

WIPP PA
User's Manual

for

NUTS (Version 2.05)

Document Version 2.05

WPO #46002

JUNE 18, 1997

Information Only

CONTENTS

1.0 INTRODUCTION	6
1.1 Software Identifier	6
1.2 Points of Contact.....	6
2.0 FUNCTIONAL REQUIREMENTS	7
3.0 REQUIRED USER TRAINING AND/OR BACKGROUND	8
4.0 DESCRIPTION OF THE MODEL AND METHODS	10
4.1 Background	10
4.1.1 NUTS's General Role in the CCA PA.....	10
4.1.2 Overview of NUTS's Capabilities.....	11
4.2 Theoretical Foundations of Transport in Fractured Porous Media Having Dual-Porosity and Dual-Permeability	12
4.2.1 Introduction to Fractured Porous Media	12
4.2.2 Basic Dissolution Processes.....	15
4.3 Differential Equations Governing Transport in Fractured Porous Media	16
4.3.1 Advective Transport.....	16
4.3.2 Advective Transport Across Fracture/Matrix Interfaces	19
4.3.3 Diffusive and Dispersive Transports.....	20
4.3.4 Asymmetric Dispersion.....	23
4.3.5 Sorption-Desorption	24
4.3.6 Decay.....	27
4.3.7 The Overall Mass-Transport Equation.....	28
4.3.8 Initial and Boundary Conditions	29
4.4 Discretization Methods	30
4.4.1 Discrete Analogues of the Conservation of Mass Equations.....	37
4.4.2 One-Point Upstream-Winding Discretization of Transport Equations	39
4.4.3 Two-Point Upstream-Winding Discretization of Transport Equations	43
4.4.4 Mid-Point Weighting Discretization of Transport Equations	48
4.5 Other Important Physical and Chemical Aspects of the Modeling.....	53
4.5.1 Temperature Dependency	53
4.5.2 Theoretical Foundations for the Gas-Transport Model Phase	54
4.5.3 Precipitate Models in NUTS	56
5.0 INHERENT CAPABILITIES AND LIMITATIONS OF THE SOFTWARE.....	62
6.0 USER INTERACTIONS WITH THE SOFTWARE	63
6.1 Exercising NUTS Interactively	63
6.2 User-Interactive Options Afforded by NUTS-SA.....	63
6.3 User-Interactive Input Sequence	64
6.3.1 Is This a DEBUG Run? (Y/N) {N}.....	64
6.3.2 The Default Input-File Name is NUTS.IN.....	64
6.3.3 The Default Output File Name is NUTS.OUT	64
6.3.4 Read the material map from BRAGFLO input file (Y/N)? [Y].....	65
6.3.5 Is This a Test Run? (Y/N) {N}	65
6.3.6 Enter Type of Output File {ASC}.....	65

6.3.7	Enter Type of Porous Medium {MATRIX}	66
6.3.8	Enter Number of Phases {1}	66
6.3.9	Enter the Type of Phase (Liquid/Gas) {L}	66
6.3.10	Enter the Type of Compiler (<i>character*32</i>)	66
6.3.11	Enter the name of the computer system	67
6.4	NUTS's Parameter Statement	67
6.5	Input/Output Files Used by NUTS-SA	70
6.5.1	ASCII Radioisotope Input File.....	71
6.5.1.A	Program Controls and Flags.....	71
6.5.1.A1	Site Description and Flags	71
6.5.1.A2	Single-Porosity Fracture Controls and Flags	76
6.5.1.A3	Single-Porosity Matrix Controls and Flags.....	80
6.5.1.A4	Dual-Porosity/Dual-Permeability Controls and Flags	84
6.5.1.A5	Sorption Equilibrium Constant (k_d)Temperature-Dependency Information .	85
6.5.1.A6	Molecular-Diffusion Temperature-Dependency Flags	85
6.5.1.A7	Printing-Frequency Information	86
6.5.1.A8	External Source Flag.....	86
6.5.1.A9	Time-Minimum Limit	87
6.5.1.A10	Intrusion Time Interpolation and Extrapolation	87
6.5.1.B	Material Map Input.....	88
6.5.1.C	Properties Input	90
6.5.1.C1	Physical Properties	90
6.5.1.C2	Component Molecular-Diffusion Input.....	93
6.5.1.C3	Rock-Density Input	94
6.5.1.C4	Waste-Matrix Input	94
6.5.1.C5	Single-Porosity Fracture Input	95
6.5.1.C6	Single-Porosity Matrix Input.....	103
6.5.1.C7	Dual-Porosity or Dual-Permeability Input	110
6.5.2	ASCII Flux-Field Input File (homogeneous)	110
6.5.2.A	Title Information	110
6.5.2.B	Time Information.....	111
6.5.2.C	Dimension Information	111
6.5.2.D	Fracture Information	111
6.5.2.E	Matrix Information	112
6.5.2.F	Dual-Porosity/Dual-Permeability Information	112
6.5.3	ASCII Flux-Field Input File (heterogeneous)	113
6.5.3.A	Title Information	113
6.5.3.B	Time Information.....	113
6.5.3.C	Dimension Information	113
6.5.3.D	Fracture Information	114
6.5.3.E	Matrix Information	115
6.5.3.F	Dual-Porosity/Dual-Permeability Information	116
6.5.4	Binary Flux-Field Input File	116
6.5.5	Binary Output Files	116
6.5.5.A	Binary Output Files (BRAGFLO Style)	116
6.5.5.B	Binary Output Files (NUTS Style).....	116
6.5.6	ASCII Output Files.....	117
6.5.6.A	Extensive ASCII Output File	117

6.5.6.B Specific-Ranges ASCII Output File	117
6.5.7 Debug (Scratch) ASCII Output File.....	117
6.6 Generating an Executable for NUTS-SA:.....	117
6.7 Input/Output (I/O) and Other Files Used by NUTS-CC	117
6.7.1 User-Interactive Input.....	117
6.7.2 NUTS-CC Parameter Statement.....	118
6.7.3 Other Files Used by NUTS-CC.....	118
6.7.4 Binary (CDB) Flux-Field and the Associated ASCII Input Filea	119
6.7.5 NUTS's Binary (CDB) Output File	119
6.7.6 NUTS's Properties Binary (CDB) Input File.....	121
6.8 Generating an Executable for NUTS-CC.....	122
6.9 Exercising NUTS from a Command Line.....	122
7.0 DESCRIPTION OF NUTS'S INPUT FILES	123
8.0 ERROR MESSAGES	124
9.0 DESCRIPTION OF NUTS'S OUTPUT FILES	125
10.0 REFERENCES	126
11.0 APPENDICES	130
Appendix A: Examples of ASCII Radioisotope Input Files.....	130
Appendix B: Example of a Nine-Grid-Block, One-Time Material Map	143
Appendix C: Example of an ASCII Flux-Field Input File (homogeneous)	144
Appendix D: Example of an ASCII Flux-Field Input File (heterogeneous)	145
Appendix E: Example of NUTS's Extensive ASCII Output File	147
Appendix F: Command File to Build NUTS-SA's Executable	160
Appendix G: Command File to Build NUTS-CC's Executable.....	161
Appendix H: Command Files and ASCII Files that Execute NUTS on the WIPP Alpha-VAX Cluster.....	162
Appendix I: Listing of NUTS's Error Messages	178
Appendix J: An Example of NUTS's Debug Output File.....	188
Appendix K: FORTRAN Program to Convert NUTS's Binary Output to a Readable Format .	199
Appendix L: Review Forms	221

LIST OF FIGURES

3-1	The code sequence for NUTS as it is exercised in the 1996 CCA.....	9
4-1	Schematic Diagram of Fractured REV.....	15
4-2	Schematic Diagram Showing the REV and the Fluxes q^*	17
4-3	Symmetric versus Asymmetric Dispersion Fronts.....	24
4-4	Point-Center Grid in Finite-Difference Approximation.....	31
4-5	Block-Center Grid in Finite-Difference Approximation.....	33

LIST OF TABLES

1	NUTS Features Used in the CCA Calculations.....	13
2	Variables used in NUT_PARAM.INC.....	69
3	Constants used in NUT_PARAM.INC.....	70
4	CDB Output Variables and Their Description	121

1.0 INTRODUCTION

This document is the User's Manual for the WIPP radioisotope-mobilization-and-decay code named NUTS (version number 2.05). It discusses the code, its execution, and its performance in the context of the 1996 Waste Isolation Pilot Project (WIPP) Compliance Certification Application (CCA) Performance Assessment (PA), and in that context only. The manual identifies the code, its authors and expert consultants (Section 1). It describes the code's WIPP-PA purposes and functions (Section 2), provides recommended user training (Section 3), outlines the code's theoretical basis and numerical methods (Section 4), its inherent capabilities and limitations (Section 5), describes user interactions (Section 6), input files (Section 7), error messages (Section 8), and output files (Section 9), and provides examples of relevant input, output, and debug files in its Appendices as well as calculations of interest (distributed throughout). Efforts have also been made to assist users in understanding how to make the required input-option decisions, to provide insights into the structure and use of the code's various input and output files, and explain the code's operation. Several examples of input files are included to help in running NUTS efficiently for cases likely to be encountered.

1.1 Software Identifier

Code Name: NUTS, a NUclide* Transport System

WIPP Prefix: NUT

Version Number: 2.05

Date: 30 May 1997 (CMS Build Date)

Platform: FORTRAN 77 for OpenVMS AXP, ver. 6.1, on a DEC Alpha VAX (see Section 2.3)

1.2 Points of Contact

Sponsor: Ali A. Shinta
Applied Physics, Inc.
2201 Buena Vista Drive, SE (fourth floor)
Albuquerque NM 87106
Voice: 505 766 9629
Fax: 505 766 9125

Consultant: Palmer Vaughn
Sandia National Laboratories
Department 6849
Albuquerque NM 87185
Voice: 505 848 0678
Fax: 505 848 0705

* Throughout this document, the word "Nuclide" is intended to be synonymous with "Isotope," and never to mean "an atomic nucleus with regard to energy level."

2.0 FUNCTIONAL REQUIREMENTS

- R.1 First order decay of radioactive constituents in multiple member chains.
- R.2 Retardation due to sorption of the chemical constituents via a linear equilibrium isotherm.
- R.3 One-dimensional convection-dispersion-decay with spatially constant velocity and dispersion coefficients.
- R.4 The ability to execute to completion under a data range geometry, and dimensions similar to that anticipated in the WIPP PA calculation and resulting in a qualitatively correct mass balance.
- R.5 Time-varying Source Terms, with a slab source (non-point source).
- R.6 Two-dimensional transport with convective, dispersion, decay, sorption, restricted to flow fields that are aligned with the grid if dispersion is not zero (i.e, no cross-terms are required in the governing equations) and restricted to dispersivities that are consistent with first-order upwind differencing.
- R.7 Non-uniform, or stretched Cartesian grids.
- R.8 Spatially variable material coefficients (e.g. Saturation, Porosity, Dispersivity, Flow-Fields).
- R.9 Solubility limit with,
 - A) an explicit linear precipitation model and colloid preferential solubility at the inventory limit (This requirement is only tested by a regression test in Test Case #8).
 - B) the implicit precipitation model .
- R.10 Implementation of Neumann boundary conditions.
- R.11 Implementation of Dirichlet conditions.

NUTS has a considerable number of external-interface requirements. They are listed in Section 2.4 of the code's Requirements Document (WPO#37924).

3.0 REQUIRED USER TRAINING AND/OR BACKGROUND

To exercise NUTS, users should have (1) basic knowledge of open VMS, (2) basic facility with Digital Command Language, and (3) an overall understanding of Sandia's CAMDAT database, which is used in virtually every WIPP code (Rechard, 1992; and Rechard et al., 1993). User's should also have (4) access to the WIPP cluster of Alpha-VAX microcomputers or their functional equivalents.

To manipulate and/or interpret the results of NUTS as it is exercised in WIPP PAs, users should have (1) a basic understanding of the chemistry of radioactive decay, solution chemistry, sorption chemistry, fluid flow and fluid transport in porous media; (2) a basic understanding of partial-differential equations and integral calculus, as they apply in the mathematical formulation of physical principles and especially of conservation of mass, (3) a generalist's conceptual understanding of numerical methods as they are applied to the numerical solution of the boundary-value problems of mathematical physics and chemistry, and (4) a basic overview understanding of the WIPP PA process, including conceptual models, scenarios, inventories, release routes, uncertainty sampling, input-data vectors, and a general familiarity with the files and functions of the WIPP codes that run prior to exercising NUTS, principally BRAGFLO and ALGEBRA, and those that exercise immediately after NUTS, principally CCDFGF (WIPP PA Dept [5 Volumes], 1992). An annotated flow diagram of NUTS's 1996 CCA code sequence is shown in Figure 3.1.

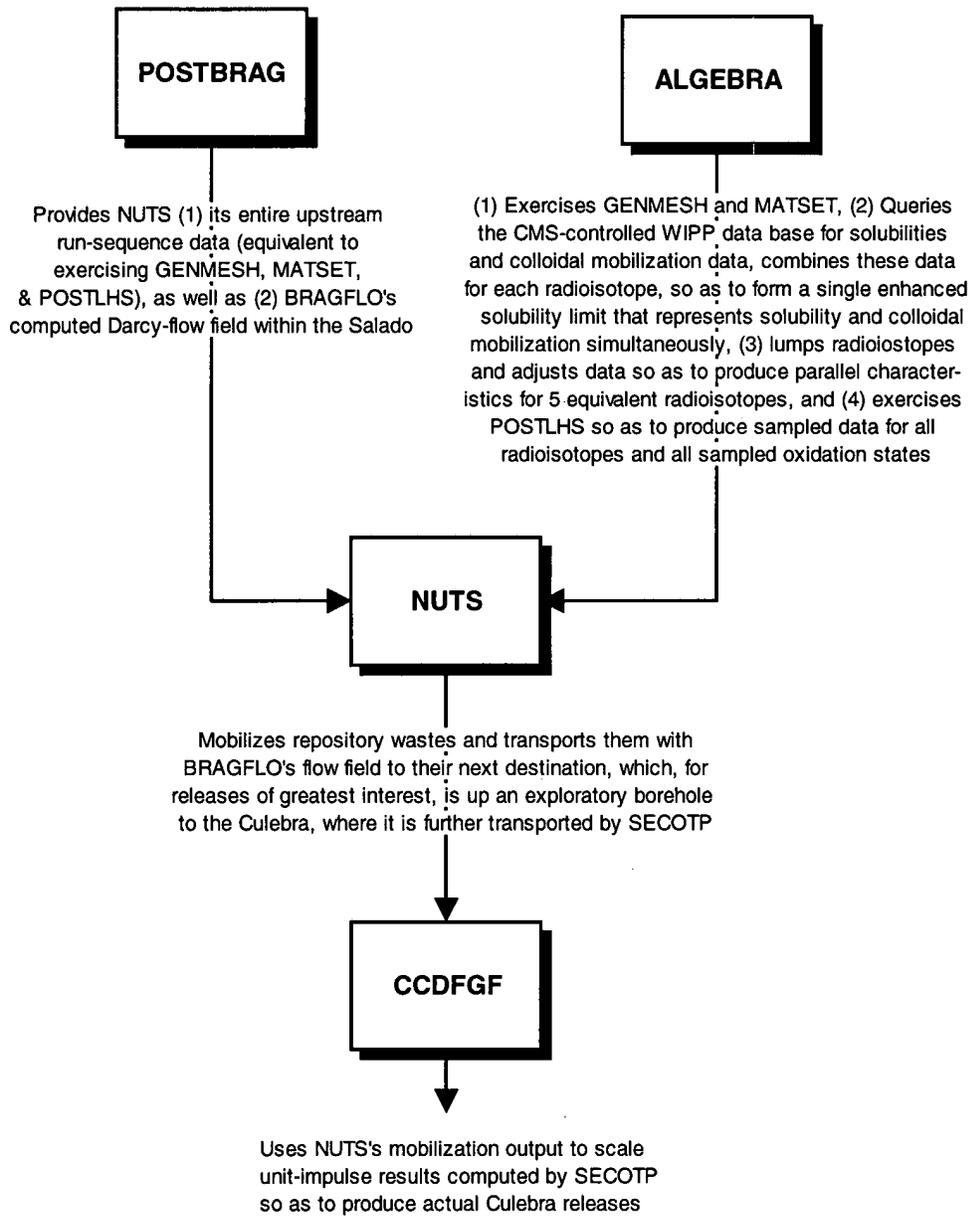


Figure 3.1: The code sequence for NUTS as it is exercised in the 1996 CCA.

4.0 DESCRIPTION of the MODEL and METHODS

4.1 Background

4.1.1 NUTS's General Role in the CCA PA

NUTS's principal CCA PA role is to estimate the radioactive contaminant load mobilized into the brine phase of the brine/gas mixture that seeps or flows through and around the WIPP repository's decommissioned waste panels. Mobilization by any process, for example dissolution or suspension on colloids, is modeled as taking place instantaneously. The contaminants introduced into the brine are the aged radioisotopes that are assumed to reside in the repository at the time of decommissioning plus any progeny of those radioisotopes that may have been produced through natural decomposition. Thus, a detailed inventory of the radioactive isotopes stored in the repository at the time of decommissioning must be provided to NUTS before it may be exercised.

NUTS plays no role in the physics of the fluid flow in or near the repository. The time history of the detailed (Darcy) volumetric flow rate throughout the repository must be provided to NUTS by an independent hydrological-flow code. In the case of the CCA, that code is BRAGFLO.

In the CCA WIPP assessment scenarios, brine is assumed to enter the repository panels in either of two very different ways, namely: (a) by natural seepage from the surrounding Salado formation, and (b) by various sorts of unnatural flows induced by exploratory boreholes. In the case of undisturbed operation, brine can seep into the repository through the disturbed rock zone from the surrounding undisturbed halite and marker beds. In the case of a repository breached by an exploratory borehole, the brine could flow into and through the repository via the pipe-like channels created by the borehole(s). The most intense flow would occur if the borehole also penetrated a deep pressurized brine pocket, and some boreholes are assumed to do so.

The natural time scales associated with deep, tight-media, groundwater flows are typically centuries, and WIPP intrusion scenarios include temporal lapses of millennia prior to the hypothesized breaching of repository waste panels by boreholes. Moreover, EPA regulations extend to 10,000 years after decommissioning. On these time scales, radioisotopes of interest exhibit significant natural decay by which they transform to other radioactive and non-radioactive isotopes and/or compounds in well-established ways. Thus, it is required to quantify the decay process and maintain a running record of the decayed contents of the repository as well as all the products of decomposition from the time of decommissioning onward to 10,000 years. NUTS does that.

In the context of the CCA, NUTS has a single principal area of application. It is this: given the defining characteristics of the flow of uncontaminated brine into the repository's waste panels, NUTS is required to model the mobilization of radioisotopes by the brine that wets the waste panels, and to calculate, given the fluid flow rates, the rate at which each isotope that was initially present in the repository, including the radioactive progeny of those isotopes, exits the repository via the now contaminated brine flow. The exit rate of contaminants depends on (i) the outflow rate of the brine and (ii) the concentration of isotopes within the flowing brine. The first

quantity is provided in advance by BRAGFLO, which analyses the 2-phase Darcy flow throughout the Castile, Salado, Rustler, and Dewey Lake Formations throughout the entire Land Withdrawal Act region. NUTS is required to calculate the second quantity (concentration of each radioisotope), and to do so by treating all the important physical processes that can affect transport via a brine carrier flowing through a porous medium.

Mobilization is assumed to take place instantaneously and to maximum capacity, i.e. to the "solubility" limit, given adequate stocks. In the CCA, the "solubility limit" is actually a maximum mobilized concentration that includes the maximum concentration of an element mobilized on four types of colloids as well as the solubility of the element. Construction of this maximum mobilized concentration is performed before the NUTS calculation using ALGEBRA.

4.1.2 Overview of NUTS's Capabilities

NUTS is a multidimensional, multicomponent, radioactive-contaminant transport code. It is designed to apply to single-porosity (SP), dual-porosity (DP), and/or dual-permeability (DPM) porous media. In two dimensions, the discretization employs five-point finite-difference methods. NUTS is designed to treat an enormously broad spectrum of transport problems. However, many of its options have been disabled in the CCA calculations for the sake of simplicity. CCA-disabled options are described briefly herein, and are clearly noted as disabled for CCA calculations.

The model simulates radioisotope transport through porous media and includes first-order radioactive decay processes. However, the simulator is not limited to radioactive materials. The transport of materials of all kinds, radioactive or not, may be simulated. In CCA applications, only radioisotopes are transported by NUTS.

In treating sorption between the waste and the media that surround it, NUTS allows for three types of sorption isotherms. They are: (1) linear, (2) Freundlich, and (3) Langmuir equilibrium isotherms. **Only option (1) is tested for regulatory calculations, and none was used in the CCA calculations. A type of sorption onto colloids, however, was modeled in the ALGEBRA calculation run prior to NUTS.**

NUTS normally models hydrodynamic dispersion under the assumption that the porous media through which the transport occurs are dispersively isotropic. However, for CCA calculations, both molecular and mechanical dispersion are taken to be zero.

"Dissolution" of waste components into the brine carrier medium is modeled, and the possibility of their precipitation during migration is included, should solutes exceed their local "solubility" limits. Precipitates are required to undergo decay and are permitted to "redissolve" in the brine if the concentration drops below the solubility limit.

Representation of multiple radioactive sites (repositories) is also possible. In that case, the contribution from each site to the component concentration and precipitation, if any, at each computational node can be found. A similar technique may be used to treat a daughter isotope

generated from the decay of different parents. **However, use of this aspect of the code is not contemplated for the WIPP CCA PA, and it is therefore not tested.**

The system of governing partial-differential equations (mass conservation for each radioisotopic constituent) is discretized and solved sequentially to determine the contribution from a decaying parent to the immediate daughter. In the sequential method, the solution proceeds progressively from the top of each radioactive chain. Therefore, the contribution to any daughter from its decaying parent is available. This approach avoids the numerical problems associated with inverting a large, sparsely-populated matrix in which the bands are not well structured.

NUTS was developed specifically to assist in performance assessments of the WIPP. **It is that application, and specifically the CCA calculation, that will be described herein.** However, in its broadest sense, NUTS can be used in other applications involving contaminant transport of all kinds through porous media. Table-1 provides a summary for the features used in the CCA calculations.

4.2 Theoretical Foundations of Transport in Fractured Porous Media having Dual-Porosity and Dual-Permeability

In the textbook approach to dual-permeability, the partial differential equations (PDEs) representing transport in a fractured, porous medium are normally written as (1) the PDEs governing transport along the fractures, and (2) the PDEs governing transport through the matrix. The two PDEs are coupled by a transfer function that represents material transport between the two media (Kazemi et al, 1976; Hill and Thomas, 1985; Litvak, 1985; Coats, 1989). That is the approach taken herein.

4.2.1 Introduction to Fractured Porous Media

A fracture can be defined as a surface of discontinuity resulting from loss of cohesion of rock materials due to rupture. Most fractures in brittle rocks of low porosity occur as a result of tectonic processes. These fractures are usually large in size and extent, and are called macrofractures. In less brittle rocks with higher intergranular porosities, the resulting fractures are usually smaller and less extensive, and are called microfractures, or fissures. Fractures generated by the state of stress in the earth's crust can be attributed to (van Golf-Racht, 1982):

1. folding and faulting;
2. rock volume shrinkage and the dehydration process;
3. volume shrinkage associated with heat loss; and
4. differential stress on the plane of weakness due to deep erosion of overburden rocks.

Table 1. NUTS Features Used in the CCA Calculations

NUTS Features	Features in the CCA Calculations	
	Used	Disabled
Single-porosity, fracture		x
Single-porosity, matrix	x	
Double-porosity		x
Double-permeability		x
Advective transport	x	
Diffusive-dispersive transport (Symmetric and Asymmetric)		x
Sorption		x
Colloid transport ⁺		x
Decay	x	
Gas phase transport		x
Temperature dependency		x
Multiple sites		x
Precipitation*(Implicit and Explicit)	x	
Solubility limit	x	
Interior sources	x	

⁺ Colloids are lumped with dissolved inventory.

* Explicit precipitate is used in the CCA calculations. Implicit precipitate, however, is used in the EPA PAV calculations.

Thus, fractured porous media are typically conventional sedimentary materials possessing discontinuities introduced later by some sort of physical activity. Diagenesis processes, such as mineral deposition (calcite), and dissolution of matrix and formation of stylolites may be associated with fracture formation. Fractures may be regular or irregular, closed and filled with cement, or open to flow. Most fractured systems, occurring in a large variety of rock mineralogy, are described adequately (for hydrological purposes) by specifying the fracture spacing, orientation, length, and degree of cementation. These data are usually obtained from cores and outcrops coupled with rock mechanics studies, well logging, and seismic-profile distortion. Because of the inherent complexity associated with the characterization of these mineral discontinuities, porous-medium parameters such as porosity, permeability, etc., normally have a wide range of numerical uncertainty. Accordingly, the classical models of fluid flow and transport in porous media have been shown to be inadequate to represent a heterogeneous system in which the heterogeneities are due to fractures, vugs, and channels. A new and more complex model is required, the complexity deriving from the existence of two entirely different but interconnected paths for fluid flow and transport. Because natural fractures tend to be irregular in their distribution, orientation, and extension, some sort of reasonable idealization is required as a basis for a reasonable physical and mathematical description of the medium.

The basic theory of fluid flow in fractured porous media was introduced by Barenblatt and Zeltov (1960), who applied a continuum approach. Their approach is similar to the approach used in classical fluid-dynamical theories in which the physical properties of a cluster containing many molecules is allowed to persist in the limit of smallness. Barenblatt and Zeltov choose a smallest physical scale that lies between microscopic and macroscopic representations of fluid properties in regular porous media. In their approach, a representative elementary volume (REV), also called a control volume, is considered. It is much larger than pore size (i.e., it contains many pores and grains) but much smaller than the total domain considered. The REV's average porosity, permeability, pressure, saturation, concentration, and flux are ascribed to the mathematical point residing at the center of the REV (see Bear, 1972), which forms the basis of their hypothesis for a continuum. In applying a strictly continuum approach to dual-porosity or dual-permeability systems, one can assign a pair of average properties to each mathematical point; one representing fracture properties and one representing matrix properties. In other words, the medium consists of two overlapping continua that communicate one with another. In this formulation, the REV contains many matrix blocks and fractures (see Figure 4.1). In actual double-porosity conceptualizations, the fractures, because of their high permeability relative to the matrix and high intensity (small spacing), are assumed to constitute a continuous medium. The matrix blocks between them are discontinuous and form the main storage region for the fluid. In double-permeability formulations, fracture spacings are larger, and matrix-to-matrix flow is permitted in the mathematical formulation of the problem*.

* For more information about these conceptualizations, the reader is referred to Beckner *et al.* (1991), Chen *et al.* (1987), Dean and Lo (1988), DeSwaan (1978, 1982), Duguid and Lee (1977), Dutra and Aziz (1991), Dykhuzeev (1990), Evans (1982), Gilman and Kazemi (1983, 1988), Gilman (1986), Huang-Zhang (1983), Kazemi and Merrill (1979), Kazemi *et al.* (1989), Narasimhan (1982), Pruess and Narasimhan (1985), Reiss *et al.* (1973), Rossen (1979), Saidi (1983), Shinta and Kazemi (1993), Sonier *et al.* (1986), Thomas *et al.* (1983), Wu and Pruess (1988).

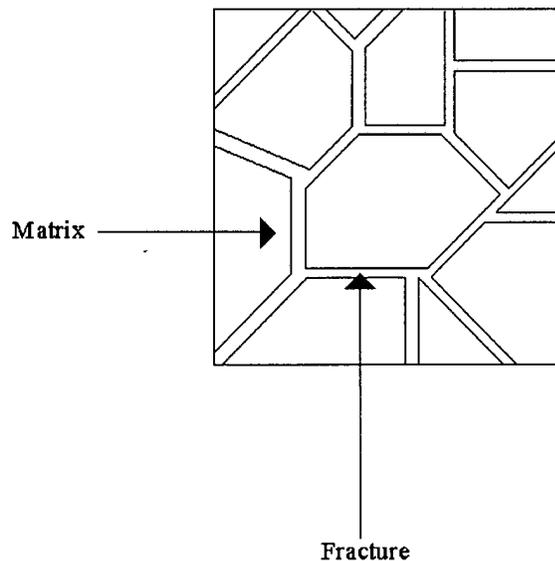


Figure 4.1 Schematic Diagram of Fractured REV.

4.2.2 Basic Dissolution Processes

When salty ground water comes into contact with repository wastes, dissolution of the soluble isotopes begins to take place and continues until equilibrium concentrations are attained, or until all undissolved supplies of radioisotopes are consumed. The capacity of WIPP brines to dissolve radioisotopes is controlled by temperature, pressure, and chemical properties of the brine and of the solutes. The main physical principal on which NUTS's transport equations are based is conservation of mass, and the principal dependant variable in which those equations are written is C , the concentration of solute (i.e., radioisotope) in the underlying fluid (i.e., brine). A pair of conservation equations arises for each radioisotope, one for the fracture domain and one for the matrix domain. The equation pairs are coupled one to another by transfer functions that represent the flux of a given radioisotope between the matrix and fracture domains of the flow. The equation pairs are also coupled one with another by radioactive decay processes. Thus, a complex system of coupled partial differential equations is anticipated.

The concentration, C , of a given radioisotope must be non-negative everywhere in the domain. However, it also has an upper bound, equal to the maximum solubility limit for each element of the waste in whichever brine or combination of brines is present. There are two principal sources of brine, one of Salado origin and one of Castile origin (WIPP PA Dept, 1992, vol 1), and the brines are chemically different. The different brines come into play in the different scenarios. NUTS must be given the solubility limit of all soluble radioactive elements as part of its input data. If radioisotope concentrations should exceed their solubility limits, NUTS will precipitate the excess in the grid block. NUTS subjects precipitates to decay and treats precipitation as a reversible process. If the concentration of a solute drops below the solubility limit, WIPP brines are permitted to redissolve from the precipitate a mass sufficient to restore the solubility limit of

that solute, assuming sufficient precipitate is available. Further details on precipitation and its numerical implementation is available in Section 4.5.3 of this manual.

4.3 Differential Equations Governing Transport in Fractured Porous Media

The mathematical formulation for the dual-continuum approach requires a mass-conservation equation for each continuum. Each such equation contains a coupling term (transfer function) that represents intercommunication (i.e., transport) between the two domains. The continuity-equation development that follows is general and therefore applies to both matrix and fracture domains, provided that the coupling term is added. In porous media, the principal focus is on the transport of a solute whose intensity is measured in terms of a fluid concentration measured in Kg of solute per Kg of fluid. The mass-conservation equation (continuity equation) is derived by considering a REV shaped like a rectangular parallelepiped in the direction of the Cartesian coordinates (x, y, z) , and having an infinitesimal volume of $\delta x \delta y \delta z$. The mass conservation principle can be stated in words as follows:

$$\text{Mass Flux in} - \text{Mass Flux out} + \text{Source / Sink} = \text{Accumulation}$$

Boundary fluxes and internal sources derive from a myriad of physical processes, which we will describe and quantify individually in the subsections that follow.

4.3.1 Advective Transport

The mass-flux vector of the solute is represented as q^* , and its three Cartesian components as q^*_x , q^*_y , q^*_z , as shown in Figure 4.2.

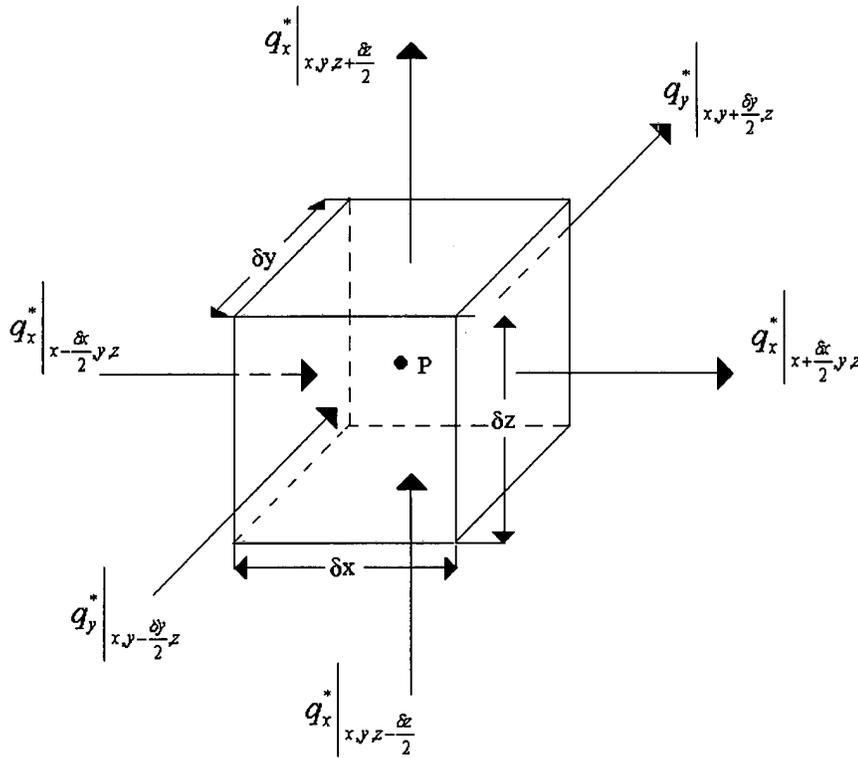


Figure 4.2 Schematic Diagram Showing the REV and the Fluxes q^* .

The net mass inflow associated with flow in the x-direction during the time period δt is:

$$\left[q_x^* \Big|_{x-\frac{\delta x}{2}, y, z} - q_x^* \Big|_{x+\frac{\delta x}{2}, y, z} \right] \delta y \delta z \delta t. \quad 4.1$$

Using Taylor's series to expand around the central point $P(x,y,z)$ of the REV, Equation 4.1 becomes:

$$\left[\left\{ q_x^* - \frac{\delta x}{2} \frac{\partial q_x^*}{\partial x} + \frac{(\delta x)^2}{8} \frac{\partial^2 q_x^*}{\partial x^2} - \dots \right\} - \left\{ q_x^* + \frac{\delta x}{2} \frac{\partial q_x^*}{\partial x} + \frac{(\delta x)^2}{8} \frac{\partial^2 q_x^*}{\partial x^2} + \dots \right\} \right] \delta y \delta z \delta t. \quad 4.2$$

Simplifying and neglecting higher order terms (which go to zero in the limit), Equation 4.2 reduces to:

$$-\frac{\partial q_x^*}{\partial x} \delta x \delta y \delta z \delta t.$$

Repeating the same procedure in the other two directions, the net mass of solute accumulated due to flux across the boundaries during the time period δt is:

$$-\left[\frac{\partial q_x^*}{\partial x} + \frac{\partial q_y^*}{\partial y} + \frac{\partial q_z^*}{\partial z} \right] \delta x \delta y \delta z \delta t . \quad 4.3$$

The net mass of solute accumulated in the control volume from internal sources (and lost to sinks) during the time period δt is:

$$q_{s/s}^* C'_{i/p} \delta x \delta y \delta z \delta t , \quad 4.4$$

where all symbols are defined in the summary list below. The net increase in mass of solute in the control volume during time δt may also be written as the local time rate of change of solute mass in the volume times δt , that is, as:

$$\frac{\partial(\phi S \rho C')}{\partial t} \delta x \delta y \delta z \delta t . \quad 4.5$$

Combining the terms in Equations 4.3, 4.4 and 4.5 according to the word equation stated in Section 4.3 results in:

$$-\left[\frac{\partial q_x^*}{\partial x} + \frac{\partial q_y^*}{\partial y} + \frac{\partial q_z^*}{\partial z} \right] \delta x \delta y \delta z \delta t + q_{s/s}^* C'_{i/p} \delta x \delta y \delta z \delta t = \frac{\partial(\phi S \rho C')}{\partial t} \delta x \delta y \delta z \delta t . \quad 4.6$$

Dividing Equation 4.6 by $\delta x \delta y \delta z \delta t$ leads to:

$$-\nabla \cdot q^* + q_{s/s}^* C'_{i/p} = \frac{\partial(\phi S \rho C')}{\partial t} , \quad 4.7$$

where the nomenclature is defined as follows:

$$q^* = \rho v C' \quad 4.8$$

and

$q_{s/s}^*$ = solvent sink/source mass rate per unit volume (kg/s/m^3)

ρ = solvent density (kg/m^3)

ϕ = porosity of porous medium (dimensionless fraction)

v = solvent advective velocity (m/s)

C' = solute concentration (kg/kg)

$C'_{i/p}$ = injected/produced solute concentration (kg/kg)

t = time (s)

S = saturation (dimensionless fraction)

In Equation 4.7, if q^* is replaced by its equivalent value (Equation 4.8), and $\rho C'$ is replaced by C , then Equation 4.7 becomes:

$$-\nabla \cdot vC + qC_{i/p} = \frac{\partial(\phi SC)}{\partial t}, \quad 4.9$$

where:

C = solute concentration (kg/m^3)

q = solvent sink/source volumetric rate per unit volume ($\text{m}^3/\text{s}/\text{m}^3$)

4.3.2 Advective Transport Across Fracture/Matrix Interfaces

Equation 4.9 as written applies to the fluid domain as a whole (i.e., matrix plus fracture). However, in our conceptualization, we divided the domain into two separate flow regimes, matrix and fracture, and we allowed the two to intercommunicate through a transfer function. The transfer function represents mass flux from the fracture to the matrix or vice versa. Therefore, the transfer function that appears as a source in the matrix equation will appear as a sink (i.e., the same term, but with its sign reversed) in the fracture equation. That is, what is lost from one continuum is gained by the other. The continuity equation that applies to either domain is written as follows:

$$-\nabla \cdot v_f C_f + q_f C_{f/i/p} - \bar{v}_{m/f} C_{m/f} = \frac{\partial(\phi SC)_f}{\partial t}, \text{ for the fracture, and} \quad 4.10a$$

$$-\nabla \cdot v_m C_m + q_m C_{m/i/p} + \bar{v}_{m/f} C_{m/f} = \frac{\partial(\phi SC)_m}{\partial t}, \text{ for the matrix,} \quad 4.10b$$

where $C_{m/f}$ is the upstream concentration of solute in kg/m^3 in either the fracture or the matrix depending on the direction of the solvent flow (see Section 4.4.1 for the computation of the

upstream weighting factor), and $\bar{\tau}$ is the volumetric exchange rate, $m^3/s/m^3$ of solvent mass transfer between the fracture and the matrix.

Transfers at fracture/matrix interfaces are dominated by a nearly steady-state Darcian flow represented by $\tau_{wm/f}$. The magnitude of these flow terms and their idealization are controlled and provided by the fluid flow code as volumetric rates.

The fracture /matrix transfer function τ for the brine could be evaluated in the fluid flow code by the following semi-steady-state formula:

$$\tau_w = \sigma k_m \left(\frac{k_r}{\mu} \right)_{m/f} [P_f - P_m - \gamma_{m/f} (D_f - D_m)], \quad 4.11$$

where

- D depth [m],
- k permeability [m^2],
- k_r relative permeability [dimensionless],
- P pressure [Pa],
- γ static pressure gradient [Pa/m],
- μ viscosity [Pa/s],
- σ shape factor [m^{-2}].

Subscripts

- f = fracture
- m = matrix
- w = brine

4.3.3 Diffusive and Dispersive Transports

Advection represents the transport of solute as it is carried along by the motion of the fluid. Diffusion and dispersion represent transport of the solute due to its motion (or apparent motion) through the fluid. Molecular diffusion results from thermal motion of solute molecules. In porous media, the diffusion coefficient (sometimes called the apparent molecular diffusion coefficient) is calculated from the free-water molecular diffusion coefficient D_m^* adjusted for the tortuous path and the water content (saturation) of the transported phase of interest. Mechanical dispersion is the mixing process resulting from the fluctuation in the ground-water velocity due to the heterogeneity of the porous media. Thus, if a sharp contaminant front is introduced into a porous-medium flow, the front will rapidly smeared as though an efficient diffusion mechanism were at work, whereas an abrupt interface would be expected to persist if only advective transport was at work. Among the processes that lead to thickening and distortion of a hypothetical front are sorption-desorption processes, molecular diffusion, and mechanical mixing

(Bear, 1988). The last two are usually combined into a single effect called hydrodynamic dispersion.

In practical applications of solute-transport theory, it is customary to account for both the molecular and mechanical components of hydrodynamic dispersion. NUTS utilizes Fick's law to describe the dispersive contribution to mass transfer in both the fracture and matrix components of the flow. In vector notation, it is written:

$$F_D = \phi SK \nabla C, \quad 4.12$$

where F_D is the dispersion flux. Generally (but **not** in CCA calculations), NUTS treats the material dispersivity as a 2nd-rank tensor, with the additional assumption that the porous transport medium is isotropic. Bear (1988) has shown that this assumption is equivalent to representing the dispersivity tensor as a positive-definite, symmetric, 2nd-rank tensor. The most general dispersivity tensor would be the one shown in component form by the following matrix:

$$K = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix}. \quad 4.13$$

However, because the dispersion tensor is assumed to be symmetric, six of its coefficients are related as follows: $K_{xy} = K_{yx}$, $K_{xz} = K_{zx}$, and $K_{yz} = K_{zy}$, or in indicial notation,

$$K_{ij} = \frac{D_m^*}{\tau^*} \delta_{ij} + \frac{\alpha_T}{\phi S} |v| \delta_{ij} + \frac{v_i v_j}{\phi S |v|} (\alpha_L - \alpha_T), \quad 4.14$$

where nomenclature is defined in the listing below. If the chosen coordinate system happens to coincide with the principal axes of dispersion, all the off-diagonal values will become zero in the tensor matrix. In that coordinate system, the dispersion coefficient in the x-direction for the fracture flow and the matrix flow become (Bear, 1993):

$$K_{fx} = \left\{ \frac{D_m^*}{\tau_f^*} + \frac{1}{\phi_f S_f |\bar{v}_f|} \left[\alpha_{Lf} v_{fx}^2 + \alpha_{Tf} (v_{fy}^2 + v_{fz}^2) \right] \right\}, \quad 4.15$$

and

$$K_{mx} = \left\{ \frac{D_m^*}{\tau_m^*} + \frac{1}{\phi_m S_m |\bar{v}_m|} \left[\alpha_{Lm} v_{mx}^2 + \alpha_{Tm} (v_{my}^2 + v_{mz}^2) \right] \right\}, \quad 4.16$$

where the terms between braces are the x-component of the dispersivity tensor within (i) the fracture flow and (ii) the matrix flow, respectively, and where,

- D_m^* = free water molecular diffusion coefficient [m^2/s],
 τ^* = tortuosity⁺ [dimensionless],
 α_L = longitudinal dispersivity [m],
 α_T = transverse dispersivity [m],
 δ_{ij} = kronecker delta [dimensionless],

and the velocity vector \vec{v} is given by:

$$|\vec{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}. \quad 4.17$$

where v_x , v_y , v_z are the fluid flow velocities [m/s] in x , y , z directions, respectively. Similar formulations apply for the y and z components of the dispersivity tensor. The present version of NUTS applies the above formulation to diagonal components of the dispersivity tensor and ignores off-diagonal contributions.

In addition to dispersion in the fracture and matrix continua, Fick's law is used to represent the hydrodynamic dispersion transport between the fracture and the matrix, and is written as:

$$F_{Dmf} = \phi_m S_m K_{mf} (C_f - C_m) \quad 4.18$$

Bear (1993) gives the dispersion coefficient for fracture/matrix flow as:

$$K_{wmf} = \sigma \left\{ \frac{D_m^*}{\tau_m^*} + \alpha_m \frac{|v_{mf}|}{\phi_m S_m} \right\}, \quad 4.19$$

where

v_{mf} is the velocity of liquid transfer [m/s], and σ is a shape factor.

Knowing that the fracture aperture is very small compared with the matrix block dimensions, fracture/matrix dispersion in Equation 4.19 is mainly controlled by matrix properties.

The shape factor σ is defined, in general, as

$$\sigma = \sum_s \frac{A_m}{V_m d}, \quad 4.20$$

where

⁺ Throughout this document, tortuosity is defined as the ratio of the square of the tortuous path length to the square of the mean path length. This definition results in tortuosities greater than or equal to unity. However, the reader should bear in mind that some formulations found in the literature use the inverse of the definition, leading to tortuosities less than or equal unity.

S is all the faces in the matrix block subjected to flow,
 A_m is the surface area of the face subjected to flow [m^2],
 V_m is the volume of the matrix block [m^3], and
 d is the distance between the face and the center of the matrix block [m].

Sigma plays the role of Δn in $\Delta C/\Delta n$, which is the normal gradient to an interfacial surface. For a rectangular parallelepiped matrix block, the shape factor is calculated (Kazemi et al., 1976) to be,

$$\sigma = 4 \left(\frac{1}{L_x^2} + \frac{1}{L_y^2} + \frac{1}{L_z^2} \right), \quad 4.21$$

where L_x , L_y , and L_z are the block dimensions in x, y, and z directions.

Collecting intermediate results, if Equations 4.12 and 4.18 are added to Equations 4.10, the resulting equations will be

$$\nabla \cdot \phi_f S_f K_f \nabla C_f - \nabla \cdot v_f C_f + q_f C_{f/p} - \bar{\tau}_{m/f} C_{m/f} - \phi_m S_m K_{m/f} (C_f - C_m) = \frac{\partial(\phi S C)_f}{\partial t} \quad 4.22$$

for the fracture, and

$$\nabla \cdot \phi_m S_m K_m \nabla C_m - \nabla \cdot v_m C_m + q_m C_{m/p} + \bar{\tau}_{m/f} C_{m/f} + \phi_m S_m K_{m/f} (C_f - C_m) = \frac{\partial(\phi S C)_m}{\partial t} \quad 4.23$$

for the matrix.

4.3.4 Asymmetric Dispersion

The mechanical-dispersion expression introduced in the transport equation is little more than a global expression that endeavors to compensate for our lack of detailed information regarding the microscopic nature of the heterogeneities inherent in porous media. However, examination of Equations 4.15 and 4.16 suggests the effective (diagonal) dispersion coefficient is always positive, otherwise the equation is mathematically unstable. Therefore, mechanical dispersion always follow and enhance the trends established by molecular diffusion. It is reasonable to assume molecular diffusion follows Fick's law because of the random thermal motion of molecules. Therefore, molecular diffusion can expand symmetrically in all directions providing only that there exists a concentration gradient. This argument, if applied to the mechanical dispersion, will lead a well-known problem in which dispersion is directed opposite to the direction of the flow. Opposite-direction dispersion is physically incorrect and is associated with *a priori* ignorance of the flow direction (i.e., v^2 is always positive). In order to overcome this problem, NUTS has a module for Asymmetric dispersion that allows the dispersion to go only in

the direction of the flow regardless of method used to discretize the transport equation⁺. Because the method used is conservative, the only effect this treatment has on the transport, is shifting the centroid of the mass further downstream. In other words, the movement of the transport front will be little faster in the case of Asymmetric dispersion, as is depicted in Figure 4.3.

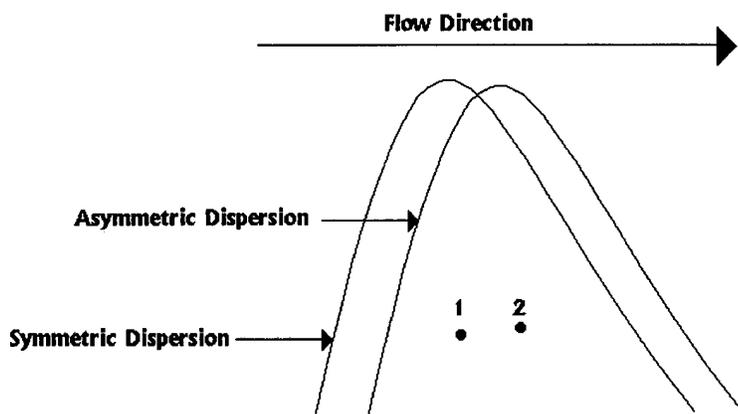


Figure 4.3 Symmetric versus Asymmetric Dispersion Fronts.

Asymmetric Dispersion is implemented numerically by computing a binary (0,1) parameter, κ as follows:

$$\kappa = \begin{cases} 1 & \text{if the concentration gradient is in the same direction as the flow gradient} \\ 0 & \text{otherwise} \end{cases}$$

The parameter κ is then used as a multiplier only to the dispersion part of the transport equation (the molecular diffusion part is symmetric). This manipulation is performed while preparing the diagonal entries of the numerical matrix to assure mass conservation of the system.

4.3.5 Sorption-Desorption

Sorption refers to the ability of certain solids to extract substances preferentially from solution and deposit them on their surfaces. The solution can be either gaseous or liquid. Two general

⁺ Dispersivity coefficients are not constants and an alternative way for asymmetric dispersion treatment is to deduce them from the time rate of change of the "moments-of-inertia" tensor. They will be zero at the source and will become constants only after several correlation lengths of the heterogeneity of the porous media have been sampled.

types of sorption are treated; physical sorption and chemical sorption. Physical sorption is a reversible process and arises on account of van der Waal intermolecular forces on the sorbing solid and the sorbed substance. In porous media, the sorbed substance may not react with the sorbing solid, but it can penetrate it via interstices, if a wettability condition is in effect. Chemical sorption is a result of chemical interaction between the sorbate and the sorbent. The chemical bond associated with this attraction is generally much greater than that found in the physical sorption. Ion exchange is sometimes classified as a kind of the chemical sorption.

Desorption is the process in which the solid phase releases some of the sorbed material back into solution. Sorption-desorption as a process depends on, pressure, temperature, and the chemical composition of the sorbate and the sorbent. It is usually described by an empirical relationship relating the concentration of the solute and the concentration of sorbate at a constant temperature, and is called an "isotherm." If a first-order kinetic model is used to describe the sorption-desorption process, one can write the following reaction-rate equation,

$$\frac{\partial R}{\partial t} = k_1 C - k_2 C_s, \quad 4.24$$

where R is the net rate of reaction, k_1 is the sorption reaction constant, k_2 is the desorption reaction constant, C is the concentration of sorbent in kg per kg of fluid, C_s is the concentration of sorbate in kg/kg dry porous media. In the limit of sorptive equilibrium,

$$\frac{\partial R}{\partial t} \rightarrow 0 \quad \text{as } t \rightarrow \infty, \text{ and}$$

$$\frac{k_1}{k_2} = \chi = \frac{C_s}{C} \Rightarrow C_s = \chi C, \quad 4.25$$

where χ is a parameter that depends on the shape of the equilibrium isotherm and the equilibrium conditions. For a linear isotherm $\chi = k_d$, the linear sorption equilibrium constant. In porous media, the mass rate of sorption per unit volume, F_s , in kg/s/m³ can therefore be determined from

$$F_s = \frac{\partial}{\partial t} (1 - \phi) \rho_s C_s, \quad 4.26$$

where

$$\rho_s = \text{rock grain density kg/m}^3.$$

The mass transfer between a solute and the solid surface that bounds it is controlled by the chemical properties of the solid surface, the liquid transport medium, and the solute. Sorption manifests itself in the transport equation as a retarding factor and is expressed in NUTS by various empirical relations called "isotherms." As the name suggests, these isotherms are

families of empirical graphs of sorbate concentration versus solute concentration at constant temperature. NUTS accommodates three equilibrium isotherms. They are the:

1. Linear sorption isotherm in which adsorbate concentration is related to solute concentration by:

$$C_s^{n+1} = k_d C_w^{n+1}, \quad 4.27$$

where

$n+1$ represents the implicitness of the concentration (current time step), and k_d is the equilibrium partition coefficient.

2. Freundlich sorption isotherm in which sorbate concentration is related to solute concentration by:

$$C_s^{n+1} = x_1 (C_w^{n+1})^{\frac{1}{x_2}} = x_1 (C_w^n)^{\frac{1-x_2}{x_2}} C_w^{n+1} = \chi C_w^{n+1} \quad 4.28$$

where n refers to the former time step, $\chi = x_1 (C_w^n)^{\frac{1-x_2}{x_2}}$, and x_1 and x_2 are the Freundlich distribution coefficient and the Freundlich coefficient, respectively.

3. Langmuir sorption isotherm, which, together with the linear isotherm, is one of the most common isotherms used in the literature today, and is defined by:

$$C_s^{n+1} = \chi_1 C_w^{n+1} \quad \text{where } \chi_1 = x_1 \left(1 - \frac{x_2 C_w^n}{1 + x_2 C_w^n} \right) \quad 4.29$$

where the empirical parameters x_1 and x_2 are the Langmuir distribution coefficient and the Langmuir coefficient, respectively.

NUTS permits sorption in the fractures and the matrix, but it has the user-controlled ability to bypass sorption in either flow field. Moreover, it is possible to assign any specified isotope as sorbable or nonsorbable, regardless of the flow regime.

Sorption and desorption are **not** included in the CCA calculation. However, sorption onto colloidal materials suspended within the flow is included, and is treated exactly as dissolution is treated, i.e., the effective maximum solubility coefficients are increased so as to represent the sum of all maximum allowable concentrations associated with (i) dissolution and (ii) all allowable forms of colloidal mobilization.

4.3.6 Decay

Radioisotopes are naturally unstable and spontaneously emit clusters of particles equivalent to helium atoms until, eventually, they arrive at stable configurations. Thus, the mass of daughter products is always lower than that of the parent radioisotope. Rates of disintegration vary over a wide range. Some are so slow that the parent radioisotopes are regarded as stable for all practical purposes. The mode of the observed disintegration can be classified into two categories: 1) disintegration associated with a change in atomic mass, such as alpha particle emission and spontaneous fission, and 2) disintegration without a change in atomic mass, such as the emission of positive or negative beta particles and the capture by the nucleus of an electron in the $n=1$ shell. Regardless of the mode of disintegration, the rate of the process is governed by a first-order rate law. Thus, the differential equation governing the mass of radioisotope residing at the top of a decay chain is of the form

$$\frac{\partial N}{\partial t} = -\lambda N, \quad 4.30$$

where λ , the proportionality constant, is called the decay constant, and N is the number of atoms present at a given time, t . The integration of Equations like 4.30 leads to solutions of the form

$$N = N_0 e^{-\lambda t}, \quad 4.31$$

where N is the number of the atoms at any time, t , and N_0 is the number of the atoms at zero time. The decay constant is related to the half life by the following relationship:

$$\lambda = \frac{\ln 2}{t_{1/2}}, \quad 4.32$$

where $t_{1/2}$ is the half life. The decay constant and, therefore, the half life are fixed physical characteristics of each unstable radioisotope and are independent of its state of chemical combination, the presence of electric and magnetic fields, the temperature, and the pressure. Because there are Avogadro's number of atoms in each mole of a substance and a mass equivalent to the molecular weight in each mole, Equations 4.30 and 4.31 can be written in terms of mass M as:

$$\frac{\partial M}{\partial t} = -\lambda M, \quad 4.33$$

and

$$M = M_0 e^{-\lambda t}, \quad 4.34$$

where

M = mass of the solute at any time t (kg), and

M_0 = mass of the solute at the initial time t_0 (kg) .

In porous media, the total mass of a radioisotope is distributed between the dissolved mass and the adsorbed mass. Therefore, from Equation 4.33, the rate at which mass is transported by decay, F_{decay} , in $kg/s/m^3$, can be written in terms of concentration as

$$F_{Decay} = -\lambda(\phi SC + (1-\phi)\rho_s C_s) \quad 4.35$$

Equation 4.35 represents a sink (i.e., loss of mass) in which the first and the second terms in the bracket refer to the decay of the dissolved and sorbed mass of the isotope, respectively. If there is a parent(s) for the substance in question, the decay of the parent(s) will act as a source in the transport equation and have a value equal to the rate of growth of the daughter. The rate of growth due to parent(s) decay, F_g in $kg/s/m^3$ is

$$F_g = \sum_j (\phi SC + (1-\phi)\rho_s C_s)_{pj} \lambda_{pj} \quad 4.36$$

where J is the total number of parent substances and p_j is a parent index.

4.3.7 The Overall Mass-Transport Equation

Amassing all the source/sink effects associated with sorption and radioactive decay (Equations 4.26, 4.35, and 4.36) within the continuity equation thus far derived (Equations 4.22 and 4.23), the overall mass-transport equation applicable to either hydrological regime is:

$$\begin{aligned} \nabla \cdot \phi SK \nabla C - \nabla \cdot vC + C^* q \pm \bar{\tau} C_{m/f} \pm \phi_m S_m K_{m/f} (C_f - C_m) = \\ \frac{\partial}{\partial t} (\phi SC + (1-\phi)\rho_s C_s) + (\phi SC + (1-\phi)\rho_s C_s) \lambda - \\ \sum_{j=1}^J (\phi SC + (1-\phi)\rho_s C_s)_{pj} \lambda_{pj} \end{aligned} \quad 4.37$$

In fractured porous media, if the total porosity, ϕ_t is defined as $\phi_t = \phi_m + \phi_f$, where ϕ_m is the matrix porosity and ϕ_f is the fracture porosity, then the partial differential equations representing the transport in the two geophysical domains (fractured, and porous media) can be written as follows* :

* Please note that the adsorbed concentration (C_s) is defined as kg solute per kg rock weight in the entire grid block. This is why, ϕ_t is used in stead of ϕ_m or ϕ_f in the sorption term of Equations 4.37 to 4.39 (Bear, 1993).

Fracture Equation:

$$\begin{aligned} & \nabla \cdot \phi_{wf} S_{wf} K_{wf} \nabla C_{wf} - \nabla \cdot v_{wf} C_{wf} + C_{wf}^* q_{wf} - \bar{\tau} C_{wm/f} - \phi_{wm} S_{wm} K_{wm/f} (C_{wf} - C_{wm}) = \\ & \frac{\partial}{\partial t} (\phi_{wf} S_{wf} C_{wf} + (1 - \phi_t) \rho_s C_{sf}) + (\phi_{wf} S_{wf} C_{wf} + (1 - \phi_t) \rho_s C_{sf}) \lambda - \\ & \sum_{j=1}^J (\phi_{wf} S_{wf} C_{wf} + (1 - \phi_t) \rho_s C_{sf})_{pj} \lambda_{pj} \end{aligned} \quad 4.38$$

Matrix Equation:

$$\begin{aligned} & \nabla \cdot \phi_{wm} S_{wm} K_{wm} \nabla C_{wm} - \nabla \cdot v_{wm} C_{wm} + C_{wm}^* q_{wm} + \bar{\tau} C_{wm/f} + \phi_{wm} S_{wm} K_{wm/f} (C_{wf} - C_{wm}) = \\ & \frac{\partial}{\partial t} (\phi_{wm} S_{wm} C_{wm} + (1 - \phi_t) \rho_s C_{sm}) + (\phi_{wm} S_{wm} C_{wm} + (1 - \phi_t) \rho_s C_{sm}) \lambda - \\ & \sum_{j=1}^J (\phi_{wm} S_{wm} C_{wm} + (1 - \phi_t) \rho_s C_{sm})_{pj} \lambda_{pj} \end{aligned} \quad 4.39$$

where f =fracture, m =matrix, w = brine, s =solid, t =total, m/f = fracture/matrix, and $C^*=C_{i,p}$.

4.3.8 Initial and Boundary Conditions

The system of transport equations (fracture and matrix) given above is part of a boundary-value problem. Therefore, some sort of (i) initialization condition is required throughout the domain, and (ii) boundary conditions for all time are required for the dependent variable $C(x,y,z,t)$. The initial condition of the matrix equation in the three-dimensional domain $(0,X; 0,Y; \text{ and } 0,Z)$ is

$$C_{wm}(x, y, z, 0) = f_m(x, y, z) \quad 0 \leq x \leq X, \quad 0 \leq y \leq Y, \quad 0 \leq z \leq Z. \quad 4.40$$

Similarly, the fracture is initialized by

$$C_{wf}(x, y, z, 0) = f_f(x, y, z) \quad 0 \leq x \leq X, \quad 0 \leq y \leq Y, \quad 0 \leq z \leq Z. \quad 4.41$$

To assure that the system is in initial equilibrium, there must be no pressure or concentration gradients between the two continua, which leads to:

$$f_f(x, y, z) = f_m(x, y, z), \quad 4.42$$

which must hold along their common boundary. The boundary conditions in the fracture and/or the matrix may be either Dirichlet (first kind) or Neumann boundary (second kind) conditions. If they are Dirichlet conditions, the dependent variable itself is specified at the boundaries, as follows:

$$\begin{aligned}
 C(0, y, z, t) &= g_{1x}(t) & t \geq 0, \\
 C(X, y, z, t) &= g_{1X}(t) & t \geq 0, \\
 C(x, 0, z, t) &= g_{1y}(t) & t \geq 0, \\
 C(x, Y, z, t) &= g_{1Y}(t) & t \geq 0, \\
 C(x, y, 0, t) &= g_{1z}(t) & t \geq 0, \\
 C(x, y, Z, t) &= g_{1Z}(t) & t \geq 0.
 \end{aligned}
 \tag{4.43}$$

If they are Neumann boundary conditions, the normal derivatives of the dependent variable are specified at the boundaries, as follows

$$\begin{aligned}
 \frac{\partial}{\partial x}[C(0, y, z, t)] &= g_{2x}(t) & t \geq 0, \\
 \frac{\partial}{\partial x}[C(X, y, z, t)] &= g_{2X}(t) & t \geq 0, \\
 \frac{\partial}{\partial y}[C(x, 0, z, t)] &= g_{2y}(t) & t \geq 0, \\
 \frac{\partial}{\partial y}[C(x, Y, z, t)] &= g_{2Y}(t) & t \geq 0, \\
 \frac{\partial}{\partial z}[C(x, y, 0, t)] &= g_{2z}(t) & t \geq 0, \\
 \frac{\partial}{\partial z}[C(x, y, Z, t)] &= g_{2Z}(t) & t \geq 0.
 \end{aligned}
 \tag{4.44}$$

These conditions are sometimes specified internally in the domain and called point source/sink.

In NUTS applications, Dirichlet boundary conditions are input controlled and can be specified at both the outer boundaries or internally at sources. Neumann boundary conditions, on the other hand are restricted to domains with either vertical or horizontal boundaries (no curved boundaries), and are not input controlled and used by default to specify no-diffusive-dispersive transport at the outer boundaries of the simulation domain, i.e., $D_{xx} \frac{\partial C}{\partial x} = 0$, $D_{yy} \frac{\partial C}{\partial y} = 0$, and

$$D_{zz} \frac{\partial C}{\partial z} = 0, \text{ where } D_{xx}, D_{yy}, \text{ and } D_{zz} \text{ are the diagonal component of the dispersion tensor.}$$

Therefore, the function $g_2(t)$ in Equation 4.44 is internally set to zero (concentration gradient of values other than zero is not considered in the boundary treatment of NUTS).

4.4 Discretization Methods

The partial differential equations and boundary/initial conditions derived in Section 4.3 and representing transport in a dual-porosity, dual-permeability medium can be solved analytically

for only the very simplest of problems, problems that bear virtually no relationship to real-world ground-water flows like those of the WIPP. To treat real-world problems, it is necessary to turn to numerical techniques, of which several are available.

The most common numerical methods used in modern hydrology are the finite-difference and finite-element (variational) methods, and particularly the Galerkin method. NUTS uses the finite-difference method to transform the time-space continuum problem into a solvable discrete mathematical model. An *algebraic* equation approximating the partial differential equation at each point of the domain of interest is derived. The derivational methods are described in this Section.

The first important step of the derivation is to define a convenient mesh (grid) over the entire spatial domain of interest.

For simplicity, consider the discrete representation of a continuous independent variable, x , between 0 and 1, as shown in Figure 4.4.

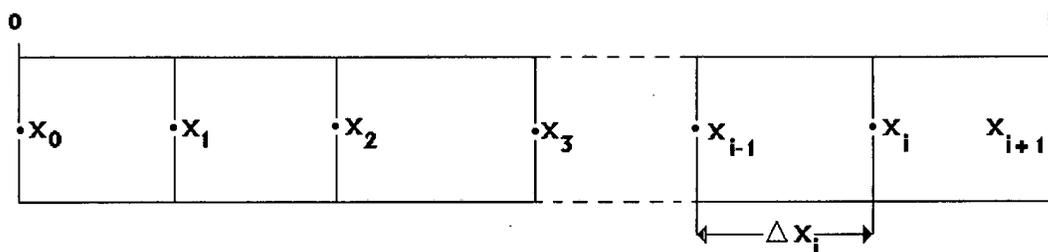


Figure 4.4 Point-Center Grid in Finite-Difference Approximation.

As is shown, the continuous domain is replaced by a set of discrete points whose spacing may be equal or unequal, and the discrete variable x is defined at those points. Clearly, the smaller the spacing the greater the resolution, but the more complex the resulting algebraic problem. Optimum spacings normally depend on the type of problem, the solution requirements, and the available resources. In Figure 4.4, the discrete values of x are denoted by x_i , $i = 0, 1, 2, \dots$. The algebra is a little easier if the intervals are of equal width Δx , where Δx is called the finite difference.

$$x_i = i\Delta x . \tag{4.45}$$

In that case, neighboring points are related to one another by

$$x_{i+1} = x_i + \Delta x , \tag{4.46}$$

and Δx is the distance between successive grid points.

$$x_{i-1} = x_i - \Delta x . \tag{4.47}$$

The function $C(x)$, which we think of as one of the unknown dependent variables of our problem, is normally defined on the continuous interval $0,1$. We will replace it by its finite-difference approximation C_i , which is defined only at the points $x = x_i$, $i = 0, 1, \dots, I+1$, and is represented by

$$C(x_i) = C_i . \tag{4.48}$$

Since our governing equations are partial differential equations, it is essential to have finite-difference expressions for the derivatives of an unknown function, as well as the function itself. The derivative of the function C_i can be determined by Taylor's expansion C_i in the neighborhood of the point x . Thus for the function $C(x + \Delta x)$,

$$C(x + \Delta x) = C(x) + \frac{\partial C}{\partial x} \Delta x + \frac{\partial^2 C}{\partial x^2} \frac{\Delta x^2}{2!} + \frac{\partial^3 C}{\partial x^3} \frac{\Delta x^3}{3!} + \dots + R_n , \tag{4.49}$$

where the derivatives are evaluated at x and the remainder term R_n is given by

$$R_n = \frac{1}{n!} \left(\Delta x \frac{\partial}{\partial x} \right)^n C(x + \zeta \Delta x) , \tag{4.50}$$

and ζ lies between 0 and 1, so the argument of C lies between x and $x + \Delta x$. Equation 4.50 is sometimes written in the form

$$R_n = O[|\Delta x|^n] , \tag{4.51}$$

in which $|\Delta x|^n$ refers to the order of the truncation error in the Taylor series, and means there exists a positive constant M such that

$$|R_n| \leq M(|\Delta x|^n) \text{ as } \Delta x \rightarrow 0 . \tag{4.52}$$

If we use a more compact notation, Equation 4.49 may be written as

$$C_{i+1} = C_i + \Delta x C_{ix} + \frac{(\Delta x)^2}{2!} C_{ixx} + \frac{(\Delta x)^3}{3!} C_{ixxx} + \dots + R_n \tag{4.53}$$

Similarly, the function $C(x - \Delta x)$ can be written as

$$C_{i-1} = C_i - \Delta x C_{ix} + \frac{(\Delta x)^2}{2!} C_{ixx} - \frac{(\Delta x)^3}{3!} C_{ixxx} + \dots + R_n \tag{4.54}$$

where $C_x = \frac{\partial C}{\partial x}$, $C_{xx} = \frac{\partial^2 C}{\partial x^2}$, etc., and all derivatives are evaluated at the point i . By adding and then subtracting Equations 4.53 and 4.54, one can arrive at finite-difference approximations for the first two derivatives of C_i , as follows:

$$C_{ix} = \frac{C_{i+1} - C_i}{\Delta x} + O(\Delta x), \quad 4.55$$

$$C_{ix} = \frac{C_i - C_{i-1}}{\Delta x} + O(\Delta x), \quad 4.56$$

$$C_{ix} = \frac{C_{i+1} - C_{i-1}}{2\Delta x} + O[(\Delta x)^2], \quad 4.57$$

$$C_{ixx} = \frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta x)^2} + O[(\Delta x)^2] \quad 4.58$$

Equations 4.55, 4.56, 4.57 are known, respectively, as forward-, backward-, and central-difference formulae for the first derivative of C_i . Equation 4.58 gives a finite-difference approximation for the second derivative of C_i . All these formulae arise from truncated infinite series. Thus, truncation errors are involved. Truncation errors vary in these formulations, being first-order in the forward and backward difference formulae, and second-order in the central-difference and the second derivative.

In NUTS's discretization of the transport equation, a block-centered grid is used, which is consistent with BRAGFLO's discretization. With a block-centered grid, the location of the advective component of the transport is evaluated not at grid points, but at the interior points mid way between grid points. This formulation leads to introduction of new points at which the discretized variable is defined, namely, at $C_{i+1/2}$ and $C_{i-1/2}$, in addition to the discrete points i where the concentration C_i was defined earlier (see Figure 4.5).

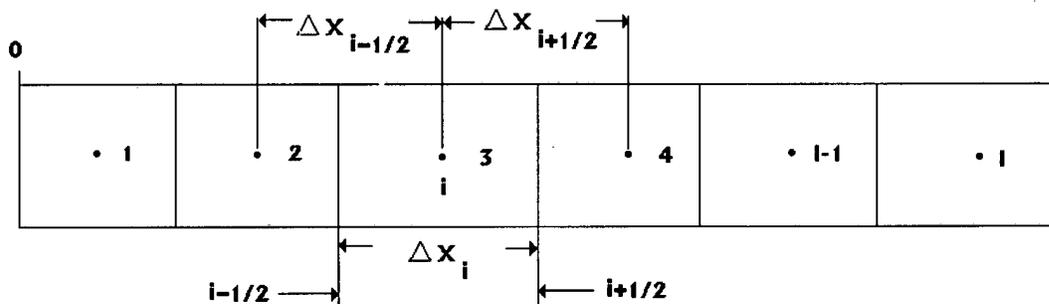


Figure 4.5 Block-Center Grid in Finite-Difference Approximation.

in Figure 4.5, $\Delta x_{i+\frac{1}{2}} = \frac{\Delta x_i + \Delta x_{i+1}}{2}$ is the distance between the neighboring points i and $i+1$.

Various methods are used to approximate the value of the function at these interfacial points, each method having a characteristic accuracy. Among the most common methods are:

1. Mid-point weighting approximation. In this second-order accurate method, the concentration at the interface is approximated as follows:

$$C_{i-\frac{1}{2}} = (0.5 + \mu_1)C_i + (0.5 - \mu_1)C_{i-1}, \quad 4.59$$

and

$$C_{i+\frac{1}{2}} = (0.5 + \mu_2)C_{i+1} + (0.5 - \mu_2)C_i, \quad 4.60$$

where

$$\mu_1 = \frac{1}{4} \frac{\Delta x_{i-1} - \Delta x_i}{\Delta x_{i-1} + \Delta x_i}, \text{ and } \mu_2 = \frac{1}{4} \frac{\Delta x_i - \Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}. \quad 4.61$$

From Equation 4.61, it is obvious that μ_1 and μ_2 are both zero for a uniform grid, wherein $C_{i+\frac{1}{2}}$ and $C_{i-\frac{1}{2}}$ become simple averages of neighboring values.

2. One-point upstream weighting. The interfacial concentration in this first-order accurate method is approximated by the concentration of the upstream block as follows:

$$C_{i-\frac{1}{2}} = \begin{cases} C_i & \text{if the flow from } i \text{ to } i-1 \\ C_{i-1} & \text{if the flow from } i-1 \text{ to } i \end{cases}. \quad 4.62$$

Similarly

$$C_{i+\frac{1}{2}} = \begin{cases} C_i & \text{if the flow from } i \text{ to } i+1 \\ C_{i+1} & \text{if the flow from } i+1 \text{ to } i \end{cases}. \quad 4.63$$

3. Two-point upstream weighting. This method is second-order accurate. The interfacial concentration in the two-point upstream weighting is approximated by the concentration of the two grid blocks upstream of the interface as follows:

$$C_{i-\frac{1}{2}} = \left\{ \begin{array}{ll} (1 + \mu_1)C_i - \mu_1 C_{i+1} & \text{if the flow from } i \text{ to } i-1 \\ (1 + \mu_2)C_{i-1} - \mu_2 C_{i-2} & \text{if the flow from } i-1 \text{ to } i \\ \text{where } \mu_1 = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}} \text{ and } \mu_2 = \frac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_{i-2}} & \end{array} \right\}, \quad 4.64$$

and

$$C_{i+\frac{1}{2}} = \left\{ \begin{array}{ll} (1 + \mu_1)C_i - \mu_1 C_{i-1} & \text{if the flow from } i \text{ to } i+1 \\ (1 + \mu_2)C_{i+1} - \mu_2 C_{i+2} & \text{if the flow from } i+1 \text{ to } i \\ \text{where } \mu_1 = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}} \text{ and } \mu_2 = \frac{\Delta x_{i+1}}{\Delta x_{i+1} + \Delta x_{i+2}} & \end{array} \right\}. \quad 4.65$$

4. Leonard method. This third-order accurate method uses two grid blocks upstream and one grid block downstream. In this method and for a uniform grid, if the flow in Figure 4.5 is from the left to the right, then

$$C_{i-\frac{1}{2}} = \frac{2C_i + 5C_{i-1} - C_{i-2}}{6}, \quad 4.66$$

and

$$C_{i+\frac{1}{2}} = \frac{2C_{i+1} + 5C_i - C_{i-1}}{6}. \quad 4.67$$

5. Total-variation-diminishing (TVD) flux-limiter methods. In this class of methods, the flux, $F = vC$, and not the concentration is used as the independent variable. It starts with the average of F ,

$$F_{i+\frac{1}{2}} = f_i + \frac{\Delta x}{2} \frac{f_{i+1} - f_i}{\Delta x} \quad 4.68$$

which is written as the first two terms of a Taylor expansion. Then, the slope-term contribution is multiplied by a flux-limiter function, ϕ , so that

$$F_{i+\frac{1}{2}} = f_i + \phi(r_i)A_{i+\frac{1}{2}} \quad 4.69$$

where $A_{i+\frac{1}{2}}$ is the slope-term contribution and $\phi(r)$, the flux-limiter term, is chosen in a way that gives a value of approximately 1.0 in smooth regions of the concentration profile. In regions where oscillations could occur due to the unrestrained slope-term contributions, $\phi(r)$ is allowed to vary so as to eliminate spurious solutions. The variable r is defined as the ratio of successive slope terms, namely:

$$r_i = \frac{A_{i-\frac{1}{2}}}{A_{i+\frac{1}{2}}} \quad 4.70$$

Many satisfactory flux-limiter functions are available in the literature. A widely used example is the van Leer limiter, which is defined as:

$$\phi(r_i) = \frac{|r_i| + r_i}{1 + |r_i|} \quad 4.71$$

The flux limiter should be calculated to be TVD, which means the total variation in the concentration is non increasing. If the total variation in the concentration at a certain time is defined by:

$$TV(C^n) = \sum |C_{i+1}^n - C_i^n| \quad 4.72$$

then in the TVD,

$$TV(C^{n+1}) \leq TV(C^n) \quad 4.73$$

where $n+1$ refers to the next time step. The formulation for $F_{i-\frac{1}{2}}$ is similar. Using Equation 4.69, it is clear that TVD is a mid-point scheme, when $\phi(r)=1$; a one-point upstream weighting scheme, when $\phi(r)=0$; and a two-point upstream weighting scheme, when $\phi(r)=r$. Other constraints apply to the flux-limiter function, depending on the degree of implicitness in the solution, but they are beyond the scope of this manual.

Thus far, the discretized values of the function and its spatial derivatives have been introduced. For time dependent problems, which arise in WIPP PAs, the dependent variables, generically denoted C_i are also functions of time $C_i = C_i(t)$. Therefore, at each grid point x_i , (i) a continuous or (ii) a discrete solution in time must also be calculated. If the time derivative is not discretized, the system of governing equations can, in principle, be solved by any of the methods traditionally used to solve ordinary differential equations. Euler's method, and the Runge-Kutta method are popular choices. However, the more common approach is to discretize temporal derivatives as well as spatial derivatives. In finite-difference formulations, the time derivative is

approximated by one of the equations given above (Equation 4.55 to 4.58). A straightforward solution can be achieved by approximating the time derivative using the forward-difference method, as shown below:

$$\left(\frac{\partial C_i}{\partial t}\right)^n = \frac{C_i^{n+1} - C_i^n}{\Delta t}, \quad 4.74$$

where Δt refers to the time-step size, $n+1$ signifies the next time in the discrete sequence, and n is the present time. When the spatial distribution of the dependent variable C_i is discretized at the time level n , the system of equations can be solved explicitly. On the other hand, if the spatial distribution of the dependent variable C_i is discretized at the time level $n+1$, a system of I equations must be solved. This method is called the implicit method. The implicit and explicit methods can also be combined so as to form a third integration method. Which method of solution to choose is not wholly arbitrary. Questions of stability, convergence, consistency, and the order of the method must be considered. Consequently, a versatile code must provide for several alternatives.

NUTS Version 2.05 accommodates the first three weighting methods mentioned above, namely: one-point upstream, two-point upstream, and mid-point weighting. For temporal variabilities, NUTS's solution method can, in principle, vary from fully explicit to fully implicit. This spectrum of methods is cumbersome to describe. Therefore, *for illustration purposes, an exemplary case will be shown in detail, but then details will be omitted from the discretized equations that follow.* For the example, consider the hyperbolic, one-dimensional, purely-advective, partial differential equation given as Equation 4.75, in which the time derivative is discretized according to Equation 4.74:

$$(1-\theta)\left(-u\frac{\partial C}{\partial x}\right)^{n+1} + \theta\left(-u\frac{\partial C}{\partial x}\right)^n = \frac{C_i^{n+1} - C_i^n}{\Delta t}. \quad 4.75$$

In Equation 4.75: if $\theta = 1$, the temporal scheme is explicit; if $\theta = 0$, the temporal scheme is implicit; and if $\theta = 1/2$, the scheme is mixed, and this particular scheme is known as the Crank-Nicolson method.

4.4.1 Discrete Analogues of the Conservation of Mass Equations

Let us now consider the partial differential equations that govern transport in the fracture and the matrix continua. In this section, these equations will be discretized fully implicitly in time and at the grid block interfaces in space, whereas in the subsequent sections, the linear system for different numerical implementations will be developed.

For the fracture, the discretized mass-transport equation in the brine is:

$$\begin{aligned}
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i+\frac{1}{2}} (C_{wfi+1}^{n+1} - C_{wfi}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i-\frac{1}{2}} (C_{wfi}^{n+1} - C_{wfi-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j+\frac{1}{2}} (C_{wff+1}^{n+1} - C_{wff}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j-\frac{1}{2}} (C_{wff}^{n+1} - C_{wff-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k+\frac{1}{2}} (C_{wfk+1}^{n+1} - C_{wfk}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k-\frac{1}{2}} (C_{wfk}^{n+1} - C_{wfk-1}^{n+1}) + \\
 & q_{wfi+\frac{1}{2}}^{n+1} C_{wfi+\frac{1}{2}}^{n+1} - q_{wfi-\frac{1}{2}}^{n+1} C_{wfi-\frac{1}{2}}^{n+1} + q_{wff+\frac{1}{2}}^{n+1} C_{wff+\frac{1}{2}}^{n+1} - q_{wff-\frac{1}{2}}^{n+1} C_{wff-\frac{1}{2}}^{n+1} + q_{wfk+\frac{1}{2}}^{n+1} C_{wfk+\frac{1}{2}}^{n+1} - q_{wfk-\frac{1}{2}}^{n+1} C_{wfk-\frac{1}{2}}^{n+1} \\
 & + C_{wfi}^{*n+1} Q_{wfi}^{n+1} - \tau_{wm/fi}^{n+1} (\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1}) - \\
 & (\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi/fi}^{n+1}) (C_{wfi}^{n+1} - C_{wmi}^{n+1}) = \tag{4.76} \\
 & \frac{V_{Ri}}{\Delta t} \left[\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \}^{n+1} - \{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \}^n \right] + \\
 & V_{Ri} \left[\phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right]^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \} \lambda_l \right]^{n+1}.
 \end{aligned}$$

Similarly, for the matrix continuum, the discretized mass-transport equation in the brine is:

$$\begin{aligned}
 & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{i+\frac{1}{2}} (C_{wmi+1}^{n+1} - C_{wmi}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{i-\frac{1}{2}} (C_{wmi}^{n+1} - C_{wmi-1}^{n+1}) + \\
 & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{j+\frac{1}{2}} (C_{wmj+1}^{n+1} - C_{wmj}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{j-\frac{1}{2}} (C_{wmj}^{n+1} - C_{wmj-1}^{n+1}) + \\
 & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{k+\frac{1}{2}} (C_{wmk+1}^{n+1} - C_{wmk}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{k-\frac{1}{2}} (C_{wmk}^{n+1} - C_{wmk-1}^{n+1}) + \\
 & q_{wmi+\frac{1}{2}}^{n+1} C_{wmi+\frac{1}{2}}^{n+1} - q_{wmi-\frac{1}{2}}^{n+1} C_{wmi-\frac{1}{2}}^{n+1} + q_{wmj+\frac{1}{2}}^{n+1} C_{wmj+\frac{1}{2}}^{n+1} - q_{wmj-\frac{1}{2}}^{n+1} C_{wmj-\frac{1}{2}}^{n+1} + q_{wmk+\frac{1}{2}}^{n+1} C_{wmk+\frac{1}{2}}^{n+1} - q_{wmk-\frac{1}{2}}^{n+1} C_{wmk-\frac{1}{2}}^{n+1} \\
 & + C_{wmi}^{*n+1} Q_{wmi}^{n+1} + \tau_{wm/fi}^{n+1} (\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1}) + (\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi/fi}^{n+1}) (C_{wfi}^{n+1} - C_{wmi}^{n+1}) = \tag{4.77} \\
 & \frac{V_{Ri}}{\Delta t} \left[\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \}^{n+1} - \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \}^n \right] + \\
 & V_{Ri} \left[\phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right]^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \} \lambda_l \right]^{n+1}
 \end{aligned}$$

In Equations 4.76 and 4.77 above, the dispersion coefficients K_{wf}^{n+1} and K_{wm}^{n+1} have a unit of (m³/s), and are related to, but different from those in Equations 4.15 and 4.16 that have a unit of m²/s. For instance, the relationship between these coefficients at $i + \frac{1}{2}$ interface of the matrix continuum is:

$$K_{wm}^{n+1} (eq. 4.77) = \frac{A_i}{\Delta x_{i+\frac{1}{2}}} \{ K_{wm}^{n+1} (eq. 4.16) \}$$

where A_i is the area normal to the flow at the grid block i and $\Delta x_{i+\frac{1}{2}}$ is the distance between x_i and x_{i+1} .

4.4.2 One-Point Upstream-Winding Discretization of Transport Equations

Fracture Equation

$$\begin{aligned}
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i+\frac{1}{2}} (C_{wfi+1}^{n+1} - C_{wfi}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i-\frac{1}{2}} (C_{wfi}^{n+1} - C_{wfi-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j+\frac{1}{2}} (C_{wffj+1}^{n+1} - C_{wffj}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j-\frac{1}{2}} (C_{wffj}^{n+1} - C_{wffj-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k+\frac{1}{2}} (C_{wfk+1}^{n+1} - C_{wfk}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k-\frac{1}{2}} (C_{wfk}^{n+1} - C_{wfk-1}^{n+1}) + \\
 & q_{wfi+\frac{1}{2}}^{n+1} (\omega_{fi+1} C_{wfi}^{n+1} + (1 - \omega_{fi+1}) C_{wfi+1}^{n+1}) - q_{wfi-\frac{1}{2}}^{n+1} (\omega_{fi} C_{wfi-1}^{n+1} + (1 - \omega_{fi}) C_{wfi}^{n+1}) + \\
 & q_{wff+\frac{1}{2}}^{n+1} (\omega_{ffj+1} C_{wffj}^{n+1} + (1 - \omega_{ffj+1}) C_{wffj+1}^{n+1}) - q_{wff-\frac{1}{2}}^{n+1} (\omega_{ffj} C_{wffj-1}^{n+1} + (1 - \omega_{ffj}) C_{wffj}^{n+1}) + \\
 & q_{wfk+\frac{1}{2}}^{n+1} (\omega_{fk+1} C_{wfk}^{n+1} + (1 - \omega_{fk+1}) C_{wfk+1}^{n+1}) - q_{wfk-\frac{1}{2}}^{n+1} (\omega_{fk} C_{wfk-1}^{n+1} + (1 - \omega_{fk}) C_{wfk}^{n+1}) + \\
 & C_{wfi}^{n+1} Q_{wfi}^{n+1} - \tau_{wmi}^{n+1} (\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1}) - (\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi}^{n+1}) (C_{wfi}^{n+1} - C_{wmi}^{n+1}) = \quad 4.78 \\
 & \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \}^{n+1} - \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \}^n + \\
 & V_{Ri} \{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \{ \{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \} \lambda_l \}^{n+1}.
 \end{aligned}$$

By collecting similar terms, the above equation can be represented by the following linear system,

$$\begin{aligned}
 & A_f C_{wfk-1}^{n+1} + B_f C_{wffj-1}^{n+1} + C_f C_{wfi-1}^{n+1} + D_{1f} C_{wffjk}^{n+1} + D_{2f} C_{wmi}^{n+1} + \\
 & E_f C_{wfi+1}^{n+1} + F_f C_{wffj+1}^{n+1} + G_f C_{wfk+1}^{n+1} = R_f^{n+1} \quad 4.79
 \end{aligned}$$

where

$$\begin{aligned}
 A_f &= HD_{wfk-\frac{1}{2}}^{n+1} - \omega_{fk} q_{wfk-\frac{1}{2}}^{n+1}, \\
 B_f &= HD_{wff-\frac{1}{2}}^{n+1} - \omega_{ffj} q_{wff-\frac{1}{2}}^{n+1}, \\
 C_f &= HD_{wfi-\frac{1}{2}}^{n+1} - \omega_{fi} q_{wfi-\frac{1}{2}}^{n+1}, \\
 E_f &= HD_{wfi+\frac{1}{2}}^{n+1} - (1 - \omega_{fi+1}) q_{wfi+\frac{1}{2}}^{n+1},
 \end{aligned}$$

$$\begin{aligned}
 F_f &= HD_{w_{fj+\frac{1}{2}}}^{n+1} - (1 - \omega_{fj+1})q_{w_{fj+\frac{1}{2}}}^{n+1}, \\
 G_f &= HD_{w_{fk+\frac{1}{2}}}^{n+1} - (1 - \omega_{fk+1})q_{w_{fk+\frac{1}{2}}}^{n+1}, \\
 D_{1f} &= -HD_{w_{fk-\frac{1}{2}}}^{n+1} - HD_{w_{fj-\frac{1}{2}}}^{n+1} - HD_{w_{fi-\frac{1}{2}}}^{n+1} - HD_{w_{fk+\frac{1}{2}}}^{n+1} - HD_{w_{fj+\frac{1}{2}}}^{n+1} - HD_{w_{fi+\frac{1}{2}}}^{n+1} - \\
 & (1 - \omega_{fk})q_{w_{fk-\frac{1}{2}}}^{n+1} - (1 - \omega_{fj})q_{w_{fj-\frac{1}{2}}}^{n+1} - (1 - \omega_{fi})q_{w_{fi-\frac{1}{2}}}^{n+1} + \omega_{fk+1}q_{w_{fk+\frac{1}{2}}}^{n+1} + \omega_{fj+1}q_{w_{fj+\frac{1}{2}}}^{n+1} + \omega_{fi+1}q_{w_{fi+\frac{1}{2}}}^{n+1} - \\
 & \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{w_{fi}} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} + V_{Ri} \{ \phi_{fi} S_{w_{fi}} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} \lambda - \tau_{wm/fi}^{n+1} \omega_{m/fi} - \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1}, \\
 D_{2f}^{n+1} &= -\tau_{wm/fi}^{n+1} (1 - \omega_{m/fi}) + \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1},
 \end{aligned}$$

and

$$R_f^{n+1} = \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{w_{fi}} C_{w_{fi}} + (1 - \phi_t) \rho_s \xi_{fi} C_{w_{fi}} \}^n - C_{w_{fi}}^* Q_{w_{fi}}^{n+1} - V_{Ri} \sum_{l=1}^L \{ \{ \phi_{fi} S_{w_{fi}} C_{w_{fi}} + (1 - \phi_t) \rho_s \xi_{fi} C_{w_{fi}} \} \lambda_l \}^{n+1}.$$

In the above formulations HD is the hydrodynamic dispersion and is defined at every fluid-solid interface. The convention adopted in NUTS for calculating the HD is that the values of the porosity, saturation, dispersivities, and tortuosity are weighted upstream. Therefore, $HD_{w_{fi-1/2}}$ is defined as follows:

$$\begin{aligned}
 HD_{w_{fi-\frac{1}{2}}}^{n+1} &= \left(\omega_{fi} \phi_{fi} S_{w_{fi}} + (1 - \omega_{fi}) \phi_{fi-1} S_{w_{fi-1}} \right) \frac{\Delta y_i \Delta z_i}{\Delta x_i + \Delta x_{i-1}} \left\{ \frac{D_m^*}{\tau_f^*} + \right. \\
 & \left. \frac{1}{\omega_{fi} \phi_{fi} S_{w_{fi}} + (1 - \omega_{fi}) \phi_{fi-1} S_{w_{fi-1}}} \frac{1}{\sqrt{v_{xf}^2 + v_{yf}^2 + v_{zf}^2}} \left(\alpha_{L_f} v_{xf}^2 + \alpha_{T_f} (v_{yf}^2 + v_{zf}^2) \right) \right\}
 \end{aligned}$$

Similar equations can be written for the HD of the other interfaces. The values of ξ used in R_f and D_{1f} are sorption coefficients and depend mainly on the isotherm selected for use. Hence, ξ is equivalent to; (i) k_d in the linear adsorption isotherm, and (ii) χ in the Freundlich and Langmuir isotherms, as presented in Equations 4.27 to 4.29. In the above formulations, ω refers to the upstream weighting and has values of 0 and 1. The value of ω is calculated depending on the sign of the velocity. For two adjacent blocks $i-1$, and i the value of ω_i is:

$$\omega_i = \begin{cases} 1 & \text{if the flow from } i-1, j, k \text{ to } i, j, k \\ 0 & \text{otherwise} \end{cases}$$

similarly, the weighting parameters in the other direction are

$$\omega_j = \begin{cases} 1 & \text{if the flow from } i, j-1, k \text{ to } i, j, k \\ 0 & \text{otherwise} \end{cases}$$

and

$$\omega_k = \begin{cases} 1 & \text{if the flow from } i, j, k-1 \text{ to } i, j, k \\ 0 & \text{otherwise} \end{cases}$$

Similar convention is used in calculating the matrix/fracture weighting factor which is:

$$\omega_{m/f} = \begin{cases} 1 & \text{if flow from the fracture to the matrix} \\ 0 & \text{otherwise} \end{cases}$$

Matrix Equation:

$$\begin{aligned} & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{i+\frac{1}{2}} (C_{wmi+1}^{n+1} - C_{wmi}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{i-\frac{1}{2}} (C_{wmi}^{n+1} - C_{wmi-1}^{n+1}) + \\ & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{j+\frac{1}{2}} (C_{wmj+1}^{n+1} - C_{wmj}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{j-\frac{1}{2}} (C_{wmj}^{n+1} - C_{wmj-1}^{n+1}) + \\ & (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{k+\frac{1}{2}} (C_{wmk+1}^{n+1} - C_{wmk}^{n+1}) - (\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1})_{k-\frac{1}{2}} (C_{wmk}^{n+1} - C_{wmk-1}^{n+1}) + \\ & q_{wmi+\frac{1}{2}}^{n+1} (\omega_{mi+1} C_{wmi}^{n+1} + (1 - \omega_{mi+1}) C_{wmi+1}^{n+1}) - q_{wmi-\frac{1}{2}}^{n+1} (\omega_{mi} C_{wmi-1}^{n+1} + (1 - \omega_{mi}) C_{wmi}^{n+1}) + \\ & q_{wmj+\frac{1}{2}}^{n+1} (\omega_{mj+1} C_{wmj}^{n+1} + (1 - \omega_{mj+1}) C_{wmj+1}^{n+1}) - q_{wmj-\frac{1}{2}}^{n+1} (\omega_{mj} C_{wmj-1}^{n+1} + (1 - \omega_{mj}) C_{wmj}^{n+1}) + \\ & q_{wmk+\frac{1}{2}}^{n+1} (\omega_{mk+1} C_{wmk}^{n+1} + (1 - \omega_{mk+1}) C_{wmk+1}^{n+1}) - q_{wmk-\frac{1}{2}}^{n+1} (\omega_{mk} C_{wmk-1}^{n+1} + (1 - \omega_{mk}) C_{wmk}^{n+1}) + \\ & C_{wmi}^{n+1} Q_{wmi}^{n+1} + \tau_{wm/fi}^{n+1} (\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1}) + (\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi/fi}^{n+1}) (C_{wfi}^{n+1} - C_{wmi}^{n+1}) = \quad 4.80 \\ & \frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \}^{n+1} - \frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \}^n + \\ & V_{Ri} \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \{ \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smli} \} \lambda_l \}^{n+1}. \end{aligned}$$

Corresponding to the fracture-flow equation, the linear system is:

$$\begin{aligned} & A_m C_{wmk-l}^{n+1} + B_m C_{wmj-l}^{n+1} + C_m C_{wmi-l}^{n+1} + D_{1m} C_{wmijk}^{n+1} + D_{2m} C_{wifjk}^{n+1} + \\ & E_m C_{wmi+l}^{n+1} + F_m C_{wmj+l}^{n+1} + G_m C_{wmk+l}^{n+1} = R_m^{n+1} \quad 4.81 \end{aligned}$$

where

$$\begin{aligned}
 A_m &= HD_{wmk-\frac{1}{2}}^{n+1} - \omega_{mk} q_{wmk-\frac{1}{2}}^{n+1}, \\
 B_m &= HD_{wmj-\frac{1}{2}}^{n+1} - \omega_{mj} q_{wmj-\frac{1}{2}}^{n+1}, \\
 C_m &= HD_{wmi-\frac{1}{2}}^{n+1} - \omega_{wi} q_{wmi-\frac{1}{2}}^{n+1}, \\
 E_m &= HD_{wmi+\frac{1}{2}}^{n+1} + (1 - \omega_{mi+1}) q_{wmi+\frac{1}{2}}^{n+1}, \\
 F_m &= HD_{wmj+\frac{1}{2}}^{n+1} + (1 - \omega_{mj+1}) q_{wmj+\frac{1}{2}}^{n+1}, \\
 G_m &= HD_{wmk+\frac{1}{2}}^{n+1} + (1 - \omega_{mk+1}) q_{wmk+\frac{1}{2}}^{n+1},
 \end{aligned}$$

$$\begin{aligned}
 D_{1m}^{n+1} &= -HD_{wmk-\frac{1}{2}}^{n+1} - HD_{wmj-\frac{1}{2}}^{n+1} - HD_{wmi-\frac{1}{2}}^{n+1} - HD_{wmk+\frac{1}{2}}^{n+1} - HD_{wmj+\frac{1}{2}}^{n+1} - HD_{wmi+\frac{1}{2}}^{n+1} \\
 &+ (1 - \omega_{mk}) q_{wmk-\frac{1}{2}}^{n+1} - (1 - \omega_{mj}) q_{wmj-\frac{1}{2}}^{n+1} - (1 - \omega_{mi}) q_{wmi-\frac{1}{2}}^{n+1} + \omega_{mk+1} q_{wmk+\frac{1}{2}}^{n+1} + \omega_{mj+1} q_{wmj+\frac{1}{2}}^{n+1} + \omega_{mi+1} q_{wmi+\frac{1}{2}}^{n+1} - \\
 &\frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^{n+1} + V_{Ri} \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^{n+1} \lambda + \tau_{wm/fi}^{n+1} (1 - \omega_{mi/fi}) - \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1},
 \end{aligned}$$

$$D_{2m}^{n+1} = \tau_{wm/fi}^{n+1} \omega_{mi/fi} + \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1},$$

and

$$\begin{aligned}
 R_m^{n+1} &= \frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} C_{wmi} \}^n - C_{wmi}^* Q_{wmi}^{n+1} - \\
 &V_{Ri} \sum_{l=1}^L \{ \{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} C_{wmi} \} \lambda_l \}^{n+1}.
 \end{aligned}$$

In a similar fashion, the HD in the matrix at I-1/2 interface is defined as follows:

$$\begin{aligned}
 HD_{wmi-\frac{1}{2}}^{n+1} &= (\omega_{mi} \phi_{mi} S_{wmi} + (1 - \omega_{mi}) \phi_{mi-1} S_{wmi-1}) \frac{\Delta y_i \Delta z_i}{\Delta x_i + \Delta x_{i-1}} \left\{ \frac{D_m^*}{\tau_m^*} + \right. \\
 &\left. \frac{1}{\omega_{mi} \phi_{mi} S_{wmi} + (1 - \omega_{mi}) \phi_{mi-1} S_{wmi-1}} \frac{1}{\sqrt{v_{xm}^2 + v_{ym}^2 + v_{zm}^2}} (\alpha_{Lm} v_{xm}^2 + \alpha_{Tm} (v_{ym}^2 + v_{zm}^2)) \right\}
 \end{aligned}$$

As in the fracture, similar equations can be written for the HD of the other interfaces. Also, the values of ξ used in R_m and D_{1m} are sorption coefficients and depend mainly on the isotherm selected for use in the matrix continuum. Moreover, ω refers to the upstream weighting factor

and has values of 0 and 1. The value of ω is calculated depending on the sign of the velocity in the matrix and has the same convention used in the fracture.

4.4.3 Two-Point Upstream-Winding Discretization of Transport Equations

In this method, two approaches for discretization are used. In the first approach, the equation is discretized fully implicitly as follows:

Fracture Equation

$$\begin{aligned}
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i+\frac{1}{2}} (C_{wfi+1}^{n+1} - C_{wfi}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{i-\frac{1}{2}} (C_{wfi}^{n+1} - C_{wfi-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j+\frac{1}{2}} (C_{wffj+1}^{n+1} - C_{wffj}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{j-\frac{1}{2}} (C_{wffj}^{n+1} - C_{wffj-1}^{n+1}) + \\
 & (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k+\frac{1}{2}} (C_{wfk+1}^{n+1} - C_{wfk}^{n+1}) - (\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1})_{k-\frac{1}{2}} (C_{wfk}^{n+1} - C_{wfk-1}^{n+1}) + \\
 & q_{wfi+\frac{1}{2}}^{n+1} \left[\omega_{fi+1} \left\{ (1 + \mu_{i1}) C_{wfi}^{n+1} - \mu_{i1} C_{wfi-1}^{n+1} \right\} + (1 - \omega_{fi+1}) \left\{ (1 + \mu_{i2}) C_{wfi+1}^{n+1} - \mu_{i2} C_{wfi+2}^{n+1} \right\} \right] - \\
 & q_{wfi-\frac{1}{2}}^{n+1} \left[\omega_i \left\{ (1 + \mu_{i3}) C_{wfi-1}^{n+1} - \mu_{i3} C_{wfi-2}^{n+1} \right\} + (1 - \omega_i) \left\{ (1 + \mu_{i4}) C_{wfi}^{n+1} - \mu_{i4} C_{wfi+1}^{n+1} \right\} \right] + \\
 & q_{wff+\frac{1}{2}}^{n+1} \left[\omega_{ff+1} \left\{ (1 + \mu_{j1}) C_{wffj}^{n+1} - \mu_{j1} C_{wffj-1}^{n+1} \right\} + (1 - \omega_{ff+1}) \left\{ (1 + \mu_{j2}) C_{wffj+1}^{n+1} - \mu_{j2} C_{wffj+2}^{n+1} \right\} \right] - \\
 & q_{wff-\frac{1}{2}}^{n+1} \left[\omega_{ff} \left\{ (1 + \mu_{j3}) C_{wffj-1}^{n+1} - \mu_{j3} C_{wffj-2}^{n+1} \right\} + (1 - \omega_{ff}) \left\{ (1 + \mu_{j4}) C_{wffj}^{n+1} - \mu_{j4} C_{wffj+1}^{n+1} \right\} \right] + \tag{4.82} \\
 & q_{wfk+\frac{1}{2}}^{n+1} \left[\omega_{fk+1} \left\{ (1 + \mu_{k1}) C_{wfk}^{n+1} - \mu_{k1} C_{wfk-1}^{n+1} \right\} + (1 - \omega_{fk+1}) \left\{ (1 + \mu_{k2}) C_{wfk+1}^{n+1} - \mu_{k2} C_{wfk+2}^{n+1} \right\} \right] - \\
 & q_{wfk-\frac{1}{2}}^{n+1} \left[\omega_{fk} \left\{ (1 + \mu_{k3}) C_{wfk-1}^{n+1} - \mu_{k3} C_{wfk-2}^{n+1} \right\} + (1 - \omega_{fk}) \left\{ (1 + \mu_{k4}) C_{wfk}^{n+1} - \mu_{k4} C_{wfk+1}^{n+1} \right\} \right] + \\
 & C_{wfi}^{n+1} Q_{wfi}^{n+1} - \tau_{wmi/fi}^{n+1} \left(\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1} \right) - (\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi/fi}^{n+1}) (C_{wfi}^{n+1} - C_{wmi}^{n+1}) = \\
 & \frac{V_{Ri}}{\Delta t} \left[\left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^{n+1} - \left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^n \right] +
 \end{aligned}$$

$$V_{Ri} \left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\} \lambda_l \right]^{n+1}.$$

Matrix Equation:

$$\begin{aligned} & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{i+\frac{1}{2}} \left(C_{wmi+1}^{n+1} - C_{wmi}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{i-\frac{1}{2}} \left(C_{wmi}^{n+1} - C_{wmi-1}^{n+1} \right) + \\ & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{j+\frac{1}{2}} \left(C_{wmj+1}^{n+1} - C_{wmj}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{j-\frac{1}{2}} \left(C_{wmj}^{n+1} - C_{wmj-1}^{n+1} \right) + \\ & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{k+\frac{1}{2}} \left(C_{wmk+1}^{n+1} - C_{wmk}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{k-\frac{1}{2}} \left(C_{wmk}^{n+1} - C_{wmk-1}^{n+1} \right) + \\ & q_{wmi+\frac{1}{2}}^{n+1} \left[\omega_{mi+1} \left\{ (1 + \mu_{i1}) C_{wmi}^{n+1} - \mu_{i1} C_{wmi-1}^{n+1} \right\} + (1 - \omega_{mi+1}) \left\{ (1 + \mu_{i2}) C_{wmi+1}^{n+1} - \mu_{i2} C_{wmi+2}^{n+1} \right\} \right] - \\ & q_{wmi-\frac{1}{2}}^{n+1} \left[\omega_{mi} \left\{ (1 + \mu_{i3}) C_{wmi-1}^{n+1} - \mu_{i3} C_{wmi-2}^{n+1} \right\} + (1 - \omega_{mi}) \left\{ (1 + \mu_{i4}) C_{wmi}^{n+1} - \mu_{i4} C_{wmi+1}^{n+1} \right\} \right] + \\ & q_{wmj+\frac{1}{2}}^{n+1} \left[\omega_{mj+1} \left\{ (1 + \mu_{j1}) C_{wmj}^{n+1} - \mu_{j1} C_{wmj-1}^{n+1} \right\} + (1 - \omega_{mj+1}) \left\{ (1 + \mu_{j2}) C_{wmj+1}^{n+1} - \mu_{j2} C_{wmj+2}^{n+1} \right\} \right] - \quad 4.83 \\ & q_{wmj-\frac{1}{2}}^{n+1} \left[\omega_{mj} \left\{ (1 + \mu_{j3}) C_{wmj-1}^{n+1} - \mu_{j3} C_{wmj-2}^{n+1} \right\} + (1 - \omega_{mj}) \left\{ (1 + \mu_{j4}) C_{wmj}^{n+1} - \mu_{j4} C_{wmj+1}^{n+1} \right\} \right] + \\ & q_{wmk+\frac{1}{2}}^{n+1} \left[\omega_{mk+1} \left\{ (1 + \mu_{k1}) C_{wmk}^{n+1} - \mu_{k1} C_{wmk-1}^{n+1} \right\} + (1 - \omega_{mk+1}) \left\{ (1 + \mu_{k2}) C_{wmk+1}^{n+1} - \mu_{k2} C_{wmk+2}^{n+1} \right\} \right] - \\ & q_{wmk-\frac{1}{2}}^{n+1} \left[\omega_{mk} \left\{ (1 + \mu_{k3}) C_{wmk-1}^{n+1} - \mu_{k3} C_{wmk-2}^{n+1} \right\} + (1 - \omega_{mk}) \left\{ (1 + \mu_{k4}) C_{wmk}^{n+1} - \mu_{k4} C_{wmk+1}^{n+1} \right\} \right] + \\ & C_{wmi}^{*n+1} Q_{wmi}^{n+1} + \tau_{wmi/fi}^{n+1} \left(\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1} \right) + \left(\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmi/fi}^{n+1} \right) \left(C_{wfi}^{n+1} - C_{wmi}^{n+1} \right) = \end{aligned}$$

$$\frac{V_{Ri}}{\Delta t} \left[\left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^{n+1} - \left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^n \right] +$$

$$V_{Ri} \left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\} \lambda_l \right]^{n+1},$$

where

$$\mu_{i1} = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}}, \mu_{i2} = \frac{\Delta x_{i+1}}{\Delta x_{i+1} + \Delta x_{i+2}}, \mu_{i3} = \frac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}}, \mu_{i4} = \frac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_{i-2}},$$

$$\mu_{j1} = \frac{\Delta y_j}{\Delta y_j + \Delta y_{j-1}}, \mu_{j2} = \frac{\Delta y_{j+1}}{\Delta y_{j+1} + \Delta y_{j+2}}, \mu_{j3} = \frac{\Delta y_j}{\Delta y_j + \Delta y_{j+1}}, \mu_{j4} = \frac{\Delta y_{j-1}}{\Delta y_{j-1} + \Delta y_{j-2}},$$

$$\mu_{k1} = \frac{\Delta z_k}{\Delta z_k + \Delta z_{k-1}}, \mu_{k2} = \frac{\Delta z_{k+1}}{\Delta z_{k+1} + \Delta z_{k+2}}, \mu_{k3} = \frac{\Delta z_k}{\Delta z_k + \Delta z_{k+1}}, \text{ and } \mu_{k4} = \frac{\Delta z_{k-1}}{\Delta z_{k-1} + \Delta z_{k-2}}$$

the corresponding linear system of the fracture-flow equation is:

$$A_{2f} C_{wfk-2}^{n+1} + A_{1f} C_{wfk-1}^{n+1} + B_{2f} C_{wff-2}^{n+1} + B_{1f} C_{wff-1}^{n+1} + C_{2f} C_{wfi-2}^{n+1} + C_{1f} C_{wfi-1}^{n+1} + D_{1f} C_{wffjk}^{n+1} + D_{2f} C_{wmiik}^{n+1} + E_{1f} C_{wfi+1}^{n+1} + E_{2f} C_{wfi+2}^{n+1} + F_{1f} C_{wff+1}^{n+1} + F_{2f} C_{wff+2}^{n+1} + G_{1f} C_{wfk+1}^{n+1} + G_{2f} C_{wfk+2}^{n+1} = R_f^{n+1}, \quad 4.84$$

where

$$A_{1f} = HD_{wfk-\frac{1}{2}}^{n+1} - \omega_{fk} (1 + \mu_{k3}) q_{wfk-\frac{1}{2}}^{n+1} - \omega_{fk+1} \mu_{k1} q_{wfk+\frac{1}{2}}^{n+1},$$

$$A_{2f} = -\omega_{fk} \mu_{k3} q_{wfk-\frac{1}{2}}^{n+1},$$

$$B_{1f} = HD_{wff-\frac{1}{2}}^{n+1} - \omega_{ff} (1 + \mu_{j3}) q_{wff-\frac{1}{2}}^{n+1} - \omega_{ff+1} \mu_{j1} q_{wff+\frac{1}{2}}^{n+1},$$

$$B_{2f} = -\omega_{ff} \mu_{j3} q_{wff-\frac{1}{2}}^{n+1},$$

$$C_{1f} = HD_{wfi-\frac{1}{2}}^{n+1} - \omega_{fi} (1 + \mu_{i3}) q_{wfi-\frac{1}{2}}^{n+1} - \omega_{fi+1} \mu_{i1} q_{wfi+\frac{1}{2}}^{n+1},$$

$$C_{2f} = -\omega_{fi} \mu_{i3} q_{wfi-\frac{1}{2}}^{n+1},$$

$$E_{1f} = HD_{wfi+\frac{1}{2}}^{n+1} + (1 - \omega_{fi+1}) (1 + \mu_{i2}) q_{wfi+\frac{1}{2}}^{n+1} + (1 - \omega_{fi}) \mu_{i4} q_{wfi-\frac{1}{2}}^{n+1},$$

$$E_{2f} = -(1 - \omega_{fi+1}) \mu_{i2} q_{wfi+\frac{1}{2}}^{n+1},$$

$$F_{1f} = HD_{wff+\frac{1}{2}}^{n+1} + (1 - \omega_{fj+1})(1 + \mu_{j2})q_{wff+\frac{1}{2}}^{n+1} + (1 - \omega_{fj})\mu_{j4}q_{wff-\frac{1}{2}}^{n+1},$$

$$F_{2f} = -(1 - \omega_{fj+1})\mu_{j2}q_{wff+\frac{1}{2}}^{n+1},$$

$$G_{1f} = HD_{wfk+\frac{1}{2}}^{n+1} + (1 - \omega_{fk+1})(1 + \mu_{k2})q_{wfk+\frac{1}{2}}^{n+1} + (1 - \omega_{fk})\mu_{k4}q_{wfk-\frac{1}{2}}^{n+1},$$

$$G_{2f} = -(1 - \omega_{fk+1})\mu_{k2}q_{wfk+\frac{1}{2}}^{n+1},$$

$$D_{1f}^{n+1} = -HD_{wfk-\frac{1}{2}}^{n+1} - HD_{wff-\frac{1}{2}}^{n+1} - HD_{wfi-\frac{1}{2}}^{n+1} - HD_{wfk+\frac{1}{2}}^{n+1} - HD_{wff+\frac{1}{2}}^{n+1} - HD_{wfi+\frac{1}{2}}^{n+1} -$$

$$q_{wfk-\frac{1}{2}}^{n+1}(1 - \omega_{fk})(1 + \mu_{k4}) - q_{wff-\frac{1}{2}}^{n+1}(1 - \omega_{fj})(1 + \mu_{j4}) - q_{wfi-\frac{1}{2}}^{n+1}(1 - \omega_{fi})(1 + \mu_{i4}) +$$

$$q_{wfk+\frac{1}{2}}^{n+1}\omega_{fk+1}(1 + \mu_{k1}) + q_{wff+\frac{1}{2}}^{n+1}\omega_{fj+1}(1 + \mu_{j1}) + q_{wfi+\frac{1}{2}}^{n+1}\omega_{fi+1}(1 + \mu_{i1}) -$$

$$\frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} + V_{Ri} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} \lambda - \tau_{w/mfi}^{n+1} \omega_{m/fi} - \phi_m^{n+1} S_{wm}^{n+1} K_{wmlfi}^{n+1},$$

$$D_{2f}^{n+1} = -\tau_{wmlfi}^{n+1} (1 - \omega_{m/fi}) + \phi_m^{n+1} S_{wm}^{n+1} K_{wmlfi}^{n+1},$$

and

$$R_f^{n+1} = \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^n C_{wfi}^n - V_{Ri} \sum_{l=1}^L \left\{ \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} \lambda_l C_{wfli}^{n+1} \right\} - C_{wfi}^{n+1} Q_{wfi}^{n+1},$$

and the linear system of the matrix equation is:

$$A_{2m} C_{wmk-2}^{n+1} + A_{1m} C_{wmk-1}^{n+1} + B_{2m} C_{wmj-2}^{n+1} + B_{1m} C_{wmj-1}^{n+1} + C_{2m} C_{wmi-2}^{n+1} + C_{1m} C_{wmi-1}^{n+1} + D_{1m} C_{wmijk}^{n+1} + D_{2m} C_{wifjk}^{n+1} +$$

$$E_{1m} C_{wmi+1}^{n+1} + E_{2m} C_{wmi+2}^{n+1} + F_{1m} C_{wmj+1}^{n+1} + F_{2m} C_{wmj+2}^{n+1} + G_{1m} C_{wmk+1}^{n+1} + G_{2m} C_{wmk+2}^{n+1} = R_m^{n+1}, \quad 4.85$$

where

$$A_{1m} = HD_{wmk-\frac{1}{2}}^{n+1} - \omega_{mk} (1 + \mu_{k3}) q_{wmk-\frac{1}{2}}^{n+1} - \omega_{mk+1} \mu_{k1} q_{wmk+\frac{1}{2}}^{n+1},$$

$$A_{2m} = -\omega_{mk} \mu_{k3} q_{\omega_{mk}-\frac{1}{2}}^{n+1},$$

$$B_{1m} = HD_{\omega_{mj}-\frac{1}{2}}^{n+1} - \omega_{mj} (1 + \mu_{j3}) q_{\omega_{mj}-\frac{1}{2}}^{n+1} - \omega_{mj+1} \mu_{j1} q_{\omega_{mj}+\frac{1}{2}}^{n+1},$$

$$B_{2m} = -\omega_{mj} \mu_{j3} q_{\omega_{mj}-\frac{1}{2}}^{n+1},$$

$$C_{1m} = HD_{\omega_{mi}-\frac{1}{2}}^{n+1} - \omega_{mi} (1 + \mu_{i3}) q_{\omega_{mi}-\frac{1}{2}}^{n+1} - \omega_{mi+1} \mu_{i1} q_{\omega_{mi}+\frac{1}{2}}^{n+1},$$

$$C_{2m} = -\omega_{mi} \mu_{i3} q_{\omega_{mi}-\frac{1}{2}}^{n+1},$$

$$E_{1m} = HD_{\omega_{mi}+\frac{1}{2}}^{n+1} + (1 - \omega_{mi+1}) (1 + \mu_{i2}) q_{\omega_{mi}+\frac{1}{2}}^{n+1} + (1 - \omega_{mi}) \mu_{i4} q_{\omega_{mi}-\frac{1}{2}}^{n+1},$$

$$E_{2m} = -(1 - \omega_{mi+1}) \mu_{i2} q_{\omega_{mi}+\frac{1}{2}}^{n+1},$$

$$F_{1m} = HD_{\omega_{mj}+\frac{1}{2}}^{n+1} + (1 - \omega_{mj+1}) (1 + \mu_{j2}) q_{\omega_{mj}+\frac{1}{2}}^{n+1} + (1 - \omega_{mj}) \mu_{j4} q_{\omega_{mj}-\frac{1}{2}}^{n+1},$$

$$F_{2m} = -(1 - \omega_{mj+1}) \mu_{j2} q_{\omega_{mj}+\frac{1}{2}}^{n+1},$$

$$G_{1m} = HD_{\omega_{mk}+\frac{1}{2}}^{n+1} + (1 - \omega_{mk+1}) (1 + \mu_{k2}) q_{\omega_{mk}+\frac{1}{2}}^{n+1} + (1 - \omega_{mk}) \mu_{k4} q_{\omega_{mk}-\frac{1}{2}}^{n+1},$$

$$G_{2m} = -(1 - \omega_{mk+1}) \mu_{k2} q_{\omega_{mk}+\frac{1}{2}}^{n+1},$$

$$D_{1m}^{n+1} = -HD_{\omega_{mk}-\frac{1}{2}}^{n+1} - HD_{\omega_{mj}-\frac{1}{2}}^{n+1} - HD_{\omega_{mi}-\frac{1}{2}}^{n+1} - HD_{\omega_{mk}+\frac{1}{2}}^{n+1} - HD_{\omega_{mj}+\frac{1}{2}}^{n+1} - HD_{\omega_{mi}+\frac{1}{2}}^{n+1} -$$

$$q_{\omega_{mk}-\frac{1}{2}}^{n+1} (1 - \omega_{mk}) (1 + \mu_{k4}) - q_{\omega_{mj}-\frac{1}{2}}^{n+1} (1 - \omega_{mj}) (1 + \mu_{j4}) - q_{\omega_{mi}-\frac{1}{2}}^{n+1} (1 - \omega_{mi}) (1 + \mu_{i4}) +$$

$$q_{\omega_{mk}+\frac{1}{2}}^{n+1} \omega_{mk+1} (1 + \mu_{k1}) + q_{\omega_{mj}+\frac{1}{2}}^{n+1} \omega_{mj+1} (1 + \mu_{j1}) + q_{\omega_{mi}+\frac{1}{2}}^{n+1} \omega_{mi+1} (1 + \mu_{i1}) -$$

$$\frac{V_{Ri}}{\Delta t} \left\{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \right\}^{n+1} + V_{Ri} \left\{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \right\}^{n+1} \lambda + \tau_{w/mfi}^{n+1} (1 - \omega_{m/fi}) - \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1},$$

$$D_{2m}^{n+1} = \tau_{wm/fi}^{n+1} \omega_{m/fi} + \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1},$$

and

$$R_m^{n+1} = \frac{V_{Ri}}{\Delta t} \left\{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \right\}^n C_{wmi}^n - V_{Ri} \sum_{l=1}^L \left\{ \left\{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mli} \right\}^{n+1} \lambda_l C_{wmli}^{n+1} \right\} - C_{wmi}^{*n+1} Q_{wmi}^{n+1}.$$

The second approach in the discretization of the two-point upstream winding is the split-operator method. In the split-operator method, the equation is discretized similarly but the time level for the concentration that appears in the second diagonal of the numerical matrix in each dimension (i.e., $i-2, i+2, j-2, j+2, k-2$, and $k+2$ diagonals) is treated explicitly. This, in turn, will lead to a linear system similar to Equations 4.84 and 4.85 except that the right-hand side of the equation will be equivalent to

$$R_f^{n+1} = R_f^{n+1} (\text{Equation 4.84}) + A_{2f} C_{wfk-2}^n + B_{2f} C_{wff-2}^n + C_{2f} C_{wfi-2}^n + E_{2f} C_{wfi+2}^n + F_{2f} C_{wff+2}^n + G_{2f} C_{wfk+2}^n \quad 4.86$$

for the fracture, and

$$R_m^{n+1} = R_m^{n+1} (\text{Equation 4.85}) + A_{2m} C_{wmk-2}^n + B_{2m} C_{wmj-2}^n + C_{2m} C_{wmi-2}^n + E_{2m} C_{wmi+2}^n + F_{2m} C_{wmj+2}^n + G_{2m} C_{wmk+2}^n \quad 4.87$$

for the matrix. The accuracy of the split operator depends on the size of the time step, which resides somewhere between that of the one-point upwinding and that of the two-point upwinding. However, the saving in computation time is substantial.

4.4.4 Mid-Point Weighting Discretization of Transport Equations

Fracture Equation

$$\left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{i+\frac{1}{2}} (C_{wfi+1}^{n+1} - C_{wfi}^{n+1}) - \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{i-\frac{1}{2}} (C_{wfi}^{n+1} - C_{wfi-1}^{n+1}) +$$

$$\left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{j+\frac{1}{2}} (C_{wff+1}^{n+1} - C_{wff}^{n+1}) - \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{j-\frac{1}{2}} (C_{wff}^{n+1} - C_{wff-1}^{n+1}) +$$

$$\left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{k+\frac{1}{2}} (C_{wfk+1}^{n+1} - C_{wfk}^{n+1}) - \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1} \right)_{k-\frac{1}{2}} (C_{wfk}^{n+1} - C_{wfk-1}^{n+1}) +$$

$$\begin{aligned}
 & q_{wfi+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{i2}) C_{wfi+1}^{n+1} + (0.5 - \mu_{i2}) C_{wfi}^{n+1} \right] - q_{wfi-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{i1}) C_{wfi}^{n+1} + (0.5 - \mu_{i1}) C_{wfi-1}^{n+1} \right] + \\
 & q_{wfi+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{j2}) C_{wfi+1}^{n+1} + (0.5 - \mu_{j2}) C_{wfi}^{n+1} \right] - q_{wfi-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{j1}) C_{wfi}^{n+1} + (0.5 - \mu_{j1}) C_{wfi-1}^{n+1} \right] + \quad 4.88 \\
 & q_{wfk+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{k2}) C_{wfk+1}^{n+1} + (0.5 - \mu_{k2}) C_{wfk}^{n+1} \right] - q_{wfk-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{k1}) C_{wfk}^{n+1} + (0.5 - \mu_{k1}) C_{wfk-1}^{n+1} \right] + \\
 & C_{wfi}^{n+1} Q_{wfi}^{n+1} - \tau_{wmlfi}^{n+1} \left(\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1} \right) - \left(\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmlfi}^{n+1} \right) \left(C_{wfi}^{n+1} - C_{wmi}^{n+1} \right) = \\
 & \frac{V_{Ri}}{\Delta t} \left[\left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^{n+1} - \left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^n \right] + \\
 & V_{Ri} \left\{ \phi_{fi} S_{wfi} C_{wfi} + (1 - \phi_t) \rho_s C_{sfi} \right\}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\left\{ \phi_{fi} S_{wfi} C_{wfli} + (1 - \phi_t) \rho_s C_{sfli} \right\} \lambda_l \right]^{n+1}
 \end{aligned}$$

Matrix Equation

$$\begin{aligned}
 & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{i+\frac{1}{2}} \left(C_{wmi+1}^{n+1} - C_{wmi}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{i-\frac{1}{2}} \left(C_{wmi}^{n+1} - C_{wmi-1}^{n+1} \right) + \\
 & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{j+\frac{1}{2}} \left(C_{wmj+1}^{n+1} - C_{wmj}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{j-\frac{1}{2}} \left(C_{wmj}^{n+1} - C_{wmj-1}^{n+1} \right) + \\
 & \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{k+\frac{1}{2}} \left(C_{wmk+1}^{n+1} - C_{wmk}^{n+1} \right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1} \right)_{k-\frac{1}{2}} \left(C_{wmk}^{n+1} - C_{wmk-1}^{n+1} \right) + \\
 & q_{wmi+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{i2}) C_{wmi+1}^{n+1} + (0.5 - \mu_{i2}) C_{wmi}^{n+1} \right] - q_{wmi-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{i1}) C_{wmi}^{n+1} + (0.5 - \mu_{i1}) C_{wmi-1}^{n+1} \right] + \\
 & q_{wmj+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{j2}) C_{wmj+1}^{n+1} + (0.5 - \mu_{j2}) C_{wmj}^{n+1} \right] - q_{wmj-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{j1}) C_{wmj}^{n+1} + (0.5 - \mu_{j1}) C_{wmj-1}^{n+1} \right] + \quad 4.89 \\
 & q_{wmk+\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{k2}) C_{wmk+1}^{n+1} + (0.5 - \mu_{k2}) C_{wmk}^{n+1} \right] - q_{wmk-\frac{1}{2}}^{n+1} \left[(0.5 + \mu_{k1}) C_{wmk}^{n+1} + (0.5 - \mu_{k1}) C_{wmk-1}^{n+1} \right] + \\
 & C_{wmi}^{n+1} Q_{wmi}^{n+1} + \tau_{wmlfi}^{n+1} \left(\omega_{m/fi} C_{wfi}^{n+1} + (1 - \omega_{m/fi}) C_{wmi}^{n+1} \right) + \left(\phi_{mi}^{n+1} S_{wmi}^{n+1} K_{wmlfi}^{n+1} \right) \left(C_{wfi}^{n+1} - C_{wmi}^{n+1} \right) = \\
 & \frac{V_{Ri}}{\Delta t} \left[\left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^{n+1} - \left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^n \right] +
 \end{aligned}$$

$$V_{Ri} \left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\}^{n+1} \lambda - V_{Ri} \sum_{l=1}^L \left[\left\{ \phi_{mi} S_{wmi} C_{wmi} + (1 - \phi_t) \rho_s C_{smi} \right\} \lambda_l \right]^{n+1}$$

where

$$\mu_{i1} = \frac{1}{4} \frac{\Delta x_{i-1} - \Delta x_i}{\Delta x_{i-1} + \Delta x_i}, \quad \mu_{i2} = \frac{1}{4} \frac{\Delta x_i - \Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}}, \quad \mu_{j1} = \frac{1}{4} \frac{\Delta y_{j-1} - \Delta y_j}{\Delta y_{j-1} + \Delta y_j}, \quad \mu_{j2} = \frac{1}{4} \frac{\Delta y_j - \Delta y_{j+1}}{\Delta y_j + \Delta y_{j+1}},$$

$$\mu_{k1} = \frac{1}{4} \frac{\Delta z_{k-1} - \Delta z_k}{\Delta z_{k-1} + \Delta z_k}, \quad \text{and} \quad \mu_{k2} = \frac{1}{4} \frac{\Delta z_k - \Delta z_{k+1}}{\Delta z_k + \Delta z_{k+1}}$$

A similar linear system as in Equations 4.79 and 4.81 is formed, in which

$$A_f = HD_{wfk-\frac{1}{2}}^{n+1} - (0.5 + \mu_{k1}) q_{wfk-\frac{1}{2}}^{n+1}$$

$$A_m = HD_{wnk-\frac{1}{2}}^{n+1} - (0.5 + \mu_{k1}) q_{wnk-\frac{1}{2}}^{n+1}$$

$$B_f = HD_{wff-\frac{1}{2}}^{n+1} - (0.5 + \mu_{j1}) q_{wff-\frac{1}{2}}^{n+1}$$

$$B_m = HD_{wmj-\frac{1}{2}}^{n+1} - (0.5 + \mu_{j1}) q_{wmj-\frac{1}{2}}^{n+1}$$

$$C_f = HD_{wfi-\frac{1}{2}}^{n+1} - (0.5 + \mu_{i1}) q_{wfi-\frac{1}{2}}^{n+1}$$

$$C_m = HD_{wmi-\frac{1}{2}}^{n+1} - (0.5 + \mu_{i1}) q_{wmi-\frac{1}{2}}^{n+1}$$

$$E_f = HD_{wfi+\frac{1}{2}}^{n+1} + (0.5 + \mu_{i2}) q_{wfi+\frac{1}{2}}^{n+1}$$

$$E_m = HD_{wmi+\frac{1}{2}}^{n+1} + (0.5 + \mu_{i2}) q_{wmi+\frac{1}{2}}^{n+1}$$

$$F_f = HD_{wff+\frac{1}{2}}^{n+1} + (0.5 + \mu_{j2}) q_{wff+\frac{1}{2}}^{n+1}$$

$$F_m = HD_{wmj+\frac{1}{2}}^{n+1} + (0.5 + \mu_{j2}) q_{wmj+\frac{1}{2}}^{n+1}$$

$$G_f = HD_{wfk+\frac{1}{2}}^{n+1} + (0.5 + \mu_{k2})q_{wfk+\frac{1}{2}}^{n+1}$$

$$G_m = HD_{wmk+\frac{1}{2}}^{n+1} + (0.5 + \mu_{k2})q_{wmk+\frac{1}{2}}^{n+1}$$

and

$$D_{jf}^{n+1} = -HD_{wfk-\frac{1}{2}}^{n+1} - HD_{wfi-\frac{1}{2}}^{n+1} - HD_{wfi-\frac{1}{2}}^{n+1} - HD_{wfk+\frac{1}{2}}^{n+1} - HD_{wfi+\frac{1}{2}}^{n+1} - HD_{wfi+\frac{1}{2}}^{n+1} - q_{wfk-\frac{1}{2}}^{n+1}(0.5 + \mu_{ki}) -$$

$$q_{wfi-\frac{1}{2}}^{n+1}(0.5 + \mu_{j1}) - q_{wfi-\frac{1}{2}}^{n+1}(0.5 + \mu_{i1}) + q_{wfk+\frac{1}{2}}^{n+1}(0.5 - \mu_{k2}) + q_{wfi+\frac{1}{2}}^{n+1}(0.5 - \mu_{j2}) + q_{wfi+\frac{1}{2}}^{n+1}(0.5 - \mu_{i2}) -$$

$$\frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} + V_{Ri} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} \lambda - \tau_{w/mfi}^{n+1} \omega_{m/fi} - \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1}$$

$$D_{1m}^{n+1} = -HD_{wmk-\frac{1}{2}}^{n+1} - HD_{wmj-\frac{1}{2}}^{n+1} - HD_{wmi-\frac{1}{2}}^{n+1} - HD_{wmk+\frac{1}{2}}^{n+1} - HD_{wmj+\frac{1}{2}}^{n+1} - HD_{wmi+\frac{1}{2}}^{n+1} - q_{wmk-\frac{1}{2}}^{n+1}(0.5 + \mu_{k1}) -$$

$$q_{wmj-\frac{1}{2}}^{n+1}(0.5 + \mu_{j1}) - q_{wmi-\frac{1}{2}}^{n+1}(0.5 + \mu_{i1}) + q_{wmk+\frac{1}{2}}^{n+1}(0.5 - \mu_{k2}) + q_{wmj+\frac{1}{2}}^{n+1}(0.5 - \mu_{j2}) + q_{wmi+\frac{1}{2}}^{n+1}(0.5 - \mu_{i2}) -$$

$$\frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^{n+1} + V_{Ri} \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^{n+1} \lambda + \tau_{w/mfi}^{n+1} (1 - \omega_{m/fi}) - \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1}$$

$$D_{2f}^{n+1} = -\tau_{wm/fi}^{n+1} (1 - \omega_{m/fi}) + \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1}$$

$$D_{2m}^{n+1} = \tau_{wm/fi}^{n+1} \omega_{m/fi} + \phi_m^{n+1} S_{wm}^{n+1} K_{wm/fi}^{n+1}$$

$$R_f^{n+1} = \frac{V_{Ri}}{\Delta t} \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^n C_{wfi}^n - V_{Ri} \sum_{l=1}^L \left\{ \{ \phi_{fi} S_{wfi} + (1 - \phi_t) \rho_s \xi_{fi} \}^{n+1} \lambda_l C_{wfli}^{n+1} \right\} - C_{wfi}^{*n+1} Q_{wfi}^{n+1}$$

$$R_m^{n+1} = \frac{V_{Ri}}{\Delta t} \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^n C_{wmi}^n - V_{Ri} \sum_{l=1}^L \left\{ \{ \phi_{mi} S_{wmi} + (1 - \phi_t) \rho_s \xi_{mi} \}^{n+1} \lambda_l C_{wmli}^{n+1} \right\} - C_{wmi}^{*n+1} Q_{wmi}^{n+1}$$

It is worth mentioning that in all the above methods no fictitious (ghost) grid points are used in the discretization of the boundary conditions. In two-point upwinding, there might be, depending on the direction of the flow, a need for such points. However, one-point upstream is used instead. This has been done in the problem at hand because the source is located in the middle of the

Banded matrix solver is used to invert the numerical matrix. Two techniques are applied to reduce the size of the numerical matrix to more manageable dimensions. In the first technique, an optimum dimension (in standard ordering) is chosen to define the connectivity of the grid blocks. The guideline for this choice is the minimum number of grid blocks. Hence, for a two dimensional problem in x and y , the numbering will start from $\min(x,y)$ and proceed to the next dimension. The second technique is used to straighten the numerical matrix diagonals and limit the calculation to the entries between the uppermost and the lowermost diagonals. Therefore, for a single-porosity system in two dimensions, a pentagonal matrix of $IBW \cdot G$ is inverted instead of $G \cdot G$ matrix, where IBW is the band width.

4.5 Other Important Physical and Chemical Aspects of the Modeling

4.5.1 Temperature Dependency

At the user's option, NUTS can implement a temperature⁺ dependency for selected parameters. For example, solubility limit is considered either as (1) a constant or (2) it can be evaluated from data correlations that account for its thermal dependency in non-isothermal processes.

Additionally, molecular diffusivity varies with temperature, and the following derived formula is used to account for that dependency:

$$D_m^*(T) = D_{mref}^* \frac{T}{T_{ref}} \frac{\mu_{ref}}{\mu(T)}, \quad 4.90$$

where

D_{mref}^* is the molecular diffusion evaluated at the temperature T_{ref} ,
 $D_m^*(T)$ is the molecular diffusion at temperature T ,
 μ_{ref} is the reference viscosity at T_{ref} ,
 $\mu(T)$ solvent viscosity at temperature T , and
 T_{ref} is the reference temperature.

Sorption is also a temperature-dependent process. The following is a zeroth-order formulation derived to account preliminarily for the temperature dependency of k_d . It can also be applied to the Freundlich and Langmuir coefficients with fair accuracy.

$$k_{d2} = k_{d1} \exp\left[\frac{\xi(T_2 - T_1)}{T_2 T_1}\right], \quad \text{where } \xi = \frac{\Delta H^o}{R}. \quad 4.91$$

In the above formulation, k_{d1} , and k_{d2} are equilibrium partition constants at temperatures T_1 and T_2 , respectively. H^o is the enthalpy of sorption at reference temperature T^o , and R is the universal gas constant.

⁺ Temperatures should be provided to NUTS as an input by suitable thermal simulator.

4.5.2 Theoretical Foundations for the Gas-Transport Model Phase

In gas transport problems, it is essential to consider the chemical interaction between the flowing gas and the constituents of the porous medium through which the gas flows. That interaction or phase equilibrium can be classified as (1) gas/solid-phase equilibrium described by sorption-desorption processes, and (2) gas/liquid-phase equilibrium.

The description of multiphase equilibrium in porous media is complex and requires a compositional simulator that strongly couples the transport with the flow equations. This kind of treatment arises because some mixtures like, for example VOCs + H₂O, are observed under certain conditions of pressure, temperature and composition, as a single phase (liquid or vapor), a double-phase (vapor-liquid or liquid-liquid), or a triple-phase (vapor-liquid-liquid) fluid. To (a) describe such a system appropriately and (b) to predict the number and form of the phases in equilibrium, the transport by the liquid and gas phases should be calculated simultaneously. In that event, the solute would be partitioned according to its equilibrium constant between the phases in equilibrium.

Such a compositionally equilibrated system requires not only a detailed description of the pressure, temperature and initial composition fields in the computational domain, but also is very demanding in regard to computation time and resources. Because of data limitations, NUTS considers the porous medium and its liquid contents as a retardation factor for the constituents transported by the gas phase (Doctor et al., 1992). The phenomena described below are included in NUTS.

1. Solid-/gas-phase equilibrium is treated using the same linear, equilibrium, sorption isotherm described above for liquid-phase flows.
2. Liquid-/gas-phase equilibrium is modeled through gas solubility in the brine as a function of temperature. The equation that describes such equilibria is

$$K_{LG} = \frac{C_L}{C_G}, \quad 4.92$$

where C_L and C_G are the concentration of the solute in the liquid and gas phases, respectively. The value of the ratio on the right-hand side is represented as a linear function of temperature, namely:

$$\frac{C_L}{C_G} = A - BT, \quad 4.93$$

where A and B are the intercept and slope, respectively, of the empirical data. This linear representation is valid only for a limited range of temperatures and only at low pressures, where compositional dependency is usually weak.

Because migration in the liquid phase is much slower than in the gas phase, and because the gas- and the liquid-phase transports are decoupled in NUTS, liquid transport of a constituent

transported abundantly by the gas phase will not be tracked. With the above mentioned assumptions, the transport in the gas phase of a dual-permeability system can be described by the following finite-difference equations:

Equation for Material Transport in the Fracture Component of the Gaseous Flow:

$$\begin{aligned} & \Delta_x [\phi_f S_{Gf} K_{Gf} \Delta_x C_{nGf}] + \Delta_y [\phi_f S_{Gf} K_{Gf} \Delta_y C_{nGf}] + \Delta_z [\phi_f S_{Gf} K_{Gf} \Delta_z C_{nGf}] + \\ & \Delta_x q_{Gf} C_{nGf} + \Delta_y q_{Gf} C_{nGf} + \Delta_z q_{Gf} C_{nGf} + C_{nGf}^* Q_{Gf} - \tau_{Gm/f} C_{nGm/f} - \phi_m S_{Gm} K_{Gm/f} (C_{nGf} - C_{nGm}) = \\ & \frac{V_R}{\Delta t} \Delta_t [\phi_f S_{Gf} C_{nGf} + (1 - \phi_t) \rho_s C_{nsf} + \phi_f (1 - S_{Gf}) K_{LG} C_{nGf}] + \\ & V_R [\phi_f S_{Gf} C_{nGf} + (1 - \phi_t) \rho_s C_{nsf} + \phi_f (1 - S_{Gf}) K_{LG} C_{nGf}] \lambda_n - \\ & V_R \sum_{j=1}^J \left\{ \left[\phi_f S_{Gf} C_{nGf_{pj}} + (1 - \phi_t) \rho_s C_{nsf_{pj}} + \phi_f (1 - S_{Gf}) K_{LG} C_{nGf_{pj}} \right] \lambda_{npj} \right\} \quad n = 1, 2, \dots, N \end{aligned} \quad 4.94$$

Equation for Material Transport in the Matrix Component of the Gaseous Flow:

$$\begin{aligned} & \Delta_x [\phi_m S_{Gm} K_{Gm} \Delta_x C_{nGm}] + \Delta_y [\phi_m S_{Gm} K_{Gm} \Delta_y C_{nGm}] + \Delta_z [\phi_m S_{Gm} K_{Gm} \Delta_z C_{nGm}] + \\ & \Delta_x q_{Gm} C_{nGm} + \Delta_y q_{Gm} C_{nGm} + \Delta_z q_{Gm} C_{nGm} + C_{nGm}^* Q_{Gm} + \tau_{Gm/f} C_{nGm/f} + \phi_m S_{Gm} K_{Gm/f} (C_{nGf} - C_{nGm}) = \\ & \frac{V_R}{\Delta t} \Delta_t [\phi_m S_{Gm} C_{nGm} + (1 - \phi_t) \rho_s C_{nsm} + \phi_m (1 - S_{Gm}) K_{LG} C_{nGm}] + \\ & V_R [\phi_m S_{Gm} C_{nGm} + (1 - \phi_t) \rho_s C_{nsm} + \phi_m (1 - S_{Gm}) K_{LG} C_{nGm}] \lambda_n - \\ & V_R \sum_{j=1}^J \left\{ \left[\phi_m S_{Gm} C_{nGm_{pj}} + (1 - \phi_t) \rho_s C_{nsm_{pj}} + \phi_m (1 - S_{Gm}) K_{LG} C_{nGm_{pj}} \right] \lambda_{npj} \right\} \quad n = 1, 2, \dots, N \end{aligned} \quad 4.95$$

where

- C_{nG} = solute concentration of component n in the gas phase [kg/m³],
- C_{nG}^* = injected or produced solute concentration of component n [kg/m³],
- C_{ns} = sorbate concentration of component n [kg/kg],
- K = dispersion coefficient [m²/s],
- K_{LG} = gas-liquid equilibrium constant [dimensionless],
- N = total number of nuclides,
- Q = gas injection/production rate [m³/s],
- q = gas interfacial volumetric rate [m³/s],
- S = gas saturation, fraction,
- t = time [s],
- V_R = grid block volume [m³],
- Δ_t = value at time n+1 - value at time n,
- Δ_x = central finite difference operator in x-direction,

- Δ_y = central finite difference operator in y-direction,
 Δ_z = central finite difference operator in z-direction,
 λ = decay constant [s^{-1}],
 ϕ = porosity [dimensionless],
 ρ = density [kg/m^3],
 τ = matrix /fracture transfer function [m^3/s],

Subscripts:

- f = fracture
 m = matrix
 m/f = matrix/fracture
 n = component number
 pj = parent number
 s = solid phase (rock)
 t = total (fracture + matrix)
 G = gas phase

In the above formulae, the delta notation (Δ) is used for central finite difference operator and applies to the indices. For example, the term $\Delta_x q_{Gm} C_{nGm}$ is a compact notation refers to the

$$\{q_{Gm} C_{nGm}\}_{i+\frac{1}{2}} - \{q_{Gm} C_{nGm}\}_{i-\frac{1}{2}}.$$

4.5.3 Precipitate Models in NUTS:

In the 1996 CCA PA calculation, the principle of the conservation of mass led to N governing partial-differential equations for advective transport of the dissolved N radioisotopes with decay in a single porosity matrix domain. The l^{th} constituent equation ($l = 1, 2, \dots, N$) is:

$$-\nabla \bullet v_w C_l + Q^* + q = \frac{\partial}{\partial t} (\phi S_w C_l) + (\phi S_w C_l) \lambda_l - \phi S_w \sum_{j=1}^{j_l} C_{l,j} \lambda_j \quad 4.96$$

where the nomenclature and the subscripts are defined as follows:

- $\nabla \bullet$ = divergence operator
 Q^* = source/sink mass rate per unit volume ($kg/s/m^3$) due to injection/ production
 q = source/sink mass rate per unit volume ($kg/s/m^3$) due to precipitate/dissolution
 ϕ = porosity (dimensionless fraction given by BRAGFLO)
 v = advective velocity vector (m/s and given by BRAGFLO)
 C = solute concentration (kg/m^3 , the principal unknown)
 t = time (s)
 N = number of isotopes
 S = saturation (dimensionless fraction given by BRAGFLO)
 λ = decay constant (s^{-1} and known data)
 w = brine (the only phase treated in the 1996 CCA PA)

- l = isotope index
- j = parent isotope index
- J_l = number of parent isotopes for the l^{th} isotope

Notice that in Equation 4.96, NUTS disregards sorption and diffusion/dispersion process. The system of N transport equations as given above constitutes only part of the relevant boundary-value problem. To complete the problem specification it is necessary to 1) give the initial concentration throughout the domain at time zero, and 2) describe the numerical implementation of the dissolution process (solid and dissolved waste interaction). A description for the precipitate governing equation will follow shortly.

In the 1996 CCA PA, NUTS's two-dimensional grid domain was identical to BRAGFLO's grid domain, that is, it includes the entire land withdrawal boundary region. That domain is referred to herein as $x \in [0, X]$ and $y \in [0, Y]$. NUTS's initial conditions are:

$$C_l(x, y, 0) = f_l(x, y) \quad 0 < x < X, \quad 0 < y < Y, \quad 4.97$$

where $f_l(x, y)$ represents the initial concentration of the l^{th} radioisotope and is understood to be known at every point in the grid domain. For all calculations, the initial concentrations outside the waste regions are zero. However, within the repository region, NUTS initializes the concentration and the solid phase isotope mass, hereafter called precipitate in each grid block as:

$$\left. \begin{aligned} C_l(x, y, 0) &= C_{IS}^0 \\ P_l^0 &= T_l^0 - \phi S_w C_{IS}^0 \end{aligned} \right\} \text{ if } T_l^0 \geq M_l^0 \quad 4.98$$

for the cases where the total mass T_l^0 (kg/m^3 bulk volume) of the isotope l is $\geq M_l^0$, the mass (kg/m^3 bulk volume) required to maintain the isotope solubility limit C_{IS}^0 (kg/m^3 brine) and where, P_l^0 (kg/m^3 bulk volume) is the precipitate, and the superscript 0 refers to the initial time (time = 0). In the cases where $T_l^0 < M_l^0$, the initial concentration and precipitate are,

$$\left. \begin{aligned} C_l(x, y, 0) &= \frac{T_l^0}{\phi S_w} \\ P_l^0 &= 0 \end{aligned} \right\} \text{ if } T_l^0 < M_l^0 \quad 4.99$$

In reality, when ground water flows into the repository and come into contact with the waste, dissolution of the soluble isotopes takes place and continues until equilibrium concentrations are attained, or until all undissolved supplies of radioisotopes are consumed. The capacity of the brines to dissolve radioisotopes is controlled by temperature, pressure and the composition of both the brine and the waste. This process is dynamic and requires extensive equilibrium calculations which are done neither in NUTS nor in the chemistry dissolution model coupled with NUTS. Instead, NUTS must be given the effective solubility limit (through the Source

Term) for each element or isotope in the waste. If the element solubility limit is given, NUTS will allocate the solubility for the isotopes according to their mole fractions (in the implicit precipitate treatment and as will be explained later, allocation using mole fractions may lead to highly nonlinear system). If radioisotope concentrations should exceed their solubility limits, NUTS will precipitate the excess in the grid block. The precipitated mass is subjected to decay and the precipitation is treated as a reversible process. If the concentration of a solute drops below the solubility limit, the brine is permitted to dissolve from the precipitate a mass sufficient to restore the solubility limit of that solute or until the precipitate is exhausted. The precipitate calculations can be done with two different numerical implementation* :

Explicit treatment. In the explicit method, the precipitate and dissolved species concentrations are adjusted at the end of the time step in each grid block. The equations to be solved are Equation 4.96 (with $q = 0$) and the solid phase equation

$$\sum_{j=1}^{J_l} \lambda_{lj} P_{lj} - \lambda_l P_l = \frac{\partial P_l}{\partial t} \quad 4.100$$

for C_l^{n+1} and P_l^{n+1} , respectively. The following adjustments are done at the end of each time step:

a) If $T_l^{n+1} \geq M_l^{n+1}$, then

$$(C_l^{n+1})_{adjusted} = C_{IS}^{n+1} \quad \text{and,} \quad 4.101a$$

$$(P_l^{n+1})_{adjusted} = P_l^{n+1} \{Eq. 4.100\} + (\phi S_w)^{n+1} (C_l^{n+1} - C_{IS}^{n+1}), \quad \text{if } C_l^{n+1} > C_{IS}^{n+1} \quad 4.101b$$

or,

$$(P_l^{n+1})_{adjusted} = P_l^{n+1} \{Eq. 4.100\} - (\phi S_w)^{n+1} (C_l^{n+1} - C_{IS}^{n+1}), \quad \text{if } C_l^{n+1} < C_{IS}^{n+1} \quad 4.101c$$

b) If $T_l^{n+1} < M_l^{n+1}$, then

$$(C_l^{n+1})_{adjusted} = C_l^{n+1} \{Eq. 4.96\} + \frac{P_l^{n+1} \{Eq. 4.100\}}{(\phi S_w)^{n+1}} \quad 4.102$$

where n+1 refer to the new time step. From Equations 101 and 102, we notice that the method is explicit only in the sense that the exchange (adjustment) between the dissolved and the precipitate mass of the species is taking place at the end of the time step.

Even though the explicit method is non-iterative and precipitate calculation requires almost no additional computational time, it may suffer from two drawbacks:

* Even though dispersion and adsorption are neglected in the current use (Equation 4.96), the numerical implementation of the precipitation models is taking into consideration the NUTS capabilities described in Equations 4.38 and 4.39.

- 1) Because the concentration is adjusted at the end of the time step, the concentration profile for the grid blocks that has no precipitate and adjacent to the non-depleted grid blocks, may show some zigzagging behavior.
- 2) In some extreme case when the flow into precipitate-depleted intrusion borehole is mainly coming from the repository, the concentration may not reach the solubility limit even if the influxes are all at the solubility limit. Consequently, the releases via the borehole into the overlying geological units may be underestimated.

Implicit treatment. In the implicit treatment the system of equations to be solved is Equation 4.96 subjected to the constraint,

$$C_i^{n+1} \leq C_{IS}^{n+1} \quad 4.103$$

and the solid phase equation

$$-q + \sum_{j=1}^{J_i} \lambda_{ij} P_{ij} - \lambda_i P_i = \frac{\partial P_i}{\partial t} \quad 4.104$$

where P is defined as mass precipitate per grid block volume (kg/m^3).

The system of Equations 4.96, 4.103, and 4.104 is solved in NUTS using the following procedure:

- 1) At the beginning of each time step, initialize the precipitate in each grid block from the old time step as follows:

$$P_i^k = P_i^n \quad 4.105$$

where k and n refers to the first iteration and old time step indices, respectively.

- 2) For each grid block i with $P_i^k > 0$, and assuming that enough precipitate is available to maintain the solubility limit, set Dirichlet condition by the penalty method⁺ at a concentration equal to the solubility limit. Then solve Equation 4.96 for the concentration, C_i^{k+1} in the whole computational domain.
- 3) From the computed concentration in step 2, evaluate the precipitate/dissolution source/sink term required to maintain Dirichlet condition (only in the grid blocks with $P_i^k > 0$) by solving for the residual of Equation 4.96 as follows:

⁺ Becker, E.B., Cary, G.F., and J.T. Oden (1981): *Finite Elements: An Introduction*, Volume 1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

$$q_i^{k+1} = \frac{\partial}{\partial t}(\phi S_w C_i) + \nabla \cdot v_w C_i - Q^* + (\phi S_w C_i) \lambda_i - \phi S_w \sum_{p=1}^{J_i} C_{i,p} \lambda_p \quad 4.106$$

Equation 4.106 is solved in the discretized form. Since all the entries for this equation are available after the solution of Equation 4.96, the value of the residual (q_i^{k+1}) can be directly solved for by substituting of the solubility limit for the concentration of the grid block i and the computed concentration (from Equation 4.96) for the surrounding grid blocks (five-point stencil in two-dimensional grid).

- 4) Evaluate the new precipitate, P_i^{k+1} from Equation 4.104, using q_i^{k+1} from step 3.
- 5) Check the sign of the P_i^{k+1} ,
 - a) If $P_i^{k+1} < 0$, then the precipitate available is not enough to maintain the solubility limit. Therefore, set $P_i^{k+1} = 0$, $q_i^{k+1} = \frac{P_i^k}{\Delta t}$, and turn Dirichlet condition off in that grid block.
 - b) If $P_i^{k+1} \geq 0$, then the precipitate available is enough to maintain the solubility limit. Therefore, leave Dirichlet condition active in that grid block.
 - c) Because some grid blocks with no precipitate ($P_i^n = 0$) may acquire precipitate due to dissolved daughter isotopes ingrowth, compare the computed concentration, C_i^{k+1} in each grid block with the solubility limit. If $C_i^{k+1} > C_{IS}^{n+1}$ then,

$$P_i^{k+1} = S_w \phi (C_i^{k+1} - C_{IS}^{n+1}) \quad 4.107$$

Please notice that this positive value of P_i^{k+1} will activate the Dirichlet condition in step 2.

- 6) Evaluate the change in the precipitate as follows:

$$\Delta P_i = P_i^{k+1} - P_i^k \quad 4.108$$

- 7) If $|\Delta P_i| \leq \epsilon$, where ϵ is a small tolerance, the procedure is done, otherwise update q and P and go to step 2.

It is worth mentioning here that in the cases when multiple isotopes compete on the same solubility limit, the solubility limit of each isotope is evaluated as:

$$C_{IS}^{n+1} = F^{n+1} C_e \quad 4.109$$

where,

F = mole fraction of the isotope and
C_e = element solubility limit (kg/m³ brine)

Knowing that F is a function of C_i ($F = f(C_i)$), the system of Equations 4.96 and 4.103 becomes highly nonlinear⁺ and may not converge in the implicit treatment (in the explicit treatment, F^{n+1} is known from C_i^{n+1}). One recommended way to overcome this problem is by linearization of Equation 4.106 through assigning a solubility limit for each isotope. In the CCA calculations, both isotopes ²³⁸Pu and ²³⁹Pu share the solubility of the plutonium and may require such a kind of treatment.

⁺ The source of the nonlinearity is the form of the function F. This function, for example, in three-isotope system in which only C₁ and C₂ compete on the same solubility is, $F_1 = \frac{C_1}{C_1 + C_2}$, $F_2 = \frac{C_2}{C_1 + C_2}$, and $F_3 = 1.0$.

Therefore, for C₁ and C₂, the function is rational and hence nonlinear. Please notice that this function does not depend on the concentration of other isotopes not competing on the same solubility.

5.0 INHERENT CAPABILITIES AND LIMITATIONS OF THE SOFTWARE

NUTS's inherent capabilities and limitations are listed below.

1. The flow and transport equations are decoupled, which is permitted only if solute concentrations are dilute. Decoupling does not affect fluid density, viscosity, or other related physical properties.
2. Local equilibrium is assumed to exist between interactive phases.
3. The radioisotopes transported by NUTS are assumed to be thermodynamically stable within their solubility limits. No gas phases or radioisotope-rich liquid phases are allowed. However, precipitation is possible and is calculated when and if solute concentrations exceed their solubility limits.
4. Dissolved radioisotopes are assumed to be in equilibrium with their precipitates. Precipitation is treated as a reversible process and is controlled by the value of solute concentration compared to the solubility limit, which defines the upper bound for dissolution.
5. Sorption is assumed to be controlled entirely by one of three equilibrium isotherms.
6. Dispersion is assumed to obey Fick's second law with dispersivities appropriate to an isotropic porous medium. Thus, the dispersivity tensor reduces to two numerical constants for the two principal flow directions.
7. The solubility of each isotope is evaluated as follows:

$$\text{Isotopic Solubility} = \text{Mole Fraction of that particular isotope} \times \text{the elemental solubility}$$
8. Gas-liquid equilibrium is assumed to obey the linear relationship represented by Equation 4.92, with temperature dependence given by Equation 4.93.
9. Because gases travel much faster than liquids, the liquid phase is regarded as a retarding agent to the constituents transported by the gas phase.

These and other inherent capabilities and limitations of the model are discussed item for item in Sections 4.0 and 6.0, and in the examples.

6.0 USER INTERACTIONS WITH THE SOFTWARE

6.1 Exercising NUTS Interactively

Assuming that the user is running under the PA environment (the symbols and the logicals have to be set up), NUTS may be exercised either interactively or through command files. To exercise NUTS interactively, type "NUTS" at the VMS \$ prompt and strike the carriage return. The title page and disclaimer will scroll by on screen. The program will then prompt the user with a series of questions including the names of a sequence of required input and output files. The sequence of queries is addressed in order in Section 6.2. The files themselves are discussed in detail in the remainder of this chapter, and illustrative examples are given in the appendices. If all NUTS's prompts are answered satisfactorily, the final "carriage return" will cause NUTS to exercise. If the run is successful, a "normal completion" notice will appear on screen. If not, an error message will result.

NUTS may be compiled on PC, VAX, and UNIX systems with minimal effort. However, for the WIPP CCA PA, NUTS will operate exclusively in a cluster of Alpha-VAX microcomputers. Consequently, the other applications will not be discussed explicitly in this manual. However, to gain the flexibility to run on different platforms, NUTS was intentionally designed to run in two slightly different configurations. The first is called the "stand-alone" configuration and is written NUTS-SA. The second, designed specifically for the WIPP's Compliance Assessment Methodology Controller (CAMCON), is called CAMCON configuration and is written NUTS-CC. The main differences between the two are (1) the types of libraries with which each compiles, (2) the flexibilities associated with the two sets of input/output capabilities, and (3) that NUTS-SA and NUTS-CC have two *different* executables. NUTS-CC will be used exclusively for the WIPP CCA PA. However, because NUTS-CC's operational procedures are a subset of NUTS-SA, we will begin with a discussion of the Stand-Alone input/output features. That discussion will be followed with the requirements for and limitations of NUTS-CC.

6.2 User-Interactive Options Afforded by NUTS-SA

NUTS requires the following three types of Input/Output:

1. A User-Interactive Input Sequence
2. A Parameter Statement
3. Input/Output Files

These three types of input/output* are reviewed in the three subsections that follow.

* The units adopted in NUTS are SI units unless specified otherwise.

6.3 User-Interactive Input Sequence

A limited number of control flags are available for user specification during the course of executing the program. These controls are part of the input and mainly specify the following information:

- a) Names, sources, and types of the I/O files (ASCII, binary, etc.).
- b) The type of porous media to be modeled (single-, or dual-porosity and/or permeability).
- c) The number of phases in the transporting medium (usually 1), and the type of phase (gas or liquid).
- d) The type of the compiler and terminal (for QA purposes).

The following is a description of user-interactive input*:

6.3.1 Is This a DEBUG Run? (Y/N) {N}

Accepts Y, N, y, or n. Press the carriage-return key to select the default option.

6.3.2 The Default Input-File Name is NUTS.IN

Do You Want To Use This Name (Y/N)--> {NUTS.IN}

Accepts Y, N, y, & n. Press the carriage-return key to select the default option.

- a. If the answer is "Y" or "y" the program will proceed to the next question.
- b. If the answer is "N" or "n", the program will prompt the following:

Enter NUTS Input File Name

The user may specify an input-file name up to 80 characters in length and strike the carriage-return key, wherein the sequence of questions will proceed.

If instead, the user strikes the "Return key" the following will appear on screen:

The Default Input File Name NUTS.IN

Is Assigned To The Input File

Do You Like To Change It (Y/N)

Accepts Y, N, y, or n. Press the carriage-return key to select the default option.

If the answer is "Y", the program will start again from "b". Otherwise it will proceed.

If an invalid input filename is entered, the program will deliver an error message and terminate.

6.3.3 The Default Output File Name is NUTS.OUT

Do You Want To Use This Name (Y/N)--> {NUTS.OUT}

Accepts Y, N, y, or n. Press the carriage-return key to select the default option.

* Quantities listed within braces are the default-option replies to program queries. They are selected by pressing the carriage return.

Depending on the answer, the program will proceed exactly as in (6.3.2). If an original filename is assigned, be sure to use a character*80 name.

6.3.4 Read the material map from BRAGFLO input file (Y/N)? {Y}

Accepts Y, N, y, or n. Press the carriage-return key to select the default option.

- a. If the answer is "Y" or "y", NUTS will read the material map from BRAGFLO's input file**.
- b. If the answer is "N" or "n", NUTS will expect to read the material map from its own ASCII input file.

6.3.5 Is This a Test Run? (Y/N) {N}

Accepts Y, N, y, or n. Press the carriage-return key to select the default option.

- a. If the answer is "Y" or "y", NUTS will ask the following:

Enter name of TEST file

The user may specify an original filename up to 80 characters in length.

- b. If the answer is "N" or "n", the program will prompt the following:

Enter BRAGFLO input binary/CDB file name.

The user may specify an original filename up to 80 characters in length.

NUTS will then inquire:

Will BRAGFLO data be read from CDB or binary+ ?

Enter BIN for binary or CDB for CDB {BIN}

Accepts BIN, bin, Bin, CDB, cdb, or Cdb. Press the carriage-return key to select the default option.

If an invalid filename is entered (whether a test-file or BRAGFLO-file name), the program will issue an error message and abort.

6.3.6 Enter Type of Output File {ASC}

The following output files are allowed:

1. **ASC for ASCII file**
2. **BIN for binary file**
3. **CDB for CAMCON database**
4. **ASC-BIN for ASCII and binary files**
5. **ASC-CDB for ASCII and CDB files**
6. **BIN-CDB for binary and CDB files**
7. **ASC-BIN-CDB for ASCII, binary, and CDB files**

Accept ASC, BIN, CDB, ASC-BIN, ASC-CDB, BIN-CDB, or ASC-BIN-CDB†.

**NUTS expects to open a BRAGFLO ASCII input file that has the same name as BRAGFLO's binary file, but has an INP extension

+ NUTS has the ability to read BRAGFLO binary outputs before and after post-processing into CDB

† Because of some technical limitation in CAMCON system, we recommend using CDB option in NUTS_CC version of the code.

The case of the characters (either upper or lower case) and the sequence of the combination of the file types is not restricted; i.e., ASC-BIN, BIN-ASC, asc-bin, or bin-asc will all be allowed. However, the combination between upper and lower cases is not allowed (Example: ASC-bin is not allowed).

If "BIN" is among the options required by the user, NUTS will prompt:

Enter Binary Output Format

The following types are allowed

- 1. Enter BRAG OR brag FOR BRAGFLO Output Format**
- 2. Enter NUTS OR nuts FOR NUTS Output Format**

Accepts BRAG, brag, NUTS, or nuts

6.3.7 Enter Type of Porous Medium {MATRIX}

The following types are allowed:

- 1. "FRACTURE, Fracture, fracture, FRACT, fract, Fract, F, or f " for fracture calculations.**
- 2. "MATRIX, Matrix, matrix, mat, MAT, Mat, M, or m" for matrix calculations.**
- 3. "DUAL-POROSITY, dual-porosity, Dual-porosity, DP, Dp, dp, D-P, d-p, or D-p" for dual-porosity.**
- 4. " DUAL-PERMEABILITY, dual-permeability, DPM, dpm, D-PM, d-pm, Dpm, or D-pm" for dual-permeability.**

Accepts FRACTURE, Fracture, fracture, FRACT, fract, Fract, F, f; MATRIX, Matrix, matrix, mat, MAT, Mat, M, m; DUAL-POROSITY, dual-porosity, Dual-porosity, DP, Dp, dp, D-P, d-p, or D-p; DUAL-PERMEABILITY, dual-permeability, DPM, dpm, D-PM, d-pm, Dpm, or D-pm. Press the carriage-return key to select the default option.

6.3.8 Enter Number of Phases {1}

Accepts an integer ≥ 1

6.3.9 Enter the Type of Phase (Liquid/Gas) {L}

The following types are allowed:

- 1. LIQUID, liquid, Liquid, L, l, LIQ, liq for liquid phase**
- 2. GAS, gas, Gas, G, g, GS, gs for gas phase**

Accepts LIQUID, liquid, Liquid, L, l, LIQ, liq; GAS, gas, Gas, G, g, GS, or gs. Press the carriage-return key to select the default option.

6.3.10 Enter the Type of Compiler (character*32)

The following are allowed:

- 1. Enter VAX for VAX SYSTEM**
- 2. Enter UNIX for UNIX SYSTEM**
- 3. Enter PC for PC LAHEY SYSTEM**

Accepts VAX, vax, UNIX, unix, PC, or pc.

6.3.11 Enter the name of the computer system

Ex: CRAY, GATEWAY2000, ALPHA, etc.

6.4 NUTS's Parameter Statement

NUTS uses an INCLUDE file (NUT_PARAM.INC) that contains a single **PARAMETER** statement to set the dimensions of different kinds of arrays used in the computations. Each parameter value is allotted space based on the needs of the WIPP's PA calculations. For larger problems, the parameters of NUT_PARAM.INC can be easily adjusted. As in any parameter statement, the size of the array may be greater (but not less) than that required by the problem. An example of a parameter statement used for Compliance Certification Application (CCA) calculations is as follows:

```
C
C      Note* :
C      For two-dimensional problems, SET MBW ≥ {(MAX(MX,MY)+1)* # OF PHASES* # OF
C                                     CONTINUUM - # OF CONTINUUM]*2 + 1} * 2

      INTEGER
+      MX, MY, MZ, NB, NS,
+      NC, NP, NDR, MBW, NFVARA, NMVARA,
+      NFVARB, NMVARB, NFVARC, NMVARC,
+      NVARTIT, NVPR, MXHIV, MVHIV,
+      MGVAR, NMATM, MMATTIME, MWASTE,
+      IHNTVARB, IHGNTVARB,
+      MXPOINT, MYPOINT
      PARAMETER
+      (
+      MX          = 200,
+      MY          = 40,
+      MZ          = 1,
+      NB          = 8000,
+      NS          = 2,
+      NC          = 9,
+      NP          = NC,
+      NDR         = 2,
+      MBW         = 410,
+      NFVARA      = 14,
+      NMVARA      = 14,
+      NFVARB      = 14,
+      NMVARB      = 14,
+      NFVARC      = 14,
+      NMVARC      = 14,
+      NVARTIT     = 28,
+      NVPR        = 300,
+      MXHIV       = 540,
+      MVHIV       = MXHIV,
+      MGVAR       = 10,
+      NMATM       = 80,
+      MMATTIME    = 50,
+      MWASTE      = 50,
+      IHNTVARB    = 28*NC,
+      IHGNTVARB   = 10*NC,
+
+      MXPOINT     = 30,
+      MYPOINT     = 30)
```

* This statement calculates a size of numerical matrix suitable for two-point upstream weighting option. For other options use half this size

The above parameter statement variables are declared to satisfy the maximum memory requirement anticipated in the CCA calculations. In general, numbers smaller than 1 are not allowed in these declarations. The definition of the above-mentioned parameters is given in Table 2 for the variables and Table 3 for the constants⁺.

⁺ These parameters are used in the parameter statement and should not be changed.

Table 2. Variables used in NUT_PARAM.INC

Parameter	Description
MX	Maximum number of grid blocks in x-direction.
MY	Maximum number of grid blocks in y-direction.
MZ	Maximum number of grid blocks in z-direction.
NB	Maximum total number of the grid blocks (MX*MY*MZ)
NS	Maximum number of radioactive sites.
NC	Maximum number of isotopes or constituents.
NP	Maximum number of parents for each isotopes.
NDR	Maximum number of rock density ranges.
MBW	Maximum numerical matrix band width.
NVPR*	Number of element array maps from BRAGFLO.
MXHIV*	Maximum total number of history variables allowed in BRAGFLO.
MVHIV*	Maximum number of history variables per output distribution in BRAGFLO.
MGVAR*	Number of global variables printed out in an ASCII or binary BRAGFLO output file.
NMATM	Maximum number of materials specified in BRAGFLO
MMATTIME	Maximum number of times material map is specified in BRAGFLO
MWASTE	Maximum number of waste regions.
IHNTVARB**	Maximum total number of history variables to be printed in BRAGFLO style binary file.
IHGNTVARB**	Maximum total number of global variables to be printed in BRAGFLO style binary file.
MXPOINT	Maximum number of rows allowed in an interpolation table
MYPOINT	Maximum number of columns allowed in an interpolation table

* These parameters are related to BRAGFLO output and relevant only when BRAGFLO binary output is dealt with directly by NUTS (not through the CAMCON).

** These parameters are relevant only when NUTS binary output is to be post-processed with BRAGFLO binary output

Table 3. Constants used in NUT_PARAM.INC

Parameter	Description
NFVARA	Number of fracture arrays to be output in an ASCII file = 14.
NMVARA	Number of matrix arrays to be output in an ASCII file = 14.
NFVARB	Number of fracture arrays to be output in a binary file = 14.
NMVARB	Number of matrix arrays to be output in a binary file = 14.
NFVARC	Number of fracture arrays to be output in a CDB file = 14.
NMVARC	Number of matrix arrays to be output in a CDB file = 14.
NVARTIT	Number of descriptive variables = 28.

6.5 Input/Output Files Used by NUTS-SA

NUTS is capable of accommodating different combinations of input/output files in each run, depending upon the user's specifications and requirements. We will classify these files as input, output, and debug* files. The convention generally used in naming NUTS's input/output files is to form a root name by concatenating the names of the flux-field input file and the radioisotope input file, and then adding any of ASC, BIN, or DBG as extensions. A subset or the entire list of these files may be opened in a single application. The files are briefly listed below, and then described in considerable detail in the subsections that follow.

1. An ASCII radioisotope input file
2. An ASCII flux-field input file
3. A binary flux-field input file (output from BRAGFLO) if not provided as an ASCII
4. Binary output files
 - a. BRAGFLO-type binary output file (to be post-processed by BRAGFLO's post-processor)
 - b. NUTS-type binary output file

* Strictly speaking, debug files are text output files. However, we prefer to classify them separately.

5. ASCII output file
 - a. Extensive output file
 - b. Specific ranges output file
6. Debug ASCII output files

An expanded description of NUTS's input and output files follows.

6.5.1 ASCII Radioisotope Input File

This file consists of pieces of information provided by different procedures and controlled by the flags listed in the NUTS's ASCII input file, and also in the user interactive input. The input can be categorized as:

- 1) General input the user will encounter in all NUTS input files (isotope-related input).
- 2) Input related to the type of the porous media encountered (Continuum-related input).
- 3) Material-properties-map input.

The subsections that follow, numbered **6.5.1.n**, where **n** is various combinations of letters and numbers, are a sequential description of the ASCII radioisotope input file. Samples of ASCII input files are given in Appendix A.

6.5.1.A Program Controls and Flags

The very first part of the ASCII-radioisotope (A/R) input file is mainly flags that control both subsequent inputs and the computation itself. To simplify the readability of the manual, these flags and most of the description that follows will be presented in a form resembling computer language.

6.5.1.A1 Site Description and Flags

Line 1. Descriptor: NUTS title of the run. *Accepts a character string up to 100 characters as a descriptive line.*

Line 2. NUTS_TITLE.

NUTS_TITLE: This is a descriptive line for the run. The title is saved and concatenated with BRAGFLO's titles when they are printed out, which are simple clues for the user as to which NUTS and BRAGFLO runs were considered. *Accepts a character string up to 100 characters in length.*

Line 3. Descriptor: Number of sites and materials. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 4. NSITES, NMAT_INPUT

NSITES: is the number of contamination sites used in the run. Any grid block or combination of grid blocks can be a site. Thus, this is a useful representation if similar constituents from different grid blocks are to be tracked or the overlap of contamination from different places is to be assessed. *Accepts an integer 0 < NSITES ≤ the total number of the grid blocks.*

NMAT_INPUT: is the number of porous materials encountered in the domain. It describes the number of materials having different properties such as porosity, permeability, density, etc. This number is compared in NUTS with the number read from BRAGFLO ASCII input file while reading the material map. If the two numbers are different, NUTS will output an aborting message and stop. *Accepts an integer 0 < NMAT_INPUT ≤ the total number of the grid blocks.*

DO I = 1, NSITES

Line 5. SITE_NAME(I), NCOMPONENT(I)

SITE_NAME: represents the name of the site to be modeled. *Accepts a 20 character string.*

NCOMPONENT: is the number of radioactive components to be modeled in each site. Similar components from different sites or parents are considered to be distinct, i.e., if ²³⁸U exists at two sites, it will be counted as a separate constituent at each site.

NCOMPONENT is the total number of all components that pre-exist, plus any that are anticipated to appear as a result of the decay process. *Accepts an integer > 0.*

END DO

Line 6. Descriptor: 1. Site name, 2. Component, Daughter, Parent, and Group names. *Accepts a character string up to 100 characters in length as a descriptive line.*

M = 0

DO I = 1, NSITES

Line 7. SNAME(I)

SNAME: a dummy variable for the SITE_NAME. SITE_NAME is renamed as a precaution in case of an input error, which, if it occurs, will not affect the stored name. *Accepts a 20-character string.*

DO J = 1, NCOMPONENT(I)

M = M + 1

**Line 8. COMPONENT_NAME(I,M), DAUGHTER_NAME(M),
PARENT_NAME(M), GROUP_NAME(M)**

COMPONENT_NAME: the name of the radioactive isotope⁺ . *Accepts a 20-character string.*

DAUGHTER_NAME : for isotopes that decay into a daughter, this variable refers to the name of the next isotope in the decay chain. For constituents that do not undergo decay or whose half-lives are greater than the simulation time, the character string 'NONE' in upper case should be used. *Accepts a 20-character string.*

PARENT_NAME: the name of the preceding isotope in the decay chain. If the component has no parent, the character string 'NONE' in upper case should be used. *Accepts a 20-character string.*

GROUP_NAME: Refers to the element name of the isotope. *Accepts a 20-character string.*

END DO
END DO

Note: In the Do Loop of Line 8, it is very important to strictly follow the chain from the top down to the last member. The branches in diverging or converging chains should be treated as a separate chain. This may require an adjustment in the decay constant, inventory, and/or the name of the component at which the convergence or the divergence occurs. It is also necessary in the above entries to be consistent in the spelling and the letter case of the component, daughter, parent and group names throughout the rest of the input, except "NONE" that should be used in upper case.

Line 9. Descriptor: 1. Number of element, 2. element name, temperature dependency of solubility and table look-up solubility. *Accepts a character string up to 100 characters as a descriptive line.*

Line 10. NOELEMENT

NOELEMENT: Number of elements. *Accepts an integer $>0 < \text{NOELEMENT} \leq$ the total number of isotopes.*

DO I = 1, NOELEMENT

Line 11. ELEMENT_NAME(I), ELTEMP_SOLB(I), SOLB_TABLE(I)

ELEMENT_NAME: the element's name. It should be identical to GROUP_NAME. *Accepts a 20-character string.*

ELTEMP_SOLB: a logical flag that indicates the temperature dependency of the solubility of the element. *Accepts T or F.*

SOLB_TABLE: a logical flag to indicate whether or not the solubility of the element will be provided by a table. *Accepts T or F.*

⁺ Throughout this manual, isotope, radioactive isotope, component, and constituent are used interchangeably. They all refer to the chemical species to be modeled and only their properties will define whether that species is radioactive or not.

END DO

Note: A classical definition of the element is a substance that can not be decomposed by chemical means into simpler substances. However, in this manual, element is used to refer to the ionic state of all compounds of an element in more than one oxidation state that compete on the same solubility.

Line 12. Descriptor: Colloidal transport (T/F). *Accepts a character string up to 100 characters as a descriptive line.*

Line 13. COLLOID

COLLOID: a logical flag to indicate whether colloidal transport is intended. *Accepts a T or F.*

If the answer above is T, then enter Line 14 to Line 18.

Line 14. Descriptor: 1. Number of colloids
2. location of the colloid in Line 8.
3. location of the dissolved species of the same kind.
4. preferential solubility.

Accepts a character string up to 100 characters as a descriptive line.

Line 15. NCOLLOID

NCOLLOID: Number of colloids to be transported. *Accepts an integer $0 < \text{NCOLLOID} \leq$ the total number of isotopes.*

Line 16. (LOC_COLLOID(I), I = 1, NCOLLOID)

LOC_COLLOID: Location of the colloid in the input sequence of line 8 of Section 6.5.1.A1. *Accepts an integer $0 < \text{LOC_COLLOID} \leq$ the total number of isotopes.*

Line 17. (IDIS(LOC_COLLOID(I)), I = 1, NCOLLOID)

IDIS: Location of the dissolved species competing with the colloid, in going to solution, in the input sequence of line 8 of Section 6.5.1.A1. *Accepts an integer $0 < \text{IDIS} \leq$ the total number of isotopes.*

Line 18. (PREF_SOLB(LOC_COLLOID(I)), I = 1, NCOLLOID)

PREF_SOLB: Logical flag that identifies whether the colloid has preference to go to solution. Accepts a T or F.

Note: Colloid transport in NUTS is used in a simple form. Even though the colloids are treated like the none-colloid species in having a distinguished name and properties, transport with colloid has two different treatments. The first treatment is the ability to scale the velocity of the brine by a scaling factor. This will adjust the velocity field for the whole transport process. The second treatment is the preferential solubility for the colloids.

The preferential solubility is given to the colloid only at the inventory limit. Therefore, if a dissolved and a colloid ^{238}U starts with some part in solution and some precipitate, at the inventory limit, the colloid will use up its own precipitate first, then the precipitate of the dissolved part of ^{238}U followed by the non-colloid dissolved ^{238}U and up to the solubility limit of the colloid each time.

Line 19. Descriptor: pH required (Y/N). *Accepts a character string up to 100 characters as a descriptive line.*

Line 20. PHREQ

PHREQ: Flag related to brine-pH information. It is associated with the solubility calculation. pH will be required if correlations are to be used to determine the solubility of an element. *Accepts Y, y, N, or n.*

Line 21. Descriptor: Order of the numerical method. *Accepts a character string up to 100 characters as a descriptive line.*

Line 22. MORDER:

MORDER: Order of the method. The following options are allowed:
MORDER =1, for one-point upstream winding.
MORDER =2, for two-point upstream winding.
MORDER =3, for split-operator two-point upstream winding.
MORDER =4, for mid-point weighting.
Accepts an integer $0 < \text{MORDER} \leq 4$.

Line 23. Descriptor: Degree of implicitness. *Accepts a character string up to 100 characters as a descriptive line.*

Line 24. BETA2:

BETA2: A multiplier used in the differential equation to weight the contribution from the explicit and implicit parts of the transport equation. The following values of BETA2 are of special interest:
BETA2 = 1 , for fully implicit.
BETA2 = 0 , for fully explicit.
BETA2 = 1/2 , for Cranck-Nicolson.
Accepts a real number $0 \leq \text{BETA2} \leq 1$.

Line 25. Descriptor: Explicit of implicit precipitate numerical treatment (T/F). *Accepts a character string up to 100 characters as a descriptive line.*

Line 26. EXIMPRECIP:

EXIMPRECIP: A logical flag to specify whether implicit or explicit precipitate is used. In the explicit precipitate treatment, the concentration is compared with the solubility limit at the end of the time step. The

implicit method, on the other hand, is an iterative and constrains the concentration to the solubility limit. For more information, the reader is referred to Section 4.5.3 of this manual.

EXIMPRECIP accept two values, "false" for the explicit and "true" for the implicit treatment.

If implicit precipitate is true, then

Line 26. NITR, TOL

NITR: Number of iteration in the implicit treatment. *Accept an integer NITR>0*

TOL: Convergence tolerance. *Accept a real number TOL>0*

6.5.1.A2 Single-Porosity Fracture Controls and Flags

These controls are applied if a single-porosity, fracture medium is to be modeled:

Fracture Sorption Controls

Line 1. Descriptor: Is fracture sorption required (Y/N)
Accepts a character string up to 100 characters in length as a descriptive line.

Line 2. ADSTYPEF

ADSTYPEF: This character identifies whether or not sorption is required.

Accepts Y, y, N, or n.

If the above answer is 'Y' or 'y', then input lines 3 and 4.

Line 3. FRAC_ISOOTHERM

FRAC_ISOOTHERM: Type of sorption isotherm. The following are considered by NUTS:

Linear isotherm. *Accepts L or l.*

Freundlich isotherm. *Accepts F, or f.*

Langmuir isotherm. *Accepts LA, la, or La.*

DO I = 1, NUCLIDE ' NUCLIDE = Total number of constituents'

Line 4. NAME(I), FSORPTION(I), FADSTEMPDEP(I)

NAME: Dummy variable for the name of the component.

Accepts a 20-character string.

FSORPTION: Character to identify whether the component is sorbable.

Accepts ADSORP for sorbable constituents, or NON_ADSORP for non-sorbable constituents.

FADSTEMPDEP: Logical flag for temperature dependency of the fracture sorption isotherm. *Accepts T, or F.*

END DO

Fracture Dispersion Controls

Line 5. Descriptor: Does dispersion take place within the fracture (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 6. FDISPREQ

FDISPREQ: Fracture-dispersion flag.
Accepts Y, y, N, or n.

Line 7. Descriptor: Do you have symmetrical dispersion in the fracture (F/T) and will data be read from NUTS ASCII (F/T).
Accepts a character string up to 100 characters in length as a descriptive line. The input on Line 8 is required only when the answer to line 6 is Y or y.

Line 8. FSYMDISP, FDSPNUTINPUT

FSYMDISP: Logical flag to specify whether mechanical dispersion is symmetrical or only downstream (default = downstream).
Accepts T or F.

FDSPNUTINPUT: Logical flag to specify whether the following dispersion data will be read from NUTS ASCII input (if the flag is true) and from the CDB (if the flag is false) (default = NUTS input).
Accepts T or F.

Fracture Source Controls

Line 9. Descriptor: Does injection/production occur in the fracture (Y/N)
Accepts a character string up to 100 characters in length as a descriptive line.

Line 10. FSTATUSINJ

FSTATUSINJ: Character to inform NUTS injection/production is to be activated in the fracture. *Accepts Y, y, N, or n.*

Line 11. Descriptor: Do Dirichlet boundary conditions apply in the fracture (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 12. FDIRICHLET

FDIRICHLET: Logical flag to inform NUTS that Dirichlet boundary condition will be considered in some computational nodes. *Accepts T or F.*

Note: Dirichlet boundary conditions in NUTS are implemented by using the penalty method⁺. In this method, the main diagonal of the linear equation of the grid block is multiplied by a large number, β and the right-hand side of the same equation is replaced by $\beta * C_{dir}$, where C_{dir} is Dirichlet boundary condition concentration. As long as the multiplier

⁺ Becker, E.B., Cary, G.F., and J.T. Oden (1981): *Finite Elements: An Introduction*, Volume 1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

is a very large number, the concentration of the grid block will be maintained at Dirichlet condition.

Fracture Initial Concentration Controls

Line 13. Descriptor: Is manual initialization required in the fracture (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 14. FMAN_CONC_INIT

FMAN_CONC_INIT: Logical flag to inform NUTS that the concentration in the fracture is initialized manually.
Accepts T or F.

Fracture Printed Element Variables Controls

This module specifies the format (ASCII, BINARY or CAMCON DATA BASE [CDB]) in which fracture concentration, precipitation, etc. to be represented in the output files.

If an ASCII file is to be output, then:

Line 15. Descriptor: Variables to be printed as an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. (IPRNTFA(I), I = 1, NFVARA)

IPRNTFA: This integer array has fourteen flags for the fracture-element variables to be printed as an ASCII file.
Choose 1 to print the variable and 0 to omit the variable from printing.

This flag will activate or deactivate printing of the following variables:

- IPRNTFA(1): Fracture total mass of dissolved constituents in the brine in each grid block (FTMPG of the D-N in the brine)⁺, kg.
- IPRNTFA(2): Fracture total mass of precipitated constituents in each grid block (FTMPG of the P-N), kg.
- IPRNTFA(3): Fracture total mass of dissolved, precipitated and sorbed constituents in each grid block (FTMPG of the D-N, P-N, and A-N), kg.
- IPRNTFA(4): Fracture total dissolved curies of all constituents in each grid block (FTCPG of the D-N in the brine), curies.

⁺ In IPRNTFA(1 to 14), the text in the bracket refers to the name of the variable in the output file.

- IPRNTFA(5): Fracture total precipitated curies of all constituents in each grid block (FTCPG of the P-N), curies.
- IPRNTFA(6): Fracture total dissolved, precipitated, and sorbed curies of all constituents in each grid block (FTCPG of the D-N, P-N, and A-N), curies.
- IPRNTFA(7): Fracture volumetric concentration of the dissolved isotope (FVCPG of the D-N in the brine), kg/(m³ brine).
- IPRNTFA(8): Fracture dissolved mass in the brine of a certain constituent (FMPG of the D-N in the brine), kg.
- IPRNTFA(9): Fracture precipitated mass of a certain constituent (FMPG of the P-N), kg.
- IPRNTFA(10): Fracture soil base concentration of the isotope (FMPG of the D-N, P-N, and A-N), mg/(kg soil).
- IPRNTFA(11): Fracture curies of volumetric concentration of a certain constituent (FCPG of the nuclide VC), curies/(m³ brine).
- IPRNTFA(12): Fracture curies of dissolved mass in the brine of a certain constituent (FCPG of the D-N), curies.
- IPRNTFA(13): Fracture curies of precipitated mass of a certain constituent (FCPG of the P-N), curies.
- IPRNTFA(14): Fracture curies of dissolved, precipitated and sorbed mass of a certain constituent (FCPG of the D-N, P-N, and A-N), curies.

Line 17. Descriptor: Spatial range (s) of data to be printed in an ASCII file for specific variable.
Accepts a character string up to 100 characters in length as a descriptive line.

Line 18. ASC_FPRINT_RANGE

ASC_FPRINT_RANGE: Allows a range(s) of data to be output in an ASCII file.
Accepts Y, y, N, or n.

If a BINARY* file is to be output, then:

Line 19. Descriptor: Variables to be printed to a BINARY file. *Accepts a character string up to 100 character in length as a descriptive line.*

Line 20. (IPRNTFB(I), I = 1, NFVARB)

IPRNTFB: integer array similar to IPRNTFA(I) has fourteen flags for the element fracture variables to be printed in a binary file.
Choose 1 to print the variable, and 0 to omit printing.

If a spatial range(s) is to be printed (line 18 is "Y" or 'y'), then:

* Lines 19 and 20 are required for the CDB output in the NUT-CC version of the code.

Line 21. Descriptor: Ranges to be printed in an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 22. NAFR:

NAFR: Number of ranges to be printed. *Accepts an integer $0 < \text{NAFR} \leq$ the total number of grid blocks.*

DO I = 1, NAFR

Line 23. IARANGE(I), MFASTRTI(I), MFASTRTJ(I), MFASTRTK(I), MFAENDI(I), MFAENDJ(I), and MFAENDK(I)

IARANGE: Range number. *Accepts an integer $0 < \text{IARANGE} \leq \text{NAFR}$.*
MFASTRT_{I,J,K} I,J,K indices of the grid block starting the range
Accepts an integer > 0 and $I \leq \text{NX}$, $J \leq \text{NY}$, $K \leq \text{NZ}$.
MFAENDI,J,K I,J,K indices of the grid block ending the range
Accepts an integer > 0 and $I \leq \text{NX}$, $J \leq \text{NY}$, $K \leq \text{NZ}$.

Line 24. (IPRNSFA(I,J), J = 1, NFVARA)

IPRNSFA: integer array similar to IPRNTFA

END DO

6.5.1.A3 Single-Porosity Matrix Controls and Flags

These controls are applied if a single-porosity matrix is to be modeled.

Matrix Sorption

Line 1. Descriptor: Is matrix sorption required (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. ADSTYPEM

ADSTYPEM: This character activates or deactivates sorption in the matrix.
Accepts Y, y, N, or n. If the answer is 'Y' or 'y' then:

Line 3. MAT_ISOOTHERM

MAT_ISOOTHERM: one of three types of matrix sorption isotherm, namely:
Linear isotherm. *Accepts L or l.*
Freundlich isotherm. *Accepts F, or f.*
Langmuir isotherm. *Accepts LA, la, or La.*

DO I = 1, NUCLIDE

Line 4. NAME(I), MSORPTION(I), MADSTEMPDEP(I)

NAME: Dummy variable for the name of the component. *Accepts a 20-character string.*

MSORPTION: Character to identify whether the component is sorbable.

Accepts ADSORP for sorbable or NON_ADSORP for non-sorbable constituents.

MADSTEMPDEP: logical flag for temperature dependency of the matrix sorption isotherm. *Accepts T or F.*

END DO

Matrix Dispersion Controls

Line 5. Descriptor: Do you have dispersion in the matrix (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 6. MDISPREQ

MDISPREQ: Matrix-dispersion flag. *Accepts 'Y' or 'y' to consider, and 'N' or 'n' to omit dispersion in the matrix.*

Line 7. Descriptor: Do you have symmetric dispersion in the matrix (F/T), and will data be read from NUTS ASCII (F/T). *Accepts a character string up to 100 characters in length as a descriptive line. The input on line 8 is required only when the answer to line 6 is Y or y.*

Line 8. MSYMDISP, MDSPNUTINPUT

MSYMDISP: Logical flag to specify whether mechanical dispersion is symmetric or only downstream (default = downstream). *Accepts T or F.*

MDSPNUTINPUT: Logical flag to specify whether the following dispersion data will be read from NUTS ASCII input (if the flag is true) and from the CDB (if the flag is false) (default = NUTS input). *Accepts T or F.*

Matrix Source Controls

Line 9. Descriptor: Is injection/production present in the matrix (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 10. MSTATUSINJ

MSTATUSINJ: Informs NUTS whether or not injection/production is present in the matrix. *Accepts Y, y, N, or n.*

Line 11. Descriptor: Are Dirichlet boundary conditions present in the matrix (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 12. MDIRICHLET

MDIRICHLET: Logical flag to inform NUTS that Dirichlet boundary condition will be considered at some computational nodes. *Accepts T or F.*

Matrix Initial Concentration Controls

Line 13. Descriptor: Is manual initialization present in the matrix (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Information Only

Line 14. MMAN_CONC_INIT

MMAN_CONC_INIT: Logical flag informing NUTS that the concentration in the matrix is initialized manually. *Accepts T or F.*

Matrix Initial Concentration from NUTS Undisturbed CDB

In the CCA calculations, two intrusion times are considered in BRAGFLO: 350 and 1000 years. In probabilistic evaluation of the intrusion times, times other than 350 and 1000 years need to be considered for the transport purposes. This step has been done by interpolating the flow fields results from BRAGFLO for 350 years to perform the transport with an intrusion that occurs at 100 years. Similarly, the flow fields of the 1000 years intrusion are extrapolated to intrusions that take place at 3000, 5000, 7000, and 9000 years. To account for the fact that the period before the intrusion is undisturbed, the output CDB from undisturbed NUTS transport calculations is used to initialize the calculation at that specific time of intrusion.

Line 15. Descriptor: Concentration is initialized from NUTS undisturbed CDB (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. CONC_CDB_INITIALIZATION

CONC_CDB_INITIALIZATION: Logical flag informing NUTS that the concentration in the matrix is initialized from the undisturbed CDB. *Accepts T or F.*

Matrix Printed Element Variable Controls

This module specifies the output-file format (ASCII, BINARY or CAMCON DATA BASE [CDB]) of results for matrix concentration, precipitation, etc.

If an ASCII file is to be output, then:

Line 17. Descriptor: Variables to be printed in an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 18. (IPRNTMA(I), I = 1, NMVARA)

IPRNTMA: An integer array containing fourteen flags for the matrix-element variables to be printed in an ASCII file.
Choose 1 to print and 0 not to print the variable in question.
IPRNTMA(I) refers to following variables:

IPRNTMA(1): Matrix total mass of dissolved constituents in the brine in each grid block (MTMPG of the D-N in the brine)⁺, kg.

IPRNTMA(2): Matrix total mass of precipitated constituents in each grid block (MTMPG of the P-N), kg.

⁺ In IPRNTMA(1 to 14), the text in the bracket refers to the name of the variable in the output file.

- IPRNTMA(3): Matrix total mass of dissolved, precipitated and sorbed constituents in each grid block (MTMPG of the D-N, P-N, and A-N), kg.
- IPRNTMA(4): Matrix total dissolved curies of all constituents in each grid block (MTCPG of the D-N in the brine), curies.
- IPRNTMA(5): Matrix total precipitated curies of all constituents in each grid block (MTCPG of the P-N), curies.
- IPRNTMA(6): Matrix total dissolved, precipitated, and sorbed curies of all constituents in each grid block (MTCPG of the D-N, P-N, and A-N), curies.
- IPRNTMA(7): Matrix volumetric concentration of the dissolved isotope (MVCPG of the D-N in the brine), kg/(m³ brine).
- IPRNTMA(8): Matrix dissolved mass in the brine of a certain constituent (MMPG of the D-N in the brine), kg.
- IPRNTMA(9): Matrix precipitated mass of a certain constituent (MMPG of the P-N), kg.
- IPRNTMA(10): Matrix soil base concentration of the isotope (MMPG of the D-N, P-N, and A-N), mg/(kg soil).
- IPRNTMA(11): Matrix curies of volumetric concentration of a certain constituent (MCPG of the nuclide VC), curies/(m³ brine).
- IPRNTMA(12): Matrix curies of dissolved mass in the brine of a certain constituent (MCPG of the D-N), curies.
- IPRNTMA(13): Matrix curies of precipitated mass of a certain constituent (MCPG of the P-N), curies.
- IPRNTMA(14): Matrix curies of dissolved, precipitated and sorbed mass of a certain constituent (MCPG of the D-N, P-N, and A-N), curies.

Line 19. Descriptor: Spatial range (s) of data to be printed in an ASCII file for specific variables. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 20. ASC_MPRINT_RANGE

ASC_MPRINT_RANGE: This character *accepts Y, y, N, or n* answers for the range(s) of data to be output in an ASCII file.

If a BINARY* file is to be output, then:

Line 21. Descriptor: Matrix variables to be printed in a BINARY file. *Accepts a character string up to 100 characters in length as a descriptive line.*

* Lines 21 and 22 are required for the CDB output in the NUT-CC version of the code.

Line 22. (IPRNTMB(I), I = 1, NMVARB)

IPRNTMB: A fourteen-flag integer array similar to IPRNTMA(I)
Choose 1 to print, or 0 to omit the element matrix variables in a binary file.

If spatial range(s) is to be printed (Line 20 is "Y" or "y"), then:

Line 23. Descriptor: Ranges to be printed in an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 24. NAMR

NAMR: Number of ranges to be printed. *Accepts an integer $0 < \text{NAMR} \leq$ the total number of grid blocks.*

DO I = 1, NAMR

Line 25. IARANGE(I), MASTRTI(I), MASTRTJ(I), MASTRTK(I), MAENDI(I), MAENDJ(I), MAENDK(I)

IARANGE: Range number. *Accepts an integer $> 0 < \text{IARANGE} \leq \text{NAMR}$.*

MASTRTI,J,K: I,J,K Indices of the grid block starting the range. *Accepts an integer > 0 and $I \leq \text{NX}$, $J \leq \text{NY}$, $K \leq \text{NZ}$.*

MAENDI,J,K: I,J,K indices of the grid block ending the range. *Accepts an integer > 0 and $I \leq \text{NX}$, $J \leq \text{NY}$, $K \leq \text{NZ}$.*

Line 26. (IPRNSMA(I,J), J = 1, NMVARA)

IPRNSMA: an integer array similar to IPRNTMA

END DO

6.5.1.A4 Dual-Porosity/Dual-Permeability Controls and Flags

If a dual-porosity or a dual-permeability medium is to be modeled, the flag sequence will be as follows:

- a. Enter the flags described in Section 6.5.1.A2, Line 1 to Line 24
- b. Enter the flags described in Section 6.5.1.A3, Line 1 to Line 8
- c. Enter the following flags for

Fracture/Matrix Dispersion

Line 8a. Descriptor: Is fracture/matrix dispersion present (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 8b. MFDISPREQ

MFDISPREQ: Fracture/matrix dispersion flag.

Accepts 'Y' or 'y' if dispersion from matrix to fracture is present and 'N' or 'n' if it is not.

- d. Enter the flags described in Section 6.5.1.A3, Line 9 to Line 26.

6.5.1.A5 Sorption Equilibrium Constant (k_d) Temperature-Dependency Information

This module provides NUTS with information about the constituents in which k_d is temperature dependent.

Line 1. Descriptor: Component name, reference temperature, and sorption exponential coefficient. *Accepts a character string up to 100 characters in length as a descriptive line.*

IF either ADSTYPEF or ADSTYPEM (above) was selected as 'Y' or 'y', then:

DO I = 1, NUCLIDE

IF either FADSTEMPDEP(I) or MADSTEMPDEP(I) (above) was selected as "TRUE," then:

Line 2.NAME(I), REFTEMPKD(I), ADSEXP_COEFF(I)

NAME: Dummy variable for the name of the constituent. *Accepts a 20-character string.*

REFTEMPKD: The reference temperature at which k_d is evaluated. *Upper and lower bounds are not provided.*

ADSEXP_COEFF: Sorption exponential coefficient. This parameter is defined as $\frac{\Delta H^0}{R}$,

where H^0 is the enthalpy of sorption at reference temperature T^0 , and R is the gas constant (see Equation 4.91). *Upper and lower bounds are not provided.*

END DO

6.5.1.A6 Molecular-Diffusion Temperature-Dependency Flags

These flags inform NUTS whether the molecular diffusion of a given constituent is temperature dependent in cases where **FDISPREQ** or **MDISPREQ** or **MFDISPREQ** is required ('Y' or 'y'). If none is required, no information will be provided by this item.

If FDISPREQ or MDISPREQ or MFDISPREQ is 'Y' or 'y', then:

Line 1. Descriptor: Component, name, and molecular-diffusion temperature dependency flag. *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NUCLIDE

Line 2.NAME(I), DMOLTEMDEP(I)

NAME: Dummy variable for the name of the constituent. *Accepts a 20-character string.*

DMOLTEMDEP: Logical flag for temperature dependency of the molecular diffusion. *Accepts T or F.*

END DO

Information Only

6.5.1.A7 Printing-Frequency Information

This module specifies the frequency at which results will be printed to ASCII, BINARY, or CAMCON DATA BASE (CDB) output files.

If an ASCII file is to be output, then:

Line 1. Descriptor: Printing frequency to an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. IPRFRQA, TAMAX

IPRFRQA: Frequency of printing to an ASCII file, e.g.: 1 if every time step is to be printed, 2 for every other time step and so on. *Accepts an integer > 0.*

TAMAX: Maximum time between printings. If this time is exceeded, NUTS will print the results regardless of the frequency specified above. *Accepts a real number $0 < TAMAX \leq$ the total simulation time.*

If a BINARY file is to be output, then:

Line 3. Descriptor: Printing frequency to a binary file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 4. IPRFRQB, TBMAX

IPRFRQB: Frequency of printing to binary file. *Accepts an integer > 0.*

TBMAX: Maximum time allowed between printing interval. *Accepts a real number $0 < TBMAX \leq$ the total simulation time.*

If a CDB file is to be output, then:

Line 5. Descriptor: Printing frequency to a CDB file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 6. IPRFRQC, TCMAX

IPRFRQC: Frequency of printing to a CDB file. *Accepts an integer > 0.*

TCMAX: Maximum time between printings. *Accepts a real number $0 < TCMAX \leq$ the total simulation time.*

6.5.1.A8 External Source Flag

This flag identifies whether there is a dynamic source interfacing with NUTS. Two kinds of dynamic sources are used in NUTS. The first kind is an Actinide Source Submodel that has the ability to look at the distribution of the waste canisters, their corrosion and breach of the radioactive material, the chemical reaction of the waste with the brine, and the transport of the

waste (by diffusion) from the breached canister to the porous media. This repository-scale calculation can be conducted through an interface between NUTS and the Actinide Source Submodel and is not used in the CCA calculations. Another dynamic source model is used in NUTS for testing purposes. This source is a sort of interpolating function that has the ability to change the form (spike, slab, triangle, double hump, etc.) and the amount of the release. This function is built in NUTS and used to generate the source in many of the test cases used in the code evaluation (see the QA documentation package NUTS).

Line 1. Descriptor: Is external source interfacing present (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. STOCKMAN

STOCKMAN: Logical flag to inform NUTS about the external source model. *Accepts T or F.*

6.5.1.A9 Time-Minimum Limit

When a high degree of non-linearity is encountered in BRAGFLO calculations, the automatic time stepping module in BRAGFLO will reduce the time step to whatever value leads to the solution convergence. This value in some occasions could be as low as a fraction of a second. Because NUTS reads the time step information from the CDB, which is stored in single precision, and because of the precision (time is read and written in seconds), the difference between two successive times (time step size) for such a highly nonlinear step, can be zero. This inaccuracy in the time information is limited to a few time steps and irrelevant for the transport calculations. However, in the mathematical formulation of the transport equation, the time step appears as a divisor; in these instances, a division by zero problem will halt the calculation. To avoid this problem a time step minimum limit in the order of 10^{-16} or less is used whenever zero time step is encountered.

Line 1. Descriptor: Input timestep lower limits. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. TIMELIMIT

TIMELIMIT: Time lower limit. *The value should not exceed 1.E-16 s*

6.5.1.A10 Intrusion Time Interpolation and Extrapolation

Line 1. Descriptor: Actual time of intrusion, Interpolated/Extrapolated time, and Tolerance. *Accepts a character string up to 100 characters in length as a descriptive line.*

If concentration initialization is to be done from the CDB (CONC_CDB_INITIALIZATION is true) then:

Line 2. TIME_INTRUSION, RESET_TIME_INTRUSION, TIMETOL

TIME_INTRUSION: The actual intrusion time specified in BRAGFLO run (350 and 1000 yrs for the CCA calculations). *Accept real number >0.*

RESET_TIME_INTRUSION: Interpolated/Extrapolated intrusion time. *Accept real number >0.*

TIMETOL:* Accepted tolerance to initialize at closest time to the **RESET_TIME_INTRUSION**. *Accept real number >0.*

6.5.1.B Material Map Input

NUTS's interactive input will determine the location from which NUTS will read the material map. If the user specified **BRAGFLO** input as the source for the material map, NUTS will open BRAGFLO's input file and read the material map and various reference condition parameters. NUTS assumes BRAGFLO's input and output files have the same name with an extension **.INP** for input and **.BIN** or **.CDB** for the output. If the user specified NUTS as a source for the material map, NUTS will proceed reading the input **file in the following sequence:**

Line 1. Descriptor: Number of times for specifying material map. *Accepts a character string up to *17 characters as a descriptive line.*

Note: NUTS will key on this label to start reading the material map, regardless of whether BRAGFLO or NUTS input is specified. Because of the format of BRAGFLO input, **this label should be used exactly preceded by two or three empty spaces (see the example in Appendix B).**

Line 2. NMATTIMES

NMATTIMES: Number of times the material map will change during simulation time.
Accepts an integer > 0.

DO I = 1, NMATTIMES

Line 3. Descriptor(I): Start time for material map. *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 4. TIMEMAST(I)

TIMEMAST: Starting time for material map (s).
Accepts a real number > 0.

Line 5. Descriptor(I): Material-type grid map. *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 6.(((IMAT(I,J,K), I = 1, NX), J = 1, NY), K = 1, NZ)

IMAT(I,J,K): An array that holds the material map of the grid blocks for the entire spatial domain, where NX, NY, and NZ are the number of

* NUTS will read the time in the CDB file and add the tolerance. If the total of time + TIMETOL \geq RESET_TIME_INTRUSION, this time will be picked up for initialization.

the grid blocks in x-,y-, and z-directions, respectively. *Accepts an integer > 0.*

END DO

Line 7. Descriptor: Material name. *Accepts a character string up to 17 characters in length as a descriptive line.*

DO I = 1, NMAT

Line 8. J, MAT_NAME(I)

J: Material number in the material map. *Accepts an integer.*

MAT_NAME: Material name. *Accepts a 50-character string.*

Note: NMAT is the number of the material specified in the material map. NMAT is compared with the input material number (NMAT_INPUT). If the two numbers do not agree, *NUTS will print a warning message and abort.*

END DO

Line 9. Descriptor: Number of waste regions. *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 10. NWST

NWST: Number of waste regions. If the $NWST > NSITES = 1$, NUTS will combine the waste regions into one site. In the cases when $NWST > NSITES > 1$, NUTS will print a warning message on screen and it will terminate.

Accepts an integer > 0 and = NSITES.

Line 11. Descriptor:

Names of waste regions.

Accepts a character string up to 17 characters in length as a descriptive line.

Line 12+ . (MAT_WASTE(I), I = 1, NWST)

Line 13. (MAT_WASTE(I), I = 1, NWST)

MAT_WASTE: Waste region material index. *Accepts any integer.*

If the material map input is from BRAGFLO, NUTS will key on ' REFERENCE TEMPE' to read the following:

Line 13. RTEMP_BRAG, RPRES_BRAG

RTEMP_BRAG: Reference temperature.

Accepts $273.15 \leq RTEMP_BRAG \leq 473.15$

⁺ In BRAGFLO input, there is two sets for the waste region indices. The first set (Line 12) refers to the waste material regions before placing the waste, whereas the second set designates the waste material regions after placing the waste. The first set is used in BRAGFLO for the five years initialization from time = -5 years to zero. At time = 0 years, the waste will be placed which in BRAGFLO is done by changing the material map of the waste to the second set of indices. NUTS is only concerned about the second set of indices, the reason why the same variable name (MAT_WASTE) is used in Lines 12 and 13.

RPRES_BRAG: Reference pressure.
Accepts $0 \leq \text{RPRES_BRAG} \leq 1.E10$

If the material map input is from BRAGFLO, NUTS will also key on " SALT% DEN" to read:

Line 14. A1, A2, A3, A4, A5

A1: Reference brine density.
A2, A3, and A4: BRAGFLO parameters irrelevant to NUTS applications.
A5: Brine compressibility.

Note: A1 and A2 are not checked directly, however they are used to calculate density under reservoir conditions. Since the density bounds are checked, A1 and A2 are implicitly checked.

If the material map input is from BRAGFLO, NUTS will also key on "# LAMBDA" to read:

DO I=1, NMAT

Line 15, IA1, A2, BRESD_MAT(I)

IA1, A2: BRAGFLO parameters irrelevant to NUTS applications.
BRESD_MAT: Brine residual saturation of the material.

END DO

An example of the material map with a 9-grid-block spatial domain for one time is given in Appendix B.

6.5.1.C Properties Input

6.5.1.C1 Physical Properties

Line 1. Descriptor: End of material map and start of physical properties input. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. Descriptor: Element solubility input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS's input file and not the CDB (this flag is set in the user interactive input . As in Appendix H, "CDB" refers NUTS to the CDB input file to read components properties, and "N" refers to NUTS ASCII input file as the source for the input), then:

DO I = 1, NOELEMENT

If the solubility is not temperature dependent and not table look-up (ELTEMP_SOLB(I) and SOLB_TABLE(I) are false) then:

Line 3. NAME(I), ALOG_ELEMNT_SOLB_LIMIT(I)

NAME: Dummy variable for the element name. *Accepts a 20-character string.*

ALOG_ELEMNT_SOLB_LIMIT: Log₁₀ of the element solubility limit (log mol/liter).

Accepts any real number for which Log₁₀ inverse exists.

END DO

DO I = 1, NOELEMENT

If the solubility is table look-up (SOLB_TABLE(I) is true) then:

Line 4. NAME(I),EQSPACED(I),ID_INT

NAME: Dummy variable for the element name. *Accepts a 20-character string.*

EQSPACED: Logical flag for equally spaced table entries. *Accepts T or F.*

ID_INT: Interpolation table dimensions identifier. *Accepts two values: ID_INT = 1 for 1D interpolation, and ID_INT = 2 for 2D interpolation tables*

If 1D table (solubility is a function of Eh) then:

Line 5. NROW1D

NROW1D: Number of rows in the table. *Accepts an integer > 0.*

DO K = 1, NROW1D

Line 6. ROW1D, FX

ROW1D: Row entry (refers to the Eh) in the table. *Accepts any real number.*

FX : Table entries (refers to the solubility, kg.mol/m³). *Accepts any real number.*

END DO

If 2D table (solubility is a function of pH and CO₂ pressure) then:

Line 7. NROW(I), NCOLUMN(I)

NROW: Number of rows in the table. *Accepts an integer > 0.*

NCOLUMN: Number of columns in the table. *Accepts an integer > 0.*

Line 8. (COLUMN(J,I), J = 1,NCOLUMN(I)

COLUMN: Column entry in two-dimensional table (represents the value of pH). *Accepts any real number.*

DO K = 1, NROW(I)

Line 9. ROW(K,I),(FX(K,NNC,I), NNC = 1, NCOLUMN(I))

ROW: Row entry in two-dimensional table (represents the value of CO₂ pressure in a unit consistent with BRAGFLO). *Accepts any real number.*

FX: Table entries (refers to the solubility, kg.mol/m³). *Accepts any real*

number.

END DO

END DO

Note: In the above table the units *must* be consistent.

Information Only

DO I = 1, NOELEMENT

If the solubility is temperature dependent (ELTEMP_SOLB(I) = .TRUE.) then:

Line 10. NAME(I), C0(I),C1(I),C2(I),C3(I),C4(I),C5(I)

NAME: Dummy variable for the element name. *Accepts a 20-character string.*

C0-5+ : Coefficients of the correlation polynomial. *Accepts any real number.*

END DO

Line 11. Descriptor: Name, molecular (or atomic) weight, inventory, half-life. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS's input file and not the CDB, then:

DO I = 1, NUCLIDE

Line 12. NAME(I),XMOLWT(I),CUINVCHD(I),CUINVRHD(I),HALF-LIFE(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

XMOLWT: Molecular or atomic weight of the component (kg-mol). *Accepts any real number $0 \leq XMOLWT \leq 1$.*

CUINVCHD: Curies of contact handled inventory (Ci). *Accepts any real number ≥ 0 .*

CUINVRHD: Curies of remote handled inventory (Ci). *Accepts any real number ≥ 0 .*

HALF-LIFE: Component half-life (s). *Accepts any real number ≥ 0 .*

Note1: In NUTS, contact and remote handled inventories are added and distributed volumetrically through out the waste region(repository). The total inventory is then converted from Ci to kg and used to initialize the concentration. NUTS locates the waste region through the input of section 6.5.1.C4. For stable constituents (non-radioactive), the values of these entries should be given in kg.

Note2: HALF-LIFE of "0" is used as a flag to identify stable components.

END DO

If phase type is gas, then:

Line 13. Descriptor: Component name, intercept and slope of equilibrium line. *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NUCLIDE

Line 14. NAME(I), EQCI(I), EQCS(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

EQCI: Intercept of gas-liquid equilibrium line (Equation 4.93). *Accepts any meaningful real number.*

+ This option is a fifth degree polynomial function to evaluate the solubility in the following form:

$$\ln S = C0 + \frac{C1}{T} + \frac{C2}{T^2} + C3T + C4 \ln(T) + C5pH$$
, where T refers to the temperature, and S is the solubility in kg/m³. The value of C0 to C5 should be evaluated from the solubility data.

Information Only

EQCS: Slope of gas-liquid equilibrium line (Equation 4.93). *Accepts any meaningful real number.*

END DO

Line 15. Descriptor: Ground-water pH Input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If PHREQ is 'Y' or 'y', then:

If the data are to be read from NUTS's input file and not the CDB, then:

Line 16. (RPH(I), I = 1, NMAT)

RPH: pH of ground water for the range specified by the material map index. *Accepts $0 \leq RPH \leq 14$.*

Line 17. Descriptor: Standard conditions brine density if a test run. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the run is a test run (densities can not be read from BRAGFLO), then:

Line 18. RBR_DEN

RBR_DEN: Standard condition brine density. *Accepts $0 \leq RBR_DEN \leq 2000$.*

6.5.1.C2 Component Molecular-Diffusion Input

Line 1. Descriptor: Molecular diffusion input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If MDISPREQ or FDISPREQ or MFDISPREQ is 'Y' or 'y', then:

If the data are to be read from NUTS input file and not the CDB, then input lines 2 and 3:

DO I = 1, NUCLIDE

Line 2. NAME(I), DMOL(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

DMOL: Molecular diffusion of the component (m^2/s). *Accepts a real number $0 \leq DMOL \leq$ a number that gives a molecular diffusional velocity⁺ of 0.6 m/s*

END DO

Line 3. Descriptor: Reference viscosity and temperature. *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NUCLIDE

If the molecular diffusion is temperature dependent (DMOLTEMDEP(I) is true) then:

C2.1.3 VISREF(I), TREF(I)

⁺ diffusional velocity is calculated as $= DMOL * distance$ (in the direction of the flow)/area (perpendicular to the direction of the flow). The value of 0.6 m/s is based on a Reynolds number = 1 in the porous medium, which assures a linear velocity dependence (i.e., satisfies Darcy's Law).

VISREF: Reference viscosity at which molecular diffusion is measured (Pa/s).

Accepts a real number > 0.

TREF: Reference temperature at which molecular diffusion is measured (K).

Accepts a real number > 0.

END DO

6.5.1.C3 Rock-Density Input

This module reads the density of each material mentioned in the material map.

Line 1. Descriptor: Rock density. *Accepts a character string up to 100 characters in length as a descriptive line.*

The density input is required if:

- If soil base concentration is required to be output (option IPRNTFA(10) or IPRNTFB(10) or IPRNTFC(10) or IPRNTMA(10) or IPRNTMB(10) or IPRNTMC(10) is equal to 1).
- If sorption is to be calculated (ADSTYPEM or ADSTYPEF is "Y" or "y").

and provided that data are to be read from NUTS input file and not the CDB, then,

Line 2. (RROCK_DENS(I), I = 1, NMAT)

RROCK_DENS: Rock grain density for the range specified by the material map index (kg/m³). *Accepts a real number $0 \leq RROCK_DENS \leq 5000$.*

6.5.1.C4 Waste-Matrix Input

In the cases where the waste is placed in a certain grid block belonging to a certain site, this input specifies the number of constituents, which constituents they are, and where in the grid-block map these constituents are located. It is this matrix that defines where to initialize the concentrations of the waste material.

Line 1. Descriptor: Waste matrix input. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. NWSTCOMP_TOTAL

NWSTCOMP_TOTAL: Number of Component placed in the waste. *Accepts an integer $0 \leq NWSTCOMP_TOTAL \leq$ the total number of components.*

If NWSTCOMP_TOTAL is zero, then stop waste matrix input here. Otherwise,

DO I = 1, NWSTCOMP_TOTAL

Line 3. NAME(I),LOCWASTINP(I),NWASTREGION(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCWASTINP: Location of the component in the input (COMPONENT_NAME list).
Accepts an integer $0 \leq \text{LOCWASTINP} \leq$ the total number of components.

NWASTREGION: Waste site number. *Accepts an integer $0 \leq \text{NWASTREGION} \leq \text{NSITES}$.*

END DO

Line 4. Descriptor: (1. site name, number of grid; 2. indices of waste matrix). *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NSITES

Line 5. SNAME(I),NGRIDSITE(I)

SNAME: A dummy character for the site name. *Accepts a 20-character string.*
NGRIDSITE: Number of grid block of waste in that particular site. *Accepts an integer $0 \leq \text{NGRIDSITE} \leq$ the total number of grid blocks.*

Line 6. (IS(I,J),JS(I,J),KS(I,J), J=1, NGRIDSITE(I))

IS,JS,KS: i,j, and k indices of the grid blocks in the waste site. *Accepts an integers $1 \leq \text{IS} \leq \text{NX}$, $1 \leq \text{JS} \leq \text{NY}$, and $1 \leq \text{KS} \leq \text{NZ}$.*

6.5.1.C5 Single-Porosity Fracture Input

If a single-porosity fracture medium is to be modeled, then:

Fracture Sorption Input

This module inputs the constituent sorption properties of the fracture. Three equilibrium isotherms are considered depending on the value of FRAC_ISOOTHERM.

Line 1. Descriptor: Fracture sorption. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS's input file and not the CDB, then:

If ADSTYPEF is 'N' or 'n', then stop the input for this module here. Otherwise,

If FRAC_ISOOTHERM is 'L' or 'l' then:

DO I = 1, NUCLIDE

Line 2. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 3. (RXLF(J), J = 1, NMAT)

RXLF: Linear sorption coefficient (k_d) of the fracture for the material specified by the material map index (m^3/kg). *Accepts a number > 1.*

END DO

If FRAC_ISOOTHERM is 'F' or 'f' then:

DO I = 1, NUCLIDE

Line 4. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 5.(RXFDCF(J), J = 1, NMAT)

RXFDCF: Freundlich distribution coefficient of the fracture for the material specified by the material map index (m^3 fluid/kg solid). *Upper and lower bounds are not specified.*

Line 6.(RXFCF(J), J = 1, NMAT)

RXFCF: Freundlich coefficient of the fracture for the material specified by the material map index (dimensionless). *Upper and lower bounds are not specified.*

END DO

If FRAC_ISOOTHERM is 'LA' or 'la' or 'La', then:

DO I = 1, NUCLIDE

Line 7. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 8. (RXLDCF(J), J = 1, NMAT)

RXLDCF: Langmuir distribution coefficient of the fracture for the material specified by the material map index (m^3 fluid/kg solid). *Upper and lower bounds are not specified.*

Line 9. (RXLCF(J), J = 1, NMAT)

RXLCF: Langmuir coefficient of the fracture for the material specified by the material map index (dimensionless). *Upper and lower bounds are not specified.*

END DO

Fracture Dispersion Input

This module inputs dispersion parameters for the fracture.

Line 10. Descriptor: Fracture dispersion. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS's input file and not the CDB, then:

If FDISPREQ is 'N' or 'n' stop the input for this module here. Otherwise,

Line 11. Descriptor: Longitudinal dispersivity in the fracture. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 12. (RALPHALF(I), I = 1, NMAT)

RALPHALF: Longitudinal dispersivity of the fracture for the material specified by the material map index (m). *Accepts a real number $0 \leq \text{RALPHALF} \leq$ the smallest hydrological dimension of the spatial domain.*

Line 13. Descriptor: Transverse dispersivity in the fracture. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 14. (RALPHATF(I), I = 1, NMAT)

RALPHATF: Transverse dispersivity of the fracture for the material specified by the material map index (m). *Accepts a real number $0 \leq \text{RALPHATF} \leq$ longitudinal dispersivity.*

Line 15. Descriptor: Tortuosity in the fracture. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. (RTORF(I), I = 1, NMAT)

RTORF: Fracture tortuosity for the material specified by the material map index (dimensionless). *Accepts a real number ≥ 1 .*

Fracture Source Input

Line 17. Descriptor: Start fracture source input for injection/production. *Accepts a character string up to 100 characters in length as a descriptive line.*

If FSTATUSINJ is 'N' or 'n', then skip the next Lines 18 to 25, otherwise,

Line 18. NTMNINJCOMP

NTMNINJCOMP: Total number of injected components. *Accepts an integer $0 < \text{NTMNINJCOMP} \leq$ total number of component*

DO J = 1, NTMNINJCOMP

Line 19. NAME(J), LOCINPM(J), NMTBLOKINJ(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCINPM: Location of the injected component in the component input sequence.

Accepts an integer $1 \leq \text{LOCINPM} \leq$ total number of component.

NMTBLOKINJ: Total number of grid blocks at which this component is injected.

Accepts an integer $1 \leq \text{NMTBLOKINJ} \leq$ total number of grid blocks.

Line 20. (INJBIM(I,J), INBJJM(I,J), INJBKM(I,J), I = 1, NMTBLOKINJ(J))

INJBIM: I index of the injection block. *Accepts an integer $1 \leq \text{INJBIM} \leq$ number of grid blocks in x direction.*

INBJJM: J index of the injection block. *Accepts an integer $1 \leq \text{INBJJM} \leq$ number of grid blocks in y direction.*

INJBKM: K index of the injection block. *Accepts an integer $1 \leq \text{INJBKM} \leq$ number of grid blocks in z direction.*

END DO

DO I = 1, NTMNINJCOMP

Line 21. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

**Line 22. (CONCMIJKINJ(INJBIM(J,I), INJBJM(J,I), INJBKM(J,I)), J = 1,
NMTBLOKINJ(I))**

CONCMIJKINJ: Concentration of the injected component (kg/m^3). *Accepts a real number ≥ 0 .*

END DO

Line 23. Descriptor: Injection production rates. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 24. NMBLOCKSOURCE, TIMEFSTRT, TIMEFEND

NMBLOCKSOURCE: Number of grid blocks having the source. *Accepts an integer $1 \leq \text{NMBLOCKSOURCE} \leq \text{total number of grid blocks}$.*

TIMEFSTRT: Starting time of injection/production in fracture (s). *Accepts a real number $0 \leq \text{TIMEFSTRT} \leq \text{TIMEFEND}$*

TIMEFEND: Ending time of injection/production in fracture (s). *Accepts a real number $0 \leq \text{TIMEFEND}$*

DO I = 1, NMBLOCKSOURCE

Line 25. IQIM(I), IQJM(I), IQKM(I), RATE(I)

IQIM, IQJM, IQKM: I,J,K indices of the injection/production block. *Accepts an integer $1 \leq \text{IQIM}, \text{IQJM}, \text{IQKM} \leq \text{total number of grid blocks in } x, y, z \text{ direction}$.*

RATE: Injection/production rate (m^3/s). *Accepts a real number ≥ 0 .*

END DO

Line 26. Descriptor: Dirichlet boundary condition in the fracture. *Accepts a character string up to 100 characters in length as a descriptive line.*

If FDIRICHLET then:

Line 27. NTMDIRCOMP, TYPEDIR

NTMDIRCOMP: Number of components with Dirichlet boundary condition (DBC). *Accepts an integer $1 \leq \text{NTMDIRCOMP} \leq \text{total number of component}$.*

TYPEDIR: Character to identify whether DBCs are specified in the repository only or may occur elsewhere in the domain. *Accepts two values; (1) GENERAL: DBCs occur only in the repository, or (2) NOT-GENERAL: DBCs may occur anywhere.*

DO J = 1, NTMDIRCOMP

If TYPEDIR is GENERAL or general, then:

Line 28. NAME(J), LOCMDIRINP(J), GCONDIR(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCMDIRINP: Location of the component in the input sequence. *Accepts an integer*
 $1 \leq \text{LOCMDIRINP} \leq \text{total number of component.}$

GCONDIR: Dirichlet B.C.s on Concentration (kg/m^3). *Accepts a real number ≥ 0 .*

Else If TYPEDIR is NOT-GENERAL, then:

Line 29. NAME(J), LOCMDIRINP(J), NMTDIRBLOK(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCMDIRINP: Location of the component in the input sequence. *Accepts an integer*
 $1 \leq \text{LOCMDIRINP} \leq \text{total number of components.}$

NMTDIRBLOK: Total number of grid blocks with D.B.C. *Accepts an integer*
 $1 \leq \text{NMTDIRBLOK} \leq \text{total number of grid blocks.}$

Line 30. (IDRBIM(I,J), IDRBJM(I,J), IDRBKM(I,J), I = 1, NMTDIRBLOK(J))

IDRBIM, IDRBJM, IDRBKM: I,J,K indices of the D.B.C grid blocks. *Accepts an integer*
 $1 \leq \text{IDRBIM}, \text{IDRBJM}, \text{IDRBKM} \leq \text{total number of grid blocks in } x, y, z \text{ direction, respectively.}$

END DO

DO I = 1, NTMDIRCOMP

Line 31. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 32. (DIRCONCMIJK(IDRBIM(J,I), IDRBJM(J,I), IDRBKM(J,I)), J = 1, NMTDIRBLOK(I))

DIRCONCMIJK: DBC on concentration of the ijk th component (kg/m^3). *Accepts a real number ≥ 0 .*

END DO

Fracture Dynamic Source Function Input

This module is created to generate a dynamic source function in the fracture. The shape of the source here can vary by changing the correlation parameters that specify the time range and the normalization factor. The form of the source function is given as $Q_0 * \hat{g}$, where Q_0 is the strength of the source in kg/s and \hat{g} is a dimensionless function. The parameter \hat{g} is an interpolated function calculated linearly in each interpolation range by the following formula:

$$\hat{g} = \frac{G_c(T - T_b) + G_b(T_c - T)}{T_c - T_b}$$

where,

T_b = interpolation range starting time, s

T_c = interpolation range ending time, s

T = simulation time , $T_b < T \leq T_c$
G_b = source normalization factor (multiplier) at time T_b
G_c = source normalization factor (multiplier) at time T_c

To add flexibility to the source shape, five interpolation ranges are allowed in the input. The source function is activated by the same flag used for the Actinide Source submodel. This source is usually needed in the testing processes outside the CAMCON environment.

Line 33. Descriptor: Testing dynamic source input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If (.NOT. STOCKMAN) skip Line 34 to Line 40.

Line 34. NTMDEPSRC

NTMDEPSRC: Total number of the radioactive isotopes that have time dependent source.
Accepts a real number $0 \leq \text{NTMDEPSRC} \leq \text{total number of component}$.

If NTMDEPSRC is zero skip line 35 to Line 40.

DO J = 1, NTMDEPSRC

Line 35. NAME(J), LOCSRCINP(J), NMTBLOKSRC(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*
LOCMDIRINP: Location of the component in the input sequence. *Accepts an integer $1 \leq \text{LOCMDIRINP} \leq \text{total number of components}$.*
NMTBLOKSRC: Total number of grid blocks with dynamic source. *Accepts an integer $1 \leq \text{NMTBLOKSRC} \leq \text{total number of grid blocks}$.*

Line 36. (ISRCI(I,J), ISRCJ(I,J), ISRCK(I,J), I = 1, NMTBLOKSRC(J))

ISRCI: I index of the grid block that has the source.
ISRCJ: J index of the grid block that has the source.
ISRCK: K index of the grid block that has the source.

Accepts an integer $1 \leq \text{ISRCI}, \text{ISRCJ}, \text{ISRCK} \leq \text{total number of grid blocks in } x, y, z \text{ direction, respectively}$.

END DO

DO J = 1, NTMDEPSRC

Line 37. NAME(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 38. (QC0IJK(ISRCI(I,J), ISRCJ(I,J), ISRCK(I,J), I = 1, NMTBLOKSRC(J))

QC0IJK: The source strength of the grid block in kg/s. *Accepts any real number.*

Line 39. ((TBIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TCIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TDIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TEIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TFIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J))),
I = 1, NMTBLOKSRC(J))

TBIJK: Start of time range 1.
TCIJK: Start of time range 2.
TDIJK: Start of time range 3.
TEIJK: Start of time range 4.
TFIJK: Start of time range 5.

Accepts a real number ≥ 0

Line 40.

((GBIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GCIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GDIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GEIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GFIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J))),
I = 1, NMTBLOKSRC(J))

GBIJK: Start of normalization factor range 1.
GCIJK: Start of normalization factor range 2.
GDIJK: Start of normalization factor range 3.
GEIJK: Start of normalization factor range 4.
GFIJK: Start of normalization factor range 5.

Accepts a real number ≥ 0

END DO

Fracture Concentration Initialization Input

Line 41. Descriptor: Initial concentration in the fracture if any. *Accepts character string up to 100 characters in length as a descriptive line.*

If FMAN_CONC_INIT is true, then:

Line 42. NINITCOMP

NINITCOMP: Number of components to be initialized. *Accepts $1 \leq \text{integer} \leq \text{total number of components}$.*

DO J = 1, NINITCOMP

Line 43. NAME(J),LOCINP(J)NBLOKINI(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCINP: Location of the component in the input sequence. *Accepts an integer*
 $1 \leq \text{LOCINP} \leq \text{total number of components.}$

NBLOKINI: Total number of grid blocks initialized. *Accepts an integer*
 $1 \leq \text{NBLOKINI} \leq \text{total number of grid blocks.}$

Line 44. (INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

NII, INIJ, INIK: I,J,K indices of the initialized grid blocks. *Accepts an integer*
 $1 \leq \text{INII}, \text{INIJ}, \text{INIK} \leq \text{total number of grid blocks in x, y, z direction, respectively.}$

END DO

DO J = 1, NINITCOMP

Line 45. NAME(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 46. CONCIJKINI(INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

CONCIJKINI: Initialized concentration of the grid. *Accepts a real number ≥ 0 .*

END DO

Fracture Colloid Transport Scaling Factors Input

Colloid scaling factors are arbitrary multipliers to scale up the velocity of the brine in the fracture. The new scaled velocity is therefore, $v_{fxs} = \theta v_{fx}$, where v_{fx} is the original brine x-direction velocity in the fracture, v_{fxs} is the x-direction scaled velocity in the fracture, and θ is the scaling factor. Similar manipulation can be considered in y- and z-direction velocities. This implementation is intended to account for the situations where colloid sorbed actinides may travel at a velocity higher than the velocity of dissolved actinides. **Because the whole velocity field is scaled up by this procedure, colloids should be handled in separate simulation runs.**

Line 47. Descriptor: Colloidal transport scaling factors. *Accepts a character string up to 100 characters in length as a descriptive line.*

If colloidal transport is required, then:

Line 48. Descriptor: Scaling factor for velocities in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 49. (SCALEX(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEX: Scaling factors of x-direction velocities. *Accepts an integer ≥ 1 .*

Line 50. Descriptor: Scaling factor for velocities in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 51. (SCALEY(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEY: Scaling factors of y-direction velocities. *Accepts an integer ≥ 1 .*

Line 52. Descriptor: Scaling factor for velocities in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 53. (SCALEZ(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEZ: Scaling factors of z-direction velocities. *Accepts an integer ≥ 1 .*

6.5.1.C6 Single-Porosity Matrix Input

If a single-porosity matrix is to be modeled, then:

Matrix Sorption Input

This module inputs the constituent sorption properties of the matrix. Three equilibrium isotherms are considered, depending on the value of MAT_ISOOTHERM.

Line 1. Descriptor: Matrix sorption. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS's input file and not the CDB, then:
If ADSTYPEM is 'N' or 'n', stop the input for this module here. Otherwise,

If MAT_ISOOTHERM is 'L' or 'l', then:

DO I = 1, NUCLIDE

Line 2. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 3. (RXLM(J), J = 1, NMAT)

RXLM: Linear sorption coefficient (k_d) of the matrix for the material specified by the material map index (m^3/kg). *Accepts a number ≥ 0*

END DO

If MAT_ISOOTHERM is 'F' or 'f' then:

DO I = 1, NUCLIDE

Line 4. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 5. (RXFDCM(J), J = 1, NMAT)

RXFDCM: Freundlich distribution coefficient of the matrix for the material specified by the material map index (m^3/kg). *Upper and lower bounds are not specified.*

Line 6. (RXFCM(J), J = 1, NMAT)

RXFCM: Freundlich coefficient of the matrix for the material specified by the material map index (dimensionless). *Upper and lower bounds are not specified.*

END DO

If MAT_ISOOTHERM is 'LA' or 'la' or 'La', then:

DO I = 1, NUCLIDE

Line 7. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Information Only

Line 8. (RXLDCM(J), J = 1, NMAT)

RXLDCM: Langmuir distribution coefficient of the Matrix for the material specified by the material map index (m^3 fluid/kg solid). *Upper and lower bounds are not specified.*

Line 9. (RXLCM(J), J = 1, NMAT)

RXLCM: Langmuir coefficient of the matrix for the material specified by the material map index (dimensionless). *Upper and lower bounds are not specified.*

END DO

Matrix Dispersion Input

This module inputs dispersion parameters for the Matrix.

Line 10. Descriptor: Matrix dispersion. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the data are to be read from NUTS input file and not the CDB, then:

If MDISPREQ is 'N' or 'n' stop the input for this module here, otherwise,

Line 11. Descriptor: Longitudinal dispersivity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 12. (RALPHALM(I), I = 1, NMAT)

RALPHALM: Longitudinal dispersivity of the matrix for the material specified by the material map index (m). *Accepts a real number $0 \leq RALPHALM \leq$ the smallest hydrological dimension of the spatial domain.*

Line 13. Descriptor: Transverse dispersivity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 14. (RALPHATM(I), I = 1, NMAT)

RALPHATM: Transverse dispersivity of the matrix for the material specified by the material map index (m). *Accepts a real number $0 \leq RALPHATM \leq$ longitudinal dispersivity.*

Line 15. Descriptor: Tortuosity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. (RTORM(I), I = 1, NMAT)

RTORM: Matrix tortuosity for the material specified by the material map index (dimensionless). *Accepts a real number ≥ 1 .*

Matrix Source Input

Line 17. Descriptor: Start matrix source input for injection/production. *Accepts a character string up to 100 characters in length as a descriptive line.*

If MSTATUSINJ is 'N' or 'n', then skip Lines 18 to 25. Otherwise,

Line 18. NTMNINJCOMP

NTMNINJCOMP: Total number of injected components. *Accepts an integer*

$0 < \text{NTMNINJCOMP} \leq \text{total number of components.}$

DO J = 1, NTMNINJCOMP

Line 19. NAME(J), LOCINPM(J), NMTBLOKINJ(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCINPM: Location of the injected component in the component input sequence.

Accepts an integer $1 \leq \text{LOCINPM} \leq \text{total number of components.}$

NMTBLOKINJ: Total number of grid blocks at which this component is injected.

Accepts an integer $1 \leq \text{NMTBLOKINJ} \leq \text{total number of grid blocks.}$

Line 20. (INJBIM(I,J), INBJM(I,J), INJBKM(I,J), I = 1, NMTBLOKINJ(J))

INJBIM: I index of the injection block. *Accepts an integer $1 \leq \text{INJBIM} \leq \text{total number of grid blocks in x direction.}$*

INBJM: J index of the injection block. *Accepts an integer $1 \leq \text{INBJM} \leq \text{total number of grid blocks in y direction.}$*

INJBKM: K index of the injection block. *Accepts an integer $1 \leq \text{INJBKM} \leq \text{total number of grid blocks in z direction.}$*

END DO

DO I = 1, NTMNINJCOMP

Line 21. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 22. (CONCMIJKINJ(INJBIM(J,I), INBJM(J,I), INJBKM(J,I)), J = 1, NMTBLOKINJ(I))

CONCMIJKINJ: Concentration of the injected component (kg/m^3). *Accepts a real number ≥ 0 .*

END DO

Line 23. Descriptor: Injection production rates. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 24. NMBLOCKSOURCE, TIMEFSTRT, TIMEFEND

NMBLOCKSOURCE: Number of grid blocks having the source.

Accepts an integer $1 \leq \text{NMBLOCKSOURCE} \leq \text{total number of grid blocks.}$

TIMEFSTRT: Starting time of injection/production in the matrix (s).

Accepts a real number $0 \leq \text{TIMEFSTRT} \leq \text{TIMEFEND.}$

TIMEFEND: Ending time of injection/production in the matrix (s).

Accepts a real number $0 \leq \text{TIMEFEND.}$

DO I = 1, NMBLOCKSOURCE

Line 25. IQIM(I), IQJM(I), IQKM(I), RATE(I)

IQIM, IQJM, IQKM: I,J,K indices of the injection/production block. *Accepts an integer $1 \leq \text{IQIM}, \text{IQJM}, \text{IQKM} \leq \text{total number of grid blocks in x, y, z direction, respectively.}$*

RATE: Injection/production rate (m^3/s). *Accepts a real number ≥ 0 .*

Information Only

END DO

Line 26: Descriptor: Dirichet boundary condition in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

If MDIRICHLET, then:

Line 27. NTMDIRCOMP, TYPEDIR

NTMDIRCOMP: Number of components with Dirchlet boundary condition (DBC).

Accepts $1 \leq \text{NTMDIRCOMP} \leq \text{total number of components}$.

TYPEDIR: Character to identify whether DBCs are specified in the repository only or elsewhere in the domain. *Accepts two values (1) GENERAL; for the whole repository DBC, or (2) NOT-GENERAL; for DBC specified in specific places.*

DO J = 1, NTMDIRCOMP

If TYPEDIR is GENERAL or general then:

Line 28. NAME(J), LOCMDIRINP(J), GCONDIR(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCMDIRINP: Location of the component in the input sequence. *Accepts an integer*

$1 \leq \text{LOCMDIRINP} \leq \text{total number of components}$.

GCONDIR: Dirchlet boundary condition on concentration (kg/m^3). *Accepts a real number ≥ 0 .*

Elseif TYPEDIR is NOT-GENERAL or not-general, then:

Line 29. NAME(J), LOCMDIRINP(J), NMTDIRBLOK(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCMDIRINP: Location of the component in the input sequence.

Accepts an integer $1 \leq \text{LOCMDIRINP} \leq \text{total number of components}$.

NMTDIRBLOK: Total number of grid blocks with D.B.C.

Accepts an integer $1 \leq \text{NMTDIRBLOK} \leq \text{total number of grid blocks}$.

Line 30. (IDRBIM(I,J), IDRBJM(I,J), IDRBKM(I,J), I = 1, NMTDIRBLOK(J))

IDRBIM, IDRBJM, IDRBKM: I,J,K indices of the D.B.C grid blocks. *Accepts an integer*

$1 \leq \text{IDRBIM}, \text{IDRBJM}, \text{IDRBKM} \leq \text{total number of grid blocks in } x, y, z \text{ direction, respectively.}$

Endif

END DO

DO I = 1, NTMDIRCOMP

Line 31. NAME(I)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 32. (DIRCONCMIJK(IDRBIM(J,I), IDRBJM(J,I), IDRBKM(J,I)), J = 1, NMTDIRBLOK(I))

DIRCONCMIJK: Concentration of the component having DBC (kg/m^3). *Accepts a real number ≥ 0 .*

END DO

Information Only

Matrix Dynamic Source Function Input

This module is created to generate a dynamic source function in the matrix. The shape of the source here can vary by changing the correlation parameters that specify the time range and the normalization factor. The form of the source function is given as $Q_0 * \hat{g}$, where Q_0 is the strength of the source in kg/s and \hat{g} is a dimensionless function. The parameter \hat{g} is an interpolated function calculated linearly in each interpolation range by the following formula:

$$\hat{g} = \frac{G_c(T - T_b) + G_b(T_c - T)}{T_c - T_b}$$

where,

- T_b = interpolation range starting time, s
- T_c = interpolation range ending time, s
- T = simulation time $> T_b$ and $\leq T_c$
- G_b = source normalization factor (multiplier) at time T_b
- G_c = source normalization factor (multiplier) at time T_c

To add flexibility to the source shape, five interpolation ranges are allowed in the input. The source function is activated by the same flag used for the Actinide Source submodel. This source is usually needed in the testing processes outside the CAMCON environment.

Line 33. Descriptor: Testing Dynamic source input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If (.NOT. STOCKMAN) skip Line 34 to Line 40.

Line 34. NTMDEPSRC

NTMDEPSRC: Total number of the radioactive isotopes that have time dependent source.
Accepts a real number $0 \leq \text{NTMDEPSRC} \leq \text{total number of component}$.

If NTMDEPSRC is zero skip line 35 to Line 40.

DO J = 1, NTMDEPSRC

Line 35. NAME(J), LOCSRCINP(J), NMTBLOKSRC(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*
LOCMDIRINP: Location of the component in the input sequence.
Accepts an integer $1 \leq \text{LOCMDIRINP} \leq \text{total number of components}$.
NMTBLOKSRC: Total number of grid blocks with dynamic source.

Accepts an integer $1 \leq \text{NMTBLOKSRC} \leq$ total number of grid blocks.

Line 36. (ISRCI(I,J), ISRCJ(I,J), ISRCK(I,J), I = 1, NMTBLOKSRC(J))

ISRCI: I index of the grid block that has the source.

ISRCJ: J index of the grid block that has the source.

ISRCK: K index of the grid block that has the source.

Accepts an integer $1 \leq \text{ISRCI}, \text{ISRCJ}, \text{ISRCK} \leq$ total number of grid blocks in x, y, z direction, respectively.

END DO

DO J = 1, NTMDEPSRC

Line 37. NAME(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 38. (QC0IJK(ISRCI(I,J), ISRCJ(I,J), ISRCK(I,J), I = 1, NMTBLOKSRC(J))

QC0IJK: The source strength of the grid block in kg/s. *Accepts any real number.*

Line 39.

**((TBIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TCIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TDIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TEIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
TFIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J))),
I = 1, NMTBLOKSRC(J))**

TBIJK: Start of time range 1.

TCIJK: Start of time range 2.

TDIJK: Start of time range 3.

TEIJK: Start of time range 4.

TFIJK: Start of time range 5.

Accepts a real number ≥ 0

Line 40.

**((GBIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GCIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GDIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GEIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J)),
GFIJK(ISRCI(I,J),ISRCJ(I,J),ISRCK(I,J))),
I = 1, NMTBLOKSRC(J))**

GBIJK: Start of normalization factor range 1.

GCIJK: Start of normalization factor range 2.

GDIJK: Start of normalization factor range 3.

GEIJK: Start of normalization factor range 4.

Information Only

GFIJK: Start of normalization factor range 5.

Accepts a real number ≥ 0

END DO

Matrix Concentration Initialization Input

Line 41. Descriptor: Initial concentration in the matrix if any. *Accepts a character string up to 100 characters in length as a descriptive line.*

If MMAN_CONC_INIT is true, then:

Line 42. NINITCOMP

NINITCOMP: Number of components to be initialized. *Accepts $1 \leq NINITCOMP \leq total$ number of components.*

DO J = 1, NINITCOMP

Line 43. NAME(J),LOCINP(J)NBLOKINI(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

LOCINP: Location of the component in the input sequence. *Accepts an integer $1 \leq LOCINP \leq total$ number of components.*

NBLOKINI: Total number of grid blocks initialized. *Accepts an integer $1 \leq NBLOKINI \leq total$ number of grid blocks.*

Line 44. (INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

INII, IINIJ, INIK: I,J,K indices of the initialized grid blocks. *Accepts an integer $1 \leq INII, IINIJ, INIK \leq total$ number of grid blocks in x, y, z direction, respectively.*

END DO

DO J = 1, NINITCOMP

Line 45. NAME(J)

NAME: Dummy variable for component name. *Accepts a 20-character string.*

Line 46. CONCIJKINI(INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

CONCIJKINI: Initialized concentration, kg/s. *Accepts a real number ≥ 0*

Matrix Colloid Transport Scaling Factors Input

Colloid scaling factors are arbitrary multipliers to scale up the velocity of the brine in the matrix. The new scaled velocity is therefore, $v_{mzs} = \theta v_{mx}$, where v_{mx} is the original brine x-direction velocity in the matrix, v_{mzs} is the x-direction scaled velocity in the matrix, and θ is the scaling factor. Similar manipulation can be considered in y- and z-direction velocities. This implementation is intended to account for the situations where colloid sorbed actinides may travel at a velocity higher than the velocity of dissolved actinides. **Because the whole velocity field is scaled up by this procedure, colloids should be handled in separate simulation runs.**

Line 47. Descriptor: Colloidal transport scaling factors. *Accepts a character string up to 100 characters in length as a descriptive line.*

If colloidal transport required, then:

Line 48. Descriptor: Scaling factor for velocities in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 49. (SCALEX(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEX: Scaling factors of x-direction velocities. *Accepts an integer ≥ 1 .*

Line 50. Descriptor: Scaling factor for velocities in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 51. (SCALEY(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEY: Scaling factors of y-direction velocities. *Accepts an integer ≥ 1 .*

Line 52. Descriptor: Scaling factor for velocities in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 53. (SCALEZ(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEZ: Scaling factors of z-direction velocities. *Accepts an integer ≥ 1 .*

6.5.1.C7 Dual-Porosity or Dual-Permeability Input

If the medium is dual-porosity or dual-permeability, then:

Input Section 6.5.1.C5, lines 1 to 53 and Section 6.5.1.C6, lines 1 to 53 following the same order and conditions mentioned above.

6.5.2 ASCII Flux-Field Input File (homogeneous)

An example of NUTS's ASCII flux-field input file for homogeneous properties is given in Appendix C.

Line 1. Descriptor: Is the input homogeneous (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. DUMP

DUMP: Logical flag to specify whether the following input is homogeneous. The value of this parameter is important only for CDB runs. *Accepts T or F.*

6.5.2.A Title Information

Line 3. Descriptor: Test run title. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 4. TEST_TITLE

TEST_TITLE: This is a descriptive line of the test run. The line is saved and concatenated with NUTS's title when printed out to give an idea about both NUTS and test runs considered. *Accepts up to a 100-character string.*

6.5.2.B Time Information

Line 5. Descriptor: Time step size and number of time steps. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 6. DT, NTIMESTEP

DT: Time step size (s). *Accepts a real number ≥ 0 .*

NTIMESTEP: Number of time steps required to run NUTS. *Accepts an integer ≥ 1 .*

6.5.2.C Dimension Information

Line 7. Descriptor: Number of grid blocks in x, y, z. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 8. NX, NY, NZ

NX: Number of grid blocks in x-direction. *Accepts an integer ≥ 1 .*

NY: Number of grid blocks in y-direction. *Accepts an integer ≥ 1 .*

NZ: Number of grid blocks in z-direction. *Accepts an integer ≥ 1 .*

Line 9. Descriptor: Dimension of grid blocks in x, y, z. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 10. DXTEST, DYTEST, DZTEST

DXTEST: Dimension of the grid block in x-direction (m). *Accepts a real number > 0 .*

DYTEST: Dimension of the grid block in y-direction (m). *Accepts a real number > 0 .*

DZTEST: Dimension of the grid block in z-direction (m). *Accepts a real number > 0 .*

6.5.2.D Fracture Information

If the medium to be modeled is fractured, then:

Line 11. Descriptor: Fracture porosity and saturation. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 12. PHIFINT, SWFINT

PHIFINT: Initial porosity of the fracture (fraction). *Accepts a real number $0 \leq PHIFINT \leq 1$.*

SWFINT: Initial saturation of the fracture (fraction). *Accepts a real number $0 \leq SWFINT \leq 1$.*

Line 13. Descriptor: Fracture Constant velocity field. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 14. TVELXF, TVELYF, TVELZF

TVELXF: Fracture velocity of the phase in x-direction (m/s). *Accepts a real number $TVELXF \leq 0.6$*

TVELYF: Fracture velocity of the phase in y-direction (m/s). *Accepts a real number*
 $TVELYF \leq 0.6$

TVELZF: Fracture velocity of the phase in z-direction (m/s). *Accepts a real number*
 $TVELZF \leq 0.6$

Line 15. Descriptor: Block temperature and brine viscosity. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. TTEMP_FRAC, TVISW_FRAC

TTEMP_FRAC: Temperature of the grid block (K). *Accepts a real number > 273.15*

TVISW_FRAC: Fracture brine viscosity (Pa/s). *Accept a real number > 0*

6.5.2.E Matrix Information

If the matrix to be modeled is a continuum, then:

Line 17. Descriptor: Matrix porosity and saturation. *Accept character string up to 100 chars as a descriptive line*

Line 18. PHIMINT, SWMINT

PHIMINT: Initial porosity of the matrix (fraction). *Accepts a real number*
 $0 \leq PHIMINT \leq 1$

SWMINT: Initial saturation of the matrix (fraction). *Accepts a real number*
 $0 \leq SWMINT \leq 1$

Line 19. Descriptor: Matrix constant velocity field. *Accept character string up to 100 chars as a descriptive line.*

Line 20. TVELXM, TVELYM, TVELZM

TVELXM: Matrix velocity of the phase in x-direction (m/s). *Accepts a real number*
 $TVELXM \leq 0.6$

TVELYM: Matrix velocity of the phase in y-direction (m/s). *Accepts a real number*
 $TVELYM \leq 0.6$

TVELZM: Matrix velocity of the phase in z-direction (m/s). *Accepts a real number*
 $TVELZM \leq 0.6$

Line 21. Descriptor: Block temperature and brine viscosity. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 22. TTEMP_MAT, TVISW_MAT

TTEMP_MAT: Temperature of the grid block (K). *Accepts a real number > 273.15 .*

TVISW_MAT: Matrix brine viscosity (Pa/s). *Accepts a real number > 0 .*

6.5.2.F Dual-Porosity/Dual-Permeability Information

If a dual-porosity or dual-permeability medium is to be modeled, then:

Line 23. Descriptor: Shape factor and the transfer function. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 24. TSIGMA, TTAUW

TSIGMA: Shape factor (m^{-2}). *Accepts a real number > 0 .*

TTAUW: Fracture/matrix transfer function (m^3/s). *Accepts a real number > 0 .*

6.5.3 ASCII Flux-Field Input File (heterogeneous)

An example of NUTS's ASCII flux-field input file for heterogeneous properties is given in Appendix D.

Line 1. Descriptor: Is the input homogeneous (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 2. DUMP

DUMP: Logical flag to specify whether the following input is homogeneous. The value of this parameter is important only for CDB runs. *Accepts T or F.*

6.5.3.A Title Information

Line 3. Descriptor: Test run title. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 4. TEST_TITLE

TEST_TITLE: This is a descriptive line of the test run. The line is saved and concatenated with NUTS title when printed out to give an idea about both NUTS and test runs considered. *Accepts up to a 100-character string.*

6.5.3.B Time Information

Line 5. Descriptor: Time step size and number of time steps. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 6. DT, NTIMESTEP

DT: Time step size (s). *Accepts a real number > 0 .*

NTIMESTEP: Number of time steps required to run NUTS. *Accepts an integer ≥ 1 .*

6.5.3.C Dimension Information

Line 7. Descriptor: Number of grid blocks in x, y, z. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 8. NX, NY, NZ

NX: Number of grid blocks in x-direction. *Accepts an integer ≥ 1 .*

NY: Number of grid blocks in y-direction. *Accepts an integer ≥ 1 .*

NZ: Number of grid blocks in z-direction. *Accepts an integer ≥ 1 .*

- Line 9. Descriptor:** Dimension of grid blocks in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 10. ((DX3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
DX3D: Dimension of the grid block in x-direction (m). *Accepts a real number > 0*
- Line 11. Descriptor:** Dimension of grid blocks in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 12. ((DY3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
DY3D: Dimension of the grid block in y-direction (m). *Accept a real number > 0*
- Line 13. Descriptor:** Dimension of grid blocks in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 14. ((DZ3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
DZ3D: Dimension of the grid block in z-direction (m). *Accepts a real number > 0.*

6.5.3.D Fracture Information

If the continuum to be modeled is fractured, then:

- Line 15. Descriptor:** Fracture porosities. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 16. ((PHIFINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
PHIFINT3D: Porosity of the fracture (fraction). *Accepts a real number*
 $0 \leq PHIFINT3D \leq 1.$
- Line 17. Descriptor:** Fracture saturations. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 18. ((SWFINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
SWFINT3D: Saturation of the fracture (fraction). *Accepts a real number*
 $0 \leq SWFINT3D \leq 1.$
- Line 19. Descriptor:** Fracture velocity field in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 20. ((VELWXF3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
VELWXF3D: Fracture velocity of the phase in x-direction (m/s). *Accepts a real number*
 $VELWXF3D \leq 0.6.$
- Line 21. Descriptor:** Fracture velocity field in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 22. ((VELWYF3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
VELWYF3D: Fracture velocity of the phase in y-direction (m/s). *Accepts a real number*
 $VELWYF3D \leq 0.6.$
- Line 23. Descriptor:** Fracture velocity field in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*
- Line 24. ((VELWZF3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)**
VELWZF3D: Fracture velocity of the phase in z-direction (m/s). *Accepts a real number*
 $VELWZF3D \leq 0.6.$
- Line 25. Descriptor:** Fracture grid block brine temperature. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 26. ((FTEMP3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

FTEMP3D: Temperature of the brine in the grid block (K). *Accepts a real number > 273.15.*

Line 27. Descriptor: Fracture grid block brine viscosity. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 28. ((FVIS3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

FVIS3D: Fracture brine viscosity (Pa/s). *Accepts a real number > 0.*

6.5.3.E Matrix Information:

If the matrix to be modeled is a continuum, then:

Line 29. Descriptor: Matrix porosities. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 30. ((PHIMINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

PHIMINT3D: Porosity of the matrix (fraction). *Accepts a real number $0 \leq PHIMINT3D \leq 1$*

Line 31. Descriptor: Matrix saturations. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 32. ((SWMINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

SWMINT3D: Saturation of the matrix (fraction). *Accepts a real number $0 \leq SWMINT3D \leq 1$.*

Line 33. Descriptor: Matrix velocity field in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 34. ((VELWXM3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

VELWXM3D: Matrix velocity of the phase in x-direction (m/s). *Accepts a real number $VELWXM3D \leq 0.6$.*

Line 35. Descriptor: Matrix velocity field in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 36. ((VELWYM3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

VELWYM3D: Matrix velocity of the phase in y-direction (m/s). *Accepts a real number $VELWYM3D \leq 0.6$.*

Line 37. Descriptor: Matrix velocity field in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 38. ((VELWZM3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

VELWZM3D: Matrix velocity of the phase in z-direction (m/s). *Accepts a real number $VELWZM3D \leq 0.6$.*

Line 39. Descriptor: Matrix Block temperature. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 40. ((XMTEMP3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

XMTEMP3D: Matrix Block brine temperature (K). *Accepts a real number > 273.15*

Line 41. Descriptor: Matrix Block brine viscosity. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 42. ((XMVIS3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

XMVIS3D: Matrix brine viscosity (Pa/s). *Accepts a real number > 0.*

6.5.3.F Dual-Porosity/Dual-Permeability Information

If medium to be modeled has dual-porosity or dual-permeability, then:

Line 43. Descriptor: Shape factor. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 44. ((SIGMA3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

SIGMA3D: Shape factor (m^2). *Accepts a real number ≥ 0 .*

Line 45. Descriptor: Transfer function. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 46. ((TAUW3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

TAUW3D: Fracture/matrix transfer function (m^3/s). *Accepts a real number ≥ 0 .*

6.5.4 Binary Flux-Field Input File

This file is the binary BRAGFLO output. It is read only for the single porosity matrix. It provides NUTS with the following variables:

- a. QA information
- b. Time information
- c. Grid blocks dimension and volumetric information
- d. Grid block saturation, pressure, velocity or fluxes field, and fluid densities

A full description of this file and the variables printed is available in BRAGFLO User's Manual.

6.5.5 Binary Output Files

6.5.5.A Binary Output Files (BRAGFLO Style)

This NUTS binary output is meant to be post-processed with the results of BRAGFLO. Therefore, the sequence and type of information in this file are in harmony with BRAGFLO's binary output (for a detailed description, the reader is referred to BRAGFLO User's Manual). The post-processor used for this objective is BINCOMBO_NV.

6.5.5.B Binary Output Files (NUTS Style)

To illustrate the structure of NUTS's output and to enable the reader to post-process its results, a FORTRAN program that reads NUTS's binary output file is given in Appendix K. The first part of the output, nested by an IF(IFLAGTIME .EQ. 0) statement, contains QA information, input echoes, and variable initializations. This information is output to the binary file once, at time zero. Temporally-varying quantities are an exception to the once-at-time-zero rule. They may be

reported additionally at subsequent time steps, but only variables specified in the input controls will be written out at subsequent times.

6.5.6 ASCII Output Files

6.5.6.A Extensive ASCII Output File

This file contains information similar to that in the NUTS-style binary output file. An illustrative example of NUTS's extensive ASCII output file is given in Appendix E.

6.5.6.B Specific-Ranges ASCII Output File

The purpose of this file is to produce an ASCII output file for output variables of interest to the user, and in specified spatial ranges. Its feature can be activated from the input file together with the ASCII file option.

6.5.7 Debug (Scratch) ASCII Output File

This feature produces a scratch or debug file in ASCII format that identifies the beginning and the end of the major procedures in NUTS, and writes the main variables associated with those procedures. The file stays open for only one time step. Therefore, it will echo some of the input data and the time-dependent variables at the first time step. The purposes of this output are (1) to facilitate debugging, and (2) to enable verification of numerical calculations.

6.6 Generating an Executable for NUTS-SA:

A command file to build NUTS-SA's executable is provided in Appendix F. This command file can build either a debug executable or a normal executable. In the building process, the main program `NUT_MAIN.FOR` will be compiled and will be linked to the `NUT_LIB_STANDALONE.FOR`.

6.7 Input/Output (I/O) and Other Files Used by NUTS-CC

Being a subset of NUTS-SA, NUTS-CC uses the following I/O files:

1. User Interactive Input
2. Parameter Statements
3. Input/Output Files

6.7.1 User-Interactive Input

A limited number of control flags are provided by the user while executing the program. These controls are part of the input and mainly specify the following:

- a) Name of the flux fields filename. *Accepts an 80-character string.*
- b) Type of the run, whether it is a test. *Accepts Y or N.*
- c) Type of the output file (ASCII/Binary). *Accepts ASC, BIN or ASC-BIN.*
- d) NUTS input filename. *Accepts an 80-character string.*
- e) NUTS output filename. *Accepts an 80-character string.*
- f) NUTS output debug filename. *Accepts an 80-character string.*
- g) Type of fluid. *Accepts L or G.*
- h) BRAGFLO ASCII input filename (*.INP) if the answer in b) is N and CANCEL if the answer in b) is Y. *Accepts an 80 character string.*
- i) Will NUTS input or the data base be used to provide chemical constituent properties and rock properties. *Accepts N for NUTS or CDB for the data base.*
- j) Properties CDB when the answer in i) is CDB and CANCEL otherwise. *Accepts an 80 character string.*
- k) NUTS undisturbed scenario CDB output file name used to initialize certain kind of intrusion computation. This kind of initialization is required only for intrusion time other than that used in BRAGFLO. *Accepts NUTS undisturbed scenario CDB output file name for intrusion calculation and up to 80 character or CANCEL otherwise.*

An example of such an interactive input included in a command file is as follows:

```
$ NUTS -                ! The Executable
  TEST_CON_DISP_DEC.IN - !Test Flux Field Input File
  Y -                   !Test Run? (y)
  ASC -                 !ASC/BIN Output
  NUT_CON_DISP_DEC.IN - !Nuts Input File
  NUT_CON_DISP_DEC.OUT - !NUTS Output ASCII File
  NUT_CON_DISP_DEC.DBG - !DEBUG File
  L -                   !Liquid/Gas
  CANCEL -              !(Would have been BRAGFLO.INP)
  N -                   !Read Nuclide Data from NUTS Input
  CANCEL -              ! Properties CDB input file
  CANCEL                ! NUTS undisturbed scenario output file
$ EXIT
```

6.7.2 NUTS-CC Parameter Statement

NUTS-CC uses the same parameter statement (NUT_PARAM.INC) used by NUTS-SA. In addition, the CAMCON SYSTEM requires additional parameters. These parameters are included in NUT_CDBXFER.INC.

6.7.3 Other Files Used by NUTS-CC

NUTS_CC accesses different combinations of files in each run depending on user specifications. These files can be classified as input, output, or debug files. The names of these files are provided in the interactive input given earlier in this chapter. A specific combination or all of the following files can be opened simultaneously:

1. ASCII radioisotope input file
2. Either (A) ASCII flux fields input files
 - a. Homogenous properties file
 - b. Heterogeneous properties file
3. Or (B) Binary flux-field input files (CDB files from BRAGFLO's output), and BRAGFLO ASCII input file associated with this case.
4. Binary (CDB) output files
5. Extensive ASCII output file.
6. Properties CDB input file.
7. NUTS undisturbed scenario output file for initialization of well bore intrusion calculations (the same output of item 4 above).
8. Debug (scratch) ASCII output files.

The files in items 1, 2, 5 and 8 are similar to those used in NUTS-SA. The only difference between the two is in the way the binary files are treated. In NUTS-SA, if two ASCII files are used as inputs (one for the nuclides and rock properties, and the other for the flux fields), then a binary output file can be generated. However, in NUTS-CC, a binary CDB is generated only if the flux fields are read from a binary CDB.

6.7.4 Binary (CDB) Flux-Field and the Associated ASCII Input Files

NUTS-CC uses this file to read the output from BRAGFLO. Time = zero information will be read first. It includes some QA information, the geometrical description of the problem, and initial saturation and porosity data. At each timestep thereafter, brine pressure (PRESBRIN), porosity (POROS), gas saturation (SATGAS), interface volumetric fluxes (FLOWBRX, FLOWBRY), and well fluxes (WELLBRIN) will be read. BRAGFLO ASCII input file is used here to provide NUTS with the material maps information, brine reference density and compressibility, and brine residual saturation.

6.7.5 NUTS's Binary (CDB) Output File

In the binary (CDB) output file, the following variables will be added to the input CDB data:

CNCNJM#: Concentration of the injected component.

FLUXIM#: Mass flux at the interface I (the convention is to put the flux at the left interface (i-1) of each grid block (i,j,k) and the sign is positive to the right).

FLUXJM#: Mass flux at the interface J (the convention is to put the flux at the lower interface (j-1) of each grid block (i,j,k) and the sign is positive in upward direction).

CNDMMS#: Mass of the condensed component # (has a non-zero value only in gas transport).

Depending on user specifications, concentration or mass of the component(s) will be output to the CDB file and in any form specified by the output array control flags (IPRNTMB (1-14)) as shown in Table 4.

Table 4. CDB Output Variables and Their Description

Printing Flag	Description of the Matrix Variable	NUTS CDB output variable name
IPRNTMB(1)	Total mass of dissolved constituents in the brine in each grid block, kg.	SM_DIS
IPRNTMB(2)	Total mass of precipitated constituents in each grid block, kg.	SM_PREC
IPRNTMB(3)	Total mass of dissolved, precipitated and adsorbed constituents in each grid block, kg.	SM_TOT
IPRNTMB(4)	Total dissolved curies of all constituents in each grid block, Ci.	CSM_DIS
IPRNTMB(5)	Total precipitated curies of all constituents in each grid block, Ci.	CSM_PREC
IPRNTMB(6)	Total dissolved, precipitated, and adsorbed curies of all constituents in each grid block, Ci.	CSM_TOT
IPRNTMB(7)	Volumetric concentration of the dissolved isotope, kg/(m ³ brine).	CM#
IPRNTMB(8)	Dissolved mass in the brine of a certain constituent, kg.	BMDISM#
IPRNTMB(9)	Precipitated mass of a certain constituent, kg.	BMPRCM#
IPRNTMB(10)	Soil base concentration of the isotope, mg/(kg soil).	ADPRCNM#
IPRNTMB(11)	Curies of volumetric concentration of a certain constituent, Ci/(m ³ brine).	VOLMCC#
IPRNTMB(12)	Curies of dissolved mass in the brine of a certain constituent, Ci.	DISMMC#
IPRNTMB(13)	Curies of precipitated mass of a certain constituent, Ci.	PRCPMMC#
IPRNTMB(14)	Curies of dissolved, precipitated and adsorbed mass of a certain constituent, Ci.	TOTMMC#

In the above notations, # refers to the location of the component in the input sequence, and M refers to the matrix continuum. Similar notation is also available for the fracture continuum.

6.7.6 NUTS's Properties Binary (CDB) Input File

When the answer to item (j) in the user interactive input (Section 6.7.1) is CDB, NUTS-CC expects to read the isotopes' property from a CDB file instead of NUTS's ASCII file. These

properties include but are not limited to the inventories, atomic weight, solubility limit, and half-life of the isotope. Other properties that are not used in the CCA calculations, such as pH of the brine, sorption coefficients, grain densities, molecular diffusion, dispersivities, and tortuosities, can also be read from the same file provided that the flag related to the respective module is activated in NUTS's ASCII input (through certain flags, one can activate sorption, dispersion, etc.).

6.8 Generating an Executable for NUTS-CC

A command file that builds NUTS-CC's executable is provided in Appendix G. This command file builds either (i) a debug file or (ii) a normal executable. In the building process, the main program `NUT_MAIN.FOR` will be compiled and linked with `NUT_CDBLIB.FOR`, `NUT_LIB.FOR`, and linked with `CAMDAT_LIB`, `CAMCON_LIB`, and `CAMSUPES_LIB`.

6.9 Exercising NUTS from a Command Line

Appendix H provides several examples of command files (*.com) to run NUTS from WIPP's Alpha-VAX cluster.

7.0 DESCRIPTION OF NUTS'S INPUT FILES

NUTS's input files are discussed in detail in Sections 6.3.1 (ASCII radioisotope file) and 6.3.2 (ASCII flux-field input file). Appendices A, B, C, and D provide examples of NUTS's ASCII Radioisotope input files, its flux-field input files (homogenous and heterogeneous cases), and a sample material map, all designed to run on the WIPP Alpha-VAX microcomputer cluster.

8.0 ERROR MESSAGES

Selected error messages and steps toward their correction are listed in Section 6.0, at the places where they may arise.

A complete listing of NUTS's error and warning messages is given in Appendix I. Error messages will be written on the screen and in the debug file and will cause NUTS to stop. Warning messages will be written only in the debug file and will generally not result in an abort. General warning messages will appear on the screen if NUTS detects undesirable value(s) in the input data.

9.0 DESCRIPTION OF NUTS'S OUTPUT FILES

NUTS's output is capable of appearing in various user-readable and user non-readable formats, depending on user inputs. To assist the user, examples in both formats are included. The first, which is user readable, is NUTS's normal debug file. An annotated version of that file appears in Appendix J.

The second example is of a user-readable output file, namely, NUTS's Extensive ASCII Output File. It is by far the most complete, user-readable, NUTS output file.

Several of NUTS's important input and output files may, at the user's option, be written in binary format. As a service to users, we have provided a translation program that renders NUTS's binary files readable. That program has not been QAed and consequently it should not be regarded as an official part of NUTS's QA documentation. However, to assist the user, the translation program is included herein as Appendix K.

10.0 REFERENCES

Asfari, A., and P.A. Witherspoon (1973). "Numerical Simulation of Naturally Fractured Reservoirs", Paper SPE 4290, presented at the 3rd Numerical Simulation of Reservoir Performance Symposium, Houston, TX, 10-12 Jan.

Bear, J. (1988). "Dynamic of Fluids in Porous Media", Dover Publications, Inc., NY.

Bear, J. (1993) " Flow and Contaminant Transport in Fractured Rock", Academic Press, Inc.

Beckner, B.L., Firoozabadi, A., and K. Aziz (1988). "Modeling Transverse Imbibition in Double-Porosity Simulators", Paper SPE 17414 presented at the SPE California Regional Meeting, Long Beach, CA 23-25 March.

Beckner, B.L., Chan, H.M., McDonald, A.E., Wooten, S.O., and T.A. Jones (1991). "Simulating naturally fractured Reservoirs Using Subdomain Method", paper SPE 21241, presented at the 11th SPE Symposium on reservoir Simulation, Anaheim, CA, 17-20 Feb.

Chen, W.H., Wasserman, M.L., and R.E. Fitzmorris (1987). "A Thermal Simulator for Naturally Fractured Reservoirs", paper SPE 16008 presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, TX, 1-4, Feb.

Coats, K.H. (1989). " Implicit Compositional Simulation of Single-Porosity and Dual-Porosity Reservoirs", Paper SPE 18427 presented at the 10th SPE Symposium on Reservoir Simulation, Houston, TX, Feb. 6-8, 1989.

Dean, R.H., and L.L. Lo (1988) " Simulations of Naturally Fractured Reservoirs", SPE Reservoir Engineering, 638-648.

DeSwaan, A. (1978). "Theory of Waterflooding in Fractured Reservoirs", SPEJ+, 117-122.

DeSwaan A.D. (1982). "A Three-Phase Model for Fractured Reservoirs Presenting Fluid Segregation", paper SPE 10510, presented at the 6th SPE Symposium on Reservoir Simulation, New Orleans, LA Jan. 31- Feb. 3.

Doctor, P.G. et al. (1992). "An Example Postclosure Risk Assessment Using the Potential Yucca Mountain Site", Pacific Northwest Laboratory, Washington.

Duguid, J.O., and P.C.Y. Lee (1977). "Flow in Fractured Porous Media", Water Resources Research, Vol. 13, No. 3, 558-566.

+ SPEJ refers to the Society of Petroleum Engineers Journal

Dutra, T.V., and K. Aziz (1991). "A New Double-Porosity Reservoir Model for Oil/Water Flow Problems", paper SPE 21248, presented at the 11th SPE Symposium on reservoir Simulation, Anaheim, CA, 17-20 Feb.

Dykhuizen, R.C. (1990). "A new Coupling Term for Dual-Porosity Models", Water Resources Research, Vol. 26, No. 2, 351-356.

Evans, R.D. (1982) "A Proposed Model for Multi-Phase Flow Through naturally Fractured Reservoir", SPEJ, 669-680.

Gilman, J.R. and H. Kazemi (1983). "Improvement in Simulation of naturally Fractured Reservoirs", SPEJ 695-707.

Gilman, J.R. (1986). "An Efficient Finite-Difference Method for Simulating Phase Segregation in the Matrix Blocks in Double-Porosity Reservoirs", SPE Reservoir Engineering, 403-413.

Gilman, J.R. and H. Kazemi (1988). "Improved Calculations for Viscous and gravity Displacement in Matrix Blocks in Dual-Porosity Simulators", JPT, 60-70.

Hill, A.C. and G.W. Thomas. (1985). "A New Approach for Simulating Complex Fractured Reservoirs", Paper SPE 13537 presented at the 1985 Res. Sim. Sym., Dallas, TX, 10-13 Feb.

Huang_Zhang C. (1983). "Numerical Simulation of Conning Behavior in a Naturally Fractured reservoir", SPEJ, 879-884.

Kazemi, H., L.S. Merrill, K.L. Porterfield and P.R. Zeman. (1976). "Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs", SPEJ 317-326.

Kazemi, H., and L.S. Merrill (1979). "Numerical Simulation of Water Imbibition in Fractured Cores", SPEJ 175-182.

Kazemi, H., Gilman, J.R., and A.M. El-Sharkawy (1989). "Analytical and Numerical Solution of Oil Recovery From Fractured Reservoirs Using Empirical Transfer Functions", paper SPE 19849 presented at the 64th SPE Annual Meeting, San Antonio, TX, 8-11 Oct.

Litvak, B.L. (1985). "Simulation and Characterization of Naturally Fractured Reservoirs", Proc. Res. Characterization Technical Conf., Dallas Academic Press, NY.

Narasimhan, T.N. (1982). "Multi-Dimensional Numerical Simulation of Fluid Flow in Fractured porous Media", Water Resources Research, Vol. 18, No. 4, 1235-1247.

Pruess, K. and T.N. Narasimhan (1985). "A Practical Method for Modeling Fluid and Heat Flow in fractured Porous Media", SPEJ, 15-26.

Rechard, R. P. , A. P. Gilkey, H. J. Iuzzolino, D. K. Rudeen, and K. A. Byle. (1993). Programmer's Manual for CAMCON: Compliance Assessment Methodology Controller. SAND90 - 1984. Sandia National Laboratories, Albuquerque NM.

Rechard, R. P., ed. (1992). User's Reference Manual for CAMCON: Compliance Assessment Methodology Controller; Version 3.0. SAND90 - 1983. Sandia National Laboratories, Albuquerque NM.

Reiss, L.H., Bossie-Codreanu, D.N., and L. DuPrey (1973). "Flow in Fissured Reservoirs", Paper SPE 4343, presented at the 2nd Annual European Meeting of SPE-AIME, London, England, 2-3 April.

Rossen, R.H. (1977). "Simulation of Naturally Fractured Reservoirs with Semi-Implicit Source Terms", SPEJ 201-210.

Saidi, A. (1983). "Simulation of Naturally Fractured Reservoirs", Paper SPE 12270 presented at the SPE Symposium on Reservoir Simulation, San Francisco, 16-18 Nov.

Shinta, A.A. and H. Kazemi. (1993). "Tracer Transport in Characterization of Dual-Porosity Reservoirs", Paper SPE 26636 presented at the 1993 SPE Annual Tech. Conf and Exh., Houston, TX, 3-6 Oct.

Sonier, F., Souillard, P., and F.T. Blaskovich (1986). "Numerical Simulation of Naturally Fractured Reservoirs", SPE Reservoir Engineering, 1114-1122.

Thomas, L.K., Dixon, T.N., and R.G. Pierson (1983). "Fractured Reservoir Simulation", SPEJ, 42-54.

WIPP PA Department (1992). Preliminary Performance Assessment for the Waste Isolation Pilot Plant, December 1992. Volume 1: Third Comparison with 40 CFR 191, Subpart B. SAND92 - 0700/1. Sandia National Laboratories, Albuquerque NM.

WIPP PA Department (1992). Preliminary Performance Assessment for the Waste Isolation Pilot Plant, December 1992. Volume 2: Technical Basis. SAND92 - 0700/2. Sandia National Laboratories, Albuquerque NM

WIPP PA Department (1992). Preliminary Performance Assessment for the Waste Isolation Pilot Plant, December 1992. Volume 3: Model Parameters. SAND92 - 0700/3. Sandia National Laboratories, Albuquerque NM.

WIPP PA Department (1992). Preliminary Performance Assessment for the Waste Isolation Pilot Plant, December 1992. Volume 4: Uncertainty and Sensitivity Analyses for 40 CFR 191, Subpart B. SAND92 - 0700/4. Sandia National Laboratories, Albuquerque NM.

WIPP PA Department (1992). Preliminary Performance Assessment for the Waste Isolation Pilot Plant, December 1992. Volume 5: Uncertainty and Sensitivity Analyses of Gas and Brine Migration for Undisturbed Performance. SAND92 - 0700/5. Sandia National Laboratories, Albuquerque NM.

Wu, Y.S., and K. Pruess (1988). "A Multiple-Porosity Method for Simulation of Naturally Fractured Petroleum Reservoirs", SPE Reservoir Engineering, 327-336.

11.0 APPENDICES

Appendix A: Examples of ASCII Radioisotope Input Files

Example 1:

```
** NUTS TITLE **
'CONVECTION-DISPERSION-DECAY IN ONE DIMENSION PROBLEM TEST'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,2
'TEST4_SITE' 1
** (1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'TEST4_SITE'
'COMP1' 'DAUGHT1' 'NONE' 'ELM1'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
1
'ELM1' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY( IS PH REQUIRED (Y/N)) **
'N'
** METHOD ORDER,ONE-POINT=1,TWO-POINT=2,SPLIT-OPERATOR=3,MID-POINT=4 **
1
** DEGREE OF IMPLICITNESS **
1.
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'Y'
** MATRIX SYMMETRIC DISPERSION AND DATA PROVIDED BY NUTS(T/F): ANSWER IF DISP.=Y **
.TRUE. .TRUE.
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.TRUE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.TRUE.
** MATRIX INITIAL.IN THE CDB FOR INTERPOLATED INTRUSION TIME IN E1 SCENARIO **
.FALSE.
** PRINT FLAGS OF MATRIX VARIABLES IN AN ASCII FILE**
0,0,0,0,0,0,1,0,0,0,0,0,0,0
** SPECIFIC RANGE OF DATA TO BE PRITED IN AN ASCII FILE **
'N'
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** TEMP. DEPEND.OF MOLECULAR DIFFUSION (ENTER DATA IF DISPER. IS (Y)) **
'COMP1' .FALSE.
** PRINTING FREQUENCY IN AN ASCII FILE **
1,1.E10
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.FALSE.
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **
NUMBER OF TIMES FOR SPECIFYING MATERIAL MAP
1
START TIME FOR MAP 1
0.D0
MATERIAL TYPE GRID MAP
1 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2
# NAME
1 REGION1
```

```
2 REGION2
NUMBER OF WASTE REGIONS
1
MAT_WASTE1
1
1
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,...,NELEMENT **
'ELM1' 7
** (COMP. NAME, MOL. (ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...,NUCLIDE **
'COMP1' 0.238 0.0 0.0 7.7016E12
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN ( ASCII FILE FOR FLUX FIELD) **
1000.
** MOLECULAR DIFFUSION OF EACH COMPONENT **
'COMP1' 1.E-9
** REFERENCE VISCOSITY AND TEMPERATURE **
** ROCK DENSITY INPUT **
** WASTE MATRIX INPUT **
1
'COMP1' 1 1
** (1.SITE NAME, # OF GRID IN THE SITE 2. INDECES), 1....NSITES **
'TEST4_SITE' 1
1,1,1
** MATRIX SORPTION INPUT **
** MATRIX DISPERSION INPUT **
** MATRIX LONGITUDINAL MATERIAL DISPERSIVITIES **
10.0 10.0
** MATRIX TRANSVERSE MATERIAL DISPERSIVITIES **
0.0 0.0
** MATRIX MATERIAL TORTUOSITY **
10.0 10.0
** MATRIX SOURCE (INJECTION/PRODUCTION) INPUT IF ANY **
** MATRIX DIR. B.CS. (REP.='GENERAL, ANYWHERE='NOT_GENERAL') **
1 'GENERAL'
'COMP1' 1 1.0
** TIME DEPENDENT SOURCE IN THE MATRIX **
** MATRIX CONCENTRATION INITIALIZATION **
1
'COMP1' 1 1
1 1 1
'COMP1'
1.0
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Example 2:

```
** NUTS RUN TITLE **
'2D TEST, ADVECTIVE IN X, DISPERSIVE IN X & Y, DECAY AND SORPTION, DT=5 Y'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,2
'NMVP_SITE' 1
**(1. SITE, 2. COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'NMVP_SITE'
'U234' 'DAUGHT1' 'NONE' 'U'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
1
'U' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY( IS PH REQUIRED (Y/N)) **
'N'
** ORDER OF THE METHOD **
1
** DEGREE OF IMPLICITNESS **
1.
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'Y'
'L'
'U234' 'ADSORP' .FALSE.
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'Y'
** DOES MATRIX HAVE SYMMETRIC DISPERSION (T/F): ANSWER IF DISPERSION IS Y **
.TRUE. .TRUE.
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.TRUE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.FALSE.
** MATRIX INITIAL. IN THE CDB FOR INTERPOLATED INTRUSION TIME IN E1 SCENARIO **
.FALSE.
** PRINT FLAGS OF MATRIX VARIABLES IN AN ASCII FILE**
0,0,0,0,0,0,1,0,0,0,0,0,0,0
** SPECIFIC RANGE OF DATA TO BE PRITED IN AN ASCII FILE **
'N'
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** TEMP. DEPEND. OF MOLECULAR DIFFUSION (ENTER DATA IF DISP. IS (Y)) **
'U234' .FALSE.
** PRINTING FREQUENCY IN AN ASCII FILE **
200,1.E12
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.TRUE.
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **
NUMBER OF TIMES FOR SPECIFYING MATERIAL MAP
1
START TIME FOR MAP 1
0.D0
MATERIAL TYPE GRID MAP
239*2 11*1 14*2 8*2 11*1 14*2 8*2 11*1 14*2 693*2
# NAME
1 REGION1
2 REGION2
NUMBER OF WASTE REGIONS
1
MAT_WASTE1
1
1
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,..,NELEMENT **
'U' 7
```

```
** (COMP. NAME, MOL. (ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...,NUCLIDE **
'U234' 0.234 0.0 0.0 7.7156621D12
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN ( ASCII FILE FOR FLUX FIELD) **
1000.
** MOLECULAR DIFFUSION OF EACH COMPONENT **
'U234' 2.7D-10
** REFERENCE VISCOSITY AND TEMPERATURE **
** ROCK DENSITY INPUT **
2500.0 2500.0
** WASTE MATRIX INPUT **
1
'U234' 1 1
** (1. SITE NAME, # OF GRID IN THE SITE, 2. INDECES), 1....NSITES **
'NMVP_SITE' 33
9,8,1 10,8,1 11,8,1 12,8,1 13,8,1
14,8,1 15,8,1 16,8,1 17,8,1 18,8,1
19,8,1
9,9,1 10,9,1 11,9,1 12,9,1 13,9,1
14,9,1 15,9,1 16,9,1 17,9,1 18,9,1
19,9,1
9,10,1 10,10,1 11,10,1 12,10,1 13,10,1
14,10,1 15,10,1 16,10,1 17,10,1 18,10,1
19,10,1
** MATRIX SORPTION INPUT **
'U234'
2.222222222D-5 2.222222222D-5
** MATRIX DISPERSION INPUT **
** MATRIX LONGITUDINAL MATERIAL DISPERSIVITIES **
100.0 100.0
** MATRIX TRANSVERSE MATERIAL DISPERSIVITIES **
5.0 5.0
** MATRIX MATERIAL TORTOUSITY **
10.0 10.0
** MATRIX SOURCE (INJECTION/PRODUCTION) INPUT IF ANY **
** MATRIX DIR. B.CS. (REP.='GENERAL, ANYWHERE='NOT_GENERAL') **
1 'NOT_GENERAL'
'U234' 1 128
 33,1,1 33,2,1 33,3,1 33,4,1 33,5,1
 33,6,1 33,7,1 33,8,1 33,9,1 33,10,1
 33,11,1 33,12,1 33,13,1 33,14,1 33,15,1
 33,16,1 33,17,1 33,18,1 33,19,1 33,20,1
 33,21,1 33,22,1 33,23,1 33,24,1 33,25,1
 33,26,1 33,27,1 33,28,1 33,29,1 33,30,1
 33,31,1
 1,31,1 2,31,1 3,31,1 4,31,1 5,31,1
 6,31,1 7,31,1 8,31,1 9,31,1 10,31,1
11,31,1 12,31,1 13,31,1 14,31,1 15,31,1
16,31,1 17,31,1 18,31,1 19,31,1 20,31,1
21,31,1 22,31,1 23,31,1 24,31,1 25,31,1
26,31,1 27,31,1 28,31,1 29,31,1 30,31,1
31,31,1 32,31,1 33,31,1
 1,1,1 1,2,1 1,3,1 1,4,1 1,5,1
 1,6,1 1,7,1 1,8,1 1,9,1 1,10,1
 1,11,1 1,12,1 1,13,1 1,14,1 1,15,1
 1,16,1 1,17,1 1,18,1 1,19,1 1,20,1
 1,21,1 1,22,1 1,23,1 1,24,1 1,25,1
 1,26,1 1,27,1 1,28,1 1,29,1 1,30,1
 1,31,1
 1,1,1 2,1,1 3,1,1 4,1,1 5,1,1
 6,1,1 7,1,1 8,1,1 9,1,1 10,1,1
11,1,1 12,1,1 13,1,1 14,1,1 15,1,1
16,1,1 17,1,1 18,1,1 19,1,1 20,1,1
21,1,1 22,1,1 23,1,1 24,1,1 25,1,1
26,1,1 27,1,1 28,1,1 29,1,1 30,1,1
31,1,1 32,1,1 33,1,1
'U234'
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0.
```


Example 3:

```
** NUTS TITLE **  
'NUTS TRACER TEST FOR E01_DOWN SENARIO'  
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **  
1,38  
'WIPP_SITE' 2  
** (1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **  
'WIPP_SITE'  
'WASTE#1' 'NONE' 'NONE' 'WASTE'  
'CASTL#1' 'NONE' 'NONE' 'CASTL'  
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **  
2  
'WASTE' .FALSE. .FALSE.  
'CASTL' .FALSE. .FALSE.  
** COLLOIDAL TRANSPORT FLAG (T/F) **  
.FALSE.  
** PH DEPENDENT SOLUBILITY (IS PH REQUIRED (Y/N)) **  
'N'  
** ORDER OF THE METHOD ***  
1  
** DEGREE OF IMPLICITNESS **  
1.  
** IMPLICIT PRECIPITATE 1.(T/F), IF TRUE, 2. # OF ITERATION, TOLERANCE **  
.FALSE.  
** IS MATRIX SORPTION REQUIRED (Y/N) **  
'N'  
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **  
'N'  
** DOES MATRIX HAVE SYMMETRIC DISPERSION (T/F): ANSWER IF DISPERSION IS Y **  
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **  
'N'  
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **  
.TRUE.  
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **  
.FALSE.  
** MATRIX INITIAL.IN THE CDB FOR INTERPOLATED INTRUSION TIME IN E1 SCENARIO **  
.FALSE.  
** PRINT FLAGS OF MATRIX VARIABLES IN A BINARY FILE **  
0,0,0,0,0,0,1,0,0,0,0,0,0,0  
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **  
** PRINTING FREQUENCY IN A BINARY FILE **  
1,12  
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **  
.FALSE.  
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **  
1.D-18  
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **  
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***  
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,..,NELEMENT **  
'WASTE' 1.  
'CASTL' 1.  
** (COMP. NAME, MOL.(ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,..,NUCLIDE **  
'WASTE#1' .1 0. 0. 0.  
'CASTL#1' .1 0. 0. 0.  
** GROUND WATER PH INPUT **  
** STANDARD BR. DENS. IF NOT BRAGFLO RUN (READ ASCII FILE FOR FLUX FIELD) **  
** MOLECULAR DIFFUSION OF EACH COMPONENT **  
** ROCK DENSITY INPUT **  
** WASTE MATRIX INPUT (LOCATION OF THE WASTE) **  
1  
'WASTE#1' 1 1  
** (1. SITE_NAME, # OF THE GRID IN THE SITE, 2. INDECES ),1,..,NSITES **  
'WIPP_SITE' 30  
 8,8,1 9,8,1 10,8,1 12,8,1 13,8,1 14,8,1 15,8,1 16,8,1  
 17,8,1 18,8,1 8,9,1 9,9,1 10,9,1 12,9,1 13,9,1 14,9,1  
 15,9,1 16,9,1 17,9,1 18,9,1 8,10,1 9,10,1 10,10,1 12,10,1  
 13,10,1 14,10,1 15,10,1 16,10,1 17,10,1 18,10,1  
** MATRIX SORPTION INPUT **  
** MATRIX DISPERSION INPUT **
```

```
** MATRIX SOURCE INPUT (INJECTED NUCLIDES IF ANY) **  
** MATRIX DIR. B.CS. INPUT (REP.='GENERAL',ANYWHERE= 'NOT_GENERAL') **  
2 'NOT_GENERAL'  
'WASTE#1'      1      30  
  8,8,1    9,8,1    10,8,1  12,8,1    13,8,1    14,8,1    15,8,1    16,8,1  
 17,8,1    18,8,1     8,9,1    9,9,1    10,9,1    12,9,1    13,9,1    14,9,1  
 15,9,1    16,9,1    17,9,1    18,9,1     8,10,1    9,10,1    10,10,1    12,10,1  
13,10,1    14,10,1    15,10,1  16,10,1    17,10,1    18,10,1  
'CASTL#1'      2      27  
  3,1,1    4,1,1    5,1,1    6,1,1    7,1,1    8,1,1    9,1,1  
10,1,1    11,1,1    12,1,1    13,1,1    14,1,1    15,1,1    16,1,1  
17,1,1    18,1,1    19,1,1    20,1,1    21,1,1    22,1,1    23,1,1  
24,1,1    25,1,1    26,1,1    27,1,1    28,1,1    29,1,1  
'WASTE#1'  
  1.  1.  1.  1.  1.  1.  1.  1.  
  1.  1.  1.  1.  1.  1.  1.  1.  
  1.  1.  1.  1.  1.  1.  1.  1.  
'CASL#1'  
  1.  1.  1.  1.  1.  1.  1.  1.  
  1.  1.  1.  1.  1.  1.  1.  1.  
  1.  1.  1.  1.  1.  1.  
** TIME DEPENDENT SOURCE IN THE MATRIX **  
** MATRIX CONCENTRATION INITIALIZATION **  
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Example 4:

```
** NUTS TITLE **
'1D TEST, ADVEC. AND DISP. IN X, 3 MEMBER DECAY AND SORPTION'
** 1.# OF SITES,# OF MATERIAL, (2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,2
'CHAIN_SITE' 3
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'CHAIN_SITE'
'IS01' 'DAUGHT1' 'NONE' 'EL1'
'DAUGHT1' 'DAUGHT2' 'IS01' 'EL1'
'DAUGHT2' 'DAUGHT3' 'DAUGHT1' 'EL1'
** 1.# OF ELEMENT, (2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
1
'EL1' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY( IS PH REQUIRED (Y/N)) **
'N'
***** ORDER OF THE METHOD *****
1
***** DEGREE OF IMPLICITNESS *****
1.D0
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'Y'
'L'
'IS01' 'ADSORP' .FALSE.
'DAUGHT1' 'ADSORP' .FALSE.
'DAUGHT2' 'ADSORP' .FALSE.
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'Y'
** DOES MATRIX HAVE SYMMETRIC & NUTS DISPER.(T/F): ANSWER IF DISPER.IS Y **
.TRUE. .TRUE.
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.TRUE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.TRUE.
** MATRIX INITIAL. IN THE CDB FOR INTERPOLATED INTRUSION TIME IN E1 SCENARIO **
.FALSE.
** PRINT FLAGS OF MATRIX VARIABLES IN AN ASCII FILE**
0,0,0,0,0,0,1,0,0,0,0,0,0,0
** SPECIFIC RANGE OF DATA TO BE PRITED IN AN ASCII FILE **
'N'
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** TEMP. DEPEND. OF MOLECULAR DIFUSION (ENTER DATA IF DISP. IS (Y)) **
'IS01' .FALSE.
'DAUGHT1' .FALSE.
'DAUGHT2' .FALSE.
** PRINTING FREQUENCY IN AN ASCII FILE **
25,1.D12
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.FALSE.
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **
NUMBER OF TIMES FOR SPECIFYING MATERIAL MAP
1
START TIME FOR MAP 1
0.D0
MATERIAL TYPE GRID MAP
1*1 76*2
# NAME
1 FORM1
2 FORM2
NUMBER OF WASTE REGIONS
```

```
1
MAT_WASTE1
1
1
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,...,NELEMENT **
'EL1'      7.D0
** (COMP. NAME, MOL. (ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...,NUCLIDE **
'ISO1'     0.1D0    0.D0    0.D0    1.366548D10
'DAUGHT1'  0.1D0    0.D0    0.D0    4.734000D8
'DAUGHT2'  0.1D0    0.D0    0.D0    2.064024D11
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN ( ASCII FILE FOR FLUX FIELD) **
1000.D0
** MOLECULAR DIFFUSION OF EACH COMPONENT **
'ISO1'     0.D0
'DAUGHT1'  0.D0
'DAUGHT2'  0.D0
** REFERENCE VISCOSITY AND TEMPERATURE **
** ROCK DENSITY INPUT **
1.11111111D0  1.11111111D0
** WASTE MATRIX INPUT **
3
'ISO'      1      1
'DAUGHT1'  2      1
'DAUGHT1'  3      1
** (1.SITE NAME, # OF GRID IN THE SITE 2. INDECES OF THE BLOCKS)1...NSITES **
'CHAIN_SITE'  1
1,1,1
** MATRIX SORPTION INPUT **
'ISO1'
935.1D0      935.1D0
'DAUGHT1'
7793.233D0   7793.233D0
'DAUGHT2'
311.633D0    311.633D0
** MATRIX DISPERSION INPUT **
** MATRIX LONGITUDINAL MATERIAL DISPERSIVITIES **
2.5908D0     2.5908D0
** MATRIX TRANSVERSE MATERIAL DISPERSIVITIES **
0.0D0        0.0D0
** MATRIX MATERIAL TORTOUSITY **
10.0D0       10.0D0
** MATRIX SOURCE (INJECTION/PRODUCTION) INPUT IF ANY **
** MATRIX DIR. B.CS. (REP.='GENERAL, ANYWHERE='NOT_GENERAL') **
3  'NOT_GENERAL'
'ISO1'      1      2
1,1,1  77,1,1
'DAUGHT1'   2      2
1,1,1  77,1,1
'DAUGHT2'   3      2
1,1,1  77,1,1
'ISO1'
1.D0        0.D0
'DAUGHT1'
0.01D0      0.D0
'DAUGHT2'
0.004D0     0.D0
** TIME DEPENDENT SOURCE IN THE MATRIX ****
** MATRIX CONCENTRATION INITIALIZATION **
3
'ISO1'      1      1
1,1,1
'DAUGHT1'   1      1
1,1,1
'DAUGHT2'   1      1
1,1,1
'ISO1'
1.D0
'DAUGHT1'
```

0.01D0
'DAUGHT2'
0.004D0

** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **

Information Only

Example 5:

```
** NUTS TITLE **
'1D TEST, FOR PRECIPITATION AND PREFERENTIAL COLLOID SOLUBILITY'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,2
'SRC_SITE' 2
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'SRC_SITE'
'SOLUBLE1' 'NONE' 'NONE' 'ELSOL1'
'COLLOID1' 'NONE' 'NONE' 'ELCOL1'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
2
'ELSOL1' .FALSE. .FALSE.
'ELCOL1' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.TRUE.
**1. # OF COLLOIDS,(2. LOC. COLLOID,3. LOC OF THE RELATED DISS)1...NCOLLOID **
1
2
1
.TRUE.
** PH DEPENDENT SOLUBILITY( IS PH REQUIRED (Y/N)) **
'N'
***** ORDER OF THE METHOD *****
1
***** DEGREE OF IMPLICITNESS *****
1.
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'N'
** DISPERSION IS SYMMETRIC (T/F), INPUT FROM NUTS (T/F) ***
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.TRUE.
** MATIRX INITIAL. IN THE CDB FOR INTERPOLATED INTURSION TIME IN E1 SCENARIO **
.FALSE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.FALSE.
** PRINT FLAGS OF MATRIX VARIABLES IN AN ASCII FILE**
0,0,0,0,0,0,0,1,1,0,0,0,0
** SPECIFIC RANGE OF DATA TO BE PRITED IN AN ASCII FILE **
'N'
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** PRINTING FREQUENCY IN AN ASCII FILE **
1,1.E12
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.FALSE.
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **
NUMBER OF TIMES FOR SPECIFYING MATERIAL MAP
1
START TIME FOR MAP 1
0.D0
MATERIAL TYPE GRID MAP
1*2 1*1 1*2
# NAME
1 REPOS
2 LEFTRIGHT
NUMBER OF WASTE REGIONS
1
MAT_WASTE1
1
1
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
```

```
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,...,NELEMENT **
'ELSOL1'      1.DO
'ELCOL1'      0.9542425D0
** (COMP. NAME, MOL. (ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...,NUCLIDE **
'SOLUBLE1'    0.1  1.1578947  0.  0.
'COLLOID1'    0.1  1.0421053  0.  0.
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN ( ASCII FILE FOR FLUX FIELD) **
1000.
** MOLECULAR DIFFUSION OF EACH COMPONENT **
** ROCK DENSITY INPUT **
** WASTE MATRIX INPUT **
2
'SOLUBLE1'    1      1
'COLLOID1'    2      1
** (1. SITE NAME, # OF GRID 2. INDECES OF WASTE MATRIX)1...NSITE **
'NMVP_SITE'   1
2,1,1
** MATRIX SORPTION INPUT **
** MATRIX DISPERSION INPUT **
** MATRIX SOURCE (INJECTION/PRODUCTION) INPUT IF ANY **
** MATRIX DIR. B.CS. (REP.='GENERAL, ANYWHERE='NOT_GENERAL') **
2 'NOT_GENERAL'
'SOLUBLE1'    1      1
3,1,1
'COLLOID1'    2      1
3,1,1
'SOLUBLE1'
0.
'COLLOID1'
0.
** TIME DEPENDENT SOURCE IN THE MATRIX ****
** MATRIX CONCENTRATION INITIALIZATION **
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
** SCALING FACTORS IN X-DIRECTION **
1.  1.  1.
** SCALING FACTORS IN Y-DIRECTION **
1.  1.  1.
** SCALING FACTORS IN Z-DIRECTION **
1.  1.  1.
```

Appendix B: Example of a Nine-Grid-Block, One-Time Material Map

This example shows the exact input sequence for a one-time material map.

```
{Two spaces} Number of times for specifying material map
1
Start time for map 1
0.E0
Material type grid map
1 2 3
4 5 6
7 8 9
# Name
1 S_halite
2 Drz_0
3 Trans_zn
4 S_mb139
5 Waste
6 S_mb138
7 Cavity 1
8 Cavity 2
9 Imperm_z
Number of Waste Regions
1
MAT_WASTEI
5
```

Appendix C: Example of an ASCII Flux-Field Input File (Homogeneous)

```
** DO YOU HAVE HOMOGENEOUS PROPERTIES? (T/F) **  
T  
** TITLE OF THE TEST RUN **  
'STAND ALONE FLUX FIELD TEST'  
** TIME STEP SIZE AND NUMBER **  
3.1536E4 100  
** # OF BLOCKS IN X,Y,Z **  
4      3      1  
** DIMENSION OF BLOCKS IN X,Y,Z **  
100.   100.   100.  
** POROSITY AND SATURATION OF THE MATRIX **  
0.3    .6  
** VELOCITIES IN X,Y,Z IN THE MATRIX **  
4.E-8  4.E-8  0.  
** MATRIX FLUID TEMPERATURE AND VISCOSITY **  
300.   1.E-2
```

Appendix D: Example of an ASCII Flux-Field Input File (Heterogeneous)

```
** DO YOU HAVE HOMOGENEOUS GEOMETRICAL PROPERTIES? (T/F) **
F
** TITLE OF THE TEST RUN **
'CONSTANT FIELD FOR CONVECTION-DISPERSION-DECAY IN 1D TEST'
** TIME STEP SIZE (Sec.) AND NUMBER **
8.64E5      40
** # OF GRID BLOCKS IN X,Y,Z **
60      1      1
** DIMENSION (M) OF GRID BLOCKS IN X **
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
** DIMENSION (M) OF GRID BLOCKS IN Y **
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
** DIMENSION (M) OF GRID BLOCKS IN Z **
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0 20.0
** POROSITY OF THE MATRIX GRID BLOCKS **
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
** SATURATION OF THE MATRIX GRID BLOCKS **
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
** MATRIX VELOCITY (M/s) FIELD IN X-DIRECTION **
0.0 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6 1.E-6
** MATRIX VELOCITY (M/s) FIELD IN Y-DIRECTION **
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
** MATRIX VELOCITY (M/s) FIELD IN Z-DIRECTION **
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
** MATRIX GRID BLOCKS FLUID TEMPERATURE (K) **
300.0 300.0 300.0 300.0 300.0 300.0 300.0 300.0 300.0 300.0 300.0
```

300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0

** MATRIX GRID BLOCKS FLUID VISCOSITY (Pa.Sec) **

1.E-2									
1.E-2									
1.E-2									
1.E-2									
1.E-2									
1.E-2									

Appendix E: Example of NUTS's Extensive ASCII Output File

```
IFLAGTIME      =      0
TIME STEP #    =      0
TIME (Sec.)    =      0.0000E+00
DATE OF THE RUN = 17-SEP-95
TIME OF THE RUN = 03:00:50
PROGRAM NAME    = NUTS
VERSION NUMBER  = 1.02ZO
REVISION DATE  = 15-06-95
COMPUTER SYSTEM = ALPHA
IS IT A TEST FILE? Y
OUTPUT FILE TYPE = ASC
RUN TITLE = CONVECTION-DISPERSION-DECAY IN ONE DIMENSION PROBLEM TEST_CONSTANT
FIELD FOR INPUT FILE NAME = U1:[AASHINT.CMS_NUTS.TEST.TEST4]NUCONVECTION-DISPERSION
BRAGFLO/TEST-RUN TITLE = CONSTANT FIELD FOR CONVECTION-DISPERSION-DECAY IN 1D TEST
NUTS-RUN TITLE = CONVECTION-DISPERSION-DECAY IN ONE DIMENSION PROBLEM TEST
NUTS T_CON_DISP_DEC.IN;1
BRAGFLO/TEST-RUN INPUT FILE NAME =
U1:[AASHINT.CMS_NUTS.TEST.TEST4]TEST_CON_DISP_DEC.IN;1
NUTS OUTPUT PREFIX = NUT
ASCII FILE OUTPUT FILE NAME =U1:[AASHINT.CMS_NUTS.TEST.TEST4]NUT_CON_DISP_DEC.OUT
DEBUG FILE OUTPUT FILE NAME = U1:[AASHINT.CMS_NUTS.TEST.TEST4]NUT_CON_DISP_DEC.DBG
*****
Type of the porous media      = M
Fracture is a continuum?      = F
Matrix is a continuum?        = T
Single-porosity calculation?  = T
Dual-porosity calculation?    = F
Dual-permeability calculation? = F
```

D I M E N S I O N A L I N F O R M A T I O N

```
-----
No. of Dimensions      = 1
No. of Blocks in x-direction = 60
No. of Blocks in y-direction = 1
No. of Blocks in z-direction = 1
Calculation Will be Conducted in X Direction
*****
No. of Continuum = 1
No. of Phases = 1
*****
```

---- Grid Block Length (x-direction)----

0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02

---- Grid Block Width (y-direction)----

0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02
0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02	0.2000E+02


```
COMP1          1          1
*****
Total Waste Volume (m3) of TEST4_SITE          =    0.80000E+04
*****
```

```
COMPONENT      MoL. Wt.
-----
COMP1          0.2380E+00
*****
```

A D S O R P T I O N I N F O R M A T I O N

N
!!!! No Sorption Calculation Is Considered!!!!
!!!!!!!!!!!!!!!!!!!! in the Matrix !!!!!!!!!!!!!!!!!!!

R O C K D E N S I T Y I N F O R M A T I O N

```
0.3000E+04  0.3000E+04  0.3000E+04  0.3000E+04  0.3000E+04
*****
```

L O N G . D I S P . I N F O R M A T I O N

!!!!!!! R O C K M A T R I X !!!!!!!

```
0.1000E+02  0.1000E+02  0.1000E+02  0.1000E+02  0.1000E+02
```

T R A N S . D I S P . I N F O R M A T I O N

!!!!!!! R O C K M A T R I X !!!!!!!

```
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
```

0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

ROCK TOR . INFORMATION

!!!!!!! ROCK MATRIX !!!!!!!

0.1000E+02 0.1000E+02 0.1000E+02 0.1000E+02 0.1000E+02
0.1000E+02 0.1000E+02 0.1000E+02 0.1000E+02 0.1000E+02

---- Matrix Injection History----

NO MATRIX INJECTION HISTORY AVAILABLE????????

MATRIX DIRICHLET B.CS. ARE AVAILABLE = T
FRACTURE DIRICHLET B.CS. ARE AVAILABLE = F

Matrix Dirichlet B.CS. Concentration of COMP1

0.10000E+01 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00

*** TITLE OF PRINTED MATRIX VARIABLES ***

MVCPG⁺ OF THE D-N IN THE BRINE

*** MATRIX PRINT FLAG IN THE ASCII FILE ***

0 0 0 0 0 0 1 0 0 0 0 0 0 0

MATRIX INITIALIZATION

---- Matrix Porosity Initialization----

0.2000E+00 0.2000E+00 0.2000E+00 0.2000E+00 0.2000E+00
0.2000E+00 0.2000E+00 0.2000E+00 0.2000E+00 0.2000E+00

+ see section 6.5.1.A3 for the definition of printed variables.


```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```

BRINE VOLUME IN EACH BLOCK

```
0.1600E+04 0.1600E+04 0.1600E+04 0.1600E+04 0.1600E+04
```

COMP1

MVCPG OF THE D-N IN THE BRINE

```
0.10000E+01 0.23026E+00 0.53018E-01 0.12208E-01 0.28109E-02
0.64722E-03 0.14903E-03 0.34314E-04 0.79010E-05 0.18192E-05
0.41889E-06 0.96452E-07 0.22209E-07 0.51137E-08 0.11775E-08
0.27111E-09 0.62426E-10 0.14374E-10 0.33097E-11 0.76207E-12
0.17547E-12 0.40403E-13 0.93030E-14 0.21421E-14 0.49322E-15
0.11357E-15 0.26150E-16 0.60211E-17 0.13864E-17 0.31922E-18
0.73503E-19 0.16925E-19 0.38970E-20 0.89730E-21 0.20661E-21
0.47573E-22 0.10954E-22 0.25222E-23 0.58075E-24 0.13372E-24
0.30790E-25 0.70895E-26 0.16324E-26 0.37587E-27 0.86546E-28
0.19928E-28 0.45885E-29 0.10565E-29 0.24327E-30 0.56014E-31
0.12898E-31 0.29697E-32 0.68380E-33 0.15745E-33 0.36254E-34
0.83476E-35 0.19221E-35 0.44261E-36 0.10240E-36 0.29944E-37
```

MATRIX MATERIAL BALANCE FOR NUCLIDE COMP1*

```
MATRIX MAXBLOCMBE% = 0.2118E-21
MATRIX MBE% = 0.1010E-04
MATRIX CMBE% = 0.1010E-04
MATRIX CRES = -0.2517E-21
MATRIX CSRC = 0.4786E+03
```

IFLAGTIME = 1
TIME STEP # = 2
TIME (Sec.) = 0.1728E+07

MATRIX FLUXES CROSSING J-1 AND I-1 BOUNDARIES
OF COMPONENT COMP1

```
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
```

* the definitions of the material balance entries are: MAXBLOCMBE% is the grid blocks maximum residual (kg/s), MBE% is the constituent percentage total material balance error in the spatial domain, CMBE% is the constituent percentage time cumulative total material balance error in the spatial domain, CRES is the time cumulative total residual in the spatial domain (kg), and CSRC is the time cumulative total sources in the spatial domain(kg).

```
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
*****
0.00000E+00 0.52063E-03 0.21215E-03 0.70097E-04 0.21032E-04
0.59693E-05 0.16338E-05 0.43592E-06 0.11413E-06 0.29444E-07
0.75088E-08 0.18968E-08 0.47541E-09 0.11837E-09 0.29304E-10
0.72192E-11 0.17709E-11 0.43278E-12 0.10541E-12 0.25598E-13
0.61994E-14 0.14978E-14 0.36106E-15 0.86865E-16 0.20859E-16
0.50007E-17 0.11969E-17 0.28608E-18 0.68285E-19 0.16279E-19
0.38762E-20 0.92197E-21 0.21907E-21 0.52004E-22 0.12334E-22
0.29227E-23 0.69204E-24 0.16374E-24 0.38712E-25 0.91464E-26
0.21596E-26 0.50960E-27 0.12018E-27 0.28326E-28 0.66729E-29
0.15711E-29 0.36975E-30 0.86976E-31 0.20450E-31 0.48063E-32
0.11291E-32 0.26515E-33 0.62243E-34 0.14606E-34 0.34262E-35
0.80343E-36 0.18834E-36 0.44136E-37 0.10337E-37 0.00000E+00
*****
```

MATRIX FLOW RATE AT INTERFACE I

```
0.4000E-03 0.4000E-03 0.4000E-03 0.4000E-03 0.4000E-03
*****
```

MATRIX FLOW RATE AT INTERFACE J

```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
*****
```

MATRIX TIME DEPENDENT VARIABLES

MATRIX PRODUCTION RATE FROM EACH GRID BLOCK

```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
*****
```

MATRIX INJECTION RATE INTO EACH GRID BLOCK

```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```



```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
*****
```

MATRIX INJECTION RATE INTO EACH GRID BLOCK

```
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
*****
```

BRINE VOLUME IN EACH BLOCK

```
0.1600E+04 0.1600E+04 0.1600E+04 0.1600E+04 0.1600E+04
*****
```

COMP1

MVCPG OF THE D-N IN THE BRINE

```
0.10000E+01 0.99423E+00 0.98175E+00 0.95918E+00 0.92329E+00
0.87191E+00 0.80466E+00 0.72337E+00 0.63193E+00 0.53556E+00
0.43992E+00 0.35011E+00 0.26998E+00 0.20182E+00 0.14634E+00
0.10302E+00 0.70477E-01 0.46898E-01 0.30389E-01 0.19194E-01
0.11829E-01 0.71204E-02 0.41904E-02 0.24132E-02 0.13612E-02
0.75268E-03 0.40831E-03 0.21748E-03 0.11381E-03 0.58563E-04
0.29648E-04 0.14777E-04 0.72551E-05 0.35109E-05 0.16754E-05
0.78887E-06 0.36665E-06 0.16830E-06 0.76323E-07 0.34211E-07
0.15163E-07 0.66478E-08 0.28839E-08 0.12384E-08 0.52656E-09
0.22176E-09 0.92530E-10 0.38263E-10 0.15685E-10 0.63752E-11
0.25701E-11 0.10278E-11 0.40787E-12 0.16064E-12 0.62802E-13
0.24378E-13 0.93980E-14 0.36023E-14 0.14022E-14 0.78821E-15
*****
```

MATRIX MATERIAL BALANCE FOR NUCLIDE COMP1

```
MATRIX MAXBLOCMBE% = 0.4337E-18
MATRIX MBE% = 0.7946E-05
MATRIX CMBE% = 0.3300E-03
MATRIX CRES = -0.2574E-16
MATRIX CSRC = 0.1461E+05
*****
```

Information Only

```
Stand-alone build file:
$ nuts_main = "NUT_MAIN"
$ if pl .eqs. " "
$ then
$ inquire/nopunct/local inp "Debug? Y/N (n) "
$ if inp .eqs. "Y"
$ then
$ pl="DEBUG"
$ else
$ pl= " "
$ endif
$ endif
$ if pl .eqs. "DEBUG"
$ then
$ fortran_switches = "/separate/noopt/debug"
$ link_switches = "/debug/exe=nuts_debug_standalone"
$ appended_info = "a DEBUG version of NUTS"
$ else
$ nuts == "$" + $environment("DEFAULT") + "nuts_debug"
$ fortran_switches = "/separate/assume=dummy"
$ link_switches = "/exe=nuts_standalone"
$ appended_info = "NUTS"
$ nuts == "$" + $environment("DEFAULT") + "nuts"
$ endif
$ write sys$output "Compiling and Linking " + appended_info
$ write sys$output " Compiling " + nuts_main
$ fortran 'fortran_switches', 'nuts_main'
$ write sys$output " Compiling NUT_LIB_STANDALONE "
$ lib/cre nut_lib_standalone
$ lib/fortran 'fortran_switches', nut_lib_standalone
$ lib/repl nut_lib_standalone nut_lib_standalone
$ write sys$output " Linking..."
$ link 'link_switches', nuts_main, nut_lib_standalone/1
$ exit
```

Appendix F: Command File to Build NUTS-SA's Executable

Appendix G: Command File to Build NUTS-CC's Executable

```
$ nuts_main = "NUT_MAIN"
$
$ if p1 .eqs. ""
$   then
$     inquire/nopunct/local inp "Debug? Y/N (n) "
$     if inp .eqs. "Y"
$       then
$         p1="DEBUG"
$       else
$         p1= ""
$       endif
$     endif
$   endif
$
$ if p1 .eqs. "DEBUG"
$   then
$     fortran_switches = "/separate/noopt/debug"
$     link_switches = "/debug/exe=nuts_debug"
$     appended_info = "a DEBUG version of NUTS"
$     nuts == "$" + f$environment("DEFAULT") + "nuts_debug"
$   else
$     fortran_switches = "/separate/assume=dummy"
$     link_switches = "/exe=nuts"
$     appended_info = "NUTS"
$     nuts == "$" + f$environment("DEFAULT") + "nuts"
$   endif
$
$ write sys$output "Compiling and Linking " + appended_info
$
$ write sys$output "  Compiling " + nuts_main
$ fortran 'fortran_switches' 'nuts_main'
$
$ write sys$output "  Compiling NUT_LIB"
$ lib/cre nut_lib
$ fortran 'fortran_switches' nut_lib!.for_standalone
$ lib/cre nut_lib nut_lib
$
$ write sys$output "  Compiling NUTS_CDBLIB"
$ lib/cre nut_cdblib
$ fortran 'fortran_switches' nut_cdblib
$ lib/cre nut_cdblib nut_cdblib
$
$ write sys$output "  Linking..."
$ link 'link_switches' 'nuts_main',nut_lib/1,nut_cdblib/1, -
      camdat_lib/1, camcon_lib/1, camsupes_lib/1
$
$ exit
```

Appendix H: Command Files and ASCII Files that Execute NUTS on the WIPP Alpha-VAX Cluster

This Appendix includes examples of some general cases encountered in the CCA calculation and the QA testing and validation documents using the NUTS_CC executable and one general case of NUTS_SA executable. For further details, see explanations of inputs in Chapter 6.

Example (1): Single-porosity, matrix, tracer test conducted by implementing Dirichlet boundary conditions in the waste region with concentration equal 1 and infinite solubility. There are two input files: Binary input for the flux field from BRAGFLO (BF3_CCA_R1_S1_V23.CDB) and an ASCII file provided by the user (NUT_CCA_SCN_R1_S1.INP). The material map is read from BRAGFLO ASCII input (BF2_CCA_R1_S1_V23.INP). In this example two output files are required, they are: ASCII debug file (NUT_CCA_SCN_R1_S1_V023.OUT) and NUTS' CDB binary file (NUT_CCA_SCN_R1_S1_V023.CDB). The constituent properties are provided by NUTS and Liquid phase transport is conducted.

Interactive Input Command File (TRACER.COM)

```
$ SET DEFAULT DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]
$ NUTS:==$N1:[NOBACK.AASHINT.NMVP.NMVP96.CCASRC.TEMP]NUTS.EXE ! NUTS EXECUTABLE PATH.
$ NUTS -
  DISK$TINA_CCA3:[BF.DATA.R1S1]BF3_CCA_R1_S1_V23.CDB -! BRAGFLO CDB INPUT.
  N - ! TEST (Y/N)?.
  BIN - ! ASC/BIN (CDB) OUTPUT.
  NUT_CCA_SCN_R1_S1.INP - ! NUTS ASCII INPUT.
  NUT_CCA_SCN_R1_S1_V023.CDB - !NUTS CDB OUTPUT.
  NUT_CCA_SCN_R1_S1_V023.OUT - !NUTS DEBUG FILE.
  L - ! LIQUID/GAS TRANSPORT.
  DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]BF2_CCA_R1_S1_V23.INP - ! BRAGFLO ASC INPUT.
  N - ! NUCLIDE INPUT DATA (CDB (C)/NUTS (N)) .
  CANCEL - ! PROPERTY CDB (SOLUBILITY DATA).
  CANCEL ! NUTS UNDISTRUBED CDB OF THE SAME VECTOR.
$ EXIT
```

NUTS' ASCII Input (NUT_CCA_SCN_R1_S1.INP)

```
** NUTS TITLE **
'NUTS TRACER SCREENING TEST FOR CCA R1S1 (UNDISTURBED SCENARIO) '
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,46
'WIPP_SITE' 1
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'WIPP_SITE'
'TWASTE' 'NONE' 'NONE' 'WASTE'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
1
'WASTE' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY(IS PH REQUIRED (Y/N)) **
'N'
** ORDER OF THE METHOD **
1
** DEGREE OF IMPLICITNESS **
1.DO
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
```

```
** IS MATRIX SORPTION REQUIRED (Y/N) **  
'N'  
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **  
'N'  
** DOES MATRIX HAVE SYMMETRIC DISPERSION (T/F): ANSWER IF DISPERSION IS Y **  
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **  
'N'  
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **  
.TRUE.  
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **  
.FALSE.  
** OPEN NUTS UNDISTURBED CDB FOR INTRUSION TIME OTHER THAN 350,1000 YRS **  
.FALSE.  
** PRINT FLAGS OF MATRIX VARIABLES IN A BINARY FILE **  
0,0,0,0,0,0,1,0,0,0,0,0,0,0  
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **  
** PRINTING FREQUENCY IN A BINARY FILE **  
1,1.D14  
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **  
.FALSE.  
** MINIMUM LIMIT OF TIME TO BE SET IF ZERO ENCOUNTERED **  
1.D-18  
** INTRUSION TIME, ITERPOLATED INTRUSION TIME, TOLERANCE **  
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***  
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,...,NELEMENT **  
'WASTE' -2.D0  
** (COMP. NAME, MOL.(ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...,NUCLIDE **  
'TWASTE' .1D0 0.D0 0.D0 0.D0  
** GROUND WATER PH INPUT **  
** STANDARD BR. DENS. IF NOT BRAGFLO RUN (READ ASCII FILE FOR FLUX FIELD) **  
** MOLECULAR DIFFUSION OF EACH COMPONENT **  
** ROCK GRAIN DENSITY INPUT (REQUIRED ONLY IF SORPTION OR SOIL BASE CONC.) **  
** WASTE MATRIX INPUT (1.# OF ISO,2.NAME, LOC. IN THE INPUT, WASTE SITE #) **  
1  
'TWASTE' 1 1  
*** (1.SITE NAME, NUMBER OF GRIDS IN THE SITE 2.INDECES)1...NSITES ***  
'WIPP_SITE' 30  
 9,8,1 10,8,1 11,8,1 12,8,1 13,8,1 14,8,1 15,8,1  
17,8,1 18,8,1 19,8,1  
 9,9,1 10,9,1 11,9,1 12,9,1 13,9,1 14,9,1 15,9,1  
17,9,1 18,9,1 19,9,1  
 9,10,1 10,10,1 11,10,1 12,10,1 13,10,1 14,10,1 15,10,1  
17,10,1 18,10,1 19,10,1  
** MATRIX SORPTION INPUT **  
** MATRIX DISPERSION INPUT **  
** MATRIX SOURCE INPUT (INJECTED NUCLIDES IF ANY) **  
** MATRIX DIR. B.CS. INPUT (REP.='GENERAL',ANYWHERE='NOT_GENERAL') **  
1 'NOT_GENERAL'  
'TWASTE' 1 30  
 9,8,1 10,8,1 11,8,1 12,8,1 13,8,1 14,8,1 15,8,1  
17,8,1 18,8,1 19,8,1  
 9,9,1 10,9,1 11,9,1 12,9,1 13,9,1 14,9,1 15,9,1  
17,9,1 18,9,1 19,9,1  
 9,10,1 10,10,1 11,10,1 12,10,1 13,10,1 14,10,1 15,10,1  
17,10,1 18,10,1 19,10,1  
'TWASTE'  
1.D0  
1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0  
1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0 1.D0  
** TIME DEPENDENT SOURCE IN THE MATRIX **  
** MATRIX CONCENTRATION INITIALIZATION **  
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Example (2): Single-porosity, matrix, radiological transport with five isotopes. Three single isotopes and one chain of two members are used. The repository location is described by the waste matrix input. The inventories, atomic weights, half-life, and solubilities are provided by the properties CDB. There are three input files: CDB Binary input file for the flux field from

BRAGFLO (BF3_CCA_R1_S4_V23.CDB) and an ASCII file provided by the user (NUT_CCA_ISO_R1_S4.INP). The material map is read from BRAGFLO ASCII input (BF2_CCA_R1_S4_V23.INP), whereas the radioactive material properties are read from the CDB binary input file NUT_CCA_PROP_R1_S4_V023.INP.

In this example, liquid transport is conducted and two output files are required, they are: ASCII debug file (NUT_CCA_ISO_R1_S4_V023.OUT) and NUTS' CDB binary file (NUT_CCA_ISO_R1_S4_V023.CDB).

Interactive Input Command File (ISO.COM)

```
$ SET DEFAULT DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]
$ NUTS:==$N1:[NOBACK.AASHINT.NMVP.NMVP96.CCASRC.TEMP]NUTS.EXE ! NUTS EXECUTABLE PATH.
$ NUTS -
  DISK$TINA_CCA3:[BF.DATA.R1S4]BF3_CCA_R1_S4_V23.CDB - ! BRAGFLO CDB INPUT.
  N - ! TEST (Y/N)?
  BIN - ! ASC/BIN (CDB) OUTPUT.
  NUT_CCA_ISO_R1_S4.INP - ! NUTS ASCII INPUT.
  NUT_CCA_ISO_R1_S4_V023.CDB - ! NUTS CDB OUTPUT.
  NUT_CCA_ISO_R1_S4_V023.OUT - ! NUTS DEBUG FILE.
  L - ! LIQUID/GAS TRANSPORT.
  DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]BF2_CCA_R1_S4_V23.INP - ! BRAGFLO ASC INPUT.
  CDB - ! NUCLIDE INPUT DATA (CDB (C)/NUTS (N)) .
  NUT_CCA_PROP_R1_S4_V023.INP - ! PROPERTY CDB (SOLUBILITY DATA).
  CANCEL ! NUTS UNDISTRUBED CDB OF THE SAME VECTOR.
$ EXIT
```

NUTS' ASCII Input (NUT_CCA_ISO_R1_S4.INP)

```
** NUTS TITLE **
'NUTS RADIOLOGICAL TRANSPORT FOR CCA R1S4 (E2 SCENARIO, 350 YRS INTRUSION)'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,49
'WIPP_SITE' 5
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'WIPP_SITE'
'AM241L' 'NONE' 'NONE' 'AML'
'PU239L' 'NONE' 'NONE' 'PUL'
'PU238L' 'NONE' 'NONE' 'PUL'
'U234L' 'TH230L' 'NONE' 'UL'
'TH230L' 'TH230D' 'NONE' 'THL'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
4
'AML' .FALSE. .FALSE.
'PUL' .FALSE. .FALSE.
'UL' .FALSE. .FALSE.
'THL' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY(IS PH REQUIRED (Y/N)) **
'N'
** ORDER OF THE METHOD **
1
** DEGREE OF IMPLICITNESS **
1.D0
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2.# OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'N'
** DOES MATRIX HAVE SYMMETRIC DISPERSION (T/F): ANSWER IF DISPERSION IS Y **
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
```

```
'N'  
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **  
.FALSE.  
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **  
.FALSE.  
** OPEN NUTS UNDISTURBED CDB FOR INTRUSION TIME OTHER THAN 350,1000 YRS **  
.FALSE.  
** PRINT FLAGS OF MATRIX VARIABLES IN A BINARY FILE **  
0,0,0,0,0,0,1,1,1,0,0,0,0,1  
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **  
** PRINTING FREQUENCY IN A BINARY FILE **  
1,1.D12  
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **  
.FALSE.  
** MINIMUM LIMITS OF TIME TO BE SET IF ZERO ENCOUNTERED **  
1.D-18  
** INTRUSION TIME, ITERPOLATED INTRUSION TIME, TOLERANCE **  
** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***  
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,...NELEMENT **  
** (COMP. NAME, MOL.(ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,...NUCLIDE **  
** GROUND WATER PH INPUT **  
** STANDARD BR. DENS. IF NOT BRAGFLO RUN (READ ASCII FILE FOR FLUX FIELD) **  
** MOLECULAR DIFFUSION INPUT **  
** ROCK DENSITY INPUT **  
** WASTE MATRIX INPUT (LOCATION OF THE WASTE) **  
5  
'AM241L' 1 1  
'PU239L' 2 1  
'PU238L' 3 1  
'U234L' 4 1  
'TH230L' 5 1  
*** (1.SITE NAME, NUMBER OF GRIDS IN THE SITE 2.INDECES)1...NSITES ***  
'WIPP_SITE' 30  
 9,8,1 10,8,1 11,8,1 12,8,1 13,8,1 14,8,1 15,8,1  
17,8,1 18,8,1 19,8,1  
 9,9,1 10,9,1 11,9,1 12,9,1 13,9,1 14,9,1 15,9,1  
17,9,1 18,9,1 19,9,1  
 9,10,1 10,10,1 11,10,1 12,10,1 13,10,1 14,10,1 15,10,1  
17,10,1 18,10,1 19,10,1  
** MATRIX SORPTION INPUT **  
** MATRIX DISPERSION INPUT **  
** MATRIX SOURCE INPUT (INJECTED NUCLIDES IF ANY) **  
** MATRIX DIR. B.CS. INPUT (REP.='GENERAL',ANYWHERE='NOT_GENERAL') **  
** TIME DEPENDENT SOURCE IN THE MATRIX **  
** MATRIX CONCENTRATION INITIALIZATION **  
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Example (3): Single-porosity, matrix, radiological transport with five isotopes. Three single isotopes and one chain of two members are used. The repository location is described by the waste matrix input. The inventories, atomic weights, half-life, and solubilities are provided by the properties CDB. There are four input files: CDB Binary input file for the flux field from BRAGFLO (BF3_CCA_R1_S5_V23.CDB) and an ASCII file provided by the user (NUT_CCA_INT3_R1_S5_V023.INP). The material map is read from BRAGFLO ASCII input (BF2_CCA_R1_S5_V23.INP), whereas the radioactive material properties are read from the CDB binary input file NUT_CCA_PROP_R1_S5_V023.INP. There is an additional CDB input file required to initialization the intrusion time calculation at 3000 years. This file is the output of NUTS for the undisturbed scenario of vector 23 (NUT_CCA_ISO_R1_S1_V023.INP).

In this example, liquid transport is conducted and two output files are required, they are: ASCII debug file (NUT_CCA_INT3_R1_S5_V023.OUT) and NUTS' CDB binary file (NUT_CCA_INT3_R1_S5_V023.CDB).

Interactive Input Command File (INT3.COM)

```
$ SET DEFAULT DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]
$ NUTS:==$N1:[NOBACK.AASHINT.NMVP.NMVP96.CCASRC.TEMP]NUTS.EXE ! NUTS EXECUTABLE PATH.
$ NUTS -
  DISK$TINA_CCA3:[BF.DATA.R1S5]BF3_CCA_R1_S5_V23.CDB -      ! BRAGFLO CDB INPUT.
  N -                                                         ! TEST (Y/N)? .
  BIN -                                                       ! ASC/BIN (CDB) OUTPUT.
  NUT_CCA_INT3_R1_S5_V023.INP -                               ! NUTS ASCII INPUT.
  NUT_CCA_INT3_R1_S5_V023.CDB -                               ! NUTS CDB OUTPUT.
  NUT_CCA_INT3_R1_S5_V023.OUT -                               ! NUTS DEBUG FILE.
  L -                                                         ! LIQUID/GAS TRANSPORT.
  DISK$TINA_CCA3:[BF.AASHINT.CCA3.INPUT]BF2_CCA_R1_S5_V23.INP - ! BRAGFLO ASC INPUT.
  CDB -                                                       ! NUCLIDE INPUT DATA (CDB (C)/NUTS (N)).
  NUT_CCA_PROP_R1_S5_V023.INP -                               ! PROPERTY CDB (SOLUBILITY DATA).
  NUT_CCA_ISO_R1_S5_V023.CDB                                 ! NUTS UNDISTRUBED CDB OF THE SAME VECTOR.
$ EXIT
```

NUTS' ASCII Input (NUT_CCA_INT3_R1_S5_V023.INP)

```
** NUTS TITLE **
'NUTS RADIOLOGICAL TRANSPORT FOR CCA R1S5V023 (E2 SCENARIO, 3000 YRS INTRUSION)'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,49
'WIPP_SITE'      5
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'WIPP_SITE'
'AM241L' 'NONE' 'NONE' 'AML'
'PU239L' 'NONE' 'NONE' 'PUL'
'PU238L' 'NONE' 'NONE' 'PUL'
'U234L'  'TH230L' 'NONE' 'UL'
'TH230L' 'TH230D' 'NONE' 'THL'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
4
'AML'  .FALSE. .FALSE.
'PUL'  .FALSE. .FALSE.
'UL'   .FALSE. .FALSE.
'THL'  .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** PH DEPENDENT SOLUBILITY(IS PH REQUIRED (Y/N)) **
'N'
** ORDER OF THE METHOD **
1
```

```
** DEGREE OF IMPLICITNESS **
1.D0
** IMPLICIT PRECIPITATE 1.(T/F),IF TRUE,2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS MATRIX SORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'N'
** DOES MATRIX HAVE SYMMETRIC DISPERSION (T/F): ANSWER IF DISPERSION IS Y **
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.FALSE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.FALSE.
** OPEN NUTS UNDISTURBED CDB FOR INTRUSION TIME OTHER THAN 350,1000 YRS **
.TRUE.
** PRINT FLAGS OF MATRIX VARIABLES IN A BINARY FILE **
0,0,0,0,0,0,1,1,1,0,0,0,0,1
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** PRINTING FREQUENCY IN A BINARY FILE **
1,1.D12
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.FALSE.
** MINIMUM LIMITS OF TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** INTRUSION TIME, ITERPOLATED INTRUSION TIME, TOLERANCE **
0.3155693D11 0.946708D11 0.D0
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,..,NELEMENT **
** (COMP. NAME, MOL.(ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,..,NUCLIDE **
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN (READ ASCII FILE FOR FLUX FIELD) **
** MOLECULAR DIFFUSION INPUT **
** ROCK DENSITY INPUT **
** WASTE MATRIX INPUT (LOCATION OF THE WASTE) **
5
'AM241L' 1 1
'PU239L' 2 1
'PU238L' 3 1
'U234L' 4 1
'TH230L' 5 1
*** (1.SITE NAME, NUMBER OF GRIDS IN THE SITE 2.INDECES)1...NSITES ***
'WIPP_SITE' 30
 9,8,1 10,8,1 11,8,1 12,8,1 13,8,1 14,8,1 15,8,1
17,8,1 18,8,1 19,8,1
 9,9,1 10,9,1 11,9,1 12,9,1 13,9,1 14,9,1 15,9,1
17,9,1 18,9,1 19,9,1
 9,10,1 10,10,1 11,10,1 12,10,1 13,10,1 14,10,1 15,10,1
17,10,1 18,10,1 19,10,1
** MATRIX SORPTION INPUT **
** MATRIX DISPERSION INPUT **
** MATRIX SOURCE INPUT (INJECTED NUCLIDES IF ANY) **
** MATRIX DIR. B.CS. INPUT (REP.='GENERAL',ANYWHERE='NOT_GENERAL') **
** TIME DEPENDENT SOURCE IN THE MATRIX **
** MATRIX CONCENTRATION INITIALIZATION **
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Example (4): Single-porosity, matrix, radiological transport of three isotopes in single chain, and implicit precipitate treatment. The waste location is described by the material map. The inventories, atomic weights, half-life, and solubilities are provided by NUTS ASCII input file. There are two ASCII input files provided by the user, NUT_PUT.IN and NUT_TEST_PUT.IN. The material map and the radioactive material properties are read from NUTS ASCII input file NUT_PUT.IN, whereas the flux and rock properties data are read from NUT_TEST_PUT.IN. In this example, liquid transport is conducted and two output files are required, they are: the ASCII debug file NUT_TP_PUT.DBG and the regular NUTS ASCII output file NUT_TP_PUT.OUT.

Interactive Input Command File (PRECIP.COM)

```
$!  
$ WRITE SYSS$OUTPUT " "  
$ WRITE SYSS$OUTPUT " 2-dimentional Hypothetical Repository"  
$ WRITE SYSS$OUTPUT " Three member decay chain"  
$ WRITE SYSS$OUTPUT " "  
$!  
$! *****  
$! *  
$! RUN NUTS ON 2D PURE ADVECTIVE WITH UNIFORM VELOCITY FIELD *  
$! *  
$! *****  
$!  
$! POINT TO THE NAME AND LOCATION OF NUTS EXECUTABLE  
$!  
$ NUTS:== $WP$PRODROOT:[NUT.EXE]NUTS_QA0205.EXE  
$ NUTS -  
    NUT_TEST_TP_PUT.IN - !TEST FLUX FIELD INPUT  
    Y - !TEST RUN? (y)  
    ASC - !ASC/BIN OUTPUT  
    NUT_TP_PUT.IN - !NUTS INPUT FILE  
    NUT_TP_PUT.OUT - !NUTS OUTPUT ASCII FILE  
    NUT_TP_PUT.DBG - !DEBUG FILE  
    L - !LIQUID/GAS  
    CANCEL - !(WOULD HAVE BEEN BRAGFLO.INP)  
    N - !READ NUCLIDE DATA FROM NUTS INPUT  
    CANCEL - !PROPERTY CDB  
    CANCEL !ONLY IN INTRUSION, UNDISTURBED NUTS RESULTS OF THE SAME VECTOR  
$!  
$ WRITE SYSS$OUTPUT " "  
$ WRITE SYSS$OUTPUT " End of the test"  
$ WRITE SYSS$OUTPUT " "  
$!  
$ EXIT
```

NUTS' ASCII Input (NUT_PUT.IN)

```
** NUTS TITLE **  
'NUTS INPUT FOR 2D PRECIPITATE TEST IN UNIFORM VELOCITY FIELD'  
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **  
1,3  
'PRECIP_SITE' 3  
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **  
'PRECIP_SITE'  
'PU238' 'U234' 'NONE' 'PU'  
'U234' 'TH230' 'PU238' 'U'  
'TH230' 'NONE' 'U234' 'TH'  
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **  
3  
'PU' .FALSE. .FALSE.  
'U' .FALSE. .FALSE.  
'TH' .FALSE. .FALSE.  
** COLLOIDAL TRANSPORT FLAG (T/F) **  
.FALSE.
```

```
** PH DEPENDENT SOLUBILITY( IS PH REQUIRED (Y/N)) **
'N'
***** ORDER OF THE METHOD *****
1
***** DEGREE OF IMPLICITNESS *****
1.
***** IMPLICIT PRECIPITATE (F/T) *****
.TRUE.
40 1.D-6
** IS MATRIX ADSORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE MATRIX (Y/N) **
'N'
** DISPERSION IS SYMMETRIC (T/F), INPUT FROM NUTS (T/F) ***
** DO YOU HAVE INJECTION/PRODUCTION IN THE MATRIX (Y/N) **
'N'
** DO YOU HAVE DIRICHLET B.CS. IN THE MATRIX (F/T) **
.TRUE.
** MATIRX INITIAL. IN THE CDB FOR INTERPOLATED INTURSION TIME IN E1 SCENARIO **
.FALSE.
** IS CONCENTRATION INITIALIZED MANUALLY IN THE MATRIX (F/T) **
.FALSE.
** PRINT FLAGS OF MATRIX VARIABLES IN AN ASCII FILE**
0,0,0,0,0,0,1,0,0,0,0,0,0
** SPECIFIC RANGE OF DATA TO BE PRITED IN AN ASCII FILE **
'N'
** TEMP. DEPEND. OF Kd (ENTER DATA IF ADSORP. IS (Y) AND TEMP. DEPEND.) **
** PRINTING FREQUENCY IN AN ASCII FILE **
10,1.E12
** DO YOU HAVE EXTERNAL NUCLIDE SOURCE? (T/F) **
.FALSE.
** MINIMUM LIMITS OF SATURATION AND TIME TO BE SET IF ZERO ENCOUNTERED **
1.D-18
** IF CDB INITIAL., INTRUSION TIME, INTERPOLATED TIME, TOLERANCE IN SECONDS **
NUMBER OF TIMES FOR SPECIFING MATERIAL MAP
1
START TIME FOR MAP 1
0.D0
MATERIAL TYPE GRID MAP
1*3 5*1 1*2 5*1 1*3
1*3 5*1 1*2 5*1 1*3
1*3 5*1 1*2 5*1 1*3
6*3 1*2 6*3
6*3 1*2 6*3
6*3 1*2 6*3
6*3 1*2 6*3
6*3 1*2 6*3
6*3 1*2 6*3
6*3 1*2 6*3
# NAME
1 WASTE
2 BOREHOLE
3 IMPERMEABLE
NUMBER OF WASTE REGIONS
1
WASTE
1
1
*** END MATERIAL MAP AND START NUCLIDES PROPERTIES ***
** IF NOT TEMP. DEPEND. (ELEMENT NAME, SOLUBILITY LIMIT) 1,..,NELEMENT **
'PU' -6.40787D0
'U' -9.15607D0
'TH' -1.00535D1
** (COMP. NAME, MOL. (ATOMIC) WT., INITIAL INVENTS., HALF LIFE)1,..,NUCLIDE **
'PU238' 0.238 2.610D6 1.450D3 2.769D9
'U234' 0.234 2.255D3 2.007D2 7.716D12
'TH230' 0.230 2.9606D0 1.2456D-1 2.430D12
** GROUND WATER PH INPUT **
** STANDARD BR. DENS. IF NOT BRAGFLO RUN ( ASCII FILE FOR FLUX FIELD) **
1000.
```

```
** MOLECULAR DIFFUSION OF EACH COMPONENT **
** ROCK DENSITY INPUT **
** WASTE MATRIX INPUT **
3
'PU238'    1    1
'U234'    2    1
'TH230'   3    1
** (1. SITE NAME, # OF GRID 2. INDECES OF WASTE MATRIX)1...NSITE **
'PRECIP_SITE'  30
2,1,1 3,1,1 4,1,1 5,1,1 6,1,1 8,1,1 9,1,1 10,1,1 11,1,1 12,1,1
2,2,1 3,2,1 4,2,1 5,2,1 6,2,1 8,2,1 9,2,1 10,2,1 11,2,1 12,2,1
2,3,1 3,3,1 4,3,1 5,3,1 6,3,1 8,3,1 9,3,1 10,3,1 11,3,1 12,3,1
** MATRIX ADSORPTION INPUT **
** MATRIX DISPERSION INPUT **
** MATRIX SOURCE (INJECTION/PRODUCTION) INPUT IF ANY **
** MATRIX DIR. B.CS. (REP.='GENERAL, ANYWHERE='NOT_GENERAL') **
3 'NOT_GENERAL'
'PU238'    1    1
7,10,1
'U234'    2    1
7,10,1
'TH230'   3    1
7,10,1
'PU238'
0.D0
'U234'
0.D0
'TH230'
0.D0
** TIME DEPENDENT SOURCE IN THE MATRIX ****
** MATRIX CONCENTRATION INITIALIZATION **
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

Flux and Hydrological ASCII Input (NUT_TEST_PUT.IN)

```
** DO YOU HAVE HOMOGENEOUS GEOMETRICAL PROPERTIES? (T/F) **
F
** TITLE OF THE TEST RUN **
'FLUX FIELD FOR PRECIPITATE 2D TEST IN UNIFORM VELOCITY FIELD'
** TIME STEP SIZE (Sec.) AND NUMBER **
3.1536D7    100
** # OF GRID BLOCKS IN X,Y,Z **
13  10  1
** DIMENSION (M) OF GRID BLOCKS IN X **
6*80.E0 1*0.2D0 6*80.E0
** DIMENSION (M) OF GRID BLOCKS IN Y **
130*80.0
** DIMENSION (M) OF GRID BLOCKS IN Z **
130*10.0
** POROSITY OF THE MATRIX GRID BLOCKS **
130*0.2
** SATURATION OF THE MATRIX GRID BLOCKS **
130*1.0
** MATRIX VELOCITY (M/Sec) FIELD IN X-DIRECTION **
0.E0 6*1.E-11 6*-1.E-11
0.E0 6*1.E-11 6*-1.E-11
0.E0 6*1.E-11 6*-1.E-11
13*0.E0
13*0.E0
13*0.E0
```

```
13*0.E0
13*0.E0
13*0.E0
13*0.E0
** MATRIX VELOCITY (M/Sec) FIELD IN Y-DIRECTION **
13*0.E0
6*0.E0 1*1.E-11 6*0.E0
** MATRIX VELOCITY (M/Sec) FIELD IN Z-DIRECTION **
130*0.0
** MATRIX GRID BLOCKS FLUID TEMPERATURE (K) **
130*300.0
** MATRIX GRID BLOCKS FLUID VISCOSITY (Pa.Sec) **
130*1.E-2
```

Example (5): Dual-permeability, One Site, Eleven Component, No Temperature Dependency, No waste emplacement, Injection in the Matrix only, No Dirichlet B.Cs., No External Dynamic Source, Material-map from Input, and writing in an ASCII output.

Interactive Input Command File (DUAL.COM)

```
$ RUN N1: [NOBACK.AASHINT.NMVP.NMVP96.CCASRC.TEMP]NUTS_STANDALONE
N
N
DUALNUTS.IN
Y
N
Y
DUALTEST.IN
ASC
DPM
1
L
VAX
ALPHA
```

DUALNUTS.IN

```
** NUTS TITLE **
'NUTS TEST RUN EXAMPLE'
** 1.# OF SITES, # OF MATERIAL, (2.SITE NAME, # OF COMP. TO BE MODELED)1,...,NSITES **
1 7
'TEST_SITE' 11
** (1.SITE NAME, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,...,NSITES **
'TEST_SITE'
'PU239' 'U235' 'NONE' 'PU'
'U235' 'PA231' 'PU239' 'U'
'PA231' 'NONE' 'U235' 'PA'
'NP237' 'U233' 'NONE' 'NP'
'U233' 'NONE' 'NP237' 'U'
'U238' 'U234' 'NONE' 'U'
'U234' 'NONE' 'U238' 'U'
'I129' 'NONE' 'NONE' 'I'
'TC99' 'NONE' 'NONE' 'TC'
'SE79' 'NONE' 'NONE' 'SE'
'U235' 'NONE' 'NONE' 'U'
** 1.# OF ELEMENT, (2.ELEMENT NAME, TEMP. DEPENDENCY, TABLE LOOK-UP)1,...,NELEMENT **
7
'U' .FALSE. .FALSE.
'PU' .FALSE. .FALSE.
'PA' .FALSE. .FALSE.
'NP' .FALSE. .FALSE.
'I' .FALSE. .FALSE.
'TC' .FALSE. .FALSE.
'SE' .FALSE. .FALSE.
** COLLOIDAL TRANSPORT FLAG (T/F) **
.FALSE.
** pH DEPENDENT SOLUBILITY (IS pH REQUIRED (Y/N)) **
'N'
** ORDER OF THE METHOD **
1
** DEGREE OF IMPLICITNESS **
1.
** IMPLICIT PRECIPITATE 1. (T/F), IF TRUE, 2. # OF ITERATION, TOLERANCE **
.FALSE.
** IS FRACTURE SORPTION REQUIRED (Y/N) **
'N'
** DO YOU HAVE DISPERSION IN THE FRACTURE (Y/N) **
'Y'
```



```
'U233 '      0.233      0.      0.      5.00E12
'U238 '      0.238      0.      0.      1.41E17
'U234 '      0.234      0.      0.      7.72E12
'I129 '      0.129     110.     0.      4.95E14
'TC99 '      0.99       0.      0.      6.72E12
'SE79 '      0.79       0.      0.      2.05E12
'U236 '      0.236     1.45E4  0.      7.39E14
** GROUND WATER pH INPUT **
** STANDARD BRINE DENSITY IF NOT BRAGFLO RUN (ASCII FILE FOR FLUX FIELD) **
1000.
** MOLECULAR DIFFUSION OF EACH COMPONENT **
'PU239'      1.00E-9
'U235 '      1.00E-9
'PA231'      1.00E-9
'NP237'      1.00E-9
'U233 '      1.00E-9
'U238 '      1.00E-9
'U234 '      1.00E-9
'I129 '      1.00E-9
'TC99 '      1.00E-9
'SE79 '      1.00E-9
'U236 '      1.00E-9
** REFERENCE VISCOSITY AND TEMPERATURE **
** ROCK DENSITY INPUT **
1993  1746  1838  2021  2283  1714  2366
** WASTE MATRIX INPUT **
3
'PU239'      1      1
'I129 '      8      1
'U236 '      11     1
** (SITE NAME, NO. OF GRID IN THE SITE, 2. INDICES), 1,..., NSITES **
'TEST_SITE'  1
28 15 1
** FRACTURE SORPTION INPUT **
** FRACTURE DISPERSION INPUT **
** FRACTURE LONGITUDINAL MATERIAL DISPERSIVITIES **
7*1.D0
** FRACTURE TRANSVERSE MATERIAL DISPERSIVITIES **
7*1.D0
** FRACTURE MATERIAL TORTUOSITY **
7*1.D0
** FRACTURE SOURCE INPUT (INJECTION PRODUCTION) IF ANY **
** FRACTURE DIRICHLET BOUNDARY CONDITIONS, IF ANY **
** TIME DEPENDENT SOURCE IN THE FRACTURE **
** FRACTURE MANUAL CONCENTRATION INITIALIZATION **
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE FRACTURE **
** MATRIX SORPTION **
'PU239'
1.00E-1  4.00E-2  1.00E-1  4.00E-2  1.00E-1  4.00E-2  1.00E-1
'U235 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
'PA231'
2.00E-3  4.00E-3  5.00E-4  4.00E-3  1.25E-3  4.00E-3  2.00E-3
'NP237'
2.00E-3  4.00E-3  5.00E-4  4.00E-3  1.25E-3  4.00E-3  2.00E-3
'U233 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
'U238 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
'U234 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
'I129 '
0.00E0  0.00E0  0.00E0  0.00E0  0.00E0  0.00E0  0.00E0
'TC99 '
0.00E0  0.00E0  0.00E0  0.00E0  0.00E0  0.00E0  0.00E0
'SE79 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
'U236 '
2.50E-3  1.00E-2  2.00E-3  1.00E-2  2.25E-3  1.00E-2  2.50E-3
** MATRIX DISPERSION INPUT **
```

```
** MATRIX LONGITUDINAL MATERIAL DISPERSIVITIES **
1.30E02 1.70E02 1.10E02
1.70E02 1.20E02 1.70E02
1.30E02
** MATRIX TRANSVERSE MATERIAL DISPERSIVITIES **
1.30E01 1.70E01 1.10E01
1.70E01 1.20E01 1.70E01
1.30E01
** MATRIX MATERIAL TORTUOSITY **
10. 10. 10.
10. 10. 10.
10.
** MATRIX SOURCE INPUT (INJECTED NUCLIDES IF ANY) **
11
'PU239'          1      1
28              15     1
'U235 '         2      1
28              15     1
'PA231'         3      1
28              15     1
'NP237'         4      1
28              15     1
'U233 '         5      1
28              15     1
'U238 '         6      1
28              15     1
'U234 '         7      1
28              15     1
'I129 '         8      1
28              15     1
'TC99 '         9      1
28              15     1
'SE79 '        10     1
28              15     1
'U236 '        11     1
28              15     1
'PU239'
1.E-7
'U235 '
1.E-7
'PA231'
1.E-7
'NP237'
1.E-7
'U233 '
1.E-7
'U238 '
1.E-7
'U234 '
1.E-7
'I129 '
1.E-7
'TC99 '
1.E-7
'SE79 '
1.E-7
'U236 '
1.E-7
** INJECTION RATE IN THE MATRIX **
1      0.      1.E30
28     15     1      1.E0
** MATRIX DIRICHLET B.CS. INPUT, IF ANY **
** TIME DEPENDENT SOURCE IN THE MATRIX **
** MATRIX MANUAL CONCENTRATION INITIALIZATION **
** COLLOID TRANSPORT VELOCITY SCALING FACTORS IN THE MATRIX **
```

DUALTEST.IN

```
** DO YOU HAVE HOMOGENEOUS PROPERTIES? (T/F) **
T
** TITLE OF THE TEST RUN **
'FLUX FIELD TEST'
** TIME STEP SIZE AND NUMBER **
3.1536E4      100
** # OF GRID BLOCKS IN X,Y,Z **
58      22      1
** DIMENSION OF GRID BLOCKS IN X,Y,Z **
100.      100.      100.
** POROSITY AND SATURATION OF THE FRACTURE **
0.01      0.6
** VELOCITIES IN X,Y,Z IN THE FRACTURE **
8.E-6      8.E-6      0.
** FRACTURE TEMPERATURE AND FLUID VISCOSITY **
298.15      1.E-3
** POROSITY AND SATURATION OF THE MATRIX **
0.3      .6
** VELOCITIES IN X,Y,Z IN THE MATRIX **
4.E-8      4.E-8      0.
** MATRIX TEMPERATURE AND FLUID VISCOSITY **
298.15      1.E-3
** SHAPE FACTOR, AND TRANSFER FUNCTION **
0.08      4.E-10
```

Appendix I: Listing of NUTS's Error Messages

Error Messages are categorized according to the files in which they arise.

BRAGFLO INPUT

Aborting Messages:

1. Number of grid blocks in x can not be #. Use at least "1" grid block. NUTS will be aborted.
2. Number of grid blocks in y can not be #. Use at least "1" grid block. NUTS will be aborted.
3. Number of grid blocks in z can not be #. Use at least "1" grid block. NUTS will be aborted.
4. Grid block length in x can not be # for block number. Use positive number > 0 NUTS will be aborted.
5. Grid block length in y can not be # for block number. Use positive number > 0 NUTS will be aborted.
6. Grid block length in z can not be # for block number. Use positive number > 0 NUTS will be aborted.
7. Viscosity of the brine can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
8. Grid block porosity can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
9. Grid block saturation can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
10. Darcian Velocity in x can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
11. Darcian Velocity in y can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
12. Darcian Velocity in z can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
13. Grid block brine pH can not be # for block number. Use positive number. NUTS will be aborted.
14. Flux in x can not be # for block number i,j,k. Numbers based on $v \leq 0.6$ m/s are appropriate. NUTS will be aborted.
15. Flux in y can not be # for block number i,j,k. Numbers based on $v \leq 0.6$ m/s are appropriate. NUTS will be aborted.
16. Flux in z can not be # for block number i,j,k. Numbers based on $v \leq 0.6$ m/s are appropriate. NUTS will be aborted.
17. Density of brine can not be # for block number i,j,k. Use a positive number. NUTS will be aborted.
18. Grid block brine viscosity as # is high. This value is not for a liquid. NUTS will be aborted.
19. Grid block saturation can not be # for block number i,j,k. Use numbers <= 1. NUTS will be aborted.
20. Grid block porosity can not be # for block number i,j,k. Use numbers <= 1. NUTS will be aborted.
21. This message will appear in the debug file:
SERIOUS ERROR IS DETECTED IN BRAGFLO BINARY FILE (BRAGFLO file name) NUTS RUN IS TERMINATED, PLEASE SEE THE ABOVE WRITTEN MESSAGES
22. This message will appear on the screen:
SERIOUS ERROR IS DETECTED IN BRAGFLO BINARY FILE(BINARY FILE NAME) NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS debug file) TO IDENTIFY THE PROBLEM.

Warning Messages:

1. Grid block PCO2 # is too high for block number i,j,k.
2. Grid block Brine Pressure # is too high for block number i,j,k.
3. Reference temperature as # is low. Use > 273.15 K.
4. Reference temperature as # is high. This may need thermal simulator.
5. Reference brine pressure as # is too high.
6. Temperature of the grid block is # for block number i,j,k. Use > 273.15 K.
7. Grid block dimension in x as # is high for grid block i,j,k. This may lead to high numerical dispersion.
8. Grid block dimension in y as # is high for grid block i,j,k. This may lead to high numerical dispersion.

9. Grid block dimension in z as # is high for grid block i,j,k. This may lead to high numerical dispersion.
10. Grid block temperature as # is high for grid block i,j,k. This may need thermal simulator.
11. Grid block brine pH as # is high for grid block i,j,k. Use practical value.
12. Grid block brine density as # is high for grid block i,j,k. Use practical value
13. This message will appear on the screen:
Undesirable value(s) is detected in the input file (BRAGFLO binary file name)
For further information, please check (NUTS debug file name).
14. This message will appear in NUTS debug file:
Undesirable value(s) is detected in the input file (BRAGFLO binary file name)
For more information, please check above written messages.

Normal Completion Message:

Upon completion of reading the binary input file successfully, this message will appear both on the screen and in the NUTS debug file.

NORMAL COMPLETION READING THE INPUT FILE FROM BRAGFLO
(BRAGFLO binary file name)

FLUX FIELD ASCII INPUT:

Aborting Messages

1. Number of grid blocks in x can not be #. Use at least "1" grid block. NUTS will be aborted.
2. Number of grid blocks in y can not be #. Use at least "1" grid block. NUTS will be aborted.
3. Number of grid blocks in z can not be #. Use at least "1" grid block. NUTS will be aborted.
4. Grid block length in x can not be # for block number. Use positive number > 0 NUTS will be aborted.
5. Grid block length in y can not be # for block number. Use positive number > 0 NUTS will be aborted
6. Grid block length in z can not be # for block number. Use positive number > 0 NUTS will be aborted.
7. Viscosity of the brine can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
8. Grid block porosity can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
9. Grid block saturation can not be # for block number i,j,k. Use positive number. NUTS will be aborted.
10. Darcian Velocity in x can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
11. Darcian Velocity in y can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
12. Darcian Velocity in z can not be # for block number i,j,k. Numbers <= 0.6 are appropriate. NUTS will be aborted.
13. Density of brine can not be # for block number i,j,k. Use a positive number. NUTS will be aborted.
14. Grid block brine viscosity as # is high. This value is not for a liquid. NUTS will be aborted.
15. Grid block saturation can not be # for block number i,j,k. Use numbers <= 1. NUTS will be aborted.
16. Grid block porosity can not be # for block number i,j,k. Use numbers <= 1. NUTS will be aborted.
17. Time step size can not be #. Use positive number. NUTS will be aborted.
18. Number of time steps can not be #. Use positive number. will be aborted.
19. This message will appear in the debug file:
SERIOUS ERROR IS DETECTED IN THE TEST FILE (file name) NUTS RUN IS TERMINATED,
PLEASE SEE THE ABOVE WRITTEN MESSAGES
20. This message will appear on the screen:
SERIOUS ERROR IS DETECTED IN THE TEST FILE (file name) NUTS RUN IS TERMINATED.
PLEASE CONSULT WITH (NUTS debug file) TO IDENTIFY THE PROBLEM.

Warning Messages:

1. Temperature of the grid block is # for block number i,j,k. Use > 273.15.
2. Grid block dimension in x as # is high for grid block i,j,k. This may lead to high numerical dispersion.
3. Grid block dimension in y as # is high for grid block i,j,k. This may lead to high numerical dispersion.
4. Grid block dimension in z as # is high for grid block i,j,k. This may lead to high numerical dispersion.
5. Grid block temperature as # is high for grid block i,j,k. This may need thermal simulator.
6. Grid block brine density as # is high for grid block i,j,k use practical value.
7. You have # character title. Title should not exceed 100 character.
8. This message will appear on the screen.
Undesirable value(s) is detected in the input file (input file name). For further information. Please check (debug file name).
9. This message will appear in the debug file.
Undesirable value(s) is detected in the input file (input file name). For more information, please check above written messages.

Normal Completion Message

Upon completion of reading the test input file successfully, this message will appear both on the screen and in the NUTS debug file.

Information Only

NORMAL COMPLETION READING THE INPUT FILE (INPUT FILE NAME)

NUTS ASCII Input

Aborting messages

1. Number of disposal sites can not be #. Use at least "1" site. NUTS will be aborted.
2. Number of sites # can not be > number of grid blocks #. They can be equal. NUTS will be aborted.
3. Number of material can not be #. Use at least "1" material. NUTS will be aborted.
4. # of material # can not be > # of grid blocks. They can be equal. NUTS will be aborted.
5. Number of the component in site # in the input can not be #. Use a number >= 1. NUTS will be aborted.
6. Number of elements can not be #. Use at least "1" element. NUTS will be aborted.
7. Number of element # can not be > number of nuclides #. They can be equal. NUTS will be aborted.
8. Number of NUCLIDE can not be #. Use at least "1" nuclide. NUTS will be aborted.
9. Solubility of element number # in the input can not be #. Use a positive number. NUTS will be aborted.
10. Molecular weight of component number in the input can not be #. Use a positive number. NUTS will be aborted.
11. Molecular weight of component number in the input can not be > 1. The conversion factors are based on Kg Mol and not gm Mol. NUTS will be aborted.
12. Contact handled inventory of component number # in the input can not be #. Use a positive number. NUTS will be aborted.
13. Remote handled inventory of component number # in the input can not be #. Use a positive number. NUTS will be aborted.
14. Half-life of component number # in the input can not be #. Use a positive number. NUTS will be aborted.
15. Brine pH can not be # for material number #. Use positive number. NUTS will be aborted.
16. Molecular diffusion of component number # in the input can not be #. Use a positive number. NUTS will be aborted.
17. Diffusion Velocity in x can not be # for block number i,j,k. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
18. Diffusion Velocity in y can not be # for block number i,j,k. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
19. Diffusion Velocity in z can not be for block number i,j,k. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
20. Rock density can not be # for material number #. Use a positive number. NUTS will be aborted.
21. Number of waste component can not be #. Use a positive number. NUTS will be aborted.
22. Number of waste component can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
23. Location of waste component in the input can not be #. Use a positive number. NUTS will be aborted.
24. Location of waste component can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
25. The # of the waste region in the input can not be #. Use a positive number. NUTS will be aborted.
26. The # of the waste region in the input can not be #. Use a number < # which is total # of sites. NUTS will be aborted.
27. the # of grid block in each site can not be #. Use a number < # which is total number of grid blocks. NUTS will be aborted.
28. the # of grid block in each site can not be #. Use a number >= #. NUTS will be aborted.
29. This message will appear in NUTS debug file
SERIOUS ERROR IS DETECTED IN THE INPUT FILE (NUTS INPUT FILE NAME). NUTS RUN IS TERMINATED. PLEASE SEE ABOVE WRITTEN MESSAGES.
30. This message will appear on the screen
SERIOUS ERROR IS DETECTED IN THE INPUT FILE (NUTS INPUT FILE NAME). NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS DEBUG FILE NAME) TO IDENTIFY THE PROBLEM.
31. Linear sorption coefficient of component number # in the material number # can not be #. Use a positive number. NUTS will be aborted.

32. Rock longitudinal dispersivity can not be # for material number #. Use a positive number. NUTS will be aborted.
33. Rock transverse dispersivity can not be # for material number #. Use a positive number. NUTS will be aborted.
34. Number of injected nuclide can not be #. Use a positive number. NUTS will be aborted.
35. Number of injected nuclide can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
36. Location of injected nuclide in the input can not be #. Use a positive number. NUTS will be aborted.
37. Location of injected nuclide can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
38. Total # of injection grids in the input can not be #. Use a positive number. NUTS will be aborted.
39. Total # of injection grids in the input can not be #. Use a number < # which is total # of grids. NUTS will be aborted.
40. Injection grid I-index in the input can not be #. Use a positive number. NUTS will be aborted.
41. Injection grid I-index in the input can not be #. Use a number < # which is total # of grids in I-dir. NUTS will be aborted.
42. Injection grid J-index in the input can not be #. Use a positive number. NUTS will be aborted.
43. Injection grid J-index in the input can not be #. Use a number < # which is total # of grids in J-dir. NUTS will be aborted.
44. Injection grid K-index in the input can not be #. Use a positive number. NUTS will be aborted.
45. Injection grid K-index in the input can not be #. Use a number < # which is total # of grids in K-dir. NUTS will be aborted.
46. Number of the grids with a source can not be #. Use a positive number. NUTS will be aborted.
47. Number of the grids with a source can not be #. Use a number < # which is total # of grids. NUTS will be aborted.
48. Source grid I-index in the input can not be #. Use a positive number. NUTS will be aborted.
49. Source grid I-index in the input can not be #. Use a number < # which is total # of grids in I-dir. NUTS will be aborted.
50. Source grid J-index in the input can not be #. Use a positive number. NUTS will be aborted.
51. Source grid J-index in the input can not be #. Use a number < # which is total # of grids in J-dir. NUTS will be aborted.
52. Source grid K-index in the input can not be #. Use a positive number. NUTS will be aborted.
53. Source grid K-index in the input can not be #. Use a number < # which is total # of grids in K-dir. NUTS will be aborted.
54. Source rate is zero in source block #. If you have a source, you should have a rate. NUTS will be aborted.

55. Source starting time can not be #. Use a positive number or 0. NUTS will be aborted.
56. Source starting time can not be # Which is > ending time #. NUTS will be aborted.
57. Source ending time can not be #. Use a positive number or 0. NUTS will be aborted.
58. Number nuclides having Dirichlet B.Cs can not be #. Use a positive number. NUTS will be aborted.
59. Number nuclides having Dirichlet B.Cs can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
60. Location of nuclides having Dirichlet B.Cs in the input can not be #. Use a positive number. NUTS will be aborted.
61. Location of nuclide having Dirich. B.Cs can not be #. 'Use a number < # which is total # of the nuclides. NUTS will be aborted.
62. Total # of grids having Dirich. B.Cs can not be #. Use a positive number. NUTS will be aborted.
63. Total # of grids having Dirich. B.Cs can not be #. Use a number < # which is total # of grids. NUTS will be aborted.
64. Dirichlet grid I-index in the input can not be #. Use a positive number. NUTS will be aborted.
65. Dirichlet grid I-index in the input can not be #. Use a number < # which is total # of grids in I-dir. NUTS will be aborted.
66. Dirichlet grid J-index in the input can not be #. Use a positive number. NUTS will be aborted.
67. Dirichlet grid J-index in the input can not be #. Use a number < # which is total # of grids in J-dir. NUTS will be aborted.
68. Dirichlet grid K-index in the input can not be #. Use a positive number. NUTS will be aborted.
69. Dirichlet grid K-index in the input can not be #. Use a number < # which is total # of grids in K-dir. NUTS will be aborted.
70. Injected conc. of component # in grid i,j,k can not be #. Please use a positive number. NUTS will be aborted.
71. Dirichlet B.C. conc. of component # in grid i,j,k can not be #. Please use a positive number. NUTS will be aborted.
72. Number nuclides initialized can not be #. Use a positive number. NUTS will be aborted.
73. Number of initialized nuclide can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
74. Location of initialized nuclides in the input can not be #. Use a positive number. NUTS will be aborted.
75. Location of initialized nuclide can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
76. Total # of initialized grids in the input can not be #. Use a positive number. NUTS will be aborted.
77. Total # of initialized grids can not be #. Use a number < # which is total # of grids. NUTS will be aborted.

78. Initialized grid I-index in the input can not be #. Use a positive number. NUTS will be aborted.
79. Initialized grid I-index in the input can not be #. Use a number < # which is total # of grids in I-dir. NUTS will be aborted.
80. Initialized grid J-index in the input can not be #. Use a positive number. NUTS will be aborted.
81. Initialized grid J-index in the input can not be #. Use a number < # which is total # of grids in J-dir. NUTS will be aborted.
82. Initialized grid K-index in the input can not be #. Use a positive number. NUTS will be aborted.
83. Initialized grid K-index in the input can not be #. Use a number < # which is total # of grids in K-dir. NUTS will be aborted.
84. Initialized conc. of component # in grid i,j,k can not be #. Please use a positive number. NUTS will be aborted.
85. Scaling factor for colloid transport in x-direction for grid block i,j,k can not be #. Use a number >= 1. NUTS will be aborted.
86. Scaling factor for colloid transport in y-direction for grid block i,j,k can not be #. Use a number >= 1. NUTS will be aborted.
87. Scaling factor for colloid transport in z-direction for grid block i,j,k can not be #. Use a number >= 1. NUTS will be aborted.
88. This message will appear in the debug file:

SERIOUS ERROR IS DETECTED IN THE INPUT FILE (NUTS INPUT FILE NAME) NUTS RUN IS TERMINATED. PLEASE SEE ABOVE WRITTEN MESSAGES
89. This message will appear on the screen:

SERIOUS ERROR IS DETECTED IN THE INPUT FILE (NUTS INPUT FILE NAME) NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS DEBUG FILE NAME) TO IDENTIFY THE PROBLEM.
90. Number of time dependent source nuclides can not be #. Use a positive number. NUTS will be aborted.
91. Number of time dependent source nuclides can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
92. Location of time dependent source nuclides can not be #. Use a positive number. NUTS will be aborted.
93. Location of time dependent source nuclides can not be #. Use a number < # which is total # of the nuclides. NUTS will be aborted.
94. Total # of time dependent source grids in the input can not be #. Use a positive number. NUTS will be aborted.
95. Total # of time dependent source grids can not be #. Use a number < # which is total # of grids. NUTS will be aborted.
96. Time dependent source grid I-index can not be #. Use a positive number. NUTS will be aborted.
97. Time dependent source grid I-index can not be #. Use a number < # which is total # of grids in I-dir. NUTS will be aborted.
98. Time dependent source grid J-index can not be #. Use a positive number. NUTS will be aborted.
99. Time dependent source grid J-index can not be #. Use a number < # which is total # of grids in J-dir. NUTS will be aborted.

100. Time dependent source grid K-index can not be #. Use a positive number. NUTS will be aborted.
101. Time dependent source grid K-index can not be #. Use a number < # which is total # of grids in K-dir. NUTS will be aborted.
102. Number of colloids used as # is > the total number of isotopes #. Please set NCOLLOID <= #. NUTS will be terminated.
103. Number of colloids used as # can not be 0 or negative, please use a number <= #. NUTS will be terminated.
104. Location of the colloid used as # is > the total number of isotopes #. Please set the location <= #. NUTS will be terminated.
105. Location of the colloid used as # can not be negative, please use a positive number <= #. NUTS will be terminated.

Warning Messages

1. Brine pH as # for material number # is high!! check it please.
2. Rock density as # for material number # is high. Please check.
3. This message will appear on the screen
Undesirable value(s) is detected in the input file (NUTS Input File Name). For further information. Please check (NUTS Debug File Name).
4. This message will appear in NUTS debug file
Undesirable value(s) is detected in the input file (NUTS Input File Name). For more information, please check above written messages.
5. Rock longitudinal dispersivity input as # for material number # is larger than the maximum hydrological distance #. Please check the value.
6. Rock transverse dispersivity input as # for material number # is larger than longitudinal dispersivity #. Please check the value.
7. Linear sorption coefficient of component number # in the material number # is very high as #. Use a positive number <=1.
8. Undesirable value(s) is detected in the input file (NUTS Input File Name) For further information. Please check (NUTS Debug File Name).
9. Undesirable value(s) is detected in the input file (NUTS Input File Name) For more information, please check above written messages
10. Only 1 for 1.st order or 2 for 2.nd order, 3 for 2.nd order with split operator, and 4 for central difference are allowed. The order of the method requested as # is not available in NUTS.
11. Only BETA2 of [0,1] interval is accepted, 0 is explicit solution, 1/2 is Crank Niclson, 1 is fully implicit. The value requested as # is not available in NUTS (if BETA2 > 1, NUTS will set the solution to fully implicit and if BETA2 < 1, NUTS will set the solution to fully explicit.
12. For component #, grid block # has row entry = # < #, the minimum value in the table of that component. The value is reset to the minimum value.
13. For component #, grid block # has row entry = # > #, the maximum value in the table of that component. The value is reset to the maximum value.
14. For component #, grid block # has column entry = # < #, the minimum value in the table of that component. The value is reset to the minimum value.
15. For component #, grid block # has column entry = # > #, the maximum value in the table of that component. The value is reset to the maximum value.

Normal Completion Message:

Upon completion of reading NUTS ASCII input file successfully, these messages will appear both on the screen and in the NUTS debug file.

1. NORMAL COMPLETION READING NUCLIDES PROPERTIES FROM (NUTS INPUT FILE NAME).

Information Only

2. NORMAL COMPLETION READING THE MATRIX PROPERTIES FROM (NUTS INPUT FILE NAME) (If the matrix is a continuum).
3. NORMAL COMPLETION READING THE FRACTURE PROPERTIES FROM (NUTS INPUT FILE NAME) (If the fracture is a continuum).

Operational Messages:

Aborting Messages

1. There is inconsistency in DT due to injection/production. The program is terminated.
2. Check your input for Langmuir coefficient. It may input as negative, or negative concentration has been detected. Check and rerun the program.
3. The matrix is singular.
4. Not enough memory allocated for band width. Parameter declaration of MBW is #, while the number required is #. Please set MBW >= # in NUT_PARAM.INC. and rerun
5. Subroutine Bin_read is only capable of reading single-porosity matrix porous media output at this time. Execution is terminated in Sub. CONVERT.
6. End of file is encountered, "execution terminated:" for further calculation, check BIN_READ and rerun (Time is written).
7. Zero tortuosity value is encountered. Division by zero will follow. The program is terminated in DSPRSN2.FOR. Check tortuosity and rerun.
8. %WARNING - ARRAY OUT OF BOUND
Not enough memory allocated for x dimension. Parameter declaration of MX is #, While the number of grid blocks is #. Please set MX>= # in NUT_PARAM.INC and rerun
9. %WARNING - ARRAY OUT OF BOUND
Not enough memory allocated for y dimension. Parameter declaration of MY is #, While the number of grid blocks is #. Please set MY>= # in NUT_PARAM.INC and rerun
10. %WARNING - ARRAY OUT OF BOUND
Not enough memory allocated for z dimension. Parameter declaration of MZ is #, While the number of grid blocks is #. Please set MZ>= # in NUT_PARAM.INC
11. %WARNING - ARRAY OUT OF BOUND
Not enough memory allocated for total number of grid blocks. Parameter declaration of NB is #, While the number of grid blocks is #. Please set NB>= # in NUT_PARAM.INC and rerun
12. End of file encountered, "execution terminated", check the time! if >= 10000 yrs, normal completion (print the time in second and years).
13. Material Map of time zero is not in BRAGFLO input. Execution is terminated in subroutine MAT_MAP.
14. Material Map of time zero is not in NUTS input. Execution is terminated in subroutine MAT_MAP.
15. Inconsistency in the # of materials. Execution is terminated in MAT_MAP.
16. Unable to read parameter from BRAGFLO input
Execution is stopped
----- Diagnosis -----
17. End of file is encountered in CDB_INITM_CONC_FLAG. Program is terminated
18. End of file is encountered in PHYSPROP. Program is terminated
19. End of file is encountered in ADS_INPUT. Program is terminated
20. End of file is encountered in MATDSP_INPUT. Program is terminated
21. End of file is encountered in FRCDSP_INPUT. Program is terminated
22. End of file is encountered in COMPMOL_DIFFUSION. Program is terminated
23. End of file is encountered in ROCK_DENSITY_INPUT. Program is terminated
24. End of file is encountered in WASTE_MATRIX_INPUT. Program is terminated
25. End of file is encountered in SITE_FLAG. Program is terminated
26. End of file is encountered in MATADS_FLAG. Program is terminated
27. End of file is encountered in FRCADS_FLAG. Program is terminated
28. End of file is encountered in KDTEMPDEP_FLAG. Program is terminated
29. End of file is encountered in MATDSP_FLAG. Program is terminated
30. End of file is encountered in FRCDSP_FLAG. Program is terminated
31. End of file is encountered in MATFRCDSP_FLAG. Program is terminated
32. End of file is encountered in MOLDIFTEMDEP_FLAG. Program is terminated
33. End of file is encountered in MATSRC_FLAG. Program is terminated
34. End of file is encountered in FRCPRINT_FLAG. Program is terminated
35. End of file is encountered in FRCSRC_FLAG. Program is terminated

36. End of file is encountered in MATPRINT_FLAG. Program is terminated
37. End of file is encountered in PRNTFREQ_FLAG. Program is terminated
38. End of file is encountered in EXTSRC_FLAG. Program is terminated
39. End of file is encountered in INTRUSION_TIME_SUB. Program is terminated
40. End of file is encountered in ZEROLIMIT. Program is terminated
41. End of file is encountered in INITF_CONC_MAN. Program is terminated
42. End of file is encountered in INITM_CONC_MAN. Program is terminated

43. End of file is encountered in CONC_INIT_INPUT. Program is terminated
44. End of file is encountered in SCALE_FACT_COLLOID_INPUT. Program is terminated
45. End of file is encountered in SOURCE_INPUT. Program is terminated
46. ***** WARNING *****
Zero value index is detected. Execution is terminated in IND_INTERPOLATE SUB.
47. ***** WARNING *****
Zero value index is detected. Execution is terminated in IND1D_INTERPOLATE SUB.
48. Division by zero. Element (element name)
49. Not enough memory allocated for the nuclides. Parameter declaration of NC is #, While the number of radionuclides is #. Please set NC>= # in NUT_PARAM.INC and rerun
50. No data available to read after # time steps of BRAGFLO. The program is terminated.
51. **WARNING !!! WARNING !!!WARNING !!! WARNING !!!**
Number of sites specified by NUTS input is different from the number given in BRAGFLO. Execution will be terminated.
52. Number of iteration for the implicit precipitate exceeded #
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! NUTS IS ABORTED !!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Please run BRAGFLO or the fluid flow code with refined time step and re-run NUTS

Warning Messages

1. **WARNING !!! WARNING !!!**
DT<=0 is detected DT will be set To #
2. **WARNING !!! WARNING !!!WARNING !!! WARNING !!!**
Number of sites specified by NUTS input is "1", different from number of waste regions ">1" specified by BRAGFLO. Number of waste regions will be collapsed to number of sites specified by NUTS input.
3. **!!! WARNING !!!**
For component # grid block # has row entry = # < #, the minimum value in the table of that component. The value is reset to the minimum value.
4. **!!! WARNING !!!**
For component # grid block # has row entry = # > #, the maximum value in the table of that component. The value is reset to the maximum value.
5. **!!! WARNING !!!**
For component # grid block # has column entry = # < #, the minimum value in the table of that component. The value is reset to the minimum value.
6. **!!! WARNING !!!**
For component # grid block # has column entry = # > #, the maximum value in the table of that component. The value is reset to the maximum value.
7. No sorption isotherm has been specified. No sorption will be considered.

Appendix J: An Example of NUTS's Debug Output File

NUTS_PA96 2.02 PROD PA96 05/23/96

05/24/96 11:45:54

N	NN	UU	UU	TTTTTT	SSSSSS	PPPPPP	AAAAA	9999	6666			
NN	NN	UU	UU	TT	SS	PP	PP	AA	AA	9	9	6
NNN	NN	UU	UU	TT	SS	PP	PP	AA	AA	9	9	6
NN	N	NN	UU	UU	TT	SSSSS	PPPPPP	AAAAAAA	99999	66666		
NN	NNN	UU	UU	TT	SS	PP	AA	AA	9	6	6	
NN	NN	UU	UU	TT	SS	PP	AA	AA	9	6	6	
NN	N	UUUUU	TT	SSSSSS	PP	AA	AA	9999	6666			

NUTS - Nuclide Transport System Model

NUTS_PA96 Version 2.02
PROD PA96 Built 05/23/96
Written by Ali Shinta
Sponsored by Ali Shinta

Run on 05/24/96 at 11:45:54
Run on ALPHA AXP BEATLE OpenVMS V6.1

Prepared for
Sandia National Laboratories
Albuquerque, New Mexico 87185-5800
for the United States Department of Energy
under Contract DE-AC04-76DP00789

Disclaimer

This computer program 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 any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed or represents that its use would not infringe privately owned rights. 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, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

FILE ASSIGNMENTS:

POSTBRAG_T CDB file:
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_TEST_CDPDC.IN;1
Written on 01/24/96 12:00:08
Is this a TEST RUN? (Y/N):
Y
I/O: (ASC, BIN, ASC-BIN):
ASC

NUTS Input data file:
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_CDPDC.IN;8
Written on 05/24/96 11:39:43
NUTS Output CDB file:
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_CDPDC.OUT
NUTS Diagnostic/Debug data file:
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_CDPDC.DBG
Calculate for LIQUID or GAS? (L/G):
L
BRAGFLO input file:

Read NUCLIDE related Material Properties from CDB or from NUTS input file (CDB/N:
N
SOLUBILITY AND INVENTORY CDB:
UNDISTRUBED SCENARIO CDB:

!!!!!!!!!!!!!! DIGNOSTIC MESSAGES !!!!!!!!!!!!!!!
!!!!!!!!!!!!!! TEST FILE INPUT !!!!!!!!!!!!!!!

NORMAL COMPLETION READING THE INPUT FILE
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_TEST_CDPDC.IN;1

***** START SUBROUTINE ONED3DINDEX *****

***** END SUBROUTINE ONED3DINDEX *****
<><> START READING FROM READ_RAD SUB. <><>

!!!!!!!!!!!!!! DIGNOSTIC MESSAGES !!!!!!!!!!!!!!!
!!!!!!!!!!!!!! NUTS INPUT !!!!!!!!!!!!!!!
!!!!!!!!!!!!!! GENERAL PROPERTIES !!!!!!!!!!!!!!!

NORMAL COMPLETION READING NUCLIDES PROPERTIES FROM
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_CDPDC.IN;8

NORMAL COMPLETION READING THE MATRIX PROPERTIES FROM
WP\$TESTROOT:[NUT.TEST.TEST4]NUT_CDPDC.IN;8

<><> FINISH READING FROM READ_RAD. <><>

<><> START INITIALIZATION IN INITIALIZATION SUB. <><>
<><> END INITIALIZATION IN INITIALIZATION SUB. <><>

***** START SUBROUTINE STOTALMAS *****

***** END SUBROUTINE STOTALMAS *****

***** START SUBROUTINE NTOTALMAS *****

***** END SUBROUTINE NTOTALMAS *****

***** START SUBROUTINE CURIES_VALUES *****


```
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
```

```
END END END END END END END END END END
BYE BYE BYE BYE BYE BYE BYE BYE BYE BYE
||||||||||||||||||||||||||||||||||||||||
*****
```

```
***** START SUBROUTINE STOTALMAS *****
***** END SUBROUTINE STOTALMAS *****
***** START SUBROUTINE NTOTALMAS *****
***** END SUBROUTINE NTOTALMAS *****
***** START SUBROUTINE CURIES_VALUES *****
***** END SUBROUTINE CURIES_VALUES *****
***** START SUBROUTINE CNTOTALMAS *****
***** END SUBROUTINE CNTOTALMAS *****
```

Appendix K: FORTRAN Program to Convert NUTS's Binary Output to a Readable Format

NUTS_BIN_READ Procedure:

A FORTRAN listing for the procedures that can be used to read the binary files from NUclide Transport System (NUTS) is provided in this appendix. A discription of the variables and a descriptive line before the main functions of the procedure are also included to increase the readability of the subroutines.

```
C
C***** START SUBROUTINE NUTS_BIN_READ *****
C
C          SUBROUTINE NUTS_BIN_READ
C
C-----
C
C          NUTS_BIN_READ
C
C          Purpose:
C          -----
C          This subroutine reads NUTS output binary file
C
C          Author:      Ali A. Shinta
C          -----
C
C          Call:  BRE33D, BRE23D
C
C          Arguments:
C          -----
C  DXGRID          Grid block length in x-direction
C  DYGRID          Grid block length in y-direction
C  DZGRID          Grid block length in z-direction
C  ZTIME          Total time
C  IFLAGTIME       Flag = 0 for time = 0 and 1 otherwise
C  NO_TIMESTEP    Time step no.
C  SDATE          Date of the run
C  STIME          Time of the run
C  NPNAME         Program name
C  NVERSION       Program version no.
C  NREVDATA       Revision date
C  NCPUNAME       Name of the machine on which the run is conducted
C  ANSWERTEST    Flag to tell if the input is from BRAGFLO or stand alone file
C  FINFILETYPE    Output files type
C  COMBTITLE      A combination of NUTS title and BRAGFLO or TEST run title
C  INTITLE        BRAGFLO or TEST run title
C  NUTS_TITLE     NUTS title
C  IDIMENSION     No. of dimensions
C  RADINPUT       Nuts input file name
C  FILE_NAME      BRAGFLO or test run input file name
C  RADOUTDBG     NUTS ASCII debug output file name
C  RADOUTPUT     NUTS output file name
C  RADOUTBIN     Output Binary name
C  RADOUTASC     Output ASCII file name
C  MEDIUM        Type of the porous medium (fracture, matrix, etc..)
C  FRACTUR       Logical flag to identify fracture continuum
C  MATRIX        Logical flag to identify matrix continuum
C  SINGLE_POROSITY Logical flag to identify single-porosity system
C  DUAL_POROSITY Logical flag to identify dual-porosity
C  DUAL_PERMEABILITY Logical flag to identify dual-permeability system
C  NBLOCK        Total number of grid blocks
C  NX OR IMAX    No. of grid blocks in x-direction
C  NY OR JMAX    No. of grid blocks in y-direction
C  NZ OR KMAX    No. of grid blocks in z-direction
```

C NOCONTINUM No. of continuum
C NO_PHASES Number of phases
C X Grid block x-coordinate
C Y Grid block x-coordinate
C Z Grid block x-coordinate
C AXGRID Interblock area in x-direction (I,I-1)
C AYGRID Interblock area in y-direction (J,J-1)
C AZGRID Interblock area in z-direction (K,K-1)
C VRGRID Grid volume
C PHREQ Flag to identify pH dependency (Y/N)
C PHGRID Grid block ground water pH
C NSITES No. of radioactive sites
C NUCLIDE Total number of radionuclide
C SITE_NAME Name of the disposal site
C COMPONENT_NAME Component name
C NAME Component name
C Daughter_NAME Component daughter name
C ELEMENT_NAME Element name
C GROUP_NAME Group under which the isotope is classified, i.e: U for U235,238,.....etc.
C PARENT_NAME Component parent name
C NCOMPONENT No. of the components in each site
C WASTEVOL Site's waste vol in M3.
C COMPINT Initial inventory of the component in Kg.
C XMOLWT Molecular weight
C XLAMDA Decay constant of the component in 1/s.
C RAD Character to identify the component radioactivity (radioactive and not_radioactive)
C NOELEMENT No. of the input elements
C ELSOLB_LIMIT_COR Element solubility limit (Kg/M3) of block I
C ELTEMP_SOLB Logical variable to identify if the element solubility is temperature dependent
C ELEMNT_SOLB_LIMIT Element solubility limit
C C0-5 Solubility parameters for temperature dependency correlation
C PHASETYPE Phase type to be transported (liquid or gas)
C EQCI Intercept of the equilibrium line for gas solubility in the ground water
C EQCS Slope of the equilibrium line for gas solubility in the ground water
C IWASTE Waste matrix, 1=waste node, 0=no-waste node.
C ADSTYEM or F Flag to identify whether sorption is considered (Y/N)
C MAT_ISOOTHERM Type of the adsorp. isotherm in the matrix
C L/l=Linear,F/f=Freundlich,LA/La/la=Langmuir
C FRAC_ISOOTHERM Type of the adsorp. isotherm in the fracture
C L/l=Linear,F/f=Freundlich,LA/La/la=Langmuir
C MSORPTION Flag to identify whether the component is sorbable in the matrix (ADSORP/NON_ADSORP)
C MADSTEMPDEP Logical flag to identify whether the component sorption is temp. dependent
C FSORPTION Flag to identify whether the component is sorbable in the matrix (ADSORP/NON_ADSORP)
C FADSTEMPDEP Logical flag to identify whether the component sorption is temp. dependent

C XLMGRID Linear sorption coefficient in the matrix ({M3 fluid}/{Kg solid})
C XLFGRID Linear sorption coefficient in the fracture ({M3 fluid}/{Kg solid})
C XFDCMGRID Freundlich coefficient in the matrix (dimensionless).
C XFDCMGRID Freundlich distribution coefficient in the matrix ({M3 fluid}/{Kg solid}).
C XLCMGRID Langmuir coefficient in the matrix (dimensionless).
C XLDCMGRID Langmuir distribution coefficient in the matrix
C XFCFGRID Freundlich coefficient in the fracture (dimensionless).
C XFDCFGRID Freundlich distribution coefficient in the fracture ({M3 fluid}/{Kg solid}).
C XLCFGRID Langmuir coefficient in the fracture (dimensionless).
C XLDCFGRID Langmuir distribution coefficient in the fracture
C REFTEMPKD Reference temperature of Kd measurment
C ADSEXPCEFF Coefficient to adjust sorption at new temperature

C RHOGRID Grid block rock density (kg/m3)
C MDISPREQ Flag to identify whether there is dispersion in the matrix (Y/N)
C FDISPREQ Flag to identify whether there is dispersion in the matrix (Y/N)
C MFDISPREQ Flag to identify whether there is dispersion between matrix and fracture (Y/N)
C ALLMGRID Longitudinal matrix dispersivity
C ALLFGRID Longitudinal fracture dispersivity
C ALTMGRID Transverse matrix dispersivity
C ALTFGRID Transverse fracture dispersivity
C TORMGRID Matrix tortosity
C TOREGRID Matrix tortosity
C DMOL Molecular diffusion at reference temperature (M2/s.)
C DMOLTEMDEP Logical flag to identify whether the molecular diffusion is temperature dependent (.T./F.)
C TREF Reference temperature (K)
C VISREF Viscosity at reference temperature TREF (Pa/s)
C FSTATUSINJ Flag to identify whether there is injection in the fracture (Y/N)
C MSTATUSINJ Flag to identify whether there is injection in the matrix (Y/N)
C STOCKMAN Logical flag to identify whether NUTS interfaces with actinide source submodel
C TIMEMSTRT Starting time of injection/production in the matrix (s)
C TIMEMEND Ending time of injection/production in the matrix (s)
C TIMEFSTRT Starting time of injection/production in the fracture (s)
C TIMEFEND Ending time of injection/production in the fracture (s)
C CONCINJMGRID Concentration of the injected component j in grid i of the matrix (kg/m3)
C CONCINJFGRID Concentration of the injected component j in grid i of the fracture (kg/m3)
C FDIRICHLET Flag to identify whether dirichlet B.C. is available in the fracture (T/F)
C MDIRICHLET Flag to identify whether dirichlet B.C. is available in the matrix (T/F)
C CONCDIRMGRID Specified concentration for D.B.C. in the matrix block (kg/m3)
C CONCDIRFGRID Specified concentration for D.B.C. in the matrix block (kg/m3)
C IPRFRQA Frequency of print in ASCII file (ex:5=each 5 timesteps)
C IPRFRQB Frequency of print in BINARY file
C IPRNTFA Flag to print the fracture variable in ASCII file
C IPRNTFB Flag to print the fracture variable in BINARY file
C IPRNTMA Flag to print the matrix variable in ASCII file
C IPRNTMB Flag to print the matrix variable in BINARY file
C IPRNTMC Flag to print the matrix variable in CDB file
C VAR Printed variable name
C DEBUG Logical flag to generate debug file and variables
C SWMOLDGRID Matrix saturation at time level n
C PORMOLDGRID Matrix porosity at time level n
C SWFOLDGRID Fracture saturation at time level n
C PORFOLDGRID Fracture porosity at time level n
C BRVOLGRID Brine volume in each matrix grid (m3)
C SUMM_DISGRID Matrix total mass per grid block of the dissolved isotopes
C SUMF_DISGRID Fracture total mass per grid block of the dissolved isotopes
C SUMM_PRECIPGRID Matrix total mass per grid block of the precipitated isotopes
C SUMF_PRECIPGRID Fracture total mass per grid block of the precipitated isotopes
C SUMM_TOTALGRID Matrix total mass per grid block of the dissolved, precipitated, and sorbed isotopes
C SUMF_TOTALGRID Fracture total mass per grid block of the dissolved, precipitated, and sorbed isotopes
C CSUMM_DISGRID Matrix total curies per grid block of the dissolved isotopes
C CSUMF_DISGRID Fracture total curies per grid block of the dissolved isotopes
C CSUMM_PRECIPGRID Matrix total curies per grid block of the precipitated isotopes
C CSUMF_PRECIPGRID Fracture total curies per grid block of the precipitated isotopes
C CSUMM_TOTALGRID Matrix total curies per grid block of the dissolved, precipitated, and sorbed isotopes
C CSUMF_TOTALGRID Fracture total curies per grid block of the dissolved, precipitated, and sorbed isotopes
C CM Concentration of the isotope in the matrix at time level n+1
C CMOLD Concentration of the isotope in the matrix at time level n
C CF Concentration of the isotope in the fracture at time level n+1
C CFOLD Concentration of the isotope in the fracture at time level n
C BLOCF_DIS_MASSGRID Fracture mass of dissolved isotope per grid volume Kg/(M3 rock).

C BLOCM_DIS_MASSGRID Matrix mass of dissolved isotope per grid volume in Kg/(M3 rock) .
C BLOCF_PRC_MASSGRID Fracture mass of precipitated isotope per grid volume Kg/(M3 rock) .
C BLOCM_PRC_MASSGRID Matrix mass of precipitated isotope per grid volume Kg/(M3 rock) .
C ADPRCONM_TOTAL_MASSGRID Total mass resulted from dissolved, precipitated and sorbed isotopes in the matrix
C ADPRCONF_TOTAL_MASSGRID Total mass resulted from dissolved, precipitated and sorbed isotopes in the fracture
C VOLM_CONC_CURIESGRID Equivalent curies of the volumetric concentration in the matrix
C VOLF_CONC_CURIESGRID Equivalent curies of the volumetric concentration in the fracture
C DISM_MASS_CURIESGRID Curies of the dissolved mass per grid block in the matrix
C DISF_MASS_CURIESGRID Curies of the dissolved mass per grid block in the fracture
C PRCIPM_MASS_CURIESGRID Curies of the precipitated mass per grid block in the matrix
C PRCIPF_MASS_CURIESGRID Curies of the precipitated mass per grid block in the matrix
C TOTALM_MASS_CURIESGRID Curies of the total mass (dissolved + sorbed + Precipitate) per grid block in the matrix
C TOTALF_MASS_CURIESGRID Curies of the total mass (dissolved + sorbed + Precipitate) per grid block in the fracture
C FLUXJM1MGRID Fluxes crossing grid block lower interface in y-direction in the matrix (kg/s)
C FLUXJF1MGRID Fluxes crossing grid block lower interface in y-direction in the fracture (kg/s)
C FLUXIM1MGRID Fluxes crossing grid block left interface in x-direction in the matrix (kg/s)
C FLUXIF1MGRID Fluxes crossing grid block left interface in x-direction in the fracture (kg/s)
C CONDMMASSGRID Gas mass that dissolve in the grid block brine of the matrix (kg)
C CONDFMASSGRID Gas mass that dissolve in the grid block brine of the matrix (kg)
C QWXF_GRID Water interblock rate in x-direction in the fracture (m3/s)
C QWYF_GRID Water interblock rate in y-direction in the fracture (m3/s)
C QWZF_GRID Water interblock rate in z-direction in the fracture (m3/s)
C QWXM_GRID Water interblock rate in x-direction in the matrix (m3/s)
C QWYM_GRID Water interblock rate in y-direction in the matrix (m3/s)
C QWZM_GRID Water interblock rate in z-direction in the matrix (m3/s)
C QPRMGRID Matrix production rate (M3 /s)
C QINMGRID Matrix injection rate (M3 /s)
C BLOCKMBMAX Maximum residual among matrix blocks
C CMBIM Isotope material balance error for the time step in the matrix
C CMBTM Cumulative isotope material balance error in the matrix
C SUMRTM Cumulative residuals in the matrix
C SUMQTM Cumulative sources in the matrix
C BLOCKMBFMAX Maximum residual among fracture blocks
C CMBIF Isotope material balance error for the time step in the fracture
C CMBTF Cumulative isotope material balance error in the fracture
C SUMRTF Cumulative residuals in the fracture
C SUMQTF Cumulative sources in the fracture
C JBIN NUTS binary output file unit number

C*****
C*****

C-----
C

```
INCLUDE 'PARAMBR.INC'  
INCLUDE 'BINRHEAD.INC'  
COMMON/D3SIZE/NX,NY,NZ  
INTEGER NX,NY,NZ,I,J,KK
```

C
C----- READ TIME INFORMATION -----
C

```
      READ(JBIN) ZTIME, IFLAGTIME, NO_TIMESTEP
C
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! START READING INITIAL VARIABLES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C
      IF(IFLAGTIME .EQ. 0) THEN
C
C----- READ PROGRAM HEADINGS (QA INFORMATION) -----
C
      READ (JBIN) SDATE
      READ (JBIN) STIME
      READ (JBIN) NPNAME
      READ (JBIN) NVERSION
      READ (JBIN) NREVDATA
      READ (JBIN) NCPUNAME
C
C----- READ THE RUN TITLES AND FILES' TYPE -----
C
      READ(JBIN) ANSWERTEST, FINFILETYPE
      READ(JBIN) COMBTITLE, INTITLE, NUTS_TITLE
C
C----- READ THE FILES' NAMES -----
C
      READ(JBIN) RADINPUT, FILE_NAME, RADOUTPUT
      IF(FINFILETYPE .EQ. 'ASC' .OR. FINFILETYPE .EQ. 'ASC-BIN'
1 .OR. FINFILETYPE .EQ. 'ASC-CDB' .OR. FINFILETYPE .EQ.
2 'ASC-BIN-CDB' .OR. DEBUG)
3 READ(JBIN) RADOUTASC
      IF(FINFILETYPE .EQ. 'BIN' .OR. FINFILETYPE .EQ. 'ASC-BIN'
1 .OR. FINFILETYPE .EQ. 'BIN-CDB' .OR. FINFILETYPE .EQ.
2 'ASC-BIN-CDB')
3 READ(JBIN) RADOUTBIN
      IF(FINFILETYPE .EQ. 'CDB' .OR. FINFILETYPE .EQ. 'ASC-CDB'
1 .OR. FINFILETYPE .EQ. 'BIN-CDB' .OR. FINFILETYPE .EQ.
2 'ASC-BIN-CDB')
3 READ(JBIN) RADOUTCDB
C
C----- READ MEDIA TYPE AND LOGICAL CONTROLS -----
C
      READ(JBIN) MEDIUM, FRACTURE, MATRIX, SINGLE_POROSITY,
      & DUAL_POROSITY, DUAL_PERMEABILITY
C
C----- READ THE TOTAL NUMBER OF GRID BLOCKS -----
C
      READ(JBIN) NBLOCK
C
C----- READ DIMENSIONS OUTPUT -----
C
      READ(JBIN) IDIMENSION, NX, NY, NZ, DIRECTION
      READ(JBIN) NOCONTINUUM, NO_PHASES
C
C----- Grid Grid Block Length (x-direction)
C
      CALL BRE33D(DXGRID,JBIN)
C
C----- Grid Grid Block Width (y-direction)
C
      CALL BRE33D(DYGRID,JBIN)
C
C----- Grid Grid Block Hight (z-direction)
C
      CALL BRE33D(DZGRID,JBIN)
C
C----- READ Grid Block X,Y,Z COORDINATE
C
      CALL BRE33D(X,JBIN)
      CALL BRE33D(Y,JBIN)
      CALL BRE33D(Z,JBIN)
```

```
C
C----- Read Grid Block Volume -----
C
      CALL BRE33D(VRGRID,JBIN)
C
C----- Read Interface Area in x-direction -----
C
      CALL BRE33D(AXGRID,JBIN)
C
C----- Read Interface Area in y-direction -----
C
      CALL BRE33D(AYGRID,JBIN)
C
C----- Read Interface Area in z-direction -----
C
      CALL BRE33D(AZGRID,JBIN)
C
C----- Read Ground Water pH -----
C
      READ(JBIN) PHREQ
      IF(PHREQ .EQ. 'Y' .OR. PHREQ .EQ. 'y') CALL BRE33D(PHGRID,JBIN)
C
C----- WASTE INFORMATION -----
C
      READ(JBIN) NUCLIDE, NSITES
C
C----- SITE INFORMATION -----
C
      READ(JBIN) (SITE_NAME(I), NCOMPONENT(I), WASTEVOL(I),
& I=1, NSITES)
      READ(JBIN) (NAME(I), DAUGHTER_NAME(I), PARENT_NAME(I),
& GROUP_NAME(I), I=1, NUCLIDE)
C
      READ(JBIN) (XMOLWT(I), RAD(I), XLAMDA(I), COMPINT(I), I=1, NUCLIDE)
C
C----- SOLUBILITY DATA -----
C
      READ(JBIN) NOELEMENT
      READ(JBIN) (ELTEMP_SOLB(I), I = 1, NOELEMENT)
      DO I = 1, NOELEMENT
        IF(.NOT. ELTEMP_SOLB(I)) THEN
C
C<><><><><> ELEMENT NAME, TEMP. DEPENDENCY, SOLUBILITY LIMIT
C
          READ(JBIN) ELEMENT_NAME(I), ELEMNT_SOLB_LIMIT(I)
          ELSE
C
C<><><><><> ELEMENT NAME, TEMP. DEPENDENCY, CORRELATION PARAMETERS
C
          READ(JBIN) ELEMENT_NAME(I), C0(I), C1(I), C2(I),
& C3(I), C4(I), C5(I)
          ENDIF
          END DO
C
C----- GAS EQUILIBRIUM LINE CONSTANTS -----
C
      IF(PHASETYPE .EQ. 'G')
& READ(JBIN) (EQCI(I), EQCS(I), I = 1, NUCLIDE)
C
C----- WASTE MATRIX -----
C
      READ(JBIN) ((IWASTE(I, J), I=1, NBLOCK), J=1, NUCLIDE)
C
C----- SORPTION INPUT INFORMATION -----
C
C***** MATRIX SORPTION
C
      IF(MATRIX) THEN
      READ(JBIN) ADSTYPEM
      IF(ADSTYPEM .EQ. 'N' .OR. ADSTYPEM .EQ. 'n') GO TO 100
```

```
      READ(JBIN) MAT_ISOOTHERM
      READ(JBIN) (NAME(J),MSORPTION(J),MADSTEMPDEP(J),J=1,NUCLIDE)
C
C----- LINEAR DISTRIBUTION COEFFICIENTS -----
C
      IF(MAT_ISOOTHERM.EQ.'L'.OR.MAT_ISOOTHERM.EQ.'l') THEN
      DO 10 J = 1, NUCLIDE
      CALL BRE23D(XLMGRID,JBIN,J)
10    CONTINUE
      ENDIF
C
C----- FREUNDLICH DISTRIBUTION COEFFICIENTS -----
C
      IF(MAT_ISOOTHERM.EQ.'F'.OR.MAT_ISOOTHERM.EQ.'f') THEN
      DO 20 J = 1, NUCLIDE
      CALL BRE23D(XFDCMGRID,JBIN,J)
20    CONTINUE
      DO 30 J = 1, NUCLIDE
      CALL BRE23D(XFCMGRID,JBIN,J)
30    CONTINUE
      ENDIF
C
C----- LANGMUIR DISTRIBUTION COEFFICIENTS -----
C
      IF(MAT_ISOOTHERM.EQ.'LA'.OR.MAT_ISOOTHERM.EQ.'La'
&      .OR.MAT_ISOOTHERM.EQ.'la') THEN
      DO 40 J = 1, NUCLIDE
      CALL BRE23D(XLDCMGRID,JBIN,J)
40    CONTINUE
      DO 50 J = 1, NUCLIDE
      CALL BRE23D(XLCMGRID,JBIN,J)
50    CONTINUE
      ENDIF
100   CONTINUE
      ENDIF
C
C***** FRACTURE SORPTION
C
      IF(FRACTURE) THEN
      READ(JBIN) ADSTYPEF
      IF(ADSTYPEF.EQ.'N'.OR.ADSTYPEF.EQ.'n') GO TO 200
      READ(JBIN) FRAC_ISOOTHERM
      READ(JBIN) (NAME(J),FSORPTION(J),FADSTEMPDEP(J),J=1,NUCLIDE)
C
C----- LINEAR DISTRIBUTION COEFFICIENTS -----
C
      IF(FRAC_ISOOTHERM.EQ.'L'.OR.FRAC_ISOOTHERM.EQ.'l') THEN
      DO 110 J = 1, NUCLIDE
      CALL BRE23D(XLFGRID,JBIN,J)
110   CONTINUE
      ENDIF
C
C----- FREUNDLICH DISTRIBUTION COEFFICIENTS -----
C
      IF(FRAC_ISOOTHERM.EQ.'F'.OR.FRAC_ISOOTHERM.EQ.'f') THEN
      DO 120 J = 1, NUCLIDE
      CALL BRE23D(XFDCFGRID,JBIN,J)
120   CONTINUE
      DO 130 J = 1, NUCLIDE
      CALL BRE23D(XFCFGRID,JBIN,J)
130   CONTINUE
      ENDIF
C
C----- LANGMUIR DISTRIBUTION COEFFICIENTS -----
C
      IF(FRAC_ISOOTHERM.EQ.'LA'.OR.FRAC_ISOOTHERM.EQ.'La'
&      .OR.FRAC_ISOOTHERM.EQ.'la') THEN
      DO 140 J = 1, NUCLIDE
      CALL BRE23D(XLDCFGRID,JBIN,J)
140   CONTINUE
```

```
      DO 150 J = 1, NUCLIDE
      CALL BRE23D(XLCFGRID,JBIN,J)
150    CONTINUE
      ENDIF
200    CONTINUE
      ENDIF
C
C----- REFERENCE TEMP. AND SORPTION EXPONENT -----
C
      IF((ADSTYPEF .EQ. 'Y' .OR. ADSTYPEF .EQ. 'y') .OR.
& (ADSTYPEM .EQ. 'Y' .OR. ADSTYPEM .EQ. 'Y')) THEN
      DO I = 1, NUCLIDE
      IF(FADSTEMPDEP(I) .OR. MADSTEMPDEP(I))
& READ(JBIN)NAME(J),REFTEMPKD(J),ADSEXP_COEFF(J)
      END DO
      ENDIF
C
C----- ROCK DENSITY OUTPUT -----
C
      IF(ADSTYPEM .EQ. 'N' .OR. ADSTYPEM .EQ. 'n' .AND.
& ADSTYPEF .EQ. 'N' .OR. ADSTYPEF .EQ. 'n') GO TO 300
      CALL BRE33D(RHOGRID,JBIN)
C
300    CONTINUE
C
C----- DISPERSION -----
C
      IF(SINGLE_POROSITY .AND. FRACTURE) READ(JBIN) FDISPREQ
      IF(SINGLE_POROSITY .AND. MATRIX) READ(JBIN) MDISPREQ
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY)
& READ(JBIN) MDISPREQ,MFDISPREQ,FDISPREQ
C
C***** MATRIX
C
      IF(MATRIX) THEN
      IF(MDISPREQ .EQ. 'N' .OR. MDISPREQ .EQ. 'n') GO TO 400
C
C----- MATRIX LONGITUDINAL DISPERSIVITY -----
C
      CALL BRE33D(ALLMGRID,JBIN)
C
C----- ROCK TRANSVERSE DISPERSIVITY -----
C
      CALL BRE33D(ALTMGRID,JBIN)
C
C----- ROCK TORTOUSITY DATA -----
C
      CALL BRE33D(TORMGRID,JBIN)
C
400    CONTINUE
      ENDIF
C
C***** FRACTURE
C
      IF(FRACTURE) THEN
      IF(FDISPREQ .EQ. 'N' .OR. FDISPREQ .EQ. 'n') GO TO 500
C
C----- FRACTURE LONGITUDINAL DISPERSIVITY -----
C
      CALL BRE33D(ALLFGRID,JBIN)
C
C----- FRACTURE TRANSVERSE DISPERSIVITY -----
C
      CALL BRE33D(ALTFFGRID,JBIN)
C
C----- FRACTURE TORTOUSITY DATA -----
C
      CALL BRE33D(TORFGRID,JBIN)
500    CONTINUE
```

```

                                ENDIF
C
C-----MOLECULAR DIFFUSION DATA -----
C
      IF((MDISPREQ .EQ. 'Y' .OR. MDISPREQ .EQ. 'y') .OR.
& (FDISPREQ .EQ. 'Y' .OR. MDISPREQ .EQ. 'y')) THEN
      READ(JBIN) (DMOLTEMDEP(I), I = 1, NUCLIDE)
      DO I = 1, NUCLIDE
      IF(.NOT. DMOLTEMDEP(I)) THEN
C
C<><><><><> MOLECULAR DIFFUSION,TEMP. DEPENDENCY
C
      READ(JBIN) DMOL(I)
      ENDIF
      IF(DMOLTEMDEP(I)) THEN
C
C<><><><><> MOLECULAR DIFFUSION,TEMP. DEPENDENCY,REF. TEMP & VISCOSITY
C
      READ(JBIN) DMOL(I),TREF(I),VISREF(I)
      ENDIF
      END DO
      ENDIF
C
C----- INJECTION HISTORY -----
C
      IF(SINGLE_POROSITY .AND. FRACTURE)
& READ(JBIN) FSTATUSINJ,STOCKMAN
C
      IF(SINGLE_POROSITY .AND. MATRIX)
& READ(JBIN) MSTATUSINJ,STOCKMAN
C
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY)
& READ(JBIN) MSTATUSINJ,FSTATUSINJ,STOCKMAN
C
      IF(STOCKMAN) GO TO 901
C
C***** MATRIX
C
      IF(MATRIX) THEN
      IF(MSTATUSINJ .EQ. 'N' .OR. MSTATUSINJ .EQ. 'n') GO TO 700
      READ(JBIN) TIMEMSTRT, TIMEFEND
      DO 600 J = 1,NUCLIDE
      CALL BRE23D(CONCINJMGRID,JBIN,J)
600    CONTINUE
700    CONTINUE
      ENDIF
C
C***** FRACTURE
C
      IF(FRACTURE) THEN
      IF(FSTATUSINJ .EQ. 'N' .OR. FSTATUSINJ .EQ. 'n') GO TO 900
      READ(JBIN) TIMEFSTRT, TIMEFEND
      DO 800 J = 1,NUCLIDE
      CALL BRE23D(CONCINJFGRID,JBIN,J)
800    CONTINUE
900    CONTINUE
      ENDIF
901  CONTINUE
C
C----- SPECIFY DIRICHLET B.C. IF ANY -----
C
      IF(SINGLE_POROSITY .AND. FRACTURE) READ(JBIN) FDIRICHLET
      IF(SINGLE_POROSITY .AND. MATRIX) READ(JBIN) MDIRICHLET
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY)
& READ(JBIN) MDIRICHLET,FDIRICHLET
C
C***** MATRIX
C
      IF(MATRIX) THEN
      IF(MDIRICHLET) THEN
```

```
          DO 902 J = 1,NUCLIDE
          CALL BRE23D(CONCDIRMGRID,JBIN,J)
902      CONTINUE
          ENDIF
      C
      C***** FRACTURE
      C
          IF(FRACTURE) THEN
          IF(FDIRICHLET) THEN
          DO 904 J = 1,NUCLIDE
          CALL BRE23D(CONCDIRFGRID,JBIN,J)
904      CONTINUE
          ENDIF
      C----- READ PRINT FLAGS OF THE BINARY FILE -----
      C----- SPECIFY NUMBER OF OUTPUT ARRAYS AND THEIR TITLES -----
      C
          IF(FINFILETYPE .EQ. 'BIN' .OR. FINFILETYPE .EQ. 'ASC-BIN'
          1 .OR. FINFILETYPE .EQ. 'BIN-CDB' .OR. FINFILETYPE .EQ.
          2 'ASC-BIN-CDB'.OR. DEBUG) THEN
      C
      C***** MATRIX
      C
          IF(MATRIX) THEN
          READ(JBIN) (IPRNTMB(I), I = 1, NMVARB)
          DO 910 I = 1, NVARTIT/2
          IF(IPRNTMB(I) .GT. 0) READ(JBIN) VAR(I)
910      CONTINUE
          ENDIF
      C
      C***** FRACTURE
      C
          IF (FRACTURE) THEN
          READ(JBIN) (IPRNTFB(I), I = 1, NFVARB)
          KK = 0
          DO 920 I = NVARTIT/2+1, NVARTIT
          KK = KK + 1
          IF(IPRNTFB(KK) .GT. 0) READ(JBIN) VAR(I)
920      CONTINUE
          ENDIF
      C
      C***** READ PRINT FLAGS OF ASCII FILES
      C
          IF(FINFILETYPE .EQ. 'ASC' .OR. FINFILETYPE .EQ. 'ASC-BIN'
          1 .OR. FINFILETYPE .EQ. 'ASC-CDB' .OR. FINFILETYPE .EQ.
          2 'ASC-BIN-CDB'.OR. DEBUG) THEN
      C
      C***** FRACTURE OUTPUT FLAGS
      C
          IF(FRACTURE) READ(JBIN) (IPRNTFA(I), I = 1, NFVARA)
      C
      C***** MATRIX OUTPUT FLAGS
      C
          IF(MATRIX) READ(JBIN) (IPRNTMA(I), I = 1, NMVARA)
          ENDIF
      C
      C***** READ PRINT FLAGS OF CDB FILES
      C
          IF(FINFILETYPE .EQ. 'CDB' .OR. FINFILETYPE .EQ. 'ASC-CDB'
          1 .OR. FINFILETYPE .EQ. 'BIN-CDB' .OR. FINFILETYPE .EQ.
          2 'ASC-BIN-CDB'.OR. DEBUG) THEN
      C
      C***** FRACTURE OUTPUT FLAGS
      C
          IF(FRACTURE) READ(JBIN) (IPRNTFC(I), I = 1, NFVARC)
      C
      C***** MATRIX OUTPUT FLAGS
      C
```

```
      IF(MATRIX) READ(JBIN) (IPRNTMC(I), I = 1, NMVARC)
      ENDIF
C
C===== START MATRIX INITIALIZATION =====
C
C----- Matrix Porosity Initialization -----
C
      IF(MATRIX) THEN
      CALL BRE33D(PORMOLDGRID,JBIN)
C
C----- Matrix Saturation Initialization -----
C
      CALL BRE33D(SWMOLDGRID,JBIN)
C
C----- BRINE VOLUME Initialization -----
C
      CALL BRE33D(BRVOLGRID,JBIN)
C
C----- TOTAL DISSOLVED MASS PER GRID BLOCK -----
C
      IF(IPRNTMB(1) .GT. 0) CALL BRE33D (SUMM_DISGRID,JBIN)
C
C-----TOTAL PRECIPITATED MASS PER GRID BLOCK -----
C
      IF(IPRNTMB(2) .GT. 0) CALL BRE33D (SUMM_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED MASS PER GRID BLOCK -----
C
      IF(IPRNTMB(3) .GT. 0) CALL BRE33D (SUMM_TOTALGRID,JBIN)
C
C----- TOTAL DISSOLVED CURIES PER GRID BLOCK -----
C
      IF(IPRNTMB(4) .GT. 0) CALL BRE33D (CSUMM_DISGRID,JBIN)
C
C----- TOTAL PRECIPITATED CURIES PER GRID BLOCK -----
C
      IF(IPRNTMB(5) .GT. 0) CALL BRE33D (CSUMM_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED CURIES PER GRID BLOCK -----
C
      IF(IPRNTMB(6) .GT. 0) CALL BRE33D (CSUMM_TOTALGRID,JBIN)
C
      DO 1000 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
C
C----- INITIAL VOLUMETRIC CONCENTRATION -----
C
      IF(IPRNTMB(7) .GT. 0) CALL BRE23D(CMOLDGRID,JBIN,J)
C
C----- INITIAL DISSOLVED MASS -----
C
      IF(IPRNTMB(8) .GT. 0) CALL BRE23D(BLOCM_DIS_MASSGRID,JBIN,J)
C
C----- INITIAL MASS OF THE PRECIPITATE -----
C
      IF(IPRNTMB(9) .GT. 0) CALL BRE23D(BLOCM_PRC_MASSGRID,JBIN,J)
C
C----- INITIAL TOTAL MASS ON SOIL BASIS -----
C
      IF(IPRNTMB(10) .GT. 0)
      & CALL BRE23D(ADPRCONM_TOTAL_MASSGRID,JBIN,J)
C
C----- INITIAL CURIES OF VOLUMETRIC CONCENTRATION -----
C
      IF(IPRNTMB(11) .GT. 0)
      & CALL BRE23D(VOLM_CONC_CURIESGRID,JBIN,J)
C
C----- INITIAL CURIES OF DISSOLVED NUCLIDES -----
```

```
C
      IF(IPRNTMB(12) .GT. 0)
        & CALL BRE23D(DISM_MASS_CURIESGRID,JBIN,J)
C----- INITIAL CURIES OF THE PRECIPITATE -----
C
      IF(IPRNTMB(13) .GT. 0)
        & CALL BRE23D(PRCIPM_MASS_CURIESGRID,JBIN,J)
C----- INITIAL CURIES OF TOTAL MASS -----
C
      IF(IPRNTMB(14) .GT. 0)
        & CALL BRE23D(TOTALM_MASS_CURIESGRID,JBIN,J)
      1000 CONTINUE
          ENDIF
C
C===== START FRACTURE INITIALIZATION =====
C----- Fracture Porosity Initialization -----
C
          IF(FRACTURE) THEN
            CALL BRE33D(PORFOLDGRID,JBIN)
C----- Fracture Saturation Initialization -----
C
            CALL BRE33D(SWFOLDGRID,JBIN)
C----- TOTAL DISSOLVED MASS PER GRID BLOCK -----
C
            IF(IPRNTFB(1) .GT. 0) CALL BRE33D (SUMF_DISGRID,JBIN)
C-----TOTAL PRECIPITATED MASS PER GRID BLOCK -----
C
            IF(IPRNTFB(2) .GT. 0) CALL BRE33D (SUMF_PRECIPGRID,JBIN)
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED MASS PER GRID BLOCK -----
C
            IF(IPRNTFB(3) .GT. 0) CALL BRE33D (SUMF_TOTALGRID,JBIN)
C----- TOTAL DISSOLVED CURIES PER GRID BLOCK -----
C
            IF(IPRNTFB(4) .GT. 0) CALL BRE33D (CSUMF_DISGRID,JBIN)
C----- TOTAL PRECIPITATED CURIES PER GRID BLOCK -----
C
            IF(IPRNTFB(5) .GT. 0)CALL BRE33D (CSUMF_PRECIPGRID,JBIN)
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED CURIES PER GRID BLOCK -----
C
            IF(IPRNTFB(6) .GT. 0)CALL BRE33D (CSUMF_TOTALGRID,JBIN)
C
            DO 2000 J = 1, NUCLIDE
              READ(JBIN) NAME(J)
C----- INITIAL VOLUMETRIC CONCENTRATION -----
C
              IF(IPRNTFB(7) .GT. 0)CALL BRE23D(CFOLDGRID,JBIN,J)
C----- INITIAL DISSOLVED MASS -----
C
              IF(IPRNTFB(8) .GT. 0)CALL BRE23D(BLOCF_DIS_MASSGRID,JBIN,J)
C----- INITIAL MASS OF THE PRECIPITATE -----
C
              IF(IPRNTFB(9) .GT. 0)CALL BRE23D(BLOCF_PRC_MASSGRID,JBIN,J)
C----- TOTAL MASS ON SOIL BASIS -----
C
```

```
      IF(IPRNTFB(10) .GT. 0)
      & CALL BRE23D(ADPRCONF_TOTAL_MASSGRID,JBIN,J)
C
C----- INITIAL CURIES OF VOLUMETRIC CONCENTRATION -----
C
      IF(IPRNTFB(11) .GT. 0)
      & CALL BRE23D(VOLF_CONC_CURIESGRID,JBIN,J)
C
C----- INITIAL CURIES OF DISSOLVED NUCLIDES -----
C
      IF(IPRNTFB(12) .GT. 0)
      & CALL BRE23D(DISF_MASS_CURIESGRID,JBIN,J)
C
C----- INITIAL CURIES OF THE PRECIPITATE -----
C
      IF(IPRNTFB(13) .GT. 0)
      & CALL BRE23D(PRCIPF_MASS_CURIESGRID,JBIN,J)
C
C----- INITIAL CURIES OF TOTAL MASS -----
C
      IF(IPRNTFB(14) .GT. 0)
      & CALL BRE23D(TOTALF_MASS_CURIESGRID,JBIN,J)
2000 CONTINUE
C
      ENDIF
      RETURN
      ENDIF
C
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! END OF INTIALIZATION AND INPUT ECHO
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
C
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! OUTPUT TIME DEPENDENT VARIABLES !!!!!!!!!!!!!!!!!!!!!
C
C%%%%%%%%%% SPECIAL INTEREST VARIABLES OUTPUT %%%%%%%%%%%
C
C----- MASS FLUXES CROSSING J-1 AND I-1 BOUNDARIES -----
C***** MATRIX
C
      IF(SINGLE_POROSITY .AND. MATRIX) THEN
      DO 2500 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
      CALL BRE23D(FLUXJM1MGRID,JBIN,J)
      CALL BRE23D(FLUXIM1MGRID,JBIN,J)
2500 CONTINUE
      ENDIF
C
C***** FRACTURE
C
      IF(SINGLE_POROSITY .AND. FRACTURE) THEN
      DO 2600 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
      CALL BRE23D(FLUXJM1FGRID,JBIN,J)
      CALL BRE23D(FLUXIM1FGRID,JBIN,J)
2600 CONTINUE
      ENDIF
C
C***** DUAL-POROSITY OR DUAL-PERMEABILITY
C
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY) THEN
      DO 2700 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
      CALL BRE23D(FLUXJM1TGRID,JBIN,J)
      CALL BRE23D(FLUXIM1TGRID,JBIN,J)
2700 CONTINUE
      ENDIF
C
C----- MASS DISSOLVED IN THE BRINE FROM GAS PHASE -----
C
      IF(PHASETYPE .EQ. 'G') THEN
```

```
C
C***** MATRIX
C
      IF(SINGLE_POROSITY .AND. MATRIX) THEN
      DO 2710 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
      CALL BRE23D(CONDMMASSGRID,JBIN,J)
2710  CONTINUE
      ENDIF

C
C***** FRACTURE
C
      IF(SINGLE_POROSITY .AND. FRACTURE) THEN
      DO 2720 J = 1, NUCLIDE
      READ(JBIN) NAME(J)
      CALL BRE23D(CONDFMASSGRID,JBIN,J)
2720  CONTINUE
      ENDIF

C
C***** DUAL POROSITY OR DUAL PERMEABILITY
C
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY) THEN
      DO 2730 J = 1, NUCLIDE
      READ(JBIN) NUCLIDE
      CALL BRE23D(CONDMMASSGRID,JBIN,J)
      CALL BRE23D(CONDFMASSGRID,JBIN,J)
2730  CONTINUE
      ENDIF
      ENDIF

C
C----- SOURCE FROM THE ACTINIDE SOURCE SUBMODEL -----
C Note that the source is in kg/s and injected only in the fracture
C
      IF(STOCKMAN) THEN
      DO 2740 J = 1,NUCLIDE
      READ(JBIN) NUCLIDE
      CALL BRE23D(CONCINJFGRID,JBIN,J)
2740  CONTINUE
      ENDIF

C
C----- BRINE INTERFACES VOLUMETRIC FLUXES (M3/S) -----
C***** FRACURE
C
      IF(SINGLE_POROSITY .AND. FRACTURE) THEN
      CALL BRE33D (QWXF_GRID,JBIN)
      CALL BRE33D (QWYF_GRID,JBIN)
      ENDIF

C
C***** MATRIX
C
      IF(SINGLE_POROSITY .AND. MATRIX) THEN
      CALL BRE33D (QWXM_GRID,JBIN)
      CALL BRE33D (QWYM_GRID,JBIN)
      ENDIF

C
C***** DUAL-PERMEABILITY OR POROSITY
C
      IF(DUAL_POROSITY .OR. DUAL_PERMEABILITY) THEN
      CALL BRE33D (QWXF_GRID,JTEMP)
      CALL BRE33D (QWYF_GRID,JTEMP)
      CALL BRE33D (QWXM_GRID,JTEMP)
      CALL BRE33D (QWYM_GRID,JTEMP)
      ENDIF

CI
CI=====I
CI                      MATRIX                      I
CI=====I
C
      IF(MATRIX) THEN
C
```

```
C----- PRODUCTION RATE OF EACH BLOCK -----
C
C      CALL BRE33D (QPRMGRID,JBIN)
C
C----- INJECTION RATE OF EACH BLOCK -----
C
C      CALL BRE33D (QINMGRID,JBIN)
C
C----- BRINE VOLUME IN EACH GRID BLOCK-----
C
C      CALL BRE33D(BRVOLGRID,JBIN)
C
C***** USER SPECIFIED VARIABLES OUTPUT *****
C
C----- TOTAL DISSOLVED MASS PER GRID BLOCK -----
C
C      IF(IPRNTMB(1) .GT. 0) CALL BRE33D (SUMM_DISGRID,JBIN)
C
C-----TOTAL PRECIPITATED MASS PER GRID BLOCK -----
C
C      IF(IPRNTMB(2) .GT. 0) CALL BRE33D (SUMM_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED MASS PER GRID BLOCK -----
C
C      IF(IPRNTMB(3) .GT. 0) CALL BRE33D (SUMM_TOTALGRID,JBIN)
C
C----- TOTAL DISSOLVED CURIES PER GRID BLOCK -----
C
C      IF(IPRNTMB(4) .GT. 0) CALL BRE33D (CSUMM_DISGRID,JBIN)
C
C----- TOTAL PRECIPITATED CURIES PER GRID BLOCK -----
C
C      IF(IPRNTMB(5) .GT. 0) CALL BRE33D (CSUMM_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED CURIES PER GRID BLOCK -----
C
C      IF(IPRNTMB(6) .GT. 0) CALL BRE33D (CSUMM_TOTALGRID,JBIN)
C
C      DO 3000 J = 1, NUCLIDE
C      READ(JBIN) NAME(J)
C
C===== VOLUMETRIC CONCENTRATION =====
C
C      IF(IPRNTMB(7) .GT. 0) CALL BRE23D(CMGRID,JBIN,J)
C
C===== DISSOLVED MASS =====
C
C      IF(IPRNTMB(8) .GT. 0) CALL BRE23D(BLOCM_DIS_MASSGRID,JBIN,J)
C
C===== MASS OF THE PRECIPITATE =====
C
C      IF(IPRNTMB(9) .GT. 0) CALL BRE23D(BLOCM_PRC_MASSGRID,JBIN,J)
C
C===== TOTAL MASS ON SOIL BASIS =====
C
C      IF(IPRNTMB(10) .GT. 0)
C      & CALL BRE23D(ADPRCONM_TOTAL_MASSGRID,JBIN,J)
C
C===== CURIES OF VOLUMETRIC CONCENTRATION =====
C
C      IF(IPRNTMB(11) .GT. 0)
C      & CALL BRE23D(VOLM_CONC_CURIESGRID,JBIN,J)
C
C===== CURIES OF DISSOLVED NUCLIDES =====
C
C      IF(IPRNTMB(12) .GT. 0)
C      & CALL BRE23D(DISM_MASS_CURIESGRID,JBIN,J)
```

```
C
C===== CURIES OF THE PRECIPITATE =====
C
      IF(IPRNTMB(13) .GT. 0)
        & CALL BRE23D(PRCIPM_MASS_CURIESGRID,JBIN,J)
C
C===== CURIES OF TOTAL MASS =====
C
      IF(IPRNTMB(14) .GT. 0)
        & CALL BRE23D(TOTALM_MASS_CURIESGRID,JBIN,J)
C
C----- MATRIX MATERIAL BALANCE ERROR -----
C
      READ(JBIN)NAME(J),BLOCKMBMMAX(J),CMBIM(J),CMBTM(J),
        & SUMRTM(J),SUMQTM(J)
3000 CONTINUE
                                ENDIF
C
CI=====I
CI                                FRACTURE                                I
CI=====I
C
      IF(FRACTURE) THEN
C
C----- PRODUCTION RATE OF EACH BLOCK -----
C
        CALL BRE33D (QPRFGRID,JBIN)
C
C----- INJECTION RATE OF EACH BLOCK -----
C
        CALL BRE33D (QINFGRID,JBIN)
C
C***** USER SPECIFIED VARIABLES OUTPUT *****
C
C----- TOTAL DISSOLVED MASS PER GRID BLOCK -----
C
        IF(IPRNTFB(1) .GT. 0) CALL BRE33D (SUMF_DISGRID,JBIN)
C
C-----TOTAL PRECIPITATED MASS PER GRID BLOCK -----
C
        IF(IPRNTFB(2) .GT. 0) CALL BRE33D (SUMF_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED MASS PER GRID BLOCK -----
C
        IF(IPRNTFB(3) .GT. 0) CALL BRE33D (SUMF_TOTALGRID,JBIN)
C
C----- TOTAL DISSOLVED CURIES PER GRID BLOCK -----
C
        IF(IPRNTFB(4) .GT. 0) CALL BRE33D (CSUMF_DISGRID,JBIN)
C
C----- TOTAL PRECIPITATED CURIES PER GRID BLOCK -----
C
        IF(IPRNTFB(5) .GT. 0) CALL BRE33D (CSUMF_PRECIPGRID,JBIN)
C
C----- THE TOTAL OF DISSOLVED, PRECIPITATED, -----
C----- AND SORBED CURIES PER GRID BLOCK -----
C
        IF(IPRNTFB(6) .GT. 0) CALL BRE33D (CSUMF_TOTALGRID,JBIN)
        DO 4000 J = 1, NUCLIDE
          READ(JBIN) NAME(J)
C
C===== VOLUMETRIC CONCENTRATION =====
C
        IF(IPRNTFB(7) .GT. 0) CALL BRE23D(CFGRID,JBIN,J)
C
C===== DISSOLVED MASS =====
C
        IF(IPRNTFB(8) .GT. 0) CALL BRE23D(BLOCF_DIS_MASSGRID,JBIN,J)
```



```
      VAROUTPUT(I,J,K,ICOUNT) = 0.0
      VAROUTPUT(I,J,K,ICOUNT) = ARRAY(I,J,K)
20    CONTINUE

C
      RETURN
C
C***** END OF SUBROUTINE  BRE23D *****
C
      END
C*****

C*****
C----- BINRHEAD.INC -----
C*****
C
C-----
C
C          BINRHEAD.INC
C
C      Purpose:
C      -----
C      This include file has the common blocks and the declarations
C      required for NUTS_BIN_READ.
C
C      Author:      Ali A. Shinta
C      -----
C
C      Call: NONE
C      Called By: NUTS_BIN_READ
C
C      Arguments:
C      -----
C      AS DEFINED PREVIOUSLY
C-----
C
C      IMPLICIT NONE
C-----
C
C      COMMON/DIMENSION/DXGRID(MX,MY,MZ),DYGRID(MX,MY,MZ),
C      & DZGRID(MX,MY,MZ),VRGRID(MX,MY,MZ),AXGRID(MX,MY,MZ),
C      & AYGRID(MX,MY,MZ),AZGRID(MX,MY,MZ),X(MX,MY,MZ),
C      & Y(MX,MY,MZ),Z(MX,MY,MZ),NBLOCK,IDIMENSION,
C      & DIRECTION
C
C      REAL DXGRID,DYGRID,DZGRID,VRGRID,AXGRID,AYGRID,AZGRID,X,Y,Z
C
C      INTEGER NBLOCK,IDIMENSION
C
C      CHARACTER*10 DIRECTION
C-----
C      COMMON/PROPERTIES/RHOGRID(MX,MY,MZ),QPRMGRID(MX,MY,MZ),
C      & QPRFGRID(MX,MY,MZ),QINMGRID(MX,MY,MZ),QINFGRID(MX,MY,MZ),
C      & PHGRID(MX,MY,MZ),PORFOLDGRID(MX,MY,MZ),PORMOLDGRID(MX,MY,MZ),
C      & SWFOLDGRID(MX,MY,MZ),SWMOLDGRID(MX,MY,MZ),PHREQ
C
C      REAL RHOGRID,QPRMGRID,QPRFGRID,QINMGRID,QINFGRID,PHGRID,
C      & PORFOLDGRID,PORMOLDGRID,SWFOLDGRID,SWMOLDGRID
C
C      CHARACTER*20 PHREQ
C-----
C      COMMON/MISCELL/ZTIME,IFLAGTIME,NUCLIDE,NO_PHASES,NOCONTINUM,
C      & NO_TIMESTEP
C
C      REAL ZTIME
C
C      INTEGER IFLAGTIME,NUCLIDE,NO_PHASES,NOCONTINUM,NO_TIMESTEP
```

```
C-----
COMMON/UFNAMETITLES/JBIN, FINFILETYPE, ANSWERTEST, RADINPUT,
& RADOUTPUT, RADOUTBIN, RADOUTASC, RADOUTCDB, VAR (NVARTIT) ,
& NUTS_TITLE, INTITLE, COMBTITLE, FILE_NAME, DEBUG
C
C      INTEGER JBIN
C
C      CHARACTER*20 FINFILETYPE, ANSWERTEST
C      CHARACTER*35 VAR
C      CHARACTER*80 RADINPUT, RADOUTPUT, RADOUTBIN, RADOUTASC,
& FILE_NAME, RADOUTCDB
C      CHARACTER*100 NUTS_TITLE, INTITLE
C      CHARACTER*132 COMBTITLE
C
C      LOGICAL DEBUG
C-----
COMMON/FMMASSES/ADPRCONF_TOTAL_MASSGRID (MX, MY, MZ, NC) ,
& ADPRCONM_TOTAL_MASSGRID (MX, MY, MZ, NC) ,
& BLOCF_DIS_MASSGRID (MX, MY, MZ, NC) ,
& BLOCM_DIS_MASSGRID (MX, MY, MZ, NC) ,
& BLOCF_PRC_MASSGRID (MX, MY, MZ, NC) ,
& BLOCM_PRC_MASSGRID (MX, MY, MZ, NC) ,
& SUMF_DISGRID (MX, MY, MZ) ,
& SUMF_PRECIPGRID (MX, MY, MZ) ,
& SUMF_TOTALGRID (MX, MY, MZ) ,
& SUMM_DISGRID (MX, MY, MZ) ,
& SUMM_PRECIPGRID (MX, MY, MZ) ,
& SUMM_TOTALGRID (MX, MY, MZ)
C
C      REAL ADPRCONF_TOTAL_MASSGRID, ADPRCONM_TOTAL_MASSGRID,
& BLOCF_DIS_MASSGRID, BLOCM_DIS_MASSGRID, BLOCF_PRC_MASSGRID,
& BLOCM_PRC_MASSGRID, SUMF_DISGRID, SUMF_PRECIPGRID,
& SUMF_TOTALGRID, SUMM_DISGRID, SUMM_PRECIPGRID,
& SUMM_TOTALGRID
C-----
COMMON/FMCURIES/VOLM_CONC_CURIESGRID (MX, MY, MZ, NC) ,
& DISM_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& PRCIPM_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& TOTALM_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& VOLF_CONC_CURIESGRID (MX, MY, MZ, NC) ,
& DISF_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& PRCIPF_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& TOTALF_MASS_CURIESGRID (MX, MY, MZ, NC) ,
& CSUMF_DISGRID (MX, MY, MZ) , CSUMF_PRECIPGRID (MX, MY, MZ) ,
& CSUMF_TOTALGRID (MX, MY, MZ) , CSUMM_DISGRID (MX, MY, MZ) ,
& CSUMM_PRECIPGRID (MX, MY, MZ) , CSUMM_TOTALGRID (MX, MY, MZ)
C
C      REAL VOLM_CONC_CURIESGRID, DISM_MASS_CURIESGRID,
& PRCIPM_MASS_CURIESGRID, TOTALM_MASS_CURIESGRID,
& VOLF_CONC_CURIESGRID, DISF_MASS_CURIESGRID,
& PRCIPF_MASS_CURIESGRID, TOTALF_MASS_CURIESGRID,
& CSUMF_DISGRID, CSUMF_PRECIPGRID,
& CSUMF_TOTALGRID, CSUMM_DISGRID,
& CSUMM_PRECIPGRID, CSUMM_TOTALGRID
C-----
COMMON/BACPRINTFLAGS/IPRNTFB (NFVARB) , IPRNTMB (NMVARB) ,
& IPRNTFA (NFVARA) , IPRNTMA (NMVARA) ,
& IPRNTFC (NFVARC) , IPRNTMC (NMVARC)
C
C      INTEGER IPRNTFB, IPRNTMB, IPRNTFA, IPRNTMA, IPRNTFC, IPRNTMC
C-----
COMMON/FMCONCMATBERR/CMBIF (NC) , CMBTF (NC) , SUMRTF (NC) ,
& SUMQTF (NC) , BLOCKMBFMAX (NC) , CMBIM (NC) , CMBTM (NC) ,
& SUMRTM (NC) , SUMQTM (NC) , BLOCKMBMMAX (NC)
C
C      REAL CMBIF, CMBTF, SUMRTF, SUMQTF, BLOCKMBFMAX, CMBIM, CMBTM,
& SUMRTM, SUMQTM, BLOCKMBMMAX
C-----
COMMON/RADSITE/ELEMNT_SOLB_LIMIT (NC) , C0 (NC) , C1 (NC) , C2 (NC) ,
```

```
& C3 (NC) , C4 (NC) , C5 (NC) , XMOLWT (NC) , COMPINT (NC) , WASTEVOL (NS) ,  
& XLAMDA (NC) , IWASTE (NB, NC) , NCOMPONENT (NS) , NSITES, NOELEMENT,  
& RAD (NC) , ELTEMP_SOLB (NC)  
C  
REAL ELEMNT_SOLB_LIMIT, C0, C1, C2, C3, C4, C5, XMOLWT, COMPINT,  
& WASTEVOL, XLAMDA  
C  
INTEGER IWASTE, NCOMPONENT, NSITES, NOELEMENT  
C  
CHARACTER*20 RAD  
C  
LOGICAL ELTEMP_SOLB  
C-----  
COMMON/NAMES/PARENT_NAME (NC) , NAME (NC) ,  
& DAUGHTER_NAME (NC) , ELEMENT_NAME (NC) ,  
& GROUP_NAME (NC) , SITE_NAME (NS)  
C  
CHARACTER*20 PARENT_NAME, NAME, DAUGHTER_NAME, ELEMENT_NAME,  
& GROUP_NAME, SITE_NAME  
C-----  
COMMON/ADSORPPROP/XLFGGRID (MX, MY, MZ, NC) , XLMGRID (MX, MY, MZ, NC) ,  
& XFDCFGGRID (MX, MY, MZ, NC) , XFDCMGRID (MX, MY, MZ, NC) ,  
& XFCFGGRID (MX, MY, MZ, NC) , XFCMGRID (MX, MY, MZ, NC) ,  
& XLDCMGRID (MX, MY, MZ, NC) , XLDCFGGRID (MX, MY, MZ, NC) ,  
& XLCMGRID (MX, MY, MZ, NC) , XLCFGGRID (MX, MY, MZ, NC) ,  
& REFTEMPKD (NC) , ADSEXPCEFF (NC) ,  
& FSORPTION (NC) , MSORPTION (NC) , ADSTYPEF ,  
& ADSTYPEM, MAT_ISOTHERM, FRAC_ISOTHERM,  
& MADSTEMPDEP (NC) , FADSTEMPDEP (NC)  
C  
REAL XLFGGRID, XLMGRID, XFDCFGGRID, XFDCMGRID, XFCFGGRID, XFCMGRID,  
& XLDCMGRID, XLDCFGGRID, XLCMGRID, XLCFGGRID, REFTEMPKD, ADSEXPCEFF  
C  
CHARACTER*20 FSORPTION, MSORPTION, ADSTYPEF, ADSTYPEM,  
& MAT_ISOTHERM, FRAC_ISOTHERM  
C  
LOGICAL MADSTEMPDEP, FADSTEMPDEP  
C-----  
COMMON/DISPPROP/ALLFGGRID (MX, MY, MZ) , ALTFGRID (MX, MY, MZ) ,  
& TORFGRID (MX, MY, MZ) , ALLMGRID (MX, MY, MZ) , ALTMGRID (MX, MY, MZ) ,  
& TORMGRID (MX, MY, MZ) , DMOL (NC) , VISREF (NC) , TREF (NC) , MDISPREQ,  
& FDISPREQ, MFDISPREQ, DMOLTEMDEP (NC)  
C  
REAL ALLFGGRID, ALTFGRID, TORFGRID, ALLMGRID, ALTMGRID,  
& TORMGRID, DMOL, VISREF, TREF  
C  
CHARACTER*20 MDISPREQ, FDISPREQ, MFDISPREQ  
C  
LOGICAL DMOLTEMDEP  
C-----  
COMMON/CONCFM/CFGRID (MX, MY, MZ, NC) , CMGRID (MX, MY, MZ, NC) ,  
& CFOLDGRID (MX, MY, MZ, NC) , CMOLDGRID (MX, MY, MZ, NC)  
C  
REAL CFGRID, CMGRID, CFOLDGRID, CMOLDGRID  
C-----  
COMMON/CONCSOURCE/CONCINJFGRID (MX, MY, MZ, NC) ,  
& CONCDIRMGRID (MX, MY, MZ, NC) , CONCDIRFGRID (MX, MY, MZ, NC) ,  
& CONCINJMGRID (MX, MY, MZ, NC) , TIMEMSTRT, TIMEMEND, TIMEFSTRT,  
& TIMEFEND, MDIRICHLET, FDIRICHLET, MSTATUSINJ, FSTATUSINJ, STOCKMAN  
C  
REAL CONCINJFGRID, CONCDIRMGRID, CONCDIRFGRID, CONCINJMGRID,  
& TIMEMSTRT, TIMEMEND, TIMEFSTRT, TIMEFEND  
C  
CHARACTER*20 MSTATUSINJ, FSTATUSINJ  
C  
LOGICAL MDIRICHLET, FDIRICHLET, STOCKMAN  
C-----  
COMMON/PORMEDIA/MEDIUM, FRACTURE, MATRIX, SINGLE_POROSITY,  
& DUAL_POROSITY, DUAL_PERMEABILITY  
C
```

```

      CHARACTER*24 MEDIUM
C
      LOGICAL FRACTURE, MATRIX, SINGLE_POROSITY, DUAL_POROSITY,
      & DUAL_PERMEABILITY
C-----
      COMMON/HEADINGS/NPNAME, NVRSION, NREVDATA, NCPUNAME, SDATE, STIME
C
      CHARACTER SDATE*9, STIME*8, NCPUNAME*32
      CHARACTER*8 NPNAME, NVRSION, NREVDATA
C-----
      COMMON/INTERFLUXES/FLUXJM1MGRID (MX, MY, MZ, NC) ,
      & FLUXJM1FGRID (MX, MY, MZ, NC) , FLUXJM1TGRID (MX, MY, MZ, NC) ,
      & FLUXIM1MGRID (MX, MY, MZ, NC) , FLUXIM1FGRID (MX, MY, MZ, NC) ,
      & FLUXIM1TGRID (MX, MY, MZ, NC)
C
      REAL FLUXJM1MGRID, FLUXJM1FGRID, FLUXJM1TGRID,
      & FLUXIM1MGRID, FLUXIM1FGRID, FLUXIM1TGRID
C-----
      COMMON/BCONDMAS/CONDMMASSGRID (MX, MY, MZ, NC) ,
      & CONDFMASSGRID (MX, MY, MZ, NC) , EQCI (NC) , EQCS (NC) ,
      & PHASETYPE
C
      REAL CONDMMASSGRID, CONDFMASSGRID, EQCI, EQCS
C
      CHARACTER PHASETYPE*10
C-----
      COMMON/BRVOLGRID/BRVOLGRID (MX, MY, MZ)
      REAL*4 BRVOLGRID
C-----
      COMMON/BRINERATES/QWXF_GRID (MX, MY, MZ) , QWYF_GRID (MX, MY, MZ) ,
      & QWXM_GRID (MX, MY, MZ) , QWYM_GRID (MX, MY, MZ)
      REAL*4 QWXF_GRID, QWYF_GRID, QWXM_GRID, QWYM_GRID
C
C***** END OF BINRHEAD.INC *****
C
```

Appendix L: Review Forms

As a convenience to users, this section contains a complete history of all the Quality-Assurance Review Forms issued during the review of this User's Manual. Review forms are issued after the User's Manual has been completed and reviewed. Thus, they will be appended to the manual after it has been paginated and therefore they may not, themselves, be paginated.

WPO# 46002

WASTE ISOLATION PILOT PLANT Sandia National Laboratories	User's Manual Reviewer's Form Form Number: 459-B Effective: 9/12/96	
	Procedure: <u>QAP 19-1</u>	Page <u>1</u> of <u>1</u>

Reviewer Instructions:

Check "Yes" for each item reviewed and found acceptable.
 Check "No" for each item which requires further work.
 Check "N/A" for items which are not applicable.
 Check "N/R" for items not reviewed (only if there are multiple reviewers).
 Complete a User's Manual Review Form for each reviewer.
 Prior to sign-off of the User's Manual, all "No" items shall be appropriately addressed by the code sponsor so that "Yes" or "N/A" may be checked.
 Include this form as part of the baseline User's Manual.
 The reviewer may document the rationale of the check marks in a memorandum included with this form in the User's Manual.

Does the user's manual contain as appropriate:

- | | | | | |
|--|---|-----------------------------|------------------------------|------------------------------|
| 1. A statement(s) of functional requirements (consistent with those in the RD) and system limitations? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 2. An explanation of the mathematical model and numerical models? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 3. Physical and mathematical assumptions? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 4. The capabilities and limitations inherent in the software? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 5. Instructions that describe the user's interaction with the software? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 6. The identification of input parameters, formats, and valid ranges? * | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 7. Messages initiated as a result of improper input and how the user can respond? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 8. The identification and description of output specifications and formats? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 9. A description of any required training necessary to use the software? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |
| 10. The identification of components of the code that were not tested? | <input checked="" type="checkbox"/> Yes | <input type="checkbox"/> No | <input type="checkbox"/> N/A | <input type="checkbox"/> N/R |

11. **Reviewer's Signature:**
Alex H. Treadway Alex H. Treadway 8/8/97
 Reviewer's Name (print) Signature Date

* Some of the ranges are based on user experience and training

WPO #46002

WASTE ISOLATION PILOT PLANT Sandia National Laboratories	Review Comment (Software) Form Number: 299-A Effective: 9/12/96
Procedure: QAP 19-1 Page <u>1</u> of <u>2</u>	

SECTION I

1. Software Name: <u>NUTS</u>	Version ID: <u>2.05</u>
----------------------------------	----------------------------

2. Type of Comment: (check one) Required Suggested

3. Software Classification (check one)
(Refer to QAP 19-1 Appendix A for software classification definitions)

<input type="checkbox"/> Firmware DAS	<input type="checkbox"/> Purchased DAS	<input type="checkbox"/> Developed DAS
<input type="checkbox"/> OFF-THE-SHELF-SW	<input type="checkbox"/> QA-Vendor-SW	<input type="checkbox"/> Vendor-SW
<input checked="" type="checkbox"/> SNL-SW		

4. Comment: (check one, attach pages as needed) Technical Document

See Attachment

5. Alex H. Treadway Alex H. Treadway 7/24/97
 Reviewer (Print) Signature Date

SECTION II

6. Comment Response: (check one; attach pages as needed) Agree Disagree

see attachment

Information Only

WASTE ISOLATION PILOT PLANT Sandia National Laboratories	Review Comment (Software)	
	Form Number: 299-A	Effective: 9/12/96
Procedure: <u>19-1</u>		Revision: <u>2</u>
		Page <u>2</u> of <u>2</u>

SECTION III - SIGNATURES			
7.	<u>Ali A. Shinta</u> Code Sponsor's Name (print)	<u>Ali A. Shinta</u> Signature	<u>8/8/97</u> Date
8.	<u>Alex H. Treadway</u> Reviewer's Name (Agreement) (print)	<u>Alex H. Treadway</u> Signature	<u>8/8/97</u> Date

WPO#: 46002 SWCF File Code: SWCF-A 1.4.0/6.1

Keywords: NUTS: NUT: UM : : ,SFT

A or C WBS #(s) Alpha Code

Information Only

SQA Review of NUTS User's Manual for Version 2.05

General Comments

- (1) 'Langumir' -> 'Langmuir' throughout document
- (2) I did not see any messages which shows that the implicit treatment for precipitation was successful or failed to converge. If the algorithm fails to converge, then an error has occurred and the code should abort with an error message!

Page Comments

- (1) pg 7, R.9 - Does this requirement include nonlinear solubilities?
- (2) pg 7, R.10 - Is there a discussion of the implementation of the Neumann boundary condition? There does not seem to be any input for this boundary condition.
- (3) pg 11, last para - 'Ignoring lateral dispersion': do you mean lateral, which I consider to be the same as transverse, or do you mean longitudinal which does cause streamwise elongation. The numerical dispersion due to the grid definitely exceeds the likely physical dispersion, which I presume has not been determined. I think your signal argument is weak.
- (4) pg 13, Table 1 - Should Asymmetric dispersion be added to the table? Does 'Precipitation' include both explicit and implicit formulations?
- (5) pg 15, foot note - 'Wu and Ruess' -> 'Wu and Pruess'; 'Thomas et al' -> 'Thomas *et al.*'
- (6) pg 16, Sect 4.3 - Are the vertical bars at both ends of your mass conservation equation necessary?
- (7) pg 19, middle - 'solvent density ρ (a constant)'; With a variable pressure (and temperature) the density is not a constant and also is not a constant unless the concentration is trace. Actually, $C=\rho C'$.
- (8) pg 21, 1st para - 'the front will rapidly thicken' -> 'the front will be rapidly smeared'
- (9) pg 22, middle - 'Dispersion in the fracture and matrix is generally small'; Do you have a reference for this statement?
- (10) pg 23, bottom - The effective (diagonal) dispersion coefficient is always positive, otherwise the equation is mathematically unstable. Furthermore the dispersivity coefficients are not constants but can be deduced from the time rate of change of the 'moments-of-inertia' tensor. They will be zero at the source and will become constants only after several correlation lengths of the heterogeneity of the porous media have been sampled. This type of formulation also removes or minimizes upstream dispersion.
- (11) pg 24, bottom - Are both vertical bars needed?
- (12) pg 26, Eq 4.28 - I presume the $\frac{1}{x_2}$ and $\frac{1-x_2}{x_2}$ are powers. I suggest you interchange the n and n+1 with these powers so that they appear correct.
- (13) pg 30, Neumann BC - I presume from this discussion that you cannot 'approximate' a curved boundary but are restricted to domains with either vertical or horizontal boundaries. If so, you should so state. Moreover, how are the $g_2(t)$ entered?
- (14) pg 31, top - 'These conditions are sometimes'; How do you specify a Neumann BC internally? No transport is controlled by the velocity normal to the boundary being zero, not by the normal derivative of the concentration being zero! You can have a no flux BC but I do not see

how your code handles that BC unless it is hard coded.

(15) pg 36, TVD discussion - Do you lag the limiters since TVD is a nonlinear algorithm and uses 5 grid blocks along a given coordinate direction? Are there any stability restrictions since it appears possible to have a zero or negative number on the diagonal if the Courant number is too large? In Eq. 4.72, the sum should be over the absolute value of the difference.

(16) pg 38, Eq 4.76 - You should state the units for K_{wf}^{n+1} (m^3/s) which one determines from HD and is eventually stated in Eq. 4.94. Otherwise the units of the equation will be wrong using the information for the dispersion tensor. 'We will discretize them fully implicitly ...' appears to be awkwardly worded.

(17) pg 40, 'HD =' - Is $\omega_{fi}\phi_{fi}S_{wfi} + (1 - \omega_{fi})\phi_{fi-1}S_{wfi-1}$ the multiplier of the remaining expression? If so, it should be in parenthesis, otherwise only the last part of this expression will be considered to be the multiplier.

(18) pp 38 -52 - I have not verified these equations.

(19) pg 52, 2nd para - 'or interior grid points are used ... boundary conditions' An explanation is needed.

(20) pg 54 - I do not think that the Colloid and Lumped Radioisotope discussion is germane to the User's Manual. These topics are analysis issues. Also it may not be appropriate to reference the CCA Analysis Package.

(21) pg 54, 1st para - The sentence construction starting with 'In this calculation' and continuing to the end of the paragraph, I found to be very awkward. Please reword.

(22) pg 57, bottom - It would help to state that the delta notation you are using for the central finite difference operator applies to the indices, otherwise the equation units are incorrect.

(23) pg 58, 4.5.5 - The nomenclature section is misaligned. Also a sentence to the effect 'We will describe the precipitate governing equations shortly' should be added since the precipitation/dissolution process is governed by a pair of equations. Is diffusion/dispersion neglected in general or only for the current use? If diffusion/dispersion is not neglected, does the algorithm still converge?

(24) pg 59, 2nd para - $(0,X) \rightarrow [0,X]$, similarly for y. $0 \leq x \leq X \rightarrow 0 < x < X$, similarly for y.

(25) pg 59, 3rd para - 'hereinafter called precipitate' \rightarrow 'hereafter called precipitate,'

(26) pg 59, middle - You talk about mass but use units of kg/m^3 . You need to state, as you do on page 61, that the units are $kg/bulk\ volume$ or $kg/grid\ block\ volume$.

(27) pg 59, last para - 'the capacity' \rightarrow 'The capacity'; 'which is not done neither in' \rightarrow 'which are done neither in'; Does allocation using mole fractions cause the process to be nonlinear?

(28) pg 60, Eq 4.101a - Is M_l^{n+1} given? If so, from where? After reading the discussion on the explicit treatment, I do not think the algorithm is a solution to the governing equations in general and probably only works with a 'small enough' time step. It may be better to delete the explicit treatment discussion and option and use only the implicit treatment.

(29) pg 60, 1) - What is the zigzagging due to? I think you mean 'chattering' or oscillatory behavior of the solution. Moreover what is the effect of the solubility allocation or partitioning, which can be nonlinear?

(30) pg 6, 2nd para - You refer to the problem as an optimization problem. What is your objective function? What are the real constraints? Do you have a reference for you statement. Also when people refer to the penalty method in the context of optimization they usually mean one multiplies

the constraint formulation by a large number and then adds this expression to the objective function to be minimized/maximized, assuming that minimizing the new objective function will minimize both the original objective function and satisfy the constraints. Unfortunately this approach does not always work. I think what you are actually describing in your algorithm is closer to a feedback control problem. You may want to provide references for your statements, particularly for 'the penalty method'.

(31) pg 61, 2) - Should 'concentration equivalent to' be 'concentration equal to'?

(32) pg 61, 3) - 'by solving Equation 4.96'; It appears to me that you are evaluating 4.96 by forming the difference equation and substituting the solubility limit for the grid block plus the values of the concentrations in the five point stencil grid blocks and thus computing the residual, that is, q , which is similar to the way BRAGFLO computes the source/sink value needed to maintain a Dirichlet boundary condition. Is this correct? If so, an expanded discussion would be helpful.

(33) pg 61, 4) - What is the effect of the $\sum_{j=1}^{J_i} \lambda_{ij} P_{ij}$ term? Is the evaluation of Eq. 4.104 done

sequentially?

(34) pg 62, middle - 'activate Dirichlet' -> 'activate the Dirichlet'; 5) and 6) -> 6) and 7)

(35) pg 62, 6) - Should be $|\Delta P_i| \leq \epsilon$

(36) pg 62, last para - 'may not converge': Due to what? too big a time step from BRAGFLO? Is

F a function of only C_1 or is it a function of other isotopes, that is, is it more like $\frac{C_1}{C_1 + C_2}$, which

is a rational function and hence nonlinear?

(37) pg 64, 1st para - You probably should mention that the symbols and logicals have to be setup as well or at least mention that you are assuming that the user is running under the PA environment.

(38) pg 76, Line 22. MORDER - Which of these inputs is for TVD?

(39) pg 77, top - Section ????????

(40) pg 103, middle - 'multiplier' -> 'multipliers'; You may need to check for other places where this has occurred. You may want to bold the last sentence of the colloid discussion.

(41) pg 106, Line 24 - 'ithe' -> 'in the'

(42) pg 116, Line 34 - You are describing VELWXM3D but refer to TVELXM3D, any particular reason?

(43) pg 119, k) - 'udisturbed' -> 'undisturbed'; 'initialized certain kind of'?

(44) pg 123, 6.8 - I think that the *.FOR is the wrong extension for the *_LIB libraries.

(45) pg 127, top - 'witherspoon' -> 'Witherspoon'

(46) pg 127, DeSwann - 1978; SPEJ(?

(47) pg 127, Doctor - Who are the remaining authors?

(48) pg 127, Duguid - Do you have the Vol. and No.?

(49) pg 168, bottom - Any reason for the bolding of the last phrase?

(50) pg 186, 3. - What does the blank line after 'This message will appear ...' signify?

(51) pg 187, 2. - 'may mistakenly input'?

(52) pg 187, 5. - 'in this time' -> 'at this time'

Responses to the review comments on the NUTS User's Manual Version 2.05

General Comments:

- 1) "Langumir" is corrected to "Langmuir" throughout the document.
- 2) There is a message in the code that read as:

```

"Number of iteration for the implicit precipitate exceeded #
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! NUTS IS ABORTED !!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

Please run BRAGFLO or the fluid flow code with refined time step and re-run NUTS"

Page comments:

- 1) No the requirement does not include nonlinear solubility. I will make it clear in the RD/VVP, the solubility tested is linear.

- 2), 13), 14) The last paragraph of Section 4.3.8 "Initial and Boundary Conditions" have been modified to the following:

"In NUTS applications, Dirichlet boundary conditions are input controlled and can be specified at both the outer boundaries or internally at sources. Neumann boundary conditions, on the other hand are restricted to domains with either vertical or horizontal boundaries (no curved boundaries), and are not input controlled and used by default to specify no-diffusive-dispersive transport at the outer boundaries of the simulation domain, i.e., $D_{xx} \frac{\partial C}{\partial x} = 0$, $D_{yy} \frac{\partial C}{\partial y} = 0$, and $D_{zz} \frac{\partial C}{\partial z} = 0$, where D_{xx} , D_{yy} , and D_{zz} are the diagonal component of the dispersion tensor. Therefore, the function $g_2(t)$ in Equation 4.44 is internally set to zero (concentration gradient of values other than zero is not considered in the boundary treatment of NUTS)."

- 3) The Users Manual is not the right document to discuss and justify assumptions. Therefore, the whole argument is removed.
- 4) "Dispersion" in Table 13 is changed to "Dispersion (Symmetric and Asymmetric)" and a footnote is added for the precipitate read as:

"Explicit precipitate is used in the CCA calculations. Implicit precipitate, however, is used in the EPA PAV calculations."

- 5) Editorial: Corrected as suggested.
- 6) Since the vertical bar is bothering you, I removed it.
- 7) Changed to : " In Equation 4.7, if q^* is replaced by its equivalent value (Equation 4.8), and $\rho C'$ is replaced by C , then Equation 4.7 becomes:"

8) Wu and Ruess and Thomas et al are corrected to Wu and Pruess and Thomas *et al* , respectively.

9) the paragraph is changed to:

“In addition to dispersion in the fracture and matrix continua, Fick’s law is used to represent the hydrodynamic dispersion transport between the fracture and the matrix, and is written as:

$$F_{Dm/f} = \phi_m S_m K_{m/f} (C_f - C_m)$$

11) changed to:

$$K = \begin{cases} 1 & \text{if the concentration gradient is in the same direction as the flow gradient} \\ 0 & \text{otherwise} \end{cases}$$

12) Editorial: corrected as suggested (n and n+1 are interchanged with the powers).

13) See Comment 2.

14) See Comment 2.

15)

a) The TVD is not in this version of the Code, It is available in non-QA version. There is an underlined statement in the U.M. in Section 4.4 (page 37) mentioned that TVD is not in this version . The statement is:

“NUTS Version 2.05 accommodates the first three weighting methods mentioned above, namely: one-point upstream, two-point upstream, and mid-point weighting.”

b) Yes, you are right, the limiters lag in time and the procedure is iterative.

c) The TVD criteria is

$$\frac{\Delta t}{\Delta x} v_i^{n+1} \left(1 - \frac{\Phi_{i-1/2}}{2} + \frac{\Phi_{i+1/2}}{2r_{i+1/2}} \right) \geq 0 \tag{1}$$

where v_i is the effective velocity or the flux gradient defined by the spatial derivatives

(for example, $v_i = \frac{F_i - F_{i-1}}{C_i - C_{i-1}}$). If the value of the $\Phi(r)$ is constrained by

$$2 \geq \Phi(r) \geq 0 \text{ and } 2 \geq \frac{\Phi(r)}{r} \geq 0 \tag{2}$$

then Equation 1 is satisfied for all CFL numbers. If the limiters are TVD, there is no restriction on the size of the time step. Therefore, the solution is stable as long as it converges satisfying the constraints of Equation 2+ .

+ Rubin B., and Blunt, M.J.:” Higher order Implicit Flux Limiting Schemes for Black Oil Simulation”, Paper SPE 21222 presented at the 11th SPE Symposium on Reservoir Simulation held in Anaheim, California, Feb. 17-20, 1991.

d) Equation 4.72 is corrected.

16) The sum is made over the absolute value of the difference in Equation 4.77.

“In Equations 4.76 and 4.77 above, the dispersion coefficients K_{wf}^{n+1} and K_{wm}^{n+1} have a unit of (m^3/s), and are related to, but different from those in Equations 4.15 and 4.16 that have a unit of m^2/s . For instance, the relationship between these coefficients at $i + \frac{1}{2}$ interface of the matrix continuum is:

$$K_{wm}^{n+1} (eq. 4.77) = \frac{A_i}{\Delta x_{i+\frac{1}{2}}} \{K_{wm}^{n+1} (eq. 4.16)\}$$

where A_i is the area normal to the flow at the grid block i and $\Delta x_{i+\frac{1}{2}}$ is the distance between x_i and x_{i+1} .”

The first paragraph of section 4.4.1 is changed to:

“Let us now consider the partial differential equations that govern transport in the fracture and the matrix continua. In this section, these equations will be discretized fully implicitly in time and at the grid block interfaces in space, whereas in the subsequent sections, the linear system for different numerical implementations will be developed.”

17) The term $\omega_{fi} \phi_{fi} S_{wfi} + (1 - \omega_{fi}) \phi_{fi-1} S_{wfi-1}$ is placed in parentheses.

18) The viewer has no comment.

19) the sentence is changed to “no fictitious (ghost) grid points are used”.

20) and 21) Lumped isotopes and colloid mobilization are removed.

22) The following statement is added

“In the above formulae, the delta notation (Δ) is used for central finite difference operator and applies to the indices. For example, the term $\Delta_x q_{Gm} C_{nGm}$ is a compact notation refers to the $\{q_{Gm} C_{nGm}\}_{i+\frac{1}{2}} - \{q_{Gm} C_{nGm}\}_{i-\frac{1}{2}}$.”

23) The paragraph is aligned. A sentence is added as suggested in the following paragraph:

“Notice that in Equation 4.96, NUTS disregards sorption and diffusion/dispersion process. The system of N transport equations as given above constitutes only part of the relevant boundary-value problem. To complete the problem specification it is necessary to 1) give the initial concentration throughout the domain at time zero, and 2) describe the

numerical implementation of the dissolution process (solid and dissolved waste interaction). A description for the precipitate governing equation will follow shortly.

the following footnote is also added:

“Even though dispersion and adsorption are neglected in the current use (Equation 4.96), the numerical implementation of the precipitation models is taking into consideration the NUTS capabilities described in Equations 4.38 and 4.39.”

24) Editorial: Corrected as requested.

25) Hereinafter changed to hereafter.

26) Bulk volume is added after m^3 in different places.

27) Editorial typos are corrected. After “NUTS will allocate the solubility for the isotopes according to their mole fractions” the following statement is added:

“in the implicit precipitate treatment and as will be explained later, allocation using mole fractions may lead to highly nonlinear system”

28) In the explicit treatment, the value of M_i^{n+1} is always known. If there is one isotope belongs to a certain element, there will be no competition and the solubility limit and, therefore the mass required to maintain it, will be constant. If there will be several isotopes competing on the same element solubility, their mole fractions (from the concentration) will be known at the end of the time step and thus, the value of M_i^{n+1} will be available (see Equation 4.109). Therefore, the nonlinearity introduced by the mole fraction is not there anymore.

29) The oscillatory behavior is due to the fact that the fluxes are not solubility limited. allocation of solubility will not impact the explicit solution because M_i^{n+1} will be available.

30) The statement “Although the system of Equations 4.96, 4.103, and 4.104 is an optimization problem, it is handled in NUTS differently using the following procedure:” is changed to

“The system of Equations 4.96, 4.103, and 4.104 is solved in NUTS using the following procedure:”

A reference for the penalty method is provided as a footnote.

31) Corrected to equal to.

32) Item 3 is replaced by the following:

- “3) From the computed concentration in step 2, evaluate the precipitate/dissolution source/sink term required to maintain Dirichlet condition (only in the grid blocks with $P_i^k > 0$) by solving for the residual of Equation 4.96 as follows:

$$q_i^{k+1} = \frac{\partial}{\partial t} (\phi S_w C_i) + \nabla \cdot v_w C_i - Q^* + (\phi S_w C_i) \lambda_i - \phi S_w \sum_{p=1}^{J_i} C_{i,p} \lambda_p \quad 4.106$$

Equation 4.106 is solved in the discretized form. Since all the entries for this equation is available after the solution of Equation 4.96, the value of the residual (q_i^{k+1}) can be directly solved for by substituting of the solubility limit for the concentration of the grid block i and the computed concentration (from Equation 4.96) for the surrounding grid blocks (five-point stencil in two-dimensional grid).”

- 33) This is the growth from the parent decay and is available because the whole solution progress sequentially.

34) Editorial: Corrected as requested.

35) Editorial: Corrected as requested.

36) the following footnote is added:

“+ The source of the nonlinearity is the form of the function F. This function, for example, in three-isotope system in which only C_1 and C_2 compete on the same solubility is, $F_1 = \frac{C_1}{C_1 + C_2}$, $F_2 = \frac{C_2}{C_1 + C_2}$, and

$F_3 = 1.0$. Therefore, for C_1 and C_2 , the function is rational and hence nonlinear. Please notice that this function does not depend on the concentration of other isotopes not competing on the same solubility.”

37) The first sentence is changed from “NUTS may be exercised either interactively or through command files.” to:

“Assuming that the user is running under the PA environment (the symbols and the logicals have to be set up), NUTS may be exercised either interactively or through command files.”

38) None:

TVD is not included in any QA version. It is available in a non-QA version.

39) Corrected. The section is 4.5.3.

40) Editorial: Corrected as requested.

41) Editorial: Corrected as requested.

42) Changed to VELWXM3D.

43) Editorial: Corrected as requested.

44) Corrected to:

WP0#46002

“In the building process, the main program NUT_MAIN.FOR will be compiled and linked with NUT_CDBLIB.FOR, NUT_LIB.FOR, and linked with CAMDAT_LIB, CAMCON_LIB, and CAMSUPES_LIB.”

45) Editorial: Corrected as requested.

46) Editorial: Corrected to SPEJ+, 117-122..

47) Other authors are:

P.W. Eslinger

D.M. Elwood

D.W. Engel

M.D. Freshley

A.M. Liebetrau

P.W. Reimus

D.L. Streng

J.E. Tarnner

A.E. Van Luik

48) the following information is added

“Vol. 13, No. 3”

49) Editorial: Corrected by adjusting the font.

50) Editorial: The blank line is removed

51) Editorial: changed to it may input.

52) Editorial: Corrected as suggested.

+ SPEJ refers to the Society of Petroleum Engineers Journal