NLAYER)*DSP(NLAYER,1)/DZ(NM1)
      DW(NODES) = EPSIN*(DSO(1)/DXIN)*CIN(1)
C
      DO K = 1, NLAYER
        DO I = LDIV(K-1)+1, LDIV(K)-1
          DUMMYP = PAIR0*DEXP(ALPHA*ZKNOT(I))+PRESSN(I)
          DUMCON = DT*(VELOC(I)*VELOC(I)/DAIR(I)) -
&          (PRESSN(I)-PRESSO(I))/DUMMYP
&          - DT*VELOC(I)*ALPHA
          DUMCON = DUMCON/(1.0D0+RET(K,1))
          DENOM = FDZ(I)
          AW(I) = -DT*RATIO(I-1,1) / DENOM
          BW(I) = 1.0D0 + DT *
&          (LAM(1)+XLSPM(K,1)
&          + (RATIO(I-1,1)+RATIO(I,1))/DENOM)
          BW(I) = BW(I)+DUMCON
          CW(I) = -DT*RATIO(I,1) / DENOM
          DW(I) = CSTR(I,1) +
&          DT*(1.0D0-EPS(K))/(EPS(K)*(1.0D0+RET(K,1))) *
&          (PARTDN(K) * SOURCE(I) +
&          CONLM(K,1)*SSP(I))
        END DO
        IF (K .LT. NLAYER) THEN
          I = LDIV(K)
          EDVDZ = EPS(K)*DSP(K,1) / DZ(I-1)
          EDVDZ1 = EPS(K+1)*DSP(K+1,1) / DZ(I)
          AW(I) = -EDVDZ
          BW(I) = EDVDZ + EDVDZ1
          CW(I) = -EDVDZ1
          DW(I) = 0.0D0
        END IF
      END DO
C
      CW(1) = -EPS(1)*(DSP(1,1)/DZ(1))
      BW(1) = EPS(1)*(DSP(1,1)/DZ(1)) + EPSOT*(DSOT(1)/DXOUT)
      AW(1) = 0.0D0
      DW(1) = EPSOT*COT(1)*(DSOT(1)/DXOUT)
C
      CALL THOMAS(NODES,AW,BW,CW,DW,XNOW,CTEMP,DTEMP)
C
      DO I = 1, NODES
        CNEW(I,1) = XNOW(I)
      END DO
C
      CTSIR(1) = (1.0D0 - 0.5D0*DT*LAM(1)) /
&      (1.0D0 + 0.5D0*DT*LAM(1))
      DO K = 1, NLAYER
        CTIR1(K,1) = (0.5D0*DT)*LTIRR(K,1)
        CTIR1(K,1) = CTIR1(K,1)/(1.0D0 + 0.5D0*DT*LAM(1))
      END DO
      CTIR2(1) = 0.0D0
C
      SECOND THRU LAST RADIONUCLIDE
C
      DO NU = 2, NUCS
        CW(NODES) = 0.0D0
        BW(NODES) = EPS(NLAYER)*(DSP(NLAYER,NU)/DZ(NM1)) +
&        EPSIN*(DSO(NU)/DXIN)
        AW(NODES) = -EPS(NLAYER)*DSP(NLAYER,NU)/DZ(NM1)
        DW(NODES) = EPSIN*(DSO(NU)/DXIN)*CIN(NU)
C
        DO K = 1, NLAYER
          DO I = LDIV(K-1)+1, LDIV(K)-1

```

```

      DUMMYP = PAIRO*DEXP (ALPHA*ZKNOT(I)) +PRESSN(I)
      DUMCON = DT*(VELOC(I)*VELOC(I)/DAIR(I)) -
&          (PRESSN(I)-PRESSO(I))/DUMMYP -
&          DT*VELOC(I)*ALPHA
      DUMCON = DUMCON/(1.0D0+RET(K,NU))
      DENOM = FDZ(I)
      AW(I) = -DT*RATIO(I-1,NU) / DENOM
      BW(I) = 1.0D0 + DT *
&          (LAM(NU) + XLSPM(K,NU) +
&          (RATIO(I-1,NU)+RATIO(I,NU))/DENOM)
      BW(I) = BW(I) + DUMCON
      CW(I) = -DT * RATIO(I,NU) / DENOM
      DW(I) = CSTR(I,NU) + DT*CONLM(K,NU)*CNEW(I,NU-1)
    END DO
    IF (K .LT. NLayer) THEN
      I = LDIV(K)
      EDVDZ = EPS(K)*DSP(K,NU) / DZ(I-1)
      EDVDZ1 = EPS(K+1)*DSP(K+1,NU) / DZ(I)
      AW(I) = -EDVDZ
      BW(I) = EDVDZ + EDVDZ1
      CW(I) = -EDVDZ1
      DW(I) = 0.0D0
    END IF
  END DO
C
  CW(1) = -EPS(1)*(DSP(1,NU)/DZ(1))
  BW(1) = EPS(1)*(DSP(1,NU)/DZ(1)) + EPSOT*(DSOT(NU)/DXOUT)
  AW(1) = 0.0D0
  DW(1) = EPSOT*COT(NU)*(DSOT(NU)/DXOUT)
C
  CALL THOMAS (NODES,AW,BW,CW,DW,XNOW,CTEMP,DTEMP)
C
  DO I = 1, NODES
    CNEW(I,NU) = XNOW(I)
  END DO
C
  CTSIR(NU) = (1.0D0 - 0.5D0*DT*LAM(NU)) /
&          (1.0D0 + 0.5D0*DT*LAM(NU))
  DO K = 1, NLayer
    CTIR1(K,NU) = (0.5D0*DT)*LTIRR(K,NU)
    CTIR1(K,NU) = CTIR1(K,NU) / (1.0D0 + 0.5D0*DT*LAM(NU))
  END DO
  CTIR2(NU) = (0.5D0*LAM(NU-1)*DT) /
&          (1.0D0 + 0.5D0*DT*LAM(NU))
&
  END DO
C
C  CALCULATE THE IRREVERSIBLY BOUND CONCENTRATIONS
C
  DO J = 1, NODES
    K = JTOK(J)
    STIRN(J,1) = CTSIR(1)*STIRO(J,1) +
&          CTIR1(K,1)*(CNEW(J,1)+COLD(J,1))
    DO NU = 2, NUCS
      STIRN(J,NU) = CTSIR(NU)*STIRO(J,NU) +
&          CTIR1(K,NU)*(CNEW(J,NU)+COLD(J,NU)) +
&          CTIR2(NU)*(STIRN(J,NU-1)+STIRO(J,NU-1))
    END DO
  END DO
C
C  CALCULATE MASSES IN THE TRANSPORT AND FATE REGION NEXT
C
  DO NU = 1, NUCS
    DO J = 2, NODES
      K = JTOK(J)
      DZD = DZ(J-1)

```

```

      MTF(J-1,NU) = EPS(K)*DZD*AREA*
&               0.5D0*(CNEW(J,NU)+CNEW(J-1,NU))
      MTS(J-1,NU) = (1.0D0-EPS(K))*DZD*AREA*FREV(K,NU)*PARTDN(K)*
&               KEQ(K,NU)*0.5D0*(CNEW(J,NU)+CNEW(J-1,NU))
      MTIRR(J-1,NU) = (1.0D0-EPS(K))*DZD*AREA*FIRR(K,NU)*
&               PARTDN(K)*0.5D0*(STIRN(J,NU)+STIRN(J-1,NU))
      END DO
C
C   COMPUTE THE EXIT END MASS FLUXES AND THE CUMULATIVE FLUXES.
C
      IF (VELOC(1) .LE. 0.0D0) THEN
        QFLXN(NU) = -EPS(1)*DSP(1,NU)*(CNEW(2,NU)-CNEW(1,NU))/DZ(1)
&               + EPS(1)*VELOC(1)*CNEW(1,NU)
      ELSE
        QFLXN(NU) = -EPSOT*DSOT(NU)*(CNEW(1,NU)-COT(NU))/DXOUT
&               + VELOT*COT(NU)
      ENDIF
      QFLXN(NU) = QFLXN(NU) * SA(NU)
C
      CUMFN(NU) = CUMFO(NU)+DT*AREA*(QFLXN(NU)+QFLXO(NU))*0.5D0
      END DO
C
C   ADVANCE TIME ONE DT STEP AND CHECK FOR THE NEXT PRINT TIME.
C
      T = T + DT
      TOUT = T / 8.64D+4
C
      IF (IPRINT .EQ. NOUT) THEN
        WRITE(*, 27) 'Time = ', T
        IPRINT = 1
      ELSE
        IPRINT = IPRINT+1
      ENDIF
C
      1200 DO I = 1, NODES
        PRESSO(I) = PRESSN(I)
        DO NU = 1, NUCS
          COLD(I,NU) = CNEW(I,NU)
          STIRO(I,NU) = STIRN(I,NU)
        END DO
      END DO
C
      DO NU = 1, NUCS
        QFLXO(NU) = QFLXN(NU)
        CUMFO(NU) = CUMFN(NU)
      END DO
C
      TEST = T - TPRT(KKK)
      IF (TEST .GE. 0.0D0) THEN
        WRITE(21,33) TOUT,
&               (QFLXN(NU), NU = 1, NUCS),
&               (CUMFN(NU), NU = 1, NUCS)
        WRITE(22,33) TOUT,
&               (PRESSN(NDEPTS(I))+
&               PAIRO*DEXP(ALPHA*ZKNOT(NDEPTS(I))),
&               I = 1, NCOL)
        WRITE(23,33) TOUT, (VELOC(NDEPTS(I)), I = 1, NCOL,
&               LU = 24
        DO NU = 1, MIN(NUCS,4)
          WRITE(LU,33) TOUT, (SA(NU)*CNEW(NDEPTS(I),NU), I = 1, NCOL)
          LU = LU + 1
        END DO
Cold  WRITE(88,2020) TOUT, PRESSN(1)+PAIRO, DSOT(1), DT, FROZEN
      IREC = IREC + 1
      CALL BRITE9(IREC)

```

```

C      KKK = KKK + 1
C
      IF (KKK .GT. NPRT) DONE = .TRUE.
      IF (T .GT. TEND) DONE = .TRUE.
      END IF
      IHDTC = IHDTC - 1
      IF (.NOT. DONE) GO TO 3000
C
1300 STOP 'Normal FORTRAN termination'
      4 FORMAT(A)
      22 FORMAT(5X, A, 1P, 5E13.5)
      27 FORMAT(1X, A, 1P, E12.4)
      30 FORMAT(1X,F10.4,1P, 4E12.4)
      31 FORMAT(1X,29(1X,E12.4))
      32 FORMAT(1X,I6,F10.4,1P, 9E12.4)
      33 FORMAT(1X, F10.4,1P,18E12.4)
2010 FORMAT(/,' DT ', A, ' by ', I2, ' to ', G13.4,
      & ' due to ', A)
2020 FORMAT(1X, F10.4, 1P, 3E12.4, L4)
      END
*VELCTY
      SUBROUTINE VELCTY(COURANT, COURNTOK, COURFAIL)
      INCLUDE 'comsize.v09'
      INCLUDE 'comlayr.v09'
      INCLUDE 'comcons.v09'
      INCLUDE 'comctrl.v09'
      INCLUDE 'comgeom.v09'
      INCLUDE 'compump.v09'
      INCLUDE 'comrnxp.v09'
      DOUBLE PRECISION COURANT
      LOGICAL COURNTOK, COURFAIL
C..Logicals--
      INTEGER K
      LOGICAL COURTEST
      DOUBLE PRECISION DZ1, DZ2, P1, P2, P3, PPDZ
*****
* Compute the gas velocity at each node. If the Courant criterion fails
* at any node, then COURNTOK will be set FALSE, and COURFAIL will be set
* true. otherwise, COURNTOK, is TRUE and COURFAIL is false. The reason
* for two flags being set for the same information is part of the logic
* used in the adaptive time step reduction scheme, above.
*
*
* D. E. Cawlfieid
*****
      COURNTOK = .TRUE.
C..
C..Do first node . . .
C..
      print *, 'pressn(1)=', pressn(1) c      print *, 'pressn(2)=', pressn(2)
C      print *, 'pressn(3)=', PRESSN(3)
C      DZ1 = DZ(1)
      DZ2 = DZ(2)
      P1 = - PRESSN(1)*(2.0D0*DZ1 + DZ2)/DZ1
      P2 = (PRESSN(2)/(DZ1*DZ2))* (DZ1 + DZ2)**2
      P3 = - PRESSN(3)*DZ1/DZ2
      PPDZ = (P1 + P2 + P3) / (DZ1 + DZ2)
      VELOC(1) = -(KAPPA(1)/(ETA*AIR*EPS(1)))*
      & (PPDZ-ALPHA*PRESSN(1))
      COURANT = VELOC(1)*DT/DZ1
      COURTEST = COURNTOK .AND. ABS(COURANT) .GT. 1.0D0
      IF (COURTEST) THEN
        WRITE(*,2010) 1, COURANT
        COURFAIL = .TRUE.
        COURNTOK = .FALSE.
      END IF

```

```

C..
C..Do interior nodes . . .
C..
      DO K = 2, NM1
        DZ1 = DZ(K-1)
        DZ2 = DZ(K)
        P1 = - PRESSN(K-1) * DZ2 / DZ1
        P2 =  PRESSN(K ) * (DZ2**2 - DZ1**2) / (DZ1*DZ2)
        P3 =  PRESSN(K+1) * DZ1 / DZ2
        PPDZ = (P1 + P2 + P3) / (DZ1 + DZ2)
        VELOC(K) = -(KAPPA(JTOK(K)) / (ETAIR*EPS(JTOK(K)))) *
&          (PPDZ-ALPHA*PRESSN(K))
        COURANT = VELOC(K)*DT/DZ(K)
        COURTEST = COURNTOK .AND. ABS(COURANT) .GT. 1.0D0
        IF (COURTEST) THEN
          WRITE(*,2010) K, COURANT
          COURFAIL = .TRUE.
          COURNTOK = .FALSE.
        END IF
      END DO
      VELOC(NODES) = 0.0D0
      RETURN
2010 FORMAT(' Courant number > 1 at node ', I6, ':', 1P, G13.5)
      END

```

*READ1

Last Revision: April 8, 1993

C

Source File:9READ1.FOR

```

SUBROUTINE READ1
  INCLUDE 'comsize.v09'
  INCLUDE 'comlayr.v09'
  INCLUDE 'comgeom.v09'
  INCLUDE 'comname.v09'
  INCLUDE 'comasi.v09'
  INCLUDE 'compump.v09'
  INCLUDE 'comrnxp.v09'
  INCLUDE 'comctrl.v09'
  INCLUDE 'comnumr.v09'
  INCLUDE 'comcons.v09'
  INTEGER          I, J, NU, LU, JTMP
  CHARACTER         CZC
  CHARACTER*80      CNAME, FNAME*63
  INTEGER          JFROMZ
  DOUBLE PRECISION  ZFROMJ, ZTMP, DELTA
  EXTERNAL          ZFROMJ, JFROMZ

```

```

*****
*  Read the input data file. This routine obtains the "root" filename      *
*  (usually from the command line) by calling GET_ROOT, opens the required  *
*  input and output files, and reads the input data file.                  *
*                                                                           *
*                               D. E. Cawlfeld                               *
*****

```

```

      CALL GET_ROOT(ROOT, L12)

```

C..

```

      FNAME = ROOT(1:L12) // '.dat'
      OPEN(UNIT=10, FILE=FNAME, STATUS='OLD')

```

C..

```

      FNAME = ROOT(1:L12) // '.o20'
      OPEN(UNIT=20, FILE=FNAME, STATUS='UNKNOWN')

```

C..

```

      FNAME = ROOT(1:L12) // '.o21'
      OPEN(UNIT=21, FILE=FNAME, STATUS='UNKNOWN')

```

C..

```

      FNAME = ROOT(1:L12) // '.o22'
      OPEN(UNIT=22, FILE=FNAME, STATUS='UNKNOWN')

```



```

C..      FNAME = ROOT(1:L12) // '.o23'
        OPEN(UNIT=23 ,FILE=FNAME, STATUS='UNKNOWN')
C..
        INQUIRE (UNIT=10,NAME=FINP)
        CALL DATE(DAYT)
        CALL TIME(TYME)
        WRITE(20,2000) FINP(1:L12+4), DAYT, TYME
2000 FORMAT(' Input File: ', A, T40, 'Date: ', A, T60, 'Time: ', A)
C
C I. RUN CONTROL
C
        WRITE(*,*)
        WRITE(*,*)'Read in the CASCADR9.DAT data input file.      '
        WRITE(*,*)
        WRITE(*,*)' NOTE! Mixed input format. Any units '
        WRITE(*,*)'can be used as long as they are CONSISTANT. '
        WRITE(*,*)
        WRITE(*,*)' I. READ RUN CONTROL INPUT DATA '
        WRITE(*,*)
C
C THREE DESCRIPTOR LINES HEAD THE DATA FILE FOR VERSION 3
C
        I = 0
        READ (10,4) CNAME
        DO WHILE (CNAME(1:1) .NE. '$')
            I = I + 1
            IF (I .LE. 3) XNAME(I) = CNAME
            WRITE(20,5) CNAME
            READ (10,4) CNAME
        END DO
        WRITE(20,*)
        WRITE(20,*)
C
C BEGIN DATA READ IN STATEMENTS... MOSTLY FREE FOR READ FORMAT.
C
*****
* NFLAG - Description
* 1 - Data Preview Only.
* 2 - List System Control Constants to Report
* 3 - List Initial Concentrations to Report
* 4 - List Initial Masses to Report
* 5 - List Initial Velocity & Concentration to Screen
* 6 - Debug by listing intermediate CSTR to Screen
* 7 - Read in Barometric Pressure from file
* 8 - Read in DSOT table from file (24 hour cycle).
* 9 - Use the curvature of the barometric pressure wave to (help)
*    adjust the time step size. Only applicable if option 7 is
*    used also.
* 10 - Unused, unassigned.
*
*                                D. E. Cawlfieid
*****
        READ(10,4) DUMMY
        READ(10,*) (NFLAG(I),I=1,10)
C..
        READ(10,4) DUMMY
        READ(10,*) NLayer, NODES, NUCS, NOUT
        NM1 = NODES-1
        NM2 = NM1-1
        IF (NLayer .GT. MLayer) THEN
            PRINT *, 'Too many layers; change MLayer and re-compile.'
            STOP
        END IF
        IF (NODES .GT. NZSET) THEN
            PRINT *, 'Too many nodes, change NZSET and re-compile.'

```

```

        STOP 1010
    ENDIF
    IF (NUCS .GT. MNUC) THEN
        PRINT *, 'Too many nuclides, change MNUC and re-compile.'
        STOP 1020
    ENDIF
C..
    READ(10,4) DUMMY
    READ(10,*) CZC
    IF (CZC .EQ. 'a') CZC = 'A'
    IF (CZC .EQ. 'r') CZC = 'R'
    READ(10,4) DUMMY
    IF (CZC .EQ. 'A') THEN ! Z Absolute
        PRINT *, 'Reading Absolute Z Coordinates . . .'
        READ (10,*) (ZKNOT(J), J = 1, NODES)
        DO J = 1, NM1
            DELTA = ZKNOT(J+1) - ZKNOT(J)
            IF (DELTA .LE. 0.0D0) THEN
                WRITE(*,1030) 'Z', DELTA, J
                STOP 1030
            ELSE
                DZ(J) = DELTA
            END IF
        END DO
        ZOFF = ZKNOT(1)
    ELSE IF (CZC .EQ. 'R') THEN
        PRINT *, 'Reading Delta-Z''s . . .'
        READ (10,*) ZOFF, (DZ(J), J = 1, NM1)
        ZKNOT(1) = ZOFF
        DO J = 2, NODES
            ZKNOT(J) = ZKNOT(J-1) + DZ(J-1)
        END DO
    ELSE
        PRINT *, '*** READ1: Bad Flag, not ''A'' or ''R''!'
        STOP 1040
    END IF
C..
    READ(10,4) DUMMY
    READ(10,*) CZC
    IF (CZC .EQ. 'a') CZC = 'A'
    IF (CZC .EQ. 'r') CZC = 'R'
    READ(10,4) DUMMY
    J = 1
    LDIV(0) = 1
    LAYER(0) = 0.0
    DO I = 1, NLAYER
        IF (I .LT. NLAYER) THEN
            READ(10,*) LAYER(I)
            IF (CZC .EQ. 'R') THEN
                LAYER(I) = LAYER(I-1) + LAYER(I)
            END IF
        ELSE
            LAYER(I) = ZKNOT(NODES)
        END IF
        DO WHILE (ZKNOT(J+1) .LT. LAYER(I))
            JTOK(J) = I
            J = J + 1
        END DO
        IF (LAYER(I) - ZKNOT(J) .LT. ZKNOT(J+1) - LAYER(I)) THEN
            LDIV(I) = J
        ELSE
            LDIV(I) = J + 1
            JTOK(J) = I
            J = J + 1
        END IF
    END IF

```

```

END DO
JTOK(NODES) = NLAYER

READ(10,4) DUMMY
READ(10,*) NCOL, CZC
IF (CZC .EQ. 'n') CZC = 'N'
IF (CZC .EQ. 'z') CZC = 'Z'
NCOL = MIN(NCOL, MCOL)
READ(10,4) DUMMY
IF (CZC .EQ. 'N') THEN
  READ(10,*) (NDEPTS(I), I = 1, NCOL)
  IF (NDEPTS(NCOL) .GE. NODES) THEN
    PRINT *, 'Adjusting NDEPTS(ncol) < NODES.'
    NDEPTS(NCOL) = MIN(NDEPTS(NCOL), NODES-1) ! For Masses...
  END IF
  DO J = 1, NCOL
    ZTMP = ZFROMJ(NDEPTS(J))
    IF (ZTMP .EQ. -9.D9) THEN
      PRINT *, '*** READ1: Bad print node! ***'
      STOP 1050
    ELSE
      ZDEPTS(J) = ZTMP
    END IF
  END DO
ELSE IF (CZC .EQ. 'Z') THEN
  READ(10,*) (ZDEPTS(I), I = 1, NCOL)
  DO J = 1, NCOL
    JTMP = JFROMZ(ZDEPTS(J))
    IF (JTMP .EQ. -9) THEN
      PRINT *, '*** READ1: Bad print Znode! ***'
      STOP 1060
    ELSE
      NDEPTS(J) = JTMP
    END IF
  END DO
  IF (NDEPTS(NCOL) .GE. NODES) THEN
    PRINT *, 'Adjusting NDEPTS(ncol) < NODES.'
    NDEPTS(NCOL) = MIN(NDEPTS(NCOL), NODES-1) ! For Masses...
    ZDEPTS(NCOL) = ZFROMJ(NDEPTS(NCOL))
  END IF
ELSE
  PRINT *, '*** READ1: Bad Flag, not ''N'' nor ''Z''!'
  STOP 1070
ENDIF

C..
READ(10,4) DUMMY
READ(10,*) T0

C..
READ(10,4) DUMMY
READ(10,*) TEND

C..
READ(10,4) DUMMY
READ(10,*) NPRT
DO I = 1, NPRT
  TPRT(I) = ((TEND-T0)/DBLE(NPRT))*DBLE(I) + T0
END DO

C..
READ(10,4) DUMMY
READ(10,*) DT0

C..
Cold WRITE(*,*)
Cold WRITE(*,*) 'Approximate print out times are: '
Cold WRITE(*,8) (TPRT(I), I=1, NPRT)
Cold WRITE(20,*) 'Approximate print out times are: '
Cold WRITE(20,8) (TPRT(I), I=1, NPRT)

```

```

Cold WRITE(20,*)
Cold WRITE(20,*)
      8 FORMAT(10X,1P,5E12.4)
C
C II. SOURCE REGION AND POROUS MEDIUM TRANSPORT AND FATE DATA
C
      WRITE(*,*)
      WRITE(*,*)' II. POROUS MEDIUM DATA. '
      WRITE(*,*)

      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) PARTDN(I), EPS(I), KAPPA(I)
      END DO
      READ(10,4) DUMMY
      READ(10,*) EPSIN, EPSOT
      READ(10,4) DUMMY
      READ(10,*) (DSO (NU), NU = 1, NUCS)
      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) (DSP (I,NU), NU = 1, NUCS)
      END DO
      READ(10,4) DUMMY
      READ(10,*) (DSOT(NU), NU = 1, NUCS)
      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) (FREV(I, NU), NU = 1, NUCS)
      END DO
      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) (FIRR(I, NU), NU = 1, NUCS)
      END DO
      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) (KEQ(I, NU), NU = 1, NUCS)
      END DO
      READ(10,4) DUMMY
      READ(10,*) (THALF(NU), NU = 0, NUCS)
      DO NU = 0, NUCS
        IF (THALF(NU) .NE. 0.0) THEN
          LAM(NU) = 0.69314718D0 / THALF(NU)
        ELSE
          LAM(NU) = 1.0D308
        END IF
      END DO
C Atomic masses, Specific activities
      READ(10,4) DUMMY
      READ(10,*) (AM(NU), NU = 0, NUCS)
      READ(10,4) DUMMY
      READ(10,*) (SA(NU), NU = 0, NUCS)
      READ(10,4) DUMMY
      DO I = 1, NLAYER
        READ(10,*) (LTIRR(I,NU), NU = 1, NUCS)
      END DO
C
C II. INLET-OUTLET CHARACTERIZING PARAMETERS PARAMETERS
C
      WRITE(*,*)
      WRITE(*,*)' III. INLET-OUTLET CONCS. '
      WRITE(*,*)
      READ(10,4) DUMMY
      READ(10,*) AREA
      READ(10,4) DUMMY
      READ(10,*) (COT(NU), NU = 1, NUCS)
      READ(10,4) DUMMY

```

```

      READ(10,*) (CIN(NU), NU = 1, NUCS)
C
C  READ IN THE POROUS MEDIUM AND INLET-OUTLET GEOMETRIC NODAL SPACINGS.
C
      WRITE(*,*)
      WRITE(*,*) ' IV: GEOMETRIC SPACINGS '
      WRITE(*,*)
      READ(10,4) DUMMY
      READ(10,*) DXIN
      READ(10,4) DUMMY
      READ(10,*) DXOUT
      WRITE(*,*)
      WRITE(*,*) ' V: AIR PUMPING PARAMETERS '
      WRITE(*,*)
      READ(10,4) DUMMY
      READ(10,*) TAIR, RG, MASSA, GRAV, RHOWAT
      READ(10,4) DUMMY
      READ(10,*) PAIRO, NPRES
      READ(10,4) DUMMY
      READ(10,*) (POC(I), I = 1, 15)
      READ(10,4) DUMMY
      READ(10,*) (POS(I), I = 1, 15)
      READ(10,4) DUMMY
      READ(10,*) (LP (I), I = 1, 15)
C..
C..Open remaining output files (up to 4) . . .
C..
      LU = 24
      FNAME = ROOT(1:L12) // '.ONN'
      DO NU = 1, MIN(NUCS,4)
        WRITE(FNAME(L12+3:L12+4), '(I2)') LU
        OPEN(UNIT=LU ,FILE=FNAME, STATUS='UNKNOWN')
        LU = LU + 1
      END DO
      RETURN
      2 FORMAT(1X,A,$)
      4 FORMAT(A)
      5 FORMAT(1X,A)
      1030 FORMAT('/', ' *** OOPS *** DELTA', A1, ' = ', 1P, G13.6,
        & ' ' is <= Zero at J = ', 0P, I6, '!', '/')
      END
*READ2
      SUBROUTINE READ2
      INCLUDE 'comsize.v09'
      INCLUDE 'comlayr.v09'
      INCLUDE 'comgeom.v09'
      INCLUDE 'compump.v09'
      INCLUDE 'comrnxp.v09'
      INTEGER I, J
      INTEGER NSRC, ISRC, JBEG, JEND
      DOUBLE PRECISION ZBEG, ZEND, CTMP, STMP
      CHARACTER CZC, DUMMY
      INTEGER JFROMZ
      DOUBLE PRECISION ZFROMJ
      EXTERNAL ZFROMJ, JFROMZ
*****
* Routine to process the initial radionuclide distribution parameters as a *
* series of "layers". There is always a layer "0", which is the background *
* concentration, pressure, etc., for the entire column. The other "layers" *
* are intended to be the source regions. These may be specified by either *
* nodal positions, or z-depths. *
* *
* D. E. Cawlfieid, March 92
*****

```

```

      READ (10,4) DUMMY
      READ (10,*) NSRC, CZC
      IF (CZC .EQ. 'n') CZC = 'N'
      IF (CZC .EQ. 'z') CZC = 'Z'
      READ (10,4) DUMMY
      IF (CZC .EQ. 'N') THEN
C..Always process "0"th source . . .
        READ (10,*) ISRC, JBEG, JEND, CTMP, STMP
        IF (ISRC .NE. 0) THEN
          PRINT *, '*** READ2: Source region "0" not first! ***'
          STOP 2010
        ENDIF
        JBEG = 1
        JEND = NODES
        DO J = JBEG, JEND
          COLD(J,1) = CTMP
          SSP(J) = STMP
        END DO
C..Now process other source layers . . .
        DO I = 1, NSRC
          READ (10,*) ISRC, JBEG, JEND, CTMP, STMP
          ZBEG = ZFROMJ(JBEG)
          ZEND = ZFROMJ(JEND)
          IF (ZBEG .EQ. -9.D9 .OR. ZEND .EQ. -9.D9 .OR.
&           JBEG .GT. JEND) THEN
            PRINT *, '*** READ2: Bad region definition. ***'
            STOP 2020
          END IF
          DO J = JBEG, JEND
            COLD(J,1) = CTMP
            SSP(J) = STMP
          END DO
        ELSE IF (CZC .EQ. 'Z') THEN
C..Always process "0"th source . . .
          READ (10,*) ISRC, ZBEG, ZEND, CTMP, STMP
          IF (ISRC .NE. 0) THEN
            PRINT *, '*** READ2: Source region "0" not first! ***'
            STOP 2040
          ENDIF
          JBEG = 1
          JEND = NODES
          DO J = JBEG, JEND
            COLD(J,1) = CTMP
            SSP(J) = STMP
          END DO
C..Now process other source layers . . .
          DO I = 1, NSRC
            READ (10,*) ISRC, ZBEG, ZEND, CTMP, STMP
            JBEG = JFROMZ(ZBEG)
            JEND = JFROMZ(ZEND)
            IF (JBEG .EQ. -9 .OR. JEND .EQ. -9 .OR.
&           JBEG .GT. JEND) THEN
              PRINT *, '*** READ2: Bad region definition. ***'
              STOP 2050
            END IF
            DO J = JBEG, JEND
              COLD(J,1) = CTMP
              SSP(J) = STMP
            END DO
          ELSE
            PRINT *, '*** READ2: Unknown flag, not "N" or "Z"! '
            STOP 2070
          END IF

```

```

      RETURN
4  FORMAT (A)
      END
*READ3
      SUBROUTINE READ3
      INCLUDE 'comsize.v09'
      INCLUDE 'comname.v09'
      INCLUDE 'comlayr.v09'
      INCLUDE 'compump.v09'
      INTEGER      I, NU
*****
*  Routine which reads the table of DSOT's, which are the average dispersion *
*  coefficients at the boundary layer region. This routine is called (and *
*  the external table required) only if NFLAG(8) is set (==1). The external *
*  table of boundary layer dispersion coefficients then provided as one *
*  entry per hour of a 24 hour day. *
*                               D. E. Cawlfieid *
*****
      DO I = 0, 23
        READ(88,*,ERR=998,END=999) (DSOT0(I,NU), NU = 1, NUCS)
      END DO
      RETURN
998 PRINT *, '*** READ3 *** Error in data file.'
      STOP 3998
999 PRINT *, '*** READ3 *** Early EOF in data file.'
      STOP 3999
      END
*JFROMZ
      INTEGER FUNCTION JFROMZ(Z)
      INCLUDE 'comsize.v09'
      INCLUDE 'comgeom.v09'
      INTEGER      J
      DOUBLE PRECISION Z
*****
*  Routine to find the "closest" J node, given a depth Z. *
*  Here, "closest" is defined as the next node down. *
*                               D. E. Cawlfieid, March 92 *
*****
      J = -9
      IF (Z .LT. ZKNOT(1)) THEN
        PRINT *, '*** JFROMZ: Z < ZKNOT(1) ***'
      ELSE IF (Z .GT. ZKNOT(NODES)) THEN
        PRINT *, '*** JFROMZ: Z > ZKNOT(NODES) ***'
      ELSE
        J = 1
        DO WHILE (Z .GT. ZKNOT(J))
          J = J + 1
        END DO
      END IF
      JFROMZ = J
      RETURN
      END
*ZFROMJ
      DOUBLE PRECISION FUNCTION ZFROMJ(J)
      INCLUDE 'comsize.v09'
      INCLUDE 'comgeom.v09'
      INTEGER      J
      DOUBLE PRECISION Z
*****
*  Routine to find the depth Z given a node number J. *
*                               D. E. Cawlfieid, March '92 *
*****
      Z = -9.D9
      IF (J .LT. 1) THEN
        PRINT *, '*** ZFROMJ: J < 1 ***'

```

```

ELSE IF (J .GT. NODES) THEN
  PRINT *, '*** ZFROMJ: J > NODES ***'
ELSE
  Z = ZKNOT(J)
END IF
ZFROMJ = Z
RETURN
END
*GET_ROOT
SUBROUTINE GET_ROOT(ROOT, L12)
CHARACTER*(*)      ROOT
INTEGER            NARGC, L12
INTEGER            IARGC, LEN
integer*2          status
*****
* Get the "root" name, which is used to name all of the other files used by *
* used by the program. If possible, obtain the root name from the command *
* line. This is easily done on most Unix and DOS implementations; other *
* systems may require modification. *
* For non-command line systems, create an IARGC routine which returns 0. *
* When IARGC returns a value of zero, then the program will get the "root" *
* name interactively from the user terminal. *
*                               D. E. Cawlfieid, April 92. *
*****
  NARGC = IARGC()
  IF (NARGC .EQ. 0) THEN
    WRITE(*,2000) 'Enter ROOT of filenames > '
    READ (*,2010) ROOT
  ELSE
    CALL GETARG(1, ROOT, status)
  END IF
  L12 = LEN(ROOT)
  DO WHILE (ROOT(L12:L12) .EQ. ' ' .AND. L12 .GE. 1)
    L12 = L12 - 1
  END DO
  RETURN
2000 FORMAT(1X, A, $)
2010 FORMAT(A)
END

```

*REPT1
C

Last Revision: April 15, 1992
Source File: C8REPT1.FOR

```

SUBROUTINE REPT1
INCLUDE 'comsize.v09'
INCLUDE 'comlayr.v09'
INCLUDE 'comgeom.v09'
INCLUDE 'comname.v09'
INCLUDE 'comasi.v09'
INCLUDE 'compump.v09'
INCLUDE 'comrxp.v09'
INCLUDE 'comctrl.v09'
INCLUDE 'comnumr.v09'
INCLUDE 'comcons.v09'
INTEGER      NU, I
*****
* This routine writes a summary of several internal system control *
* constants which were formed from the values in the input data file. This *
* report is written once onto the output report, providing that NFLAG(2) is *
* set (==1). This report is occasionally useful to debug a new data set. *
*                               D. E. Cawlfieid *
*****

```



```

WRITE(20,*)
WRITE(20,*)' Writing out system control constants. '
WRITE(20,*)
DO NU = 1, NUCS
  WRITE(20,2022) ' CTSIR(' , NU, ') is ' , CTSIR(NU)
  WRITE(20,2023) ' CTIR1(I' , NU, ') are ' ,
& (CTIR1(I,NU), I = 1, NLAYER)
  WRITE(20,2022) ' CTIR2(' , NU, ') is ' , CTIR2(NU)
  WRITE(20,2023) ' XLSPM(I' , NU, ') are ' ,
& (XLSPM(I,NU), I = 1, NLAYER)
  WRITE(20,2023) ' CONLM(I' , NU, ') are ' ,
& (CONLM(I,NU), I = 1, NLAYER)
END DO
WRITE(20,22)' OMEG(1-5): ' ,OMEG(1),OMEG(2),OMEG(3),OMEG(4),
& OMEG(5)
WRITE(20,22)' OMEG(6-10): ' ,OMEG(6),OMEG(7),OMEG(8),OMEG(9),
& OMEG(10)
WRITE(20,22)' OMEG(11-15): ' ,OMEG(11),OMEG(12),OMEG(13),OMEG(14),
& OMEG(15)
WRITE(20,22) ' DAIR(1) is ' ,DAIR(1)
22 FORMAT(5X,A,1P,5E13.5)
2022 FORMAT(5X, A, I1.1, A, 1P, E13.5)
2023 FORMAT(5X, A, I1.1, A, 1P, 100E13.5)
WRITE(20,*)
WRITE(20,*)
WRITE(*,*)
WRITE(*,*)
WRITE(*, 22) 'Depth or distance to water table is (cm) ' , DEPTH
WRITE(20,22) 'Depth or distance to water table is (cm) ' , DEPTH
WRITE(20,*)
WRITE(*,*)
RETURN
END
*REPT2
SUBROUTINE REPT2
  INCLUDE 'comsize.v09'
  INCLUDE 'comgeom.v09'
  CHARACTER CH1, CH2
  INTEGER IP1, IP2, N, N2, LUN
*****
* This routines produces a table of nodes, depths, and delta-z's onto the *
* output report. It also flags those nodes which are output horizons with *
* a ">" mark. This report is very useful to check out the spacing table, *
* and requested print horizons. *
* D. E. Cawlfieid *
*****
  LUN = 20
  IF (NFLAG(10) .EQ. 1) LUN = 06
  WRITE(LUN,2000)
  2000 FORMAT(/,T25,'T a b l e o f N o d e s ' ,/)
  N2 = NODES / 2
C..
C..Set pointer IP2 to middle of NDEPTS list . . .
C..
  IP1 = 1
  IP2 = 1
  DO WHILE (N2+1 .GT. NDEPTS(IP2) .AND. IP2 .LT. NCOL)
    IP2 = IP2 + 1
  END DO
C..
C..List main body of table . . .
C..
  DO N = 1, N2-1
    CH1 = ' '
    CH2 = ' '

```

```

      IF (N .EQ. NDEPTS(IP1)) THEN
        CH1 = '>'
        IP1 = MIN(IP1+1,NCOL)
      END IF
      IF (N2+N .EQ. NDEPTS(IP2)) THEN
        CH2 = '>'
        IP2 = MIN(IP2+1,NCOL)
      END IF
      WRITE(LUN,2010) CH1, N, ZKNOT(N), DZ(N),
&                      CH2, N2+N, ZKNOT(N2+N), DZ(N2+N)
    END DO
C..
C..List last line of table . . .
C..
      IF (MOD(NODES,2) .EQ. 0) THEN
        CH1 = ' '
        CH2 = ' '
        IF (N2 .EQ. NDEPTS(IP1)) CH1 = '>'
        IF (NODES .EQ. NDEPTS(IP2)) CH2 = '>'
        WRITE(LUN,2010) CH1, N2, ZKNOT(N2), DZ(N2),
&                      CH2, NODES, ZKNOT(NODES), 0.0D0
      ELSE
        CH1 = ' '
        CH2 = ' '
        IF (N2 .EQ. NDEPTS(IP1)) THEN
          CH1 = '>'
          IP1 = MIN(IP1+1,NCOL)
        ENDIF
        IF (N2+N2 .EQ. NDEPTS(IP2)) THEN
          CH2 = '>'
          IP2 = MIN(IP2,NCOL)
        ENDIF
        WRITE(LUN,2010) CH1, N2, ZKNOT(N2), DZ(N2),
&                      CH2, N2+N2, ZKNOT(N2+N2), DZ(N2+N2)
      CH2 = ' '
      IF (NODES .EQ. NDEPTS(IP2)) CH2 = '>'
      WRITE(LUN,2012) CH2, NODES, ZKNOT(NODES), 0.0D0
    END IF
    RETURN
2010 FORMAT(T05, A, 1X, I5, 1P, 2E12.4, 0P,
&          T45, A, 1X, I5, 1P, 2E12.4)
2012 FORMAT(T45, A, 1X, I5, 1P, 2E12.4)
END

```

*BRITE
C

Last Revision: July 1, 1992
Source File: C9BRITE.FOR

```

SUBROUTINE BRITE_INIT(IREC)
  INTEGER IREC
  INCLUDE 'comsize.v09'
  INCLUDE 'comlayr.v09'
  INCLUDE 'comgeom.v09'
  INCLUDE 'comname.v09'
  INCLUDE 'comasi.v09'
  INCLUDE 'compump.v09'
  INCLUDE 'comrnxp.v09'
  INCLUDE 'comctrl.v09'
  INCLUDE 'comnumr.v09'
  INCLUDE 'comcons.v09'
  INTEGER J
  CHARACTER FNAME*63

```

```

*****
* Routine to open and initialize the binary output file. The binary output *
* file is used to hold the results of the modling run for post processing *
* (plotting). *
* D. E. Cawlfieid, Mar 92 *
*****
      FNAME = ROOT(1:L12) // '.bin'
      OPEN (60, FILE=FNAME,
& FORM='UNFORMATTED', ACCESS='SEQUENTIAL',
& STATUS='UNKNOWN')
      WRITE(60) VERSION
      WRITE(60) FINP, DAYT, TYME
      WRITE(60) NODES, NUCS, NCOL, NPRT, NPRES, NOUT
      WRITE(60) T0, TEND, DT0
      WRITE(60) XNAME(1)
      WRITE(60) XNAME(2)
      WRITE(60) XNAME(3)
      WRITE(60) (DZ(J), J = 1, NM1)
      WRITE(60) (ZKNOT(J), J = 1, NODES)
      WRITE(60) (ZDEPTS(J), J = 1, NCOL)
      WRITE(60) (NDEPTS(J), J = 1, NCOL)
      WRITE(60) IREC, TOUT
      RETURN
      END
*BRITE9
      SUBROUTINE BRITE9(IREC)
      INTEGER IREC
      INCLUDE 'comsize.v09'
      INCLUDE 'comlayr.v09'
      INCLUDE 'comgeom.v09'
      INCLUDE 'comname.v09'
      INCLUDE 'comasi.v09'
      INCLUDE 'compump.v09'
      INCLUDE 'comrnxp.v09'
      INCLUDE 'comctrl.v09'
      INCLUDE 'comnumr.v09'
      INCLUDE 'comcons.v09'
      INTEGER I, NU
*****
* Write the current state of the run onto the binary output file. *
* D. E. Cawlfieid, Mar 92 *
*****
      WRITE(60) IREC, TOUT
      WRITE(60) (PRESSN(NDEPTS(I))+PAIR0*DEXP(ALPHA*ZKNOT(NDEPTS(I))),
& I = 1, NCOL)
      WRITE(60) (VELOC(NDEPTS(I)), I = 1, NCOL)
      WRITE(60) (QFLXN(NU), NU = 1, NUCS)
      WRITE(60) (CUMFN(NU), NU = 1, NUCS)
      DO NU = 1, NUCS
        WRITE(60) (SA(NU)*CNEW(NDEPTS(I),NU), I = 1, NCOL)
        WRITE(60) (STIRN(NDEPTS(I),NU), I = 1, NCOL)
        WRITE(60) (MTF(NDEPTS(I),NU), I = 1, NCOL)
        WRITE(60) (MTS(NDEPTS(I),NU), I = 1, NCOL)
        WRITE(60) (MTIRR(NDEPTS(I),NU), I = 1, NCOL)
      END DO
      RETURN
      END

```

*AIRPUMP9

Last Revision: December 1, 1992
Source File: AIRPUMP9.FOR

```
C
SUBROUTINE AIRPUMP(COURFAIL, IHDTC, FROZEN)
  INCLUDE 'comsize.v09'
  INCLUDE 'comlayr.v09'
  INCLUDE 'comgeom.v09'
  INCLUDE 'comasi.v09'
  INCLUDE 'compump.v09'
  INCLUDE 'comctrl.v09'
  INCLUDE 'comname.v09'
  INCLUDE 'comnumr.v09'
  INCLUDE 'comcons.v09'

C..Locals --
  DOUBLE PRECISION SUMPRS, TNOW
  LOGICAL FIRST, COURFAIL
  INTEGER I, K

C..The following is for splineing the baro data . . .
  INTEGER MX, IN
  PARAMETER (MX=1000)
  DOUBLE PRECISION DA(MX), BP(MX), DA1, DAN, BPY2(MX), BNEW
  COMMON /FUTZ1/ DA, BP
  COMMON /FUTZ2/ BPY2
  DOUBLE PRECISION FDZ, DZD, KVDZM, KVDZP, CON1
  EXTERNAL HUNT ! Use sequential search in table

C..The following are for the Curvature Heuristic . . .
  CHARACTER FNAME*63
  INTEGER IHDTC, IHDTC, ID
  DOUBLE PRECISION KURVTR, DTOLD
  LOGICAL THA1, THA2, THAW, FROZEN

*****
* Air pumping in dry, isothermal soils of Area 5, NTS. Complex mixture of up*
* to 15 sinusoids giving a variation about a fixed mean in atmospheric *
* pressure at the half space boundary [0,+∞). *
* David E. Cawfield, 92 *
*****

C..
C..The following is an Arithmetic Statement F . . .
C..
  FDZ(K) = DZ(K-1) + DZ(K)
  DATA FIRST/.TRUE./, IN/0/, DA1, DAN/2*1.D30/ C..
C..Get new Surface Barometric Pressure . . .
C..
  DTOLD = DT
  TNOW = T + DT
  IF (NFLAG(7) .EQ. 0) THEN ! Use Fourier Series for P
    SUMPRS = 0.0D0
    DO I = 1, NPRES
      SUMPRS = SUMPRS + POC(I)*DCOS(OMEG(I)*TNOW) +
&      POS(I)*DSIN(OMEG(I)*TNOW)
    END DO
    PRESSN(1) = 1.0D+3*SUMPRS
  ELSE ! Use Data File for P
    IF (FIRST) THEN
      CALL SETUP_SPLINE(MX, IN, DA, BP)
      CALL SPLINE(DA, BP, IN, DA1, DAN, BPY2)
      FNAME = ROOT(1:12) // '.kvr'
      OPEN(87, FILE=FNAME, STATUS='UNKNOWN')
      WRITE(87,2000)
      FIRST = .FALSE.
    END IF
    CALL SPLINT(DA, BP, BPY2, IN, HUNT, TNOW, BNEW, KURVTR)

C..
C..Adjust the current DT depending upon the curvature of the spline, KURVTR.
C..This action will also depend upon whether we have already made an
```

C..adjustment (FROZEN), and/or whether or not we have encountered problems
C..with the Courant number in VELCTY (as flagged by COURFAIL).
C..

```

      IF (.NOT. COURFAIL) THEN
        THA1 = IHDTC .LE. 0
        THA2 = IHDTK .LE. 0
        THAW = THA1 .AND. THA2
        IF (THAW .AND. KURVTR .LT. 50.0D0) THEN
          DT = MIN(DT0, 2.0D0*DT)
          IF (DT .NE. DTOLD)
&          WRITE(*,2010) ' raised', 02, DT, 'relaxation', KURVTR
&          , T + DT
          FROZEN = .FALSE.
        ELSE
          IF (NFLAG(9) .EQ. 1) THEN
            IF (.NOT. FROZEN .AND. KURVTR .GE. 50.0D0) THEN
              CALL KRAZY(DA, BP, BPY2, IN, KURVTR, ID, IHDTK)
              WRITE(*,2010) 'lowered', ID, DT, 'curvature.',
&              KURVTR, T + DT
              FROZEN = .TRUE.
            ELSE
              write(*,'(A,$)') ' .'
              IHDTK = IHDTK - 1
            END IF
            ! FROZEN restriction
          END IF
            ! Only when flag 9 set
          END IF
            ! KURVTR selection
          END IF
            ! COURFAIL restriction
          IF (DT .NE. DTOLD) THEN
            ! in case we changed dt
            TNOW = T + DT
            CALL SPLINT(DA, BP, BPY2, IN, HUNT, TNOW, BNEW, KURVTR)
          END IF
          PRESSN(1) = BNEW - PAIRO
        END IF

```

C..
C..Re-compute AA, BB, CC, and DD . . .
C..

```

      DD(1) = PRESSN(1)
      DO K = 1, NLAYER
        ! Re-compute AA, BB & CC
        DO I = LDIV(K-1)+1, LDIV(K)-1
          CON1 = 2.0D0*DAIR(I)*DT
          DZD = FDZ(I)
          AA(I) = -CON1 / (DZ(I-1)*DZD)
          BB(I) = 1.0D0 + CON1/(DZ(I-1)*DZ(I)) + DT*KADZ(I)
          CC(I) = -CON1 / (DZ(I)*DZD)
          DD(I) = PRESSO(I)
        END DO
        IF (K .LT. NLAYER) THEN
          I = LDIV(K)
          KVDZM = KAPPA(K) / DZ(I-1)
          KVDZP = KAPPA(K+1) / DZ(I)
          AA(I) = -KVDZM
          BB(I) = KVDZM + KVDZP + (KAPPA(K+1) - KAPPA(K))*ALPHA
          CC(I) = -KVDZP
          DD(LDIV(K)) = 0.0D0
        END IF
      END DO
      DD(NODES) = 0.0D0

```

C..
CALL THOMAS (NOLES, AA, BB, CC, DD, XNOW, CTEMP, DTEMP)

C..
DO I = 1, NODES
PRESSN(I) = XNOW(I)
END DO

C..
RETURN

```

2000 FORMAT('      "T1"      ', '      "T2"      ', '      "T3"      ',
&      '      "K1"      ', '      "K2"      ', '      "K3"      ',
&      '      "DTOLD"    ', '      "DTNEW"    ', '      "IHDTK"    ')
2010 FORMAT(/, ' DT ', A, ' by ', I2, ' to ', 1P, G12.4,
&      ' due to ', A, 1P, 2G12.4)
END
*KRAZY
SUBROUTINE KRAZY(DA, BP, BPY2, IN, KURVTR, ID, IHDTK)
INCLUDE 'comsize.v09'
INCLUDE 'comctrl.v09'
INTEGER      IN, ID, IHDTK
DOUBLE PRECISION DA(IN), BP(IN), BPY2(IN), KURVTR
C..Locals --
INTEGER      MTRIP, ITRIP
PARAMETER    (MTRIP=5000)
DOUBLE PRECISION TLOCAL, DTLOCL, DTOLD, BNEW
DOUBLE PRECISION T1, TMAX, T3, K1, KVRMAX, K3
EXTERNAL     LOCATE ! Use bisection search into table
*****
* This routine examines the local curvature of the barometric data and      *
* applies a heuristic to reduce the time step if the curvature is too      *
* "sharp". This routine is called only if NFLAG(9) is set.                  *
*****
      DTOLD = DT
      TLOCAL = T + DT
      TMAX = TLOCAL
      KVRMAX = KURVTR
      DTLOCL = DT0 / 32.0D0
      DTLOCL = MIN(DTLOCL, 0.25D0*DTOLD)
C..
C..Find local maximum curvature on this bend . . .
C..
      ITRIP = 0
      T1 = TLOCAL
      K1 = KVRMAX
      DO WHILE (KURVTR .GE. 50.0D0)
          T3 = TLOCAL
          K3 = KURVTR
          TLOCAL = TLOCAL + DTLOCL
          CALL SPLINT(DA, BP, BPY2, IN, LOCATE, TLOCAL, BNEW, KURVTR)
          IF (KURVTR .GT. KVRMAX) THEN
Cdbg          print *, kvrmax, kurvtr, tmax, tlocal, itrip
              KVRMAX = KURVTR
              TMAX = TLOCAL
          END IF
          ITRIP = ITRIP + 1
          IF (ITRIP .GT. MTRIP) STOP 'Serious problem in KRAZY'
      END DO
C..
C..Use local maximum curvature to set new DT (DTL) . . .
C..
      IF (KVRMAX .LT. 100.0D0) THEN
          DT = DT0 / 8.0D0
          DT = MIN(DTOLD, DT)
          ID = 8
      ELSE IF (KVRMAX .LE. 150.0D0) THEN
          DT = DT0 / 16.0D0
          DT = MIN(DTOLD, DT)
          ID = 16
      ELSE
          DT = DT0 / 32.0D0
          DT = MIN(DTOLD, DT)
          ID = 32
      END IF

```

```

C..      IHDTK = NINT((T3-T1)/DT) + 1
          KURVTR = KVRMAX
CDBG     WRITE(*,2010) T1, TMAX, T3, K1, KVRMAX, K3, DTOLD, DT, IHDTK
          WRITE(87,2010) T1, TMAX, T3, K1, KVRMAX, K3, DTOLD, DT, IHDTK
          RETURN
2010     FORMAT(1P, 8E12.4, 0P, I6)
          END
*SET_DSOT
SUBROUTINE SET_DSOT
  INCLUDE 'comsize.v09'
  INCLUDE 'comlayr.v09'
  INCLUDE 'comasi.v09'
  INCLUDE 'comctrl.v09'
  INCLUDE 'compump.v09'
  INTEGER          NU, INDX
  DOUBLE PRECISION TCHG, TNOW
  LOGICAL          ISHOUR
  SAVE             TCHG
*****
* This routine sets the DSOT for each nuclide to the value contained in the *
* external table of DSOT's, dependant upon time of day. This table is a 24 *
* hour table which is used only if NFLAG(8) is set. *
*                               D. E. Cawlfieid *
*****
DATA TCHG/0.0D0/, INDX/0/
C..
  IF (TCHG+DT .GT. 3599.99D0) THEN      ! Integral hours.
    TCHG = 0.0D0
    ISHOUR = .TRUE.
  ELSE
    TCHG = TCHG + DT
    ISHOUR = .FALSE.
  END IF
C..
C..Possibly twaddle with the DSOT's
C..
  TNOW = T + DT
  IF (ISHOUR) THEN
    INDX = MOD(INDX+1, 24)
    DO NU = 1, NUCS
      DSOT(NU) = DSOT0(INDX,NU)
    END DO
D      WRITE(*,1010) TNOW, INDX, DSOT(1)
  END IF
  RETURN
D1010    FORMAT('*** DSOT changed at ', F12.1, ', INDX ', I2, F13.4)
  END
*SETUP_SPLINE
SUBROUTINE SETUP_SPLINE(MX, IN, DA, BP)
  INCLUDE 'comsize.v09'
  INCLUDE 'comctrl.v09'
  INTEGER          MX, I, IEOF, IN
  DOUBLE PRECISION DA(MX), BP(MX)
  DOUBLE PRECISION TEMDA, TEMBP, DBASE
  LOGICAL          DONE, LBASE, LTEND
  Last Revision: September 23, 1992

```

```

*****
* Read the Barometric Data file (BARO.DAT) in preparation for splineing. *
* The data file should have a few data extra data points at the beginning, *
* to allow for the boundary of the (natural) cubic spline. * * Similarly, *
* the data file will be read to a few points past the required stopping *
* point for boundary purposes. The data file currently in use starts at *
* several hundred days; the variable DBASE is used to adjust the data to *
* time "zero". Also, the barometric data in the file is in mbars and must *
* be converted to dynes/cm2. After the proper section of data has been *
* extracted, the routines SPLINE and SPLINT will be used to spline the *
* data table. *
*
* David E. Cawlfeld, Sept 92 *
*****
      OPEN (87, FILE='baro.dat', STATUS='UNKNOWN')
C..
C..First, find the "baseline" point in the data file, which is the first point
C..where the data is on an even day (ie-- starts at midnight).
C..
      READ(87,*,END=990) TEMDA
      I = 1
      LBASE = TEMDA .EQ. AINT(TEMDA)
      DONE = LBASE .OR. I .GT. 50
      DO WHILE (.NOT. DONE)
        READ(87,*,END=990) TEMDA
        I = I + 1
        LBASE = TEMDA .EQ. AINT(TEMDA)
        DONE = LBASE .OR. I .GT. 50
      END DO
      IF (LBASE) THEN
        DBASE = TEMDA
        WRITE(*,2000) DBASE, I
      ELSE
        PRINT *, '***SPLINBAR*** Can''t find Base Time'
        STOP 2001
      END IF
C..
C..Now, read in the data file . . .
C..If the data file has time in days, reading to TEND+1 should provide a few
C..extra points at the end for boundary purposes.
C..
      REWIND(87)
      READ(87,*,IOSTAT=IEOF,END=990) TEMDA, TEMBP
      LTEND = (TEMDA-DBASE)*8.64D+4 + T0 .GE. TEND+8.64D+4 ! iff in days !
      DONE = LTEND .OR. IEOF .NE. 0 .OR. IN .EQ. MX
      DO WHILE (.NOT. DONE)
        IN = IN + 1
        DA(IN) = (TEMDA - DBASE)*8.6400D+4 + T0
        BP(IN) = TEMBP * 1.0D3
        READ(87,*,IOSTAT=IEOF) TEMDA, TEMBP
        LTEND = DA(IN) .GE. TEND+8.64D+4 ! iff in days !
        DONE = LTEND .OR. IEOF .NE. 0 .OR. IN .EQ. MX
      END DO
      CLOSE(87)
      IF (LTEND) THEN
        PRINT *, 'Read in ', IN, ' Baro values, last was at ', DA(IN)
      ELSE
        PRINT *, '*** SPLINBAR *** Not enough data in BARO.DAT'
      END IF
      RETURN
      990 STOP '*** SPLINBAR *** Early EOF'
      2000 FORMAT(' SPLINBAR found Time Base = ', F9.4, ' at record ', I2)
      END

```


*SPLINE Last Revision: September 23, 1992
C Source File: SPLINE.FOR

```

SUBROUTINE SPLINE(X, Y, N, YP1, YPN, Y2)
IMPLICIT NONE
INTEGER N, NMAX, I, K
PARAMETER (NMAX=1000)
DOUBLE PRECISION YP1, YPN, SIG, P, QN, UN
DOUBLE PRECISION X(N), Y(N), Y2(N), U(NMAX)
*****
* This is the SPLINE routine from PRESS, et al. *
*****
IF (YP1 .GT. .99D30) THEN
  Y2(1)=0.0D0
  U(1) =0.0D0
ELSE
  Y2(1)=-0.50D0
  U(1)=(3.0D0/(X(2)-X(1)))*((Y(2)-Y(1))/(X(2)-X(1))-YP1)
ENDIF
DO 11 I=2,N-1
  SIG=(X(I)-X(I-1))/(X(I+1)-X(I-1))
  P=SIG*Y2(I-1)+2.0D0
  Y2(I)=(SIG-1.0D0)/P
  U(I)=(6.0D0*((Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/
    / (X(I)-X(I-1)))/(X(I+1)-X(I-1))-SIG*U(I-1))/P
11 CONTINUE
IF (YPN .GT. .99D30) THEN
  QN=0.0D0
  UN=0.0D0
ELSE
  QN=0.50D0
  UN=(3.0D0/(X(N)-X(N-1)))*(YPN-(Y(N)-Y(N-1))/(X(N)-X(N-1)))
ENDIF
Y2(N)=(UN-QN*U(N-1))/(QN*Y2(N-1)+1.0D0)
DO 12 K=N-1,1,-1
  Y2(K)=Y2(K)*Y2(K+1)+U(K)
12 CONTINUE
RETURN
END

```

*SPLINT Last Revision: October 22, 1992
C Source File: SPLINK.FOR

```

SUBROUTINE SPLINT(XA, YA, Y2A, N, HUNT, X, Y, KURVTR)
INTEGER N
DOUBLE PRECISION XA(N), YA(N), Y2A(N), X, Y, KURVTR
EXTERNAL HUNT
C..Locals --
INTEGER KLO, KHI, KHL
DOUBLE PRECISION FCTR, TERM, AB, YP, YPP, A, B, H
*****
* This is the SPLINT routine from PRESS, et al. *
* A modification has been made to compute the "curvature of the spline". *
* David E. Cawlfeld, Sept 92 *
*****
DATA KLO /0/ ! For first call to HUNT !
C..
C..The following are two ARITHMETIC STATEMENT FUNCTIONS
C..
FCTR(AB) = 3.0D0*AB**2 - 1.0D0
TERM(AB,KHL) = FCTR(AB)*Y2A(KHL)
C..
C..HUNT is passed as an external routine: usually HUNT (sequentially ordered)
C..or LOCATE (binary search).
C..
CALL HUNT(XA, N, X, KLO)
KHI = KLO + 1
Cdbg if (klo .lt. 0 .or. khi .gt. n) then

```

```

Cdbg      klo = klo
Cdbg      pause
Cdbg      endif
      H=XA(KHI)-XA(KLO)
      IF (H.EQ.0.) STOP 'Bad XA input.'
      A=(XA(KHI)-X)/H
      B=(X-XA(KLO))/H
      Y=A*YA(KLO)+B*YA(KHI)+
&      ((A**3-A)*Y2A(KLO)+(B**3-B)*Y2A(KHI))*(H**2)/6.0D0

```

```

C..
C..Added lines to compute curvature . . .
C..

```

```

      YP = (YA(KHI) - YA(KLO)) / H
&      + (TERM(B,KHI) - TERM(A,KLO)) * H / 6.0D0
      YP = 8.64D+4*YP / 1.0D+3

```

```

C..
      YPP = A*Y2A(KLO) + B*Y2A(KHI)
      YPP = 7.640D+9*YPP / 1.0D+3

```

```

C..
      KURVTR = YPP / SQRT(1.0D0 + YP**2)**3
      KURVTR = ABS(KURVTR)
      RETURN
      END

```

```

*HUNT

```

Last Revision: October 22, 1992
Source File: HUNT.FOR

```

C
      SUBROUTINE HUNT(XX, N, X, JLO)
      INTEGER      N,      JLO
      DOUBLE PRECISION XX(N), X

```

```

C..Locals --

```

```

      INTEGER      JHI, JM, INC
      LOGICAL      ASCND

```

```

*****
* This is the HUNT routine from Press, et al.
* This version performs a search on an ordered table XX(N), assuming
* correlated values of X. That is, it assumes that you will be searching
* the table using sequential X's.
*****

```

```

      ASCND = XX(N) .GT. XX(1)
      IF (JLO .LE. 0 .OR. JLO .GT. N) THEN
        JLO = 0
        JHI = N+1
        GO TO 3

```

```

      END IF
      INC = 1
      IF (X .GE. XX(JLO) .EQV. ASCND) THEN
1      JHI = JLO+INC
        IF (JHI .GT. N) THEN
          JHI = N+1
        ELSE IF (X .GE. XX(JHI) .EQV. ASCND) THEN
          JLO = JHI
          INC = INC+INC
          GO TO 1
        ENDIF

```

```

      ELSE
2      JHI = JLO
        JLO = JHI-INC
        IF (JLO .LT. 1) THEN
          JLO = 0
        ELSE IF (X .LT. XX(JLO) .EQV. ASCND) THEN
          JHI = JLO
          INC = INC+INC
          GO TO 2
        ENDIF

```

```

3      IF (JHI-JLO .EQ. 1) RETURN

```

```

      JM = (JHI+JLO)/2
      IF (X .GT. XX(JM) .EQV. ASCND) THEN
        JLO = JM
      ELSE
        JHI = JM
      ENDIF
      GO TO 3
    END

*LOCATE
C
      SUBROUTINE LOCATE(XX, N, X, J)
      INTEGER      N, J
      DOUBLE PRECISION XX(N), X
C..Locals --
      INTEGER      JL, JM, JU
*****
* This is the LOCATE routine from Press, et al.
* This does a bisection search for the location J of a value X in an
* ordered table XX(N).
*****
      JL = 0
      JU = N+1
      DO WHILE (JU-JL .GT. 1)
        JM = (JU+JL)/2
        IF ( (XX(N) .GT. XX(1)) .EQV. (X .GT. XX(JM)) ) THEN
          JL = JM
        ELSE
          JU = JM
        ENDIF
      END DO
      J = JL
      RETURN
    END

*EVAL
C
      SUBROUTINE EVAL(NODES, VEL, ZKNOT, ZSTR, DZ, COEF, COLD, CSTR)
      INTEGER      NODES
      DOUBLE PRECISION VEL(NODES), ZKNOT(NODES), ZSTR(NODES)
      DOUBLE PRECISION COEF(NODES), COLD(NODES), CSTR(NODES)
      DOUBLE PRECISION DZ(NODES-1)
C..Locals --
      DOUBLE PRECISION ZDUMO, ZDUM1, AAA, BBB, CCC
      INTEGER      J
*****
* Subroutine for generating the coefficients for each of the lagrange
* cubic splines.
* Note: Although the variables which are defined above have similar names
* as those which appear in various common blocks . . . the variables in
* this routine are *local copies*. Specifically, several doubly
* dimensioned arrays are singly dimensioned locally.
* D. E. Cawlfeld, Mar 92
*****
C
C EVALUATION OF SPLINE AT INTERPOLATION POINTS.
C
      DO J= 2, NODES-1
        IF (VEL(J) .GT. 0.0D0) THEN
          ZDUMO = ZKNOT(J) - ZSTR(J)
          ZDUM1 = ZSTR(J) - ZKNOT(J-1)
          AAA = (ZDUM1*ZDUM1-DZ(J-1)*DZ(J-1))*ZDUM1
          AAA = AAA*COEF(J)/(6.0D0*DZ(J-1))
          BBB = (ZDUMO*ZDUMO-DZ(J-1)*DZ(J-1))*ZDUMO

```

```

      BBB = BBB*COEF(J-1)/(6.0D0*DZ(J-1))
      CCC = COLD(J-1)*ZDUMO+COLD(J)*ZDUM1
      CCC = CCC/DZ(J-1)
      CSTR(J) = AAA+BBB+CCC
    ELSE
      ZDUMO = ZKNOT(J+1) - ZSTR(J)
      ZDUM1 = ZSTR(J) - ZKNOT(J)
      AAA = (ZDUM1*ZDUM1-DZ(J)*DZ(J))*ZDUM1
      AAA = AAA*COEF(J+1)/(6.0D0*DZ(J))
      BBB = (ZDUMO*ZDUMO-DZ(J)*DZ(J))*ZDUMO
      BBB = BBB*COEF(J)/(6.0D0*DZ(J))
      CCC = COLD(J)*ZDUMO+COLD(J+1)*ZDUM1
      CCC = CCC/DZ(J)
      CSTR(J) = AAA+BBB+CCC
    ENDIF
  END DO
C
  RETURN
END

```

```

*THOMAS9
C
      SUBROUTINE THOMAS(NNNN,AAAA,BBBB,CCCC,DDDD,XNOW,CTEMP,DTEMP)
      INTEGER NNNN,NM1,KKK,J
      DOUBLE PRECISION AAAA(NNNN),BBBB(NNNN),
&      CCCC(NNNN),DDDD(NNNN),XNOW(NNNN)
      DOUBLE PRECISION CTEMP(NNNN),DTEMP(NNNN)
      DOUBLE PRECISION DENOM
      *****
      * Subroutine THOMAS is used for computing the solution of a tridiagonal
      * system of linear equations. It is assumed that any full or partial
      * pivoting has been accomplished a priori.
      *****
C
      NM1=NNNN-1
C
C    FORWARD ELIMINATION STEP.
C
      CTEMP(1) = CCCC(1)/BBBB(1)
      DTEMP(1) = DDDD(1)/BBBB(1)
      DO 10, J = 2, NNNN
        DENOM = BBBB(J)-AAAA(J)*CTEMP(J-1)
        CTEMP(J) = CCCC(J)/DENOM
        DTEMP(J) = (DDDD(J)-AAAA(J)*DTEMP(J-1))/DENOM
      10 CONTINUE
C
C    BACKWARD SUBSTITUTION STEP.
C
      XNOW(NNNN) = DTEMP(NNNN)
      DO 20, J = 1, NM1
        KKK = NNNN-J
        XNOW(KKK) = DTEMP(KKK)-CTEMP(KKK)*XNOW(KKK+1)
      20 CONTINUE
C
      RETURN
END

```

```

*DTS
C
Last Revision: April 2, 1992
Source File: DTS.FOR
*****
* IBM versions of the Vax VMS "DATE", "TIME", and "SECNDS" routine.
* This will be placed into FUTIL.LIB as compatability modules.
* The getarg and iargc routines have been added to map into the unix FORTRAN
* environment.
*
* D. E. Cawlfieid, Feb 91
*****
*SECNDS Time *function*
C
Last Revision: February 11, 1991
Source File: SECNDS.FOR

REAL FUNCTION SECNDS(T0)
INTEGER*2 IH, IM, IS, IHU
REAL START, DAY, T0
SAVE START, DAY
*****
* IBM version of the DEC VAX FORTRAN SECNDS routine.
* Returns the number of seconds and hundredths of seconds elapsed since
* midnight. This version is designed to emulate the VAX routine of the same
* name. T0 is subtracted from SECNDS before returning.
*
* D. E. Cawlfieid, Dec 90
*****
DATA START, DAY/ 2*0.0 /
CALL GETTIM(IH, IM, IS, IHU)
SECNDS = 3600.*IH + 60.*IM + IS + 0.01*IHU
IF (SECNDS+DAY+1.0 .LT. START) DAY = DAY + 86400.
SECNDS = SECNDS + DAY
START = SECNDS - T0
SECNDS = START
RETURN
END

*DATE
SUBROUTINE DATE(DAYT)
C
Last Update: February 7, 1991
CHARACTER DAYT*9, CMO(1:12)*3
INTEGER*2 IYR, IMO, IDA
INTEGER MOD
INTRINSIC MOD
*****
* A compatibility module: Vax VMS routine for IBM FORTRAN.
* Returns the system date, as a character string. This routine is very
* system dependent; furthermore the GETDAT call is specific to MS FORTRAN
* 4.x, so this may have to be re-coded for other systems or compilers. If
* trouble is encountered, simply comment out the call to this routine--the
* user may/should enter the time anyway.
*
* David E. Cawlfieid, Spring 91
*****
DATA CMO / 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
& 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec' /
CALL GETDAT(IYR, IMO, IDA)
IYR = MOD(IYR, 1900)
WRITE(DAYT, 2000) IDA, CMO(IMO), IYR
RETURN
2000 FORMAT(I2.2, '-', A, '-', I2.2)
END

*TIME
C
Last Revision: February 7, 1991
SUBROUTINE TIME(TYME)
CHARACTER TYME*8
INTEGER*2 IHR, IMIN, ISEC, IHDT
CHARACTER CHR(0:23)*2, CAM(0:23)*2, CMIN*2

```

```

*****
* A compatibility module: Vax VMS routine for IBM FORTRAN.
* Returns the system time, as a character string. This routine is very
* system dependent; furthermore the GETTIM call is specific to MS FORTRAN
* 4.x, so this may have to be re-coded for other systems or compilers. If
* trouble is encountered, simply comment out the call to this routine--the
* user may/should enter the time anyway. A simple table look-up seems to
* be the best route to go on this.
*
* David E. Cawlfieid, Fall 91
*****
DATA CHR/
& '12','01','02','03','04','05','06','07','08','09','10','11',
& '12','01','02','03','04','05','06','07','08','09','10','11'/
DATA CAM/ 12*'AM', 12*'PM'/
CALL GETTIM(IHR, IMIN, ISEC, IHDT)
WRITE(CMIN,'(I2.2)') IMIN
TYME = CHR(IHR) // ':' // CMIN // ' ' // CAM(IHR)
RETURN
END

*IARGC
INTEGER FUNCTION IARGC()
INTEGER NARGS
*****
* Provided to map the MS-FORTRAN 5.x call into the unix f77 world.
* Returns the number of arguments in the command line.
*
* David E. Cawlfieid, Spring 92
*****
IARGC = NARGS() - 1
RETURN
END

```

APPENDIX C

CURVATURE AND SMOOTH CUBIC SPLINE CONCEPTS USED IN CASCADR9

Introduction

Figure C.1 shows a plot of the first four days of barometric pressure data in (mb). This is three hours of averaged data from the National Oceanic and Atmospheric Administration (NOAA) META 5 Station in Area 5 of NTS. The highest sampling rate at META 5 is 15 minutes or 96 samples per day. It was found that using direct barometric pressure drive at the atmosphere-alluvium interface, where the barometric pressure was held constant for a prescribed length of time (3 hours in Figure C.1), followed by a step change (numerical differentiation done in CASCADR9 air pumping submodel) to the next pressure level produced spikes (mathematically Dirac δ -function values). The following two procedures were introduced into CASCADR9 to eliminate this numerical and mathematical artifice, which in the real world are not present. Figure C.2 shows barometric pressure over the same four-day period as obtained from 1,000 spline evaluations. The splined barometric data has a smooth effect compared to raw barometric pressure data.

Smooth Cubic Spline Method (Press et al., 1986)

Careful inspection of the four-day real barometric pressure data at the atmosphere-alluvium interface shows regions where the slope changes abruptly. Since the induced advection velocity used in CASCADR9 is a derived quantity (first partial derivative of $p[z,t]$, the air pressure field in the alluvium with respect to depth into the alluvium), any discontinuities in either $p|_{z=0}$ or $\frac{\partial p}{\partial z}|_{z=0}$ are reflected, in damped fashion, to any depth z . Hence, introducing the smooth cubic Lagrange spline discussed by Press et al., (1986) beginning on page 86. The entire set of barometric air pressure data, using a typical 15 minute or some other sampling rate data, is splined using their set of FORTRAN 77 subroutines. During the simulation run of CASCADR9, since the time, Δt , can be almost any positive real number, bounded above by a user supplied $\Delta t_{\max}$, the smooth cubic spline is used as a very smooth representation of the actual barometric pressure data. Figure C.3 shows the smooth cubic spline

Raw Barometric Data

First 4 days

Figure C.1 Raw barometric pressure data from the NOAA Weather Station META 5 in Area 5 of NTS.

Splined Barometric Data 1000 Pts Over First 4 days

Figure C.2 Splined barometric data - 1000 points generated over the first 4 days.

2 of 2

Splined Barometric Data 1000 Points

Figure C.3 Typical example of 40 days of splined real NOAA META 5 barometric pressure data (heavy line). Curvature function used in helping to choose local Δt values for accurate time stepping.

representation of 40 days of real NOAA META 5 barometric pressure data. The first four days overlap with the time scale shown on Figure C.1. The actual data points used in the spline (eight per day) are buried in the heavy line of the Quattro Pro plot (Figure C.3) and are therefore not visible.

The geometric concept of local curvature was included in CASCADR9 to retain accuracy in the computation by resetting Δt to small values in time regions where the rate of change of slope $\left. \frac{dP}{dt} \right|_{z=0}$ is rapid. Curvature as defined by Kaplan (1952) pages 62 and 63, for a univariate function of time t can be written as

$$\kappa_r(t) = \frac{y''(t)}{\left(\sqrt{1 + (y'(t))^2} \right)^3} \quad (C.1)$$

Recall that a particle executing constant radius circular motion has a constant curvature κ and $1/\kappa$ is said to be the radius of curvature. This aids in understanding the geometric significance of κ . For curves in space where κ is close to zero, the radius of curvature is very large and the curve is almost a straight line, where as κ is very large, the radius of curvature is very small and y' is changing rapidly from one sign to the other. Points on Figure C.3 where the curvature is large seem to correspond nicely with the rather sharp points on the pressure curve.

In CASCADR9 the absolute time units used are in seconds. However, the heuristic based upon $\kappa_r(t)$ uses time in day units for the derivatives. Also, barometric pressure is read in millibars. It is internally converted to dyne/cm². Therefore, the first derivative of the pressure is multiplied by 86,400 sec/day and divided by 1,000 to put the units back to (mb). The second derivative is multiplied once again by 86,400 sec/day. Furthermore, internally the pressure function $P(z,t)$ is represented as per Lindstrom et al., (1992a)

$$P(z,t) = \bar{P}_{air} e^{\frac{M_{air} g z}{R_g T}} + p(z,t) \quad (\text{dyne/cm}^2) \quad (C.2)$$

where positive z is vertically downward into the alluvium.

At the atmosphere-alluvium interface there is

$$P(0,t) = \bar{P}_{air} + p(0,t) \quad . \quad (\text{dyne/cm}^2) \quad (C.3)$$

Thus, the "average" barometric pressure $\bar{P}_{air}$ is split out and $p(0,t)$ carries all the time variation in the pressure field. Therefore, defining

$$y(t) = (\bar{P}_{air} + p(0,t)) / 1,000 \quad . \quad (\text{mb}) \quad (C.4)$$

In equation C.1 y' and y'' are required. Hence, with

$$y'(t) = 86,400 \, p'(0,t) / 1,000 \quad (\text{mb/day}) \quad (C.5)$$

and

$$y'' = (86,400)^2 \, p''(0,t) / 1,000, \quad (\text{mb/day}^2) \quad (C.6)$$

therefore,

$$\kappa_r(t) = \frac{\left(\frac{(86,400)^2 \, p''(0,t)}{1,000} \right)}{\left(\sqrt{1 + \left(\frac{86,400 \, p'(0,t)}{1,000} \right)^2} \right)^3} \quad (C.7)$$

is obtained, which is the formula used in CASCADR9.

Based upon the FORTRAN 77 subroutine SPLINT (...) (Press et al., 1986)

$$\begin{aligned} P &= A(t) \, P_A(KLO) + B(t) \, P_A(KHI) \\ &+ \frac{\left((A^3(t) - A(t)) \, P_{2A}(KLO) + (B^3(t) - B(t)) \, P_{2A}(KHI) \right) H^2}{6} \end{aligned} \quad (C.8)$$

is defined,

$$A(t) = (X_A(KHI) - t)/H \quad (C.9)$$

$$B(t) = (t - X_A(KHO))/H. \quad (C.10)$$

In the above spline interpolation formula $p_A(KLO)$ and $p_A(KHI)$ are direct pressure interpolation values at knot points $X_A(KLO)$ and $X_A(KHI)$ respectively. The coefficients $p_{2A}(KLO)$ and $p_{2A}(KHI)$ are obtained from the solution of the associated linear system of interpolation equations. They are related to the matching of the local curvatures at $X_A(KLO)$ and $X_A(KHI)$. $H = X_A(KHI) - X_A(KLO)$. H is simply the length of the subinterval $[X_A(KHO), X_A(KHI)]$. The local time variable t also lies in this same interval. Recall that the smooth cubic Lagrange not only interpolates exactly at the given knot (data) points, but both first and second derivatives are continuous across each knot. This makes for a smooth (numerically and mathematically) curvature function κ_r .

The formula for p' becomes by direct differentiation

$$p'(t) = \frac{p_A(KHI) - p_A(KLO)}{H} + \frac{\left((3B^2(t) - 1) p_{2A}(KHI) - (3A^2(t) - 1) p_{2A}(KHO) \right) H}{6}, \quad (C.11)$$

and the formula for $p''(t)$ becomes by direct differentiation

$$p''(t) = A(t) p_{2A}(KLO) + B(t) p_{2A}(KHI). \quad (C.12)$$

Formulas C.11 and C.12 are then substituted into C.7 at each acceptable time of the model.

The heuristic based on the magnitude of κ_r is given in subroutine KRAZYC (. . .). The selection rules are as follows:

- 1) if $|\kappa_r| < 50.0$, leave Δt alone; if not,
- 2) if $50 \leq |\kappa_r| \leq 100$, replace Δt by $\frac{\Delta t}{8.0}$
- 3) if $100 \leq |\kappa_r| \leq 150.0$, replace Δt by $\frac{\Delta t}{16.0}$; if not
- 4) if $|\kappa_r| > 150.0$, replace Δt by $\frac{\Delta t}{32.0}$.

These values of Δt are held fixed for an integer number of times. This integer is called IHDTK and is defined via the formula (integer function)

$$\text{IHDTK} = \text{NINT} ((T_3 - T_1) / \Delta t) + 1$$

where T_1 is that point on the master time line where $|\kappa_r|$ is so large, as per the above tests, that Δt must be adjusted downward. T_3 is that point, looking ahead, on the master time line where $|\kappa_r|$ just comes back into "compliance." Thus, it is seen on the splined barometric pressure curve where $|\kappa|$ is reduced back to one of the above test values. After stepping the model on ahead at the greatly reduced Δt value, T_3 is finally reached in running time. At this point, a second simple heuristic for expanding Δt by powers of 2.0 until the Δt value used before the shrinking heuristic was invoked, is called. Thereby, Δt can be automatically shrunk and expanded to keep the accuracy of the simulation high. These heuristics are fully compatible with the Courant Checker subroutine required for stability purposes. Any Δt reset via the curvature heuristic is subordinate to the Courant Checker.

APPENDIX D

CHAIN3: AN EXTENDED MULTI-PURPOSE BATEMAN CHAIN MODEL

Introduction

Appendices B and C of Lindstrom et al., (1992b) include the chemical mass based Bateman chain and a simple Bateman chain model for air concentration in a standard 2500 ft² house, respectively. These two concepts have been wedded to form a multi-purpose Bateman chain model called CHAIN3. The below detailed version is chemically mass based. However, by inputting the actual specific activity (pCi/gm) of each radionuclide, the output is a time distribution of the activity of the first six radionuclides in the Bateman chain. The model has many possible modes of operating, three of which are: 1) straight Bateman chain of six radionuclides either chemically mass based or specific activity based starting with either nonzero or zero (except for the first radionuclide) initial mass data; 2) constant flux of radon-222 into a ventilated (or non-ventilated) house of prescribed volume, floor surface area, and room air changes per unit time; 3) 1 and 2 run in conjunction.

The chemical mass mode is obtained when all specific activities and the room air volume are set to 1.0. Otherwise the output is a table of the time course of each radionuclide in specific activity concentrations units (pCi/cm³ air). For the details and nomenclature of the following mathematical balance equations the reader should see appendices B and C of Lindstrom et al., (1992b).

Balance Equations (Figure D.1)

Radionuclide 1 (parent compound)

$$\frac{dC_0}{dt} = \frac{q_{C_0} \text{ Area}}{\text{Vol}} - \left(\lambda_0 + \frac{R_f}{\text{Vol}} \right) C_0, \quad C_0(0) = C_{00} \quad (\text{gm/cm}^3 \text{ air}) \quad (\text{D.1})$$

Figure D.1 General structure of multi-purpose Bateman chain model CHAIN3.

Radionuclide k, k = 2, 3, . . . 6 (Daughters)

$$\frac{dC_K}{dt} = \lambda_{K-1} \left(\frac{M_K}{M_{K-1}} \right) C_{K-1} - \left(\lambda_K + \frac{R_f}{Vol} \right) C_K, \quad C_K(0) = C_{K0} \quad (\text{gm/cm}^3 \text{ air}) \quad (\text{D.2})$$

Recall that the relationship between the chemical mass concentration of radionuclide k, C_K and the number of radionuclide k atoms per unit volume of air is

$$C_K = \frac{N_K M_K}{A_0}, \quad (\text{D.3})$$

where M_k is the atomic mass (g/mole) of radionuclide k and A_0 is Avogadro's number (6.023×10^{23} atoms/mole).

Solution Method

The solution method for solving the first six equations in the coupled system of equations (CASCADE system) is that of Laplace transforms (Wylie, 1960; Chapter 8 and McLachlan, 1963). Transforming equations D.1 and D.2 coupled with the initial data obtains the first six radionuclides (s is the Laplace transform variable):

Radionuclide 1 (parent)

$$\hat{C}_0(s) = \frac{C_{00}}{s + \lambda_0 + \frac{R_f}{Vol}} + \frac{q_{C_0} \frac{\text{Area}}{Vol}}{s \left(s + \lambda_0 + \frac{R_f}{Vol} \right)} \quad (\text{D.4})$$

Radionuclide 2 (1st daughter)

$$\begin{aligned} \hat{C}_1(s) = & \frac{C_{10}}{s + \lambda_1 + \frac{R_f}{Vol}} + \lambda_0 \left(\frac{M_1}{M_0} \right) \frac{C_{00}}{\left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right)} \\ & + \lambda_0 \left(\frac{M_1}{M_0} \right) \frac{q_{C_0} \frac{\text{Area}}{Vol}}{s \left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right)} \end{aligned} \quad (\text{D.5})$$

Radionuclide 3 (2nd daughter)

$$\hat{C}_2(s) = \frac{C_{20} \frac{R_f}{M_1}}{s + \lambda_2 + \frac{R_f}{Vol}} + \lambda_1 \left(\frac{M_2}{M_1} \right) \frac{C_{10}}{\left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right)} + \lambda_0 \lambda_1 \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \frac{C_{10}}{\left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right)}$$

$$+ \lambda_0 \lambda_1 \left(\frac{M_1}{M_0} \right) \left(\frac{M_2}{M_1} \right) \frac{\frac{\text{Area}}{q_{C_0}} \frac{1}{Vol}}{s \left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right)} \quad (D.6)$$

Radionuclide 4 (3rd daughter)

$$\hat{C}_3(s) = \frac{C_{30} \frac{R_f}{M_2}}{s + \lambda_3 + \frac{R_f}{Vol}} + \lambda_2 \left(\frac{M_3}{M_2} \right) \frac{C_{20}}{\left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right)} + \lambda_1 \lambda_2 \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \frac{C_{10}}{\left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right)}$$

$$+ \lambda_0 \lambda_1 \lambda_2 \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \frac{C_{00}}{\left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right)}$$

$$+ \lambda_0 \lambda_1 \lambda_2 \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \frac{\frac{\text{Area}}{q_{C_0}} \frac{1}{Vol}}{s \left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right)} \quad (D.7)$$

Radionuclide 5 (4th daughter)

$$\begin{aligned}
 \hat{C}_4(s) = & \frac{C_{40}}{s + \lambda_4 + \frac{R_f}{Vol}} + \lambda_3 \left(\frac{M_4}{M_3} \right) \frac{C_{30}}{\left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right)} + \lambda_2 \lambda_3 \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \frac{C_{20}}{\left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right)} \\
 & + \lambda_1 \lambda_2 \lambda_3 \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \frac{C_{10}}{\left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right)} \\
 & + \lambda_0 \lambda_1 \lambda_2 \lambda_3 \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \frac{C_{00}}{\left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right)} \\
 & + \lambda_0 \lambda_1 \lambda_2 \lambda_3 \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \frac{q_{C_0} \frac{Area}{Vol}}{s \left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right)}
 \end{aligned}$$

(D.8)

Radionuclide 6 (5th daughter)

$$\begin{aligned}
 \hat{C}_5(s) = & \frac{C_{50}}{s + \lambda_5 + \frac{R_f}{Vol}} + \lambda_4 \left(\frac{M_5}{M_4} \right) \left[\frac{C_{40}}{\left(s + \lambda_4 + \frac{R_f}{Vol} \right)} + \lambda_3 \lambda_4 \left(\frac{M_5}{M_4} \right) \left(\frac{M_4}{M_3} \right) \left[\frac{C_{30}}{\left(s + \lambda_3 + \frac{R_f}{Vol} \right)} \left(s + \lambda_4 + \frac{R_f}{Vol} \right) \left(s + \lambda_5 + \frac{R_f}{Vol} \right) \right] \right. \\
 & + \lambda_2 \lambda_3 \lambda_4 \left(\frac{M_5}{M_4} \right) \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left[\frac{C_{20}}{\left(s + \lambda_2 + \frac{R_f}{Vol} \right)} \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right) \left(s + \lambda_5 + \frac{R_f}{Vol} \right) \right] \\
 & + \lambda_1 \lambda_2 \lambda_3 \lambda_4 \left(\frac{M_5}{M_4} \right) \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left[\frac{C_{10}}{\left(s + \lambda_1 + \frac{R_f}{Vol} \right)} \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right) \left(s + \lambda_5 + \frac{R_f}{Vol} \right) \right] \\
 & + \lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \left(\frac{M_5}{M_4} \right) \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \left[\frac{C_{00}}{\left(s + \lambda_0 + \frac{R_f}{Vol} \right)} \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right) \left(s + \lambda_5 + \frac{R_f}{Vol} \right) \right] \\
 & + \lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \left(\frac{M_5}{M_4} \right) \left(\frac{M_4}{M_3} \right) \left(\frac{M_3}{M_2} \right) \left(\frac{M_2}{M_1} \right) \left(\frac{M_1}{M_0} \right) \left[\frac{\text{Area}}{q_{C_0} \frac{\text{Area}}{Vol}} \left(s + \lambda_0 + \frac{R_f}{Vol} \right) \left(s + \lambda_1 + \frac{R_f}{Vol} \right) \left(s + \lambda_2 + \frac{R_f}{Vol} \right) \left(s + \lambda_3 + \frac{R_f}{Vol} \right) \left(s + \lambda_4 + \frac{R_f}{Vol} \right) \left(s + \lambda_5 + \frac{R_f}{Vol} \right) \right]
 \end{aligned}$$

(D.9)

Application of the inversion theorem $C(t) = \frac{1}{2\pi i} \int_{Br_1} e^{st} \hat{c}(s) ds$ to each distribution obtains:

Radionuclide 1 (parent)

$$C_0(t) = C_{00} e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t} + \left(q_{C_0} \frac{Area}{Vol}\right) \left(\frac{1}{\lambda_0 + \frac{R_f}{Vol}} - \frac{e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{\lambda_0 + \frac{R_f}{Vol}} \right) \quad (D.10)$$

Radionuclide 2 (1st daughter)

$$C_1(t) = C_{10} e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t} + \left(\frac{M_1}{M_0}\right) C_{00} \left(\frac{\lambda_0 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{\lambda_1 - \lambda_0} + \frac{\lambda_0 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{\lambda_0 - \lambda_1} \right) \\ + \left(\frac{M_1}{M_0}\right) q_{C_0} \frac{Area}{Vol} \left(\frac{\lambda_0}{\left(\lambda_0 + \frac{R_f}{Vol}\right)\left(\lambda_1 + \frac{R_f}{Vol}\right)} + \frac{\lambda_0 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_0 + \frac{R_f}{Vol}\right)(\lambda_1 - \lambda_0)} + \frac{\lambda_0 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_1 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_1)} \right) \quad (D.11)$$

Radionuclide 3 (2nd daughter)

$$\begin{aligned}
 C_2(t) = & C_{20} e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t} + \left(\frac{M_2}{M_1}\right) C_{11} \left(\frac{\lambda_1 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{\lambda_2 - \lambda_1} + \frac{\lambda_1 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{\lambda_1 - \lambda_2} \right) \\
 & + \left(\frac{M_2}{M_0}\right) C_{00} \left(\frac{\lambda_0 \lambda_1 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)} + \frac{\lambda_0 \lambda_1 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)} + \frac{\lambda_0 \lambda_1 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)} \right) \\
 & + \left(\frac{M_2}{M_0}\right) q_{C_0} \frac{Area}{Vol} \left(\frac{\lambda_0 \lambda_1}{\left(\lambda_0 + \frac{R_f}{Vol}\right)\left(\lambda_1 + \frac{R_f}{Vol}\right)\left(\lambda_2 + \frac{R_f}{Vol}\right)} \right. \\
 & \left. + \frac{\lambda_0 \lambda_1 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_0 + \frac{R_f}{Vol}\right)(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)} + \frac{\lambda_0 \lambda_1 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_1 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)} + \frac{\lambda_0 \lambda_1 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_2 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)} \right)
 \end{aligned}
 \tag{D.12}$$

Radionuclide 4 (3rd daughter)

$$\begin{aligned}
 C_3(t) = & C_{30} e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t} + \left(\frac{M_3}{M_2}\right) C_{20} \left(\frac{\lambda_2 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{\lambda_3 - \lambda_2} + \frac{\lambda_2 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{\lambda_2 - \lambda_3} \right) \\
 & + \left(\frac{M_3}{M_1}\right) C_{10} \left(\frac{\lambda_1 \lambda_2 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} + \frac{\lambda_1 \lambda_2 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{\lambda_1 \lambda_2 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} \right) \\
 & + \left(\frac{M_3}{M_0}\right) C_{00} \left(\frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)(\lambda_3 - \lambda_0)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} \right) \\
 & + \left(\frac{M_3}{M_0}\right) q_{c0} \frac{Area}{Vol} \left(\frac{\lambda_0 \lambda_1 \lambda_2}{\left(\lambda_0 + \frac{R_f}{Vol}\right)\left(\lambda_1 + \frac{R_f}{Vol}\right)\left(\lambda_2 + \frac{R_f}{Vol}\right)\left(\lambda_3 + \frac{R_f}{Vol}\right)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_0 + \frac{R_f}{Vol}\right)(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)(\lambda_3 - \lambda_0)} \right. \\
 & \left. + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_1 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_2 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_3 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)} \right)
 \end{aligned}$$

(D.13)

Radionuclide 5 (4th daughter)

$$\begin{aligned}
 C_4(t) = & C_{40} e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t} + \left(\frac{M_4}{M_3}\right) C_{30} \left(\frac{\lambda_3 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{\lambda_4 - \lambda_3} + \frac{\lambda_3 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{\lambda_3 - \lambda_4} \right) \\
 & + \left(\frac{M_4}{M_2}\right) C_{20} \left(\frac{\lambda_2 \lambda_3 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)} + \frac{\lambda_2 \lambda_3 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)} + \frac{\lambda_2 \lambda_3 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)} \right) \\
 & + \left(\frac{M_4}{M_1}\right) C_{10} \left(\frac{\lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_4 - \lambda_1)} + \frac{\lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)} + \frac{\lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)} + \frac{\lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)} \right) \\
 & + \left(\frac{M_4}{M_0}\right) C_{00} \left(\frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)(\lambda_3 - \lambda_0)(\lambda_4 - \lambda_0)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_4 - \lambda_1)} \right. \\
 & \left. + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_4)(\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)} \right)
 \end{aligned}$$

$$\begin{aligned}
& + \left(\frac{M_4}{M_0} \right) q_{C_0} \frac{\text{Area}}{\text{Vol}} \left[\frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3}{\left(\lambda_0 + \frac{R_f}{\text{Vol}} \right) \left(\lambda_1 + \frac{R_f}{\text{Vol}} \right) \left(\lambda_2 + \frac{R_f}{\text{Vol}} \right) \left(\lambda_3 + \frac{R_f}{\text{Vol}} \right) \left(\lambda_4 + \frac{R_f}{\text{Vol}} \right)} \right. \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_0 + \frac{R_f}{\text{Vol}} \right) t}}{-\left(\lambda_0 + \frac{R_f}{\text{Vol}} \right) (\lambda_1 - \lambda_0) (\lambda_2 - \lambda_0) (\lambda_3 - \lambda_0) (\lambda_4 - \lambda_0)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_1 + \frac{R_f}{\text{Vol}} \right) t}}{-\left(\lambda_1 + \frac{R_f}{\text{Vol}} \right) (\lambda_0 - \lambda_1) (\lambda_2 - \lambda_1) (\lambda_3 - \lambda_1) (\lambda_4 - \lambda_1)} \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_2 + \frac{R_f}{\text{Vol}} \right) t}}{-\left(\lambda_2 + \frac{R_f}{\text{Vol}} \right) (\lambda_0 - \lambda_2) (\lambda_1 - \lambda_2) (\lambda_3 - \lambda_2) (\lambda_4 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_3 + \frac{R_f}{\text{Vol}} \right) t}}{-\left(\lambda_3 + \frac{R_f}{\text{Vol}} \right) (\lambda_0 - \lambda_3) (\lambda_1 - \lambda_3) (\lambda_2 - \lambda_3) (\lambda_4 - \lambda_3)} \\
& \left. + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 e^{-\left(\lambda_4 + \frac{R_f}{\text{Vol}} \right) t}}{-\left(\lambda_4 + \frac{R_f}{\text{Vol}} \right) (\lambda_0 - \lambda_4) (\lambda_1 - \lambda_4) (\lambda_2 - \lambda_4) (\lambda_3 - \lambda_4)} \right]
\end{aligned}$$

(D.14)

Radionuclide 6 (5th daughter)

$$\begin{aligned}
 C_5(t) = & C_{50} e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t} + \left(\frac{M_5}{M_4}\right) C_{40} \left(\frac{\lambda_4 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{\lambda_5 - \lambda_4} + \frac{\lambda_4 e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{\lambda_4 - \lambda_5} \right) \\
 & + \left(\frac{M_5}{M_3}\right) C_{30} \left(\frac{\lambda_3 \lambda_4 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_4 - \lambda_3)(\lambda_5 - \lambda_3)} + \frac{\lambda_3 \lambda_4 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_3 - \lambda_4)(\lambda_5 - \lambda_4)} + \frac{\lambda_3 \lambda_4 e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{(\lambda_3 - \lambda_5)(\lambda_4 - \lambda_5)} \right) \\
 & + \left(\frac{M_5}{M_2}\right) C_{20} \left(\frac{\lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)(\lambda_5 - \lambda_2)} + \frac{\lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)(\lambda_5 - \lambda_3)} + \frac{\lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)(\lambda_5 - \lambda_4)} + \frac{\lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_5)(\lambda_3 - \lambda_5)(\lambda_4 - \lambda_5)} \right) \\
 & + \left(\frac{M_5}{M_1}\right) C_{10} \left(\frac{\lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_4 - \lambda_1)(\lambda_5 - \lambda_1)} + \frac{\lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)(\lambda_5 - \lambda_2)} \right. \\
 & \left. + \frac{\lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)(\lambda_5 - \lambda_3)} + \frac{\lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)(\lambda_5 - \lambda_4)} + \frac{\lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_5)(\lambda_2 - \lambda_5)(\lambda_3 - \lambda_5)(\lambda_4 - \lambda_5)} \right)
 \end{aligned}$$

$$\begin{aligned}
& + \left(\frac{M_5}{M_0} \right) C_{00} \left(\frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{(\lambda_1 - \lambda_0)(\lambda_2 - \lambda_0)(\lambda_3 - \lambda_0)(\lambda_4 - \lambda_0)(\lambda_5 - \lambda_0)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_1)(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_1)(\lambda_4 - \lambda_1)(\lambda_5 - \lambda_1)} \right. \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)(\lambda_5 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)(\lambda_5 - \lambda_3)} \Bigg) \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_4)(\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)(\lambda_5 - \lambda_4)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{(\lambda_0 - \lambda_5)(\lambda_1 - \lambda_5)(\lambda_2 - \lambda_5)(\lambda_3 - \lambda_5)(\lambda_4 - \lambda_5)} \Bigg)
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{M_5}{M_0} \right) q_{C_0} \frac{Area}{Vol} \left(\frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4}{\left(\lambda_0 + \frac{R_f}{Vol} \right) \left(\lambda_1 + \frac{R_f}{Vol} \right) \left(\lambda_2 + \frac{R_f}{Vol} \right) \left(\lambda_3 + \frac{R_f}{Vol} \right) \left(\lambda_4 + \frac{R_f}{Vol} \right) \left(\lambda_5 + \frac{R_f}{Vol} \right)} \right. \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_0 + \frac{R_f}{Vol}\right)t}}{\left(\lambda_1 - \lambda_0 \right) \left(\lambda_2 - \lambda_0 \right) \left(\lambda_3 - \lambda_0 \right) \left(\lambda_4 - \lambda_0 \right) \left(\lambda_5 - \lambda_0 \right)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 e^{-\left(\lambda_1 + \frac{R_f}{Vol}\right)t}}{\left(\lambda_0 - \lambda_1 \right) \left(\lambda_2 - \lambda_1 \right) \left(\lambda_3 - \lambda_1 \right) \left(\lambda_4 - \lambda_1 \right) \left(\lambda_5 - \lambda_1 \right)} \Bigg)
\end{aligned}$$

$$\begin{aligned}
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \theta^{-\left(\lambda_2 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_2 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_2)(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)(\lambda_4 - \lambda_2)(\lambda_5 - \lambda_2)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \theta^{-\left(\lambda_3 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_3 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_3)(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)(\lambda_4 - \lambda_3)(\lambda_5 - \lambda_3)} \\
& + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \theta^{-\left(\lambda_4 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_4 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_4)(\lambda_1 - \lambda_4)(\lambda_2 - \lambda_4)(\lambda_3 - \lambda_4)(\lambda_5 - \lambda_4)} + \frac{\lambda_0 \lambda_1 \lambda_2 \lambda_3 \lambda_4 \theta^{-\left(\lambda_5 + \frac{R_f}{Vol}\right)t}}{-\left(\lambda_5 + \frac{R_f}{Vol}\right)(\lambda_0 - \lambda_5)(\lambda_1 - \lambda_5)(\lambda_2 - \lambda_5)(\lambda_3 - \lambda_5)(\lambda_4 - \lambda_5)} \Bigg)
\end{aligned}$$

(D.15)

One typical use for the CHAIN3 is to estimate the average room air radon concentration in (gm/cm^3) or (pCi/cm^3). Suppose that, in this case, q_{co} is specified in ($\text{gm}/\text{cm}^2\text{-sec}$) of radon-222 gas entering the room through the floor, whose surface area is given in cm^2 . The average room air change rate is modeled via R_f (cm^3/sec) and the volume (Vol [cm^3]) via the ratio R_f/Vol (1/sec). The half lives of the six radionuclides in the Vol are specified in terms of seconds. The decay constants (λ 's) are calculated internally. The user may or may not choose to specify nonzero initial air concentrations of the six radionuclides. If all the initial concentrations (gm/cm^3) are zero then the user needs to specify all six initial values as zero (double precision floating point zero). If there is no room air change then R_f is simply set to zero. The room air is assumed to be instantaneous well mixed (stirred batch reaction vessel). By specifying the specific activities all as 1.0, the room air-phase concentrations are all in chemical mass terms as (gm/cm^3). Specifying each radionuclide specific activity results in the output tables being in (pCi/cm^3).

A second typical use of CHAIN3 is estimating a straight Bateman decay chain such as that required in estimating ingrowth of the radionuclides. In this case $R_f = 0.0$, $\text{area} = 0.0$, $\text{Vol} = 1.0$, $q_{\text{co}} = 0.0$. The user may either specify the initial concentrations (masses in this case) of each radionuclide in the chain or simply choose all but the first one equal to zero. Again the output in the tables may be interpreted as either an ingrowth chemical mass or specific activity ingrowth depending upon whether all the SA's are set to 1.0 or their actual specific activity values (pCi/gm), respectively.

APPENDIX E

BATEMAN CASCADE CHAIN FOR SIX RADIONUCLIDES

* Model name: CHAIN3
* Version number: 1.0
* Date last worked on: 08-04-92
* Key descriptors:

*

*

* BATEMAN CASCADE CHAIN FOR SIX RADIONUCLIDES
* CHEMICAL MASS FORMULATION (GMS)
* ATOMIC MASS QUOTIENTS CORRECTED FORMS
*

*

* GENERAL CASE OF RADIONUC SERIES IN A HOUSE (UP TO 6 RADIONUCS):
* 1) CONSTANT PARENT INPUT FLUX ACROSS AN AREA.
* 2) PRESCRIBED INITIAL CONC. OF EACH RADIONUC.
* 3) CONSTANT VENTILATION RATE
*

C*****

C This model is the joint effort of: F. T. Lindstrom,
C D. E. Cawlfeld, D. F. Emer, and G. J. Shott

C

C Special Projects Group
C Waste Management Department
C Environment, Health, and Safety Division
C Reynolds Electrical and Engineering Co., Inc.
C P O Box 98521 M/S 738
C Las Vegas, NV 89193-8521

C*****

C

PROGRAM CHAIN3

C

DOUBLE PRECISION AMCOEF,BMCOEF
DOUBLE PRECISION CON10,CON11,CON12,CON13,CON14,CON15,CON16
DOUBLE PRECISION CON20,CON21,CON22,CON23,CON24,CON25,CON26
DOUBLE PRECISION CON30,CON31,CON32,CON33,CON34,CON35,CON36
DOUBLE PRECISION CON311
DOUBLE PRECISION CON40,CON41,CON42,CON43,CON44,CON45,CON46
DOUBLE PRECISION CON411,CON421,CON422,CON423
DOUBLE PRECISION CON511,CON521,CON522,CON523,CON531,CON532,
& CON533,CON534
DOUBLE PRECISION CON611,CON621,CON622,CON623,CON631,CON632,
& CON633,CON634,CON641,CON642,CON643,CON644,CON645
DOUBLE PRECISION CON50,CON51,CON52,CON53,CON54,CON55,CON56
DOUBLE PRECISION CON60,CON61,CON62,CON63,CON64,CON65,CON66
DOUBLE PRECISION XLAM,TOH,VOL,SA,RFLOW
DOUBLE PRECISION TIME,DT,QFLUX,AREA,ATM,ATMR,E1,E2,E3,E4,E5,E6
DOUBLE PRECISION CMASS,C0
INTEGER I,J,NEVAL
CHARACTER*8 ANS

C

COMMON/DAT1/ AMCOEF(10,10),BMCOEF(10,10)
COMMON/DAT2/ CON10,CON11,CON12,CON13,CON14,CON15,CON16


```

COMMON/DATA3/ CON20,CON21,CON22,CON23,CON24,CON25,CON26
COMMON/DATA4/ CON30,CON31,CON32,CON33,CON34,CON35,CON36
COMMON/DATA5/ CON40,CON41,CON42,CON43,CON44,CON45,CON46
COMMON/DATA6/ CON50,CON51,CON52,CON53,CON54,CON55,CON56
COMMON/DATA6/ CON60,CON61,CON62,CON63,CON64,CON65,CON66
COMMON/DATA7/ XLAM(10),TOH(10),SA(6),RFLOW
COMMON/DATA8/ TIME,DT,QFLUX,AREA,VOL,ATM(10),ATMR(10),C0(10)
COMMON/DATA9/ CMASS(10)
COMMON/DATA10/ CON311,CON411,CON421,CON422,CON423
COMMON/DATA11/ CON511,CON521,CON522,CON523,CON531,CON532,
& CON533,CON534
COMMON/DATA12/ CON611,CON621,CON622,CON623,CON631,CON632,
& CON633,CON634,CON641,CON642,CON643,CON644,CON645

C      OPEN(UNIT=20,FILE='          ',STATUS='OLD')
      OPEN(UNIT=30,FILE='          ',STATUS='NEW')

C      WRITE(*,*)' READ IN QFLUX,RFLOW,AREA,VOL,DT, AND NEVAL '
      READ(*,*) QFLUX,RFLOW,AREA,VOL,DT,NEVAL

C      WRITE(30,*)' QFLUX = ',QFLUX
      WRITE(30,*)' RFLOW = ',RFLOW
      WRITE(30,*)' AREA = ',AREA
      WRITE(30,*)' VOL = ',VOL
      WRITE(30,*)' DT = ',DT
      WRITE(30,*)' NEVAL = ',NEVAL
      WRITE(30,*)

C      DO 513 I=1,6
      READ(20,*)TOH(I),ATM(I),SA(I),C0(I)
513    CONTINUE
      DO 514 I=1,6
      XLAM(I)=0.693D0/TOH(I)
514    CONTINUE

C      WRITE OUT THE HALF LIVES AND THE DECAY CONSTANTS
C
C      WRITE(*,*)' HALF LIVES, DECAY CONSTANTS, AND SA TABLE '
      WRITE(30,*)' HALF LIVES, DECAY CONSTANTS, AND SA TABLE '
      WRITE(30,*)
      DO 515 I=1,6
      WRITE(*,*)I,TOH(I),XLAM(I),SA(I),C0(I)
      WRITE(30,*)I,TOH(I),XLAM(I),SA(I),C0(I)
515    CONTINUE
      WRITE(*,*)
      WRITE(30,*)

C      UNIVERSAL CONSTANTS
C
C      ATMR(1)=1.0D0
      ATMR(2)=ATM(2)/ATM(1)
      ATMR(3)=ATM(3)/ATM(1)
      ATMR(4)=ATM(4)/ATM(1)
      ATMR(5)=ATM(5)/ATM(1)
      ATMR(6)=ATM(6)/ATM(1)

C      CON10=XLAM(1)+RFLOW/VOL
      AMCOEF(1,1)=1.0D0/CON10
      AMCOEF(1,2)=-1.0D0/CON10

```

```

AMCOEF(1,3)=0.OD0
AMCOEF(1,4)=0.OD0
AMCOEF(1,5)=0.OD0
AMCOEF(1,6)=0.OD0
AMCOEF(1,7)=0.OD0
BMCOEF(1,1)=C0(1)
BMCOEF(1,2)=0.OD0
BMCOEF(1,3)=0.OD0
BMCOEF(1,4)=0.OD0
BMCOEF(1,5)=0.OD0
BMCOEF(1,6)=0.OD0
BMCOEF(1,7)=0.OD0

```

C

```

CON20=XLAM(2)+RFLOW/VOL
CON21=XLAM(2)-XLAM(1)
AMCOEF(2,1)=XLAM(1)/(CON10*CON20)
AMCOEF(2,2)=-XLAM(1)/(CON10*CON21)
AMCOEF(2,3)=XLAM(1)/(CON20*CON21)
AMCOEF(2,4)=0.OD0
AMCOEF(2,5)=0.OD0
AMCOEF(2,6)=0.OD0
AMCOEF(2,7)=0.OD0
BMCOEF(2,1)=XLAM(1)*ATMR(2)*C0(1)/CON21
BMCOEF(2,2)=C0(2)-XLAM(1)*ATMR(2)*C0(1)/CON21
BMCOEF(2,3)=0.OD0
BMCOEF(2,4)=0.OD0
BMCOEF(2,5)=0.OD0
BMCOEF(2,6)=0.OD0
BMCOEF(2,7)=0.OD0

```

C

```

CON30=XLAM(3)+RFLOW/VOL
CON31=(XLAM(2)-XLAM(1))*(XLAM(3)-XLAM(1))
CON32=(XLAM(1)-XLAM(2))*(XLAM(3)-XLAM(2))
CON33=(XLAM(1)-XLAM(3))*(XLAM(2)-XLAM(3))
CON311=XLAM(3)-XLAM(2)
AMCOEF(3,1)=XLAM(1)*XLAM(2)/(CON10*CON20*CON30)
AMCOEF(3,2)=-XLAM(1)*XLAM(2)/(CON10*CON31)
AMCOEF(3,3)=-XLAM(1)*XLAM(2)/(CON20*CON32)
AMCOEF(3,4)=-XLAM(1)*XLAM(2)/(CON30*CON33)
AMCOEF(3,5)=0.OD0
AMCOEF(3,6)=0.OD0
AMCOEF(3,7)=0.OD0
BMCOEF(3,1)=XLAM(1)*XLAM(2)*ATMR(3)*C0(1)/CON31
BMCOEF(3,2)=XLAM(1)*XLAM(2)*ATMR(3)*C0(1)/CON32 +
& XLAM(2)*C0(2)*(ATM(3)/ATM(2))/CON311
BMCOEF(3,3)=XLAM(1)*XLAM(2)*ATMR(3)*C0(1)/CON33 -
& XLAM(2)*C0(2)*(ATM(3)/ATM(2))/CON311 +C0(3)
BMCOEF(3,4)=0.OD0
BMCOEF(3,5)=0.OD0
BMCOEF(3,6)=0.OD0
BMCOEF(3,7)=0.OD0

```

C

```

CON40=XLAM(4)+RFLOW/VOL
CON41=CON31*(XLAM(4)-XLAM(1))
CON42=CON32*(XLAM(4)-XLAM(2))
CON43=CON33*(XLAM(4)-XLAM(3))
CON44=(XLAM(1)-XLAM(4))*(XLAM(2)-XLAM(4))*(XLAM(3)-XLAM(4))
CON411=XLAM(4)-XLAM(3)

```

```

CON421=(XLAM(3)-XLAM(2))*(XLAM(4)-XLAM(2))
CON422=(XLAM(2)-XLAM(3))*(XLAM(4)-XLAM(3))
CON423=(XLAM(2)-XLAM(4))*(XLAM(3)-XLAM(4))
AMCOEF(4,1)=XLAM(1)*XLAM(2)*XLAM(3)/(CON10*CON20*CON30*CON40)
AMCOEF(4,2)=-XLAM(1)*XLAM(2)*XLAM(3)/(CON10*CON41)
AMCOEF(4,3)=-XLAM(1)*XLAM(2)*XLAM(3)/(CON20*CON42)
AMCOEF(4,4)=-XLAM(1)*XLAM(2)*XLAM(3)/(CON30*CON43)
AMCOEF(4,5)=-XLAM(1)*XLAM(2)*XLAM(3)/(CON40*CON44)
AMCOEF(4,6)=0.OD0
AMCOEF(4,7)=0.OD0
BMCOEF(4,1)=XLAM(1)*XLAM(2)*XLAM(3)*ATMR(4)*C0(1)/CON41
BMCOEF(4,2)=XLAM(1)*XLAM(2)*XLAM(3)*ATMR(4)*C0(1)/CON42 +
& XLAM(2)*XLAM(3)*C0(2)*(ATM(4)/ATM(2))/CON421
BMCOEF(4,3)=XLAM(1)*XLAM(2)*XLAM(3)*ATMR(4)*C0(1)/CON43 +
& XLAM(2)*XLAM(3)*C0(2)*(ATM(4)/ATM(2))/CON422 +
& XLAM(3)*C0(3)*(ATM(4)/ATM(3))/CON411
BMCOEF(4,4)=XLAM(1)*XLAM(2)*XLAM(3)*ATMR(4)*C0(1)/CON44 +
& XLAM(2)*XLAM(3)*C0(2)*(ATM(4)/ATM(2))/CON423 -
& XLAM(3)*C0(3)*(ATM(4)/ATM(3))/CON411 +C0(4)
BMCOEF(4,5)=0.OD0
BMCOEF(4,6)=0.OD0
BMCOEF(4,7)=0.OD0

```

C

```

CON50=XLAM(5)+RFLOW/VOL
CON51=CON41*(XLAM(5)-XLAM(1))
CON52=CON42*(XLAM(5)-XLAM(2))
CON53=CON43*(XLAM(5)-XLAM(3))
CON54=CON44*(XLAM(5)-XLAM(4))
CON55=(XLAM(1)-XLAM(5))*(XLAM(2)-XLAM(5))*(XLAM(3)-XLAM(5))*
& (XLAM(4)-XLAM(5))
CON511=XLAM(5)-XLAM(4)
CON521=(XLAM(4)-XLAM(3))*(XLAM(5)-XLAM(3))
CON522=(XLAM(3)-XLAM(4))*(XLAM(5)-XLAM(4))
CON523=(XLAM(3)-XLAM(5))*(XLAM(4)-XLAM(5))
CON531=(XLAM(3)-XLAM(2))*(XLAM(4)-XLAM(2))*(XLAM(5)-XLAM(2))
CON532=(XLAM(2)-XLAM(3))*(XLAM(4)-XLAM(3))*(XLAM(5)-XLAM(3))
CON533=(XLAM(2)-XLAM(4))*(XLAM(3)-XLAM(4))*(XLAM(5)-XLAM(4))
CON534=(XLAM(2)-XLAM(5))*(XLAM(3)-XLAM(5))*(XLAM(4)-XLAM(5))
AMCOEF(5,1)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON10*CON20*
* CON30*CON40*CON50)
AMCOEF(5,2)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON10*CON51)
AMCOEF(5,3)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON20*CON52)
AMCOEF(5,4)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON30*CON53)
AMCOEF(5,5)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON40*CON54)
AMCOEF(5,6)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)/(CON50*CON55)
AMCOEF(5,7)=0.OD0
BMCOEF(5,1)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*ATMR(5)*C0(5)/
& CON51
BMCOEF(5,2)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*ATMR(5)*C0(5)/
& CON52 + XLAM(2)*XLAM(3)*XLAM(4)*(ATM(5)/ATM(2))*C0(2)
& /CON531
BMCOEF(5,3)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*ATMR(5)*C0(5)/
& CON53 + XLAM(2)*XLAM(3)*XLAM(4)*(ATM(5)/ATM(2))*C0(2)
& /CON532 + XLAM(3)*XLAM(4)*(ATM(5)/ATM(3))*C0(3)/CON521
BMCOEF(5,4)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*ATMR(5)*C0(5)/
& CON54 + XLAM(2)*XLAM(3)*XLAM(4)*(ATM(5)/ATM(2))*C0(2)
& /CON533 + XLAM(3)*XLAM(4)*(ATM(5)/ATM(3))*C0(3)/CON522
& +XLAM(4)*(ATM(5)/ATM(4))*C0(4)/CON511
BMCOEF(5,5)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*ATMR(5)*C0(5)/

```

```

&      CON55 + XLAM(2)*XLAM(3)*XLAM(4)*(ATM(5)/ATM(2))*C0(2)
&      /CON534 + XLAM(3)*XLAM(4)*(ATM(5)/ATM(3))*C0(3)/CON523
&      -XLAM(4)*(ATM(5)/ATM(4))*C0(4)/CON511 + C0(5)
BMCOEF(5,6)=0.0D0
BMCOEF(5,7)=0.0D0

C
CON60=XLAM(6)+RFLOW/VOL
CON61=CON51*(XLAM(6)-XLAM(1))
CON62=CON52*(XLAM(6)-XLAM(2))
CON63=CON53*(XLAM(6)-XLAM(3))
CON64=CON54*(XLAM(6)-XLAM(4))
CON65=CON55*(XLAM(6)-XLAM(5))
CON66=(XLAM(1)-XLAM(6))*(XLAM(2)-XLAM(6))*(XLAM(3)-XLAM(6))*
&      (XLAM(4)-XLAM(6))*(XLAM(5)-XLAM(6))
CON611=XLAM(6)-XLAM(1)
CON621=(XLAM(5)-XLAM(4))*(XLAM(6)-XLAM(4))
CON622=(XLAM(4)-XLAM(5))*(XLAM(6)-XLAM(5))
CON623=(XLAM(4)-XLAM(6))*(XLAM(5)-XLAM(6))
CON631=(XLAM(4)-XLAM(3))*(XLAM(5)-XLAM(3))*(XLAM(6)-XLAM(3))
CON632=(XLAM(3)-XLAM(4))*(XLAM(5)-XLAM(4))*(XLAM(6)-XLAM(4))
CON633=(XLAM(3)-XLAM(4))*(XLAM(4)-XLAM(5))*(XLAM(6)-XLAM(5))
CON634=(XLAM(3)-XLAM(6))*(XLAM(4)-XLAM(6))*(XLAM(5)-XLAM(6))
CON641=(XLAM(3)-XLAM(2))*(XLAM(4)-XLAM(2))*(XLAM(5)-XLAM(2))
&      *(XLAM(6)-XLAM(2))
CON642=(XLAM(2)-XLAM(3))*(XLAM(4)-XLAM(3))*(XLAM(5)-XLAM(3))
&      *(XLAM(6)-XLAM(3))
CON643=(XLAM(2)-XLAM(4))*(XLAM(3)-XLAM(4))*(XLAM(5)-XLAM(4))
&      *(XLAM(6)-XLAM(4))
CON644=(XLAM(2)-XLAM(5))*(XLAM(3)-XLAM(5))*(XLAM(4)-XLAM(5))
&      *(XLAM(6)-XLAM(5))
CON645=(XLAM(2)-XLAM(6))*(XLAM(3)-XLAM(6))*(XLAM(4)-XLAM(6))
&      *(XLAM(5)-XLAM(6))
AMCOEF(6,1)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON10
&      *CON20*CON30*CON40*CON50*CON60)
AMCOEF(6,2)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON10
&      *CON61)
AMCOEF(6,3)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON20
&      *CON62)
AMCOEF(6,4)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON30
&      *CON63)
AMCOEF(6,5)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON40
&      *CON64)
AMCOEF(6,6)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON50
&      *CON65)
AMCOEF(6,7)=-XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)/(CON60
&      *CON66)
BMCOEF(6,1)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON61
BMCOEF(6,2)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON62 + XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(2))*C0(2)/CON641
BMCOEF(6,3)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON63 + XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(2))*C0(2)/CON642 + XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(3))*C0(3)/CON631
BMCOEF(6,4)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON64 + XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(2))*C0(2)/CON643 + XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(3))*C0(3)/CON632 + XLAM(4)*XLAM(5)*

```

```

&      (ATM(6)/ATM(4))*C0(4)/CON621
BMCOEF(6,5)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON65 + XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(2))*C0(2)/CON644 + XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(3))*C0(3)/CON633 + XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(4))*C0(4)/CON622 + XLAM(5)*
&      (ATM(6)/ATM(5))*C0(5)/CON611
BMCOEF(6,6)=XLAM(1)*XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      ATMR(6)*C0(1)/CON66 + XLAM(2)*XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(2))*C0(2)/CON645 + XLAM(3)*XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(3))*C0(3)/CON634 + XLAM(4)*XLAM(5)*
&      (ATM(6)/ATM(4))*C0(4)/CON623 - XLAM(5)*
&      (ATM(6)/ATM(5))*C0(5)/CON611 + C0(6)
BMCOEF(6,7)=0.0D0

```

```

C
C
C      WRITE OUT THE COMPUTED AM'S

```

```

      WRITE(*,*)' CHAIN COEFFICIENTS
      WRITE(30,*)' CHAIN COEFFICIENTS
      WRITE(*,*)
      WRITE(30,*)
      WRITE(30,*)' AMCOEF S
      WRITE(30,*)
      DO 520 I=1,6
      WRITE(30,30)I, (AMCOEF(I,J),J=1,7)
520  CONTINUE
      WRITE(30,*)
      WRITE(30,*)' BMCOFF S
      WRITE(30,*)
      DO 521 I=1,6
      WRITE(30,30)I, (BMCOEF(I,J),J=1,7)
521  CONTINUE
      30  FORMAT(1X,I5,1X,10(LX,E12.6))

```

```

C
C      BEGIN THE MASTER EVALUATION LOOP
C

```

```

      WRITE(30,*)
      WRITE(*,*)' MASTER EVALUATION LOOP
      WRITE(30,*)'          MASTER EVALUATION LOOP
      WRITE(30,*)' TIME / RADIONUCLIDE CONC.S IN PCI/CM^3
      WRITE(30,*)
      DO 530 J=1,NEVAL

```

```

C
      TIME=DT*FLOAT(J-1)
C

```

```

      E1=DEXP(-CON10*TIME)
      E2=DEXP(-CON20*TIME)
      E3=DEXP(-CON30*TIME)
      E4=DEXP(-CON40*TIME)
      E5=DEXP(-CON50*TIME)
      E6=DEXP(-CON60*TIME)

```

```

C
      CMASS(1)=QFLUX*AREA*ATMR(1)*(AMCOEF(1,1)+AMCOEF(1,2)*E1)
      CMASS(1)=CMASS(1)/VOL
      CMASS(1)=CMASS(1)+BMCOEF(1,1)*E1
      CMASS(1)=SA(1)*CMASS(1)

```

```

C
      CMASS(2)=QFLUX*AREA*ATMR(2)*(AMCOEF(2,1)+AMCOEF(2,2)*E1+
&      AMCOEF(2,3)*E2)

```

```

      CMASS(2)=CMASS(2)/VOL
      CMASS(2)=CMASS(2)+BMCOEF(2,1)*E1+BMCOEF(2,2)*E2
      CMASS(2)=SA(2)*CMASS(2)
C
      CMASS(3)=QFLUX*AREA*ATMR(3)*(AMCOEF(3,1)+AMCOEF(3,2)*E1+
&      AMCOEF(3,3)*E2+AMCOEF(3,4)*E3)
      CMASS(3)=CMASS(3)/VOL
      CMASS(3)=CMASS(3)+BMCOEF(3,1)*E1+BMCOEF(3,2)*E2+BMCOEF(3,3)*E3
      CMASS(3)=SA(3)*CMASS(3)
C
      CMASS(4)=QFLUX*AREA*ATMR(4)*(AMCOEF(4,1)+AMCOEF(4,2)*E1+
&      AMCOEF(4,3)*E2+AMCOEF(4,4)*E3+AMCOEF(4,5)*E4)
      CMASS(4)=CMASS(4)/VOL
      CMASS(4)=CMASS(4)+BMCOEF(4,1)*E1+BMCOEF(4,2)*E2+BMCOEF(4,3)*E3
&      +BMCOEF(4,4)*E4
      CMASS(4)=SA(4)*CMASS(4)
C
      CMASS(5)=QFLUX*AREA*ATMR(5)*(AMCOEF(5,1)+AMCOEF(5,2)*E1+
&      AMCOEF(5,3)*E2+AMCOEF(5,4)*E3+AMCOEF(5,5)*E4+AMCOEF(5,6)*E5)
      CMASS(5)=CMASS(5)/VOL
      CMASS(5)=CMASS(5)+BMCOEF(5,1)*E1+BMCOEF(5,2)*E2+BMCOEF(5,3)*E3
&      +BMCOEF(5,4)*E4+BMCOEF(5,5)*E5
      CMASS(5)=SA(5)*CMASS(5)
C
      CMASS(6)=QFLUX*AREA*ATMR(6)*(AMCOEF(6,1)+AMCOEF(6,2)*E1+
&      AMCOEF(6,3)*E2+AMCOEF(6,4)*E3+AMCOEF(6,5)*E4+AMCOEF(6,6)*E5+
&      AMCOEF(6,7)*E6)
      CMASS(6)=CMASS(6)/VOL
      CMASS(6)=CMASS(6)+BMCOEF(6,1)*E1+BMCOEF(6,2)*E2+BMCOEF(6,3)*E3
&      +BMCOEF(6,4)*E4+BMCOEF(6,5)*E5+BMCOEF(6,6)*E6
      CMASS(6)=SA(6)*CMASS(6)
C
      WRITE(*,*)J,CMASS(1),CMASS(2),CMASS(3),CMASS(4),CMASS(5),
&      CMASS(6)
      WRITE(30,31)TIME,CMASS(1),CMASS(2),CMASS(3),CMASS(4),CMASS(5),
&      CMASS(6)
C
530  CONTINUE
    31  FORMAT(Lx,11(E12.6,Lx))
C
1000  STOP'NORMAL FORTRAN TERMINATION'
      END

```

APPENDIX F

SUMMARY OF FORTRAN LIST-DIRECTED INPUT

A list-directed input record is a sequence of values and value separators. Each value in a list-directed record must be one of the following:

- A constant, optionally multiplied by an unsigned-nonzero-integer constant. For example, 5 or 2*5 (two successive fives) are both acceptable. The integer multiplier is termed the "repeat factor", and implies replication of values as opposed to algebraic multiplication.
- A null value, optionally multiplied by an unsigned-nonzero-integer constant. For example, 5* is five successive null values.

Except in string constants, none of these may have embedded blanks. Each value separator in a list-directed input record must be one of the following:

- A comma (,).
- A forward slash (/).
- A forward slash (/) encountered as a value separator during execution of a list-directed input statement stops execution of that statement after assignment of the previous value. Any further items in the input list are treated as if they were null values. The current program does not make use of this feature.
- One or more contiguous blanks between two constants or after the last constant.

Blanks next to value separators are ignored. For example " 5 6 , 7" is equivalent to "5,6,7".

REFERENCES

- Kaplan, W. 1952. Advanced calculus. Addison-Wesley. Reading, MA. 679pp.
- Lindstrom, F. T., Cawfield, D. E., Emer, D. F., Shott, G. J., and Donahue, M. E.
1992a. CASCADER: An m-chain gas phase radionuclide transport and fate model.
Volume 1 - Basic physics and mathematics. U.S. Department of Energy Report,
USDOE/NV/10630-23. Reynolds Electrical & Engineering Co., Inc., P.O. Box 98521,
Las Vegas, NV.
- Lindstrom, F. T., Cawfield, D. E., Emer, D. F., Shott, G. J., and Donahue, M. E.,
1992b. A simulation of the transport and fate of radon-222 derived from thorium-232
low-level waste in the near surface zone of the radioactive waste management site
in Area 5 of the Nevada Test Site. U.S. Department of Energy Report,
USDOE/NV/10630-38. Reynolds Electrical & Engineering Co., Inc., P.O. Box 98521,
Las Vegas, NV.
- Lindstrom, F. T., Cawfield, D. E., Emer, D. F., Shott, G. J., and Donahue, M. E.,
1992c. CASCADER: An M-chain gas-phase radionuclide transport and fate model.
Volume 3 - Heterogeneous layered porous media. U.S. Department of Energy
Report, USDOE/NV/10630-41. Reynolds Electrical & Engineering Co., Inc., P.O. Box
98521, Las Vegas, NV.
- McLachlan, N. W. 1963. Complex variable theory and transform calculus.
Cambridge press, Cambridge, U.K. 388pp.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1986. Numerical
recipes: The art of scientific computing. Cambridge Univ. press, Cambridge, U.K.
679pp.
- Wylie, C. R. 1960. Advanced engineering mathematics. McGraw-Hill, NY. 696pp.

MATERIAL WORK IN PROCESS

Lindstrom, F. T., Cawfield, D. E., Emer, D. F., Shott, G. J., and Donahue, M. E., 1993. A simulation of the transport and fate of radon-222 derived from thorium-230 low-level waste in the near surface zone of the radioactive waste management site in Area 5 of the Nevada Test Site. U.S. Department of Energy Report, USDOE/NV/10630-58. Reynolds Electrical & Engineering Co., Inc., P.O. Box 98521, Las Vegas, NV. (work submitted for publication)

DISTRIBUTION LIST

Waste Management Department Reference Library, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 501, Las Vegas, NV 89193. (20)

F. T. Lindstrom, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 738, Las Vegas, NV 89193. (5)

D. E. Cawlfeld, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 738, Las Vegas, NV 89193. (5)

D. F. Emer, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 738, Las Vegas, NV 89193. (5)

G. J. Shott, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 738, Las Vegas, NV 89193. (5)

C. J. Sheldon, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 551, Las Vegas, NV 89193. (2)

DOE Office of Scientific and Technical Information, U.S. Department of Energy, P.O. Box 98518, M/S 505, Las Vegas, NV 89193. (2)

DOE/NV Technical Information Resource Center, U.S. Department of Energy, P.O. Box 98518, M/S 505, Las Vegas, NV 89193. (2)

DATE

FILMED

4 / 15 / 94

END