DOE/NV/10630-57
UC-721

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

VOLUME 2 - USER'S MANUAL FOR CASCADR8

BY

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

**SPECIAL PROJECTS SECTION
ENVIRONMENTAL RESTORATION AND
TECHNOLOGY DEVELOPMENT DEPARTMENT**

JUNE 1993

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

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

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

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Yao

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 diffusion. Furthermore, parent and daughter radionuclides may decay as they are transported in the soil.

This is volume two to the CASCADER series, titled CASCADR8. It embodies the concepts presented in volume one of this series. To properly understand how the CASCADR8 model works, the reader should read volume one first. This volume presents the input and output file structure for CASCADR8, and a set of realistic scenarios for buried sources of radon gas.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
ABSTRACT	iii
FIGURES	v
TABLES	vi
ACKNOWLEDGEMENT	vii
1.0 INTRODUCTION	1
2.0 OVERVIEW OF CASCADR8	2
3.0 INPUT FILE STRUCTURE	5
4.0 OUTPUT FILE STRUCTURE	15
5.0 EXAMPLE SCENARIOS	18
5.1 Study 1 - Burial Depth Effect	19
5.2 Study 2 - Various Fourier Components	19
5.3 Study 3 - Changing Gas Permeability	26
APPENDIX A INSTALLING THE PROGRAM	29
APPENDIX B NOMENCLATURE AND FORTRAN SYMBOL LIST	31
APPENDIX C FORTRAN SOURCE CODE LISTING	35
APPENDIX D SUMMARY OF FORTRAN LIST DIRECTED INPUT	67
APPENDIX E ERROR MESSAGES AND THEIR CAUSES	68
APPENDIX F VERIFICATION OF CASCADR8 USING RADON GAS AS A MODEL NOBLE GAS	72
REFERENCES	99
DISTRIBUTION LIST	100

FIGURES

<u>Number</u>	<u>Page</u>
1 Data File required by the CASCADR8 program	6
5.1.1 Burial depth study. Surface flux of radon-222 over time at five different burial depths.	20
5.2.1 Surface flux of radon-222 for very large amplitude 10-day sinusoidal pressure wave.	22
5.2.2 Surface flux of radon-222 for very large amplitude 1-day sinusoidal pressure wave.	23
5.2.3 Surface flux of radon-222 for very large amplitude 1/12 day sinusoidal pressure wave.	24
5.2.4 Radon-222 surface flux (pCi/cm ² -sec). Realistic barometric pressure wave. Source region between 8 and 9 meters below grade.	25
5.3.1 Radon-222 surface flux pCi/cm ² -sec, shallow burial, changing gas permeability.	28
F.1 The domain of definition of the transport and fate problem.	76
F.2 Ingrowth of Ra-226 from 99 grams of Th-230 at time t=0 years.	89
F.3 Verification run01 base case scenario. Analytical and numerical model comparisons. Forty-day of gas evolution.	93
F.4 Verification runs 02-06. One meter thick source region. Source rate same as base case scenario. x_T and x_B for each run listed in Table F.4.	94
F.5 Semi-log plot of mean surface flux of radon-222 vs depth of burial. Analytical case at 40-days of gas evolution.	97
F.6 Semi-log of mean surface flux of radon-222 vs depth of burial. Numerical case at 40 days of gas evolution.	98

TABLES

<u>Number</u>	<u>Page</u>
5.2.1 Amplitudes and periods used in tailoring the "Realistic" barometric pressure wave.	26
F.1 Thorium-230 waste stream summary.	88
F.2 Thorium-230 chain data (Brown and Firestone, 1986).	88
F.3 Run parameters for base case scenario.	91
F.4 Alternative depth of burial scenarios.	95
F.5 Summary of 1n (F[40]) vs mean depth of burial at 40 days gas evolution. . .	95
F.6 Summary of 1n (-F[40]) vs mean depth of burial at 40 days gas evolution. .	96

ACKNOWLEDGEMENT

The authors wish to thank Michael V. Ekedahl, Senior Systems Analyst at the University of Nevada Las Vegas (UNLV) National Super Computing Center for Energy and the Environment, for his assistance both at UNLV and at our site. 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 report (Volume 1) describing the mathematical model CASCADER, (Lindstrom et al. 1992a). As the title implies, this volume is the user's guide to running the software program (CASCADR8) which is based upon the model. Those readers wishing to understand the basic science upon which the model is built should read Volume 1 (Lindstrom et al. 1992a). This document will satisfy all demands from those readers only interested in learning how to set up data files and run the computer program. This document begins with a brief overview of how CASCADR8 runs, including certain conventions that are used in the naming of various input and output files.

2.0 OVERVIEW OF CASCADR8

When initiating the CASCADR8 program, it 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. If running this program on a Unix based system then shell scripts can easily be set up, generating several runs in succession. Batch files serve the same purpose on a personal computer (PC).

The naming of an output file 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: CASCADR8 will overwrite any pre-existing output files which have the same name. This means that each new input file should have a unique name if output files from previous runs are saved.

To begin CASCADR8, 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:

```
cascadr8 run01
```

CASCADR8 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 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, which will also result in long 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 seem to be happening almost continually are they a problem. The value of the "magic" δt 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 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 that will be 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 reports is determined by the NOUT variable in the input file. 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. Appendix A explains program

installation. Appendix B provides a cross reference listing of the mathematical symbols used in the model (see volume one) and the FORTRAN variable names used in the program. Appendix C outlines the entire FORTRAN source code listing. A summary of FORTRAN free format input (the format needed in the input data file) can be found in Appendix D. A discussion of program error messages and their causes are in Appendix E.

3.0 INPUT FILE STRUCTURE

This chapter describes in detail the data file required by the CASCADR8 program. The interface between the program and the input file is designed to provide flexibility in the format for data specification, also 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 will consider a simple data file (Figure 1) 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.,

```
%cascadr8 test
```

The data file starts with a comment section that 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.

```
THIS IS AN EXAMPLE DATA SET FOR CASCADR8.  
IT WAS MODELED AFTER DATASET C8507921.DAT,  
WHICH WAS SCENARIO 5.1 IN THE TECHNICAL MANUAL.  
THE COMMENTS IN THIS SECTION MUST END WITH A  
DOLLAR SIGN ("$$") IN THE FIRST COLUMN OF THE  
LAST LINE. THIS APPEARS ON THE NEXT LINE --  
$
```

This opening comment section can be arbitrarily long; the end is signified by a dollar sign "\$," in the first position of the 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 sections, each of which contains a small

THIS IS AN EXAMPLE DATA SET FOR CASCADR8. IT WAS MODELED AFTER DATASET C8507921.DAT, WHICH WAS SCENARIO 5.1 IN THE TECHNICAL MANUAL. THE COMMENTS IN THIS SECTION MUST END WITH A DOLLAR SIGN ("\$\$") IN THE FIRST COLUMN OF THE LAST LINE. THIS APPEARS ON THE NEXT LINE -- \$

```

SET OF FLAGS: NFLAG(I)
0,0,0,0,0,0,0,0,0,0,0
NODES   NUCS   NOUT
401     4     600
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 1.5D1 1.5D1 2.0D1
1*2.0D1
2.0D1
1.5D1 1.0D1 7.5D0 7.5D0 5.0D0 2.5D0 2.5D0 5.0D0
1.0D1 1.5D1 2.0D1 2.0D1 1.5D1 1.0D1 5.0D0 5.0D0
1.0D1 1.5D1 2.0D1
150*2.0D1
2.0D1 3.0D1 4.0D1 5.0D1 6.0D1 7.0D1 8.0D1 9.0D1 1.0D2
206*1.0D2
NDEP; then 'N' for nodes or 'Z' for actual depths
08 'Z'
PRINT AT THESE NODAL POSITIONS
0.0D0, 2.0D2, 4.0D2, 8.0D2, 1.6D3, 3.2D3, 6.4D3, 2.44D4
INITIAL TIME: T0
0.0D0
FINAL TIME: TIME
3.456D+3
NUMBER OF PRINT TIMES: NPRT
240
DT
60.0D0
PARTICLE DENSITY, PARTDN
2.6
EPSIN, EPS, EPSOT
0.45, 0.45, 1.0
DSO1, DSO2, DSO3, DSOM
0.10, 0.10, 0.10, 0.10
DSP1, DSP2, DSP3, DSPM
0.10, 0.10, 0.10, 0.10
DSOT1, DSOT2, DSOT3, DSOTM
1.00, 1.00, 1.00, 1.00
FREV1, FREV2, FREV3, FREVM
0.5, 0.5, 0.5, 0.5
FIRR1, FIRR2, FIRR3, FIRRm
0.5, 0.5, 0.5, 0.5
KEQ1, KEQ2, KEQ3, KEQm
0.0, 0.0, 0.0, 0.0
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
LTIRR1, LTIRR2, LTIRR3, LTIRRm
0.0, 1.0, 1.0, 1.0
AREA
1.0D+4
COT1, COT2, COT3, COTm
0.0, 0.0, 0.0, 0.0

```

Figure 1. Data File required by the CASCADR8 program.

```

CIN1, CIN2, CIN3, CINm
0.0D0, 0.0D0, 0.0D0, 0.0D0
DXIN
100.0D0
DXOUT
100.0D0
KAPPA, TAIR, RG, MASSA, GRAV, RHOWAT
4.00D-8, 2.93D+2, 8.3143D+7, 29.0D0, 9.8D+2, 1.0D0
PA AND NPRES
68.25D0, 2
POC'S FROM 1 TO 15
0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
POS'S FROM 1 TO 15
-1.50D0, -0.50D0, 0.0D0, 0.0D0, 0.0D0
0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
PERIODS OF EACH COMPONENT IN SECS.
3.456D+5, 8.64D+4, 1.0D0, 1.0D0, 1.0D0
1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0
1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0
pressure (t) == -1.5*sin(2*Pi*t/3.456D+5) - 0.5*sin(2*Pi*t/8.64D+4)
Number of Source Regions; 'N' for nodes, 'Z' for depths.
01 'Z'
INITIAL DISTRIBUTION DATA
00 0.0D0 2.44D4 1.299D-18, 0.0D0
01 2.0D2 3.00D2 1.299D-18, 2.5919D-16

```

Figure 1. Data File required by the CASCADR8 program (continued).

number of data values. Each section 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 contents of the line. 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 throughout the data file, lists of numbers are required; the format required is FORTRAN list-directed input. Separators for data values may be single commas, white space (spaces, tabs and new lines), or both; no other separator, such as a semicolon or slash, will be accepted. The program uses double precision for all floating point numbers, so we have used exponential and double precision notation; 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 of x, n times. Appendix D contains a summary of FORTRAN list-directed ("free form") input.

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

```
SET OF FLAGS: NFLAG(I)  
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 that 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. Write various control parameters to the report file. The parameters to write can be found in SUBROUTINE REPT1, in file C8REPT1.FOR. This will be done once at the beginning of a run, if requested. This option is mainly a development tool.
- 3 - List Initial Concentrations to Report. Which are: at each knot, the velocity (VELOC), the concentration of each radionuclide (COLD), and the reversibly bound concentrations of radionuclides (STIRO). This option is also mainly for program development and verification.
- 4 - List Initial Masses to Report. These are the free, reversibly sorbed, and irreversibly bound masses.
- 5 - List Initial Velocity & Concentration to Screen.
- 6 - Debugging option which lists an intermediate variable (CSTR) to the screen.

7-10 - Reserved.

Flags 7 - 10 are not used for CASCADR8. After the NFLAGS the data file has the following:

NODES	NUCS	NOUT
401	4	600

NODES are 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 sections 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 1.5D1 1.5D1 2.0D1
1*2.0D1
2.0D1
1.5D1 1.0D1 7.5D0 7.5D0 5.0D0 2.5D0 2.5D0 5.0D0
1.0D1 1.5D1 2.0D1 2.0D1 1.5D1 1.0D1 5.0D0 5.0D0
1.0D1 1.5D1 2.0D1
150*2.0D1
2.0D1 3.0D1 4.0D1 5.0D1 6.0D1 7.0D1 8.0D1 9.0D1 1.0D2
206*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 the number of nodes

the program will expect to read. Note the shorthand "150*2.0D1," which means "input the value 2.0D1, 150 times." If specifying "absolute" spacing ("A"), the absolute nodal positions are listed in ascending order and there is no ZOFF value given.

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

```
NDEP; then 'N' for nodes or 'Z' for actual depths
08 'Z'
PRINT AT THESE NODAL POSITIONS
0.0D0, 2.0D2, 4.0D2, 8.0D2, 1.6D3, 3.2D3, 6.4D3, 2.44D4
```

The first number, number of depths (NDEP), determines how many depths will be sampled. The 'Z' means 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+3
NUMBER OF PRINT TIMES: NPRT
240
DT
60.0D0
```

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. δt , in seconds, is the maximum, and default, step size that the simulation will take. In times when conditions are changing rapidly, δt may be reduced to preserve the accuracy of the results, but the step size will never go above the δt specified in the input file.

PARTICLE DENSITY, PARTDN

2.6

EPSIN, EPS, EPSOT

0.45, 0.45, 1.0

PARTDN is the porous medium particle density, in units of (gm/cm³). EPSIN is the porosity of the water table region, EPS is the porosity of the soil region, and EPSOT is the porosity of the boundary layer region. The porosity units are (cm³/cm³).

This version of CASCADER assumes that the soil region is homogeneous from the water table to the surface; hence there is only one EPS. Later versions allow for multiple soil regions.

DSO1, DSO2, DSO3, DSOm
0.10, 0.10, 0.10, 0.10
DSP1, DSP2, DSP3, DSPm
0.10, 0.10, 0.10, 0.10
DSOT1, DSOT2, DSOT3, DSOTm
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. DSOj is the average dispersion coefficient of the jth radionuclide in the water table region; DSPj is the jth average dispersion coefficient in the soil region; and DSOTj is the jth dispersion coefficient in the boundary layer region. The units are all (cm²/sec).

FREV1, FREV2, FREV3, FREVm
0.5, 0.5, 0.5, 0.5
FIRR1, FIRR2, FIRR3, FIRRm
0.5, 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. These quantities are dimensionless.

KEQ1, KEQ2, KEQ3, KEQm
0.0, 0.0, 0.0, 0.0

KEQj is the linear equilibrium reversible sorption coefficient for radionuclide j onto the solid particles. Units are (cm^3 air/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

```

THALFj is the half life for a stationary source of the jth radionuclide; its units are (sec).

SAj is the specific activity of the jth radionuclide, in (pico curies/gm); and AMj is the atomic mass of the jth radionuclide in (gm).

```

LTIRR1, LTIRR2, LTIRR3, LTIRRm
0.0, 1.0, 1.0, 1.0

```

LTIRRj is the rate coefficient for irreversible binding of jth radionuclide, in (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, COTm
0.0, 0.0, 0.0, 0.0
CIN1, CIN2, CIN3, CINm
0.0D0, 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/cm^3).

```

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).

KAPPA, TAIR, RG, MASSA, GRAV, RHOWAT
 4.00D-8, 2.93D+2, 8.3143D+7, 29.0D0, 9.8D+2, 1.0D0

KAPPA is the air permeability in (cm²). 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).

PA AND NPRES
 68.25D0, 2
 POC'S FROM 1 TO 15
 0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
 0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
 0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
 POS'S FROM 1 TO 15
 -1.50D0, -0.50D0, 0.0D0, 0.0D0, 0.0D0
 0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
 0.0D0, 0.0D0, 0.0D0, 0.0D0, 0.0D0
 PERIODS OF EACH COMPONENT IN SECS.
 3.456D+5, 8.64D+4, 1.0D0, 1.0D0, 1.0D0
 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0
 1.0D0, 1.0D0, 1.0D0, 1.0D0, 1.0D0

Atmospheric barometric pressure values for the simulation are produced by a Fourier sum of sines and cosines containing as many as 15 components. 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 only two components are used, since the program will try to read exactly 15 components. If fewer than 15 components are used, then the amplitudes of the remaining components should be zero on both the sine and cosine. The units for the pressures are (cm Hg), and the periods are in seconds. In the example, the barometric pressure used is

pressure (t) == -1.5*sin(2*Pi*t/3.456D+5) - 0.5*sin(2*Pi*t/8.64D+4)

where t represents the time. PA refers to the average barometric pressure (cm Hg).

Number of Source Regions; 'N' for nodes, 'Z' for depths.

```
01      'Z'
INITIAL DISTRIBUTION DATA
00      0.0D0 2.44D4  1.299D-18,  0.0D0
01      2.0D2 3.00D2  1.299D-18,  2.5919D-16
```

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, which is always specified, 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, and are optional. The second and third columns are the beginning (top) and ending (bottom) nodes for the region. 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 two to three meters deep. The fourth column contains the initial concentration, in (gm/cm³), 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 values. The final column, which is the last field, is the buried waste source function for the first radionuclide in the region; the units are (gm radionuclide/cm³ sec).

4.0 OUTPUT FILE STRUCTURE

Each output file generated by CASCADR8 is named with the same root as the input file, 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 seconds (sec). Finally, node depths are listed, along with displacement between nodes, node index numbers, and indicators of which nodes are at print depths. The print depth nodes are marked with a greater than symbol (>). The units on all distances are in centimeters (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 ($\text{pCi}/\text{cm}^2\text{-sec}$), and the third has the cumulative flow in (pCi).

The remaining ".o2x" output files are each organized into (NDEP + 1) columns; the first column is time in days, and the remaining columns contain measured quantity at a specific depth, or horizon of that file. The horizons are listed in the output file, in the same order as 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 are listed 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. The .bin file starts with an initial section containing some global information for the program. The remaining file consists of periodic dumps of the state of the simulation during execution. The number and timing of the dumps is determined by variables NPRT (number of times printed) and TEND (stopping time of computer run) that 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:

37 characters:

- the version number
- the input file name
- the date and time of execution (wall clock)

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)

3 double precision values

- the start time of the simulation (in simulation seconds; e.g., 0.0) (T0)
- the end time of the simulation, in secs (TEND)
- the default time step, in secs (DT0)

3x80 characters (the first 3 lines of the comment section of the data file)

- (NODES - 1) double precision values -- the spacing between consecutive nodes, starting at the surface
- NODES double precision values -- the absolute depth of each node
- NCOL double precision values -- the absolute depth of each node to be sampled on the dumps; i.e., the print depths

NCOL integers

- the node index numbers of the print depths

an integer, value zero

a double precision value -- the starting time of the simulation, in days

All distances are in centimeters. There will be NPRT dumps forming the bulk of the .bin file, and each dump will consist of the following information:

an integer
the index number of this dump (1, 2, ..., NPRT)

a double precision number
the current simulation time, in days

NCOL double precision values
the barometric pressure, in mbar, at each print depth

NCOL double precision values
the average gas flow velocity (seepage velocity) at each print depth, in cm/sec.

NUCS double precision values
the surface flux for each radionuclide in (pCi/cm²-sec)

NUCS double precision values
the cumulative mass loss from the surface for each radionuclide, in (pCi)

for each radionuclide, the following information is listed

NCOL double precision values
the concentration in specific activity units (pCi/gm) at each print depth

NCOL double precision values
the irreversibly bound concentration (gm/gm) at each print depth

NCOL double precision values
the mobile phase mass, in gm, at each print depth

NCOL double precision values
the reversibly sorbed mass, in gm, at each print depth

NCOL double precision values
the irreversibly sorbed mass (gm) at each print depth

Which means "for each radionuclide" the entire packet of values is printed for the first radionuclide, then the entire packet for the second radionuclide, etc.

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

5.0 EXAMPLE SCENARIOS

Several illustrations of the use of CASCADR8 were given in Lindstrom et al. (1992a). Here we use the same type of buried radium-226 source and consider in some detail: 1) the effect on the surface flux of radon-222, when the source region burial depth is increased. 2) the effect on the surface flux of radon-222 by different Fourier components in the barometric pressure wave, and 3) the partial effect on the surface flux of radon-222 due to reduced advection caused by the reduction of porous medium gas permeability. The one meter thick source region is now defined and used throughout the remainder of this report.

Suppose that at time $t=0$, 1.866×10^5 gm of radium-226 metal dust has been instantaneously mixed uniformly into a plane sheet type source region one meter thick and 100 meters in length and width respectively. The volume of the source is thus $1 \times 10^{10} \text{ cm}^3$. The source rate function $S_{SP}(x)$ as defined in Lindstrom et al. (1992a) is found via the formula

$$S_{SP}(x) = \frac{1.866 \times 10^5}{1 \times 10^{10} \text{ cm}^3} \frac{0.693}{(1600)(365.25)(86,400)} = 2.5919 \times 10^{-16} \frac{\text{gm}}{\text{cm}^3 - \text{sec}} \quad (5.0.1)$$

which is a positive constant in the region, $x_T \leq x \leq x_B$ and zero elsewhere.

Assume the same background or naturally occurring radium-226 source as in Lindstrom et al. (1992a), the gas-phase radon-222 concentration being 0.2 pCi/cm^3 which upon using the specific activity of radon-222 ($1.5377 \times 10^{17} \text{ pCi/gm}$) becomes the constant background source concentration of $1.299 \times 10^{-18} \text{ gm/cm}^3$. Using formula 2.2.6 in Lindstrom et al. (1992a), one can compute the naturally occurring radium-226 source rate function. Table 4.1.1 in Lindstrom et al. (1992a) details the values of the model parameters used in all the following scenarios.

5.1 Study 1 – Burial Depth Effect

Five simulation runs, based upon moving the source region deeper into the 4 Darcy homogeneous dry alluvial soil, have been made. The thickness of the source region is fixed at one meter. The five depths of burial then are: $x_T = 2.0, 4.0, 8.0, 16,$ and 32 meters, respectively. x_B in each case is found from the simple formula $x_B = x_T + 1.0$. The barometric pressure wave is tailored from superposing two sinusoidal waves and the "d.c." or average pressure. The "d.c." value is 910.0 mb, while the two trigonometric (trig) components are:

period = 3.456×10^5 sec (4-days) with amplitude 20 mb, and

period = $86,400$ sec (1-day) with amplitude 6.66 mb, respectively.

Five CASCADR8 runs were made, each run extending to 40 simulated days.

Figure 5.1.1 summarizes the radon-222 surface flux over time in ($\text{pCi}/\text{cm}^2\text{-sec}$) for each of the five burial depths. The attenuation of surface flux with increasing burial depth is clearly shown. Also, the increasing lag time or time to steady-state with increasing depth of burial is clearly seen. In Appendix F, the exponential decrease in surface radon-222 flux with increasing depth is derived for the case of pure diffusion transport and a fixed zero concentration surface boundary. Neither of these two qualifying conditions strictly hold in the simulations shown in Figure 5.1.1. However, a semi-log plot of the mean daily surface flux (not shown) as a function of the mean burial depth $x_m = \frac{x_T + x_B}{2}$ indicates that diffusion is the major transport pathway. Additional work in setting up engineering design studies has been done in Lindstrom et al. (1992b and 1993).

5.2 Study 2 – Various Fourier Components

Although diffusion seems to be the major method of transporting radon-222 from the source region to the alluvium-atmosphere interface ($x=0$), the induced advection does play a significant role (Lindstrom et al. 1992b, Section 4, scenario 3). Three

Surface Flux vs Time

Burial Depths Scenario

Figure 5.1.1 Burial depth study. Surface flux of radon-222 over time at five different burial depths.

simulations, each using the one meter thick source region, two meters below grade, have been made. All three simulations use the same background concentration and manmade source rate functions as was used in study 1, above. The first simulation used a barometric pressure wave made up of

$$p_{\text{air}}(0,t) = 910 - 66.64 \sin\left(\frac{2\pi t}{8.64 \times 10^5}\right) \quad (\text{mb}) \quad (5.2.1)$$

where the 8.64×10^5 sec period is equivalent to 10 days. The second simulation used a barometric pressure wave made up of

$$p_{\text{air}}(0,t) = 910 - 66.64 \sin\left(\frac{2\pi t}{8.64 \times 10^4}\right) , \quad (\text{mb}) \quad (5.2.2)$$

where the 3.64×10^4 sec period is equivalent to one day. The third simulation used a barometric pressure wave made up of

$$p_{\text{air}}(0,t) = 910 - 66.64 \sin\left(\frac{2\pi t}{7200}\right) \quad (\text{mb}) \quad (5.2.3)$$

where the 7,200 sec period is equivalent to 1/12 day. The pressure amplitude 66.64 mb has been inflated by roughly an order of magnitude over a real barometric pressure amplitude for these Fourier frequencies. Figures 5.2.1 through 5.2.4 summarize the radon-222 surface fluxes for each Fourier frequency at each of five burial depths. In all cases the model data sampling rate is 10 times/day. The figures clearly show that long wavelength components penetrate the deepest into the dry desert alluvium. The high frequency components effect is rapidly attenuated with depth. The model becomes more diffusion controlled as the frequency increases. This is another reason for deep burial to get the waste region out of the domain of influence of atmospheric pumping.

A fourth simulation using the same background concentration and manmade source region between eight and nine meters was made. The barometric pressure wave used was made up of the superposition "d.c." component and nine realistic

Surface Flux vs Time

Long Wave Fourier Components

Burial
Depths

Figure 5.2.1 Surface flux of radon-222 for very large amplitude 10-day sinusoidal pressure wave.

Surface Flux vs Time

Medium Wave Fourier Components

Figure 5.2.2 Surface flux of radon-222 for very large amplitude 1-day sinusoidal pressure wave.

Surface Flux vs Time

Short Wave Fourier Components

Figure 5.2.3 Surface flux of radon-222 for very large amplitude 1/12 day sinusoidal pressure wave.

Surface Flux vs Time

Realistic Air Pumping Scenario

Figure 5.2.4 Radon-222 surface flux (pCi/cm2-sec). Realistic barometric pressure wave. Source region between 8 and 9 meters below grade.

phases and amplitude sinusoid and cosinusoidal waves. Table 5.2.1 lists the amplitudes and periods of the trigonometric waves. These data were obtained from L. Barker (1992), and derived by Fast Fourier transforming a string (two years) of National Oceanic and Atmospheric Administration (NOAA) weather station barometric pressure data.

Table 5.2.1 Amplitudes and periods used in tailoring the "Realistic" barometric pressure wave.

Period (sec)	Amp. Cos. (mb)	Amp. Sin. (mb)
3.1536×10^7	+ 3.396	- 1.742
1.5768×10^7	+ 0.654	+ 0.684
3.942×10^6	+ 0.380	+ 0.279
8.64×10^5	- 0.465	+ 0.221
3.456×10^5	+ 0.421	- 0.089
2.592×10^5	+ 0.163	- 0.001
8.64×10^4	- 0.620	+ 1.094
4.32×10^4	+ 0.608	- 0.259
7.2×10^3	- 1.470	+ 1.470

It appears the surface flux has migrated into a steady-state distribution by day 30. Based upon the study of Figures 5.2.1 through 5.2.4, the low frequency components seem to have the most influence in moving radon-222 toward the surface. Recall that the source region is buried 8.0 meters below grade.

5.3 Study 3 – Changing Gas Permeability

In this study we investigate the partial effect of changing the gas permeability on the surface flux of radon-222 given the buried source is between two and three meters below grade and the background and manmade sources have the same strength as discussed in study 1, above. Also the same two component pressure wave used in study 1, is used here as well. The effect seen is only partial in that a lowering of the gas permeability of the dry alluvial material lowers both the porosity and the gas-phase diffusion coefficient in a non-linear fashion (Nazaroff, 1992). In this simulation only the

gas permeability parameter was changed in decade increments, hence the use of the word "partial."

Figure 5.3.1 shows the time distribution of the surface flux of radon-222 when the gas permeability of the porous medium is changed from $\kappa_{\text{air}} = 40$ Darcy ($4.0 \times 10^{-7} \text{ cm}^2$) to $\kappa_{\text{air}} = 0.04$ Darcy ($4.0 \times 10^{-10} \text{ cm}^2$) in decades without changing the porosity, ϵ , or the radon-222 diffusion coefficient, D^1 . The reduction in surface flux is dramatic, reflecting the fact that changing κ_{air} greatly affects the air pumping subsystem near the surface. Since we are assuming shallow burial here, $x_T = 2.0$ m and $x_B = 3.0$ m, the effect on the surface flux is considerable. For deep burial the effect is less dramatic.

Surface Flux vs Time

Gas Permeability Scenario

Figure 5.3.1 Radon-222 surface flux pCi/cm2-sec, shallow burial, changing gas permeability.

APPENDIX A

INSTALLING THE PROGRAM

CASCADR8 has been successfully compiled and run on the following machines and systems:

- IBM Compatible PC running MS-DOS and MS FORTRAN
- Sun SPARCstation-2 running SunOS 4.1.x and Sun FORTRAN 1.4
- VAX 8550 running VMS and VAX FORTRAN
- Cray Y/MP

Some minor variations may be required when porting between machines within this set. These modifications involve the processing of command line arguments. Subroutine "GET_ROOT," in source file "C8READ1.F" or "C8READ1.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. GET_ARG is not supported on the VAX, so for this machine the statement "NARGC = IARGC()", in GET_ROOT, will be replaced with "NARGC = 0". Then the ROOT of the input file is entered manually at the prompt. The call to GET_ARG should then be comment out, or a dummy GET_ARG subroutine, which does nothing, can be added to the program.

Both executable files and source code will be provided for any of the above machines. If users desire to run CASCADR8 on other machines, the authors may provide some assistance in porting the code over to those machines. The preferred method for

attaining these files is via anonymous file transfer protocol (ftp) site ... An alternative method is a DOS formatted, 3-1/2 inch floppy diskette, available from the authors.

In most cases CASCADR8 can be installed simply by copying the executable file and sample data files from the ftp site or diskette onto your host computer. (The executable file is called "CASCADR8.EXE" if it is the PC version, "CASCADR8" otherwise.) It can then be run, as described in chapter 2, by entering the command "cascadr8 ROOT," where ROOT is the first part of the data file name; e.g., for file "test1.dat," ROOT is "test1."

Sometimes the executable file cannot be used, or is not desired. This would be the case, for example, if the user desires to simulate more radionuclides than the maximum allowed by the program (MNUC), so this maximum must be changed and the program recompiled. To recompile, copy all source files from the /SRC directory of the diskette or ftp site. The source files include names that end with the extensions "f" or "FOR", and others with extension "v08". All files are necessary to compile the program. The "v08" files contain all common block declarations, while the "f" or "FOR" files contain the actual FORTRAN code. The PC also requires the file FUTIL.LIB, which is only used during compilation. Compilation commands are described below.

PC with MS-DOS and MS FORTRAN:

```
f1/G2/Ox *.for
```

SUN SPARCstation:

```
f77 -lv77 -O3 -native -o cascadr8 *.f
```

VAX 8550:

```
For CASCADR8, C8READ1, C8REPTI, C8BRITE, AIRPUMP8, EVAL8, THOMAS8,  
Link CASCADR8, C8READ1, C8REPT1, C8BRITE, AIRPUMP8, EVAL8, THOMAS8
```

CRAY Y/MP:

```
cf77 -Wf"-dp -ev" -o cascadr8 *.f
```

APPENDIX B

NOMENCLATURE AND FORTRAN SYMBOL LIST

Symbol	Definition	Units	Fortran 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(J)
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 $X = X_{Nx}$	(cm/sec)	VELOT(J)
D^j	jth average dispersion coefficient in soil region	(cm ² /sec)	DSP(J)
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

Symbol	Definition	Units	Fortran Variable
D_{out}^j	average dispersion coefficient of jth radionuclide in the boundary layer region	(cm ² /sec)	DSOT(J)
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(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(J)
M_j	atomic mass of jth radionuclide	(curies/gm)	AM(J)
λ_j	decay coefficient for jth radionuclide	(1/sec)	THALF(J) $\lambda_j = \frac{0.693}{THALF(J)}$
λ_{irr}^j	rate coefficient for irreversible binding of jth radionuclide	(1/sec)	LTIRR
ρ_p	porous medium particle density	(gm/cm ³)	PARTDN
f_{revj}	fraction of porous medium solids available for reversible binding of jth radionuclide	(dimensionless)	FREV(J)
f_{irrj}	fraction of porous medium solids available for irreversible binding of jth radionuclide	(dimensionless)	FIRR(J)

Symbol	Definition	Units	Fortran 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	$\left(\frac{\text{gm radionuclide}}{\text{cm}^3\text{-sec}} \right)$	SSP(I)
κ_{air}	air permeability	(cm^2)	KAPPA
K_{air}	pneumatic conductivity	cm/sec	KAIR(I)
η_{air}	air viscosity	$\frac{\text{dyne-sec}}{\text{cm}^2}$	ETAIR
ρ_{air}	air density	gm/cm ³	RHOAIR
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	cm Hg	poc(I) & pos(I)
L_i	ith barometric component period	(sec)	LP(I)
P_{air}	barometric air pressure	cm-Hg	PRESSO(J), PRESSN(J)
P_A	average barometric pressure	dyne/cm ²	PA
M_{air}	molecular mass of air	gm/mole	MASSA
R_g	gas constant	$\frac{\text{erg}}{\text{mole-}^\circ\text{K}}$	RG
T	kelvin temperature	(°K)	TAIR
D_{air}	pneumatic diffusivity	cm ² /sec	DAIR(I)

Symbol	Definition	Units	Fortran Variable
S_l^j	irreversibly bound concentration of radionuclide j in the transport and fate region $[0, L_x]$	(gm/gm)	STIRO(I) & STIRN(I)
N_x	number of equal length subintervals into which $[0, L_x]$ is subdivided	(dimensionless)	NODES
Δt	time increment between adjacent time levels	(sec)	δt
$M_{Tfree\ k, n+1}^j$	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)
34 $M_{TR\ k, n+1}^j$	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)
$M_{Tlk, n+1}^j$	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 jth radionuclide from the surface $x = 0$	gm	CUMFO(J) & CUMFN(J)

APPENDIX C

FORTRAN SOURCE CODE LISTING

```

*COMSIZE
C
* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible *
sorption, advection-dispersion equations - Bateman equations at inlet *
boundary with mass conserving flux boundary conditions.
C
COMMON BLOCKS FOR COMSIZE.V08
C*****
C..
INTEGER          ZSET, NPT,  MNUC,  MCOL
C..
PARAMETER        (NZSET=0401, NPT=600,MNUC=4, MCOL = 10)
C..
INTEGER          NODES, NM1, NM2. NUCS, NPRES, NFLAG,NOUT
COMMON/DATA19/   NODES, NM1, NM2. NUCS, NPRES, NFLAG(10), NOUT
C..
INTEGER          NDEPTS, NCOL
DOUBLE PRECISION ZDEPTS
COMMON/DATA0/    ZDEPTS(MCOL), NDEPTS(MCOL), NCOL
C..
CHARACTER        DAYT*9, TYME*8, FINP*12
COMMON/DATAD1/   DAYT  TYME,  FINP
C..
CHARACTER        VERSION*(*)
PARAMETER        (VERSION='CASCv8.5')
C..

```

```

*COMASI
C
* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible *
sorption, advection-dispersion equations - Bateman equations at inlet *
boundary with mass conserving flux boundary conditions.
C
COMMON BLOCKS FOR COMASI.V08
C*****
C..
DOUBLE PRECISION COT
COMMON/DATA9/    COT(MNUC)
C..
DOUBLE PRECISION DSOT,          EPSOT, VELOT
COMMON/DATA29/   DSOT(MNUC), EPSOT, VELOT
C..
DOUBLE PRECISION QFLXO,          QFLXN,
& CUMFO,          CUMFN
COMMON/DATA31/   QFLXO(MNUC), QFLXN(MNUC),
& CUMFO(MNUC), CUMFN(MNUC)

```

*COMCONS

Last Revision: April 4, 1992

C Source File: COMCONS.V08

* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible *
 * sorption, advection-dispersion equations - Bateman equations at inlet *
 * boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMCONS.V08

C*****

C..

DOUBLE PRECISION CTSIR, CTIR1, CTIR2
 COMMON/DATA42/ CTSIR(MNUC), CTIR1(MNUC), CTIR2(MNUC)

C..

DOUBLE PRECISION PI, OMEG, PIO2, RHOWAT
 COMMON/DATA47/ PI, OMEG(15), PIO2, RHOWAT

C..

DOUBLE PRECISION DAIR, KAIR, KADZ
 COMMON/DATA47/ DAIR(NZSET), KAIR(NZSET), KADZ(NZSET)

C..

DOUBLE PRECISION RATIO
 COMMON/DATAAD8/ RATIO(NZSET,MNUC)

C..

DOUBLE PRECISION CONLM
 COMMON/DATA53/ CONLM(MNUC)

C..

*COMCTRL

Last Revision: April 4, 1992

C Source File: COMCTRL.V08

* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible
 * sorption, advection-dispersion equations - Bateman equations at inlet
 * boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMCTRL.V08

C*****

C..

INTEGER NPRT
 DOUBLE PRECISION T0, DT, DT0, T, TEND, TNEXT, TOUT, TPRT

C..

COMMON/DATA16/ T0, DT, DT0, T, TEND, TNEXT, TOUT, TPRT(NPT),
 & NPRT

C..

*COMGEOM

Last Revision: April 16, 1992

C Source File: COMGEOM.V08

* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible
 * sorption, advection-dispersion equations - Bateman equations at inlet
 * boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMGEOM.V08

C*****

C..

DOUBLE PRECISION ZOFF, ZNOT, DZ
 COMMON/DATAD5/ ZOFF, ZNOT(NZSET), DZ(NZSET-1)

C..

DOUBLE PRECISION ZSTR
 COMMON/DATA17/ ZSTR(NZSET)

C..

DOUBLE PRECISION AREA, DXIN, DXOUT, DEPTH
 COMMON/DATA18/ AREA, DXIN, DXOUT, DEPTH

C..

*COMNAME Last Revision: June 23, 1992
C Source File: COMNAME.V08

*Euler-Lagrange time integrated, 4-chain radionuclide, irreversible *
sorption, advection-dispersion equations - Bateman equations at inlet *
boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMNAME.V08

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.V08

* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible *
sorption, advection-dispersion equations - Bateman equations at inlet *
boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMNUMR.V08

C..
DOUBLE PRECISION CTEMP, DTEMP, XNOW

C..
COMMON/DATA30/ CTEMP(NZSET), DTEMP(NZSET), XNOW(NZSET)

*COMPUMP Last Revision: April 4, 1992
C Source File: COMPUMP.V08

* Air pumping in dry, isothermal soils of Area 5, NTS. Complex admixture of
* up to 15 sinusoids giving variation about a fixed mean in atmospheric
* pressure at the half space boundary (0, $+\infty$).

C COMMON BLOCKS FOR COMPUMP.V08

C*****

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

C..
DOUBLE PRECISION KAPPA, TAIR, MASSA, RG, GRAV
COMMON/DATA45/ KAPPA, TAIR, MASSA, RG, GRAV

C..
DOUBLE PRECISION PA, POC POS, LP
COMMON/DATA46/ PA, 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
COMMON/DATA49/ ACON, BCON

C..

*COMRNX

Last Revision: April 4, 1992

C

Source File: COMRNX.V08

* Euler-Lagrange time integrated, 4-chain radionuclide, irreversible
* sorption, advection-dispersion equations - Bateman equations at inlet
* boundary with mass conserving flux boundary conditions.

C COMMON BLOCKS FOR COMRNX.V08

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(MNUC), DSO(MNUC), EPSIN, VELIN

C..

DOUBLE PRECISION PARTDN, EPS, VEL
COMMON/DATA12/ PARTDN, EPS, 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(MNUC), FREV(MNUC), FIRR(MNUC), KEQ(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(MNUC)

C..

DOUBLE PRECISION XLSPM
COMMON/DATA34/ XLSPM(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..

```

```

*CASCADR8
C

```

Last Revision: August 11, 1992
Source File: CASCADR8.FOR

```

PROGRAM CASCADR8
INCLUDE 'comsize.v08'
INCLUDE 'comgeom.v08'
INCLUDE 'comname.v08'
INCLUDE 'comasi.v08'
INCLUDE 'compump.v08'
INCLUDE 'comrnxp.v08'
INCLUDE 'comctrl.v08'
INCLUDE 'comnumr.v08'
INCLUDE 'comcons.v08'
INTEGER           IPRINT, IREC, NU
INTEGER           I, J, KKK, LU, K
DOUBLE PRECISION  YPO, YPO1
DOUBLE PRECISION  DENOM, CON1, TEST, DUMMYZ
DOUBLE PRECISION  DUMCON, COURANT
INTEGER           IADTP, IADTC, IADTH, IADTV(0:6)
LOGICAL           DONE, COURNTOK, COURFAIL
DOUBLE PRECISION  DZD, FDZ
DOUBLE PRECISION  TEMP, MU_T, ETAIR
DOUBLE PRECISION  DAIR0, KAIR0, PAIR0, RAIR0
DOUBLE PRECISION  CON0, ECON
DATA IADTV/1.0D0,6*0.0D0/
*****
* 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. *
* Up to 15 sinusoids to tailor the barometric pressure function. *
* An accuracy test is implemented by reducing the time step size if the *
* Courant test on velocity is exceeded. *
*****
* This model is the joint effort of: F. T. Lindstrom, D. E. Cawlfild *
* D. Daffern, M. E. Donahue, D. F. Emer, and Gregory J. Shott *
* *
* Special Projects Group *
* Waste Management Division *
* Reynolds Electrical and Engineering Co., Inc. *
* P O box 98521 M/S 501 Las Vegas, NV 89193-8521 *
*****
C..
C..The following is an Arithmetic Statement Function . . .
C..
      FDZ(K) = DZ(K-1) + DZ(K)
C..  CALCULATE CONSTANTS AND SYSTEM GROUP PARAMETERS
C..
      WRITE(*,*)
      WRITE(*,*)'Reading initial data . . .'
      WRITE(*,*)
      CALL READ1
      CALL READ2

```

```

CALL REPT2
CLOSE(10)
WRITE(*,*)
WRITE(*,*)'CALCULATING CONSTANTS AND SYSTEM GROUP PARAMETERS'
WRITE(*,*)

C..
PI      = 4.0D0*DATAN(1.0D0)

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..
PAIRO   = RHOWAT*GRAV*PA*13.6D0
RAIRO   = MASSA*PAIRO / (RG*TAIR)
DAIRO   = KAPPA*PAIRO / (EPS*ETAIR)
KAIRO   = KAPPA*RAIRO*GRAV / (EPS*ETAIR)

C..
C..Compute DAIR and KAIR as a function of Z . . .
C..
CON0 = MASSA*GRAV / (RG*TAIR)
DO I = 1, NODES
    ECON = EXP(CON0*ZNOT(I))
    DAIR(I) = DAIR0 * ECON
    KAIR(I) = KAIRO * ECON
    KADZ(I) = CON0 * KAIR(I)
END DO

C..
PRESSO(1) = PAIRO
PRESSN(1) = PRESSO(1)
DO K = 1, NM1
    PRESSO(K+1) = (1.0D0 + CON0*DZ(K)) * PRESSO(K)
    PRESSN(K+1) = PRESSO(K+1)
END DO
DT      = DT0

C..
AA(1) = 0.0D0
BB(1) = 1.0D0
CC(1) = 0.0D0
DO I = 2, NM1
    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
AA(NODES) = -1.0D0
BB(NODES) = 1.0D0 - CON0*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..

```

```

RET(1) = (1.0D0-EPS)*FREV(1)*PARTDN*KEQ(1) / EPS
CTSIR(1) = (1.0D0 - 0.5D0*DT*LAM(1)) /
& (1.0D0 + 0.5D0*DT*LAM(1))
CTIR1(1) = (0.5D0*DT)*LTIRR(1)
CTIR1(1) = CTIR1(1)/1.0D0 + 0.5D0*DT*LAM(1)
CTIR2(1) = 0.0D0
XLSPM(1) = (1.0D0-EPS)*FIRR(1)*LTIRR(1)/(EPS*(1.0D0+RET(1)))
CONLM(1) = AM(1)/AM(0)

C..
DO NU = 2, NUCS
  RET (NU) = (1.0D0-EPS)*FREV(NU)*PARTDN*KEQ(NU)/EPS
  CTSIR(NU) = (1.0D0 - 0.5D0*DT*LAM(NU)) /
& (1.0D0 + 0.5D0*DT*LAM(NU))
  CTIR1(NU) = (0.5D0*DT)*LTIRR(NU)
  CTIR1(NU) = CTIR1(NU) / (1.0D0 + 0.5D0*DT*LAM(NU))
  CTIR2(NU) = (0.5D0*LAM(NU-1)*DT) /
& (1.0D0 + 0.5D0*DT*LAM(NU))
  XLSPM(NU) = (1.0D0-EPS)*FIRR(NU)*LTIRR(NU) /
& (EPS*(1.0D0+RET(NU)))
  CONLM(NU) = LAM(NU-1)*(1.0D0+RET(NU-1))/(1.0D0+RET(NU))
  CONLM(NU) = (AM(NU)/AM(NU-1)) * CONLM(NU)
END DO

C..
DO NU = 1, NUCS
  DO J = 1, NM1
    RATIO(J,NU) = 2.0D0*DSP(NU) / (DZ(J)*(1.0D0+RET(NU)))
  END DO
END DO

C..
KKK = 1
DEPTH = ZKNOT(NODES)

C..
IF (NFLAD(2) .EQ. 1) THEN
  WRITE(*,*)
  WRITE(*,*)
  WRITE(*,*) ' Writing out system control constants. '
  WRITE(*,*)
  WRITE(*,*)
  CALL REPT1
END IF

C..
C..Initialization of all concentration, flux, and mass variables.
C..
DO I = 1, NODES
  SOURCE(I) = EPS*(1.0D0+RET(1))*(LAM(1)+XLSPM(1))*COLD(I,1) /
& (PARTDN*(1.0D0-EPS))
  STIRO(I,1) = LTIRR(1)*COLD(I,1)/(PARTDN*LAM(1))
  DO NU = 2, NUCS
    COLD(I,NU) = CONLM(NU)*COLD(I,NU-1)/(LAM(NU)+XLSPM(NU))
    STIRO(I,NU) = LTIRR(NU)*COLD(I,NU)/(PARTDN*LAM(NU))
    STIRO(I,NU) = STIRO(I,NU) + LAM(NU-1)*STIRO(I,NU-1)/LAM(NU)
  END DO
END DO

C..
CALL VELCTY(COURANT, COURNTOK, COURFAIL)
VELOC(NODES) = 0.0D0

C..
VELOT = (EPS*VELOC(1))/EPSOT

C..

```

```

DO NU = 1, NUCS
  IF (VELOC(1) .LE. 0.0D0) THEN
    QFLXO(NU) = - EPS*DSP(NU) * (COLD(2,NU)-COLD(1,NU)) /DZ(1)
    &          + EPS*VELOC(1)*COLD(1,NU)
  ELSE
    QFLXO(NU) = - EPSOT*DSOT(NU) * (COLD(1,NU)-COT(NU)) /DXOUT +
    &          VELOT*COT(NU)
  ENDIF
C
  QFLXO(NU) = QFLXO(NU)*SA(NU)
  CUMFO(NU) = 0.0D0
C
  DO J = 2, NODES
    DZD = DZ(J-1)
    MTF(J-1,NU) = EPS*DZD*AREA*
    &             0.5D0 (COLD(J,NU)+COLD(J-1,NU))
    MTS(J-1,NU) = (1.0D0-EPS)*DZD*AREA*FREV(NU)*PARTDN*KEG(NU)*
    &             0.5D0*(COLD(J,NU)+COLD(J-1,NU))
    MTIRR(J-1,NU) = (1.0D0-EPS)*DZD*AREA*FIRR(NU)*PARTDN*
    &             0.5D0*(STIRO(J,NU) = STIRO(J-1,NU))
  END DO
END DO
C
  T = T0
  TOUT = T / 8.64D+4
C..
C..Compute the LU factorization for the Thomas algorithm
C..as applied to the Lagrange cubic spline-advection approximation
C..
  ASOS(1) = 0.0D0
  BSOS(1) = 2.0D0*DZ(1)
  CSOS(1) = DZ(1)
C
  DO J = 2, NM1
    ASOS(J) = DZ(J-1)
    BSOS(J) = 2.0D0*FDZ(J)
    CSOS(J) = DZ(J)
  END DO
C
  ASOS(NODES) = DZ(NM1)
  BSOS(NODES) = 2.0D0*DZ(NM1)
  CSOS(NODES) = 0.0D0
C
  CSS(1) = CSOS(1)/BSOS(1)
  DO J = 2, NODES
    DENOM = BSOS(J)-ASOS(J)*CSS(J-1)
    CSS(J) = CSOS(J)/DENOM
  END DO
C..
  IF (NFLAG(5) .EQ. 1) THEN
    DO I = 1, NODES
      WRITE(*,30) ZKNOT(I),
      &          (SA(NU)*COLD(I,NU), NU = 1, NUCS)
    END DO
  END IF
C..
  IF (NFLAG(3) .EQ. 1) THEN
    WRITE(20,*)
    WRITE(20,4) ' The initial velocity and concentration output '

```



```

        WRITE(20,4) ' has the following tabular form: '
        WRITE(20,*)
        WRITE(20,2024) 'Node,' 'Zdepth,' 'Vel(z),'
&        ('COLD(z,' NU, '),', NU = 1, NUCS),
&        ('STIRO(z,' NU, '),', NU = 1, NUCS)
2024 FORMAT(1X, A6, A10, A12, 50(A9, I1, A2))
        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,'z NU, '),',
&        'Mts(z,' NU, '),',
&        'Mtirr(z,' NU, '),', NU =1, NCUS)
2026 FORMAT(1X, A10, 50(A9, I1, A2))
        WRITE(20,*)
        DO I = 2, NODES
            DUMMYZ = (KNOT(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 here if initial data preview is requested
C..
    IF (NFLAG(1) .EQ. 1) STOP 'Initial Data Preview (only).'
```

C..

```

    IREC = 0
    CALL BRITE_INIT(IREC)
    WRITE(*,*)
    WRITE(*,*)
    WRITE(*, 27) 'Time = ', T
    WRITE(20,*)
Cold WRITE(20,27) 'Time = ', T
    IPRINT = 1
    DONE = .FALSE.
C..
C..*** Begin the master computation loop ***.
C..
C..There is a time-step reduction scheme buried in the main
C..loop. If the Courant test fails (in routine VELCTY), then
C..COURNTOK returns as false and also COURFAIL returns true.
C..At this point, the time step size, DT, is reduced by a factor
C..of two. The calls to AIRPUMP and VELCTY are repeated until
C..COURNTOK becomes true. The count of the power of two reduction
C..is placed on a "stack" -- held in array IADTV using a pointer
C..IADTP. This count is also placed in local counter IADTH, which
C..is then decremented at the end of the master loop. When the
C..counter IADTH becomes zero, then the stack is checked to see
```

C..if a previous reduction needs satisfying. The stack is empty
 C..when IADTP is zero. The placing of the reductions onto a stack
 C..allows the time step reduction algorithm to catch up to an
 C..exact multiple of the initial time step, and also allows
 C..for reductions while a previous reduction is being played out.
 C..There is an arbitrary limit of 6 entries on the reduction stack.

C..
 3000 CONTINUE

C..
 C..Compute the local pressure field, and
 C..seepage velocities at each nodal point.
 C..

```

      IF (IADTP .NE. 0 .AND. IADTH .LE. 0) THEN
        DT = DT*DBLE(IADTV(IADTP))
        WRITE(*,2760) 'Restored', IADTP, IADTV(IADTP), DT
        IADTV(IADTP) = 0
        IADTP = IADTP - 1
        IADTH = IADTV(IADTP)
      END IF

```

C

```

      COURFAIL = .FALSE.
      COURNTOK = .FALSE.
      IADTC = 1
      DO WHILE (.NOT. COURNTOK)
        CALL AIRPUMP
        CALL VELCTY(COURANT, COURNTOK, COURFAIL)
        IF (.NOT. COURNTOK) THEN
          IADTC = 2*IADTC
          DT = 0.5D0*DT
          DO I = 2, NM1
            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
        END IF
      END DO

```

C

```

      IF (COURFAIL) THEN
        IADTP = IADTP + 1
        IF (IADTP .LE. 6) THEN
          IADTV(IADTP) = IADTC
          IADTH = IADTC
          WRITE(*,2760) 'Reduced', IADTP, IADTC, DT
        ELSE
          PRINT *, '*** IADTP > 6 ***'
          STOP 2760
        END IF
      END IF

```

2760 FORMAT(1X, A, ' at level ', I1, ' by ', I4, ' to', 1P, G13.6)

C

```

      TNEXT = T + DT
      VELOC(NODES) = 0.0D0
      VEL(NODES) = 0.0D0
      VELOT = (EPS*VELOC(1))/EPSOT
      VELIN = 0.0D0

```

C

```

      DO NU = 1, NUCS

```

```

DO J = 1, NM1
  VEL(J) = VELOC(J) / (1.0D0+RET(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-1)
    IF (BCON .EQ. 0.0D0) THEN
      ZSTR(J) = ZKNOT(J)-ACON*DT
    ELSE
      CON1 = (ACON+BCON*DZ(J-1))*DEXP(BCON*DT)
      CON1 = (CON1-ACON)/BCON
      ZSTR(J) = ZKNOT(J-1)+CON1
    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..Compute cubic spline estimates of concentrations at the
C..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
*****
        DO I = 2, NM1
          WRITE(20,802) (CSTR(I,NU), NU = 1, NU)
        END DO
      802 FORMAT(1X,1P,4E13.5
*****
      END IF

C..
C..Define the transport-transfer matrix coefficients. Then,
C..sequentially call the Thomas Algorithm for solving the N
C..coupled linear tridiagonal systems.
C..
C..First Radionuclide
C..
      CW(NODES) = 0.0D0
      BW(NODES) = EPS*(DSP(1)/DZ(NM1)) + EPSIN*DSO(1)/DXIN
      AW(NODES) = -EPS* DSP(1)/DZ(NM1)
      DW(NODES) = EPSIN*(DSO(1)/DXIN*CIN(1))
C
      DO I = 2, NM1
        DUMCON = DT*(VELOC(I)*VELOC(I)/DAIR(I)) -
&      (PRESSN(I)-PRESSO(I))/PRESSN(I)
&      - DT*VELOC(I)*CON0
        DUMCON = DUMCON/(1.0D0+RET(1))
        DENOM = FDZ(I)
        AW(I) = -DT*RATIO(I-1,1) / DENOM
        BW(I) = 1.0D0 + DT *
&      (LAM(1)+XLSPM(1)
&      + (RATIO(I-1,1)+RATIO(I,1))/DENOM)
        BW(I) = BW(I)+DUMCON
        CW(I) = -FT*RATIO(I,1) / DENOM
        DW(I) = CSTR(I,1) + DT*(1.0D0-EPS)/(EPS*(1.0D0+RET(1))) *
&      (PARTDN * SOURCE(I) +
&      CONLM(1)*SSP(I)*DEXP(-LAM(0)*TNEXT))
      END DO
C
      CW(1) = -EPS*(DSP(1)/DZ(1))
      BW(1) = EPS*(DSP(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))

```

```

CTIR1(1) = (0.5D0*DT)*LTIRR(1)
CTIR1(1) = CTIR1(1)/1.0D0 + 0.5D0*DT*LAM(1))
CTIR2(1) = 0.0D0
C..
C..Second thru Last Radionuclide
C..
DO NU = 2, NUCS
  CW(NODES) = 0.0D0
  BW(NODES) = EPS*(DSP(NU)/DZ(NM1)+EPSIN*(DSO(NU)/DXIN)
  AW(NODES) = -EPS* DSP(NU)/DZ(NM1)
  DW(NODES) = EPSIN*(DSO(NU)/DXIN)*CIN(NU)
C
DO I = 2, NM1
  DUMCON = DT*(VELOC(I)*VELOC(I)/DAIR(I)) -
&          (PRESSN(I)-PRESSO(I))/PRESSN(I)
&          - DT*VELOC(I)*CON0
  DUMCON = DUMCON/(1.0D0+RET(NU))
  DENOM = FDZ(I)
  AW(I) = -DT*RATIO(I-1,NU) / DENOM
  BW(I) = 1.0D0 + DT *
&          (LAM(NU) + XLSPM(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(NU)*CNEW(I,NU-1)
END DO
C
  CW(1) = -EPS*(DSP(NU)/DZ(1))
  BW(1) = EPS*(DSP(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))
  CTIR1(NU) = (0.5D0*DT)*LTIRR(NU)
  CTIR1(NU) = CTIR1(NU) / (1.0D0 + 0.5D0*DT*LAM(NU))
  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
  STIRN(J,1) = CTSIR(1)*STIRO(J,1) +
&          CTIR1(1)*(CNEW(J,1)+COLD(J,1))
  DO NU = 2, NUCS
    STIRN(J,NU) = CTSIR(NU)*STIRO(J,NU) +
&          CTIR1(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

```

```

C..
  DO NU = 1, NUCS
    DO J = 2, NODES
      DZD = DZ(J-1)
      MTF(J-1,NU) = EPS*DZD*AREA*
&      0.5D0*(CNEW(J,NU)+CNEW(J-1,NU)
      MTS(J-1,NU) = (1.0D0-EPS)*DZD*AREA*FREV(NU)*PARTDN*KEQ(NU)*
&      0.5D0*(CNEW(J,NU)+CNEW(J-1,NU))
      MTIRR(J-1,NU) = (1.0D0-EPS)*DZD*AREA*FIRR(NU)*PARTDN*
&      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*DSP(NU)*(CNEW(2,NU)-CNEW(1,NU))/DZ(1)
&      + EPS*VELOC(1)*CNEW(1,NU)
  ELSE
    QFLXN(NU) = - EPSOT*DSOT(NU)*(CNEW(1,NU)-COT(NU))/DXOT
&      + VELOT*COT(NU)
  END IF
C
  QFLXN(NU) = QFLXN(NU)*SA(NU)
  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
Cold  WRITE(20,27) '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))), 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
      IREC = IREC + 1
      CALL BRITE8(IREC)
C
      KKK = KKK + 1
C
      IF (KKK .GT. NPRT) DONE = .TRUE.
      IF (T .GT. TEND) DONE = .TRUE.
      END IF
      IADTH = IADTH - 1
      IF (.NOT. DONE) GO TO 3000
C
      1300 STOP 'Normal Fortran termination'
      4 FORMAT(A)
      22 FORMAT(5X, A, IP, 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)
      END
*VELCTY
      SUBROUTINE VELCTY(COURANT, COURNTOK, COURFAIL)
      INCLUDE 'comsize.v08'
      INCLUDE 'comcons.v08'
      INCLUDE 'comctrl.v08'
      INCLUDE 'comgeom.v08'
      INCLUDE 'compump.v08'
      DOUBLE PRECISION COURANT
      LOGICAL COURNTOK, COURFAIL
C..Logicals--
      INTEGER K
      LOGICAL COURTEST
      DOUBLE PRECISION DZ1, DZ2, P1, P2, P3, PPDZ
*****
* Calculate the gas velocity, based upon the current pressures. *
* There is a check on the Courant number imbedded in these calculations. *
* If the Courant number is exceeded (that is, becomes > 1), two logical *
* flags are set. Upon failure, COURNTOK becomes false, and COURFAIL *
* becomes true. Two logical flags are used (as opposed to one) to make *
* the logic in the time-step reduction scheme easier. *
* D. E. Cawlfeld, Summer 92. *
*****
      COURNTOK = .TRUE.
C..
C..Do first node . . .
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) = - DAIR(1)*PPDZ/PRESSN(1) + KAIR(1)
      COURANT = VELOC(1)*DT/DZ1

```

```

COURTEST = COURNTOK .AND. ABS(COURANT) .GT. 1.0D0
IF (COURTEST THEN
  PRINT *, 'Courant number > 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) = - DAIR(K)*PPDZ/PRESSN(K) + KAIR(K)
  COURANT = VELOC(K)*DT/DZ(K)
  COURTEST = COURNTOK .AND. ABS(COURANT) .G. 1.0D0
  IF (COURTEST) THEN
    PRINT *, 'Courant number > 1', COURANT
    COURFAIL = .TRUE.
    COURNTOK = .FALSE.
  END IF
END DO
VELOC(NODES) = 0.0D0
RETURN
END

```

*READ1
C

Last Revision: June 25, 1992
Source File : C8RREAD1.FOR

```

SUBROUTINE READ1
  INCLUDE 'comsize.v08'
  INCLUDE 'comgeom.v08'
  INCLUDE 'comname.v08'
  INCLUDE 'comasi.v08'
  INCLUDE 'compump.v08'
  INCLUDE 'comrnxp.v08'
  INCLUDE 'comctrl.v08'
  INCLUDE 'comnumr.v08'
  INCLUDE 'comcons.v08'
  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 initial data. The routine uses routine GET_ROOT to obtain the *
  * "root" name from the command line. See routine GET_ROOT for possible *
  * system dependencies. *
  * D. E. Cawlfieid, Summer 92. *
  *****
C
  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 CASCADR8.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,*)

*****
* 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. *
*****
READ(10,4) DUMMY
READ(10,*) (NFLAG(I),I=1,10)

C..
READ(10,4) DUMMY
READ(10,*) NODES, NUCS, NOUT
NM1 = NODES-1
NM2 = NM1 -1
IF (NODES .GT. NZSET) THEN

```

```

      PRINT *, 'Too many nodes, change NZSET and re-compile.'
      STOP 1010
    END IF
    IF (NUCS .GT. MNUC) THEN
      PRINT *, 'Too many nuclides, change MNUC and re-compile.'
      STOP 1020
    END IF

```

C..
 C..The following code branches depending whether the indicator
 C..variable CZC requests absolute or relative coordinates.
 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 = ZNOT(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 "RR"!
        STOP 1040
      END IF

```

C..
 C..The following code branches depending upon whether the indicator
 C..variable CZC request nodes or z coordinates.
 C..

```

      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
END IF

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)
    END DO

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

C..
    WRITE(*,*)
    WRITE(*,*) 'Approximate print out times are: '
    WRITE(*,8) (TPRT(I), I=1, NPRT)
    WRITE(20,*) 'Approximate print out times are: '
    WRITE(20,8) (TPRT(I), I=1, NPRT)
    WRITE(20,*)
    WRITE(20,*)
    8 FORMAT(10X,1P,5E12.4)

C
C III. SOURCE REGION AND POROUS MEDIUM TRANSPORT AND FATE DATA
C
    WRITE(*,*)
    WRITE(*,*) ' II. POROUS MEDIUM DATA. '
    WRITE(*,*)
    READ(10,4) DUMMY
    READ(10,*) PARTDN
    READ(10,4) DUMMY

```

```

READ(10,*) EPSIN, EPS, EPSOT
READ(10,4) DUMMY
READ(10,*) (DSO (NU), NU = 1, NUCS)
READ(10,4) DUMMY
READ(10,*) (DSP (NU), NU = 1, NUCS)
READ(10,4) DUMMY
READ(10,*) (DSOT (NU), NU = 1, NUCS)
READ(10,4) DUMMY
READ(10,*) (FREV (NU), NU = 1, NUCS)
READ(10,4) DUMMY
READ(10,*) (FIRR (NU), NU = 1, NUCS)
READ(10,4) DUMMY
READ(10,*) (KEQ (NU), NU = 1, NUCS)
C..Half-lives, then converted to decay constants...
READ(10,4) DUMMY
READ(10,*) (THALF (NU), NU = 0, NUCS)
DO NU = 0, NUCS
  IF (THALF (NU) .NE. 0.D0) 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
READ(10,*) (LTIRR (NU), NU = 1, NUCS)
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,*) KAPPA, TAIR, RG, MASSA, GRAV, RHOWAT
  READ(10,4) DUMMY
  READ(10,*) PA, 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..Automatically create file extensions of ".o24", ".o25", etc.
C..
      LU = 24
      DO NU = 1, MIN(NUCS,4)
        FNAME = ROOT(1:L12) // '.ONN'
        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, ' = ', IP, G13.6,
& ' is <= Zero at J = ', OP, I6, '!' //)
      END
*READ2
      SUBROUTINE READ2
      INCLUDE 'comsize.v08'
      INCLUDE 'comgeom.v08'
      INCLUDE 'compump.v08'
      INCLUDE 'comrxnp.v08'
      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. Cawlfeld, 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
        END IF
        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
        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
        END DO
      ELSE
        PRINT *, '*** READ2: Unknown flag, not ''N'' or ''Z''!'
        STOP 2070
      END IF
      RETURN
4  FORMAT(A)
END
*JFROMZ
      INTEGER FUNCTION JFROMZ(Z)
      INCLUDE 'comsize.v08'
      INCLUDE 'comgeom.v08'
      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.v08'
      INCLUDE 'comgeom.v08'
      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 *
* 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, *
* and a dummy routine for GETARG. The program will then request the "root" *
* name from the terminal. *
*
* NOTE: On most Unix based systems, there is no "STATUS" parameter in *
* routine GETARG. This will create problems on these systems unless this *
* parameter is removed from the call. *
*
*
* D. E. Cawlfild, 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: June 25, 1992
Source File: C8REPT1.FOR

```

SUBROUTINE REPT1
  INCLUDE 'comsize.v08'
  INCLUDE 'comgeom.v08'
  INCLUDE 'comname.v08'
  INCLUDE 'comasi.v08'
  INCLUDE 'compump.v08'
  INCLUDE 'comrnxp.v08'
  INCLUDE 'comctrl.v08'
  INCLUDE 'comnumr.v08'
  INCLUDE 'comcons.v08'
  INTEGER NU
*****
* Write various control parameters to the report file. Done once at the *
* beginning of a run, when requested. *
*
* D. E. Cawlfild, Spring 92.
*****
WRITE(20,*)
  WRITE(20,*) ' Writing out system control constants. '
  WRITE(20,*)
  DO NU = 1, NUCS
    WRITE(20,222) ' CTSIR(' , NU, ' ) is ' , CTSIR(NU)
    WRITE(20,222) ' CTIR1(' , NU, ' ) is ' , CTIR1(NU)
    WRITE(20,222) ' CTIR2(' , NU, ' ) is ' , CTIR2(NU)
    WRITE(20,222) ' XLSPM(' , NU, ' ) is ' , XLSPM(NU)
    WRITE(20,222) ' CONLM(' , NU, ' ) is ' , CONLM(NU)
  
```



```

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, 11.1, A 1P, E13.5)
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.v08'
INCLUDE 'comgeom.v08'
CHARACTER CH1, CH2
INTEGER IP1, IP2, N, N2, LUN
*****
* Write a table of nodes to the output file. These are listed in two broad *
* columns down the page. A special symbol(">") is printed at the nodes *
* which are used as columns in the various output reports. *
* D. E. Cawlfeld, Summer 92. *
*****
* LUN = 20
IF (NFLAG(10) .EQ. 1) LUN = 06
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)
        END IF
        IF (N2+N2) .EQ. NDEPTS(IP2)) THEN
          CH2 = '>'
          IP2 = MIN(IP2,NCOL)
        END IF
        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

Last Revision: June 23, 1992

C

Source File: C8BRITE.FOR

```

      SUBROUTINE BRITE_INIT(IREC)
      INTEGER          IREC
      INCLUDE 'comsize.v08'
      INCLUDE 'comgeom.v08'
      INCLUDE 'comname.v08'
      INCLUDE 'comasi.v08'
      INCLUDE 'compump.v08'
      INCLUDE 'comrnxp.v08'
      INCLUDE 'comctrl.v08'
      INCLUDE 'comnumr.v08'
      INCLUDE 'comcons.v08'
      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

```

*BRITE8

```

SUBROUTINE BRITE8(IREC)
INTEGER IREC
INCLUDE 'comsize.v08'
INCLUDE 'comgeom.v08'
INCLUDE 'comname.v08'
INCLUDE 'comasi.v08'
INCLUDE 'compump.v08'
INCLUDE 'comrnxp.v08'
INCLUDE 'comctrl.v08'
INCLUDE 'comnumr.v08'
INCLUDE 'comcons.v08'

```

```

INTEGER I, NU

```

```

*****
* Write the current state of the run onto the binary output file. *
* D. E. Cawlfeld, Mar 92 *
*****

```

```

WRITE(60) IREC, TOUT
WRITE(60) (PRESSN(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

```

*AIRPUMP

C

```

SUBROUTINE AIRPUMP
INCLUDE 'comsize.v08'
INCLUDE 'comgeom.v08'
INCLUDE 'comasi.v08'
INCLUDE 'compump.v08'
INCLUDE 'comctrl.v08'
INCLUDE 'comnumr.v08'
INCLUDE 'comcons.v08'

```

C..Locals --

Last Revision: August 11, 1992
Source File: AIRPUMP8.FOR

```

      DOUBLE PRECISION SUMPRS, TNOW
      INTEGER I
*****
*   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,+∞}.
*****
C..
      SUMPRS = 0.0D0
      TKNOW = T+DT
      DO I = 1, NPRES
        SUMPRS = SUMPRS + POC(I)*DCOS(OMEG(I)*TNOW +
*          POS(I)*DSIN(OMEG(I)*TNOW
      END DO
      PRESSN(1) = RHOWAT*GRAV*13.6D0*(PA + SUMPRS)
C..
      DD(1) = PRESSN(1)
      DO I = 2, NM1
        DD(I) = PRESSO(I)
      END DO
      DD(NODES) = 0.0D0
C..
      CALL THOMAS(NODES, AA, BB, CC, DD, XNOW, CTEMP, DTEMP)
C..
      DO I = 1, NODES
        PRESSN(I) = XNOW(I)
      END DO
      RETURN
      END

```

*EVAL

Last Revision: April 4, 1992

C

Source File: EVAL8.FOR

```

      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) / (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
END IF
END DO
RETURN
END

```

```

*
C                                         Last Revision: April 4, 1992
                                         Source File: THOMAS8.FOR
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
      RETURN
      END

```

```

*DTS                                         Last Revision: April 2, 1992
C                                         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. Cawlfeld, 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. Cawlfeld, 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

```

Last Update: February 7, 1991

*DATE

```

SUBROUTINE DATE(DAYT)
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. Cawlfeld, 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

```

Last Revision: February 7, 1991

*TIME
C

```

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
* 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. Cawlfeld, 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. Cawlfeld, Spring 92.              *
*****
IARGC = NARGS() - 1
      RETURN
      END

```


APPENDIX D

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."

APPENDIX E

ERROR MESSAGES AND THEIR CAUSES

Unless otherwise indicated, each error message discussed is a precursor to program termination.

- 1) Too many nodes, change NZSET and re-compile.
- 2) Too many nuclides, change MNUC and re-compile.

NZSET and MNUC are FORTRAN variables representing the maximum number of nodes and maximum number of radionuclides, respectively, that the program can handle. Set these parameters when the program is compiled so the FORTRAN compiler can determine the size of some data structures. If either NODES or NUCL are set in the data file to a value higher than its maximum value, one of the above error messages will be seen. These parameters can be increased, but then the program will need to be recompiled. To solve the problem, edit the file "comsize.v08." The line that will need to be changed looks like this:

```
PARAMETER      (NZSET=0401, NPT=600, MNUC=4, MCOL = 10)
```

After editing the parameter values, compile the program, as described in Appendix A. In general, it is good practice not to make these parameters many orders of magnitude larger than necessary, since this will slow the execution process on all machines, and possibly effect the execution process totally on some small machines.

- 3) *** OOPS *** DELTAZ = -0.500000 is <= Zero at J = 3!

The spatial nodes at which CASCADR performs all calculations must be specified, in the data file, in order from high elevation to low. If any nodes are specified out of order, CASCADR interprets this as a negative distance from one node to the next (DELTAZ), which is not allowed. To correct the problem, edit the data file, in the

portion where the nodes are listed, and put all nodes in order. This message is only given when the nodes are listed in the "absolute" mode; of course, when the nodes are listed in "relative" mode, all "DELTAZs" must be positive, as well.

4) *** READ1: Bad Flag, not 'A' or 'R'!

Specification of the computation nodes in the data file is done either in absolute mode or in relative mode, signalled by "A" or "R." If neither flag is found, the above error message will be received. See Chapter 3 for details on formatting the input file.

- 5) *** ZFROMJ: J < 1 ***
- 6) *** ZFROMJ: J > NODES ***
- 7) *** JFROMZ: Z < ZKNOT(1) ***
- 8) *** JFROMZ: Z > ZKNOT(NODES) ***

Messages 5-8 indicate that an attempt has been made to either find the depth of a node that doesn't exist (5,6) or find the node number of a depth that is not within the boundaries of the problem (7,8). These errors do not cause the program to terminate. However, they usually occur in conjunction with errors 9 or 10, which will cause termination.

- 9) *** READ1: Bad print node! ***
- 10) *** READ1: Bad print Znode! ***

Error messages 9 and 10 indicate that a print node has been selected that is not within the node boundaries (first and last node) as defined in the input file. Message 9 occurs in combination with 5 or 6, when print depths are specified by node number; message 10 occurs in combination with 7 or 8, when print depths are specified in absolute distances. To correct the error, edit the data file to insure that all print depths are within the bounds of the problem.

11) *** READ1: Bad Flag, not 'N' nor 'Z'!

Message 11 is similar to message 4, except that it occurs when the format for reading print depths has not been specified properly. 'N' indicates node numbers, 'Z' indicates absolute depths. See Chapter 3.

12) *** READ2: Source region "0" not first! ***

In the input file, the background radiation region must be the first radionuclide source region specified, and it must be numbered 0. See Chapter 3.

13) *** READ2: Bad region definition. ***

This means that a radionuclide source region is specified in the input file such that the top of the region is under the bottom. Fix the problem by making sure that all region boundaries are specified in the correct order.

14) *** READ2: Unknown flag, not 'N' or 'Z'!

Message 14 is in the same family with 11 and 4. It applies to radionuclide source region specification.

15) *** IADTP > 6 ***

Message 15 indicates the timestep had to be decreased (by half) more than 6 times. This probably means that the δt chosen was too high and, if the program was to continue, much time would be consumed thrashing back and forth with the timestep. Try decreasing the default δt , in the input file, and rerunning the program.

16) Adjusting NDEPTS(ncol) < NODES.

If the last print depth is below the last node then CASCADR will automatically select the last node as the last print depth, and will issue message 16. This will not cause termination of the program, so long as only one print depth is defined below the last node. If two or more print depths are below the last node, execution will stop, with error message 9 or 10.

17) Courant number > 1

The courant number is the measure used to determine whether the current timestep is small enough to guarantee enough accuracy. When the courant number is above 1, this means that the timestep should be reduced. This message is then sent to the screen, and execution continues. Therefore, message 17 is not an error message, per se, but it is included here since it serves to inform the user about the status of the simulation.

APPENDIX F

VERIFICATION OF CASCADR8 USING RADON GAS AS A MODEL NOBLE GAS

Introduction

An important part of the QA/QC of mathematical models of fate and transport in porous media is the verification step. By verification we mean that the output of the numerical model has been verified or checked against (compared with) the output of an analytical solution for the same underlying transport and fate equations for the same initial and boundary conditions. Most large scale simultaneous water and heat transport in the vadose zone models have never been verified in the above sense of the word. To date, there are no known analytical solutions to these nonlinear sets of equations. Fortunately, CASCADR8 is a linear model. Furthermore, if the problem is specialized somewhat by assuming a constant barometric pressure at the air-alluvium interface, then the induced advection term vanishes. If the noble gas diffusion coefficient is constant in space, the alluvium is homogeneous (automatically satisfied in CASCADR8), and that first order irreversible sorption of radon can be neglected, the basic transport and fate equation for radon gas in dry alluvium reduces to the simple parabolic form

$$\begin{aligned} \epsilon (1+R_1) \frac{\partial C_1}{\partial t} + \epsilon (1+R) \lambda_1 C_1 = \\ \epsilon D_1 \frac{\partial^2 C_1}{\partial x^2} + (1-\epsilon) S(x) \left(\frac{M_1}{M_0} \right) + \epsilon (1+R_1) \lambda_1 C_0 \end{aligned} \quad (F.1)$$

Multiplication of both sides by $1/\epsilon (1+R_1)$ with subsequent definition of the effective radon gas diffusion coefficient

$$D_{\text{eff}} = \frac{D}{1+R_1} \quad (F.2)$$

obtains the simplified form

$$\frac{\partial C_1}{\partial t} = D_{\text{eff}} \frac{\partial^2 C_1}{\partial x^2} - \lambda_1 C_1 + \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \left(\frac{S(x)}{1+R_1} \right) + \lambda C_1 . \quad (\text{F.3})$$

For the sake of consistency with the text we chose the manmade source function $S(x)$ to be

$$S(x) = \begin{cases} S_0, & x_T < x < x_B \\ 0, & \text{elsewhere} \end{cases} \quad (\text{gm/cm}^3\text{-sec}) \quad (\text{F.4})$$

where x_T is the distance in centimeters from the surface to the top of the waste region and x_B is the distance in centimeters from the surface to the bottom of the waste region. Without loss of generality, it is assumed that initially there is no radon in the alluvium. Actually, there is a naturally occurring source of radon gas in most soils. The definition of this source was made early in the text of Lindstrom et al. (1992a). In fact, since the general transport and fate model equations (Section 2) in their text are linear models, the solution C , can be superposed; that is, split into the sum of two solutions, one solution arising as the solution of the boundary value problem:

$$\frac{\partial C_{11}}{\partial t} + \lambda_1 C_{11} = D_{\text{eff}} \frac{\partial^2 C_{11}}{\partial x^2} + \lambda_0 C_0 , \quad (\text{F.5})$$

with initial condition

$$C_{11}(x, 0^+) = C_0, \quad 0 < x < \infty , \quad (\text{F.6})$$

and boundary conditions

$$\text{i) } C_{11}(0, t) = 0, \quad t > 0 , \quad (\text{F.7})$$

$$\text{ii) } \lim_{x \rightarrow \infty} \frac{\partial C_{11}}{\partial x} = 0 \text{ and } \lim_{x \rightarrow \infty} C_{11} < \infty , \quad (\text{F.8})$$

while the other solution arises as the solution to the boundary value problem:

$$\frac{\partial C_{12}}{\partial t} + \lambda_1 C_{12} = D_{\text{eff}} \frac{\partial^2 C_{12}}{\partial x^2} + \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \frac{S(x)}{1+R_1} \quad (\text{F.9})$$

with initial condition

$$C_{12}(x, 0^+) = 0, \quad 0 < x < \infty, \quad (\text{F.10})$$

and boundary conditions

$$\text{i) } C_{12}(0, t) = 0, \quad t > 0, \quad (\text{F.11})$$

$$\text{ii) } \lim_{x \rightarrow \infty} \frac{\partial C_{12}}{\partial x} = 0 \quad \text{and} \quad \lim_{x \rightarrow \infty} C_{12} < \infty. \quad (\text{F.12})$$

Then the complete solution is simply

$$C_1(x, t) = C_{11}(x, t) + C_{12}(x, t). \quad (\text{F.13})$$

For verification purposes the match between the numerical model and the analytical solution for equation F.9, "the buried manmade source scenario," was selected for demonstration. Therefore, it is required to obtain an analytical solution for partial differential equation F.9 subject to initial condition F.10 and boundary conditions F.11 and F.12.

Solution Method

The method chosen to find the analytic solution is that of Laplace transforms (McLachlan, 1963). Transforming equation F.9 obtains

$$s \hat{C} + \lambda_1 \hat{C} = D_{\text{eff}} \frac{d^2 \hat{C}}{dx^2} + \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \frac{S(x)}{s(1+R_1)} \quad (\text{F.14})$$

where s is the transform variable. Using the definition of $S(x)$, (equation F.4), equation F.14 can be written as

$$(s + \lambda_1) \hat{C} = D_{\text{eff}} \frac{d^2 \hat{C}}{dx^2}, \quad 0 < x < x_T \quad (\text{F.15})$$

$$(s + \lambda_1) \hat{C} = D_{\text{eff}} \frac{d^2 \hat{C}}{dx^2} + \left(\frac{1 - \varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \frac{S_0}{s + R_1}, \quad x_T < x < x_B \quad (\text{F.16})$$

$$(s + \lambda_1) \hat{C} = D_{\text{eff}} \frac{d^2 \hat{C}}{dx^2}, \quad x > x_B \quad (\text{F.17})$$

with the continuity requirements at x_T and x_B being:

$$\hat{C}(x_T^-, s) = \hat{C}(x_T^+, s) \quad (\text{F.18})$$

$$\left. \frac{d\hat{C}}{dx} \right|_{x_T^-} = \left. \frac{d\hat{C}}{dx} \right|_{x_T^+} \quad (\text{F.19})$$

$$\hat{C}(x_B^-, s) = \hat{C}(x_B^+, s) \quad (\text{F.20})$$

$$\left. \frac{d\hat{C}}{dx} \right|_{x_B^-} = \left. \frac{d\hat{C}}{dx} \right|_{x_B^+} \quad (\text{F.21})$$

The solution to the system of equations F.15 - F.17 subject to the conditions F.18 - F.21 can be shown to be the same as the solution of F.14 via a standard exercise in ordinary differential equations. The three subregions on which transformed distributions F.15 - F.17 are shown in Figure F.1.

The solutions of these three ordinary differential equations are:

$$\hat{C}(x, s) = \alpha_1 e^{\gamma x} + \beta_1 e^{-\gamma x}, \quad 0 < x < x_T \quad (\text{F.22})$$

$$\hat{C}(x,s) = \alpha_2 e^{\gamma x} + \beta_2 e^{-\gamma x} + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{s(s+\lambda_1)(1+R_1)}, \quad x_T < x < x_B \quad (F.23)$$

$$\hat{C}(x,s) = \alpha_3 e^{\gamma x} + \beta_3 e^{-\gamma x}, \quad x > x_B \quad (F.24)$$

where

$$\gamma = \sqrt{\frac{s+\lambda_1}{D_{\text{eff}}}} \quad (F.25)$$

satisfaction of the well-behavedness condition F.12 requires $\alpha_3 \equiv 0$. Application of the top boundary condition F.11 obtains, $\alpha_1 = -\beta_1$. Thus, the above system of equations can be rewritten as

$$\hat{C}(x,s) = \beta_1 (e^{-\gamma x} - e^{\gamma x}), \quad 0 < x < x_T, \quad (F.26)$$

$$\hat{C}(x,s) = \alpha_2 e^{\gamma x} + \beta_2 e^{-\gamma x} + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{s(s+\lambda_1)(1+R_1)}, \quad x_T < x < x_B, \quad (F.27)$$

$$\hat{C}(x,s) = \beta_3 e^{-\gamma x}, \quad x > x_B. \quad (F.28)$$

Figure F.1 The domain of definition of the transport and fate problem.

Application of equations F.18 - F.21 obtains the system of four equations in four unknowns:

$$\beta_1 e^{-\gamma x_T} - \beta_1 e^{\gamma x_T} = \alpha_2 e^{\gamma x_T} + \beta_2 e^{-\gamma x_T} + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{s(s+\lambda_1)(1+R_1)} , \quad (F.29)$$

$$- \beta_1 e^{-\gamma x_T} - \beta_1 e^{\gamma x_T} = \alpha_2 e^{\gamma x_T} - \beta_2 e^{-\gamma x_T} , \quad (F.30)$$

$$\alpha_2 e^{\gamma x_B} + \beta_2 e^{-\gamma x_B} = \beta_3 e^{-\gamma x_B} - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{s(s+\lambda_1)(1+R_1)} , \quad (F.31)$$

$$\alpha_2 e^{\gamma x_B} - \beta_2 e^{-\gamma x_B} = - \beta_3 e^{-\gamma x_B} . \quad (F.32)$$

Solution of equations F.29 - F.32 obtains

$$\beta_1 = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} (e^{-\gamma x_B} - e^{-\gamma x_T}) , \quad (F.33)$$

$$\alpha_1 = \frac{-S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} e^{-\gamma x_B} , \quad (F.34)$$

$$\beta_2 = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} (e^{-\gamma x_B} - e^{-\gamma x_T} - e^{\gamma x_T}) , \quad (F.35)$$

$$\beta_3 = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} (e^{\gamma x_B} + e^{-\gamma x_B} - e^{-\gamma x_T} - e^{\gamma x_T}) . \quad (F.36)$$

Back substitution for the coefficients yields

$$\hat{C}(x,s) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} \left(e^{-\gamma(x+x_B)} - e^{-\gamma(x_B-x)} \right. \\ \left. - e^{-\gamma(x_T+x)} + e^{-\gamma(x_T-x)} \right), \quad 0 \leq x \leq x_T, \quad (F.37)$$

$$\hat{C}(x,s) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} \left(2 - e^{-\gamma(x_B-x)} + e^{-\gamma(x+x_B)} \right. \\ \left. - e^{-\gamma(x+x_T)} - e^{-\gamma(x-x_T)} \right), \quad x_T \leq x \leq x_B, \quad (F.38)$$

$$\hat{C}(x,s) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2s(s+\lambda_1)(1+R_1)} \left(e^{-\gamma(x-x_B)} \right. \\ \left. + e^{-\gamma(x+x_B)} - e^{-\gamma(x+x_T)} - e^{-\gamma(x-x_T)} \right), \quad x \geq x_B. \quad (F.39)$$

Application of the inversion theorem (McLachlan, 1963) to transformed distributions

F.37 - F.39 obtains:

for $0 \leq x \leq x_T$,

$$C(x,t) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x+x_B)\sqrt{\frac{\lambda_1}{D_{eff}}}} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{eff}t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\ \left. \left. + e^{(x+x_B)\sqrt{\frac{\lambda_1}{D_{eff}}}} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{eff}t}} + \sqrt{\lambda_1 t} \right) \right) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{eff}t}} \right) \right\}$$

$$\begin{aligned}
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} \left\{ \frac{1}{2} \left[e^{-\left(x_B-x\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x_B-x}{2\sqrt{D_{eff}t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& \quad \left. \left. + e^{\left(x_B-x\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x_B-x}{2\sqrt{D_{eff}t}} + \sqrt{\lambda_1 t} \right) \right] - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x_B-x}{2\sqrt{D_{eff}t}} \right) \right\} \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} \left\{ \frac{1}{2} \left[e^{-\left(x+x_T\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{eff}t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& \quad \left. \left. + e^{\left(x+x_T\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{eff}t}} + \sqrt{\lambda_1 t} \right) \right] - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{eff}t}} \right) \right\} \\
& + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} \left\{ \frac{1}{2} \left[e^{-\left(x_T-x\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x_T-x}{2\sqrt{D_{eff}t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& \quad \left. \left. + e^{\left(x_T-x\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x_T-x}{2\sqrt{D_{eff}t}} + \sqrt{\lambda_1 t} \right) \right] - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x_T-x}{2\sqrt{D_{eff}t}} \right) \right\}, \tag{F.40}
\end{aligned}$$

for $x_T \leq x \leq x_B$,

$$\begin{aligned}
C(x,t) = & \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} (2 - 2e^{-\lambda_1 t}) \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2\lambda_1(1+R_1)} \left\{ \frac{1}{2} \left[e^{-\left(x_B-x\right) \sqrt{\frac{\lambda_1}{D_{eff}t}}} \operatorname{erfc} \left(\frac{x_B-x}{2\sqrt{D_{eff}t}} - \sqrt{\lambda_1 t} \right) \right. \right.
\end{aligned}$$

$$\begin{aligned}
& + e^{(x_B - x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_B - x}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x_B - x}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x_B + x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_B + x}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x_B + x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_B + x}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x_B + x}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x_T + x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_T + x}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x_T + x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_T + x}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x_T + x}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x - x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x - x_T}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x - x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x - x_T}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x - x_T}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} , \tag{F.41}
\end{aligned}$$

and for $x \geq x_B$,

$$\begin{aligned}
C(x,t) = & \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x-x_B) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x-x_B}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x-x_B) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x-x_B}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x-x_B}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& + \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x+x_B) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x+x_B) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x+x_B}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x+x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x+x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x+x_T}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} \\
& - \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ \frac{1}{2} \left(e^{-(x-x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x-x_T}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) \right. \right. \\
& + e^{(x-x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x-x_T}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) \Bigg) - e^{-\lambda_1 t} \operatorname{erfc} \left(\frac{x-x_T}{2\sqrt{D_{\text{eff}} t}} \right) \Bigg\} .
\end{aligned} \tag{F.42}$$

We have need of the large time t or limiting forms of F.40 - F.42. As $t \rightarrow \infty$ we have

$$\lim_{t \rightarrow \infty} \operatorname{erfc} \left(\frac{x - x_T}{2\sqrt{D_{\text{eff}} t}} - \sqrt{\lambda_1 t} \right) = \begin{cases} 2, & x \geq x_T, \lambda_1 > 0 \\ 1, & x \geq x_T, \lambda_1 = 0 \end{cases}$$

and

$$\lim_{t \rightarrow \infty} \operatorname{erfc} \left(\frac{x - x_T}{2\sqrt{D_{\text{eff}} t}} + \sqrt{\lambda_1 t} \right) = \begin{cases} 0, & x \geq x_T, \lambda_1 > 0 \\ 1, & x \geq x_T, \lambda_1 = 0 \end{cases}$$

The remaining complementary error functions have similar limiting forms for large time values. Thus, we can write

$$\lim_{t \rightarrow \infty} C(x, t) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ e^{-(x+x_B) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} - e^{-(x_B-x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right. \\ \left. - e^{-(x+x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} + e^{-(x_T-x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right\} \quad (\text{F.43})$$

for $0 \leq x \leq x_T$,

$$\lim_{t \rightarrow \infty} C(x, t) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ 2 - e^{-(x_B-x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} + e^{-(x_B+x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right. \\ \left. - e^{-(x_T+x) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} - e^{-(x-x_T) \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right\} \quad (\text{F.44})$$

for $x_T \leq x \leq x_B$, and

$$\lim_{t \rightarrow \infty} C(x,t) = \frac{S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right)}{2 \lambda_1 (1+R_1)} \left\{ e^{-\frac{(x-x_B) \sqrt{\lambda_1}}{\sqrt{D_{eff}}}} + e^{-\frac{(x+x_B) \sqrt{\lambda_1}}{\sqrt{D_{eff}}}} - e^{-\frac{(x+x_T) \sqrt{\lambda_1}}{\sqrt{D_{eff}}}} - e^{-\frac{(x-x_T) \sqrt{\lambda_1}}{\sqrt{D_{eff}}}} \right\} \quad (F.45)$$

for $x \geq x_B$.

CASCADR8 calculates the flux of radon gas off the surface $x=0$, the atmosphere-alluvium interface. The time dependent flux at the surface is defined as

$$F(t) = -\varepsilon D \left. \frac{\partial C}{\partial x} \right|_{x=0} \quad (SA) \quad (F.46)$$

In the transform F can be written as

$$\hat{F}(s) = -\varepsilon D \left. \frac{\partial \hat{C}}{\partial x} \right|_{x=0} \quad (SA) \quad (F.47)$$

substitution for $\hat{C}$ from F.37 into equation F.47 yields

$$\hat{F}(s) = \frac{-\varepsilon D S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \gamma}{s(s+\lambda_1)(1+R_1)} \left\{ e^{-\gamma x_T} - e^{-\gamma x_B} \right\} \quad (F.48)$$

Application of the inversion theorem gives

$$F(t) = -\varepsilon S_0 \left(\frac{1-\varepsilon}{\varepsilon} \right) \left(\frac{M_1}{M_0} \right) \sqrt{\frac{D}{1+R_1}} \frac{e^{-\lambda_1 t}}{2\pi i} \int_{B_{r1}} \frac{e^{pt}}{\sqrt{p(p-\lambda)}} \left(e^{-x_T \sqrt{\frac{p}{D_{eff}}}} - e^{-x_B \sqrt{\frac{p}{D_{eff}}}} \right) dp \quad (F.49)$$

From the tables of integral transforms (Abramowitz and Stegun, 1965) the flux function becomes

$$\begin{aligned}
 F(t) = & \frac{-\epsilon S_0 \left(\frac{1-\epsilon}{\epsilon} \right) \left(\frac{M_1}{M_0} \right) \sqrt{\frac{D}{1+R_1}}}{2\sqrt{\lambda_1}} \left\{ e^{-x_T \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_T}{2\sqrt{D_{\text{eff}}t}} - \sqrt{\lambda_1 t} \right) \right. \\
 & - e^{x_T \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_T}{2\sqrt{D_{\text{eff}}t}} + \sqrt{\lambda_1 t} \right) - e^{-x_B \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_B}{2\sqrt{D_{\text{eff}}t}} - \sqrt{\lambda_1 t} \right) \\
 & \left. + e^{x_B \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc} \left(\frac{x_B}{2\sqrt{D_{\text{eff}}t}} + \sqrt{\lambda_1 t} \right) \right\} . \quad (F.50)
 \end{aligned}$$

At large time values the surface radon flux assumes the simple form

$$\lim_{t \rightarrow \infty} F(t) = \frac{-\epsilon S_0 \left(\frac{1-\epsilon}{\epsilon} \right) \left(\frac{M_1}{M_0} \right) \sqrt{D_{\text{eff}}}}{\sqrt{\lambda_1}} \left\{ e^{-x_T \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} - e^{-x_B \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right\} . \quad (F.51)$$

Setting $x_B = x_T + L_T$, where L_T is the thickness of the source region (meters), substitution into the above long time limiting form for the surface flux of radon gas obtains

$$F(\infty) = \frac{-\epsilon S_0 \left(\frac{1-\epsilon}{\epsilon} \right) \left(\frac{M_1}{M_0} \right) \sqrt{D_{\text{eff}}}}{\sqrt{\lambda_1}} e^{-x_T \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \left(1 - e^{-L_T \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right) . \quad (F.52)$$

The character of the long time surface behavior is clearly shown in equation F.52 to be exponential with depth of burial. This equation can be used to help design disposal pits or trenches. This equation also shows that the long time surface flux is proportional to $\sqrt{D_{\text{eff}}}$. Clearly, by changing the permeability of the dry porous medium (porosity ϵ) the flux will change without affecting the exponential.

Before proceeding to the example verification distributions, we briefly develop the solution to the natural background source portion of the overall problem. That is, we outline the method for solving equation F.5 subject to conditions F.6 - F.8. Application of the Laplace transform method (previously discussed) to equations F.5 with subsequent transformation of the boundary conditions leads to the transformed distribution where

$$\hat{C}(x,s) = A e^{\gamma x} + B e^{-\gamma x} + \frac{C_0}{s},$$

$$\gamma = \sqrt{\frac{s + \lambda_1}{D_{\text{eff}}}} \quad (\text{F.53})$$

Satisfaction of the transformed boundary condition requires that

$$\hat{C}(x,s) = \frac{C_0}{s} - \frac{C_0}{s} e^{-\gamma x} \quad (\text{F.54})$$

Application of the inversion theorem (McLachlan, 1963) obtains

$$C(x,t) = C_0 - \frac{C_0 e^{-\lambda_1 t}}{2\pi i} \int_{B_{r1}} \frac{e^{pt} e^{-x\sqrt{\frac{p}{D_{\text{eff}}}}}}{p - \lambda_1} dp \quad (\text{F.55})$$

Formally, the integrand of the Bromwich contour integral indicated in equation F.54 can be rewritten as

$$\frac{1}{p - \lambda_1} = \frac{1}{2\sqrt{\lambda_1}} \left(\frac{1}{\sqrt{p} - \sqrt{\lambda_1}} - \frac{1}{\sqrt{p} + \sqrt{\lambda_1}} \right) \quad (\text{F.56})$$

so that

$$C(x,t) = C_0 - \frac{C_0 e^{-\lambda_1 t}}{2\sqrt{\lambda_1}} \frac{1}{2\pi i} \int_{B_{r1}} \frac{e^{pt} e^{-x\sqrt{\frac{p}{D_{\text{eff}}}}}}{\sqrt{p} - \sqrt{\lambda_1}} dp + \frac{C_0 e^{-\lambda_1 t}}{2\sqrt{\lambda_1}} \frac{1}{2\pi i} \int_{B_{r1}} \frac{e^{pt} e^{-x\sqrt{\frac{p}{D_{\text{eff}}}}}}{\sqrt{p} + \sqrt{\lambda_1}} dp \quad (\text{F.57})$$

These integrals can be found in Abramowitz and Stegun (1965) via formula number 29.3.88. We have then

$$C(x,t) = C_0 - \frac{C_0}{2} \left\{ e^{-x\sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc}\left(\frac{x}{2\sqrt{D_{\text{eff}}t}} - \sqrt{\lambda_1 t}\right) + e^{x\sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \operatorname{erfc}\left(\frac{x}{2\sqrt{D_{\text{eff}}t}} + \sqrt{\lambda_1 t}\right) \right\} . \quad (\text{F.58})$$

The long time or equilibrium concentration distribution is readily found from equation F.58 by computing

$$C(x,\infty) = \lim_{t \rightarrow \infty} C(x,t) = C_0 \left(1 - e^{-x\sqrt{\frac{\lambda_1}{D_{\text{eff}}}}} \right) . \quad (\text{F.59})$$

Recall that the transformed surface flux $\hat{F}(s)$ is defined as

$$\hat{F}(s) = -\varepsilon D \left. \frac{\partial \hat{C}(x,s)}{\partial x} \right|_{x=0} \quad (\text{F.47})$$

where the specific activity multiplier (SA) has been suppressed without loss of generality.

Computing $\left. \frac{\partial \hat{C}}{\partial x} \right|_{x=0}$ by using equation F.53 obtains

$$\hat{F}(s) = -\frac{\varepsilon D C_0}{s} \sqrt{\frac{s + \lambda_1}{D_{\text{eff}}}} . \quad (\text{F.60})$$

Application of the inversion theorem yields

$$F(t) = -\frac{\varepsilon D C_0 e^{-\lambda_1 t}}{\sqrt{D_{\text{eff}}}} \frac{1}{2\pi i} \int_{B_{r1}} \frac{e^{pt} \sqrt{p}}{p - \lambda} dp , \quad (\text{F.61})$$

which may formally be rewritten as

$$F(t) = - \frac{\varepsilon D C_0 e^{-\lambda_1 t}}{\sqrt{D_{\text{eff}}}} \frac{\partial}{\partial t} \left(\frac{1}{2\pi i} \int_{B_{r1}} \frac{e^{pt} \delta p}{\sqrt{p} (p - \lambda_1)} \right), \quad (\text{F.62})$$

or

$$F(t) = - \frac{\varepsilon D C_0 e^{-\lambda_1 t}}{2\sqrt{D_{\text{eff}}} \sqrt{\lambda_1}} \frac{\partial}{\partial t} \left(\frac{1}{2\pi i} \int_{B_{r1}} \left(\frac{e^{pt}}{\sqrt{p} (\sqrt{p} - \sqrt{\lambda_1})} - \frac{e^{pt}}{\sqrt{p} (\sqrt{p} + \sqrt{\lambda_1})} \right) dp \right), \quad (\text{F.63})$$

or

$$F(t) = - \frac{\varepsilon D C_0 e^{-\lambda_1 t}}{\sqrt{D_{\text{eff}}} \lambda_1} \frac{\partial}{\partial t} \left(e^{\lambda_1 t} - e^{\lambda_1 t} \operatorname{erfc} \sqrt{\lambda_1 t} \right). \quad (\text{F.64})$$

Carrying out the indicated differentiation yields

$$F(t) = - \frac{\varepsilon D C_0}{\sqrt{D_{\text{eff}}} \lambda_1} \left\{ \lambda_1 - \lambda_1 \operatorname{erfc} (\sqrt{\lambda_1 t}) + e^{-\lambda_1 t} \sqrt{\frac{\lambda_1}{\pi t}} \right\}. \quad (\text{F.65})$$

Again, we are interested in the equilibrium or steady-state surface flux of radon gas.

Thus, taking the limit on both sides of equation F.65 yields

$$F(\infty) = \lim_{t \rightarrow \infty} F(t) = -\varepsilon D C_0 \sqrt{\frac{\lambda_1}{D_{\text{eff}}}}, \quad (\text{F.66})$$

as the formula for computing the equilibrium surface flux of radon gas in the case of a constantly emitting uniformly distributed ($0 < x < +\infty$) background source of radon in the porous medium.

Verification Scenario (Base Burial Design)

The Westinghouse Materials Co. of Ohio (WEMCO) thorium waste stream currently approved for disposal at the NTS consists of thorium hydroxide and thorium oxides packaged from feed materials stored in Fernald, Ohio. Table F.1 summarizes thorium-230, the waste currently being received from WEMCO.

Table F.1 Thorium-230 waste stream summary.

approximate number of containers	1600
container type	48-gallon drums in 55-gallon overpacks
estimated total volume of drums	339 m ³
estimated total volume of waste	291 m ³
total mass of waste	288,774 kg
total mass Th-230 (impurity)	99 g
total activity	2.0 Ci in secular equilibrium with Ra-226
specific activity	6.9 (μCi/kg)
mean activity/drum volume	5.9 mCi/m ³
mean activity/waste volume	6.9 Mci/m ³
mean mass/drum	180 kg
mean activity/drum	1.2 Mci

Table F.2 details the thorium-230 decay chain physical data necessary to characterize the source term for radon-222. A separate chemical mass-based Bateman (1910) decay chain model called CHAIN1, which is detailed in Appendix B of Lindstrom et al. (1992b), was run to simulate a period of 10,000 years. Figure F.2 shows the ingrowth of radium-226 in grams given 99 grams of thorium-230 at time t=0.

Table F.2 Thorium-230 chain data (Brown and Firestone, 1986).

Radionuclide	Half Life	Atomic Mass (gm/mole)	Branching Ratio	Specific Activity (pCi/gm)
Th-230	7.7×10^4 yrs	230	1.0	2.0202×10^{10}
Ra-226	1,600 yrs	226	1.0	0.989×10^{12}
Rn-220	3.8235 days	220	1.0	1.5377×10^{17}

The mass of radium-226 currently available at the 10,000 year mark (3.15576×10^{12} sec) is computed to be 1.86 gm. By 10,000 years we are essentially at the peak

RA-226 INGROWTH (GMS)

99.0 gms Th-230 orig.

Figure F.2. Ingrowth of Ra-226 from 99 grams of Th-230 at time t=0 years.

in the radium-226 ingrowth curve. Thus, it was assumed that 1.86 gm of radium-226 will be constantly available in the 99 gm of low-level thorium waste.

The Area 5 RWMS Operations Manager decided to stack the thorium waste packages in Trench 3 in two rectangular parallelepiped nests 25 feet wide, 45 feet long, and 16 feet high. Because the bottom of Trench 3 is 25 feet below grade, the waste will be nine feet below grade when fully stacked, this is called the base six case scenario. The source term for radium-226 was then assumed to exist in the worst possible condition; namely, radium-226 is instantaneously and uniformly released within the above-defined volume. Thus, the constant source rate function required by CASCADR8 is obtained as

$$S_{se}(z) = \left(\frac{1.86 \text{ gm}}{5.057 \times 10^8 \text{ cm}^3} \right) \frac{(0.693)}{5.0492 \times 10^{10} \text{ sec}} = 5.048 \times 10^{-20} \quad (\text{gm/cm}^3/\text{sec})$$

The diffusion coefficient for radon-222 in still, dry air is approximately 0.1 cm²/sec (Nazaroff, 1992). Our current site characterization project in Area 5 indicates that the surface soils have an average porosity of 0.37 cm³/cm³ (site characterization report in progress). A tortuosity of 0.5 seems reasonable (Nazaroff, 1992), considering surface moisture contents are approximately 0.05 to 0.07 cm³/cm³ (site characterization report in process). Therefore, we set the radium-222 diffusion coefficient in the dry desert alluvium at 0.05 cm²/sec. This value provides a conservative estimate of radium-222 diffusion in the dry desert soil in Area 5 at the NTS.

It is assumed that radon-222 does not bind to any of the dry alluvial surfaces. Nazaroff (1992) does show that in the presence of pore water radon-222 is appreciably retarded in its diffusion, which is due to the partitioning into and diffusion through water films around the particles. Therefore, it is a conservative assumption that radon-222 is nonbinding.

Having the constant-in-time source rate function defined as per above, the top of the source is placed nine feet below grade and the bottom of the source is 25 feet below grade. In the following simulation the water table is 240 meters below grade. A cap of nine feet of native soil (gas permeability of 4 Darcy) is put over the top of the waste to bring the soil-atmosphere interface up to grade. No credit is taken for any of the waste packaging material.

Table F.3 summarizes the data used in making the 40-day simulation runs of CASCADR8 and the associated analytical solution evaluation (equations F.39 - F.41).

Table F.3 Run parameters for base case scenario.

Analytical Model	Numerical Model
$\varepsilon = 0.37 \text{ cm}^3/\text{cm}^3$	same
$R_1 = 0.0$	
$\lambda_1 = 2.083 \times 10^{-6} \text{ 1/sec}$	
$D_1 = 0.05 \text{ cm}^2/\text{sec}$	
$M_1 = 222.0 \text{ gm}$	
$M_o = 226.0 \text{ gm}$	
$S_o = 5.048 \times 10^{-20} \text{ gm/cm}^3\text{-sec}$	
$SA = 1.5377 \times 10^{17} \text{ pCi/gm}$	
$x_T = 280 \text{ cm}$	
$x_B = 760 \text{ cm}$	
	water table depth = 24,000.0 cm
	$C_{out} = 0.0 \text{ gm/cm}^3$
	$\Delta X_{out} = 100.0 \text{ cm}$
	$D_{out}^1 = 1.0 \times 10^{10} \text{ cm}^2/\text{sec}$
	PA = 917.0 mb (fixed in time)
	$T_{air} = 293 \text{ }^\circ\text{K}$
	$R_g = 8.3143 \times 10^7 \text{ erg/mole} \cdot \text{ }^\circ\text{K}$
	$M_{air} = 29.0 \text{ gm}$
	$g = 960.0 \text{ cm/sec}^2$
	$\rho_w = 1.0 \text{ gm/cm}^3$

The 10 user selected horizons have been specified to cover the source region, as it is this region where the gas-phase concentration of radon-222 is the highest. Figure F.3 shows the comparison of the radon-222 air-phase concentration in pico curies per cm^3 for the analytic and numerical models at the end of 40 simulated days of radon gas evolution. The corresponding radon-222 fluxes at the air-alluvium interface are 0.1178 $\text{pCi}/\text{cm}^2\text{-sec}$ for the numerical model and 0.1195 $\text{pCi}/\text{cm}^2\text{-sec}$ for the analytical model.

Verification Scenarios (Depth of Burial Cases)

The base case just discussed has demonstrated that the analytical and numerical models agree very well at 40 days. Further verification runs made to 400 days (not shown), show an even closer agreement. However, from a design of a burial procedure for thorium-230 waste, the base case scenario exceeds NESHAP (National Emission Standards for Hazardous Air borne Pollutants) by two orders of magnitude. In the spirit of combined model verification and waste disposal protocol an alternative waste disposal protocol based upon changing the source geometry from a rectangular stack of drums 16 feet thick, to a rectangular stack of drums one meter (100 cm) thick and sequentially varying the depth to the center of the now thin source region. Figure F.4 shows the comparisons of the analytical and numerical models used in the design mode (depth of burial). The values of x_T and x_B used in each run are given in Table F.4. The surface fluxes of radon-222 for both the analytical and numerical model are also given in Table F.4.

Tables F.5 and F.6 respectively summarize the natural logarithm of the surface flux estimates at 40 days for both the analytical and the numerical models. Equation F.52 states that $F(\infty)$ is a negative exponential function of depth. Figures F.5 and F.6 show respectively, semi-log plots of the mean surface flux (averaged over 24 hours on day 40) as a function of the mean X_m ($X_m = \frac{x_T + x_B}{2}$) depth of burial of the one meter thick waste region. The statistics shown in Tables F.5 and F.6 as well as the curves plotted in Figures F.5 and F.6 were generated by the commercial software package Quatro Pro 3.0. Inspection of the statistics in Tables F.5 and F.6 show that the slopes are

Figure F.3 Verification run01 base case scenario. Analytical and numerical model comparisons. Forty-day evolution.

Figure F.4 Verification runs 02-06. One meter thick source region. Source rate same as base case scenario. x_T and x_B for each run listed in Table F.4.

Table F.4 Alternative depth of burial scenarios.

Run No.	x_T (cm)	x_B (cm)	Radon-222 Surface Fluxes* (pCi/cm ² -sec)	
			Analytical Model	Numerical Model
02	280	380	5.767×10^{-2}	6.074×10^{-2}
03	660	760	4.914×10^{-3}	5.183×10^{-3}
04	1,120	1,220	2.450×10^{-4}	2.589×10^{-4}
05	1,420	1,520	3.343×10^{-5}	3.538×10^{-5}
06	2,840	2,940	5.836×10^{-10}	6.360×10^{-10}

*NESHAP = emission standards for radon-222 surface flux = 2×10^{-3} pCi/cm²-sec.
 Flux estimates for the analytical model are derived from evaluation of equation F.49 at time $t = 40$ days (3.456×10^6 sec).

Table F.5 Summary of $1n(F[40])$ vs mean depth of burial at 40 days gas evolution.

Analytical Model - Pure Diffusion Case			
Run	Mean Depth (m)	Rate (pCi/sec)	Ln[Rate]
OUT2	3.3	5.768E-02	-2.853E+00
OUT3	7.1	4.910E-03	-5.316E+00
OUT4	11.7	2.450E-04	-8.314E+00
OUT5	14.7	3.344E-05	-1.031E+01
OUT6	28.3	5.836E-10	-2.126E+01

Regression Output			
Constant			0.02509
Std Err of Y Est			0.48909
R Squared			0.99645
No. of Observations			5.00000
Degrees of Freedom			3.00000
X Coefficient(s)		-0.74004	
Std Err of Coef.		0.02551	

Table F.6 Summary of $\ln(-F[40])$ vs mean depth of burial at 40 days gas evolution.

Thorium-230 verification -- 1 m layering			
Run	Mean Depth (m)	Rate (pCi/sec)	Ln[Rate]
C9231502	3.3	6.074E-02	-2.801E+00
C9231503	7.1	5.183E-03	-5.262E+00
C9231504	11.7	2.589E-04	-8.259E+00
C9231505	14.7	3.538E-05	-1.025E+01
C9231506	28.3	6.360E-10	-2.118E+01
Regression Output			
Constant			
Std Err of Y Est			0.06756
R Squared			0.48344
No. of Observations			0.99652
Degrees of Freedom			5.00000
X Coefficient(s)		-0.73865	
Std Err of Coef.		0.02522	

virtually identical. The intercepts (constant on Y-axis) appear to be different. However, the standard error estimates are so large that a t-test fails to reject the hypothesis: the intercepts are different, except at unreasonably large p (significance) values. Therefore, we conclude that the surface flux estimates are not significantly different.

Log[Surface-Flux] vs Depth

Analytical Model - Pure Diffusion Case

Figure F.5 Semi-log plot of mean surface flux of radon-222 vs depth of burial. Analytical case at 40-days of gas evolution.

Log[Surface-Flux] vs Depth Thorium 230 Verification Suite

Figure F.6 Semi-log plot of mean surface flux of radon-222 vs depth of burial. Numerical case at 40-days of gas evolution.

REFERENCES

- Abramowitz, M. and I. A. Stegun. 1965. Handbook of mathematical functions. Dover, NY. 1046 pp.
- Lindstrom, F. T., D. E. Cawfield, D. F. Emer, G. J. Shott. 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 89193-8521. (Work in progress)
- Lindstrom, F. T., D. E. Cawfield, D. F. Emer, G. J. Shott, and M. E. Donahue. 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 89193-8521.
- Lindstrom, F. T., D. E. Cawfield, D. F. Emer, G. J. Shott, and M. E. Donahue. 1992b. 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 89193-8521.
- McLachlan, N. W. 1963. Complex variable theory and transform calculus. Cambridge press, Cambridge, U.K. 388pp.
- Nazaroff, W. W. 1992. Radon transport from soil to air reviews of geophysics. 30(2):137-160.

DISTRIBUTION LIST

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

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

Kenneth B. Been, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 738, Las Vegas, NV 89193. (5)

M. E. Donahue, Reynolds Electrical & Engineering Co., Inc. P.O. Box 98521, M/S 501, 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)

F. T. Lindstrom, 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)

END

DATE

FILMED

3/29/94

minutes