1 of 2

12
2-7-94 JS (1)

PNL-8943
UC-402

Rn3D: A Finite Element Code For Simulating Gas Flow and Radon Transport in Variably Saturated, Nonisothermal Porous Media: User's Manual, Version 1.0

D. J. Holford

January 1994

**Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830**

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

PNL-8943

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

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

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831;
prices available from (615) 576-8401. FTS 626-8401.

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

The contents of this report were printed on recycled paper

**Rn3D: A FINITE ELEMENT CODE FOR
SIMULATING GAS FLOW AND RADON
TRANSPORT IN VARIABLY
SATURATED, NONISOTHERMAL
POROUS MEDIA: USER'S MANUAL,
VERSION 1.0**

D. J. Holford

January 1994

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED ^{zp}

SUMMARY

This document is a user's manual for the Rn3D finite element code. Rn3D was developed to simulate gas flow and radon transport in variably saturated, nonisothermal porous media. The Rn3D model is applicable to a wide range of problems involving radon transport in soil because it can simulate either steady-state or transient flow and transport in one-, two- or three-dimensions (including radially symmetric two-dimensional problems). The porous materials may be heterogeneous and anisotropic. This manual describes all pertinent mathematics related to the governing, boundary, and constitutive equations of the model, as well as the development of the finite element equations used in the code. Instructions are given for constructing Rn3D input files and executing the code, as well as a description of all output files generated by the code. Five verification problems are given that test various aspects of code operation, complete with example input files, FORTRAN programs for the respective analytical solutions, and plots of model results. An example simulation is presented to illustrate the type of problem Rn3D is designed to solve. Finally, instructions are given on how to convert Rn3D to simulate systems other than radon, air, and water.

CONTENTS

SUMMARY	iii
1. INTRODUCTION	1
1.1 BACKGROUND	1
1.2 MODEL DESCRIPTION	2
1.3 MODEL APPLICABILITY	3
1.4 SYSTEM REQUIREMENTS	4
2. MATHEMATICS	5
2.1 GOVERNING EQUATIONS	5
2.1.1 Radon Transport	5
2.1.2 Fluid Flow	7
2.1.3 Constitutive Relationships	9
2.2 FINITE ELEMENT EQUATIONS	14
2.2.1 Radon Transport	14
2.2.2 Fluid Flow	16
2.3 BOUNDARY CONDITIONS	17
2.3.1 Boundary Conditions for the Flow and Transport Equations	17
2.3.2 Numerical Treatment of Transport Boundary Conditions	18
2.3.3 Numerical Treatment of Flow Boundary Conditions	18
3. CODE STRUCTURE	21
3.1 SUBROUTINES AND FUNCTIONS	21
3.2 INFORMATION FLOW BETWEEN SUBROUTINES	23
3.3 ARRAY DIMENSIONS	23
3.4 MACHINE-SPECIFIC FEATURES	23
4. DATA INPUT	25
4.1 INPUT DATA FILE FORMAT	25
4.2 DATA GROUPS AND VARIABLES	27
4.2.1 Data Group CASE	28

4.2.2	Data Group RUNTYPE	28
4.2.3	Data Group DIMENSION	29
4.2.4	Data Group TIME	30
4.2.5	Data Group GRADIENTS	31
4.2.6	Data Group CONVERGENCE	31
4.2.7	Data Group CRACK	33
4.2.8	Data Group GRID	33
4.2.9	Data Group PRESSURE	36
4.2.10	Data Group CONCENTRATION	37
4.2.11	Data Group BOUNDARY	37
4.2.12	Data Group ZONES	40
4.2.13	Data Group SOIL	41
4.2.14	Data Group FLUID	45
4.2.15	Data Group INITIAL	46
4.2.16	Data Group OUTPUT	47
5.	DATA OUTPUT	49
5.1	Rn3D OUTPUT FILES	49
6.	CODE EXECUTION	51
6.1	Rn3D EXECUTION ON UNIX SYSTEMS	51
7.	VERIFICATION OF NUMERICAL MODEL	53
7.1	STEADY-STATE, ONE-DIMENSIONAL ADVECTION, DIFFUSION, RADIOACTIVE DECAY, AND PRODUCTION	53
7.2	STEADY-STATE, TWO-DIMENSIONAL MATRIX DIFFUSION FROM A SINGLE CRACK	54
7.3	TRANSIENT, ONE-DIMENSIONAL ADVECTION AND DIFFUSION	57
7.4	TRANSIENT, ONE-DIMENSIONAL GAS FLOW IN RESPONSE TO SINUSOIDAL VARIATIONS IN SURFACE PRESSURE	58
7.5	STEADY-STATE RADIAL GAS FLOW FROM A WELL BENEATH A CONFINING LAYER	60

8. EXAMPLE SIMULATION	65
8.1 TWO-DIMENSIONAL CRACKED CONCRETE SLAB ABOVE GRAVEL LAYER AND SOIL	65
9. USING RN3D TO SIMULATE SYSTEMS OTHER THAN RADON, AIR AND WATER	69
10. REFERENCES	71
APPENDIX – Rn3D INPUT FILES, AND PROGRAMS TO COMPUTE ANALYTICAL SOLUTIONS FOR VERIFICATION AND EXAMPLE PROBLEMS	A.1

FIGURES

1.1. Schematic showing various factors affecting radon transport in soils	3
3.1. Flow of information between Rn3D subroutines	24
4.1. Types of finite element grids generated by Rn3D	30
6.1. Sample Rn3D execution file for a computer using the UNIX operating system	51
7.1. Comparison of numerical and analytical solutions for one-dimensional, steady-state advection, diffusion, radioactive decay and constant source	54
7.2. Domain for simulation of two-dimensional, steady-state matrix diffusion from a single crack	55
7.3. Comparison of Rn3D numerical (symbols) and analytical (solid curves) solutions for steady-state matrix diffusion from a single crack	57
7.4. Comparison of Rn3D numerical solution (circles) and an analytical solution for concentration profiles at various times for one-dimensional, transient advective-dispersion	58
7.5. Comparison of Rn3D and analytical solutions to one-dimensional, transient pressure variations in soil due to sinusoidally varying surface pressure	60
7.6. Domain separated from atmosphere by confining unit	61
7.7. Comparison of numerical and analytical solutions for axisymmetric gas flow to a well	64
8.1. Diagram of two-dimensional cracked concrete slab above gravel layer and soil	65
8.2. Average radon flux and surface pressure from example simulation 8.1	67

TABLES

2.1. Solubility of radon in water/air at atmospheric pressure	13
3.1. Rn3D subroutines and their purpose	21
4.1. Rn3D input files and their purpose	25
4.2. Data group names for Rn3D input	27
4.3. Acceptable values for the variable PLOTTT, which controls which variables will be output to ASCII plot files	48
5.1. Rn3D output files and their purpose	49
8.1. Properties of porous materials used in example simulation 8.1	66

1. INTRODUCTION

1.1 BACKGROUND

Radon transport in soil gas and groundwater has been studied for many years in connection with uranium exploration, as a tracer for moving air and groundwater, and as a possible predictor of seismic activity. There have been recent comprehensive reviews of the extensive literature on the subject [Nazaroff 1992; Nazaroff and Nero 1988]. The discovery in recent years of elevated levels of radon in the indoor environment has led to intensive investigation of the physical mechanisms which control the release and transport of radon through soils and into manmade structures. Nero [1988] has estimated that the radiation dose from inhaled decay products of radon is the major component of the natural radiation exposure of the general population. For the United States, the average lifetime risk of lung cancer caused by exposure to radon and its decay products is estimated to be 0.3%, causing an estimated 10,000 cases of lung cancer annually among the U.S. population of 235 million [Nero et al. 1986]. The primary source of indoor radon has been recognized to be radon that has migrated from soil. An understanding of the mechanisms and processes governing radon transport is essential to identifying both the populations at risk, as well as effective mitigation methods for reducing radon exposure risks.

Because radon is chemically unreactive under normal conditions, its concentration at a measurement point is a function of two primary factors: 1) the abundance and distribution of its parent radionuclide, radium, in the source material and 2) the efficiency of transport processes which bring about its release and movement from the source material. Because ^{222}Rn occurs within the decay series of ^{238}U and is the daughter of ^{226}Ra its concentration in soil gas is directly related to the abundance of these isotopes in the source material. However, there is no strict correlation between radon concentrations in soil water and soil gas and radium content of rocks and soil. In fact, good correlation exists only for extreme cases: if the uranium and radium content of the parent material is very low or very high, then the soil gas radon content is likely to be very low or very high. However, for the vast majority of materials which contain only modest amounts of radium, the radon concentration in adjacent soil water and/or soil gas can vary over many orders of magnitude [Rierner and Gunderson 1989; Sextro et al. 1987]. This illustrates that transport processes are dominant in controlling radon concentrations in most soils which contain modest amounts of radium.

The transport mechanisms responsible for the release of radon from the solid matrix and for its further movement through the porous media need to be fully understood in order to identify the significant factors affecting radon behavior in the subsurface environment. Environmental conditions, such as barometric pressure changes and rainfall events, are the major controlling factors of radon activity and flux in the soil. The interaction between environmental conditions and soil characteristics affects the structure, moisture content, and radon transport characteristics of the soil. The transport characteristics in turn affect the radon concentration and flux within and from the soil.

1.2 MODEL DESCRIPTION

The finite element model simulates pressure gradients in the soil and cracks caused by pressure changes at the ground surface. The resulting air flow drives the advective transport of radon gas, affecting the flux density of radon across the soil/air interface. The soil in the model may be heterogeneous and anisotropic. The pores in the soil may be completely dry, partially or completely filled with water. A uniform water pressure and temperature gradient may be specified in the model input. The cracks in the model are evenly spaced and of a uniform width and depth.

Molecular diffusion of ^{222}Rn is assumed to be governed by Fick's law [Crank 1975]. The advective flow of gas is assumed to be laminar and governed by Darcy's law. A source of radon-producing ^{226}Ra is distributed in the soil; the radium content distribution may be homogeneous or heterogeneous, and the actual radon emanation may be a function of water content.

Figure 1.1 is a schematic diagram of the model, showing various components affecting transport. Advection and diffusion are multidimensional, because the cracks do not extend down to the water table and pressure changes will result in a complex, nonuniform flow field.

Figure 1.1. Schematic showing various factors affecting radon transport in soils.

1.3 MODEL APPLICABILITY

The Rn3D model is applicable to a wide range of problems involving radon transport in soil. The model has been used to simulate the relationship between barometric pressure variations and radon fluxes from a dry, cracked gravelly, sandy loam at a site near Socorro, New Mexico [Holford et al. 1992b].

The Rn3D computer code simulates the transport of radon through various porous media such as soil, rock, gravel or concrete. This has enabled the performance of realistic simulations of radon transport from radium-bearing soils into buildings and to determine the soil properties that must be measured in order to predict radon entry rates into buildings [Owczarski et al. 1990]. Some of the most important parameters are gas-permeability, porosity, and water content. The water content of the soil void space greatly affects the rate of radon diffusion in soil because the

diffusion coefficient of radon in water is 5 times less than that in air. The permeability of the soil affects the response of the soil to transient variations in pressure differentials between the soil and the atmosphere or buildings.

Natural variations in meteorological factors and soil properties contribute to uncertainty in subsurface model predictions of radon flux, which, when coupled with a building transport model, will add uncertainty to predictions of radon concentrations in homes. A statistical uncertainty analysis using the Rn3D finite-element numerical model was conducted to assess the relative importance of these meteorological factors and the properties of the soil affecting the transport of radon [Holford et al. 1992a]. Important parameters to measure in the field to make more accurate predictions of radon fluxes are the presence of cracks (or other macropores), radium content, gas pressure gradient, and capillary pressure (which affects soil saturation and ultimately, the strength of radon diffusion).

The Rn3D computer code was used in a two-dimensional study of the effects of winds on radon concentration profiles beneath a slab-on-grade house with dry gravel of various thicknesses surrounding the outer surfaces of the slab [Owczarski et al. 1991; Owczarski et al. 1992]. This study confirmed that winds can effectively dilute sub-slab radon concentrations in slabs underlain by gravel. Results show that for a gravel depth of 0.1 m, radon concentrations are reduced by up to 50% with an underlying soil of sand or loam, 60% with silt, and 90% with clay. Reductions are larger for thicker gravel layers.

1.4 SYSTEM REQUIREMENTS

The Rn3D source code is written in FORTRAN 77, and can be found in plain-text format on an IBM-PC format disk attached to the inside back cover of this report. Simple one- and two-dimensional simulations have been run successfully on a Macintosh IIfx with a 240 mb hard disk and 20 mb of ram. The RAM and hard disk space requirements for complex two- and three-dimensional simulations are large, and have required the use of UNIX workstations such as the SUN Sparcstations and the IBM RS-6000.

2. MATHEMATICS

2.1 GOVERNING EQUATIONS

2.1.1 Radon Transport

The transport of gaseous components in soils partially saturated with water has been considered in a number of studies [Abriola 1989; Corapcioglu and Baehr 1987; Kinzelbach et al. 1991; Rogers et al. 1991; Weeks et al. 1982; Wood et al. 1993]. This section details the development of the equations governing the transport of radon in both the gas (air) and liquid (water) phases contained within the pore spaces of a soil. Conservation of mass results in the following continuity equation

$$\frac{\partial}{\partial t} [(1-s) n C_a + s n C_w] = - \frac{\partial F_{ai}}{\partial x_i} - \frac{\partial F_{wi}}{\partial x_i} - \lambda [(1-s) n C_a + s n C_w] + \phi_a + \phi_w \quad (2.1)$$

where subscript a = air phase

subscript w = water phase

subscript i implies summation over the three directions of the cartesian coordinate system

t = time

s = the relative saturation of the soil pore space with water

n = the total volumetric porosity of the soil

F_{ai} and F_{wi} = the total air and water phase flux densities

λ = ^{222}Rn decay coefficient ($\ln(2)$ / half-life of ^{222}Rn)

ϕ_a = production rate of radon in the air phase

ϕ_w = production rate of radon in the water phase

The total flux density in the air phase is the sum of transport by both molecular diffusion and mechanical dispersion, and also by advection with the bulk flow of the air phase

$$F_{ai} = - D_{aij} \frac{\partial C_a}{\partial x_j} + v_{ai} C_a \quad (2.2)$$

where $-D_{aij} \partial C_a / \partial x_j$ = the diffusive/dispersive flux density in the air phase, defined by Fick's first law [Bear 1979]

$v_{ai} C_a$ = the advective flux density in the air phase

D_a = diffusion/dispersion coefficient for radon in the air phase

v_{ai} = the darcy velocity of the air phase.

Similarly, the total flux in the water phase of the soil is given by

$$F_{wi} = -D_{wij} \frac{\partial C_w}{\partial x_j} + v_{wi} C_w \quad (2.2)$$

where D_w = diffusion/dispersion coefficient for radon in the water phase

v_{wi} = the darcy velocity of the water phase.

Substituting Equations (2.2) and (2.3) into (2.1), yields the general form of the governing equation for radon transport in unsaturated soil:

$$\begin{aligned} \frac{\partial}{\partial t} [(1-s)n C_a + s n C_w] &= \frac{\partial}{\partial x_i} \left(D_{aij} \frac{\partial C_a}{\partial x_j} + D_{wij} \frac{\partial C_w}{\partial x_j} \right) - \frac{\partial}{\partial x_i} (v_{ai} C_a + v_{wi} C_w) \\ &\quad - \lambda [(1-s)n C_a + s n C_w] + \phi_a + \phi_w \end{aligned} \quad (2.4)$$

This two-phase transport equation can be simplified by assuming that the concentration in the air phase is related to the concentration in the water phase by the Henry's Law distribution coefficient,

$$\kappa = \frac{C_w}{C_a} \quad (2.5)$$

where κ is the distribution coefficient between the radon concentration in the water phase and the concentration in the air phase.

Substituting κC_a for C_w in (2.4),

$$\begin{aligned} \frac{\partial}{\partial t} [(1-s+s\kappa)n C_a] &= \frac{\partial}{\partial x_i} \left[(D_{aij} + D_{wij} \kappa) \frac{\partial C_a}{\partial x_j} \right] - \frac{\partial}{\partial x_i} [(v_{ai} + v_{wi} \kappa) C_a] \\ &\quad - \lambda (1-s+s\kappa)n C_a + \phi_a + \phi_w \end{aligned} \quad (2.6)$$

The transport equation can be further simplified by lumping the source terms for each phase into the bulk source term ϕ , which is a function of water saturation. For the sake of brevity in

following discussions, we will define a lumped diffusion/dispersion tensor, $D_{ij} = D_{a ij} + D_{w ij} \kappa$, and a lumped velocity vector $v_i = v_{ai} + v_{wi} \kappa$. Substituting these variables into Equation (2.6) results in

$$\frac{\partial}{\partial t} [(1 - s + s \kappa) n C_a] = \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial C_a}{\partial x_j} \right) - \frac{\partial}{\partial x_i} (v_i C_a) - \lambda (1 - s + s \kappa) n C_a + \phi \quad (2.7)$$

Macropores (cracks and root holes) are assumed to be perfectly dry, with a porosity of 1, therefore transport in a macropore is governed by

$$\frac{\partial C_a}{\partial t} = \frac{\partial}{\partial x_i} \left(D_{a ij} \frac{\partial C_a}{\partial x_j} \right) - \frac{\partial}{\partial x_i} (v_{ai} C_a) - \lambda C_a \quad (2.8)$$

2.1.2 Fluid Flow

The radon transport equations and the fluid flow equations are coupled by Darcy's law in both the air and the water phases [Bear 1979]:

$$v_{ai} = - \frac{k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \quad (2.9)$$

$$v_{wi} = - \frac{k_{rw} k_{ij}}{\mu_w} \left(\frac{\partial P_w}{\partial x_j} + \rho_w g_j \right) \quad (2.10)$$

where v_a = darcy velocity of air

v_w = darcy velocity of water

k_{ra} = relative air permeability (a function of water saturation)

k_{rw} = relative water permeability (a function of water saturation)

k_{ij} = intrinsic permeability (a function of soil pore structure)

μ_a = dynamic viscosity of air

μ_w = dynamic viscosity of water

P_a = absolute air pressure

P_w = absolute water pressure

ρ_a = density of air

ρ_w = density of water

g = gravitational acceleration constant.

Currently, the only option for water flow is to specify a constant vertical pressure gradient, $\partial P_w / \partial z$.

For air flow, conservation of mass requires that the change of air mass stored within a unit volume of soil equal the net rate of air flow into that volume. The resulting continuity equation for air takes the form [Bear 1979]

$$-\frac{\partial(n_a \rho_a)}{\partial t} = \frac{\partial(v_{ai} \rho_a)}{\partial x_i} - \phi \rho_a \quad (2.11)$$

where $\phi \rho_a$ is the mass production rate of air per unit volume.

Combining the continuity equation and Darcy's law results in the general form of the governing equation for air flow in unsaturated soil

$$\frac{\partial(n_a \rho_a)}{\partial t} = \frac{\partial}{\partial x_i} \left[\frac{k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \rho_a \right] + \phi \rho_a \quad (2.12)$$

Applying the chain rule to the left-hand side of Equation (2.11) gives

$$\frac{\partial(n_a \rho_a)}{\partial t} = n_a \frac{\partial \rho_a}{\partial t} + \rho_a \frac{\partial n_a}{\partial t} = \rho_a (n_a \beta_a + \alpha_a) \frac{\partial P_a}{\partial t} \quad (2.13)$$

where α_a is the compressibility of the air-filled rock matrix, $\alpha_a = \partial n_a / \partial P_a$, and β_a is the compressibility of the gas, $\beta_a = (1/\rho_a) \partial \rho_a / \partial P_a$.

For an ideal gas, the equation of state is

$$\rho_a = \frac{M P_a}{R T} \quad (2.14)$$

where M = molecular weight

R = universal gas constant

T = absolute temperature.

For steady-state temperature distributions, then

$$\frac{\partial \rho_a}{\partial P_a} = \frac{M}{RT} \quad (2.15)$$

Substituting Equations (2.13) and (2.14) into (2.12), and assuming rock compressibility to be negligible, gives the governing equation for air flow in unsaturated porous media

$$\frac{n_a M}{RT} \frac{\partial P_a}{\partial t} = \frac{\partial}{\partial x_i} \left[\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \right] + \phi \rho_a \quad (2.16)$$

In the cracks, it is assumed that the air-filled porosity equals 1.0. Air flow in the cracks is assumed to be laminar Poiseuille flow between parallel plates [Huyakorn and Pinder 1983; White 1991]. The permeability distribution is parabolic with the maximum permeability at the center of the crack; the average permeability is given by

$$k_{z(\text{crack})} = \frac{w^2}{12} \quad (2.17)$$

where w is the width of the crack. The horizontal permeabilities in the crack, k_x and k_y are assigned values several orders of magnitude larger than that of the soil.

Air flow in holes is assumed to be represented by the equations governing Hagen-Poiseuille flow in a circular duct [White 1991]. The average permeability is given by

$$k_{z(\text{hole})} = \frac{r^2}{8} \quad (2.18)$$

where r is the radius of the hole.

2.1.3 Constitutive Relationships

Several of the model variables are dependent on soil pore saturation with water. The capillary pressure of partially saturated soil is defined as the difference in pressure between the water and the air phases, and is related to hydraulic head by the density of water and gravitational acceleration

$$p_c = p_a - p_w = -\psi \rho_w g \quad (2.19)$$

where p_c = capillary pressure
 ψ = hydraulic head

ρ_w = density of water

g = gravitational acceleration.

The hydraulic head is related to the water saturation of the soil by the following empirical relation [van Genuchten 1980]:

$$\frac{s - s_r}{s_w - s_r} = \left[\frac{1}{1 + (-\psi \alpha_b)^n} \right]^m \quad (2.20)$$

where s = relative water saturation ($0 \leq s \leq 1$)

s_r = residual relative water saturation

s_w = maximum relative water saturation

n = empirical constant correlated with pore size distribution

$m = 1 - 2/n$

α_b = inverse of the air-entry pressure.

The values of n , m , and α may be obtained for different soils by fitting the van Genuchten model to measured water retention data.

The permeability of the water phase as a function of saturation can be predicted using the variables n and m determined with Equation (2.19) [Mualem 1976b]:

$$k_{rw} = s^\xi \left[1 - (1 - s^{1/m})^m \right]^2 \quad (2.21)$$

where k = intrinsic permeability and ξ = exponent commonly fixed at 0.5. The permeability of the air phase is assumed to be related to the water permeability by [Burdine 1953]

$$k_{ra} = (1 - k_{rw})(1 - s)^2 \quad (2.22)$$

The bulk diffusion/dispersion coefficient represents the combined effects of molecular diffusion and hydrodynamic dispersion. The diffusion coefficient of a gaseous species in a porous medium decreases rapidly with increasing water content. Soil pores on a microscopic scale will branch, resulting in hydrodynamic dispersion of radon transported by advection. The effects of hydrodynamic dispersion and molecular diffusion are often considered to be additive [Bear 1979]. The estimation method proposed by Millington and Quirk [1961] and modified by Collin and

Rasmuson [1988] gives the best predictions of the diffusion coefficient for different materials at various moisture contents. A simplified version of their model yields very similar results (when $\tau = 0.7$) and is much easier to implement numerically

$$D_{a,ij} = D_A \tau_{ij} n (1-s)^3 + |v_{a,i}| \alpha_D \quad (2.23)$$

$$D_{w,ij} = D_W \tau_{ij} n s^3 + |v_{w,i}| \alpha_D \quad (2.24)$$

where τ_{ij} = tortuosity factor for a soil

D_A = diffusion coefficient of radon in pure air

D_W = diffusion coefficient of radon in pure water

$|v_{w,i}|$ and $|v_{a,i}|$ are the absolute Darcy velocities in the air and water phases, respectively

α_D = is the hydrodynamic dispersivity.

The tortuosity can be defined as the ratio of the straight-line distance between two points to the distance between the same two points by way of the connected pores. The tortuosity has been empirically determined to be between 0.01 and 0.66 for most soils [Bear 1979]. The tortuosity can be estimated for a soil if the geometric standard deviation of the pore sizes is known [Nielson et al. 1984].

$$\tau = \left\{ 1 + \left[\frac{3(1-n)}{2\pi} \right]^{2/3} \right\}^{-\left(1 + \frac{\sigma_g}{22}\right)} \quad (2.25)$$

where σ_g is the geometric standard deviation of the pore size distribution of a particular soil.

The diffusion of radon in pure water was calculated as a function of temperature [Bird et al. 1960; Hart 1986]:

$$D_W = 7.4 \times 10^{-8} \frac{(T+273.15) \sqrt{\phi_w M_w}}{(\mu_w) \left(M_{Rn} / \rho_{Rn} \right)^{0.6}} \quad (2.26)$$

where D_W = diffusion of radon in pure water (cm²/s)

T = temperature (°C)

ϕ = "association parameter" for water (given as 2.6)

M_w = molecular weight of water

μ_w = viscosity of water as a function of temperature (centipoise)

M_{Rn} = molecular weight of radon

ρ_{Rn} = density of radon at the normal boiling point (given as $4.4 \times 10^3 \text{ kg/m}^3$) [Herreman 1980].

The diffusion of radon in pure air was calculated as a function of temperature and pressure using Chapman-Enskog kinetic theory [Bird et al. 1960]:

$$D_A = 1.88292 \times 10^3 \frac{\sqrt{T^3 \left(\frac{1}{M_{Rn}} + \frac{1}{M_a} \right)}}{p \sigma_{AB}^2 \Omega_{DAB}} \quad (2.27)$$

where D_a = diffusion of radon in pure air (cm^2/s)

T = temperature (K)

M_{Rn} = molecular weight of radon

M_a = molecular weight of air.

p = absolute pressure (dyne/cm^2)

σ_{AB} = Lennard-Jones parameter

Ω_{DAB} = dimensionless function of the temperature and the intermolecular potential field.

The diffusion coefficients calculated in this manner show good agreement with compiled data [Hart 1986].

The radon source term ϕ_k is defined as

$$\phi = R \rho_b \lambda E \quad (2.28)$$

where R is the radium content of the soil, ρ_b is the soil bulk density, E is the radon emanation coefficient into the soil pore space [Nielson et al. 1984].

The soil bulk density can be calculated from

$$\rho_b = \rho_p * (1 - n) \quad (2.29)$$

where: ρ_p = particle density = 2.65 g/cm^3 .

The emanating fraction of radon from the soil was calculated from

$$E = \begin{cases} E_w s/s^* + E_a(1 - s/s^*) & s < s^* \\ E_w & s > s^* \end{cases} \quad (2.30)$$

where E_w and E_a = emanation coefficients at saturation and at dryness, and s^* = minimum moisture on the plateau of an emanation-versus-moisture curve [Nielson et al. 1984].

The properties of water and air are treated as functions of temperature and pressure in Rn3D. The viscosity of water, viscosity of air, and density of water as functions of temperature were implemented tabular functions for temperatures between 0 and 100°C [Weast 1982]. The density of air as a function of temperature and pressure was calculated from the ideal gas law [Weast 1982]:

$$\rho_a = \left(\frac{1.293}{1 + 0.00367 T} \right) \left(\frac{p}{1333.224 * 76} \right) \quad (2.31)$$

where ρ_a = density of dry air (kg/m³), T = temperature (°C), and p = absolute pressure (Pa).

The solubility of radon in water/air at atmospheric pressure, κ , was implemented as a tabular function of temperatures between 0 and 100°C (Table 2.1).

Table 2.1. Solubility of radon in water/air at atmospheric pressure [Clever 1979].

<u>Temperature, °C</u>	<u>Solubility</u>
0	0.507
10	0.340
20	0.250
30	0.195
37	0.167
50	0.138
75	0.114
100	0.106

2.2 FINITE ELEMENT EQUATIONS

2.2.1 Radon Transport

The governing equation for transport of radon gas, Equation (2.7), can be rewritten as

$$\frac{\partial}{\partial t} [(1-s+s\kappa) n C_a] - \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial C_a}{\partial x_j} \right) + \frac{\partial}{\partial x_i} (v_i C_a) + \lambda (1-s+s\kappa) n C_a - \phi = 0 \quad (2.32)$$

We begin by assuming a trial solution $\hat{C}$ for C of the form:

$$\hat{C}(x_i, t) = \sum_{J=1}^{nn} N_J(x_i) C_J(t) \quad (2.33)$$

where C_J = concentration at node J , $N_J(x_i)$ is the basis function associated with node J , and nn is the total number of nodes in the problem domain. Because $\hat{C}$ is only an approximation for C , substitution of $\hat{C}$ in Equation (2.32) results in some error such that Equation (2.32) $\neq 0$.

The Galerkin method requires that the weighted integral of the error over the whole solution region, R , must be equal to zero [Huyakorn and Pinder 1983]:

$$\int_R N_I \left\{ \frac{\partial}{\partial t} [(1-s+s\kappa) n \hat{C}_a] - \frac{\partial}{\partial x_i} \left(D_{ij} \frac{\partial \hat{C}_a}{\partial x_j} \right) + \frac{\partial}{\partial x_i} (v_i \hat{C}_a) + \lambda (1-s+s\kappa) n \hat{C}_a - \phi \right\} dR = 0 \quad (2.34)$$

where $I = 1, 2, \dots, nn$.

Application of Green's (divergence) theorem to the diffusive term gives

$$\begin{aligned} \int_R \left\{ (1-s+s\kappa) n \frac{\partial \hat{C}_a}{\partial t} N_I + D_{ij} \frac{\partial N_I}{\partial x_i} \frac{\partial \hat{C}_a}{\partial x_j} + \frac{\partial}{\partial x_i} (v_i \hat{C}_a) N_I + \lambda (1-s+s\kappa) n \hat{C}_a N_I - \phi N_I \right\} dR \\ - \int_B \left(D_{ij} N_I \frac{\partial \hat{C}_a}{\partial x_j} \vec{n}_i \right) dB = 0 \end{aligned} \quad (2.35)$$

where $\vec{n}_i$ is the unit normal vector to boundary B. The boundary integral term represents diffusive flux across a boundary.

Substitution of Equation (2.33) for $\hat{C}$ gives

$$\begin{aligned} \int_R \left\{ (1-s+s\kappa) n \frac{\partial C_J}{\partial t} N_I N_J + D_{ij} \frac{\partial N_I}{\partial x_i} \frac{\partial N_J}{\partial x_j} C_J + \frac{\partial}{\partial x_i} [v_i N_J] N_I C_J + \lambda (1-s+s\kappa) n C_J N_J N_I - \phi N_I \right\} dR \\ - \int_B \left(D_{ij} N_I \frac{\partial N_J}{\partial x_j} C_J \vec{n}_i \right) dB = 0 \end{aligned} \quad (2.36)$$

which can be more conveniently written in matrix form:

$$[Q] \left\{ \frac{\partial C_J}{\partial t} \right\} + [S] \{C_J\} - \{f\} = 0 \quad (2.37)$$

where:

$$Q_{IJ} = \sum_e \int_{R^e} (1-s+s\kappa) n N_I N_J dR \quad (2.38)$$

$$S_{IJ} = \sum_e \int_{R^e} \left(D_{ij} \frac{\partial N_I}{\partial x_i} \frac{\partial N_J}{\partial x_j} + \frac{\partial}{\partial x_i} [v_i N_J] N_I + \lambda (1-s+s\kappa) n N_J N_I \right) dR \quad (2.39)$$

$$f_I = \sum_e \int_{R^e} \left(\frac{\phi}{1-s+s\kappa} N_I \right) + \sum_e \int_{B_2^e} \left(D_{ij} N_I \frac{\partial N_J}{\partial x_j} C_J \vec{n}_i \right) dB \quad (2.40)$$

where R^e is the element region, B_2^e is a second-type boundary (see Section 2.3) and the summation is for all of the elements in the solution region.

To solve this system of equations numerically, we assuming the nodal values change linearly over a single time step and employ a finite-difference time-stepping scheme. This results in a set of algebraic equations as follows:

$$\frac{1}{\Delta t} [Q]^{m+\epsilon} \left(\{C\}^{m+1} - \{C\}^m \right) + \epsilon [S]^{m+1} \{C\}^{m+1} + (1-\epsilon) [S]^m \{C\}^m + \epsilon \{f\}^{m+1} + (1-\epsilon) \{f\}^m = 0 \quad (2.41)$$

where k and $k+1$ are the previous and current time levels, Δt the current time step, ϵ a time-weighting factor between 0 and 1, and the coefficient matrix $[Q]^{m+\epsilon}$ is evaluated using the weighted average of the current and previous nodal values of C .

Rearranging, to separate known and unknown quantities yields

$$\left(\frac{1}{\Delta t} [Q]^{m+\epsilon} + \epsilon [S]^{m+1} \right) \{C\}^{m+1} = \left(\frac{1}{\Delta t} [Q]^{m+\epsilon} - (1-\epsilon) [S]^m \right) \{C\}^m - \epsilon \{f\}^{m+1} - (1-\epsilon) \{f\}^m \quad (2.42)$$

Now, everything on the right-hand side and within the parentheses on the left-hand side of Equation (2.43) is known, and the unknown nodal values $\{C\}^{m+1}$ can be determined.

If $\epsilon = 0$, the concentration at the new time step is weighted toward the concentration at the beginning of the time step. The numerical procedure which results from this choice is called a forward difference or explicit method, because an explicit expression for the concentration at the new time step can be written. If $0 < \epsilon < 1$, then the solution is obtained using an average of the old and new concentrations. If $\epsilon = 0.5$, this is called a time-centered, or Crank-Nicholson scheme. If $\epsilon = 1$, the solution is weighed toward the concentration at the end of the time step. For a steady-state simulation, ϵ must equal 1.0, because there are no previous time steps. The selection of the best value of ϵ for transient simulations is determined by considerations of numerical stability and accuracy. The explicit scheme is stable only if time steps are kept small, the Crank-Nicholson and implicit schemes are less accurate but conditionally stable. The user is encouraged to experiment with different values of ϵ for transient simulations, and is referred to textbooks on numerical modeling for a more complete discussion [Huyakorn and Pinder 1983].

2.2.2 Fluid Flow

The governing equation for compressible, ideal gas flow may be rewritten as

$$\frac{n_a M}{RT} \frac{\partial P_a}{\partial t} - \frac{\partial}{\partial x_i} \left[\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \right] + \phi_{\rho_a} = 0 \quad (2.43)$$

Applying the Galerkin method and Green's theorem as in Equations (2.34) to (2.36) results in

$$\int_R \left[\frac{n_a M}{RT} \frac{\partial P_{aj}}{\partial t} N_I N_J - \frac{\rho_a k_{ra} k_{ij}}{\mu_a} P_{aj} \frac{\partial N_I}{\partial x_i} \frac{\partial N_J}{\partial x_j} - \frac{\rho_a k_{ra} k_{ij}}{\mu_a} \rho_a g_j \frac{\partial N_I}{\partial x_i} + \phi_{\rho_a} N_I \right] dR$$

$$- \int_B \left[\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \vec{n}_i N_I \right] dB = 0 \quad (2.44)$$

which can be more conveniently written in matrix form:

$$[T] \left\{ \frac{\partial C_J}{\partial t} \right\} + [V] \{C_J\} - \{g\} = 0 \quad (2.45)$$

where:

$$T_{IJ} = \sum_e \int_{R^e} \left(\frac{n_a M}{RT} N_I N_J \right) dR \quad (2.46)$$

$$V_{IJ} = \sum_e \int_{R^e} \left(- \frac{\rho_a k_{ra} k_{ij}}{\mu_a} \frac{\partial N_I}{\partial x_i} \frac{\partial N_J}{\partial x_j} \right) dR \quad (2.47)$$

$$g_I = \sum_e \int_{R^e} \left(- \frac{\rho_a k_{ra} k_{ij}}{\mu_a} \rho_a g_j \frac{\partial N_I}{\partial x_i} + \phi_{\rho_a} N_I \right) dR + \sum_e \int_{B_2^e} \left[\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \vec{n}_i N_I \right] dB \quad (2.48)$$

Applying a finite-difference technique to the time derivative similar to that used to obtain Equation (2.42) results in

$$\left(\frac{1}{\Delta t} [T]^{m+\varepsilon} + \varepsilon [V]^{m+1} \right) \{P\}^{m+1} = \left(\frac{1}{\Delta t} [T]^{m+\varepsilon} - (1-\varepsilon) [V]^m \right) \{P\}^m - \varepsilon \{g\}^{m+1} - (1-\varepsilon) \{g\}^m \quad (2.49)$$

2.3 BOUNDARY CONDITIONS

2.3.1 Boundary Conditions for the Flow and Transport Equations

The boundary conditions at the surface can vary in both space and time. First- (Dirichlet or prescribed value), second- (Neumann or prescribed flux), or third-type (Cauchy or mixed) boundary conditions may be implemented. At a first-type boundary, the value of pressure or concentration is prescribed, but may vary with time. At a second-type boundary, the value of air flux (in units of volume per unit area per unit time) or the diffusive radon flux (in units of mass per

unit area per unit time) may be prescribed. At a third-type boundary, the diffusive radon flux is prescribed as being proportional to the concentration gradient across the boundary.

2.3.2 Numerical Treatment of Transport Boundary Conditions

A first-type boundary is simulated by holding the value at the boundary node equal to the prescribed concentration value.

The boundary integral term from Equation (2.36)

$$\int_B \left(D_{ij} N_I \frac{\partial N_J}{\partial x_j} C_J \vec{n}_i \right) dB \quad (2.50)$$

represents the diffusive flux integrated across a boundary. At a second-type boundary

$$D_{ij} \frac{\partial N_J}{\partial x_j} C_J \vec{n}_i = F_{CB} \quad (2.51)$$

where F_{CB} is the prescribed diffusive flux on the boundary. At a third-type boundary

$$D_{ij} \frac{\partial N_J}{\partial x_j} C_J \vec{n}_i = q_C (C_J - C_B) \quad (2.52)$$

where q_C is a mass transfer coefficient and C_B is the prescribed concentration just outside the boundary. The mass transfer coefficient can be used to represent a film resistance at the ground surface.

The advective flux across a boundary is calculated by calculating the advective flux, Equation (2.2), at each boundary node and taking a weighted sum of the values. The total radon flux output by the model is the sum of the advective and diffusive fluxes across the soil surface.

2.3.3 Numerical Treatment of Flow Boundary Conditions

A first-type boundary is simulated by holding the value at the boundary node equal to the prescribed pressure value.

The boundary integral term from Equation (2.44)

$$\int_B \left[\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \vec{n}_i N_I \right] dB \quad (2.53)$$

represents the flux integrated across a boundary. At a second-type boundary

$$\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \vec{n}_i = F_{PB} \quad (2.54)$$

where F_{PB} is the prescribed flux on the boundary. At a third-type boundary

$$\frac{\rho_a k_{ra} k_{ij}}{\mu_a} \left(\frac{\partial P_a}{\partial x_j} + \rho_a g_j \right) \vec{n}_i = q_P (P_I - P_B) \quad (2.55)$$

where q_P is a mass transfer coefficient and P_B is the prescribed pressure just outside the boundary.

3. CODE STRUCTURE

3.1 SUBROUTINES AND FUNCTIONS

Table 3.1 describes the purpose of each FORTRAN subroutine in the Rn3D code.

Table 3.1. Rn3D subroutines and their purpose

<u>Subroutine/ Function</u>	<u>Purpose</u>
BCFLUX	Calculate flux across boundary: flux out (+), flux in (-)
BCVEL	Calculates outward normal velocity vectors at model boundary
BOUND	Read boundary condition data and write it to scratch file
COMMON	Initialize common block variables
COMPG	Calculates compressibility (partial differential of density w.r.t. pressure) of ideal gas mixture as a function of temperature, pressure, and mole fraction of each gas
DENG	Density of dry air (g/cm ³) assuming ideal gas as a function of pressure and temperature
DENL	Density (g/ml) of water as a function of temperature
DSOIL	Bulk diffusion coefficient of radon (cm ² /s) in soil as a function of liquid saturation
DIFG	Diffusion coefficient of radon in pure air (cm ² /s) as a function of temperature and pressure
DIFL	Diffusion coefficient of radon in pure water (cm ² /s) as a function of temperature
ELPROP	Assign material properties to each finite element. Read in soil properties, including porosity, bulk density, percent saturation, relative permeability, dry soil air permeability in the x, y and z-direction, radon diffusion coefficient in soil, radon emanation coefficient, and radium content. Also read in gas properties, including radon diffusion coefficient in pure air, air

	dynamic viscosity, half-life of radon, distribution coefficient of radon in water/air
FEFLUX	Calculates x, y, and z components of radon flux vector at element centers
FECRID	Generate 1, 2, or 3-dimensional regular finite element grid with linear, rectangular, or rectangular box elements, respectively; the grid spacing can increase by a constant ratio
GAUSSQ	Set up element data for gaussian quadrature
GETBC	Reads boundary conditions for current time step from scratch file
GETOUT	Prints out selected variables at each time step in format suitable for plotting
HEADER	Print header for output file, including code name and version, and date and time of run.
INDATA	Reads in problem title, type, and gradients
INITAL	Initialize time counter and arrays, either by reading initial conditions from last time step of binary file, or by using default values and performing a steady-state run for initial conditions
MAIN	Main program, controls time step loop, calls most subroutines
MATRX	Forms local matrix for each element, then assembles these into global matrix
ORDER	Feeds matrix element information to a SPARSPAK-A routine that determines matrix structure and most efficient ordering
PARAM	Defines parameters (Leonard-Jones, etc.) for calculation of gas and liquid viscosities, diffusion coefficients, and densities
RDOUT	Reads in parameters controlling program output.
RDZONE	Reads in material properties of each soil zone
RESET	SPARSPAK-A returns solution in solution matrix {s}, so here we transfer solution from {s} to concentration matrix {conc} or pressure matrix {pres}
RPERMG	Relative gas permeability as a function of liquid saturation by van Genuchten and Mualem's methods
RPERML	Relative liquid permeability as a function of liquid saturation by van Genuchten and Mualem's methods
SOLU	Solubility of radon in water/air as a function of temperature
STATSA	SPARSPAK-A subroutine that prints out matrix statistics
SOLVE6	SPARSPAK-A matrix solver
SOURC	Calculates radon source term at each element
SOURP	Calculates gas pressure source term at each element

SPARSPK	SPARSPAK-A subroutine containing machine-specific variable assignments
THETA	Liquid saturation as a function of capillary pressure by van Genuchten and Mualem's methods
VELOCI	Calculate x, y, and z-direction components of velocity vector
VISG	viscosity (poise) of air as a function of temperature
VISL	viscosity (poise) of water as a function of temperature

3.2 INFORMATION FLOW BETWEEN SUBROUTINES

Figure 3.1 is a flow chart depicting the steps in Rn3D program execution. There are four major program loops. The criteria governing the repetition of the flow, transport, and coupled loops are discussed in Section 4.2.6. The criteria governing the repetition of the time loop is discussed in Section 4.2.4.

3.3 ARRAY DIMENSIONS

Array dimensions are specified in a parameter statement in the file DIM.INC. To change the array dimensions for a specific problem, first the changes must be made to DIM.INC and the program RN3D must be recompiled. RN3D.F contains INCLUDE statements (one in each subroutine) which insert the contents of DIM.INC in to the body of the program as it is being compiled.

3.4 MACHINE-SPECIFIC FEATURES

A few aspects of Rn3D may need to be altered slightly for it to compile on different computers. These include:

1. Function to calculate program execution time (in subroutine MAIN)
2. Form of OPEN statement for input and output files (in subroutines MAIN, INITAL, RDOUT)
3. Values of variables RATIOL, RATIOS and MAXINT in subroutine SPARSPK

Figure 3.1. Flow of information between Rn3D subroutines.

4. DATA INPUT

4.1 INPUT DATA FILE FORMAT

Table 4.1 lists the various input files generated by Rn3D. The user must construct the ASCII input file, designated in the code as FORTRAN unit 4. The binary input file (unit 2) will have been generated by Rn3D during a previous code run (see Section 5.1). This enables the user to specify initial conditions that are the final conditions of a previous code run.

Table 4.1. Rn3D input files and their purpose.

File Type	Purpose and Contents
2 Binary input	concentration, gas and liquid pressures, and temperatures from previous code run; used as initial conditions if READIC = .true. (see Section 4.1.15)
5 ASCII input	echo input data, convergence data at each time step

Rn3D uses "namelist" input. Namelist input is a standard procedure for reading data into a FORTRAN program. Namelist input allows format-free input of whole groups of variables, or input of selected items in a group of variables. Several example ASCII input files for Rn3D are listed in the appendix. Referring to these example input files will make the information presented in this section easier to understand.

The first record of namelist input data has the special symbol "\$" (dollar sign) in column two or beyond, followed by the namelist group name. This is followed by a series of assignment statements, starting in or after column two, on the same or subsequent records, each assigning a value to a variable (or one or more values to array elements) of the specified group. The input data is terminated with "\$end", in or after column two, on a line following the last data line, as in the pattern:

```
$group-name
  variable=value [,variable=value,...]
$end
↑
column 2
```

A specific example is:

```
$time
    delt=900.,  nsteps = 100
$end
↑
column 2
```

where, in the data group 'time', the real variable 'delt' is assigned a value of 900 and the integer variable 'nsteps' is assigned a value of 100.

The following syntax rules apply for input data to be read by namelist:

- * The variables of the named group can be in any order, and any can be omitted.
- * The data must start in or after column two. Column one is totally ignored.
- * There must be at least one comma, space, or tab between variables, and one or more spaces or tabs are the same as a single space.
- * The equal sign of the assignment statement can have zero or more blanks or tabs on either side of it.
- * Only constant values can be used for the values assigned to variables or arrays. You cannot use a symbolic constant (PARAMETER).
 - * There must be at least one comma, space, or tab between constants, and one or more spaces or tabs are the same as a single space.
 - * An integer constant is a round number (...,-2,-1,0,1,2,...).
 - * A real constant may be input in decimal format (e.g. 1.0, .1, or 0.123) or scientific notation (e.g. 1.e10, 3.14e+02, .54e-04)
 - * A character constant is delimited by apostrophes (') or quotes ("), but if you start with one of those, you must finish that character constant with the same one. If you use the apostrophe as the delimiter, then to get an apostrophe in a string, use two consecutive apostrophes.
 - * A logical constant is any form of true or false value, such as .true. or .false., or any value beginning with .t, .f, etc.

For arrays only:

- * The form $r*c$ stores r copies of the constant c into an array, where r is a nonzero, unsigned integer constant, and c is any constant. For example, the input:

```
$boundary   value = 25* 1.0
$end
↑           ↑
column 2    no space
```

loads 1.0 into the first 25 elements of the array named 'value'.

4.2 DATA GROUPS AND VARIABLES

Input variables are read in groups. Rn3D expects to read the group names listed in table 4.2, at least once, in the order specified.

Table 4.2. Data group names for Rn3D input.

<u>Group Name</u>	<u>Number of Times Read</u>
CASE	1
RUNTYPE	1
DIMENSION	1
TIME	1
GRADIENTS	1
CONVERGENCE	1
CRACK	1
GRID	1
PRESSURE	1
BOUNDARY	PZONES ¹
CONCENTRATION	1
BOUNDARY	CZONES ²
ZONES	1
SOIL	1
FLUID	1
INITIAL	1
OUTPUT	1

(1) PZONES is a variable defined in data group PRESSURE.

(2) CZONES is a variable defined in data group CONCENTRATION.

Each group contains one or more variables, which may be required or optional input. The order in which the variables are listed within each group is not important. The variable and group names are shown in capital letters for clarity, however, in the input data file, case is not important.

4.2.1 Data Group CASE

This data group handles the input of a descriptive title, up to 80 characters long, to describe the simulation. This title will be printed in the header of the text output file.

The following input variables is included in data group CASE:

TITLE

Description:	problem title
Data Type:	character
Default Value:	blank

4.2.2 Data Group RUNTYPE

The data group RUNTYPE contains four variables: TWOPHASE, FLOW, TRANSPORT and COUPLED. Each variable has two or three possible values; these variables control general aspects of program execution.

The following variables are included in data group RUNTYPE:

TWOPHASE

Description:	determines whether simulation is one-phase (gas) or two-phase (gas and liquid)
Possible Values:	if .false. soil liquid saturation is automatically set to zero if .true. soil liquid saturation is a function of the difference between the liquid and gas phase pressures (see Section 2.1.3)
Data Type:	logical variable
Default Value:	.true.

FLOW

Description:	controls type of gas flow simulation
Possible Values:	if 'none' flow equations are not solved if 'steady' steady-state flow equations are solved

Data Type: if 'transient' transient flow equations are solved
character variable
Default Value: 'none'

TRANSPORT

Description: controls type of contaminant transport simulation
Possible Values: if 'none' transport equations are not solved
if 'steady' steady-state transport equations are solved
if 'transient' transient transport equations are solved
Data Type: character variable
Default Value: 'none'

COUPLED

Description: determines whether flow and transport equations are coupled through the gas-phase density
Possible Values: if .true. gas phase density is a function of contaminant concentration, temperature and pressure
if .false. gas phase density is a function of temperature and pressure only
Data Type: logical variable
Default Value: .false.

4.2.3 Data Group DIMENSION

The data group DIMENSION consists of the following variable:

DIM

Description: dimensionality of problem
Possible Values: 1: 1-D linear grid, z-axis
2: 2-D planar grid, x- and z-axes
2.5: 2-D radial grid, x- and z-axes
3: 3-D cubic grid, x-, y- and z-axes
Data Type: integer variable
Default Value: no default, input required

Figure 4.1 shows the four different types of grids.

When specifying grid, boundary, and material zones, for a 1-D problem only the z-direction must be considered, likewise, for a 2-D problem only the x- and z-dimensions must be considered. All other data input for a 2-D radial problem is exactly the same as for a 2-D problem; output from a 2-D radial problem will differ from a 2-D problem in that the 2-D radial grid is assumed to be axisymmetric about the z-axis. If the user wishes to construct a 1-D horizontal grid instead of a 1-D vertical grid as shown in Figure 4.1, simply construct the grid in the z-direction – but set the variable GRAV in data group FLUID equal to zero to prevent any gravity-induced flow.

Figure 4.1. Types of finite element grids generated by Rn3D.

4.2.4 Data Group TIME

The data group TIME specifies time stepping parameters for transient simulations. Either the flow or transport components of the simulation can be specified as transient in data group RUNTYPE.

The following variables are included in data group TIME:

<u>Variable</u>	<u>Description</u>	<u>Data Type</u>	<u>Default Value</u>
DELT	initial time step (seconds)	real variable	0.0
DTMULT	time step multiplication factor	real variable	1.0
DTMAX	maximum time step (seconds)	real variable	1×10^{30}
NSTEPS	number of time steps	integer variable	0
T0	initial time (seconds)	real variable	0.0

For example, the simulation time at the first time step is equal to T0; the simulation time at the second time step is equal to T0 + DELT*DTMULT. The time increment, DELT, grows by a factor of DTMULT at each time step. A value of 1.1 for DTMULT would cause the time increment to grow by 10% at each time step. The simulation continues until NSTEPS time steps have been completed.

4.2.5 Data Group GRADIENTS

The data group GRADIENTS specifies constant gradients for either liquid pressure (DPLDZ) or temperature (DTDZ) as a function of vertical depth. Values of liquid pressure or temperature are specified at the soil surface by defining values for the variables PL0 and TEMP0 in data group INITIAL. Then, for example, temperatures at other depths are calculated using the formula $TEMP(Z) = TEMP(Z=0) + DPLDZ*(-Z)$.

The following variables are included in data group GRADIENTS:

DPLDZ

Description:	constant vertical liquid pressure gradient
Units:	dyne/cm ²
Possible Values:	if +, liquid pressure increases downward if - liquid pressure decreases downward
Data Type:	real variable
Default Value:	0.

DTDZ

Description:	constant vertical temperature gradient
Units:	°C
Possible Values:	if + temperature increases downward if - temperature decreases downward
Data Type:	real variable
Default Value:	0.

4.2.6 Data Group CONVERGENCE

There are three instances where the equations being solved become nonlinear and the program will iterate to reach the correct solution. Those instances are (1) a gas flow simulation, because the

gas flow equations are nonlinear, (2) a third-type boundary is used in either the gas flow or contaminant transport simulations, because the flux across a surface depends on what the primary variable is calculated to be at that surface, and (3) if the variable COUPLED in the data group RUNTYPE is set equal to .true., because the flow and transport equations are then coupled through the gas density.

The following variables are included in data group CONVERGENCE:

ITERMAX

Description: maximum number of iterations per time step
Data Type: integer variable
Default Value: 1

EPS

Description: time weighting factor
Possible Values: 0.0: Fully Explicit
0.0 < EPS < 1.0: Crank-Nicholson
1.0: Fully Implicit
Data Type: real variable
Default Value: 1.0

TOLP

Description: convergence criteria for gas pressures
Data Type: real variable
Default Value: 1×10^{-7}

TOLC

Description: convergence criteria for concentrations
Data Type: real variable
Default Value: 1×10^{-7}

The code will iterate until the error between successive flow iterations is less than TOLP, and the error between successive transport iterations is less than TOLC or until ITERMAX iterations

have been completed. The time weighting factor EPS corresponds to ϵ used in the solution of the contaminant transport Equation (2.45) and similarly in the solution of the gas flow equation.

4.2.7 Data Group CRACK

Parallel, vertical cracks can be included in 2- and 3-D simulations by specifying values for the variables CWIDTH, CLENGTH, CDEPTH and NEFX in this data group.

The following variables are included in data group CRACK:

CWIDTH

Description: width of crack or hole in the x-direction (cm)
Units: centimeters
Data Type: real variable
Default Value: 0.

CLENGTH

Description: width of crack or hole in the y-direction (cm)
Units: centimeters
Data Type: real variable
Default Value: 0.

CDEPTH

Description: width of crack or hole in the z-direction (cm)
Units: centimeters
Data Type: real variable
Default Value: 0.

NEFX

Description: number of elements in the crack in the x-direction
Data Type: integer variable
Default Value: 0

4.2.8 Data Group GRID

The data group GRID allows definition of the finite-element grid. The only type of grid currently allowed is a rectangular one; irregular domains can be simulated by setting conductivities

and tortuosities to very low values to prevent flow and transport in certain portions of the grid – this can be accomplished with data groups ZONES and SOIL.

The following variables are included in data group GRID:

GZONES

Description: number of grid zones in each direction
Data type: real array with a dimension of 3
Default: for 1-dimensional problem, GZONES = 1,0,0
for 2-dimensional problem, GZONES = 1,0,1
for 3-dimensional problems, GZONES = 1,1,1

GXMAX

Description: maximum x-coordinate of grid zone
Units: centimeters
Data type: real array
Range: greater than zero

DX1

Description: initial x-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the x-direction
Range: greater than zero

DX2

Description: final x-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the x-direction
Range: greater than zero

DXMULT

Description: x-grid spacing multiplication factor
Data type: real array
Array size: number of grid zones in the x-direction
Range: greater than zero

GYMAX

Description: maximum y-coordinate of grid zone

Units: centimeters
Data type: real array
Array size: number of grid zones in the y-direction
Range: greater than zero

DY1

Description: initial y-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the y-direction
Range: greater than zero

DY2

Description: final y-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the y-direction
Range: greater than zero

DYMULT

Description: y-grid spacing multiplication factor
Data type: real array
Array size: number of grid zones in the y-direction
Range: greater than zero

GZMIN

Description: minimum z-coordinate of grid zone
Units: centimeters
Data type: real array
Array size: number of grid zones in the z-direction
Range: less than zero

DZ1

Description: initial z-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the z-direction
Range: greater than zero

DZ2

Description: final z-grid spacing
Units: centimeters
Data type: real array
Array size: number of grid zones in the z-direction
Range: greater than zero

DZMULT

Description: z-grid spacing multiplication factor
Data type: real array
Array size: number of grid zones in the z-direction
Range: greater than zero

The grid zones must be defined in series beginning at the origin ($x=0$, $y=0$, $z=0$). For instance, to construct a two-dimensional grid with two grid zones in the x-direction, no grid zones in the y-direction, and one grid zone in the z-direction, you would define $GZONES = 2,0,1$. To define the first grid zone in the x-direction with a grid spacing increasing from 1.0 cm to 10.0 cm, and the second grid zone in the x-direction with a grid spacing decreasing from 9.0 cm to 2.0 cm, you would define $DX1 = 1.0,9.0$ and $DX2 = 10.0,2.0$. In order for the grid spacing to increase in the first x-direction grid zone, $DXMULT$ would be given a number larger than 1, say 1.2; for the grid spacing to decrease in the second x-direction grid zone, $DXMULT$ would be given a number smaller than 1, say 0.8. In this case, you would define $DXMULT = 1.2,0.8$. If each of the x-direction grid zones were 100.0 cm wide, you would define the maximum value of each grid zone as $DXMAX = 100.0,200.0$; in this case the maximum x-coordinate is 200.0. The procedure is the same for the y- and z-directions except for one small change for the z-direction. In the z-direction, because it extends from the ground surface at zero to negative values at depth, the variable $ZMIN$ is used to define the extent of the grid zones.

4.2.9 Data Group PRESSURE

This data group is used to specify the number of different boundary condition zones for the gas flow portion of Rn3D. The data group BOUNDARY immediately follows data group PRESSURE if the value of PZONES is greater than zero. If the value of PZONES is set equal to 2, for example, then Rn3D will expect two separate BOUNDARY data groups to follow data group PRESSURE; if the value of PZONES is set equal to 0, then no BOUNDARY data groups will follow data group PRESSURE.

The following variable is included in data group **PRESSURE**:

PZONES

Description:	number of pressure boundary condition zones
Data type:	integer variable
Default value:	none, this variable must be defined
Range of values:	greater than or equal to zero

4.2.10 Data Group CONCENTRATION

This data group is used to specify the number of different boundary condition zones for the radon transport portion of Rn3D. The data group **BOUNDARY** immediately follows data group **CONCENTRATION** if the value of **CZONES** is greater than zero. If the value of **CZONES** is set equal to 2, for example, then Rn3D will expect two separate **BOUNDARY** data groups to follow data group **CONCENTRATION**; if the value of **CZONES** is set equal to 0, then no **BOUNDARY** data groups will follow data group **CONCENTRATION**.

The following variable is included in data group **CONCENTRATION**:

CZONES

Description:	number of concentration boundary condition zones
Data type:	integer variable
Default value:	none, this variable must be defined
Range of values:	greater than or equal to zero

4.2.11 Data Group BOUNDARY

This data group is used to specify the boundary conditions for the simulation. This data group may occur one or more times immediately following data groups **PRESSURE** and **CONCENTRATION**. The number of times this data group occurs depends on the values of variables defined in data groups **PRESSURE** and **CONCENTRATION**; please refer to Sections 4.2.9 and 4.2.10.

The following variables are included in data group **BOUNDARY**:

BXMIN

Description:	x-coordinate of left side of boundary condition zone
Units:	centimeters

Data type:	real variable
Range:	greater than or equal to zero
BXMAX	
Description:	x-coordinate of right side of boundary condition zone
Units:	centimeters
Data type:	real variable
Range:	greater than zero
BYMIN	
Description:	y-coordinate of front side of boundary condition zone
Units:	centimeters
Data type:	real variable
Range:	greater than or equal to zero
BYMAX	
Description:	y-coordinate of back side of boundary condition zone
Units:	centimeters
Data type:	real variable
Range:	greater than zero
BZMIN	
Description:	z-coordinate of bottom side of boundary condition zone
Units:	centimeters
Data type:	real variable
Range:	less than or equal to zero
BZMAX	
Description:	z-coordinate of top side of boundary condition zone
Units:	centimeters
Data type:	real variable
Range:	less than zero
NNODE	
Description:	node number of zone (for zones that encompass only one node); this variable is used in leu of BXMIN, BXMAX, etc., to define the extent of the boundary zone
Data type:	integer variable
Range:	greater than zero, less than the number of nodes

VALUE

Description: specified value of pressure or concentration for first or third type boundary condition

Units: pressure: dyne/cm², concentration: any units as long as CCONV is specified to convert them to moles/cm³

Data type: real array

Array size: number of time steps

Range: less than zero

XCOEF

Description: mass transfer coefficient for third type boundary condition in x-direction

Data type: real array

Array size: number of time steps

Range: greater than zero

YCOEF

Description: mass transfer coefficient for third type boundary condition in y-direction

Data type: real array

Array size: number of time steps

Range: greater than zero

ZCOEF

Description: mass transfer coefficient for third type boundary condition in z-direction

Data type: real array

Array size: number of time steps

Range: greater than zero

XFLUX

Description: specified flux for second type boundary condition in x-direction

Units: pressure: dyne/cm²/s, concentration: any units as long as CCONV is specified to convert them to moles/cm³/s

Data type: real array

Array size: number of time steps

Range: greater than zero

YFLUX

Description: specified flux for second type boundary condition in y-direction

Units: pressure: dyne/cm²/s, concentration: any units as long as CCONV is specified to convert them to moles/cm³/s

Data type: real array
 Array size: number of time steps
 Range: greater than zero

ZFLUX

Description: specified flux for second type boundary condition in z-direction
 Units: pressure: dyne/cm²/s, concentration: any units as long as CCONV is specified to convert them to moles/cm³/s
 Data type: real array
 Array size: number of time steps
 Range: greater than zero

The dimensions of the boundary zone are defined by specifying the minimum and maximum coordinates in each direction.

4.2.12 Data Group ZONES

This data group is used to define the dimensions of the material zones within the finite element grid. Each element may have only one material type.

The following variables are included in data group ZONES:

MZONES

Description: number of material zones
 Data type: integer variable

MXMIN

Description: x-coordinate of left side of material zone
 Data type: real array
 Array size: number of material zones
 Range: greater than zero

MXMAX

Description: x-coordinate of right side of material zone
 Data type: real array
 Array size: number of material zones
 Range: greater than zero

MYMIN

Description: y-coordinate of front side of material zone

Data type: real array
Array size: number of material zones
Range: greater than zero

MYMAX

Description: y-coordinate of back side of material zone
Data type: real array
Array size: number of material zones
Range: greater than zero

MZMIN

Description: z-coordinate of bottom side of material zone
Data type: real array
Array size: number of material zones
Range: greater than zero

MZMAX

Description: z-coordinate of top side of material zone
Data type: real array
Array size: number of material zones
Range: greater than zero

MATYPE

Description: material type number
Data type: integer array
Array size: number of material zones
Range: greater than zero

For the first material zone in each direction, $MXMIN = 0.$, $MYMIN = 0.$, and $MZMAX = 0.$ (remember that the z-direction coordinates range is less than or equal to zero).

4.2.13 Data Group SOIL

This data group is used to define the properties of each material zone defined in data group ZONES. These variables are defined in Section 2.1.3.

The following variables are included in data group SOIL:

MATS

Description: number of materials

Data type:	integer variable
------------	------------------

POROS

Description:	porosity of soil
Data type:	real array
Array size:	number of materials, MATS
Range:	0.0 to 1.0

DENB

Description:	dry bulk density of soil (g/cm ³)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

TORTX

Description:	tortuosity factor for soil in x-direction
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

TORTY

Description:	tortuosity factor for soil in y-direction
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

TORTZ

Description:	tortuosity factor for soil in z-direction
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

DIFX

Description:	x-direction diffusivity (cm)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

DIFY

Description:	y-direction diffusivity (cm)
Data type:	real array

Array size:	number of materials, MATS
Range:	greater than zero

DIFZ

Description:	z-direction diffusivity (cm)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

DISPX

Description:	x-direction dispersivity (cm)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

DISPY

Description:	y-direction dispersivity (cm)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

DISPZ

Description:	z-direction dispersivity (cm)
Data type:	real array
Array size:	number of materials
Range:	greater than zero

PERMX

Description:	intrinsic permeability in x-direction (cm ²)
Data type:	real array
Array size:	number of materials
Range:	greater than zero

PERMY

Description:	intrinsic permeability in y-direction (cm ²)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

PERMZ

Description:	intrinsic permeability in y-direction (cm ²)
--------------	--

Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

SATRES

Description:	residual saturation, minimum possible water saturation of soil
Data type:	real array
Array size:	number of materials, MATS
Range:	0.0 to 1.0

SATWET

Description:	maximum possible water saturation of soil
Data type:	real array
Array size:	number of materials, MATS
Range:	0.0 to 1.0

ALPHA

Description:	van Genuchten parameter α (see Section 2.1.3)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

VANN

Description:	van Genuchten parameter n (see Section 2.1.3)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

EXPO

Description:	van Genuchten parameter ξ (see Section 2.1.3)
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

EA

Description:	Rn: radon emanation coefficient from dry soil
Data type:	real array
Array size:	number of materials, MATS
Range:	greater than zero

EW

Description: Rn: radon emanation coefficient from wet soil
Data type: real array
Array size: number of materials, MATS
Range: greater than zero

SATMIN

Description: Rn: minimum saturation of emanation plateau
Data type: real array
Array size: number of materials, MATS
Range: greater than zero

SOURCE

Description: Rn: specific radium activity (concentration/g);
Other gases: source term (moles/cm³/s)
Data type: real array
Array size: number of materials, MATS
Range: greater than zero

DECAY

Description: Factor for exponential decay of source term with time (s⁻¹),
where source(t) = SOURCE * exp(DECAY*time)
Data type: real array
Array size: number of materials, MATS

4.2.14 Data Group FLUID

This data group is used to define various fluid properties that apply to the simulation in general. The following variables are included in data group FLUID:

HLIFE

Description: half-life of radon or other contaminant gas (s⁻¹)
Data type: real variable
Range: greater than zero

GRAV

Description: gravitational acceleration constant (cm/s²)
Data type: real variable

Range: greater than zero; typical value is 981. (or 0. if the effect of gravity is not included in the simulation)

CCONV

Description: multiplication factor to convert concentration values to moles/cm³

Data type: real variable

Range: greater than zero

4.2.15 Data Group INITIAL

This group of variables specifies the initial conditions for concentration, gas and liquid pressure, and temperature. The following variables are included in data group INITIAL:

READIC

Description: read initial conditions from binary file;

Data type: logical variable

Possible Values: if .true. Rn3D will read initial pressures, concentrations, and simulation time from binary file UNIT 2;
if .false. then Rn3D will start from constant values of concentration (C0), gas pressure (PG0), liquid pressure (PL0) and temperature (TEMP0)

C0

Description: initial concentration (any units may be used as long as CCONV is specified)

Data type: real variable

Range: greater than zero

PG0

Description: initial gas pressure (dyne/cm²)

Data type: real variable

Range: greater than zero

PL0

Description: initial liquid pressure (dyne/cm²)

Data type: real variable

Range: greater than zero

TEMP0

Description: initial temperature (°C)

Data type: real variable

Range: greater than zero

4.2.16 Data Group OUTPUT

This group of variables specifies which variables will be sent to ASCII output files in a format suitable for plotting. The following variables are included in data group INITIAL:

PLOTTT

Description: a list of numbers specifying which variables are to be printed
Data type: integer array
Range: see Table 4.3

PSTEPS

Description: a list of numbers specifying at which time step output is desired
Data type: integer array
Range: from 1 to the number of time steps
Restrictions: applies only to PLOTTT selections numbered 21 or higher

PNODES

Description: a list of numbers specifying which node output is desired
Data type: integer array
Range: from 1 to the number of nodes
Restrictions: applies only to PLOTTT selections numbered 12 and 13

Table 4.3. Acceptable values for the variable PLOTTT, which controls which variables will be output to ASCII plot files. The output files will be named fort.n, where n is the number listed in the value column of this table.

<u>Value</u>	<u>Output Variable</u>	<u>Restrictions</u>
11	Average Flux vs. Time	
12	Soil Gas Concentrations vs. Time	at selected nodes using PNODES
13	Soil Gas Pressures vs. Time	at selected nodes using PNODES
21	Fluxes (across model boundary) at selected times using PSTEPS	
22	Soil Gas Velocities (across model boundary)	at selected times using PSTEPS
31	Fluxes (at element centers)	at selected times using PSTEPS
32	Soil Gas Concentrations at nodes	at selected times using PSTEPS
33	Soil Gas Velocities (at element centers)	at selected times using PSTEPS
34	Soil Liquid Velocities (at element centers)	at selected times using PSTEPS
35	Soil Gas Pressures at nodes	at selected times using PSTEPS
36	Soil Liquid Pressures at nodes	at selected times using PSTEPS
37	Temperatures at nodes	at selected times using PSTEPS
38	Mole Fraction at nodes	at selected times using PSTEPS
39	Total Molar Concentration (P/RT) at nodes	at selected times using PSTEPS
41	Liquid Saturations (at element centers)	at selected times using PSTEPS
42	Gas Permeabilities (at element centers)	at selected times using PSTEPS
43	Liquid Permeabilities (at element centers)	at selected times using PSTEPS
44	Diffusion/Dispersion Coefficient (at element centers) at selected times using PSTEPS	
45	Source Term (at element centers) at selected times using PSTEPS	
46	Gas Phase Density (at element centers) at selected times using PSTEPS	
47	Gas Phase Viscosity (at element centers) at selected times using PSTEPS	

5. DATA OUTPUT

5.1 Rn3D OUTPUT FILES

Table 5.1 lists the various output files generated by Rn3D. The FORTRAN unit 1 binary file is useful as a restart file. The final conditions of a code run can be used as the initial conditions of a successive run (see Section 4.2.15).

Table 5.1. Rn3D output files and their purpose.

<u>Unit</u>	<u>File Type</u>	<u>Purpose and Contents</u>
1	Binary output	concentration, gas and liquid pressures, and temperatures at final time step
6	ASCII output	echo input data, convergence data at each time step
9	Binary scratch	temporary storage of boundary conditions
>11	ASCII output	plot files, see Section 5.2.16

The binary output file, referenced in the code as FORTRAN unit 1, receives the concentration, gas and liquid pressures, and temperatures at final time step. This file can be read in during subsequent runs of the model and the values used as initial conditions. The binary scratch file, referenced in the code as binary unit 9, is a temporary file for the storage of boundary conditions at each time. This file is used in order to reduce the size of the arrays needed by the code.

The ASCII output file, which is referenced in the code as FORTRAN unit 6, receives run-time data from the code. First, the input data is echoed as it is read in, this allows the user to determine where there are mistakes in the input file. After the input data is echoed, unit 6 then receives diagnostic messages from the matrix solver package, SPARSPAK-A, as it determines the most efficient ordering for the matrix equations. A message similar to the following will be printed:

```
IJBEGN - BEGIN STRUCTURE INPUT ...  
INIJ    - INPUT OF ADJACENCY PAIRS ...  
IJEND   - END OF STRUCTURE INPUT ...  
ORDRA6  - NESTED DISSECTION ORDERING ...
```

STATSA - SYSTEM-A STATISTICS ...

SIZE OF STORAGE ARRAY (MAXSA)	2000000
NUMBER OF EQUATIONS	8978
NUMBER OF OFF-DIAGONAL NONZEROS	70622
TIME FOR ORDERING	.000
STORAGE FOR ORDERING	57758.
TIME FOR ALLOCATION	*****
STORAGE FOR ALLOCATION	101780.
STORAGE FOR SOLUTION	632594.
TOTAL TIME REQUIRED	*****
MAXIMUM STORAGE REQUIRED	101780.

The SIZE OF STORAGE ARRAY (MAXSA) is the value given by the user for the parameter MSA in the include file dim.inc. The value given after MAXIMUM STORAGE REQUIRED must be less than the size of the storage array. More detail is given in the user's manual for SPARSPAK-A.

At each time step, the simulation time and time step size are printed out. For each iteration, diagnostic messages from the SPARSPAK-A subroutines INAIJ6, INBI and SOLVE6 are printed as the matrix is constructed and solved. Then the maximum pressure or concentration error for each iteration is printed in order to track the convergence of the solution. Finally, the average advective, diffusive and total radon fluxes across the model boundaries are printed out as an indication of the progress of the simulation.

6. CODE EXECUTION

6.1 Rn3D EXECUTION ON UNIX SYSTEMS

On computers using the UNIX operating system, it is simplest to execute the code using a file that contains all the commands necessary for execution. Figure 6.1 shows an execution file containing the commands to run the code twice. First, a steady-state simulation is executed. Then, a transient simulation is executed using the final conditions from the steady-state simulation (`clay_1d_ic.bin`) as initial conditions. The files named `fort.1`, `fort.2` and `fort.11` are files referenced in the FORTRAN code as units 1, 2 and 11, respectively. These files are given more specific names using the link command: `ln -s`. All files named `fort.*` (the * signifying any extension) are removed at the end of the run because they are redundant. These files are discussed in Sections 4 and 5. The input file for the steady-state run is named `clay_1d_ic.dat`, and the output file is named `clay_1d_ic.out`.

```
rm fort.*
rm tmp.*
# steady-state run
ln -s clay_1d_ic.bin fort.1
rn3d < clay_1d_ic.dat > clay_1d_ic.out
rm fort.1
# transient run
ln -s clay_1d.bin fort.1
ln -s clay_1d_ic.bin fort.2
ln -s clay_1d.flt fort.11
rn3d < clay_1d.dat > clay_1d.out
rm fort.*
```

Figure 6.1. Sample Rn3D execution file for a computer using the UNIX operating system.

7. VERIFICATION OF NUMERICAL MODEL

Five test cases were run to verify the accuracy of the numerical model by comparing it to analytical solutions found in the literature. The model consists of two main parts, one which simulates gas flow and the other which simulates radon transport. The first three cases verify the method used to simulate radon transport and the last two cases verify the simulation of gas flow.

7.1 STEADY-STATE, ONE-DIMENSIONAL ADVECTION, DIFFUSION, RADIOACTIVE DECAY, AND PRODUCTION

The partial differential equation governing one-dimensional, steady-state, advection, diffusion, radioactive decay and production is

$$\frac{D}{n} \frac{\partial^2 C}{\partial z^2} - \frac{v_z}{n} \frac{\partial C}{\partial z} - \lambda C + \frac{\phi}{n} = 0 \quad (7.1)$$

where C = concentration of radon (Bq/cm³)
 D = diffusion coefficient of radon in soil (cm²/s)
 v_z = average vertical darcy velocity of gas phase (cm/s)
 n = porosity of soil (cm³/cm³)
 λ = radioactive decay constant of radon (s⁻¹)
 ϕ = radon production rate (Bq/g/s)

The steady-state analytical solution with an upper boundary condition of $C = 0$ is given by [Schery et al. 1984]

$$C = \frac{\phi}{n \lambda} \left\{ 1 - \exp \left[\gamma \left(\frac{4 \lambda n}{D} \right) z \right] \right\} \quad (7.2a)$$

where

$$\gamma = \beta + (\beta^2 + 1)^{1/2} \quad (7.2b)$$

and

$$\beta = \frac{v_z}{(4 \lambda n D)^{1/2}} \quad (7.2c)$$

All model parameters were taken from [Schery et al. 1984]. A comparison of results produced by Rn3D and the analytical solution are shown in Figure 7.1. There is excellent agreement between the two solutions. The code for the analytical solution and the Rn3D input file are listed in the Appendix.

Figure 7.1. Comparison of numerical and analytical solutions for one-dimensional, steady-state advection, diffusion, radioactive decay and constant source.

7.2 STEADY-STATE, TWO-DIMENSIONAL MATRIX DIFFUSION FROM A SINGLE CRACK

The simulation of contaminant transport in a cracked porous media may be simplified by assuming that only vertical advection, dispersion and radioactive decay occurs in the crack, and that only horizontal diffusion and radioactive decay occurs in the soil matrix. The domain simulated is shown in Figure 7.2.

Figure 7.2. Domain for simulation of two-dimensional, steady-state matrix diffusion from a single crack.

The governing equation for transport in the crack is given by

$$D_z \frac{\partial^2 C}{\partial z^2} - v_z \frac{\partial C}{\partial z} - \lambda C + \frac{n D_x}{b} \frac{\partial C}{\partial x} \Big|_{x=b} = 0 \quad (7.3)$$

- where
- b = one-half the crack width
 - C = concentration of radon / concentration at top of crack
 - D_x = diffusion coefficient of radon in soil (cm²/s)
 - D_z = dispersion coefficient of radon in crack (cm²/s)
 - v_z = vertical darcy velocity of gas phase in crack (cm/s)
 - n = porosity of soil (cm³/cm³)
 - λ = radioactive decay constant of radon (s⁻¹)

The boundary conditions for (7.3) are that the concentration is specified at the top of the crack and that the lower boundary extends to infinity.

The governing equation for transport in the soil matrix is given by

$$D_x \frac{\partial^2 C}{\partial x^2} - \lambda C = 0 \quad (7.4)$$

The boundary conditions for the soil matrix are that the concentration at $x = b$ is equal to the concentration in the crack, and that the boundary at $x = B$ is an axis of symmetry with no concentration gradient across it.

A steady-state analytical solution [Sudicky and Frind 1982] was used for comparison with model results. The solution for transport in the crack is

$$C_c = C_1 \exp \left(\frac{v_z z}{2 D_z} \left\{ 1 - \left[1 + \frac{4}{\gamma} (1 + \beta) \right]^{1/2} \right\} \right) \quad (7.5)$$

where C_1 = concentration at lower crack boundary
 $\beta = n D_x^{1/2} \tanh(\sigma \lambda^{1/2}) / \lambda^{1/2} b$
 $\gamma = v_z^2 / \lambda D_z$

The analytical solution for transport in the soil matrix is

$$C_s = C_c \left\{ \frac{\cosh[D_x^{-1/2} \lambda^{1/2} (b - x)]}{\cosh(\sigma \lambda^{1/2})} \right\} \quad (7.6)$$

where $\sigma = D_x^{-1/2} (b - B)$
 B = one-half the crack spacing

A comparison of results obtained by Rn3D and by the analytical solution are shown in Figure 7.3. Close agreement can be seen for various profiles in the horizontal (x) and vertical (z) directions. The code for the analytical solution and Rn3D input and output files are listed in the Appendix.

Figure 7.3. Comparison of Rn3D numerical (symbols) and analytical (solid curves) solutions for steady-state matrix diffusion from a single crack.

7.3 TRANSIENT, ONE-DIMENSIONAL ADVECTION AND DIFFUSION

The one-dimensional form of the advective-diffusion equation in homogeneous, isotropic soil is

$$D \frac{\partial^2 C}{\partial z^2} - \frac{v_z}{n} \frac{\partial C}{\partial z} = \frac{\partial C}{\partial t} \quad (7.7)$$

where

- C = concentration of radon (Bq/cm³)
- D = diffusion coefficient of radon in soil (cm²/s)
- v_z = average vertical darcy velocity of gas phase (cm/s)
- n = porosity of soil (cm³/cm³)

The analytical solution for a semi-infinite column with one end held at constant concentration C_0 is [Ogata 1970]

$$\frac{C}{C_0} = 0.5 \left[\operatorname{erfc} \left(\frac{z - \frac{v_z}{n} t}{2 \sqrt{D t}} \right) + \exp \left(\frac{v_z z}{D} \right) \operatorname{erfc} \left(\frac{z + \frac{v_z}{n} t}{2 \sqrt{D t}} \right) \right] \quad (7.8)$$

where erfc represents the complementary error function [Abramowitz and Stegun 1964].

A comparison of results obtained by Rn3D and by the analytical solution are shown in Figure 7.4. Excellent agreement can be seen for concentration profiles at various time. The code for the analytical solution and the Rn3D input file are listed in the Appendix.

Figure 7.4. Comparison of Rn3D numerical solution (circles) and an analytical solution for concentration profiles at various times for one-dimensional, transient advective-dispersion

7.4 TRANSIENT, ONE-DIMENSIONAL GAS FLOW IN RESPONSE TO SINUSOIDAL VARIATIONS IN SURFACE PRESSURE

The governing equation for vertical gas transport in soil in response to sinusoidal variations in atmospheric pressure, when the effect of gravity is neglected, can be simplified to

$$\frac{\partial(P_a)}{\partial t} = \frac{k_z}{\mu_a n_a} P_0 \frac{\partial^2 P_a}{\partial z^2} \quad (7.9)$$

where P_0 is the average pressure, and all other variables are as defined in Chapter 2.

The transient analytical solution is given by [Carslaw and Jaeger 1959]

$$P_a(z,t) = P_0 + A(0) \exp\left(-\frac{z}{z_d}\right) \sin\left(\omega t - \frac{z}{z_d}\right) \quad (7.10)$$

where $A(0)$ is the amplitude of the pressure difference at the soil surface,
 ω is the angular frequency of the oscillation, 2π / period
and z_d is the damping depth, given by

$$z_d = \left(\frac{2 k_z P_0}{\mu_a n_a \omega} \right)^{1/2} \quad (7.11)$$

A sinusoidal pressure variation with an amplitude of 1000 dyne/cm² and a period of 24 hours was applied at the upper boundary. A comparison of results obtained by Rn3D and by the analytical solution are shown in Figure 7.5. The Rn3D numerical solution does not agree well with the analytical solution until two complete cycles of surface pressure have been completed and the initial conditions no longer affect the numerical solution. Even then, there is a slight disagreement between the solutions. This result is to be expected because of different assumptions underlying the numerical and analytical solutions. The analytical is restricted by the assumption of very small perturbations about the average pressure; the larger the surface pressure variation, the greater the disagreement will be between the numerical and analytical solution. The code for the analytical solution and input file for Rn3D are listed in the Appendix.

Figure 7.5. Comparison of Rn3D and analytical solutions to one-dimensional, transient pressure variations in soil due to sinusoidally varying surface pressure.

7.5 STEADY-STATE RADIAL GAS FLOW FROM A WELL BENEATH A CONFINING LAYER

In this test case, Rn3D is compared to a two-dimensional, steady-state analytical solution for radial flow to a gas injection/extraction well [Baehr and Hult 1991]. The domain simulated is shown in Figure 7.6.

A permeable unit is separated from the atmosphere by a lower-permeability confining unit. The bottom boundary is formed by the water table (impervious to air). The governing equation for two-dimensional steady-state radial flow is given by

$$k_r \frac{\partial^2 \phi}{\partial r^2} + k_r \frac{1}{r} \frac{\partial \phi}{\partial r} + k_z \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (7.12)$$

where $\phi = P^2$, k_r is the horizontal permeability (cm²), and k_z is the vertical permeability (cm²).

Figure 7.6. Domain separated from atmosphere by confining unit.

Leakage through the confining unit is assumed to be distributed across the domain and is accounted for by adding the term:

$$\frac{-1}{b} \left(\frac{k'}{b'} \right) (\phi - P_{atm}^2) \quad (7.13)$$

where b is the domain thickness (cm), b' is the confining unit thickness (cm), k' is the confining unit permeability (cm²), and P_{atm} is the atmospheric pressure.

This results in the governing equation for two-dimensional steady-state radial flow with leakage through a confining layer:

$$k_r \frac{\partial^2 \phi}{\partial r^2} + k_r \frac{1}{r} \frac{\partial \phi}{\partial r} + k_z \frac{\partial^2 \phi}{\partial z^2} - \frac{k'}{bb'} (\phi - P_{atm}^2) = 0 \quad (7.14)$$

The boundary conditions are as follows:

$$\frac{\partial \phi}{\partial z}(r, 0) = \frac{\partial \phi}{\partial z}(r, b) = 0 \quad (7.15)$$

$$\lim_{r \rightarrow \infty} \phi(r, z) = P_{\text{atm}}^2 \quad (7.16)$$

at $r = r_w$

$$\frac{\partial \phi}{\partial r} = 0 \quad 0 < z < d \quad (7.17)$$

$$\frac{\partial \phi}{\partial r} = \frac{-Q^*}{\pi k_r(1-d)r_w} \quad d < z < l \quad (7.18)$$

$$\frac{\partial \phi}{\partial r} = 0 \quad l < z < b \quad (7.19)$$

where r_w is the radius of the well and $Q^* = (Q\mu RT)/\omega$, where Q is the gas flow rate from the well in g/s (negative for withdrawal, positive for injection), μ is the dynamic viscosity of air in g/cm s, R is the universal gas constant = 8.314×10^{-7} g/cm² s² mol K, T is temperature in Kelvin, ω is the average molecular weight of the air phase in g/mol, d is the vertical distance from the bottom of the confining unit to the top of the well screen, and l is the vertical distance from the bottom of the confining unit to the bottom of the well screen.

In order to convert the boundary conditions at $r=r_w$ to a form that can be used with Rn3D, the chain rule is applied to the partial derivatives with respect to ϕ :

$$\frac{\partial \phi}{\partial r} = \frac{\partial P^2}{\partial r} = 2 P \frac{\partial P}{\partial r} \quad (7.20)$$

Applying this to the boundary condition at the well screen, Equation (7.15), yields

$$2 P \frac{\partial P}{\partial r} = \frac{-Q\mu RT}{\omega \pi k_r(1-d)r_w} \quad (7.21)$$

Rearranging, so that the left side of the equation is in the proper form to express a constant flux boundary condition in Rn3D gives:

$$\frac{\omega}{RT} \frac{k_r}{\mu} P \frac{\partial P}{\partial r} = \frac{-Q}{2\pi r_w(1-d)} \quad (7.22)$$

or:

$$\rho \frac{k_r}{\mu} \frac{\partial P}{\partial r} = \frac{-Q}{2\pi r_w(1-d)} \quad (7.23)$$

where $\rho = P\omega / RT$ is the average density of the air phase in g/cm³.

The analytical solution to Equations (7.11) through (7.16) is

$$\begin{aligned} \phi = & P_{atm}^2 + \frac{a Q^*}{\pi k_r r_w} \left\{ \frac{K_0(M_0 \frac{r}{a})}{b M_0 K_1(M_0 \frac{r_w}{a})} \right. \\ & \left. + \frac{2}{\pi(1-d)} \sum_{n=1}^{\infty} \frac{1}{n} \left[\frac{\sin(\frac{n\pi l}{b}) - \sin(\frac{n\pi d}{b})}{M_n K_1(M_n \frac{r_w}{a})} \right] K_0(M_n \frac{r}{a}) \cos(\frac{n\pi z}{b}) \right\} \end{aligned} \quad (7.24)$$

where $a^2 = k_r / k_z$, K_0 and K_1 are the zero-order and first-order modified Bessel functions of the second kind, respectively, and

$$M_n = \left[\left(\frac{n\pi}{b} \right)^2 + \frac{k'}{b k_z} \right]^{1/2} \quad (7.25)$$

Figure 7.7 shows a comparison of the results of the analytical solution and of Rn3D at depths of 80, 180, and 380 cm.

Figure 7.7. Comparison of numerical and analytical solutions for axisymmetric gas flow to a well.

Good agreement between the numerical and analytical solution can be seen at the various depths. The slight differences between the numerical and analytical solutions may be attributable to the fact that whereas the analytical uses constant gas viscosity and density, in Rn3D the gas viscosity and density are a function of pressure and therefore are spatially variable. The code for the analytical solution and input and output files are listed in the Appendix.

8. EXAMPLE SIMULATION

8.1 TWO-DIMENSIONAL CRACKED CONCRETE SLAB ABOVE GRAVEL LAYER AND SOIL

This simulation was constructed to demonstrate the effect of barometric pressure variations on radon transport into basements. A schematic diagram of the model is shown in Figure 8.1. The depth to water table was 10 m. Sinusoidal pressure variations with an amplitude of 50 Pa and a period of 0.5 hours were applied at the upper boundary. Actual barometric pressure variations can range from an amplitude of a few Pa for periods of minutes, to a few hundred Pa in diurnal variations. The passage of weather fronts can cause pressure changes on the order of 1000-2000 Pa [Narasimhan et al. 1990].

Figure 8.1. Diagram of two-dimensional cracked concrete slab above gravel layer and soil.

A high-permeability sandy soil was used in the simulation; the soil is overlain by a layer of gravel and a cracked concrete slab. Properties for all materials used in the simulations are listed in Table 8.1.

Table 8.1. Properties of porous materials used in example simulation 8.1.

	<u>Porosity</u> (m ³ /m ³)	<u>Permeability</u> (m ²)	<u>Diffusion</u> (m ² /s)	<u>Radium</u> <u>Content</u> (Bq/kg)	<u>Radon</u> <u>Emanation</u> <u>Coefficient</u>
sand	0.35	1.0×10 ⁻¹⁰	6.1×10 ⁻²	35.	0.1
gravel	0.40	3.0×10 ⁻⁷	6.1×10 ⁻²	0.	0.0
concrete	0.25	4.4×10 ⁻¹⁶	1.3×10 ⁻³	0.	0.0

Properties for gravel are typical of sub-slab aggregate [Gadgil et al. 1991]. Properties for concrete are typical of new residential dwellings [Rogers et al. 1991]. Porosities and permeabilities for the sand soil are typical values for the grain-size class [Freeze and Cherry 1979; Mualem 1976a]. Diffusion coefficients were estimated based on typical pore-size distributions for the grain-size class [Nielson et al. 1984] and the diffusion coefficient of radon in pure air [Bird et al. 1960; Hart 1986]. Radium content and emanation coefficients for the three soil types were based on soils near Socorro, New Mexico [Schery et al. 1984] and Poamoho, Hawaii [Cotter 1990]. Bulk densities for each material were calculated assuming $\rho_b = 2.65 * (1 - n)$. The crack width was 0.5 cm and the crack spacing in the concrete slab was 0.8 m.

The Rn3D input file for this simulation is listed in the Appendix. The results of the simulation are shown in Figure 8.2; note the inverse relationship between radon flux and barometric pressure. The time lag between barometric pressure changes and the response of the radon flux is short for sand due to its relatively high permeability, which allows pressure changes to propagate rapidly.

Figure 8.2. Average radon flux and surface pressure from example simulation 8.1.

9. USING RN3D TO SIMULATE SYSTEMS OTHER THAN RADON, AIR AND WATER

Rn3D has been written in such a way as to allow the simulation of the transport of other contaminants in porous media. The gas and liquid properties specific to radon, air and water are defined in several places: in the Rn3D subroutines PARAM, SOURC and SOURP, and in the Rn3D input data file. The variables defined in these subroutines are described specifically in Section 2.1.3. As a specific example, let us consider the conversion of Rn3D to simulate the system of oxygen, nitrogen and water.

In the subroutine PARAM, the molecular weights of Rn and Air must be replaced by the molecular weights of O₂ and N₂, as shown in the following excerpt:

```
c      Molecular weight for 1) trace gas, 2) carrier gas and 3) liquid
c O2
      molwgt(1)=31.9988
c N2
      molwgt(2)=28.0140
c H2O
      molwgt(3) = 18.016
```

Furthermore, the Lennard-Jones parameters for Rn and Air must be replaced by the Lennard-Jones parameters for O₂ and N₂:

```
c Lennard-Jones potentials for 1) trace gas and 2) carrier gas
c epslj =      characteristic energy / Boltzmann's constant (K)
c siglj =      collision diameter (angstroms)
c O2
      siglj(1)=3.467
      epslj(1)=106.7
c N2
      siglj(2)=3.798
      epslj(2)=71.4
```

Finally, the molar volume of oxygen at its normal boiling point is specified.

```
c molar volume of gas (cm**3/g-mol) at its normal boiling point
c (Reid,Prausnitz&Sherwood; Properties of Gases & Liquids, 1987, McGraw-Hill)
c O2
      vmol = 28.
```

Also, the source term for oxygen is modeled differently than the source term for radon. The source term for radon is a function of soil saturation, as shown in Equations (2.31), (2.32), and (2.33). The generation of radon in the soil is assumed to have no effect on the gas pressures, whereas for oxygen, the pressure source term (in subroutine SOURP) is linked to the oxygen source term (in subroutine SOURC) through the ideal gas law.

Finally, because oxygen does not radioactively decay as ^{222}Rn does, the half-life is set to an infinitely large number. This is accomplished using the input variable HLIFE in data group FLUID (see Section 4.21).

10. REFERENCES

- Abramowitz, M. and I. A. Stegun. 1964. *Handbook of Mathematical Functions*. Dover Publications, New York.
- Abriola, L. M. 1989. "Modeling multiphase migration of organic chemicals in groundwater systems-A review and assessment." *Environ. Health Perspec.* 83, 117-143.
- Baehr, A. L. and M. F. Hult. 1991. "Evaluation of unsaturated zone air permeability through pneumatic tests." *Water Resour. Res.* 27(10), 2605-2617.
- Bear, J. 1979. *Hydraulics of Groundwater*. McGraw-Hill, New York.
- Bird, R. B., W. E. Stewart and E. N. Lightfoot. 1960. *Transport Phenomena*. John Wiley, New York.
- Burdine, N. T. 1953. "Relative permeability calculations from pore-size distribution data." *Petr. Trans., Am. Inst. Mining Metall. Eng.* 198, 71-77.
- Carslaw, H. S. and J. C. Jaeger. 1959. *Conduction of Heat in Solids*. Oxford University Press, New York.
- Clever, H. L. 1979. *Krypton, Xenon and Radon - Gas Solubilities*. International Union of Pure and Applied Chemistry Solubility Data Series. Pergamon Press, New York.
- Collin, M. and A. Rasmuson. 1988. "A comparison of gas diffusivity models for unsaturated porous media." *Soil Sci. soc. Am. J.* 52, 1559-1565.
- Corapcioglu, M. Y. and A. L. Baehr. 1987. "A compositional multiphase model for groundwater contamination by petroleum products, 1. Theoretical considerations." *Water Resour. Res.* 23(1), 191-200.
- Cotter, J. M. 1990. *Simulated Transport of Radon in Soil Gas*. Master's Thesis, University of Hawaii, Manoa.
- Crank, J. 1975. *The Mathematics of Diffusion*. Oxford University Press, New York.
- Freeze, R. A. and J. A. Cherry. 1979. *Groundwater*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Gadgil, A. J., Y. C. Bonnefous, W. J. Fisk, R. J. Prill and A. Nematollahi. 1991. *Influence of Subslab Aggregate Permeability on SSV Performance*. LBL-31160, Lawrence Berkeley Laboratory, Berkeley, California.
- Hart, K. P. 1986. *Radon Exhalation from Uranium Tailings, Volume 1*. Ph.D. dissertation, University of New South Wales, Kensington, N.S.W.

- Herreman, W. 1980. "Calculation of the Liquid Density of Radon." *Cryogenics*. 20(3), 133-134.
- Holford, D. J., P. C. Owczarski, H. D. Freeman and G. W. Gee. 1992a. "Statistical uncertainty analysis of radon transport in nonisothermal, unsaturated soils." In *Indoor Radon and Lung Cancer: Reality or Myth? 29th Hanford Symposium on Health and Environment, October 15-19, 1990*, Part 1, Battelle Press, Richland, Washington.
- Holford, D. J., S. D. Schery, J. L. Wilson and F. M. Phillips. 1992b. "Modeling radon transport in dry, cracked soil." *J. of Geophys. Res.* 98(B1), 567-580.
- Huyakorn, P. S. and G. F. Pinder. 1983. *Computational Methods in Subsurface Flow*. Academic, San Diego, Calif.
- Kinzelbach, W. W., W. Schafer and J. Herzer. 1991. "Numerical modeling of nitrate and enhanced denitrification processes in aquifers." *Water Resour. Res.* 27, 1123-1135.
- Millington, R. J. and J. P. Quirk. 1961. "Permeability of porous solids." *Trans. Faraday Soc.* 57, 1200-1207.
- Mualem, Y. 1976a. *A Catalogue of the Hydraulic Properties of Unsaturated Soils*. Research Project 442, Technion, Israel Institute of Technology, Haifa, Israel.
- Mualem, Y. 1976b. "A new model for predicting the hydraulic conductivity of unsaturated porous media." *Water Resour. Res.* 12, 513-522.
- Narasimhan, T. N., Y. W. Tsang and H. Y. Holman. 1990. "On the potential importance of transient air flow in advective radon entry into buildings." *Geophys. Res. Lett.* 17(6), 82-824.
- Nazaroff, W. W. 1992. "Radon transport from soil to air." *Rev. Geophys.* 30, 137-160.
- Nazaroff, W. W. and A. V. Nero. 1988. *Radon and Its Decay Products in Indoor Air*. John Wiley, New York.
- Nero, A. V., M. B. Schwehr, W. W. Nazaroff and K. L. Revzan. 1986. "Distribution of airborne radon-222 concentrations in U.S. homes." *Science*. 234, 992.
- Nero, J., A. V. 1988. "Radon and its decay products in indoor air: an overview". In *Radon and its Decay Products in Indoor Air*. John Wiley and Sons, New York.
- Nielson, K. K., V. C. Rogers and G. W. Gee. 1984. "Diffusion of radon through soils: a pore distribution model." *Soil Sci. Soc. Am. J.* 48, 482-487.
- Ogata, A. 1970. *Theory of dispersion in a granular medium*. U.S. Geol. Surv. Prof. Paper 411-I, U.S. Government Printing Office, Washington, DC.
- Owczarski, P. C., D. J. Holford, K. W. Burk, H. D. Freeman and G. W. Gee. 1991. "Effects of Winds on Reducing Sub-Slab Radon Concentrations under Houses Laid Over Gravel Beds." In *Proceedings: The 1991 International Symposium on Radon and Radon Reduction Technology*, Volume 1. Symposium Oral Papers Opening Session and Technical Sessions 1 through 5. EPA-600/9/91-037a, November 1991, PB92-115351., Environmental Protection Agency, Research Triangle Park, North Carolina.

- Owczarski, P. C., D. J. Holford, H. D. Freeman and G. W. Gee. 1990. "Effects of changing water content and atmospheric pressure on radon flux from the surfaces of five soil types." *Geophys. Res. Lett.* 17(6), 817-820.
- Owczarski, P. C., D. J. Holford, H. D. Freeman and G. W. Gee. 1992. "Radon Transport from the Subsurface: The Roles of Certain Boundary Conditions at Subsurface/Environment Boundaries." In *Indoor Radon and Lung Cancer: Reality or Myth? 29th Hanford Symposium on Health and Environment, October 15-19, 1990*, Part 1, Battelle Press, Richland, Washington.
- Riemer, G. M. and L. C. S. Gunderson. 1989. "A direct correlation among indoor Rn, soil gas Rn and geology in the reading prong near Boyertown, Pennsylvania." *Health Physics*. 57(1), 155-160.
- Rogers, V. C., K. K. Nielson and M. A. Lehto. 1991. *Indoor Radon Entry from Concrete Foundations*. RAE-9127/1-1, Rogers and Assoc. Engineering Corp., Salt Lake City, Utah.
- Schery, S. D., D. H. Gaeddert and M. H. Wilkening. 1984. "Factors affecting exhalation of radon from a gravelly sandy loam." *J. Geophys. Res.* 89(D5), 7299-7309.
- Sextro, R. G., B. A. Moed, W. W. Nazaroff, K. L. Revzan and A. V. Nero. 1987. "Investigations of soil as a source of indoor radon". In *Radon and its Decay Products*. American Chemical Society, Washington, DC.
- Sudicky, E. A. and E. O. Frind. 1982. "Contaminant transport in fractured porous media: Analytical solutions for a system of parallel fractures." *Water Resour. Res.* 18(6), 1634-1642.
- van Genuchten, M. T. 1980. "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Sci. Soc. Am. J.* 44, 892-898.
- Weast, R. C. 1982. *CRC Handbook of Chemistry and Physics*. CRC Press, Inc., Boca Raton, Florida.
- Weeks, E. P., D. E. Earp and G. M. Thompson. 1982. "Use of atmospheric fluorocarbons F-11 and F-12 to determine the diffusion parameters of the unsaturated zone in the southern high-plains of Texas." *Water Resources Research*. 18(5), 1365-1378.
- White, F. M. 1991. *Viscous Fluid Flow*. McGraw-Hill, New York.
- Wood, B. D., C. K. Keller and D. L. Johnstone. 1993. "Carbon dioxide as a measure of microbial activity in the unsaturated zone." *Water Resources Research*. 29(3), 647-659.

APPENDIX

Rn3D INPUT FILES, AND PROGRAMS TO COMPUTE ANALYTICAL SOLUTIONS FOR VERIFICATION AND EXAMPLE PROBLEMS

APPENDIX

Rn3D INPUT FILES, AND PROGRAMS TO COMPUTE ANALYTICAL SOLUTIONS FOR VERIFICATION AND EXAMPLE PROBLEMS

A.1 Rn3D INPUT FILE FOR VERIFICATION PROBLEM 7.1

```
$case
  title = ' Steady-state One-dimensional Problem'
$end
$runtype
  twophase = .false.,
  flow = 'Steady',
  transport = 'Steady',
  coupled = .false.
$end
$dimension
  dim = 1.
$end
$time
  nsteps = 1, t0 = 0
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermax = 3, eps = 1.0, tolp = 1.0e-07, tolc = 1.0e-07
$end
$crack
$end
$grid
  gzones = 1,
  gzmin = -3000., dz1 = -0.1, dz2 = -20., dzmult = 1.2
$end
$pressure
  pzones = 1
$end
$boundary
  type = 1,
  value =
  2*1.0000000E+06
$end
$concentration
  czones = 1
```

```

$end
$boundary
  type = 1, value = 2*0.
$end
$zones
  mzones = 1,
  mzmax = 0.,
  mzmin = -3000.,
  matype = 1
$end
$soil
  mats = 1,
  poros = 0.35,
  denb = 1.5,
  tortz = 2.57142857e-1,
  permz = 1.1e-8,
  ea = .38,
  ew = .38,
  satmin = .4,
  source = 0.90
$end
$fluid
  hlfe = 3.3e+5, grav=981., cconv=2.9981e-20
$end
$initial
  readic = .false., c0 = 0., pl0 = 1.000000 pg0 = 1.0e+6, temp0 = 20.
$end
$output
  plotit = 32,33,35,44,
  psteps = 1
$end

```

A.2 FORTRAN PROGRAM TO COMPUTE ANALYTICAL SOLUTION FOR VERIFICATION PROBLEM 7.1

c Analytical solution for steady-state radon concentration profile
 c and surface flux. Includes advection, diffusion, radioactive
 c decay, and constant source. Taken from :
 c Schery, S.D., D.H. Gaeddert, and M.H. Wilkening,
 c "Factors affecting exhalation of radon from a gravelly sandy loam",
 c J. Geophysical Research, vol. 89, no. D5, pp 7299-7309, 1984.

```

      real sat,dist,diff,dpdz,cond,vel,por,decay,eman,radium,bulkdens,
      &source,cdepth,flux,z,c

```

```

      nnodes=101

```

```

delz=-0.1
dzmult=1.2
dzmax=-20.
sat=0.
dist=.26
tort=0.09/0.35
por=.35
dair=0.105962733
diff=tort*por*dair
dpdz=0.
cond=0.
vis = 1.8e-4
decay=2.1e-6
eman=.38
radium=.9
bulkdens=1.5
source=radium*eman*bulkdens*decay/por/(1-sat+sat*dist)
cdepth=source/decay
vel=dpdz*cond/vis/por/(1-sat+sat*dist)

write(6,*)' number of nodes = ',nnodes
write(6,*)' initial grid spacing = ',delz
write(6,*)' grid spacing multiplication factor = ',dzmult
write(6,*)' percent moisture saturation = ',sat
write(6,*)' Henry's Law distribution coefficient = ',dist
write(6,*)' Total Porosity = ',por
write(6,*)' Tortuosity = ',tort,' Diffusion in Air = ',dair
write(6,*)' Effective Diffusion Coefficient = ',diff
write(6,*)' Vertical Pressure gradient = ',dpdz
write(6,*)' Gas Permeability = ',cond
write(6,*)' Gas Viscosity = ',vis
write(6,*)' Radioactive Decay Constant = ',decay
write(6,*)' Emanating Fraction = ',eman
write(6,*)' Radium Content = ',radium
write(6,*)' Bulk Density = ',bulkdens
write(6,*)' Source Term = ',source
write(6,*)' Velocity = ',vel
write(6,*)

beta=vel/sqrt(4*decay*por*diff)
gamma=beta+sqrt(beta**2+1.)
flux=gamma*source/decay*sqrt(decay*por*diff)
write(6,*) 'flux(z=0) = ',flux
write(6,*)

write(6,*)'Depth (cm), Concentration (pCi/cm\u00b3\n)'

```

```

z=0.
do i=1,nnodes
c = source/decay * (1-exp(gamma*sqrt(decay*por/diff)*z))
write(6,*) z,',',c
z=z+delz
delz=delz*dzmult
delz=max(delz,dzmax)
enddo

stop
end

```

A.3. Rn3D INPUT FILE FOR VERIFICATION PROBLEM 7.2

```

$case
  title = ' Two-dimensional Problem with Crack'
$end
$runtype
  flow = 'Steady',
  transport = 'Steady',
  coupled = .false.,
  twophase = .false.
$end
$dimension
  dim = 2.
$end
$time
  nsteps = 1, t0 = 0.0
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermax = 10, eps = 1.0, tolp = 1.0d-08, tolc = 1.0d-07
$end
$crack
  cdepth = -2000., cwidth = .1, nefx=5
$end
$grid
  gzones = 1,
  gxmax = 400., dx1 = 0.01, dx2 = 20., dxmult = 2.5,
  gzmin = -2000., dz1 = 0.01, dz2 = 20., dzmult = 1.5
$end
$pressure
  pzones = 2
$end

```

```

$boundary
  type = 1, bxmin=0., bxmax=400., bzmin=0., bzmax=0.,
  value = 1.000200d+06
$end
$boundary
  type = 1, bxmin=0., bxmax=400., bzmin=-2000., bzmax=-2000.,
  value = 1.000000d+06
$end
$concentration
  czones = 2
$end
$boundary
  type = 1, bxmin=0., bxmax=0.05, bzmin=0., bzmax=0., value = 1.
$end
$boundary
  type = 1, bxmin=0., bxmax=0.05, bzmin=-2000., bzmax=-2000., value = 0.
$end
$zones
  mzones = 2,
  mxmin = 0.,0.,
  mxmax = 400.,0.05,
  mzmax = 0.,0.,
  mzmin = -2000.,-2000.,
  matype = 1,2
$end
$soil
  mats = 2,
  poros = 0.5,1.0,
  tortx = 0.3,1.d-20,
  tortz = 1.d-9,1.d-20,
  dispx = 0.,0.,
  dispz = 0.,0.1,
  permx = 1.d-20,1.d-20,
  permy = 1.d-20,1.d-20,
  permz = 1.d-20,1.d-20,
  source = 0.,0.
$end
$fluid
  hlife = 3.3d+5, grav=0., cconv=1.6611E-24
$end
$initial
  readic = .false., c0 = 0., pl0 = 1.00d+6, pg0 = 1.0d+6, temp0 = 20.
$end
$output
  plotit = 32,
  psteps = 1
$end

```

A.4. PROGRAM TO COMPUTE ANALYTICAL SOLUTION FOR VERIFICATION PROBLEM 7.2

c Analytical solution for matrix diffusion into a crack, from:
c Sudicky, E. A. and E. O. Frind, "Contaminant transport in fractured
c porous media: Analytical solutions for a system of parallel
c fractures", Water Resour. Res., 18, 1634-1642, 1982.

```
implicit none
integer mx,mz
parameter (mx=100,mz=100)
double precision difx,difz,velz,por,lambda,cwidth,cspace,
&delx,delz,gamma,beta,conc1,conc(mx,mz),x(mx),z(mz),
&dxmult,dxmax,dzmult,dzmax,sigma,dpdz,kz,visc
integer i,j,nx,nz,nefx

open(unit=16,file="sudicky.anal",form='formatted')
open(unit=18,file="sudicky.x",form='formatted')
open(unit=19,file="sudicky.z",form='formatted')

cwidth=1.d-1
cspace=800.
nefx=5
delx=cwidth/(nefx*2)
dxmult=2.5d0
dxmax=2.d+1
delz=1.d-2
dzmult=1.5d0
dzmax=2.d+1
nx=33
nz=68
write(16,*)' Crack Width = ',cwidth,' Crack Spacing = ',cspace

visc = 1.814205152257397d-004
dpdz = 1.d-1
kz = cwidth**2 / 12.
velz = dpdz*kz/visc

difx = .3178871E-01
difz = 0.1 * velz
por = 0.5
lambda = dlog(2.d0) / (3.3d+5)
conc1=1.0d0
write(16,*)' Velocity = ',velz
```



```

write(16,*)' Matrix Diffusion = ',difx,' Crack Dispersion = ',difz
write(16,*)' Porosity = ',por,' Radioactive Decay = ',lambda
write(16,*)' Upper B.C. Conc. = ',conc1

gamma = (velz)**2 / (lambda * difz)
sigma = difx**(-0.5) * (cspace-cwidth)/2.
beta = por * dsqrt(difx) * dtanh(sigma*dsqrt(lambda))
&      / (dsqrt(lambda) * cwidth/2.)
write(16,*)' gamma = ',gamma,' sigma = ',sigma,' beta = ',beta

z(1)=0.
do i=2,nz
  z(i)=z(i-1)+delz
delz=delz*dzmult
delz=dmin1(delz,dzmax)
end do
x(1)=0.
x(2)=cwidth/2.
do i=3,nfx+2
  x(i)=x(i-1)+delx
end do
do i=nfx+3,nx
delx=delx*dxmult
delx=dmin1(delx,dxmax)
  x(i)=x(i-1)+delx
end do
write(16,*)' Initial X Spacing = ',delx,' Max X Spacing = ',dxmax
write(16,*)' Initial Z Spacing = ',delz,' Max Z Spacing = ',dzmax

c calculate concentration in crack

do j=1,nz
  conc(1,j) = conc1 * dexp( velz*z(j)/(2.*difz)
&      * (1. - dsqrt(1. + 4./gamma * (1. + beta))) )
end do

c calculate concentration in soil

do j=1,nz
do i=2,nx
  conc(i,j) = conc(1,j) * (dcosh(difx**(-0.5) * dsqrt(lambda)
&      * (cspace/2. - x(i))) / dcosh(sigma*dsqrt(lambda)))
end do
end do

```

c print out results

```
write(16,*) nz,nx
write(16,*) 0.,0.
write(16,*) (z(j),j=1,nz)
write(16,*) (x(i),i=1,nx)
do j=1,nz
  write(16,*) (conc(i,j),i=1,nx)
end do

write(18,*)' Z (cm),',( 'X = ',x(i),',',i=1,nx)
do j=1,nz
  write(18,*) -z(j),',',(conc(i,j),',',i=1,nx)
end do
write(19,*)' X (cm),',( 'Z = ',-z(j),',',j=1,nz)
do i=1,nx
  write(19,*) x(i),',',(conc(i,j),',',j=1,nz)
end do
```

```
8001 format(6(e15.7,','))
```

```
stop
end
```

A.5. Rn3D INPUT FILE FOR VERIFICATION PROBLEM 7.3

```
$case
  title = ' Transient One-dimensional Advection-Diffusion'
$end
$runtype
  twophase = .false.,
  flow = 'Steady',
  transport = 'Transient',
  coupled = .false.
$end
$dimension
  dim = 1.
$end
$time
  delt = 3600., dtmult = 1.0, nsteps = 300, t0 = 0.
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermax = 3, eps = 1.0, tol p = 1.0e-07, tol c = 1.0e-07
```

```

$end
$crack
$end
$grid
  gzones = 1,
  gzmin = -3000., dz1 = -5., dz2 = -5., dzmult = 1.
$end
$pressure
  pzones = 2
$end
$boundary
  type = 1,
  bzmin = 0., bzmax = 0.,
  value = 301*1.010000E+06
$end
$boundary
  type = 1,
  bzmin = -3000., bzmax = -3000.,
  value = 301*1.000000E+06
$end
$concentration
  czones = 1
$end
$boundary
  type = 1,
  bzmin = 0., bzmax = 0.,
  value = 301*1.
$end
$zones
  mzones = 1,
  mzmax = 0.,
  mzmin = -3000.,
  matype = 1
$end
$soil
  mats = 1,
  poros = 0.35,
  denb = 1.5,
  tortz = 2.57142857e-1,
  permz = 1.1e-8,
  source = 0.90
$end
$fluid
  hlife = 1.0e+30, grav=0., cconv=2.9981e-20
$end
$initial

```

```

    readic = .false., c0 = 0., pl0 = 1.00e+6, pg0 = 1.0e+6, temp0 = 20.
$end
$output
    plotit = 32,33,35,44,
    psteps = 100,200,300
$end

```

A.6. PROGRAM TO COMPUTE ANALYTICAL SOLUTION FOR VERIFICATION PROBLEM 7.3

- c Analytical solution for one-dimensional, steady-state advective-diffusion
- c Ogata, A., Theory of dispersion in a granular medium,
- c U.S. Geol. Surv. Prof. Paper 411-I, 1970.

```

    open(unit=16,file="avedif.output",form='formatted')

    tort=0.09/0.35
    por=.35
    dair=0.105962733
    d=tort*dair
    write(16,*) ' Diffusion Coefficient = ',d,' cm2/sec'

    dpdz=0.010000e6/3000.
    permz = 1.1e-8
    visc = 1.8142e-4
    v=dpdz*permz/visc
    write(16,*) ' Velocity = ',v,' cm/sec'

    c0=1.
    c1=0.

    delt=3600.
    disp=0.5*delt*v**2
    write(16,*) ' Numerical Dispersion = ',disp,' cm2/sec'
    nsteps=300

    t=nsteps*delt
    write(16,*) ' Distance = ',v/por*t

    delz=5.
    zmax=3000.
    do z=0.,zmax,delz
        write(16,*)t,',',-z,',',conc(z,t,v,por,d,c0,c1)
    end do

    stop
end

```

```

function conc(x,t,v,por,d,c0,c1)
conc = c1 + 0.5*(c0-c1) *
& ( erfcc((x-v/por*t)/(2.*sqrt(d*t))) + exp(x*v/(d*por)) *
&   erfcc((x+v/por*t)/(2.*sqrt(d*t))) )
return
end

```

```

function erfcc(x)
c Returns the complementary error function erfc(x) with fractional error
c everywhere less than 1.2e-7. From "Numerical Recipes" by Press,
c Flannery, Teukolsky & Vetterling, Cambridge University Press, 1986.

```

```

z=abs(x)
t=1./(1.+0.5*z)
erfcc=t*exp(-z*z-1.26551223+t*(1.00002368+t*(.37409196+
&   t*(.09678418+t*(-.18628806+t*(.27886807+t*(-1.13520398+
&   t*(1.48851587+t*(-.82215223+t*.17087277)))))))))
if (x.lt.0.) erfcc=2.-erfcc
return
end

```

A.7. Rn3D INPUT FILE FOR VERIFICATION PROBLEM 7.4

```

$case
  title = ' Transient Gas Flow with Sinusoidally Varying Atmospheric Pressure'
$end
$runtype
  twophase = .false.,
  flow = 'Transient',
  transport = 'None',
  coupled = .false.
$end
$dimension
  dim = 1.
$end
$time
  delT = 3600., dtmult = 1.0, nsteps = 76, t0 = 0.
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermx = 5, eps = 1.0, tolp = 1.0e-08, tolc = 1.0e-07
$end
$crack
$end
$grid

```

```

gzones = 1,
gzmin = -3000., dz1 = -5., dz2 = -5., dzmult = 1.
$end
$pressure
pzones = 1
$end
$boundary
type = 1,
bzmin = 0., bzmax = 0.,
value = 1.0002588E+06, 1.0005000E+06,
1.0007071E+06, 1.0008660E+06, 1.0009659E+06,
1.0010000E+06, 1.0009659E+06, 1.0008660E+06,
1.0007071E+06, 1.0005000E+06, 1.0002588E+06,
1.0000000E+06, 9.9974118E+05, 9.9950000E+05,
9.9929289E+05, 9.9913397E+05, 9.9903407E+05,
9.9900000E+05, 9.9903407E+05, 9.9913397E+05,
9.9929289E+05, 9.9950000E+05, 9.9974118E+05,
1.0000000E+06, 1.0002588E+06, 1.0005000E+06,
1.0007071E+06, 1.0008660E+06, 1.0009659E+06,
1.0010000E+06, 1.0009659E+06, 1.0008660E+06,
1.0007071E+06, 1.0005000E+06, 1.0002588E+06,
1.0000000E+06, 9.9974118E+05, 9.9950000E+05,
9.9929289E+05, 9.9913397E+05, 9.9903407E+05,
9.9900000E+05, 9.9903407E+05, 9.9913397E+05,
9.9929289E+05, 9.9950000E+05, 9.9974118E+05,
1.0000000E+06, 1.0002588E+06, 1.0005000E+06,
1.0007071E+06, 1.0008660E+06,
$end
$concentration
czones = 0
$end
$zones
mzones = 1,
mzmax = 0.,
mzmin = -3000.,
matype = 1
$end
$soil

```

```

mats = 1,
poros = 0.35,
denb = 1.5,
tortz = 2.57142857e-1,
permz = 1.e-10,
source = 0.90
$end
$fluid
hlife = 1.0e+30, grav=0., cconv=2.9981e-20
$end
$initial
readic = .false., c0 = 0., pl0 = 1.00e+6, pg0 = 1.0e+6, temp0 = 20.
$end
$output
plotit = 35,
psteps = 4,28,52,76
$end

```

A.8. PROGRAM TO COMPUTE ANALYTICAL SOLUTION FOR VERIFICATION PROBLEM 7.4

```

c  FORTRAN program for computing gas pressures in soil due to sinusoidally
c  varying atmospheric pressure
c  Based on program 4.1 from G.S. Campbell, Soil Physics with BASIC
c  Written by Diana J. Holford, 9/24/92

```

```

c  ne      = number of elements
c  dt      = time step (sec)
c  tmax     = maximum simulation time (sec)
c  dz      = grid spacing (cm)
c  dzmult   = grid spacing multiplication factor
c  p0       = average pressure
c  am       = amplitude of sine wave temperature variation at surface
c  pe       = period of sine wave temperature variation at surface
c  om       = angular frequency (1/s)
c  z        = depth
c  pa       = gas pressure
c  kz       = gas permeability
c  visc     = gas viscosity
c  por      = gas-filled porosity
c  kon      = conductance
c  zd       = damping depth

```

```

implicit none
integer me
parameter(me=500)
double precision pa(me),z(me),p0,am,ti,dt,zd,pi,kon,visc,kz,por,
&om,dz,delz,tmax,dzmult,pe
integer ne,i
data pi /3.141592654/

```

```

open(16,file='sine.out',status='unknown',form='formatted')

```

```

dt=1.44d+4
tmax=8.64d+4
dz=5.d+0
dzmult=1.d+0
ne=200
write(16,*)' Time Step = ',dt,' Maximum Time = ',tmax
write(16,*)' Grid Spacing = ',dz,' Mult. Factor = ',dzmult
write(16,*)' Number of elements: ',ne

```

```

p0=1.e6
am=1.d+3
pe=8.64d+4
om=2*pi/pe
write(16,*)' Average Pressure = ',p0,' Amplitude = ',am
write(16,*)' Period = ',pe,' Angular Frequency = ',om

```

```

kz=1.d-10
visc=1.814205152257397d-004
por=3.5d-1
kon=kz*p0/(visc*por)
zd=dsqrt(2.* kon / om)
write(16,*)' Gas Permeability = ',kz,' Gas Viscosity = ',visc
write(16,*)' Porosity = ',por,' Damping depth = ',zd

```

```

c set grid spacing
z(1)=0.d0
delz=dz
do i=2,ne
    delz=dz*dzmult
    z(i)=z(i-1)+delz
end do

```

```

c time loop
ti=0.d0

```



```

do while (ti .le. tmax)
  ti=ti+dt

c  calculate analytical solution
  do i=1,ne
    pa(i)= p0 + am * dexp(-z(i)/zd) *
    &      dsin( om*ti - z(i)/zd )
  end do

  write(16,*)
  write(16,*)' Time (sec) = ',ti
  write(16,*)'Depth (cm), Pressure (dynes/cm\u2\n)'
  do i=1,ne
    write(16,1000) z(i),pa(i)
  end do

c  end time loop
end do

1000 format(2(1pe15.7,','))
end

```

A.9. Rn3D INPUT FILE FOR VERIFICATION PROBLEM 7.5

```

$case
  title = ' Baehr and Hult Application 1, Confining Unit'
$end
$runtype
  flow = 'Steady',
  transport = 'None',
  coupled = .false.,
  twophase = .false.
$end
$dimension
  dim = 2.5
$end
$time
  nsteps = 1, t0 = 0.
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermax = 100, eps = 1.0, tolp = 1.0d-08, tolc = 1.0d-07
$end
$crack

```

2 of 2

```

cwidth = 0., cdepth = 0., clength = 0.
$end
$grid
gzones = 3,
gxmin = 11.5, 300.,
gxmax = 300., 3000.,
dx1 = 0.1, 5.,
dx2 = 5., 50.,
dxmult = 1.2, 1.5,
gzmin = -50., -110., -180., -380., -547.,
dz1 = 0.01, 5., 5., 20., 20.,
dz2 = 5., 5., 20., 20., 50.,
dzmult = 1.3, 1.0, 1.3, 1.0, 1.2
$end
$pressure
pzones = 2
$end
$boundary
type = 1, bxmin=11.5, bxmax=3000., bzmin=0., bzmax=0.,
value=1.000000d+6
$end
$boundary
type = 2, bxmin=11.5, bxmax=11.5, bzmin=-110., bzmax=-50.,
xflux = -1.2d-2
$end
$concentration
czones = 0
$end
$boundary
$end
$zones
mzones = 2,
mxmin = 11.5, 11.5,
mxmax = 3000., 3000.,
mzmax = 0., -20.,
mzmin = -20., -547.,
matype = 1, 2
$end
$soil
mats = 2,
poros = 0.5, 0.5,
denb = 1.7, 1.7,
tortx = 0.6, 0.6,
tortz = 0.6, 0.6,
permx = 1.d-15, 3.2d-7,
permz = 1.9d-9, 1.2d-7,

```

```

source = 0., 0.
$end
$fluid
hlife = 3.3d+5, grav=981., cconv=1.d-25
$end
$initial
readic = .false., c0 = 0., pl0 = 9.00d+5, pg0 = 1.0d+6, temp0 = 9.1
$end
$output
plotit = 22,35,
psteps = 1
$end

```

A.10. PROGRAM TO COMPUTE ANALYTICAL SOLUTION FOR VERIFICATION PROBLEM 7.5

c Analytical solution for 2-D radial compressible gas flow to a well
c where the domain is separated from the atmosphere by a confining unit
c Baehr & Hult, "Evaluation of Unsaturated Zone Air Permeability
c Through Pneumatic Tests", WRR, vol. 27, No. 10, pp. 2605-2617.

```

implicit none
real pi,rg,kz,kr,b,kconf,bconf,p,patm,l,d,rw,q,temp,omega,visc,
&r0,rmax,dr,z0,zmax,dz,r,z,qstar,sum,mn,m0,bessk1,bessk0,a
integer n
data pi / 3.141592654 /
C-----
c Input Data
C-----
c universal gas constant (g cm2 / s2 mol K)
rg = 8.314e7
c atmospheric pressure (dynes/cm2)
c patm = 1.0132E+06
patm = 1.0000E+06
c mass flux of air (g/s) (- for withdrawal, + for injection)
q = 52.9
c average temperature (degrees Kelvin)
temp = 273.15 + 9.1
c average molecular weight of the air phase (g/mole)
omega = 28.8
c viscosity of the air phase (g/cm-s)
visc = 1.73e-4
c thickness of domain (cm)
b = 547.
c thickness of confining unit (cm)
bconf = 20.

```

```

c air permeability of confining unit (cm2)
  kconf = 1.9e-9
c air permeability of domain (cm2)
  kr = 3.2e-7
  kz = kr/2.6
c depth from top of domain to bottom of screen (cm)
  l = 110.
c depth from top of domain to top of screen (cm)
  d = 50.
c radius of well (cm)
  rw = 11.5
c grid dimensions (cm)
  r0 = rw
  rmax = 300.
  dr = 15.
  z0 = 80.
  zmax = 380.
  dz = 100.

c-----
  qstar = (q*visc*rg*temp)/omega
  a = sqrt (kr/kz)

  do z=z0,zmax,dz
  do r=r0,rmax,dr

    sum = 0.
    do n=1,200
      mn = sqrt( (n*pi/b)**2 + kconf/(b*bconf*kz) )
      sum = sum + (1./float(n)) * ( (sin(n*pi*l/b) - sin(n*pi*d/b)) /
& (mn*bessk1(mn*rw/a)) ) * bessk0(mn*r/a) *
& cos(n*pi*z/b)
    end do

    m0 = sqrt( kconf/(b*bconf*kz) )
    p = sqrt( patm**2 + a * qstar / (pi * kr * rw)
& * ( bessk0(m0*r/a) / (b*m0*bessk1(m0*rw/a))
& + 2./(pi*(l-d))*sum ) )

    write(6,*) r,z,p

  end do
end do

stop
end

```

```

real function bessj0(x)
c Returns the modified Bessel function Io(x) for any real x.
c From "Numerical Recipes" by Press, Flannery, Teukolsky & Vetterling,
c Cambridge University Press, 1986. Based on polynomial coefficients
c given by Abramowitz and Stegun.
  real*8 y,p1,p2,p3,p4,p5,p6,p7,q1,q2,q3,q4,q5,q6,q7,q8,q9
  data p1,p2,p3,p4,p5,p6,p7 / 1.0d0,3.5156229d0,3.0899424d0,
&1.2067492d0,0.2659732d0,0.360768d-1,0.45813d-2 /
  data q1,q2,q3,q4,q5,q6,q7,q8,q9 / 0.39894228d0,0.1328592d-1,
&0.225319d-2,-0.157565d-2,0.916281d-2,-0.2057706d-1,0.2635537d-1,
&-0.1647633d-1,0.392377d-2 /
  if (abs(x).lt.3.75) then
    y=(x/3.75)**2
    bessj0=p1+y*(p2+y*(p3+y*(p4+y*(p5+y*(p6+y*p7))))))
  else
    ax=abs(x)
    y=3.75/ax
    bessj0=(exp(ax)/sqrt(ax))*(q1+y*(q2+y*(q3+y*(q4+y*(q5+y*(q6+y*
& (q7+y*(q8+y*q9))))))))))
  endif
  return
end

```

```

real function bessj1(x)
c Returns the modified Bessel function I1(x) for positive real x.
c From "Numerical Recipes" by Press, Flannery, Teukolsky & Vetterling,
c Cambridge University Press, 1986. Based on polynomial coefficients
c given by Abramowitz and Stegun.
  real*8 y,p1,p2,p3,p4,p5,p6,p7,q1,q2,q3,q4,q5,q6,q7,q8,q9
  data p1,p2,p3,p4,p5,p6,p7 / 0.5d0,0.87890594d0,0.51498869d0,
&0.15084934d0,0.2658733d-1,0.301532d-2,0.32411d-3 /
  data q1,q2,q3,q4,q5,q6,q7,q8,q9 / 0.39894228d0,-0.3988024d-1,
&-0.362018d-2,0.163801d-2,-0.1031555d-1,0.2282967d-1,
&-0.2895312d-1,0.1787654d-1,-0.420059d-2 /
  if (abs(x).lt.3.75) then
    y=(x/3.75)**2
    bessj1=x*(p1+y*(p2+y*(p3+y*(p4+y*(p5+y*(p6+y*p7))))))
  else
    ax=abs(x)
    y=3.75/ax
    bessj1=(exp(ax)/sqrt(ax))*(q1+y*(q2+y*(q3+y*(q4+y*(q5+y*(q6+y*
& (q7+y*(q8+y*q9))))))))))
  if (x.lt.0.) bessj1=-bessj1
  endif
  return
end

```

```

      real function bessk0(x)
c Returns the modified Bessel function Ko(x) for positive real x.
c From "Numerical Recipies" by Press, Flannery, Teukolsky & Vetterling,
c Cambridge University Press, 1986. Based on polynomial coefficients
c given by Abramowitz and Stegun.

```

```

      real*8 y,p1,p2,p3,p4,p5,p6,p7,q1,q2,q3,q4,q5,q6,q7
      data p1,p2,p3,p4,p5,p6,p7 /-0.57721566d0,0.42278420d0,
&0.23069756d0,0.3488590d-1,0.262698d-2,0.10750d-3,0.74d-5 /
      data q1,q2,q3,q4,q5,q6,q7 / 1.25331414d0,-0.7832358d-1,
&0.2189568d-1,-0.1062446d-1,0.587872d-2,-0.251540d-2,0.53208d-3 /
      if (x.le.2.0) then
        y=x*x/4.0
        bessk0=(-log(x/2.0)*bessi0(x))+(p1+y*(p2+y*(p3+
&          y*(p4+y*(p5+y*(p6+y*p7))))))
      else
        y=(2.0/x)
        bessk0=(exp(-x)/sqrt(x))*(q1+y*(q2+y*(q3+
&          y*(q4+y*(q5+y*(q6+y*q7))))))
      endif
      return
end

```

```

      real function bessk1(x)
c Returns the modified Bessel function K1(x) for positive real x.
c From "Numerical Recipies" by Press, Flannery, Teukolsky & Vetterling,
c Cambridge University Press, 1986. Based on polynomial coefficients
c given by Abramowitz and Stegun.

```

```

      real*8 y,p1,p2,p3,p4,p5,p6,p7,q1,q2,q3,q4,q5,q6,q7
      data p1,p2,p3,p4,p5,p6,p7 / 1.0d0,0.15443144d0,-0.67278579d0,
&-0.18156897d0,-0.1919402d-1,-0.110404d-2,-0.4686d-4 /
      data q1,q2,q3,q4,q5,q6,q7 / 1.25331414d0,0.23498619d0,
&-0.3655620d-1,0.1504268d-1,-0.780353d-2,0.325614d-2,-0.68245d-3 /
      if (x.le.2.0) then
        y=x*x/4.0
        bessk1=(log(x/2.0)*bessi1(x))+(1.0/x)*(p1+y*(p2+y*(p3+
&          y*(p4+y*(p5+y*(p6+y*p7))))))
      else
        y=2.0/x
        bessk1=(exp(-x)/sqrt(x))*(q1+y*(q2+y*(q3+
&          y*(q4+y*(q5+y*(q6+y*q7))))))
      endif
      return
end

```


A.11. Rn3D INPUT FILE FOR EXAMPLE SIMULATION 8.1

```
$case
  title = ' Two-dimensional Problem with Crack'
$end
$runtype
  flow = 'Transient',
  transport = 'Transient',
  coupled = .false.,
  twophase = .false.
$end
$dimension
  dim = 2.
$end
$time
  delt = 150., dtmult = 1.0, nsteps = 48, t0 = 0.0
$end
$gradients
  dpldz = 0., dtdz = 0.0
$end
$convergence
  itermx = 3, eps = 1.0, tolp = 1.0e-08, tolc = 1.0e-07
$end
$crack
  cdepth = 10., cwidth = .5, nefx=5
$end
$grid
  gxmax = 40., dx1 = 0.01, dx2 = 20., dxmult = 1.5,
  gzmin = -5.,-10.,-15.,-20,-1000.,
  dz1 = 0.001, 1., 0.1, 1., 0.1,
  dz2 = 1., 0.1, 1., 0.1, 20.,
  dzmult = 1.2,0.8,1.2,0.8,1.2
$end
$pressure
  pzones = 1
$end
$boundary
  type = 1,
  bxmin=0., bxmax=40., bymin=0., bymax=0., bzmin=0., bzmax=0.,
  value =
  1.0002500E+06,1.0004330E+06,1.0005000E+06,
  1.0004330E+06,1.0002500E+06,1.0000000E+06,9.9975000E+05,
  9.9956700E+05,9.9950000E+05,9.9956700E+05,9.9975000E+05,
  1.0000000E+06,1.0002500E+06,1.0004330E+06,1.0005000E+06,
  1.0004330E+06,1.0002500E+06,1.0000000E+06,9.9975000E+05,
  9.9956700E+05,9.9950000E+05,9.9956700E+05,9.9975000E+05,
  1.0000000E+06,1.0002500E+06,1.0004330E+06,1.0005000E+06,
```

```

1.0004330E+06,1.0002500E+06,1.0000000E+06,9.9975000E+05,
9.9956700E+05,9.9950000E+05,9.9956700E+05,9.9975000E+05,
1.0000000E+06,1.0002500E+06,1.0004330E+06,1.0005000E+06,
1.0004330E+06,1.0002500E+06,1.0000000E+06,9.9975000E+05,
9.9956700E+05,9.9950000E+05,9.9956700E+05,9.9975000E+05,
1.0000000E+06
$end
$concentration
  czones = 1
$end
$boundary
  type = 1,
  bxmin=0., bxmax=40., bymin=0., bymax=0., bzmin=0., bzmax=0.,
  48*0.
$end
$zones
  mzones = 3,
  mxmin = 0., 0., 0.,
  mxmax = 40., 40., 40.,
  mzmax = 0., -10., -20.,
  mzmin = -10., -20., -1000.,
  matype = 2, 3, 1
$end
$soil
  mats = 3,
  poros = 0.35, 0.25, 0.40,
  denb = 1.7, 2.0, 1.6,
  tortx = 0.6, 0.013, 0.6,
  torty = 0.6, 0.013, 0.6,
  tortz = 0.6, 0.013, 0.6,
  permx = 1.e-6, 4.4e-12, 3.e-3,
  permy = 1.e-6, 4.4e-12, 3.e-3,
  permz = 1.e-6, 4.4e-12, 3.e-3,
  satres = 0., 0., 0.,
  satwet = 1., 1., 1.,
  alpha = 0.0237, 0.0166, 0.0237,
  vann = 4.886, 2.5, 4.886,
  expo = .5, .5, .5,
  ea = .38, 0., 0.,
  ew = .38, 0., 0.,
  satmin = .4, 1., 1.,
  source = 17256.86, 0., 0.
$end
$fluid
  hlife = 3.3e+5, grav=981., cconv=2.8666E-20
$end

```

```
$initial  
  readic = .true.  
$end  
$output  
  plotit = 11,32,35,  
  psteps = 39,45  
$end
```

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

12 DOE/Office of Scientific and
Technical Information

T. B. Borak
Department of Radiology and
Radiation Biology
Colorado State University
Fort Collins, CO 80523

R. Cady
U.S. Nuclear Regulatory Commission
5650 Nicholson Lane
Rockville, MD 20852

W. Fisk
90-3058
Lawrence Berkeley Laboratory
Berkeley, CA 94720

M. Gadd
Department of Radiological Health
Sciences
Colorado State University
Fort Collins, CO 80523

D. Genereux
Ralph M. Parsons Laboratory for
Water Resources and
Hydrodynamics
Department of Civil Engineering
Massachusetts Institute of
Technology
Cambridge, MA 02138

W. Graustein
Department of Agronomy and Soil
Science
University of Hawaii
1910 East-West Road
Honolulu, HI 96822

R. B. Gammage
Martin Marietta Energy Systems, Inc.
P.O. Box 2001
Oak Ridge, TN 37831-6383

D. Gonzalez
Jacobs Engineering
5301 Central Ave. NE, Suite 1400
Albuquerque, NM 87108

B. Howell
Environmental Science Division
Lawrence Livermore National
Laboratory
P.O. Box 5510, L524
Livermore, CA 94550

C. K. Keller
Department of Geology
Washington State University
Pullman, WA 99164-2812

B. K. Lamb
Department of Civil & Environmental
Engineering
Washington State University
Pullman, WA 99164-2910

B. S. Levy
Camp Dresser & McKee, Inc.
One Center Plaza
Boston, MA 02108

F. T. Lindstrom
Reynolds Electrical Engineering Co.
M/S 738
P.O. Box 98521
Las Vegas, NV 89193-8521

K. K. Nielson
Rogers & Associates Engineering
Corp.
P.O. Box 330
Salt Lake City, UT 84110

C. R. Olsen
Environmental Sciences Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6036

No. of
Copies

P. C. Owczarski
SAIC
1845 Terminal Drive, Suite 202
Richland, WA 99352

R. T. Peake
U.S. Environmental Protection
Agency
Office of Radiation Programs
401 M Street S.W.
Washington, DC 20460

F. M. Phillips
Geoscience Dept.
New Mexico Institute of Mining
and Technology
Socorro, NM 87801

M. Reimer
U.S. Geological Survey
MS 963
Denver Federal Center
Denver, CO 80225

A. W. Rose
Dept. of Geosciences
Penn State University
University Park, PA 16802

S. D. Schery
Physics Dept.
New Mexico Institute of Mining
and Technology
Socorro, NM 87801

R. R. Schumann
U.S. Department of the Interior
P.O. Box 25046, MS 939, DFC
Denver, CO 80225

J. P. Schubert
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

R. Sextro
Indoor Air Program
Lawrence Berkeley Laboratory
Berkeley, CA 94720

No. of
Copies

A. B. Tanner
U.S. Department of the Interior
Geological Survey
990 National Center
Reston, VA 22092

D. L. Thomas
S.O.E.S.T.
University of Hawaii
2525 Correa Road
Honolulu, HI 96822

S. E. Trumbore
Department of Geosciences
UC Irvine
Irvine, CA 92717-3100

Y. W. Tsang
Earth Sciences Division
Lawrence Berkeley Laboratory
Berkeley, CA 94720

J. W. Washington
Meiser & Earl, Inc.
1512 West College Ave.
State College, PA 16801

J. L. Wilson
Geoscience Dept.
New Mexico Institute of Mining
and Technology
Socorro, NM 87801

FOREIGN

P. Doyle
5-1939 Carling Avenue
Ottawa, Ontario
CANADA K2A 1E9

R. Grasty
Energy, Mines and Resources
Canada
601 Booth Street
Ottawa, Ontario
CANADA K10 0E8

C. Wray
GK Yill and Associates, Ltd.
1441 Pembina Highway
Winnipeg, Manitoba
CANADA R3T 2C4

No. of
Copies

No. of
Copies

ONSITE

52 Pacific Northwest Laboratory

M. P. Bergeron	K6-77
C. R. Cole	K6-77
J. L. Cox	P8-38
F. T. Cross	K4-13
J. L. Devary	K6-77
M. D. Freshley	K6-77
H. D. Freeman (5)	P8-38
G. W. Gee (5)	K6-77
T. R. Ginn	K6-77
D. J. Holford (20)	K6-77
A. C. James	K3-51
R. R. Kirkham	K6-77
K. R. Roberson	K6-77

J. L. Smoot	K6-77
L. W. Vail	K6-77
R. E. Wildung	P7-54
B. D. Wood	K6-77
S. K. Wurstner	K6-77
Publishing Coordination	K1-06
Technical Report Files (5)	

Routing

R. M. Ecker	SEQUIM
M. J. Graham	K6-78
P. M. Irving	K6-98
C. S. Sloane	K6-04
P. C. Hays (last)	K6-86

DATE

FILMED

3 / 17 / 94

END