Appendix E

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**Design Document Criteria**	**Form Number: NP 19-1-4** **Page 1 of 1**

1. **Software Name:** NUTS
2. **Software Version:** 2.08
3. **Document Version:** 2.08
4. **ERMS #:** 573824

Prior to sign-off of the DD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" may be checked. Include this form as part of the DD.

Are the following appropriately defined and documented in the DD?

5.	Major Software Components	☒ Yes
6.	Technical description of the software with respect to: theoretical basis, embodied mathematical model, major control flow, control logic, and data structures	☒ Yes
7.	Functionalities and interfaces of object components, functions, and subroutines.	☒ Yes
8.	Allowable or prescribed ranges for inputs and outputs	☒ Yes
9.	Verifiability: Is the design verifiable through testing or other means?	☒ Yes
10.	Consistency and Traceability: Is the design consistent with and traceable to the software's requirements?	☒ Yes
11.	Technical Feasibility: Is the design technically feasible?	☒ Yes
12.	Implementation: Is the design presented in sufficient detail to allow for implementation as computer software?	☒ Yes

13.	Sungtae Kim **Code Team/Sponsor** (*print*)	Sungtae Kim Signature	12/14/21 Date
14.	Seth King **Technical Reviewer** (*print*)	Signature on File Signature	12/14/20 21 Date
15.	Shelly R. Nielsen **QA Reviewer** (*print*)	Shelly R. Nielsen Signature	2-1-2022 Date
16.	Steve Wagner **Responsible Manager** (*print*)	Steve Wagner Signature	1/20/2022 Date
17.	Jennifer J. Long **SCM Coordinator** (*print*)	Jennifer Long Signature	1/28/22 Date

Information Only

Appendix G

NUCLEAR WASTE MANAGEMENT PROCEDURE Sandia National Laboratories	**User's Manual Criteria**	**Form Number: NP 19-1-6** **Page 1 of 1**

Does the User's Manual contain as appropriate:

1. **Software Name:** NUTS
2. **Software Version:** 2.08
3. **Document Version:** 2.08
4. **ERMS #:** 573824

Prior to sign-off of the User's Manual, all 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 User's Manual.

5.	A statement(s) of functional requirements (consistent with those in the RD)?	☒ Yes	
6.	An explanation of the mathematical model and numerical models, where applicable as based on code functionality?	☒ Yes	☐ N/A
7.	Physical and mathematical assumptions, where applicable as based on code functionality?	☒ Yes	☐ N/A
8.	The capabilities and limitations inherent in the software?	☒ Yes	
9.	Instructions that describe the user's interaction with the software?	☒ Yes	
10.	The identification of input parameters, formats, and valid ranges?	☒ Yes	
11.	Messages initiated as a result of improper input and how the user can respond?	☒ Yes	
12.	The identification and description of output specifications and formats?	☒ Yes	
13.	A description of any required training necessary to use the software?	☒ Yes	
14.	The identification of components of the code that were not tested?	☒ Yes	☐ N/A

15.	Sungtae Kim **Code Team/Sponsor** (*print*)	Sungtae Kim Signature	12/14/21 Date
16.	Seth King **Technical Reviewer** (*print*)	Signature on File Signature	12/14/2021 Date
17.	Shelly R. Nielsen **QA Reviewer** (*print*)	Shelly R. Nielsen Signature	2-1-2022 Date
18.	Steve Wagner **Responsible Manager** (*print*)	Steve Wagner Signature	1/20/2022 Date
19.	Jennifer J. Long **SCM Coordinator** (*print*)	Jennifer Long Signature	1/28/22 Date

Information Only

WIPP PA

Design Document and User's Manual

for

NUTS Version 2.08

Document Version 2.08

ERMS # 573824

February 2022

Information Only

TABLE OF CONTENTS

1.0 INTRODUCTION AND SCOPE 7
1.1 Software Identification 7
1.2 Points of Contact 7
1.3 Code Overview 7
2.0 REQUIREMENTS 9
2.1 Functional Requirements 9
2.2 External Interface Requirements 9
2.3 Other Requirements 10
2.4 Functionality Not Tested 10
3.0 MAJOR SOFTWARE COMPONENTS 12
3.1 Major Control Flow, Control Logic, and Subroutines 12
3.2 Data Structures 20
4.0 MODELS AND METHODS 27
4.1 Background 27
4.1.1 NUTS's general role in the PA 27
4.1.2 Overview of NUTS's Capabilities 28
4.2 Theoretical Foundations of Transport in Fractured Porous Media having Dual Porosity and Dual Permeability 30
4.2.1 Introduction to Fractured Porous Media 30
4.2.2 Basic dissolution processes 31
4.3 Differential Equations Governing Transport in Fractured Porous Media 32
4.3.1 Advective transport 32
4.3.2 Advective transport across fracture/matrix interfaces 35
4.3.3 Diffusive and dispersive transports 36
4.3.4 Asymmetric dispersion 38
4.3.5 Sorption–desorption 39
4.3.6 Matrix/Fracture boundary sorption 40
4.3.7 Decay 41
4.3.8 The overall mass transport equation 43
4.3.9 Initial and boundary conditions 44
4.4 Discretization Methods 45

Information Only

4.4.1 Discrete analogues of the conservation of mass equations 51
4.4.2 One-point upstream-winding discretization of transport equations 53
4.4.3 Two-point upstream-winding discretization of transport equations 57
4.4.4 Mid-point weighting discretization of transport equations 61
4.5 Other important physical and chemical aspects of the modeling 65
4.5.1 Treatment of mobilization involving colloidal materials 65
4.5.2 Lumped radionuclides 66
4.5.3 Temperature dependency 66
4.5.4 Theoretical foundations for the gas-transport model phase 67
4.5.5 Precipitate models 69

5.0 INPUT AND OUTPUT FILES 74
5.1 NUTS input ASCII file 74
5.1.1 Site description and flags 74
5.1.2 Single porosity matrix controls and flags 78
5.1.3 k_d temperature dependency information 82
5.1.4 Molecular diffusion temperature dependency flags 83
5.1.5 Printing frequency information 83
5.1.6 External source flag 84
5.1.7 Time-minimum limit 85
5.1.8 Intrusion time interpolation and extrapolation 85
5.1.9 Material map input 85
5.1.10 Properties input 88
5.2 ASCII flux field input file (homogeneous) 100
5.3 ASCII flux field input file (heterogeneous) 101
5.4 NUTS output files 103

6.0 REQUIRED USER TRAINING AND/OR BACKGROUND 108

7.0 CAPABILITIES AND LIMITATIONS 109

8.0 USER INTERACTIONS WITH THE SOFTWARE 110

9.0 ERROR MESSAGES 113

10.0 REFERENCES 124

11.0 APPENDICES 125
11.1 Appendix A – ASCII input files for the CRA19 analysis 125

Information Only

11.1.1 ASCII input file for NUTS SCN run mode for scenario 2 125
11.1.2 ASCII input file for NUTS ISO run mode for scenario 2 127
11.1.3 ASCII input file for NUTS INT run mode for scenario 2 at 100 year intrusion time 129
11.2 Appendix B – NUTS's ASCII flux field input file 131
11.2.1 Example of an ASCII flux input file (homogeneous) 131
11.2.2 Example of an ASCII flux input file (heterogeneous) 132

Information Only

LIST OF FIGURES

Figure 4-1. Schematic diagram of fractured REV 31
Figure 4-2. Schematic diagram showing the REV and the fluxes $q*$ 33
Figure 4-3. Symmetric versus asymmetric dispersion fronts.....39
Figure 4-4. Point-center grid in finite difference approximation.....46
Figure 4-5. Block-center grid in finite difference approximation.....48

Information Only

LIST OF TABLES

Table 3-1. Major program unit in NUTS 12
Table 3-2. Block data modules in NUTS 12
Table 3-3. Function modules in NUTS 12
Table 3-4. Subroutines in NUTS 12
Table 3-5. Common block names used in NUTS 20
Table 3-6. Variables and arrays used in NUTS. In variables, "F" refers to the fracture and "M" refers to the matrix. F and M may appear in the beginning, in the middle or at the end. 21
Table 4-1. NUTS features used in the CCA calculations 29
Table 5-1. Variables' names to be printed in the output ASCII file 105
Table 5-2. Variables' names to be printed in the output CDB file when a NUTS input ASCII file sets any IPRNTMA variable to 1 for the compliance calculations, where $n = 1$ for AM241L, $n = 2$ for PU239L, $n = 3$ for PU238L, $n = 4$ for U234L, and $n = 5$ for TH230L 105
Table 5-3. History variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0 106
Table 5-4. Global variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0 106
Table 5-5. Element variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0, where $n = 1$ for AM241L, $n = 2$ for PU239L, $n = 3$ for PU238L, $n = 4$ for U234L, and $n = 5$ for TH230L 107
Table 8-1. Example to execute NUTS in SCN (screening) mode 112
Table 8-2. Example to execute NUTS in ISO (isotope) mode 112
Table 8-3. Example to execute NUTS in INT (intrusion) mode 112

Information Only

1.0 INTRODUCTION AND SCOPE

This Design Document and User's Manual for NUTS Version 2.08, describes the following major features of software design: software components, theoretical basis, mathematical model, control flow and logic, data structures, and functionality and the interfaces of subroutines. It also describes in detail the allowed or prescribed ranges for input data and user interface, in accordance with the Nuclear Waste Management Procedure NP 19-1 Software Requirements (Long, 2020). This code is used by the Sandia National Laboratories' Performance Assessment (PA) Department in support of the performance assessment calculations for the Waste Isolation Pilot Plant (WIPP). It is controlled under the software configuration management system CVS (Concurrent Versions System) on the Linux cluster.

1.1 Software Identification

Code Name:	NUTS
Version:	2.08
CVS Repository:	/data/cvs/CVSLIB/WIPP_CODES/PA_CODES/NUTS
Platform:	CentOS 7/Dell PowerEdge C6420
Host:	fwm.sandia.gov
Compiler:	Gfortran/GCC 8.2.0

1.2 Points of Contact

Code Sponsor:	Sungtae Kim	sunkim@sandia.gov	(575) 234-0001

1.3 Code Overview

NUTS is a multidimensional, multicomponent radioactive material contaminant transport, single-porosity (SP), dual-porosity (DP), and dual-permeability (DPM) finite-difference simulator. The model simulates first order radioactive chain decay during radioactive material transport. However, the simulator is not limited to radioactive material transport, and any non-radioactive material can be included. Three types of sorption isotherms are considered to represent ion exchange between the solute and the surrounding formation: linear, Freundlich, and Langmuir equilibrium isotherms. Hydrodynamic dispersion is modeled with the assumption that the off diagonal dispersivities are all zero. The solubility limits of the waste components, and their precipitation during migration are included in NUTS. The precipitate is allowed to undergo decay, and to redissolve in the brine if the concentration drops below the solubility limit. Multi-radioactive-site representations are also possible, in which case the contribution from each site to the component concentration and precipitation in each computational node can be found. A similar technique is used to handle the daughters generated from the decay of different parents. Many

Information Only

options for transport equation(s) discretization are included. In the implicit solution, the system of partial differential equations is solved sequentially to determine the contribution from parent radioactive material decay to the immediate daughter. In the sequential method, the solution proceeds progressively from the top of each radioactive material chain. Therefore, the contribution to any daughter from parent decay will be available. In addition, NUTS also accounts for thermal dependency of some properties. For the WIPP PA, NUTS calculates the radioactive decay, dissolution/precipitation, and isothermal transport of multicomponent radionuclides through the rock matrix in the 2-dimensional flared grid structure.

Information Only

2.0 REQUIREMENTS

2.1 Functional Requirements

The functional requirements for NUTS, Version 2.08 are listed below.

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.

2.2 External Interface Requirements

The external interface requirements are listed below:

R.12 NUTS utilizes routines from CAMDAT_LIB, CAMCON_LIB, and CAMSUPES_LIB. Consequently, it must be linked with these libraries.

R.13 NUTS requires an ASCII input file that contains control flags. Constituent and material properties can either be provided by NUTS or by a CAMDAT input file (R.16 below).

R.14 NUTS requires an ASCII input file which contains fluxes and geometrical information (if not provided by R.15).

Information Only

R.15 NUTS requires a CAMDAT input file which contains fluxes and geometrical information, if it is not provided by the ASCII input file of R.14. In this case an additional ASCII input file associated with the CAMDAT is also required to provide the material map and some other parameters.

R.16 NUTS requires a CAMDAT property input file which contains inventories, solubility's, half-lives, dispersion and sorption material properties, etc., if it is not provided by the NUTS ASCII input file of R.13.

R.17 In certain calculations associated with intrusion scenarios, NUTS requires a CAMDAT initialization input file. This file is the same as NUTS CAMDAT output file mentioned in R.20 below.

R.18 NUTS generates an ASCII output file for debug information and warning messages.

R.19 NUTS generates an ASCII output file containing an echo for the input in addition to the run results and an optional ASCII file contains spatial specific range of data.

R.20 NUTS generates a CAMDAT output file containing the run results.

2.3 Other Requirements

There is other requirement for NUTS.

R.21 Partial brine saturation (saturation < 100%)

2.4 Functionality Not Tested

The following features are not be tested nor to be used as part of the WIPP compliance calculations:

1) The ability to perform a three-dimensional transport simulation.

2) The ability to perform multi-continuum simulation. Because NUTS is designed to treat a broad spectrum of transport problems, it has a dual-permeability model so that it has the ability to simulate transport in a fractured porous media. This makes NUTS possible to run the model as dual-permeability, dual-porosity, single-porosity fracture, and single-porosity matrix. In compliance calculations only the single-porosity matrix option is used

3) The ability to model transport in the gas phase, which must consider gas solubility in brine as a retarding factor.

4) The ability to utilize Langmuir and Freundlich sorption isotherms.

5) The ability to calculate temperature dependency of solubility, molecular diffusion, and sorption equilibrium coefficients.

6) The ability to perform Multi-site (waste site) representation. NUTS' transport simulation can be contributed from radionuclides of each site to the component concentration and precipitation. This aspect is not used for the compliance calculation, and was therefore not tested.

Information Only

7) The ability to interface with a dynamic Actinide Source Submodel. This source is called by NUTS to evaluate, on a waste canister scale, corrosion, breaching, and transport from the canister to the porous media by diffusion. The transported material out of the canisters will act as a time-dependent source in NUTS.

8) The ability to read solubility of the radioactive materials from a table look-up as a function of CO_2 pressure and pH in each grid block. Also, the ability to read metal solubility as a function of Eh in each grid block.

9) The ability to handle a daughter generated from the decay of different parents.

10) Different options for time/space discretization, including mid-point weighting, split-operator, fully explicit, and the Crank-Nicolson method.

Information Only

3.0 MAJOR SOFTWARE COMPONENTS

NUTS consists of a set of core subroutines and functions that perform its calculations as related to the functional requirements. It is linked against the object libraries CAMDAT_LIB, CAMON_LIB, and CAMSUPES_LIB and calls upon subroutines and functions in those libraries to perform input/output operations on the CDB (CAMDAT Database) binary files, perform input/output operations related to QA records, and perform memory allocation.

3.1 Major Control Flow, Control Logic, and Subroutines

The major program unit, block data, functions and subroutines are listed in Table 3-1 through Table 3-4. Their functionalities and major control flow and logic are included in the descriptions.

Table 3-1. Major program unit in NUTS

Program name	Description
PROGRAM NUTS	Parses the command line arguments, prints QA records, opens the input and output CDB files, and calls subroutines to perform multidimensional, multicomponent radionuclide contaminant transport calculations.

Table 3-2. Block data modules in NUTS

Block data name	Description
SOMECONST	Specifies some constant values of SECPERYEAR, DISINTCONST, and AVOGADRO_NO.
VARDESCRPTN	Describes the fracture and matrix variables' names to be printed.

Table 3-3. Function modules in NUTS

Function name	Description
CORSOL	Calculates the solubility limit of the element from correlations.
GHAT	Generate a dynamic source to be used in NUTS for testing purposes. The output is a source in kg/sec.
CDBPROP	Extracts material property value from CAMDAT.

Table 3-4. Subroutines in NUTS

Subroutine name	Description
ADSORP	Calculates the equilibrium adsorption coefficient for the Linear, Freundlich, and Langmuir isotherms.
AREA_INTERFACE	Calculates the blocks' interface areas. Interface area between I, I-1; J, J-1; and K, K-1 are calculated and stored in I. The area is interpolated at the interface.
ARRAY13D	Switches the block number index from 1-D double precision array to 3-D single precision array.
ARRAY23D	Switches the block number index from 2-D double precision array to 3-D single precision array.

Information Only

ASC_WRITE	Write volumetric concentration, mass of the nuclide, mass of the precipitate, and concentration on soil basis in an ASCII file.
BANDIT	Conduct Gaussian elimination on the entries bounded by the upper and lower diagonals. The matrix need to be loaded after straighten up the diagonals and identify the location of each entry in the matrix.
BONDRY	Identifies the boundaries and the connectivity of each block. By this subroutine, it is possible to convert the calculation node indicies from I,J,K to I.
CONC_MATRIX	Evaluates the matrix entries and load the matrix into the band solver. A check for optimum matrix band width is provided, in this case, the subroutine will switch to minimum one.
CONCMB	Evaluates the material balance error for each transported component per time step. Also, the cumulative material balance error is calculated.
CONVERT	Interfaces with BRAGFLO. It reads from BRAGFLO binary file the time independent data such as dimensions and calculates the interblock area. Output data from BRAGFLO in an array of a dimension more than 1-D will be transformed into 1-D array for convenience.
CONVTEST	Inputs time dependent and independent data such as dimensions, fluxes, and geological data in cases when a run is wanted to be done without interfacing with BRAGFLO. This subroutine can be used to input information for verification or test purposes. Constant porosity, saturation, dimensions, fluxes, and time step are used consistent with the usual assumptions in the analytical solutions. The data regarding chemical component properties will be input in subroutine READ_RAD.
HETRO_CONVTEST	Inputs time dependent and independent data such as dimensions, fluxes, and geological data in cases when a run is wanted to be done without interfacing with BRAGFLO. This subroutine can be used to input information for verification or test purposes. Constant but heterogenous porosity, saturation, dimensions, fluxes.
DEBUGSUB	Identifies the beginning and the end of most of the subroutines in the program. This subroutine then writes the main variables in the identified procedure.
DECAYSOURCE	Calculates the decay coefficient of each component. Decay coefficient here is equal to (Retardation factor) × (Grid block volume) × (Radioactivity decay constant).
DECAY_SUM	Calculates the contribution to the daughter from the decay of its soluble and precipitated parent.
DSPRSN1	Calls subroutine DSPRSN2 to calculate the dispersion term. The values are calculated at I-1,J-1, and K-1 interfaces and stored at I
DSPRSN2	Calculates the dispersion term coming from the molecular diffusion and hydrodynamic dispersion. Transverse velocities are averaged in upstream fashion neglecting zero values. Longitudinal velocity is taken at the interface. The term (porosity) × (saturation), dispersivities, tortousities are weighted upstream.
DSPRSNMF	Calculates the matrix fracture dispersion term. Velocity of the transfer is evaluated as an average value of a transfer velocities in x,y,z directions. Matrix dispersivity is considered as longitudinal.
FLUX1	Evaluates the flux and any term of partial differential equation that is not component dependent. This flux matrix will be used to evaluate the concentration matrix later. The truncation error in this subroutine is first order correct.

Information Only

FLUX2	Evaluates the flux and any term of partial differential equation that is not component dependent. This flux matrix will be used to evaluate the concentration matrix later. The truncation error in this subroutine is second order correct.
FLUX3	Evaluates the flux and any term of partial differential equation that is not component dependent. This flux matrix will be used to evaluate the concentration matrix later. The method used is central finite-difference.
INITIALIZATION	Initializes concentration, precipitate, parent contribution to daughter, decay coefficient, precipitate decay, and adsorption coefficient at time n for each component. Also, the initial inventory will be distributed among the waste nodes. It is assumed that the brine will not hold more than its solubility limit. The rest will be precipitated. Saturation and porosity are also initialized.
OMEGA	Calculates the upstream weighting factors for the grid blocks. It takes a value of either "0" or "1". The flow is compared between I, I-1 in I-direction; I, J-1 in J-direction; and K, K-1 in K-direction and the flags are stored in I. Single index here refers to triple index, i.e., I refers to I,J,K. Weighting factor flag of "0" refers to flow out of I and "1" for flow into I. Similar criterion is used in J and K directions.
OMEGAMF	Calculates matrix fracture upstream weighting factors for Dual-Porosity/Dual-Permeability system. It takes a value of either "0" or "1". The fluid transfer between the matrix and the fracture is compared for each Phase. This depends on the transfer function formulation.
ONED3DINDEX	Generates the 1D indices to 3D indices.
OPTIM_DIMENSION	Compares number of blocks in x- and y-dimension and generates flags to be used to determine dimensionality of the problem and to optimize the band width. This is done to reduce the size of the band width in the matrix to be inverted and to automize the calculation. The standard dimension will be XYZ. Calculations will always be conducted in this sequence, however, matrix inversion will look for the smallest direction and therefore smallest bandwidth.
PRCIPDCY	Calculates the contribution from the precipitated parent to the immediate daughter. The amount of the decay is adjusted if it ceases somewhere within any time step.
PRECIPITATE	Assumes that any component having concentration greater than the solubility limit will precipitate out of the solution. If the concentration drops below the solubility limit, the component will dissolve again within the same limit too. Solid precipitate goes under decay. The amount decayed should be subtracted from the parent and added to the daughter in solution. In cases where there is contribution to the component from different waste sites, the precipitate will be treated linearly. Similar treatment is valid for daughter from different parents.
READ_RAD	Inputs the component properties and data required for dispersion, and adsorption. Injected component properties are also included.
MAT_MAP	Reads the time dependent material maps from the BRAGFLO ASCII input file.
PHYSPROP	Inputs the component properties such as solubility, radioactivity, molecular weight, initial inventory, and formation pH.
ADS_INPUT	Inputs the component adsorption properties and distributes the data among the grid blocks.
MATDSP_INPUT	Inputs dispersion parameters for matrix dispersion.
FRCDSP_INPUT	Inputs dispersion parameters for fracture dispersion.

Information Only

COMPMOL_DIFFUSION	Inputs the molecular diffusion of the components and reference temperature and viscosity for the temperature dependent molecular diffusion.
ROCK_DENSITY_INPUT	Inputs the rock density of each layer and distributes it among the grid blocks.
WASTE_MATRIX_INPUT	Inputs the waste density of each layer and distributes it among the grid blocks.
SOURCE_INPUT	Inputs the source/sink terms. This includes production, injection, and Dirichlet type boundary conditions.
READ_PROP	Reads material property and distribute the value among its corresponding grid blocks
READ_2D_PROP	Reads material property and distribute the value among its corresponding grid blocks
SOLUBILITYF	Calculates isotopes solubility based on the mole fraction of each isotope in the solution.
TCONVERT	Interface with BRAGFLO to read time dependent data such as velocities, porosities, and saturation. Interblock flow rates are also calculated. Any array of 3-D will be converted to 1-D.
CONV3DTO1D	Convert 3-dimensional array into 1-dimensional.
UPDATE	Updates the variables after each time step to set n time level values.
VELAVG	Takes the upstream average of the velocities in the transverse directions and appropriately account for the boundaries to avoid adding a velocity value of zero. Only two velocities will come into the subroutine with values greater than zero (upstream velocities). If any one of these velocities will have zero value, it will not be included in the average.
ZERO1D	Sets a 1-D array to zero.
ZERO2D	Sets a 2-D array to zero.
ZERO3D	Sets a 3-D array to zero.
CURIES_VALUES	Calculates the curies for the printout variables.
XYZCOORD	Evaluates x,y, and z coordinates of each grid block.
STOTALMAS	Evaluates the total mass of nuclide. The total mass of nuclide represents the sum of dissolved mass, mass of the precipitate, and mass of the adsorbed nuclide.
NTOTALMAS	Evaluates the total mass of all nuclides in each grid block.
CNTOTALMAS	Evaluates the total curies of all nuclides per grid block.
BWR13D	Converts the 1D variable into 3D variable and writes it in a binary file as single precision.
BWR23D	Converts the 2D variable into 3D variable and writes in BINARY file as single precision
BWR33D	Converts the 3D variable into single precision 3D and writes it in BINARY file.
AWR13D	Converts the 1D variable into 3D variable and writes it in an ASCII file as single precision
AWR23D	Converts the 2D variable into 3D variable and writes in an ASCII file as single precision
AWR33D	Converts the 3D variable into single precision 3D and writes it in an ASCII file.
CORSOLUBILITY	Calculates the solubility limit of the element from correlations by calling function CORSOL.
MOLDIFTEMP	Evaluates temperature dependency of the molecular diffusion.
SW1_CONC_MATRIX	Preprocesses the matrix before inversion. A check for optimum matrix band width is provided. The subroutine will switch to minimum one before loading the matrix in the band solver. This subroutine is used for first order method.

Information Only

SW2_CONC_MATRIX	Preprocesses the matrix before inversion. A check for optimum matrix band width is provided. The subroutine will switch to minimum one before loading the matrix in the band solver. This subroutine is used for second order method.
ISOTOPID	Evaluates the number of component (isotope) in each element class and identify their location in the input. This is done to distribute the solubility and its limit between the corresponding isotopes.
CONCURIES	Evaluates the conversion constants of the radioactive decay in curies/Kg.
DECAYCONSTANT	Evaluates decay constants of the radioactive material. Accordingly, radioactive material is labeled as RADIOACTIVE.
COMPSLID	Identifies the immediate parent of each component and its location in the input data. This is done to track each component separately and to allocate the solubility and the precipitation.
FLAG	Inputs flags to control the calculations in NUTS.
SITE_FLAG	Inputs disposal sites information and solubility flags.
MATADS_FLAG	Inputs flags to control the calculations of matrix adsorption.
FRCADS_FLAG	Inputs flags to control the calculations of fracture adsorption.
KDTEMPDEP_FLAG	Inputs flags and kd temperature dependent parameters
MATDSP_FLAG	Inputs dispersion flags controlling matrix dispersion.
FRCDSP_FLAG	Inputs dispersion flags.
MATFRCDSP_FLAG	Inputs flags controlling fracture/matrix dispersion.
MOLDIFTEMDEP_FLAG	Inputs flags controlling temperature dependency of molecular diffusion.
MATSRC_FLAG	Inputs flags controlling injection/production source and Dirichlet boundary conditions if any in the matrix.
FRCSRC_FLAG	Inputs flags controlling injection/production source and Dirichlet boundary conditions if any in the fracture.
MATPRINT_FLAG	Inputs flags controlling the type of matrix variables that can be printed in the ASCII,BINARY, and CDB files.
FRCPRINT_FLAG	Inputs flags controlling the type of fracture variables that can be printed in an ASCII and/or BINARY files.
PRNTFREQ_FLAG	Inputs flags controlling the type of matrix variables that can be printed in an ASCII and/or BINARY files.
EXTSRC_FLAG	Inputs a flag to identify whether there is an external source model. NUTS expects to have all the nuclides information from the source model, and then subroutine READ_RAD will be bypassed.
ZEROLIMIT	Regarding the time step, it is noticed that because of the precision, zero time step may be calculated from subtracting T2 from T1. The actual value may be fraction of a second. This value is set to very tiny number to avoid division by zero.
INTRUSION_TIME_SUB	Inputs the times related to intrusion interpolation-extrapolation in time.
SNKSRC	Adjust time step size (DT) with the starting and ending of sink/source term.
TRUNCATE	Adjust the length of the character to concatenate with other character.
MBMAS	Evaluates the total mass per grid block resulted from the sum of dissolved mass, mass of the adsorbed nuclides, and the retarded gas in case of gas transport.
BLOCKMAS	Evaluates the mass per grid block of dissolved and precipitated nuclide.
BRAGCONV	Rename the single precision BRAGFLO 3-d array.
RENAMECOMP	Rename the single nuclides for comparison purposes.
MASSCURIES	Calculates masses and curies in the grid blocks.
CONTINUUM	Specifies the type of the porous media and in which continuum the calculations should be conducted.

Information Only

PROP_SWITCH	Change time dependent material properties according to the material maps.
UNITCONVERT	Converts the solubility limit from moles/m^3 to kg/m^3 and converts inventory from Curies to kg.
PRGM_INFORM	Specifies program information.
PRNT_INFORM	Specifies number, name, description, unit conversion, and units of the element and global variables to be printed.
INBRAG_BIN_WRITE	Writes a binary file to be post-processed by the same post-processor of BRAGFLO.
BOUNDJM1_FLUX	Evaluates the flux crossing block boundary. It consists of advective flow and dispersion contributions from both matrix and fracture. This subroutine is for the first order fully implicit method.
BOUNDJM2_FLUX	Evaluates the flux crossing block boundary. It consists of advective flow and dispersion contributions from both matrix and fracture. This subroutine is for the second order fully implicit method.
BOUNDJM3_FLUX	Evaluates the flux crossing block boundary. It consists of advective flow and dispersion contributions from both matrix and fracture. This subroutine is for the second order with split operator technique.
BOUNDJM4_FLUX	Evaluates the flux crossing block boundary. It is used in conjunction with generalized Crank-Nicolson.
CONDENSATION	Evaluates the equilibrium constant (K value). Experimental K values are used as follows: $$\frac{C_t^*}{C_t} = A + BT$$ where, C_t^* is the concentration of carbon ions in the liquid phase at equilibrium, C_t is the concentration of carbon ions in the gas phase at equilibrium, A is intercept, B is the slop, and T is the temperature. Pressure and composition dependency of K values is neglected (Amter, 1988).
SRCMOD_YTOTVEL	Evaluates the total velocity of the matrix and the fracture for the flux crossing the waste cans. This subroutine is required for source calculation.
TIME_PRECIP	Evaluates the amount of the precipitated nuclide for each time step.
IND_INTERPOLATE	Brackets the location of the ROWIN (x argument of the interpolated function $F(x,y)$) and COLIN (y argument of the interpolated function $F(x,y)$) in the table for further two dimensional linear interpolation.
IND_EQUAL_INTERPOLATE	Brackets the minimum row and column entries. Also equally spaced rows and column differences are found.
STEPWISE_INTERPOLATE	Generate step-wise linear interpolation functions. This function has a property of having a value of 1 at a certain node and zero everywhere. To interpolate in x, y dimension, four of such interpolation functions are required, two in each dimension.
SOLB_INTERPOLATE	Evaluates element solubility from table look-up.
AWRNG13D	Converts the 1D variable into 3D variable and write it in an ASCII file as single precision for specific range.
RANGE13D	Switches the block number index from 1-D double precision array to 3-D single precision array for specific range.
RANGE23D	Switches the block number index from 2-D double precision array to 3-D of single precision array.
AWRNG23D	Converts the 2D variable into 3D variable and write in an ASCII file as single precision.
RASC_WRITE	Write volumetric concentration, mass of the nuclide, mass of the precipitate, and concentration on soil basis in an ASCII file for specified ranges.

Information Only

DSPRSN3	Sets the molecular diffusion to zero for impermeable interfaces.
BRAG_INPUT_CHECK	Evaluates BRAGFLO input at time = 0 and verify the validity of the data range used.
TEST_INPUT_CHECK	Evaluates TEST file input at time = 0 and verify the validity of the data range used.
NUTSG_INPUT_CHECK	Evaluates NUTS file general input (apply for both fracture and matrix) at time = 0 and verify the validity of the data range used.
NUTSM_INPUT_CHECK	Evaluates MATRIX PROPERTIES in NUTS input file at time = 0 and verify the validity of the data range used.
NUTSF_INPUT_CHECK	Evaluates FRACTURE PROPERTIES in NUTS input file at time = 0 and verify the validity of the data range used.
INITF_CONC_MAN	Inputs manual initialization flag for concentration in the fracture.
INITM_CONC_MAN	Inputs manual initialization flag for concentration in the matrix.
CDB_INITM_CONC_FLAG	Initializes and reads the flag for the concentration initialization from the CDB. It is required for intrusion times other than 350, 1000 yrs.
CONC_INIT_INPUT	Initializes the concentration manually from the input.
RESET_INTRUSION_TIME	Resets the intrusion time at values different from the standard values (350, 1000) in BRAGFLO. It is required for the intrusion of 100 yr interpolated from the real 350 yrs in BRAGFLO. Also for 3000, 5000, 7000, 9000 interpolated from BRAGFLO intrusion time of 1000 yrs. In these cases, the transport parameters are initialized from NUTS Undisturbed values and the parameters from BRAGFLO are reset at the new time of intrusion.
IND1D_INTERPOLATE	Brackets the location of the ROWIN1D (input value of the interpolated function F(x)) in the table for 1D linear interpolation.
STEPWISE1D_INTERPOLATE	Generates step-wise linear interpolation function. This function has a property of having a value of 1 at a certain node and zero everywhere. To interpolate in one dimension, two of such interpolation functions are required.
IND1D_EQUAL_INTERPOLATE	Brackets the minimum row entry. Also equally spaced row differences are found.
SCALEUP_COLLOID	In this effort colloid transport is assumed to flow at maximum velocity available in the velocity profile. For a laminar flow in a pipe, the velocity profile is parabola and $v_{average} = \frac{2}{3} v_{max}$, therefore the scaling factor is $\frac{2}{3}$. However in the porous media, velocity profile is not that regular and the scaling factor could be different. This subroutine inputs the scaling factor that should be multiplied by the velocity to account for the fact that the colloid is transported faster than the dissolved material.
SCALE_FACT_COLLOID_INPUT	Inputs the scale factors for the velocities in x, y, and z directions required for colloid transport.
SOILBASECON	Evaluates the soil base concentration. Noticed that the same density is used to calculate for both fracture and matrix. Brine density in the fracture is not available now.
CONC_CHECK	Checks the bound for the concentration in second-order method. This is done because two-point upstream weighting involve extrapolation process and may not conserve the physics of the problem.
CONCMB2	Evaluates the material balance error for each transported component per time step in second order method. Also, the cumulative material balance error is calculated.
RHSEXCONT	Evaluates the flux contribution to the right hand side of the partial differential equation in case when split operator two point upstream weight is used. In

Information Only

	this technique, the contribution from the first derivative will be added explicitly to the equation.
CRANK1	Prepares the entries required to solve the system of equations implicitly.
FIRST_CRANK2	Evaluates the explicit concentration required for generalized Crank-Nicolson technique in first order methods.
SECOND_CRANK2	Evaluates the explicit concentration required for generalized Crank-Nicolson technique in second order methods.
TIMESOURCE	Identifies the gridblock and the nuclide that have a time dependent source.
BOUNDJM5_FLUX	Evaluates the flux crossing block boundary. It consists of advective flow and dispersion contributions from both matrix and fracture. This subroutine is for central difference scheme.
FLUXADJUST	Evaluates the flux crossing grid block boundary. If the input is from BRAGFLO, this subroutine will compare the saturation with the residual saturation. The flow from the block will be set to zero if its saturation is < the residual saturation. This step is done as precaution in case where the fluxes in BRAGFLO are not updated after the convergence from newton iteration is satisfied.
REALCONV	Convert real*4 into real*8.
PRC_CONVERGENCE	Evaluates the convergence in the rate of change in the precipitate.
IMPPRSRC1	Evaluates the amount of precipitate need to be used as a source in each grid block if Dirichlet condition is applied. It calculates the material balance around the grid block without the source and then evaluate the amount of the source required to maintain Dirichlet condition.
IMPPRSRC2	Evaluates if the grid block has enough precipitate to maintain Dirichlet condition. If not, the amount of the precipitate left will be used as a source. Concentration exceeding solubility limit will also be checked.
SRC_RHSADJ	Adjusts the right hand side during the iteration process in treating the precipitate as an implicit source.
MRHSADJ	Adjusts the right hand side during the iteration process in treating the precipitate as an implicit source.
GRID_WITH_PRECIP	Identifies grids with precipitate, save the old amount precipitate, and initialize some variable.
DISSOLVED_MASS	Evaluates isotope mass dissolved in the brine.
DECAY_SUM_IM	Calculates the contribution to the daughter from the decay of its soluble and precipitated parent in the precipitate fully implicit treatment.
ALONEDECAY	Evaluates the decayed mass from analytical expression.
OLDCONCDCY	Evaluates the total mass in the computational domain.
IMP_PRECIP	Evaluates the precipitated mass in the implicit treatment mainly due to the ingrowth in the daughter component.
CDB_OPENFILES	Reads the command line arguments
CDB_HEADER	Reads the header information for the CDB
INITIALIZE_CDB	Gets model sizes from input CDB and reads isotope data from CDB header
CDBOATTR	Takes a double precision vector of attribute values and applies to write them to the CDB with the given name.
CDBOVAR	Writes time-dependent variables to the CDB.
CDB_READ	Reads time-dependent values needed by NUTS from the CDB.
CDB_MAP	Checks the number of elements and dimensions.
CDB_VARNAMES	Alerts CAMDAT that there will be new time dependent variables and also instructs CAMDAT to copy over the old ones from the input CDB
CDB_WRITE	Writes results from NUTS to the opened CDB file. This routine replaces BIN_WRITE in NUTS for compatibility with CAMCON

Information Only

NUTS_STOP	Allows NUTS to use CAMCON - QAABORT abort execution.
BIN_READ	Calls CDB_READ to read BRAGFLO output binary file. The data read are mainly fluxes, porosities and saturations in addition to some QA required information. NUTS requires only the velocities at the interfaces, porosities, saturations, and production/injection rates at each grid block. The rest of the data is not important to NUTS.
BIN_WRITE	Write volumetric concentration, mass of the nuclide, mass of the precipitate, and concentration on soil basis in a binary file.
INOUTFIL	Calls subroutine CDB_OPENFILES to perform QA functions as well as to read the command line and open necessary I/O files.
FILE_OPEN	This does not have any functionality.
VAX_TIME	Evaluates the CPU run time and read date and time of the run.

3.2 Data Structures

NUTS uses standard FORTRAN variables, arrays, common blocks, and fixed constant parameters. The common blocks that are used in NUTS and are defined in 'inc' files are listed in Table 3-5. Variables and arrays are listed in Table 3-6.

Table 3-5. Common block names used in NUTS

inc file name	Common block names
cdbxfer.inc	CDB_VARIABLES_COMMON, REAL_VECTOR_COMMON, CHARACTER_VECTOR_COMMON, READ_VECTOR_COMMON, TEMP, TEMP11, TEMP12, CDB_TO_NUTS_GRID_MAP, CDBVRNAMS, CDB_HISTORY_OVERRIDE,
common.inc	ADSTEMPDEPEND, ASCPRINTFLAGS, BCONDMAS, BINCDBBRAG, BINPRINTFLAGS, BLOCKMASS, BOUNDARY, BRAGHVAR, BRAGMAP, BRAGPFLUX, BRAGPHPCO2, BRINEVOL, BTITLE, BWP1POINTERS, BWP2POINTERS, CDBPRINTFLAGS, COMPNAME, CONCF, CONCM, CONCSOURCE, CONDEQCON, COORDINATES, CORSOLPAR, CTIME, CURGRIDMASS, CURIES, DCONSTANT, DEBUGG, DIM, DIRECT, DIRICHLET, DISPCONTROL, DISSYMETRY, DOLOOPCONC, EDATETIME, FABANDS, FFABANDS, FABLOCRNGPRNTFLG, FADSDECAY, FADSDSTCF, FADSTYPE, FBBANDS, FCONCMATBERR, FCURIESOUTPUT, FDIFFLAG, FDISPM, FDISPPROP, FDISPV, FILENM, FMDISP, FMFLUXIM1, FMFLUXJM1, FMFLUXKM1, OLDFFLUX, OLDMFLUX, FMSSTIME, FMYTOTVEL, FNAME, FPRECIPTFLAG, FRACPROP, GASBANDS, GASEQCONS, GLPHASETYPE, GRIDMASS, GSIZE, HEADINGS, HGUNDES, HGVARBINPR, IDENTIFY, INDECES2D, INDECES3D, INOUT, LIMITS, LMAX, LOGSOL_INV, MABANDS, MMABANDS, MABLOCRNGPRNTFLG, MACHINE, MADSDECAY, MADSDSTCF, MADSTYPE, MATERIALS, MATFRACTRANS, MATPROP, MATRIX, MBBANDS, FFBANDS, MMBANDS, MBOLDNEWMASS, MCONCMATBERR, MCURIESOUTPUT, MDIFFLAG, MDISPM, MDISPPROP, MDISPV, MISC, MPRECIPTFLAG, NAMES, NBNC, NOFILE, OLDF, OLDM, OLDPRECIP, OMEGAFM, OUTBINARY, OUTFNMT, OUTPUTVARDES, PHPCO2, PORMEDIA, PORSATFLOW, PRINTFREQUENCY, RADSITE, RANGEOUTPUT, RANGINDX, REFERENCE, REF1POINTERS, REF2POINTERS, REGPROP, RTPDENCOMP, SDATETIME, SIZE, SOLBLIMIT, SOLBTAB, SOLUBILITY, SW1POINTERS, SW2POINTERS, TABLES, TABLESINDECES, TEMPDMOL, TEMPERATURE, TEMPSOLB, TOTALMAS, VEL, VISCOSITY, INFOSOURCE, INFOINIT, FLVEL, WASTETEST, DIGNOSTICFILE, TESTFILEINPUT, MANINITIAL, EQSPACEDXDY, TABLES1D, EHBRAG, EHFRACMAT, PRESSDEN, COLLOIDTRANS, CDBFLAG, TRUNERROR, EXPLICITRHS, GCRANKNICLSON, DFCRANK, DMCRANK, TDSOURCE, FSRCTMECOR, MSRCTMECOR, RESDSAT, PREFSOLB, CONSTANTSSS, INTRUSIONTIME, CDBINITIAL, IMEXPRECIP, DIAGADJUST, RIMPPRECIP, LIMPPRECIP, ORIGNPRECIP,

Information Only

Table 3-6. Variables and arrays used in NUTS. In variables, "F" refers to the fracture and "M" refers to the matrix. F and M may appear in the beginning, in the middle or at the end.

Variable	Description
A	Concentration matrix
A1	I,J,K-1 diagonal contribution to the conc. matrix
A2	I,J-1,K diagonal contribution to the conc. matrix
A3	I-1,J,K diagonal contribution to the conc. matrix
A4	I,J,K diagonal contribution to the conc. matrix
A5	I+1,J,K diagonal contribution to the conc. matrix
A6	I,J+1,K diagonal contribution to the conc. matrix
A7	I,J,K+1 diagonal contribution to the conc. matrix
ADPRCON_TOTAL_MASS	Total mass resulted from dissolved, precipitated and adsorbed nuclides
ADSORP_COEFF	Adsorption Coefficient at *n+1* time level
ADSRP_COEFF_OLD	Adsorption coefficient at *n* time level
ADSTYPE	Flag to identify whether adsorption is considered (Y/N)
ALPHAL	Longitudinal dispersivity
ALPHAT	Transverse dispersivity
ALPL3D,ALPL3D	Longitudinal and transverse 3D dispersivities
ANSWERTEST	Character flag to read from a test file
AX	Interblock area in *x*-direction (I,I-1)
AXISFLAG	Flag to indicate whether xy axes are switched
AY	Interblock area in *y*-direction (J,J-1)
AZ	Interblock area in *z*-direction (K,K-1)
B	Right-hand side of the matrix or the solution vector
B1	I,J,K-1 flux
B2	I,J-1,K flux
B3	I-1,J,K flux
B4A1	I,J,K flux – *n+1* time level contribution from RHS
B4A2	Matrix flux (Transfer) contribution
B4B	Contribution from *n+1* time level adsorption from RHS
B5	I+1,J,K flux
B6	I,J+1,K flux
B7	I,J,K+1 flux
BLOCF_DIS_MASS	Fracture mass of dissolved nuclide per grid volume (kg/m^3 rock)
BLOCM_DIS_MASS	Matrix mass of dissolved nuclide per grid volume in (kg/m^3 rock)
BLOCF_PRC_MASS	Fracture mass of precipitated nuclide per grid volume (kg/m^3 rock)
BLOCM_PRC_MASS	Matrix mass of precipitated nuclide per grid volume (kg/m^3 rock)
BRAGNAMBIN	BRAGFLO binary output file name
C	Concentration at *n+1* time level (kg/m^3 solution)
CC	Mass of component precipitated (kg)
CMBI	Nuclide Material balance error/timestep
CMBT	Nuclide cumulative material balance error
COLD	Concentration at *n* time level (kg/m^3 solution)
CONCINJ	Concentration of the injected component (kg/m^3 solution)
COMPINT	Initial inventory of the component (kg)
COMPONENT_NAME	Component name
CUMMASS	Total mass resulted from dissolved, precipitated and adsorbed nuclide
Daughter_NAME	Component daughter name

Information Only

DEBUG	Logical flag to debug the program
DEBUGASC	NUTS debug ASCII output file name
DECAY	Decay coefficient
DELT	BRAGFLO timestep duration
DIRECTION	The direction in which the calculation should be conducted
DIR1,2,3	Flag to involve x, y, z in the calculation
DIS_MASS_CURIES	No. of Curies in the dissolved nuclide
DISPMF	Fracture/matrix dispersion
DISPX	Dispersion coefficient in x-direction (between I,I-1)
DISPY	Dispersion coefficient in y-direction (between J,J-1)
DISPZ	Dispersion coefficient in z-direction (between K,K-1)
DISPREQ	Character flag to identify whether dispersion is considered or not.
DMOL	Molecular diffusion at reference temperature (m^2/s)
DMOLTEMDEP	Logical flag to identify whether the molecular dispersion is temperature dependent (T/F)
DT	Timestep duration
DTIN	Input DT for the source
DTOUT	Adjusted DT after the source
DX	Grid length
DXGRID	BRAGFLO grid block length in x-direction
DXTEST	Input grid block length in x-direction from test file
DY	Grid width
DYGRID	BRAGFLO grid block length in y-direction
DYTEST	Input grid block length in y-direction from test file
DZ	Grid thickness
DZGRID	BRAGFLO grid block length in z-direction
DZTEST	Input grid block length in z-direction from test file
ELEMENT_NAME	Element name
ELSOLB_LIMIT_COR	Element solubility limit (kg/m^3) of a block
ELTEMP_SOLB	Logical variable to identify if the element solubility is temperature dependent
FDISPREQ	Flag to identify whether there is dispersion in the fracture (Y/N)
FINFILETYPE	Type of output file:ASC,BIN,ASC-BIN
FRAC_ISOTHERM	Type of the adsorp. isotherm in the fracture L/l = Linear, F/f = Freundlich, LA/La/la = Langmuir
FSORPTION	Flag to identify whether the component is adsorbable in the fracture (ADSORP/NON_ADSORP)
FSTATUSINJ	Flag to identify whether there is injection in the fracture (Y/N)
GROUP_NAME	Group under which the isotope is classified i.e.: U for U235,238, etc.
HALF_LIFE	Component half-life (s)
IBW	Matrix bandwidth
IBWS	Switched matrix bandwidth
ICOUNT	Number of components
IDIMENSION	Number of dimensions
IDISOTOPE	Relate the component to its element
IDSIMCOMP	Location of the component in the input
IDP	Location of the parent in the input
IFILE	Flag to write in ASCII or binary file (0 = binary, 1=ASCII)

Information Only

IFLAGTIME	Time flag, 0 = time independent data is read from BRAGFLO 1= time dependent data is read from BRAGFLO
IJMAX	Number of blocks in *xy* dimension
IM	Half band width of the matrix
INJBLOKN	Grid block number that has injection
IPRFRQA	Frequency of print in ASCII file (ex, 5 = each 5 timesteps)
IPRFRQB	Frequency of print in BINARY file (ex, 5 = each 5 timesteps)
IPRNTFA	Flag to print the fracture variable in ASCII file
IPRNTFB	Flag to print the fracture variable in binary file
IPRNTMA	Flag to print the matrix variable in ASCII file
IPRNTMB	Flag to print the matrix variable in binary file
IRANGE	Range number
IPRECIP	Character flag for precipitate initial or time dependent precipitation
IRATECONTROL	Flag to start or stop injection/production
ISOTOPLOC	Location of the component in the input
ISOTHERM	Adsorption isotherm kind.
IT	Iteration index of time loop
IWASTE	Waste matrix, (1 = waste node, 0 = no-waste node)
J1	Flag refers to I-1
J2	Flag to control the J-1 boundary conditions
J3	Flag refers to I+1
J4	Flag refers to J+1
J5	Flag refers to K+1 and to control K+1 boundary conditions
J6	Flag refers to K-1 and to control K-1 boundary conditions
J7	Flag refers to I-1,J+1 block
J8	Flag refers to I-1,K-1 block
J9	Flag refers to I-1,K+1 block
J10	Flag refers to J-1
J11	Flag refers to I-1,J-1 block
J12	Flag refers to I+1,J-1 block
J13	Flag refers to J-1,K+1 block
J14	Flag refers to J-1,K-1 block
J15	Flag refers to I+1,K-1 block
J16	Flag refers to J+1,K-1 block
J17	Flag to control J+1 B.C'S
J18	Flag to control I+1 B.C'S
J19	Flag to control I-1 B.C'S
JASC	NUTS ASCII output file unit number
JBIN	NUTS binary output file unit number
JDBUG	Debug output file unit number
JIN	READ_RAD input file unit number
JOUT	INIT_WRITE output file unit number
JTEST	Test file (CONVTEST) input file unit number
LOCINP	Location in the input of the injected component i
LTNODWASTE	Total number of the grid blocks having component i
MAT_ISOTHERM	Type of the adsorp. isotherm in the matrix L/l = Linear, F/f = Freundlich, LA/La/la = Langmuir
MDISPREQ	Flag to identify whether there is dispersion in the matrix (Y/N)

Information Only

MFDISPREQ	Flag to identify whether there is dispersion in the matrix/fracture transfer (Y/N)
MSORPTION	Flag to identify whether the component is adsorbable in the matrix (ADSORP/NON_ADSORP)
MSTATUSINJ	Flag to identify whether there is injection in the matrix (Y/N)
NAME	Component name
NBLOCK	Number of grids
NCOMPONENT	Number of the components in each site
NDENSRANGE	Number of rock density ranges (layers).
NDSPNRANGE	Number of dispersion ranges.
MEDIUM	Type of the porous medium (either fracture or matrix)
NOCONTINUM	Number of continuum
NODWASTE	Total number of grid blocks having waste.
NOELEMENT	Number of the input elements
NO_PHASES	Number of phases
NSIMCOMP	Number of similar components in the input
NSITES	Number of radioactive sites
NTBLOKINJ	Total number of blocks with injected component *i*
NX OR IMAX	Number of grid blocks in *x*-direction
NY OR JMAX	Number of grid blocks in *y*-direction
NZ OR KMAX	Number of grid blocks in *z*-direction
PARENT_NAME	Component's parent name
PH	pH of the ground water
PHIFINT	Input fracture porosity
PHIMINT	Input matrix porosity
PHREQ	Flag to identify PH dependency (Y/N)
POR	Porosity at *n+1* time level
POROLD	Porosity at *n* time level
PORT	Total porosity at *n+1* time level
POROLDT	Total porosity at *n* time level
PRDECAY	Precipitate decay
PRECIP	Precipitate
PRCIP_MASS_CURIES	Mass of the Precipitate (*Ci*)
QIN,QININPUT	Injection rate
QPR,QPRINPUT	Production rate
QWX	Water interblock rate in *x*-direction (between I and I-1)
QWY	Water interblock rate in *y*-direction (between J and J-1)
QWZ	Water interblock rate in *z*-direction (between K and K-1)
RAD	Character to identify the component whether radioactive or not
RADINPUT	READ_RAD input file name
RADOUTDBG	NUTS ASCII debug output file name
RADOUTPUT	INIT_WRITE ASCII output file name
RADOUTBIN	NUTS output binary name
RADOUTASC	NUTS output ASCII file name
RALPHAL,RALPHAT	Longitudinal and transverse range dispersivities
RESD	Residual/timestep
RH	Right hand side of Partial Differential Equation (PDE)
RHORANGE	Rock density range (kg/m^3)
RHOSOLID	Rock density (kg/m^3)
RTOR	Tortuosity range

Information Only

SIGMA	Shape factor
SITE_NAME	Name of the disposal site. SNAME and SITE_NAME are used interchangeably
SNAME	Name of the disposal site. SNAME and SITE_NAME are used interchangeably
SOLB_LIMIT	Solubility limit
SSOLB_LIMIT	Solubility limit (kg)
SORPTION	Character to identify adsorbable components
STATUSINJ	Character flag to identify whether injection is considered or not
STOCKMAN	Logical flag to identify whether the source
SUMQ	Summation of nuclide all sources/sinks/timestep
SUMQT	Cumulative nuclide sources/sinks
SUM_PARENTS	Contribution to daughter from parents' decay
SUMRT	Cumulative residual
SW	Water saturation
SWFINT	Input fracture saturation
SWMINT	Input matrix saturation
SWOLD	Saturation at n time level
TAUW	Total fracture/matrix transfer of water
TDMOL	Temperature dependent molecular diffusion
TEMP_FRAC	Fracture temperature
TEMP_MAT	Matrix temperature
TESTFILE	Test file input file name
TIME	BRAGFLO time
TIMEASCMAX	Maximum time between timesteps allowed not to print in ASCII file
TIMEBINMAX	Maximum time between timesteps allowed not to print in binary file
TIMEEND	Ending time of injection/production
TIMELOOP	Total time
TIMESTRT	Starting time of injection/production is supplied by STOCKMAN
TITLE	Title of the run.
TNINJCOMP	Total number of injected components.
TORM,TORF	Matrix and fracture tortuosity
TOR3D	3D tortuosity
TOTAL_MASS_CURIES	Total mass of dissolved + adsorbed + Precipitate (*Ci*)
TREF	Reference temperature
VELWX	Water interblock velocity in *x*-direction (between I and I-1)
VELWY	Water interblock velocity in *y*-direction (between J and J-1)
VIS	Viscosity at temperature T
VISREF	Viscosity at reference temperature TREF
VISW_FRAC	Fracture water viscosity
VISW_MAT	Matrix water viscosity
VOL_CONC_CURIES	Volumetric concentration (*Ci*)
VR	Grid volume
WASTEVOL	Site's waste vol (m^3)
WX	Weighting factor in *x*-direction
WY	Weighting factor in *y*-direction
WZ	Weighting factor in *z*-direction
TVELX	Input interblock velocity of water phase in *x*-direction from test file
TVELY	Input interblock velocity of water phase in *y*-direction from test file
TVELZ	Input interblock velocity of water phase in *z*-direction from test file
UNITSNO	BLOCK DATA to specify files' unit numbers

Information Only

VELWZ	Water interblock velocity in *z*-direction (between K and K-1)
WWATMF	Fracture/matrix water transfer upstream weighting factor
XFC	FREUNDLICH coefficient (dimensionless).
XFDC	FREUNDLICH distribution coefficient (m^3 fluid/kg solid)
XL	Linear adsorption coefficient (m^3 fluid/kg solid)
XLAMDA	Decay constant of the component (s^{-1})
XLC	Langmuir coefficient (dimensionless)
XLDC	Langmuir distribution coefficient (m^3 fluid/kg solid)
XMOLEFRACTION	Mole fraction
XMOLWT	Molecular weight
XNOMOLE	Number of moles

Information Only

4.0 MODELS AND METHODS

4.1 Background

4.1.1 NUTS's general role in the PA

NUTS's principal 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 closed 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 radionuclides that are assumed to reside in the repository at the time of closure plus any progeny of those radionuclides that may have been produced through natural decomposition (by decay). 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, BRAGFLO.

In the 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 closure. On these time scales, radionuclides 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 closure onward to 10,000 years. NUTS does that.

NUTS has a single principal area of application. Given the defining characteristics of the flow of uncontaminated brine into the repository's waste panels, NUTS is required to model the dissolution and sorption of radionuclides, and to calculate, based on the given fluid flow rates, the rate at which each radionuclide 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 analyzes 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

Information Only

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. 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 either ALGEBRACDB or PANEL.

Because of natural decay processes, it is entirely possible to find compounds in the effluent that were not present in the initial repository inventory.

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 compliance calculations for the sake of simplicity. Compliance-disabled options are described briefly herein and are clearly noted as disabled for compliance calculations.

The model simulates radionuclide 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 compliance applications, only radionuclides 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) was 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. It is described in Section 4.5.1.**

NUTS normally models hydrodynamic dispersion under the assumption that the porous media through which the transport occurs are dispersively isotropic. However, for compliance calculations, both molecular and mechanical dispersion are taken to be zero. The effects of molecular dispersion are truly small (O(10m)). Mechanical dispersion is proportional to velocity and other physical and geometric factors (see Section 4.2.3). Ignoring transverse dispersion tends to keep solutes in the higher-velocity regions of the flow field. Ignoring lateral dispersion ignores streamwise elongation, which is reasonable given that EPA limits are defined in terms of time-integrated measures. For example, when and how a signal reaches the land-withdrawal boundary are unimportant, as long as the signal reaches the boundary in 10,000 years. Numerical dispersion and statistical dispersion (due to sampling) are both present in NUTS results, and mimic physical dispersion.

"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.

Information Only

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 PA, and it is therefore not tested.**

The system of governing partial-differential equations (mass conservation for each radionuclide) 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 compliance 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 4-1 provides a summary for the features used in the compliance calculations.

Table 4-1. NUTS features used in the CCA calculations

NUTS Features	Features in the compliance calculations	
	Used	Disabled
Single-porosity, fracture		X
Single-porosity, matrix	X	
Double porosity		X
Double permeability		X
Advective transport	X	
Diffusive dispersive transport		X
Sorption		X
Colloid transport*		X
Decay	X	
Gas phase transport		X
Temperature dependency		X
Multiple sites		X
Precipitation	X	
Solubility limit	X	
Interior sources	X	

* Colloids are lumped with dissolved inventory (see Section 4.5.1)

Information Only

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 a1, 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.

Thus, fractured porous media are typically conventional sedimentary materials that are possessing discontinuities introduced later by physical activity. Digenesis 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

Information Only

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, 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.

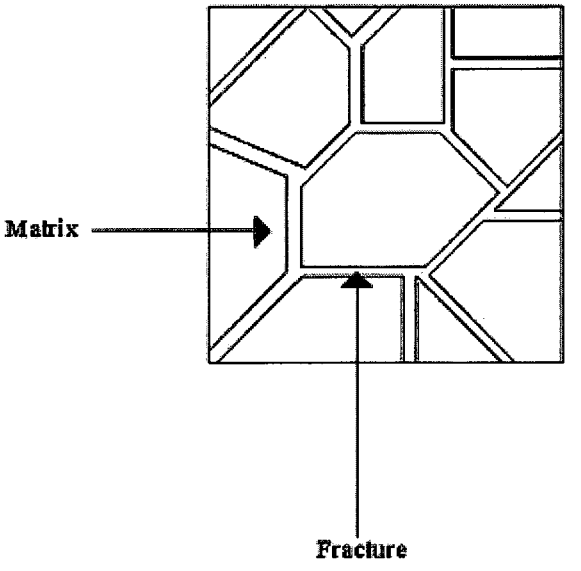


Figure 4-1. Schematic diagram of fractured REV

4.2.2 Basic dissolution processes

When brine 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,

Information Only

and the principal dependent variable in which those equations are written is C, the concentration of solute (i.e., radionuclide) in the underlying fluid (i.e., brine). A pair of conservation equations arises for each radionuclide, 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 radionuclide 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 (radionuclide's) elements as part of its input data. It allocates the solubility limit for the isotopes according to their mole fractions. 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.

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, proving 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:

$$|Mass\ Flux\ in - Mass\ Flux\ out + Source/Sink\ = Accumulation|_{REV}$$

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^*, and q_z^*, as shown in Figure 4-2.

Information Only

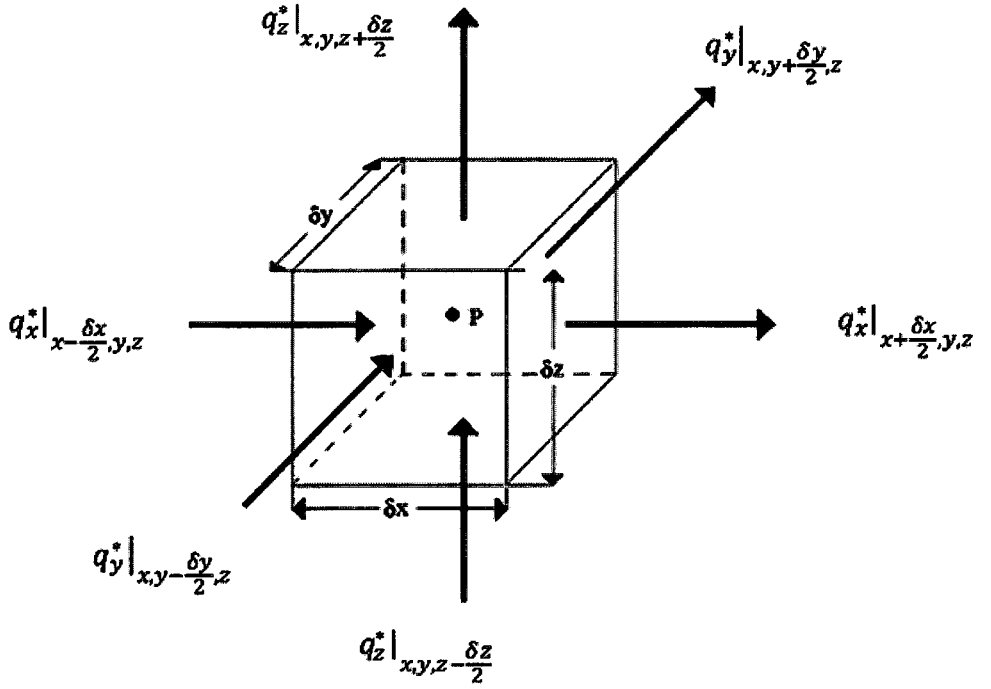


Figure 4-2. Schematic diagram showing the REV and the fluxes q^*

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

$$\left[q_x^*|_{x-\frac{\delta x}{2},y,z} - q_x^*|_{x+\frac{\delta x}{2},y,z}\right]\delta y\delta z\delta t \qquad \text{Eq. 4-1}$$

Using Taylor's series to expand around the central point P(x,y,z) of the REV, Eq. 4-1 becomes:

$$\left[\left\{q_x^* - \frac{\delta x}{2}\left(\frac{\partial q_x^*}{\partial x}\right) + \frac{(\delta x)^2}{8}\left(\frac{\partial^2 q_x^*}{\partial x^2}\right) - \cdots\right\} - \left\{q_x^* + \frac{\delta x}{2}\left(\frac{\partial q_x^*}{\partial x}\right) + \frac{(\delta x)^2}{8}\left(\frac{\partial^2 q_x^*}{\partial x^2}\right) + \cdots\right\}\right]\delta y\delta z\delta t \qquad \text{Eq. 4-2}$$

Simplifying and neglecting higher order terms (which go to zero in the limit), Eq. 4-2 reduces to:

$$-\left(\frac{\partial q_x^*}{\partial x}\right)\delta x\delta y\delta z\delta t$$

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:

$$-\left[\frac{\partial q_x^*}{\partial x} + \frac{\partial q_y^*}{\partial y} + \frac{\partial q_z^*}{\partial z}\right]\delta x\delta y\delta z\delta t \qquad \text{Eq. 4-3}$$

Information Only

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

$$q^*_{s/s} C'_{i/p} \delta x \delta y \delta z \delta t \qquad \text{Eq. 4-4}$$

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:

$$\frac{\partial(\phi S \rho C')}{\partial t} \delta x \delta y \delta z \delta t \qquad \text{Eq. 4-5}$$

Combining the terms in Eq. 4-3, Eq. 4-4 and Eq. 4-5 according to the word equation stated in Section 4.3 results in:

$$-\left[\frac{\partial q^*_x}{\partial x} + \frac{\partial q^*_y}{\partial y} + \frac{\partial q^*_z}{\partial z}\right] \delta x \delta y \delta z \delta t + q^*_{s/s} C'_{i/p} \delta x \delta y \delta z \delta t = \frac{\partial(\phi S \rho C')}{\partial t} \delta x \delta y \delta z \delta t \qquad \text{Eq. 4-6}$$

Dividing Eq. 4-6 by $\delta x \delta y \delta z \delta t$ and taking the limit as the control volume approaches zero leads to:

$$-\nabla \cdot q^* + q^*_{s/s} C'_{i/p} = \frac{\partial(\phi S \rho C')}{\partial t} \qquad \text{Eq. 4-7}$$

where the nomenclature is defined as follows:

$$q^* = \rho v C' \qquad \text{Eq. 4-8}$$

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)

If Eq. 4-7 is divided by the solvent density ρ (a constant), and q^* is replaced by its equivalent value (Eq. 4-8), Eq. 4-7 becomes:

$$-\nabla \cdot vC + qC_{i/p} = \frac{\partial(\phi SC)}{\partial t} \qquad \text{Eq. 4-9}$$

where:

Information Only

C = solute concentration (kg/ m^3)

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

Eq. 4-9 as written applies to the fluid domain as a whole (i.e., matrix plus fracture). However, as was said, we prefer to divide the domain into two separate flow regimes, matrix and fracture, and to allow the two to intercommunicate through a transfer function. The transfer function represents mass flux from the fracture to the matrix. 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. The continuity equation that applies to either domain is written as follows:

$$-\nabla \cdot v_f C_f + q_f C_{fi/p} - \bar{\tau}_{m/f} C_{m/f} = \frac{\partial(\phi SC)_f}{\partial t}, for\ the\ fracture \qquad \text{Eq. 4-10}$$

$$-\nabla \cdot v_m C_m + q_m C_{mi/p} + \bar{\tau}_{m/f} C_{m/f} = \frac{\partial(\phi SC)_m}{\partial t}, for\ the\ matrix \qquad \text{Eq. 4-11}$$

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 $\bar{\tau}_{m/f}$ is the volumetric exchange rate, $m^3/s/m^3$ of solvent mass transfer between the fracture and the matrix. The $+$ and $-$ signs in the transfer terms in Eq. 4-10 and Eq. 4-11 imply that the transfer acts as a source in one continuum and as a sink in the other. That is, what is lost from the matrix is gained by the fracture and vice versa.

4.3.2 Advective transport across fracture/matrix interfaces

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 formula:

$$\tau_w = \sigma k_m \left(\frac{k_r}{\mu}\right)_{m/f} \left[P_f - P_m - \gamma_{m/f}(D_f - D_m)\right] \qquad \text{Eq. 4-12}$$

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}).

Information Only

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 thicken 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:

$$F_D = \phi S K \nabla C \qquad \text{Eq. 4-13}$$

where F_D is the dispersion flux. Generally (but **not** in compliance 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:

$$K = \begin{vmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{vmatrix} \qquad \text{Eq. 4-14}$$

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,

$$K_{ij} = \frac{D_m^*}{\tau^*}\delta_{ij} + \frac{\alpha_T}{\phi S}|v|\delta_{ij} + \frac{v_i v_j}{\phi S|v|}(\alpha_L - \alpha_T) \qquad \text{Eq. 4-15}$$

Information Only

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):

$$K_{fx} = \left\{ \frac{D_m^*}{\tau_f^*} + \frac{1}{\phi_f S_f |\vec{v}_f|} \left[\alpha_{Lf} (v_{fx})^2 + \alpha_{Tf} \left[(v_{fy})^2 + (v_{fz})^2 \right] \right] \right\} \qquad \text{Eq. 4-16}$$

and

$$K_{mx} = \left\{ \frac{D_m^*}{\tau_m^*} + \frac{1}{\phi_m S_m |\vec{v}_m|} \left[\alpha_{Lm} (v_{mx})^2 + \alpha_{Tm} \left[(v_{my})^2 + (v_{mz})^2 \right] \right] \right\} \qquad \text{Eq. 4-17}$$

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:

$$|\vec{v}| = \sqrt{(v_x)^2 + (v_y)^2 + (v_z)^2} \qquad \text{Eq. 4-18}$$

where v_x, v_y, and 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.

Dispersion in the fracture and matrix is generally small and can be neglected in comparison to the advective part of the transport. Moreover, dispersion can always be a significant means of mass transfer between the fracture and matrix flow zones. Fick's law for the transport between the fracture and the matrix is written as:

$$F_{Dm/f} = \phi_m S_m K_{m/f} (C_f - C_m) \qquad \text{Eq. 4-19}$$

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

[+] 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.

Information Only

$$K_{wm/f} = \sigma\left\{\frac{D_m^*}{\tau_m^*} + \alpha_m \frac{|v_{m/f}|}{\phi_m S_m}\right\} \qquad \text{Eq. 4-20}$$

where

$v_{m/f}$ is the velocity of liquid transfer (m/s), and

σ is a shape factor.

In Eq. 4-20, it is assumed that the contribution from the fracture dispersion is negligible compared with the matrix (the fracture aperture is very small compared with the matrix block dimensions).

The shape factor σ is defined, in general, as

$$\sigma = \sum_s \frac{A_m}{V_m d} \qquad \text{Eq. 4-21}$$

where

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,

$$\sigma = 4\left[\frac{1}{(L_x)^2} + \frac{1}{(L_y)^2} + \frac{1}{(L_z)^2}\right] \qquad \text{Eq. 4-22}$$

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

Collecting intermediate results, if Eq. 4-13 and Eq. 4-19 are added to Eq. 4-10 and Eq. 4-11, the resulting equations will be

$$\nabla \cdot \phi_f S_f K_f \nabla C_f - \nabla \cdot v_f C_f + q_f C_{fi/p} - \bar{\tau}_{m/f} C_{m/f} - \phi_m S_m K_{m/f}(C_f - C_m) = \frac{\partial(\phi SC)_f}{\partial t}, for\ the\ fracture \qquad \text{Eq. 4-23}$$

$$\nabla \cdot \phi_m S_m K_m \nabla C_m - \nabla \cdot v_m C_m + q_m C_{mi/p} + \bar{\tau}_{m/f} C_{m/f} + \phi_m S_m K_{m/f}(C_f - C_m) = \frac{\partial(\phi SC)_m}{\partial t}, for\ the\ matrix \qquad \text{Eq. 4-24}$$

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

Information Only

microscopic nature of the heterogeneities inherent in porous media. However, examination of Eq. 4-16 and Eq. 4-17 suggests the effective dispersion coefficient will always be positive. 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.

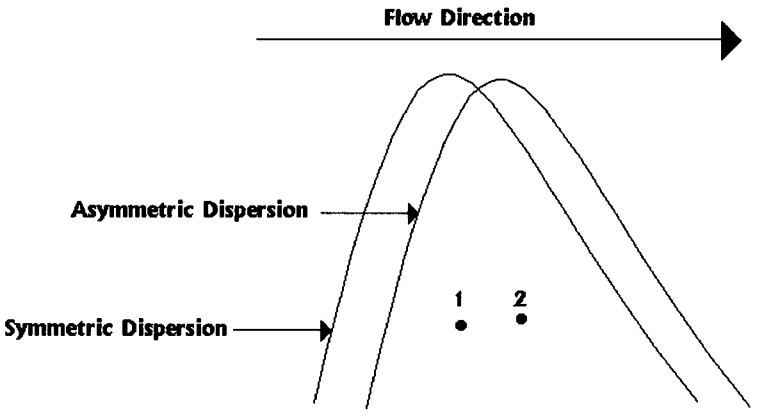


Figure 4-3. Symmetric versus asymmetric dispersion fronts

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

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

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 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

Information Only

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,

$$\frac{\partial R}{\partial t} = k_1 C - k_2 C_s \qquad \text{Eq. 4-25}$$

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,

$$\frac{\partial R}{\partial t} \to 0 \qquad as\ t \to \infty, \qquad and$$
$$\frac{k_1}{k_2} = \chi = \frac{C_s}{C} \Rightarrow C_s = \chi C \qquad \text{Eq. 4-26}$$

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. The mass rate of sorption per unit volume, F_s, in kg/s/m^3 can therefore be determined from

$$F_s = \frac{\partial}{\partial t}(1 - \phi)\rho_s C_s \qquad \text{Eq. 4-27}$$

where

ρ_s = rock grain density kg/m^3 .

4.3.6 Matrix/Fracture boundary sorption

Many dissolved WIPP chemicals are subject to sorption onto the solid surfaces that form the boundaries in the rock matrix through which they flow. This phenomenon takes place because of the locally irregular atomic and molecular properties of the surface skins of virtually all materials. 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. Boundary sorption manifests itself as a retarding factor and is expressed in NUTS through the variables C_{sm} and C_{sf}, which represent the sorbed concentration of any specific radionuclide per unit mass of rock. It is conventional to relate C_{sm} and C_{sf} to the solute concentration by any of various empirical relations called "isotherms." As the name suggests, these isotherms are families of empirical graphs of

Information Only

sorbate concentration versus solute concentration at constant temperature. NUTS accommodates three equilibrium isotherms. They are :

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

$$C_s^{n+1} = k_d C_w^{n+1} \qquad \text{Eq. 4-28}$$

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:

$$C_s^{n+1} = x_1 \left[C_w \frac{1}{x_2}\right]^{n+1} = x_1 \left[C_w \frac{1-x_2}{x_2}\right]^{n} C_w^{n+1} = \chi C_w^{n+1} \qquad \text{Eq. 4-29}$$

where n refers to the former time step, $\chi = x_1 \left[C_w \frac{1-x_2}{x_2}\right]^n$, 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:

$$C_s^{n+1} = \chi_1 C_w^{n+1} \quad where\ \chi_1 = x_1 \left[1 - \frac{x_2 C_w^n}{1 + x_2 C_w^n}\right] \qquad \text{Eq. 4-30}$$

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.

Matrix/fracture boundary sorption and desorption are **not** included in the compliance calculations. 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. See Section 4.5.1.

4.3.7 Decay

Radionuclides are naturally unstable and spontaneously emit clusters of particles equivalent to helium atoms (α particle) until, eventually, they arrive at stable configurations. Thus, the mass of daughter products is always lower than that of the parent radionuclide. Rates of disintegration vary over a wide range. Some are so slow that the parent radionuclides 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 α particle emission and spontaneous fission, and 2) disintegration without a change in atomic mass, such as the emission of positive or negative β particles and the capture by the nucleus of an electron in the

Information Only

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

$$\frac{\partial N}{\partial t} = -\lambda N \qquad \text{Eq. 4-31}$$

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 Eq. 4-31 leads to solutions of the form

$$N = N_0 e^{-\lambda t} \qquad \text{Eq. 4-32}$$

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:

$$\lambda = \frac{ln(2)}{t_{1/2}} \qquad \text{Eq. 4-33}$$

where $t_{1/2}$ is the half-life. The decay constant and, therefore, the half-life are fixed physical characteristics of each unstable radionuclide 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, Eq. 4-31 and Eq. 4-32 can be written in terms of mass M as:

$$\frac{\partial N}{\partial t}\left(\frac{M_w}{N_A}\right) = -\lambda N\left(\frac{M_w}{N_A}\right) \qquad \text{Eq. 4-34}$$

$$\frac{\partial M}{\partial t} = -\lambda M \qquad \text{Eq. 4-35}$$

and

$$M = M_0 e^{-\lambda t} \qquad \text{Eq. 4-36}$$

where $M = N\left(\frac{M_w}{N_A}\right)$ is mass of the solute at any time t (kg), M_0 is mass of the solute at the initial time t_0 (kg), M_w is molar weight of the solute (kg/mole), and N_A is the Avogadro number (= 6.02×10^{23} atoms/mole).

In porous media, the total mass of a radionuclide is distributed between the dissolved mass and the adsorbed mass. Therefore, from Eq. 4-35, the rate at which mass is transported by decay, F_{Decay}, in kg/s/m^3 , can be written in terms of concentration as

$$F_{Decay} = -\lambda(\phi SC + (1-\phi)\rho_s C_s) \qquad \text{Eq. 4-37}$$

Information Only

Eq. 4-37 represents a sink (i.e., loss of mass). 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

$$F_g = \sum_J (\phi SC + (1-\phi)\rho_s C_s)_{pj}\, \lambda_{pj} \qquad \text{Eq. 4-38}$$

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

4.3.8 The overall mass transport equation

Amassing all the source/sink effects associated with sorption and radioactive decay (Eq. 4-27, Eq. 4-37, and Eq. 4-38) within the continuity equation thus far derived (Eq. 4-23 and Eq. 4-24), the overall mass-transport equation applicable to either hydrological regime is:

$$\begin{aligned} \nabla \cdot \phi SK\nabla C - \nabla \cdot vC + C^* q \pm \bar{\tau} C_{m/f} \pm \phi_m S_m K_{m/f}(C_f - C_m) \\ = \frac{\partial(\phi SC + (1-\phi)\rho_s C_s)}{\partial t} + (\phi SC + (1-\phi)\rho_s C_s)\lambda \\ - \sum_{j=1}^{J} (\phi SC + (1-\phi)\rho_s C_s)_{pj} \lambda_{pj} C_{pj} \end{aligned} \qquad \text{Eq. 4-39}$$

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:

Fracture Equation:

$$\begin{aligned} \nabla \cdot \phi_{wf} S_{wf} K_{wf} \nabla C_{wf} - \nabla \cdot v_{wf} C_{wf} + C^*_{wf} q_{wf} - \bar{\tau} C_{wm/f} - \phi_{wm} S_{wm} K_{wm/f}(C_{wf} \\ - C_{wm}) \\ = \frac{\partial}{\partial t}(\phi_{wf} S_{wf} C_{wf} + (1-\phi_t)\rho_s C_{sf}) + (\phi_{wf} S_{wf} C_{wf} \\ + (1-\phi_t)\rho_s C_{sf})\lambda \\ - \sum_{j=1}^{J} (\phi_{wf} S_{wf} C_{wf} + (1-\phi_t)\rho_s C_{sf})_{pj} \lambda_{pj} C_{wfpj} \end{aligned} \qquad \text{Eq. 4-40}$$

Information Only

Matrix Equation:

$$\begin{aligned}\nabla \cdot \phi_{wm} S_{wm} K_{wm} \nabla C_{wm} - \nabla \cdot v_{wm} C_{wm} + C^*_{wm} q_{wm} - \bar{\tau} C_{wm/f} \\ - \phi_{wm} S_{wm} K_{wm/f} (C_{wf} - C_{wm}) \\ = \frac{\partial}{\partial t} (\phi_{wm} S_{wm} C_{wm} + (1 - \phi_t) \rho_s C_{sm}) + (\phi_{wm} S_{wm} C_{wm} \\ + (1 - \phi_t) \rho_s C_{sm}) \lambda \\ - \sum_{j=1}^{J} (\phi_{wm} S_{wm} C_{wm} + (1 - \phi_t) \rho_s C_{sm})_{pj} \lambda_{pj} C_{wmpj}\end{aligned} \qquad \text{Eq. 4-41}$$

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

4.3.9 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; and 0,Z) is

$$C_{wm}(x, y, z, 0) = f_m(x, y, z) \qquad 0 \le x \le X, 0 \le y \le Y, 0 \le z \le Z \qquad \text{Eq. 4-42}$$

Similarly, the fracture is initialized by

$$C_{wf}(x, y, z, 0) = f_f(x, y, z) \qquad 0 \le x \le X, 0 \le y \le Y, 0 \le z \le Z \qquad \text{Eq. 4-43}$$

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

$$f_f(x, y, z) = f_m(x, y, z) \qquad \text{Eq. 4-44}$$

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

$$\begin{aligned} C(0, y, z, t) &= g_{1x}(t) & t \ge 0 \\ C(X, y, z, t) &= g_{1X}(t) & t \ge 0 \\ C(x, 0, z, t) &= g_{1y}(t) & t \ge 0 \\ C(x, Y, z, t) &= g_{1Y}(t) & t \ge 0 \\ C(x, y, 0, t) &= g_{1z}(t) & t \ge 0 \\ C(x, y, Z, t) &= g_{1Z}(t) & t \ge 0 \end{aligned} \qquad \text{Eq. 4-45}$$

Information Only

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

$$\begin{aligned}
\frac{\partial}{\partial t}[C(0,y,z,t)] &= g_{2x}(t) \qquad t \geq 0 \\
\frac{\partial}{\partial t}[C(X,y,z,t)] &= g_{2X}(t) \qquad t \geq 0 \\
\frac{\partial}{\partial t}[C(x,0,z,t)] &= g_{2y}(t) \qquad t \geq 0 \\
\frac{\partial}{\partial t}[C(x,Y,z,t)] &= g_{2Y}(t) \qquad t \geq 0 \\
\frac{\partial}{\partial t}[C(x,y,0,t)] &= g_{2z}(t) \qquad t \geq 0 \\
\frac{\partial}{\partial t}[C(x,y,Z,t)] &= g_{2Z}(t) \qquad t \geq 0
\end{aligned} \qquad \text{Eq. 4-46}$$

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

In NUTS applications, Dirichlet boundary conditions can be used at both the outer boundaries or internally at sources. Neumann boundary conditions, on the other hand are used to specify no-transport at the outer boundaries of the simulation domain, i.e., $\frac{\partial C}{\partial x} = 0$, $\frac{\partial C}{\partial y} = 0$ and $\frac{\partial C}{\partial z} = 0$

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 integration 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 L, as shown in Figure 4-4.

Information Only

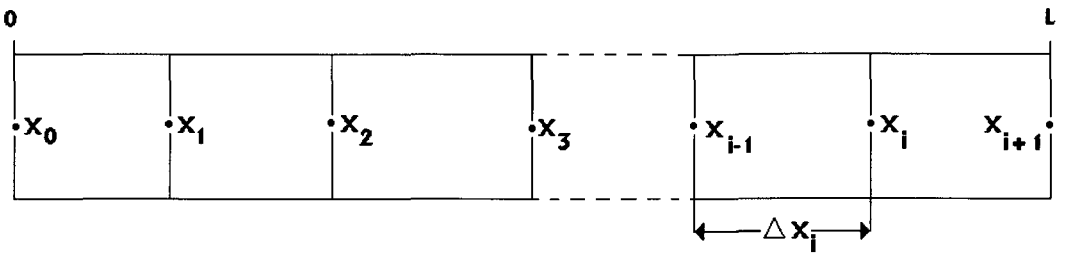


Figure 4-4. Point-center grid in finite difference approximation

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, ... The algebra is a little easier if the intervals are of equal width Δx , where Δx is called the finite difference.

$$x_i = i\Delta x \qquad \text{Eq. 4-47}$$

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

$$\begin{aligned} x_{i+1} &= x_i + \Delta x \\ x_{i-1} &= x_i - \Delta x \end{aligned} \qquad \text{Eq. 4-48}$$

and Δx is the distance between successive grid points.

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, ..., I+1 , and is represented by

$$C(x_i) = C_i \qquad \text{Eq. 4-49}$$

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 $C(x + \Delta x)$,

$$C(x+\Delta x) = C(x) + \frac{\partial C}{\partial x}\Delta x + \frac{\partial^2 C}{\partial x^2}\frac{(\Delta x)^2}{2!} + \frac{\partial^3 C}{\partial x^3}\frac{(\Delta x)^3}{3!} + \cdots + R_n \qquad \text{Eq. 4-50}$$

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

$$R_n = \frac{\partial^n C(x+\zeta\Delta x)}{\partial x^n}\frac{(\Delta x)^n}{n!} \qquad \text{Eq. 4-51}$$

Information Only

and ζ lies between 0 and 1, so the argument of C lies between x and $x+\Delta x$. Eq. 4-51 is sometimes written in the form

$$R_n = O[|\Delta x|^n] \qquad \text{Eq. 4-52}$$

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

$$|R_n| \le M(|\Delta x|^n) \quad as\ \Delta x \to 0 \qquad \text{Eq. 4-53}$$

If we use a more compact notation, Eq. 4-50 may be written as

$$C_{i+1} = C_i + C_{ix}\Delta x + C_{ixx}\frac{(\Delta x)^2}{2!} + C_{ixxx}\frac{(\Delta x)^3}{3!} + \cdots + R_n \qquad \text{Eq. 4-54}$$

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

$$C_{i-1} = C_i - C_{ix}\Delta x + C_{ixx}\frac{(\Delta x)^2}{2!} - C_{ixxx}\frac{(\Delta x)^3}{3!} + \cdots + R_n \qquad \text{Eq. 4-55}$$

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 Eq. 4-54 and Eq. 4-55, one can arrive at finite-difference approximations for the first two derivatives of C_i, as follows:

$$C_{ix} = \frac{C_{i+1} - C_i}{\Delta x} + O(\Delta x) \qquad \text{Eq. 4-56}$$

$$C_{ix} = \frac{C_i - C_{i-1}}{\Delta x} + O(\Delta x) \qquad \text{Eq. 4-57}$$

$$C_{ix} = \frac{C_{i+1} - C_{i-1}}{2\Delta x} + O[(\Delta x)^2] \qquad \text{Eq. 4-58}$$

$$C_{ixx} = \frac{C_{i+1} - 2C_i + C_{i-1}}{(\Delta x)^2} + O[(\Delta x)^2] \qquad \text{Eq. 4-59}$$

Eq. 4-56, Eq. 4-57, and Eq. 4-58 are known, respectively, as forward-, backward-, and central-difference formulae for the first derivative of C_i. Eq. 4-59 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

Information Only

discretized variable is defined, namely, at $C_{i+1/2}$ and $C_{i-1/2}$, which is in addition to the discrete points i where the concentration C_i was defined earlier (see Figure 4-5)

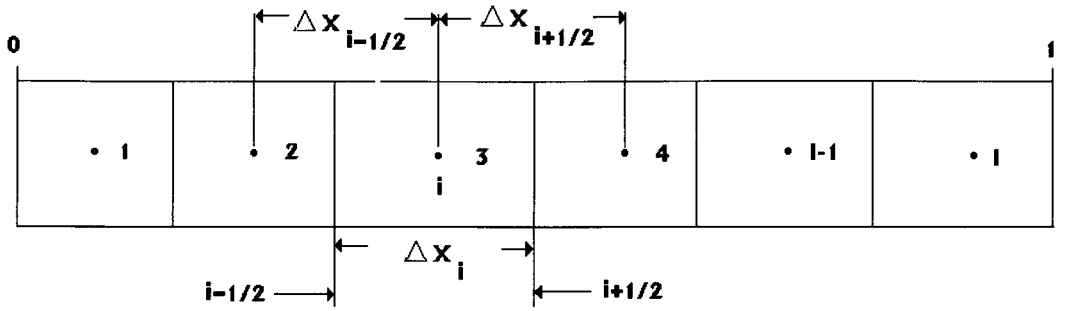


Figure 4-5. Block-center grid in finite difference approximation

where $\Delta x_{i+1/2} = \frac{\Delta x_i + \Delta x_{i+1}}{2}$ 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:

$$C_{i-\frac{1}{2}} = (0.5 + \mu_1)C_i + (0.5 - \mu_1)C_{i-1} \qquad \text{Eq. 4-60}$$

and

$$C_{i+\frac{1}{2}} = (0.5 + \mu_2)C_{i+1} + (0.5 - \mu_2)C_i \qquad \text{Eq. 4-61}$$

where

$$\mu_1 = \frac{1}{4}\frac{\Delta x_{i-1} - \Delta x_i}{\Delta x_{i-1} + \Delta x_i}, \quad \text{and} \quad \mu_2 = \frac{1}{4}\frac{\Delta x_i - \Delta x_{i+1}}{\Delta x_i + \Delta x_{i+1}} \qquad \text{Eq. 4-62}$$

From Eq. 4-62, 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:

$$C_{i-\frac{1}{2}} = \begin{cases} C_i & if\ the\ flow\ from\ i\ to\ i-1 \\ C_{i-1} & if\ the\ flow\ from\ i-1\ to\ i \end{cases} \qquad \text{Eq. 4-63}$$

Similarly

$$C_{i+\frac{1}{2}} = \begin{cases} C_i & if\ the\ flow\ from\ i\ to\ i+1 \\ C_{i+1} & if\ the\ flow\ from\ i+1\ to\ i \end{cases} \qquad \text{Eq. 4-64}$$

Information Only

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:

$$C_{i-\frac{1}{2}} = \begin{cases} (1+\mu_1)C_i - \mu_1 C_{i+1} & if\ the\ flow\ from\ i\ to\ i-1 \\ (1+\mu_2)C_{i-1} - \mu_2 C_{i-2} & if\ the\ flow\ from\ i-1\ to\ i \\ where\ \mu_1 = \dfrac{\Delta x_i}{\Delta x_i + \Delta x_{i+1}} & and\ \mu_2 = \dfrac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_{i-2}} \end{cases} \qquad \text{Eq. 4-65}$$

and

$$C_{i+\frac{1}{2}} = \begin{cases} (1+\mu_1)C_i - \mu_1 C_{i-1} & if\ the\ flow\ from\ i\ to\ i+1 \\ (1+\mu_2)C_{i+1} - \mu_2 C_{i+2} & if\ the\ flow\ from\ i+1\ to\ i \\ where\ \mu_1 = \dfrac{\Delta x_i}{\Delta x_i + \Delta x_{i-1}} & and\ \mu_2 = \dfrac{\Delta x_{i+1}}{\Delta x_{i+1} + \Delta x_{i+2}} \end{cases} \qquad \text{Eq. 4-66}$$

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

$$C_{i-\frac{1}{2}} = \frac{2C_i + 5C_{i-1} - C_{i-2}}{6} \qquad \text{Eq. 4-67}$$

and

$$C_{i+\frac{1}{2}} = \frac{2C_{i+1} + 5C_i - C_{i-1}}{6} \qquad \text{Eq. 4-68}$$

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,

$$F_{i+\frac{1}{2}} = f_i + \frac{\Delta x}{2}\frac{f_{i+1} - f_i}{\Delta x} \qquad \text{Eq. 4-69}$$

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

$$F_{i+\frac{1}{2}} = f_i + \varphi(r_i)A_{i+\frac{1}{2}} \qquad \text{Eq. 4-70}$$

where $A_{i+\frac{1}{2}}$ is the slope-term contribution and $\varphi(r_i)$, 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, $\varphi(r)$ is allowed to vary so as to eliminate spurious solutions. The variable r is defined as the ratio of successive slope terms, namely:

$$r_i = \frac{A_{i-\frac{1}{2}}}{A_{i+\frac{1}{2}}} \qquad \text{Eq. 4-71}$$

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

Information Only

$$\varphi(r_i) = \frac{|r_i| + r_i}{1 + |r_i|} \qquad \text{Eq. 4-72}$$

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:

$$TV(C^n) = \sum (C^n_{i+1} - C^n_i) \qquad \text{Eq. 4-73}$$

then in the TVD,

$$TV(C^{n+1}) \leq TV(C^n) \qquad \text{Eq. 4-74}$$

where $n + 1$ refers to the next time step. The formulation for $F_{i+\frac{1}{2}}$ is parallel. Using Eq. 4-70, it is clear that TVD is a mid-point scheme, when $\varphi(r) = 1$; a one-point upstream weighting scheme, when $\varphi(r) = 0$; and a two-point upstream weighting scheme, when $\varphi(r) = r$. 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, (1) a continuous or (2) 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 (Eq. 4-56 to Eq. 4-59). A straightforward solution can be achieved by approximating the time derivative using the forward-difference method, as shown below:

$$\left(\frac{\partial C_i}{\partial t}\right)^n = \frac{C_i^{n+1} - C_i^n}{\Delta t} \qquad \text{Eq. 4-75}$$

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 known at the time step n, the system of equations can be solved explicitly. On the other hand, if the time derivative in Eq. 4-75 is approximated using the backward-difference method and the spatial distribution of the dependent variable C_i is known at the time step $n + 1$, 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 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

Information Only

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 Eq. 4-76, in which the time derivative is discretized according to Eq. 4-75:

$$(1-\theta)\left(-u\,\frac{\partial C}{\partial x}\right)^{n+1}+\theta\left(-u\,\frac{\partial C}{\partial x}\right)^{n}=\frac{C_i^{n+1}-C_i^{n}}{\Delta t} \qquad \text{Eq. 4-76}$$

In Eq. 4-76: 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 of interest. We will discretize them in general and then develop the linear system for different numerical implementations. For the fracture, the discretized mass-transport equation in the brine is:

$$\begin{aligned}
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i+\frac{1}{2}}\left(C_{wf\,i+1}^{n+1}-C_{wf\,i}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i-\frac{1}{2}}\left(C_{wf\,i}^{n+1}-C_{wf\,i-1}^{n+1}\right)+\\
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j+\frac{1}{2}}\left(C_{wf\,j+1}^{n+1}-C_{wf\,j}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j-\frac{1}{2}}\left(C_{wf\,j}^{n+1}-C_{wf\,j-1}^{n+1}\right)\\
&+\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k+\frac{1}{2}}\left(C_{wf\,k+1}^{n+1}-C_{wf\,k}^{n+1}\right)\\
&-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k-\frac{1}{2}}\left(C_{wf\,k}^{n+1}-C_{wf\,k-1}^{n+1}\right)+\\
&q_{wf\,i+\frac{1}{2}}^{n+1}C_{wf\,i+\frac{1}{2}}^{n+1}-q_{wf\,i-\frac{1}{2}}^{n+1}C_{wf\,i-\frac{1}{2}}^{n+1}+q_{wf\,j+\frac{1}{2}}^{n+1}C_{wf\,j+\frac{1}{2}}^{n+1}-q_{wf\,j-\frac{1}{2}}^{n+1}C_{wf\,j-\frac{1}{2}}^{n+1}+\\
&q_{wf\,k+\frac{1}{2}}^{n+1}C_{wf\,k+\frac{1}{2}}^{n+1}-q_{wf\,k-\frac{1}{2}}^{n+1}C_{wf\,k-\frac{1}{2}}^{n+1}+C_{wf\,i}^{*\,n+1}Q_{wf\,i}^{n+1}-\\
&\tau_{wm/f\,i}^{n+1}\left(\omega_{m/f\,i}C_{wf\,i}^{n+1}+(1-\omega_{m/f\,i})C_{wm\,i}^{n+1}\right)-\\
&\left(\phi_{m\,i}^{n+1}S_{wm\,i}^{n+1}K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1}-C_{wm\,i}^{n+1}\right)=\\
&\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_sC_{sf\,i}\right\}^{n+1}-\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_sC_{sf\,i}\right\}^{n}\right]\\
&+V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_sC_{sf\,i}\right\}^{n+1}\lambda-\\
&V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li}+(1-\phi_t)\rho_sC_{sf\,li}\right\}\lambda_l\right]^{n+1}
\end{aligned} \qquad \text{Eq. 4-77}$$

Information Only

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

$$\begin{aligned}
&\left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{i+\frac{1}{2}} \left(C_{wm\,i+1}^{n+1} - C_{wm\,i}^{n+1}\right) - \left(\phi_m^{n+1} S_{wf}^{n+1} K_{wm}^{n+1}\right)_{i-\frac{1}{2}} \left(C_{wm\,i}^{n+1} - C_{wm\,i-1}^{n+1}\right) + \\
&\left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{j+\frac{1}{2}} \left(C_{wm\,j+1}^{n+1} - C_{wm\,j}^{n+1}\right) - \left(\phi_m^{n+1} S_{wf}^{n+1} K_{wm}^{n+1}\right)_{j-\frac{1}{2}} \left(C_{wm\,j}^{n+1} - C_{wm\,j-1}^{n+1}\right) \\
&+ \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{k+\frac{1}{2}} \left(C_{wm\,k+1}^{n+1} - C_{wm\,k}^{n+1}\right) \\
&- \left(\phi_m^{n+1} S_{wf}^{n+1} K_{wm}^{n+1}\right)_{k-\frac{1}{2}} \left(C_{wm\,k}^{n+1} - C_{wm\,k-1}^{n+1}\right) + \\
&q_{wm\,i+\frac{1}{2}}^{n+1} C_{wm\,i+\frac{1}{2}}^{n+1} - q_{wm\,i-\frac{1}{2}}^{n+1} C_{wm\,i-\frac{1}{2}}^{n+1} + q_{wm\,j+\frac{1}{2}}^{n+1} C_{wm\,j+\frac{1}{2}}^{n+1} - q_{wm\,j-\frac{1}{2}}^{n+1} C_{wm\,j-\frac{1}{2}}^{n+1} + \\
&q_{wm\,k+\frac{1}{2}}^{n+1} C_{wm\,k+\frac{1}{2}}^{n+1} - q_{wm\,k-\frac{1}{2}}^{n+1} C_{wm\,k-\frac{1}{2}}^{n+1} + C_{wm\,i}^{*\,n+1} Q_{wm\,i}^{n+1} + \\
&\tau_{wm/f\,i}^{n+1} \left(\omega_{m/f\,i} C_{wf\,i}^{n+1} + (1 - \omega_{m/f\,i}) C_{wm\,i}^{n+1}\right) + \\
&\left(\phi_{m\,i}^{n+1} S_{wm\,i}^{n+1} K_{wm/f\,i}^{n+1}\right) \left(C_{wf\,i}^{n+1} - C_{wm\,i}^{n+1}\right) = \\
&\frac{V_{R\,i}}{\Delta t} \left[\left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1 - \phi_t) \rho_s C_{sm\,i}\right\}^{n+1} - \left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1 - \phi_t) \rho_s C_{sm\,i}\right\}^{n}\right] \\
&+ V_{R\,i} \left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1 - \phi_t) \rho_s C_{sm\,i}\right\}^{n+1} \lambda - \\
&V_{R\,i} \sum_{l=1}^{L} \left[\left\{\phi_{m\,i} S_{wm\,i} C_{wm\,li} + (1 - \phi_t) \rho_s C_{sm\,li}\right\} \lambda_l\right]^{n+1}
\end{aligned} \qquad \text{Eq. 4-78}$$

Information Only

4.4.2 One-point upstream-winding discretization of transport equations

Fracture Equation

$$\begin{aligned}
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i+\frac{1}{2}}\left(C_{wf\,i+1}^{n+1}-C_{wf\,i}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i-\frac{1}{2}}\left(C_{wf\,i}^{n+1}-C_{wf\,i-1}^{n+1}\right)+\\
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j+\frac{1}{2}}\left(C_{wf\,j+1}^{n+1}-C_{wf\,j}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j-\frac{1}{2}}\left(C_{wf\,j}^{n+1}-C_{wf\,j-1}^{n+1}\right)\\
&+\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k+\frac{1}{2}}\left(C_{wf\,k+1}^{n+1}-C_{wf\,k}^{n+1}\right)\\
&-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k-\frac{1}{2}}\left(C_{wf\,k}^{n+1}-C_{wf\,k-1}^{n+1}\right)+\\
&q_{wf\,i+\frac{1}{2}}^{n+1}\left(\omega_{f\,i+1}C_{wf\,i}^{n+1}+\left(1-\omega_{f\,i+1}\right)C_{wf\,i+1}^{n+1}\right)-\\
&q_{wf\,i-\frac{1}{2}}^{n+1}\left(\omega_{f\,i}C_{wf\,i-1}^{n+1}+\left(1-\omega_{f\,i}\right)C_{wf\,i}^{n+1}\right)+\\
&q_{wf\,j+\frac{1}{2}}^{n+1}\left(\omega_{f\,j+1}C_{wf\,j}^{n+1}+\left(1-\omega_{f\,j+1}\right)C_{wf\,j+1}^{n+1}\right)-\\
&q_{wf\,j-\frac{1}{2}}^{n+1}\left(\omega_{f\,j}C_{wf\,j-1}^{n+1}+\left(1-\omega_{f\,j}\right)C_{wf\,j}^{n+1}\right)+\\
&q_{wf\,k+\frac{1}{2}}^{n+1}\left(\omega_{f\,k+1}C_{wf\,k}^{n+1}+\left(1-\omega_{f\,k+1}\right)C_{wf\,k+1}^{n+1}\right)-\\
&q_{wf\,k-\frac{1}{2}}^{n+1}\left(\omega_{f\,k}C_{wf\,k-1}^{n+1}+\left(1-\omega_{f\,k}\right)C_{wf\,k}^{n+1}\right)+\\
&C_{wf\,i}^{*\,n+1}Q_{wf\,i}^{n+1}-\tau_{wm/f\,i}^{n+1}\left(\omega_{m/f\,i}C_{wf\,i}^{n+1}+\left(1-\omega_{m/f\,i}\right)C_{wm\,i}^{n+1}\right)-\\
&\left(\phi_{m\,i}^{n+1}S_{wm\,i}^{n+1}K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1}-C_{wm\,i}^{n+1}\right)=\\
&\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n+1}-\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n}\right]\\
&+V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n+1}\lambda-\\
&V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li}+\left(1-\phi_t\right)\rho_sC_{sf\,li}\right\}\lambda_l\right]^{n+1}
\end{aligned}$$

Eq. 4-79

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

$$\begin{aligned}
&A_f\,C_{wf\,k-1}^{n+1}+B_f\,C_{wf\,j-1}^{n+1}+C_f\,C_{wf\,i-1}^{n+1}+D_{1f}\,C_{wf\,ijk}^{n+1}+D_{2f}\,C_{wm\,ijk}^{n+1}+\\
&E_f\,C_{wf\,i+1}^{n+1}+F_f\,C_{wf\,j+1}^{n+1}+G_f\,C_{wf\,k+1}^{n+1}=R_f^{n+1}
\end{aligned}$$

Eq. 4-80

where

$$A_f=HD_{wf\,k-\frac{1}{2}}^{n+1}-\omega_{f\,k}q_{wf\,k-\frac{1}{2}}^{n+1}$$

$$B_f=HD_{wf\,j-\frac{1}{2}}^{n+1}-\omega_{f\,j}q_{wf\,j-\frac{1}{2}}^{n+1}$$

$$C_f=HD_{wf\,i-\frac{1}{2}}^{n+1}-\omega_{f\,i}q_{wf\,i-\frac{1}{2}}^{n+1}$$

$$E_f=HD_{wf\,i+\frac{1}{2}}^{n+1}-\left(1-\omega_{f\,i+1}\right)q_{wf\,i+\frac{1}{2}}^{n+1}$$

Information Only

$$F_f = HD^{n+1}_{wf\,j+\frac{1}{2}} - (1-\omega_{f\,j+1})q^{n+1}_{wf\,j+\frac{1}{2}}$$

$$G_f = HD^{n+1}_{wf\,k+\frac{1}{2}} - (1-\omega_{f\,k+1})q^{n+1}_{wf\,k+\frac{1}{2}}$$

$$D_{1f} = -HD^{n+1}_{wf\,k-\frac{1}{2}} - HD^{n+1}_{wf\,j-\frac{1}{2}} - HD^{n+1}_{wf\,i-\frac{1}{2}} - HD^{n+1}_{wf\,k+\frac{1}{2}} - HD^{n+1}_{wf\,j+\frac{1}{2}} - HD^{n+1}_{wf\,i+\frac{1}{2}} -$$
$$(1-\omega_{f\,k})q^{n+1}_{wf\,k-\frac{1}{2}} - (1-\omega_{f\,j})q^{n+1}_{wf\,j-\frac{1}{2}} - (1-\omega_{f\,i})q^{n+1}_{wf\,i-\frac{1}{2}} +$$
$$\omega_{f\,k+1}q^{n+1}_{wf\,k+\frac{1}{2}} + \omega_{f\,j+1}q^{n+1}_{wf\,j+\frac{1}{2}} + \omega_{f\,i+1}q^{n+1}_{wf\,i+\frac{1}{2}} -$$
$$\frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1} + V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1}\lambda$$
$$-\tau^{n+1}_{wm/f\,i}\omega_{m/f\,i} - \phi^{n+1}_m S^{n+1}_{wm}K^{n+1}_{wm/f\,i}$$

$$D^{n+1}_{2f} = -\tau^{n+1}_{wm/f\,i}(1-\omega_{m/f\,i}) + \phi^{n+1}_m S^{n+1}_{wm}K^{n+1}_{wm/f\,i}$$

and

$$R^{n+1}_f = \frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}C_{wf\,i}\right\}^{n} - C^*_{wf\,i}Q^{n+1}_{wf\,i}$$
$$-V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li} + (1-\phi_t)\rho_s\xi_{f\,li}C_{wf\,li}\right\}\lambda_l\right]^{n+1}$$

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 tortousity are weighted upstream. Therefore, $HD_{wf\,i-\frac{1}{2}}$ is defined as follows:

$$HD^{n+1}_{wf\,i-\frac{1}{2}} = \omega_{f\,i}\phi_{f\,i}S_{wf\,i} + (1-\omega_{f\,i})\phi_{f\,i-1}S_{wf\,i-1}\frac{\Delta y_i\Delta z_i}{\frac{\Delta x_i + \Delta x_{i-1}}{2}}\left\{\frac{D^*_m}{\tau^*_f} + \right.$$
$$\frac{1}{\omega_{f\,i}\phi_{f\,i}S_{wf\,i} + (1-\omega_{f\,i})\phi_{f\,i-1}S_{wf\,i-1}}\frac{1}{\sqrt{(v_{xf})^2 + (v_{yf})^2 + (v_{zf})^2}}\left[\alpha_{Lf}(v_{xf})^2\right.$$
$$\left.\left. + \alpha_{Tf}\left((v_{xf})^2 + (v_{xf})^2\right)\right]\right\}$$

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. For example, in the linear adsorption isotherm ξ is equivalent to k_d. 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:

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

similarly, the weighting parameters in the other direction are

Information Only

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

and

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

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

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

Matrix Equation:

$$\begin{aligned}
&\left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{i+\frac{1}{2}} \left(C_{wm\,i+1}^{n+1} - C_{wm\,i}^{n+1}\right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{i-\frac{1}{2}} \left(C_{wm\,i}^{n+1} - C_{wm\,i-1}^{n+1}\right) + \\
&\left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{j+\frac{1}{2}} \left(C_{wm\,j+1}^{n+1} - C_{wm\,j}^{n+1}\right) - \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{j-\frac{1}{2}} \left(C_{wm\,j}^{n+1} - C_{wm\,j-1}^{n+1}\right) \\
&+ \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{k+\frac{1}{2}} \left(C_{wm\,k+1}^{n+1} - C_{wm\,k}^{n+1}\right) \\
&- \left(\phi_m^{n+1} S_{wm}^{n+1} K_{wm}^{n+1}\right)_{k-\frac{1}{2}} \left(C_{wm\,k}^{n+1} - C_{wm\,k-1}^{n+1}\right) + \\
&q_{wm\,i+\frac{1}{2}}^{n+1} \left(\omega_{m\,i+1} C_{wm\,i}^{n+1} + (1-\omega_{m\,i+1}) C_{wm\,i+1}^{n+1}\right) - \\
&q_{wm\,i-\frac{1}{2}}^{n+1} \left(\omega_{m\,i} C_{wm\,i-1}^{n+1} + (1-\omega_{m\,i}) C_{wm\,i}^{n+1}\right) + \\
&q_{wm\,j+\frac{1}{2}}^{n+1} \left(\omega_{m\,j+1} C_{wm\,j}^{n+1} + (1-\omega_{m\,j+1}) C_{wm\,j+1}^{n+1}\right) - \\
&q_{wm\,j-\frac{1}{2}}^{n+1} \left(\omega_{m\,j} C_{wm\,j-1}^{n+1} + (1-\omega_{m\,j}) C_{wm\,j}^{n+1}\right) + \\
&q_{wm\,k+\frac{1}{2}}^{n+1} \left(\omega_{m\,k+1} C_{wm\,k}^{n+1} + (1-\omega_{m\,k+1}) C_{wm\,k+1}^{n+1}\right) - \\
&q_{wm\,k-\frac{1}{2}}^{n+1} \left(\omega_{m\,k} C_{wm\,k-1}^{n+1} + (1-\omega_{m\,k}) C_{wm\,k}^{n+1}\right) + \\
&C_{wm\,i}^{*\,n+1} Q_{wm\,i}^{n+1} + \tau_{wm/f\,i}^{n+1} \left(\omega_{m/f\,i} C_{wf\,i}^{n+1} + (1-\omega_{m/f\,i}) C_{wm\,i}^{n+1}\right) + \\
&\left(\phi_{m\,i}^{n+1} S_{wm\,i}^{n+1} K_{wm/f\,i}^{n+1}\right) \left(C_{wf\,i}^{n+1} - C_{wm\,i}^{n+1}\right) = \\
&\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s C_{sm\,i}\right\}^{n+1} - \left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s C_{sm\,i}\right\}^{n}\right] \\
&+ V_{R\,i}\left\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s C_{sm\,i}\right\}^{n+1} \lambda - \\
&V_{R\,i} \sum_{l=1}^{L} \left[\left\{\phi_{m\,i} S_{wm\,i} C_{wm\,li} + (1-\phi_t)\rho_s C_{sm\,li}\right\} \lambda_l\right]^{n+1}
\end{aligned}$$

Eq. 4-81

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

$$\begin{aligned}
&A_m\, C_{wm\,k-1}^{n+1} + B_m\, C_{wm\,j-1}^{n+1} + C_m\, C_{wm\,i-1}^{n+1} + D_{1m}\, C_{wm\,ijk}^{n+1} + D_{2m}\, C_{wf\,ijk}^{n+1} + \\
&E_m\, C_{wm\,i+1}^{n+1} + F_m\, C_{wm\,j+1}^{n+1} + G_m\, C_{wm\,k+1}^{n+1} = R_m^{n+1}
\end{aligned}$$

Eq. 4-82

where

Information Only

$$A_m = HD^{n+1}_{wm\,k-\frac{1}{2}} - \omega_{m\,k} q^{n+1}_{wm\,k-\frac{1}{2}}$$

$$B_m = HD^{n+1}_{wm\,j-\frac{1}{2}} - \omega_{m\,j} q^{n+1}_{wm\,j-\frac{1}{2}}$$

$$C_m = HD^{n+1}_{wm\,i-\frac{1}{2}} - \omega_{m\,i} q^{n+1}_{wm\,i-\frac{1}{2}}$$

$$E_m = HD^{n+1}_{wm\,i+\frac{1}{2}} - (1-\omega_{m\,i+1}) q^{n+1}_{wm\,i+\frac{1}{2}}$$

$$F_m = HD^{n+1}_{wm\,j+\frac{1}{2}} - (1-\omega_{m\,j+1}) q^{n+1}_{wm\,j+\frac{1}{2}}$$

$$G_m = HD^{n+1}_{wm\,k+\frac{1}{2}} - (1-\omega_{m\,k+1}) q^{n+1}_{wm\,k+\frac{1}{2}}$$

$$D_{1m} = -HD^{n+1}_{wm\,k-\frac{1}{2}} - HD^{n+1}_{wm\,j-\frac{1}{2}} - HD^{n+1}_{wm\,i-\frac{1}{2}} - HD^{n+1}_{wm\,k+\frac{1}{2}} - HD^{n+1}_{wm\,j+\frac{1}{2}} - HD^{n+1}_{wm\,i+\frac{1}{2}}$$
$$-(1-\omega_{m\,k}) q^{n+1}_{wm\,k-\frac{1}{2}} - (1-\omega_{m\,j}) q^{n+1}_{wm\,j-\frac{1}{2}} - (1-\omega_{m\,i}) q^{n+1}_{wm\,i-\frac{1}{2}} +$$
$$\omega_{m\,k+1} q^{n+1}_{wm\,k+\frac{1}{2}} + \omega_{m\,j+1} q^{n+1}_{wm\,j+\frac{1}{2}} + \omega_{m\,i+1} q^{n+1}_{wm\,i+\frac{1}{2}} -$$
$$\frac{V_{R\,i}}{\Delta t}\{\phi_{m\,i} S_{wm\,i} + (1-\phi_t)\rho_s \xi_{m\,i}\}^{n+1} + V_{R\,i}\{\phi_{m\,i} S_{wm\,i} + (1-\phi_t)\rho_s \xi_{m\,i}\}^{n+1}\lambda$$
$$+\tau^{n+1}_{wm/f\,i}(1-\omega_{m/f\,i}) - \phi^{n+1}_m S^{n+1}_{wm} K^{n+1}_{wm/f\,i}$$

$$D^{n+1}_{2m} = \tau^{n+1}_{wm/f\,i}\omega_{m/f\,i} + \phi^{n+1}_m S^{n+1}_{wm} K^{n+1}_{wm/f\,i}$$

and

$$R^{n+1}_m = \frac{V_{R\,i}}{\Delta t}\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s \xi_{m\,i} C_{wm\,i}\}^n - C^*_{wm\,i} Q^{n+1}_{wm\,i}$$
$$-V_{R\,i}\sum_{l=1}^{L}[\{\phi_{m\,i} S_{wm\,i} C_{wm\,li} + (1-\phi_t)\rho_s \xi_{m\,li} C_{wm\,li}\}\lambda_l]^{n+1}$$

In a similar fashion, the HD in the matrix at $i-\frac{1}{2}$ interface is defined as follows:

$$HD^{n+1}_{wm\,i-\frac{1}{2}} = \omega_{m\,i}\phi_{m\,i} S_{wm\,i} + (1-\omega_{m\,j})\phi_{m\,i-1} S_{wm\,i-1} \frac{\Delta y_i \Delta z_i}{\frac{\Delta x_i + \Delta x_{i-1}}{2}}\left\{\frac{D^*_m}{\tau^*_m} + \right.$$
$$\frac{1}{\omega_{m\,i}\phi_{m\,i} S_{wm\,i} + (1-\omega_{m\,i})\phi_{m\,i-1} S_{wm\,i-1}} \frac{1}{\sqrt{(v_{xm})^2 + (v_{ym})^2 + (v_{zm})^2}} [\alpha_{Lm}(v_{xm})^2$$
$$\left. +\alpha_{Tm}((v_{xm})^2 + (v_{xm})^2)]\right\}$$

Similar definitions for y and ω's will be extended to the matrix discretization.

Information Only

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}
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i+\frac{1}{2}}\left(C_{wf\,i+1}^{n+1}-C_{wf\,i}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{i-\frac{1}{2}}\left(C_{wf\,i}^{n+1}-C_{wf\,i-1}^{n+1}\right)+\\
&\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j+\frac{1}{2}}\left(C_{wf\,j+1}^{n+1}-C_{wf\,j}^{n+1}\right)-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{j-\frac{1}{2}}\left(C_{wf\,j}^{n+1}-C_{wf\,j-1}^{n+1}\right)\\
&+\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k+\frac{1}{2}}\left(C_{wf\,k+1}^{n+1}-C_{wf\,k}^{n+1}\right)\\
&-\left(\phi_f^{n+1}S_{wf}^{n+1}K_{wf}^{n+1}\right)_{k-\frac{1}{2}}\left(C_{wf\,k}^{n+1}-C_{wf\,k-1}^{n+1}\right)+\\
&q_{wf\,i+\frac{1}{2}}^{n+1}\left[\omega_{f\,i+1}\left\{\left(1+\mu_{i1}\right)C_{wf\,i}^{n+1}-\mu_{i1}C_{wf\,i-1}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,i+1}\right)\left\{\left(1+\mu_{i2}\right)C_{wf\,i+1}^{n+1}-\mu_{i2}C_{wf\,i+2}^{n+1}\right\}\right]-\\
&q_{wf\,i-\frac{1}{2}}^{n+1}\left[\omega_{f\,i}\left\{\left(1+\mu_{i3}\right)C_{wf\,i-1}^{n+1}-\mu_{i3}C_{wf\,i-2}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,i}\right)\left\{\left(1+\mu_{i4}\right)C_{wf\,i}^{n+1}-\mu_{i4}C_{wf\,i+1}^{n+1}\right\}\right]+\\
&q_{wf\,j+\frac{1}{2}}^{n+1}\left[\omega_{f\,j+1}\left\{\left(1+\mu_{j1}\right)C_{wf\,j}^{n+1}-\mu_{j1}C_{wf\,j-1}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,j+1}\right)\left\{\left(1+\mu_{j2}\right)C_{wf\,j+1}^{n+1}-\mu_{j2}C_{wf\,j+2}^{n+1}\right\}\right]-\\
&q_{wf\,j-\frac{1}{2}}^{n+1}\left[\omega_{f\,j}\left\{\left(1+\mu_{j3}\right)C_{wf\,j-1}^{n+1}-\mu_{j3}C_{wf\,j-2}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,j}\right)\left\{\left(1+\mu_{j4}\right)C_{wf\,j}^{n+1}-\mu_{j4}C_{wf\,j+1}^{n+1}\right\}\right]+\\
&q_{wf\,k+\frac{1}{2}}^{n+1}\left[\omega_{f\,k+1}\left\{\left(1+\mu_{k1}\right)C_{wf\,k}^{n+1}-\mu_{k1}C_{wf\,k-1}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,k+1}\right)\left\{\left(1+\mu_{k2}\right)C_{wf\,k+1}^{n+1}-\mu_{k2}C_{wf\,k+2}^{n+1}\right\}\right]-\\
&q_{wf\,k-\frac{1}{2}}^{n+1}\left[\omega_{f\,k}\left\{\left(1+\mu_{k3}\right)C_{wf\,k-1}^{n+1}-\mu_{k3}C_{wf\,k-2}^{n+1}\right\}\right.\\
&\left.+\left(1-\omega_{f\,k}\right)\left\{\left(1+\mu_{k4}\right)C_{wf\,k}^{n+1}-\mu_{k4}C_{wf\,k+1}^{n+1}\right\}\right]+\\
&C_{wf\,i}^{*\,n+1}Q_{wf\,i}^{n+1}-\tau_{wm/f\,i}^{n+1}\left(\omega_{m/f\,i}C_{wf\,i}^{n+1}+\left(1-\omega_{m/f\,i}\right)C_{wm\,i}^{n+1}\right)-\\
&\left(\phi_{m\,i}^{n+1}S_{wm\,i}^{n+1}K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1}-C_{wm\,i}^{n+1}\right)=\\
&\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n+1}-\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n}\right]\\
&+V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+\left(1-\phi_t\right)\rho_sC_{sf\,i}\right\}^{n+1}\lambda-\\
&V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li}+\left(1-\phi_t\right)\rho_sC_{sf\,li}\right\}\lambda_l\right]^{n+1}
\end{aligned}
\qquad \text{Eq. 4-83}
$$

Information Only

Matrix Equation:

$$\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{i+\frac{1}{2}}\left(C_{wm\,i+1}^{n+1}-C_{wm\,i}^{n+1}\right)-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{i-\frac{1}{2}}\left(C_{wm\,i}^{n+1}-C_{wm\,i-1}^{n+1}\right)+$$
$$\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{j+\frac{1}{2}}\left(C_{wm\,j+1}^{n+1}-C_{wm\,j}^{n+1}\right)-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{j-\frac{1}{2}}\left(C_{wm\,j}^{n+1}-C_{wm\,j-1}^{n+1}\right)$$
$$+\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{k+\frac{1}{2}}\left(C_{wm\,k+1}^{n+1}-C_{wm\,k}^{n+1}\right)$$
$$-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{k-\frac{1}{2}}\left(C_{wm\,k}^{n+1}-C_{wm\,k-1}^{n+1}\right)+$$
$$q_{wm\,i+\frac{1}{2}}^{n+1}\left[\omega_{m\,i+1}\left\{\left(1+\mu_{i1}\right)C_{wm\,i}^{n+1}-\mu_{i1}C_{wm\,i-1}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,i+1}\right)\left\{\left(1+\mu_{i2}\right)C_{wm\,i+1}^{n+1}-\mu_{i2}C_{wm\,i+2}^{n+1}\right\}\right]-$$
$$q_{wm\,i-\frac{1}{2}}^{n+1}\left[\omega_{m\,i}\left\{\left(1+\mu_{i3}\right)C_{wm\,i-1}^{n+1}-\mu_{i3}C_{wm\,i-2}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,i}\right)\left\{\left(1+\mu_{i4}\right)C_{wm\,i}^{n+1}-\mu_{i4}C_{wm\,i+1}^{n+1}\right\}\right]+$$
$$q_{wm\,j+\frac{1}{2}}^{n+1}\left[\omega_{m\,j+1}\left\{\left(1+\mu_{j1}\right)C_{wm\,j}^{n+1}-\mu_{j1}C_{wm\,j-1}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,j+1}\right)\left\{\left(1+\mu_{j2}\right)C_{wm\,j+1}^{n+1}-\mu_{j2}C_{wm\,j+2}^{n+1}\right\}\right]-$$
$$q_{wm\,j-\frac{1}{2}}^{n+1}\left[\omega_{m\,j}\left\{\left(1+\mu_{j3}\right)C_{wm\,j-1}^{n+1}-\mu_{j3}C_{wm\,j-2}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,j}\right)\left\{\left(1+\mu_{j4}\right)C_{wm\,j}^{n+1}-\mu_{j4}C_{wm\,j+1}^{n+1}\right\}\right]+$$
$$q_{wm\,k+\frac{1}{2}}^{n+1}\left[\omega_{m\,k+1}\left\{\left(1+\mu_{k1}\right)C_{wm\,k}^{n+1}-\mu_{k1}C_{wm\,k-1}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,k+1}\right)\left\{\left(1+\mu_{k2}\right)C_{wm\,k+1}^{n+1}-\mu_{k2}C_{wm\,k+2}^{n+1}\right\}\right]-$$
$$q_{wm\,k-\frac{1}{2}}^{n+1}\left[\omega_{m\,k}\left\{\left(1+\mu_{k3}\right)C_{wm\,k-1}^{n+1}-\mu_{k3}C_{wm\,k-2}^{n+1}\right\}\right.$$
$$\left.+\left(1-\omega_{m\,k}\right)\left\{\left(1+\mu_{k4}\right)C_{wm\,k}^{n+1}-\mu_{k4}C_{wm\,k+1}^{n+1}\right\}\right]+$$
$$C_{wm\,i}^{*\,n+1}Q_{wm\,i}^{n+1}+\tau_{wm/f\,i}^{n+1}\left(\omega_{m/f\,i}C_{wf\,i}^{n+1}+\left(1-\omega_{m/f\,i}\right)C_{wm\,i}^{n+1}\right)+$$
$$\left(\phi_{m\,i}^{n+1}S_{wm\,i}^{n+1}K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1}-C_{wm\,i}^{n+1}\right)=$$
$$\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+\left(1-\phi_t\right)\rho_s C_{sm\,i}\right\}^{n+1}-\left\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+\left(1-\phi_t\right)\rho_s C_{sm\,i}\right\}^{n}\right]$$
$$+V_{R\,i}\left\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+\left(1-\phi_t\right)\rho_s C_{sm\,i}\right\}^{n+1}\lambda-$$
$$V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{m\,i}S_{wm\,i}C_{wm\,li}+\left(1-\phi_t\right)\rho_s C_{sm\,li}\right\}\lambda_l\right]^{n+1}$$

Eq. 4-84

where

$$\mu_{i1}=\frac{\Delta x_i}{\Delta x_i+\Delta x_{i-1}},\mu_{i2}=\frac{\Delta x_{i+1}}{\Delta x_{i+1}+\Delta x_{i+2}},\mu_{i3}=\frac{\Delta x_i}{\Delta x_i+\Delta x_{i+1}},\mu_{i4}=\frac{\Delta x_{i-1}}{\Delta x_{i-1}+\Delta x_{i-2}}$$
$$\mu_{j1}=\frac{\Delta y_j}{\Delta y_j+\Delta y_{j-1}},\mu_{j2}=\frac{\Delta y_{j+1}}{\Delta y_{j+1}+\Delta y_{j+2}},\mu_{j3}=\frac{\Delta y_j}{\Delta y_j+\Delta y_{j+1}},\mu_{j4}=\frac{\Delta y_{j-1}}{\Delta y_{j-1}+\Delta y_{j-2}}$$
$$\mu_{k1}=\frac{\Delta z_k}{\Delta z_k+\Delta z_{k-1}},\mu_{k2}=\frac{\Delta z_{k+1}}{\Delta z_{k+1}+\Delta z_{k+2}},\mu_{k3}=\frac{\Delta z_k}{\Delta z_k+\Delta z_{k+1}},\mu_{k4}=\frac{\Delta z_{k-1}}{\Delta z_{k-1}+\Delta z_{k-2}}$$

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

Information Only

$$A_{2f}\,C^{n+1}_{wf\,k-2} + A_{1f}\,C^{n+1}_{wf\,k-1} + B_{2f}\,C^{n+1}_{wf\,j-2} + B_{1f}\,C^{n+1}_{wf\,j-1} + C_{2f}\,C^{n+1}_{wf\,i-2} + C_{1f}\,C^{n+1}_{wf\,i-1} + D_{1f}\,C^{n+1}_{wf\,ijk} + D_{2f}\,C^{n+1}_{wm\,ijk} + E_{1f}\,C^{n+1}_{wf\,i+1} + E_{2f}\,C^{n+1}_{wf\,i+2} + F_{1f}\,C^{n+1}_{wf\,j+1} + F_{2f}\,C^{n+1}_{wf\,j+2} + G_{1f}\,C^{n+1}_{wf\,k+1} + G_{2f}\,C^{n+1}_{wf\,k+2} = R^{n+1}_{f} \qquad \text{Eq. 4-85}$$

where

$$A_{1f} = HD^{n+1}_{wf\,k-\frac{1}{2}} - \omega_{f\,k}(1+\mu_{k3})q^{n+1}_{wf\,k-\frac{1}{2}} - \omega_{f\,k+1}\mu_{k1}q^{n+1}_{wf\,k+\frac{1}{2}}$$

$$A_{2f} = -\omega_{f\,k}\mu_{k3}q^{n+1}_{wf\,k-\frac{1}{2}}$$

$$B_{1f} = HD^{n+1}_{wf\,j-\frac{1}{2}} - \omega_{f\,j}(1+\mu_{j3})q^{n+1}_{wf\,j-\frac{1}{2}} - \omega_{f\,j+1}\mu_{j1}q^{n+1}_{wf\,j+\frac{1}{2}}$$

$$B_{2f} = -\omega_{f\,j}\mu_{j3}q^{n+1}_{wf\,j-\frac{1}{2}}$$

$$C_{1f} = HD^{n+1}_{wf\,i-\frac{1}{2}} - \omega_{f\,i}(1+\mu_{i3})q^{n+1}_{wf\,i-\frac{1}{2}} - \omega_{f\,i+1}\mu_{i1}q^{n+1}_{wf\,i+\frac{1}{2}}$$

$$C_{2f} = -\omega_{f\,i}\mu_{i3}q^{n+1}_{wf\,i-\frac{1}{2}}$$

$$E_{1f} = HD^{n+1}_{wf\,i+\frac{1}{2}} + (1-\omega_{f\,i+1})(1+\mu_{i2})q^{n+1}_{wf\,i+\frac{1}{2}} + (1-\omega_{f\,i})\mu_{i4}q^{n+1}_{wf\,i-\frac{1}{2}}$$

$$E_{2f} = -(1-\omega_{f\,i+1})(1+\mu_{i2})q^{n+1}_{wf\,i+\frac{1}{2}}$$

$$F_{1f} = HD^{n+1}_{wf\,j+\frac{1}{2}} + (1-\omega_{f\,j+1})(1+\mu_{j2})q^{n+1}_{wf\,j+\frac{1}{2}} + (1-\omega_{f\,j})\mu_{j4}q^{n+1}_{wf\,j-\frac{1}{2}}$$

$$F_{2f} = -(1-\omega_{f\,j+1})\mu_{j2}q^{n+1}_{wf\,j+\frac{1}{2}}$$

$$G_{1f} = HD^{n+1}_{wf\,k+\frac{1}{2}} + (1-\omega_{f\,k+1})(1+\mu_{k2})q^{n+1}_{wf\,k+\frac{1}{2}} + (1-\omega_{f\,k})\mu_{k4}q^{n+1}_{wf\,k-\frac{1}{2}}$$

$$G_{2f} = -(1-\omega_{f\,k+1})\mu_{k2}q^{n+1}_{wf\,k+\frac{1}{2}}$$

$$\begin{aligned} D^{n+1}_{1f} = {} & -HD^{n+1}_{wf\,k-\frac{1}{2}} - HD^{n+1}_{wf\,j-\frac{1}{2}} - HD^{n+1}_{wf\,i-\frac{1}{2}} - HD^{n+1}_{wf\,k+\frac{1}{2}} - HD^{n+1}_{wf\,j+\frac{1}{2}} - HD^{n+1}_{wf\,i+\frac{1}{2}} \\ & - q^{n+1}_{wf\,k-\frac{1}{2}}(1-\omega_{f\,k})(1+\mu_{k4}) - q^{n+1}_{wf\,j-\frac{1}{2}}(1-\omega_{f\,j})(1+\mu_{j4}) \\ & - q^{n+1}_{wf\,i-\frac{1}{2}}(1-\omega_{f\,i})(1+\mu_{i4}) + q^{n+1}_{wf\,k+\frac{1}{2}}\omega_{f\,k+1}(1+\mu_{k1}) \\ & + q^{n+1}_{wf\,j+\frac{1}{2}}\omega_{f\,j+1}(1+\mu_{j1}) + q^{n+1}_{wf\,i+\frac{1}{2}}\omega_{f\,i+1}(1+\mu_{i1}) \\ & - \frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1} \\ & + V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1}\lambda - \tau^{n+1}_{wm/f\,i}\omega_{m/f\,i} \\ & - \phi^{n+1}_{m}S^{n+1}_{wm}K^{n+1}_{wm/f\,i} \end{aligned}$$

$$D^{n+1}_{2f} = -\tau^{n+1}_{wm/f\,i}(1-\omega_{m/f\,i}) + \phi^{n+1}_{m}S^{n+1}_{wm}K^{n+1}_{wm/f\,i}$$

and

$$R^{n+1}_{f} = \frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}C_{wf\,i}\right\}^{n} - C^{*n+1}_{wf\,i}Q^{n+1}_{wf\,i} - V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li} + (1-\phi_t)\rho_s\xi_{f\,li}C_{wf\,li}\right\}^{n+1}\lambda_l\right]$$

Information Only

and the linear system of the matrix equation is:

$$A_{2m}\,C^{n+1}_{wm\,k-2} + A_{1m}\,C^{n+1}_{wm\,k-1} + B_{2m}\,C^{n+1}_{wm\,j-2} + B_{1m}\,C^{n+1}_{wm\,j-1} + C_{2m}\,C^{n+1}_{wm\,i-2} + C_{1m}\,C^{n+1}_{wm\,i-1} + D_{1m}\,C^{n+1}_{wm\,ijk} + D_{2m}\,C^{n+1}_{wf\,ijk} + E_{1m}\,C^{n+1}_{wm\,i+1} + E_{2m}\,C^{n+1}_{wm\,i+2} + F_{1m}\,C^{n+1}_{wm\,j+1} + F_{2m}\,C^{n+1}_{wm\,j+2} + G_{1m}\,C^{n+1}_{wm\,k+1} + G_{2m}\,C^{n+1}_{wm\,k+2} = R^{n+1}_{m} \qquad \text{Eq. 4-86}$$

where

$$A_{1m} = HD^{n+1}_{wm\,k-\frac{1}{2}} - \omega_{m\,k}(1+\mu_{k3})q^{n+1}_{wm\,k-\frac{1}{2}} - \omega_{m\,k+1}\mu_{k1}q^{n+1}_{wm\,k+\frac{1}{2}}$$

$$A_{2m} = -\omega_{m\,k}\mu_{k3}q^{n+1}_{wm\,k-\frac{1}{2}}$$

$$B_{1m} = HD^{n+1}_{wm\,j-\frac{1}{2}} - \omega_{m\,j}(1+\mu_{j3})q^{n+1}_{wm\,j-\frac{1}{2}} - \omega_{m\,j+1}\mu_{j1}q^{n+1}_{wm\,j+\frac{1}{2}}$$

$$B_{2m} = -\omega_{m\,j}\mu_{j3}q^{n+1}_{wm\,j-\frac{1}{2}}$$

$$C_{1m} = HD^{n+1}_{wm\,i-\frac{1}{2}} - \omega_{m\,i}(1+\mu_{i3})q^{n+1}_{wm\,i-\frac{1}{2}} - \omega_{m\,i+1}\mu_{i1}q^{n+1}_{wm\,i+\frac{1}{2}}$$

$$C_{2m} = -\omega_{m\,i}\mu_{i3}q^{n+1}_{wm\,i-\frac{1}{2}}$$

$$E_{1m} = HD^{n+1}_{wm\,i+\frac{1}{2}} + (1-\omega_{m\,i+1})(1+\mu_{i2})q^{n+1}_{wm\,i+\frac{1}{2}} + (1-\omega_{m\,i})\mu_{i4}q^{n+1}_{wm\,i-\frac{1}{2}}$$

$$E_{2m} = -(1-\omega_{m\,i+1})\mu_{i2}q^{n+1}_{wm\,i+\frac{1}{2}}$$

$$F_{1m} = HD^{n+1}_{wm\,j+\frac{1}{2}} + (1-\omega_{m\,j+1})(1+\mu_{j2})q^{n+1}_{wm\,j+\frac{1}{2}} + (1-\omega_{m\,j})\mu_{j4}q^{n+1}_{wm\,j-\frac{1}{2}}$$

$$F_{2m} = -(1-\omega_{m\,j+1})\mu_{j2}q^{n+1}_{wm\,j+\frac{1}{2}}$$

$$G_{1m} = HD^{n+1}_{wm\,k+\frac{1}{2}} + (1-\omega_{m\,k+1})(1+\mu_{k2})q^{n+1}_{wm\,k+\frac{1}{2}} + (1-\omega_{m\,k})\mu_{k4}q^{n+1}_{wm\,k-\frac{1}{2}}$$

$$G_{2m} = -(1-\omega_{m\,k+1})\mu_{k2}q^{n+1}_{wm\,k+\frac{1}{2}}$$

$$D^{n+1}_{1m} = -HD^{n+1}_{wm\,k-\frac{1}{2}} - HD^{n+1}_{wm\,j-\frac{1}{2}} - HD^{n+1}_{wm\,i-\frac{1}{2}} - HD^{n+1}_{wm\,k+\frac{1}{2}} - HD^{n+1}_{wm\,j+\frac{1}{2}} - HD^{n+1}_{wm\,i+\frac{1}{2}} - q^{n+1}_{wm\,k-\frac{1}{2}}(1-\omega_{m\,k})(1+\mu_{k4}) - q^{n+1}_{wm\,j-\frac{1}{2}}(1-\omega_{m\,j})(1+\mu_{j4}) - q^{n+1}_{wm\,i-\frac{1}{2}}(1-\omega_{m\,i})(1+\mu_{i4}) + q^{n+1}_{wm\,k+\frac{1}{2}}\omega_{m\,k+1}(1+\mu_{k1}) + q^{n+1}_{wm\,j+\frac{1}{2}}\omega_{m\,j+1}(1+\mu_{j1}) + q^{n+1}_{wm\,i+\frac{1}{2}}\omega_{m\,i+1}(1+\mu_{i1}) - \frac{V_{R\,i}}{\Delta t}\{\phi_{m\,i}S_{wm\,i} + (1-\phi_t)\rho_s\xi_{m\,i}\}^{n+1} + V_{R\,i}\{\phi_{m\,i}S_{wm\,i} + (1-\phi_t)\rho_s\xi_{m\,i}\}^{n+1}\lambda + \tau^{n+1}_{wm/f\,i}(1-\omega_{m/f\,i}) - \phi^{n+1}_{m}S^{n+1}_{wm}K^{n+1}_{wm/f\,i}$$

$$D^{n+1}_{2m} = \tau^{n+1}_{wm/f\,i}\omega_{m/f\,i} + \phi^{n+1}_{m}S^{n+1}_{wm}K^{n+1}_{wm/f\,i}$$

and

Information Only

$$R_m^{n+1} = \frac{V_{R\,i}}{\Delta t}\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s \xi_{m\,i} C_{wm\,i}\}^n - C_{wm\,i}^{*n+1} Q_{wm\,i}^{n+1} - V_{R\,i} \sum_{l=1}^{L} \left[\{\phi_{m\,i} S_{wm\,i} C_{wm\,li} + (1-\phi_t)\rho_s \xi_{m\,li} C_{wm\,li}\}^{n+1} \lambda_l\right]$$

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 Eq. 4-85 and Eq. 4-86 except that the right-hand side of the equation will be equivalent to

$$R_f^{n+1} = R_f^{n+1}(\text{Eq. 4.85}) + A_{2f}\, C_{wf\,k-2}^n + B_{2f}\, C_{wf\,j-2}^n + C_{2f}\, C_{wf\,i-2}^n + E_{2f}\, C_{wf\,i+2}^n + F_{2f}\, C_{wf\,j+2}^n + G_{2f}\, C_{wf\,k+2}^n \qquad \text{Eq. 4-87}$$

for the fracture, and

$$R_m^{n+1} = R_m^{n+1}(\text{Eq. 486}) + A_{2m}\, C_{wm\,k-2}^n + B_{2m}\, C_{wm\,j-2}^n + C_{2m}\, C_{wm\,i-2}^n + E_{2m}\, C_{wm\,i+2}^n + F_{2m}\, C_{wm\,j+2}^n + G_{2m}\, C_{wm\,k+2}^n \qquad \text{Eq. 4-88}$$

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

$$\begin{aligned}
&\left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{i+\frac{1}{2}} \left(C_{wf\,i+1}^{n+1} - C_{wf\,i}^{n+1}\right) - \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{i-\frac{1}{2}} \left(C_{wf\,i}^{n+1} - C_{wf\,i-1}^{n+1}\right) + \\
&\left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{j+\frac{1}{2}} \left(C_{wf\,j+1}^{n+1} - C_{wf\,j}^{n+1}\right) - \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{j-\frac{1}{2}} \left(C_{wf\,j}^{n+1} - C_{wf\,j-1}^{n+1}\right) \\
&+ \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{k+\frac{1}{2}} \left(C_{wf\,k+1}^{n+1} - C_{wf\,k}^{n+1}\right) \\
&- \left(\phi_f^{n+1} S_{wf}^{n+1} K_{wf}^{n+1}\right)_{k-\frac{1}{2}} \left(C_{wf\,k}^{n+1} - C_{wf\,k-1}^{n+1}\right) + \\
&q_{wf\,i+\frac{1}{2}}^{n+1} \left[(0.5+\mu_{i2}) C_{wf\,i+1}^{n+1} + (0.5-\mu_{i2}) C_{wf\,i}^{n+1}\right] - \\
&q_{wf\,i-\frac{1}{2}}^{n+1} \left[(0.5+\mu_{i1}) C_{wf\,i}^{n+1} + (0.5-\mu_{i1}) C_{wf\,i-1}^{n+1}\right] + \\
&q_{wf\,j+\frac{1}{2}}^{n+1} \left[(0.5+\mu_{j2}) C_{wf\,j+1}^{n+1} + (0.5-\mu_{j2}) C_{wf\,j}^{n+1}\right] - \\
&q_{wf\,j-\frac{1}{2}}^{n+1} \left[(0.5+\mu_{j1}) C_{wf\,j}^{n+1} + (0.5-\mu_{j1}) C_{wf\,j-1}^{n+1}\right] + \\
&q_{wf\,k+\frac{1}{2}}^{n+1} \left[(0.5+\mu_{k2}) C_{wf\,k+1}^{n+1} + (0.5-\mu_{k2}) C_{wf\,k}^{n+1}\right] - \\
&q_{wf\,k-\frac{1}{2}}^{n+1} \left[(0.5+\mu_{k1}) C_{wf\,k}^{n+1} + (0.5-\mu_{k1}) C_{wf\,k-1}^{n+1}\right] + \\
&C_{wf\,i}^{*\,n+1} Q_{wf\,i}^{n+1} - \tau_{wm/f\,i}^{n+1} (\omega_{m/f\,i} C_{wf\,i}^{n+1} + (1-\omega_{m/f\,i}) C_{wm\,i}^{n+1}) - \\
&\left(\phi_{m\,i}^{n+1} S_{wm\,i}^{n+1} K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1} - C_{wm\,i}^{n+1}\right) =
\end{aligned} \qquad \text{Eq. 4-89}$$

Information Only

$$\frac{V_{R\,i}}{\Delta t}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_s C_{sf\,i}\right\}^{n+1}-\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_s C_{sf\,i}\right\}^{n}\right]$$
$$+V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,i}+(1-\phi_t)\rho_s C_{sf\,i}\right\}^{n+1}\lambda-$$
$$V_{R\,i}\sum_{l=1}^{L}\left[\left\{\phi_{f\,i}S_{wf\,i}C_{wf\,li}+(1-\phi_t)\rho_s C_{sf\,li}\right\}\lambda_l\right]^{n+1}$$

Matrix Equation

$$\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{i+\frac{1}{2}}\left(C_{wm\,i+1}^{n+1}-C_{wm\,i}^{n+1}\right)-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{i-\frac{1}{2}}\left(C_{wm\,i}^{n+1}-C_{wm\,i-1}^{n+1}\right)+$$
$$\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{j+\frac{1}{2}}\left(C_{wm\,j+1}^{n+1}-C_{wm\,j}^{n+1}\right)-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{j-\frac{1}{2}}\left(C_{wm\,j}^{n+1}-C_{wm\,j-1}^{n+1}\right)$$
$$+\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{k+\frac{1}{2}}\left(C_{wm\,k+1}^{n+1}-C_{wm\,k}^{n+1}\right)$$
$$-\left(\phi_m^{n+1}S_{wm}^{n+1}K_{wm}^{n+1}\right)_{k-\frac{1}{2}}\left(C_{wm\,k}^{n+1}-C_{wm\,k-1}^{n+1}\right)+$$
$$q_{wm\,i+\frac{1}{2}}^{n+1}\left[(0.5+\mu_{i2})C_{wm\,i+1}^{n+1}+(0.5-\mu_{i2})C_{wm\,i}^{n+1}\right]-$$
$$q_{wm\,i-\frac{1}{2}}^{n+1}\left[(0.5+\mu_{i1})C_{wm\,i}^{n+1}+(0.5-\mu_{i1})C_{wm\,i-1}^{n+1}\right]+$$
$$q_{wm\,j+\frac{1}{2}}^{n+1}\left[\left(0.5+\mu_{j2}\right)C_{wm\,j+1}^{n+1}+\left(0.5-\mu_{j2}\right)C_{wm\,j}^{n+1}\right]-$$
$$q_{wm\,j-\frac{1}{2}}^{n+1}\left[\left(0.5+\mu_{j1}\right)C_{wm\,j}^{n+1}+\left(0.5-\mu_{j1}\right)C_{wm\,j-1}^{n+1}\right]+ \qquad \text{Eq. 4-90}$$
$$q_{wm\,k+\frac{1}{2}}^{n+1}\left[(0.5+\mu_{k2})C_{wm\,k+1}^{n+1}+(0.5-\mu_{k2})C_{wm\,k}^{n+1}\right]-$$
$$q_{wm\,k-\frac{1}{2}}^{n+1}\left[(0.5+\mu_{k1})C_{wm\,k}^{n+1}+(0.5-\mu_{k1})C_{wm\,k-1}^{n+1}\right]+$$
$$C_{wm\,i}^{*\,n+1}Q_{wm\,i}^{n+1}+\tau_{wm/f\,i}^{n+1}(\omega_{m/f\,i}C_{wf\,i}^{n+1}+(1-\omega_{m/f\,i})C_{wm\,i}^{n+1})+$$
$$\left(\phi_{m\,i}^{n+1}S_{wm\,i}^{n+1}K_{wm/f\,i}^{n+1}\right)\left(C_{wf\,i}^{n+1}-C_{wm\,i}^{n+1}\right)=$$
$$\frac{V_{R\,i}}{\Delta t}\left[\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+(1-\phi_t)\rho_s C_{sm\,i}\}^{n+1}-\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+(1-\phi_t)\rho_s C_{sm\,i}\}^{n}\right]$$
$$+V_{R\,i}\{\phi_{m\,i}S_{wm\,i}C_{wm\,i}+(1-\phi_t)\rho_s C_{sm\,i}\}^{n+1}\lambda-$$
$$V_{R\,i}\sum_{l=1}^{L}\left[\{\phi_{m\,i}S_{wm\,i}C_{wm\,li}+(1-\phi_t)\rho_s C_{sm\,li}\}\lambda_l\right]^{n+1}$$

where

$$\mu_{i1}=\frac{1}{4}\frac{\Delta x_{i-1}-\Delta x_i}{\Delta x_{i-1}+\Delta x_i}\text{ and }\mu_{i2}=\frac{1}{4}\frac{\Delta x_i-\Delta x_{i+1}}{\Delta x_i+\Delta x_{i+1}}$$
$$\mu_{j1}=\frac{1}{4}\frac{\Delta y_{j-1}-\Delta y_j}{\Delta y_{j-1}+\Delta y_j}\text{ and }\mu_{j2}=\frac{1}{4}\frac{\Delta y_j-\Delta y_{j+1}}{\Delta y_j+\Delta y_{j+1}}$$
$$\mu_{k1}=\frac{1}{4}\frac{\Delta z_{k-1}-\Delta z_k}{\Delta z_{k-1}+\Delta z_k}\text{ and }\mu_{k2}=\frac{1}{4}\frac{\Delta z_k-\Delta z_{k+1}}{\Delta z_k+\Delta z_{k+1}}$$

A similar linear system as in Eq. 4-80 and Eq. 4-82 is formed, in which

Information Only

$$A_f = HD^{n+1}_{wf\,k-\frac{1}{2}} - (0.5+\mu_{k1})q^{n+1}_{wf\,k-\frac{1}{2}}$$

$$A_m = HD^{n+1}_{wm\,k-\frac{1}{2}} - (0.5+\mu_{k1})q^{n+1}_{wm\,k-\frac{1}{2}}$$

$$B_f = HD^{n+1}_{wf\,j-\frac{1}{2}} - (0.5+\mu_{j1})q^{n+1}_{wf\,j-\frac{1}{2}}$$

$$B_m = HD^{n+1}_{wm\,j-\frac{1}{2}} - (0.5+\mu_{j1})q^{n+1}_{wm\,j-\frac{1}{2}}$$

$$C_f = HD^{n+1}_{wf\,i-\frac{1}{2}} - (0.5+\mu_{i1})q^{n+1}_{wf\,i-\frac{1}{2}}$$

$$C_m = HD^{n+1}_{wm\,i-\frac{1}{2}} - (0.5+\mu_{i1})q^{n+1}_{wm\,i-\frac{1}{2}}$$

$$E_f = HD^{n+1}_{wf\,i+\frac{1}{2}} + (0.5+\mu_{i2})q^{n+1}_{wf\,i+\frac{1}{2}}$$

$$E_m = HD^{n+1}_{wm\,i+\frac{1}{2}} + (0.5+\mu_{i2})q^{n+1}_{wm\,i+\frac{1}{2}}$$

$$F_f = HD^{n+1}_{wf\,j+\frac{1}{2}} + (0.5+\mu_{j2})q^{n+1}_{wf\,j+\frac{1}{2}}$$

$$F_m = HD^{n+1}_{wm\,j+\frac{1}{2}} + (0.5+\mu_{j2})q^{n+1}_{wm\,j+\frac{1}{2}}$$

$$G_f = HD^{n+1}_{wf\,k+\frac{1}{2}} + (0.5+\mu_{k2})q^{n+1}_{wf\,k+\frac{1}{2}}$$

$$G_m = HD^{n+1}_{wm\,k+\frac{1}{2}} + (0.5+\mu_{k2})q^{n+1}_{wm\,k+\frac{1}{2}}$$

and

$$\begin{aligned} D_{1f} = &-HD^{n+1}_{wf\,k-\frac{1}{2}} - HD^{n+1}_{wf\,j-\frac{1}{2}} - HD^{n+1}_{wf\,i-\frac{1}{2}} - HD^{n+1}_{wf\,k+\frac{1}{2}} - HD^{n+1}_{wf\,j+\frac{1}{2}} - HD^{n+1}_{wf\,i+\frac{1}{2}} \\ &-(0.5+\mu_{k1})q^{n+1}_{wf\,k-\frac{1}{2}} - (0.5+\mu_{j1})q^{n+1}_{wf\,j-\frac{1}{2}} - (0.5+\mu_{i1})q^{n+1}_{wf\,i-\frac{1}{2}} \\ &+(0.5-\mu_{k2})q^{n+1}_{wf\,k+\frac{1}{2}} + (0.5-\mu_{j2})q^{n+1}_{wf\,j+\frac{1}{2}} + (0.5-\mu_{i2})q^{n+1}_{wf\,i+\frac{1}{2}} \\ &-\frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1} + V_{R\,i}\left\{\phi_{f\,i}S_{wf\,i} + (1-\phi_t)\rho_s\xi_{f\,i}\right\}^{n+1}\lambda \\ &-\tau^{n+1}_{wm/f\,i}\omega_{m/f\,i} - \phi^{n+1}_m S^{n+1}_{wm} K^{n+1}_{wm/f\,i} \end{aligned}$$

$$\begin{aligned} D_{1m} = &-HD^{n+1}_{wm\,k-\frac{1}{2}} - HD^{n+1}_{wm\,j-\frac{1}{2}} - HD^{n+1}_{wm\,i-\frac{1}{2}} - HD^{n+1}_{wm\,k+\frac{1}{2}} - HD^{n+1}_{wm\,j+\frac{1}{2}} - HD^{n+1}_{wm\,i+\frac{1}{2}} \\ &-(0.5+\mu_{k1})q^{n+1}_{wm\,k-\frac{1}{2}} - (0.5+\mu_{j1})q^{n+1}_{wm\,j-\frac{1}{2}} - (0.5+\mu_{i1})q^{n+1}_{wm\,i-\frac{1}{2}} \\ &+(0.5-\mu_{k2})q^{n+1}_{wm\,k+\frac{1}{2}} + (0.5-\mu_{j2})q^{n+1}_{wm\,j+\frac{1}{2}} + (0.5-\mu_{i2})q^{n+1}_{wm\,i+\frac{1}{2}} \\ &-\frac{V_{R\,i}}{\Delta t}\left\{\phi_{m\,i}S_{wm\,i} + (1-\phi_t)\rho_s\xi_{m\,i}\right\}^{n+1} + V_{R\,i}\left\{\phi_{m\,i}S_{wm\,i} + (1-\phi_t)\rho_s\xi_{m\,i}\right\}^{n+1}\lambda \\ &+\tau^{n+1}_{wm/f\,i}(1-\omega_{m/f\,i}) - \phi^{n+1}_m S^{n+1}_{wm} K^{n+1}_{wm/f\,i} \end{aligned}$$

Information Only

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

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

$$R_f^{n+1} = \frac{V_{R\,i}}{\Delta t}\left\{\phi_{f\,i} S_{wf\,i} C_{wf\,i} + (1-\phi_t)\rho_s \xi_{f\,i} C_{wf\,i}\right\}^n - C_{wf\,i}^{*n+1} Q_{wf\,i}^{n+1}$$
$$-V_{R\,i}\sum_{l=1}^{L}\left[\phi_{f\,i} S_{wf\,i} C_{wf\,li} + (1-\phi_t)\rho_s \xi_{f\,li} C_{wf\,li}\right]^{n+1}\lambda_l$$

$$R_m^{n+1} = \frac{V_{R\,i}}{\Delta t}\{\phi_{m\,i} S_{wm\,i} C_{wm\,i} + (1-\phi_t)\rho_s \xi_{m\,i} C_{wm\,i}\}^n - C_{wm\,i}^{*} Q_{wm\,i}^{n+1}$$
$$-V_{R\,i}\sum_{l=1}^{L}[\phi_{m\,i} S_{wm\,i} C_{wm\,li} + (1-\phi_t)\rho_s \xi_{m\,li} C_{wm\,li}]^{n+1}\lambda_l$$

It is worth mentioning that in all the above methods no fictitious or interior 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 domain with negligible or no transport near boundaries. In the cases when there is an active flow field near the boundaries, such treatment (one-point upstream) may lead to a first-order method even if two-point upstream weighting is used in the interior.

The transport equations (Eq. 4-40 and Eq. 4-41) have $N \times G$ unknowns for single porosity and $2N \times G$ unknowns for dual-porosity and dual-permeability formulations, where N is the number of equations in each grid block and G is the number of grid blocks in the simulated spatial domain. The system of governing partial differential equations (two for each transported constituent) is strongly coupled one to the other because of the strong contribution from parental decay to the concentration of the immediate daughter. Consequently, the sequential method is used to solve the system implicitly (if the implicit method is used). In the sequential method, the solution proceeds progressively from the top to the bottom of each radioactive chain. Therefore, the contribution to any daughter from parental decay will be immediately calculable.

The matrix resulting from one-point upstream winding and the mid-point weighting resembles the following for a two-dimensional system of 3 × 3 grid blocks

$$\begin{bmatrix}
f\,m & f\,0 & 0\,0 & f\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 \\
f\,m & 0\,m & 0\,0 & 0\,m & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 \\
f\,0 & f\,m & f\,0 & 0\,0 & f\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 \\
0\,m & f\,m & 0\,m & 0\,0 & 0\,m & 0\,0 & 0\,0 & 0\,0 & 0\,0 \\
0\,0 & f\,0 & f\,m & 0\,0 & 0\,0 & f\,0 & 0\,0 & 0\,0 & 0\,0 \\
0\,0 & 0\,m & f\,m & 0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,0 & 0\,0 \\
f\,0 & 0\,0 & 0\,0 & f\,m & f\,0 & 0\,0 & f\,0 & 0\,0 & 0\,0 \\
0\,m & 0\,0 & 0\,0 & f\,m & 0\,m & 0\,0 & 0\,m & 0\,0 & 0\,0 \\
0\,0 & f\,0 & 0\,0 & f\,0 & f\,m & f\,0 & 0\,0 & f\,0 & 0\,0 \\
0\,0 & 0\,m & 0\,0 & 0\,m & f\,m & 0\,m & 0\,0 & 0\,m & 0\,0 \\
0\,0 & 0\,0 & f\,0 & 0\,0 & f\,0 & f\,m & f\,0 & 0\,0 & f\,0 \\
0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,m & f\,m & 0\,m & 0\,0 & 0\,m \\
0\,0 & 0\,0 & 0\,0 & f\,0 & 0\,0 & f\,0 & f\,m & 0\,0 & 0\,0 & f\,0 \\
0\,0 & 0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,m & f\,m & 0\,0 & 0\,0 & 0\,m \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & f\,0 & 0\,0 & 0\,0 & f\,m & f\,0 & 0\,0 \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,0 & f\,m & 0\,m & 0\,0 \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & f\,0 & 0\,0 & f\,0 & f\,m & f\,0 \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,m & f\,m & 0\,m \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & f\,0 & 0\,0 & f\,0 & f\,m \\
0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,0 & 0\,m & 0\,0 & 0\,m & f\,m
\end{bmatrix}$$

Information Only

Since two-point upstream winding uses information from the grid block further in the upstream direction, the band width of the matrix will be doubled compared with the previous one. Also the matrix will be less sparse. For the same system of 3×3, the numerical matrix is:

$$\begin{bmatrix}
f & m & f & 0 & f & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f & m & 0 & m & 0 & m & 0 & m & 0 & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 \\
f & 0 & f & m & f & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 \\
0 & m & f & m & 0 & m & 0 & 0 & 0 & m & 0 & 0 & 0 & 0 & 0 & m & 0 \\
f & 0 & f & 0 & f & m & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 & f & 0 & \\
0 & m & 0 & m & f & m & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 & 0 & 0 & m \\
f & 0 & 0 & 0 & 0 & f & m & f & 0 & f & 0 & f & 0 & 0 & 0 & & \\
0 & m & 0 & 0 & 0 & f & m & 0 & m & 0 & m & 0 & m & 0 & 0 & & \\
0 & 0 & f & 0 & 0 & f & 0 & f & m & f & 0 & 0 & 0 & f & 0 & 0 & \\
0 & 0 & 0 & m & 0 & 0 & 0 & m & f & m & 0 & m & 0 & 0 & 0 & m & 0 \\
0 & 0 & 0 & f & 0 & f & 0 & f & 0 & f & m & 0 & 0 & 0 & f & 0 & \\
0 & 0 & 0 & 0 & m & 0 & m & 0 & m & f & m & 0 & 0 & 0 & 0 & m & \\
f & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 & f & m & f & 0 & f & 0 & \\
0 & m & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 & f & m & 0 & m & 0 & m & \\
0 & 0 & f & 0 & 0 & 0 & f & 0 & 0 & f & 0 & f & m & f & 0 & & \\
0 & 0 & 0 & m & 0 & 0 & 0 & 0 & m & 0 & 0 & 0 & m & f & m & 0 & m \\
0 & 0 & 0 & f & 0 & 0 & 0 & 0 & f & 0 & f & 0 & f & 0 & f & m & \\
0 & 0 & 0 & 0 & m & 0 & 0 & 0 & 0 & m & 0 & m & 0 & m & f & m &
\end{bmatrix}$$

where f refers to fracture entry, m refers to matrix entries, and 0 refers to zero entry.

Standard Gaussian elimination 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×G is inverted instead of G×G matrix, where IBW is the band width.

4.5 Other important physical and chemical aspects of the modeling

4.5.1 Treatment of mobilization involving colloidal materials

In order to minimize unnecessary computations, dissolved and colloidal mobilized species have been combined for transport within NUTS. This is justified because molecular diffusion and sorption are the only physical processes that could cause differences in the modeled transport of dissolved versus colloidal species, and since they have been set to zero, the species will transport identically. The combination of dissolved and colloidal species is performed prior to NUTS in an ALGEBRA/PANEL calculation. In this calculation, four types of colloid-related mobilization are considered, as follows: (i) condensation of hydrolyzed actinide ions into intrinsic colloids, and sorption onto already existent colloidal materials of (ii) humic, (iii) microbial, and (iv) mineral origin. Maximum concentrations for these four different types of colloidal particulates are *added* to the maximum concentration for dissolution for each of the radionuclides and (sampled) oxidation state indices. The sum of the five maximum concentrations is then used in a single dissolution-like computation that estimates the net mobilization due to colloids and dissolution combined.

Information Only

4.5.2 Lumped radionuclides

To keep computational resources at a reasonable level, and because several of the radionuclides in the inventory have been shown to be unimportant in terms of release calculations, inventory radionuclides have been (i) combined, where possible, or (ii) omitted, where justifiable, so as to reduce the list of 29 actual radionuclides to be monitored to 5 "equivalent" radionuclides. The rules were to combine the effects of some radionuclides (by adding their inventories), providing they have similar half-lives, and to omit 10 others whose effects on release values have been shown to be negligible (less than 0.01 normalized EPA units). For example, ^{233}U and ^{234}U are 30 to 100 times more important than other uranium isotopes in terms of normalized EPA releases. Thus, ^{235}U, ^{236}U, and ^{238}U can be ignored without a significant impact on release calculations. ^{233}U and ^{234}U comprise less than 1% of the total uranium inventory mass. To account for that, the solubility of uranium is decreased by a factor of 100. Lastly, since the half-lives are similar, ^{233}U and ^{234}U are combined into a single equivalent radionuclide. Thus, 5 radionuclides have thereby been reduced to one equivalent radioisotope. Similarly, ^{229}Th is absorbed into ^{230}Th, their solubility reduced by 1000 to account for their representation in the inventory, and ^{232}Th is ignored altogether on account of its negligible effect on release values. In every case, revisions favored overestimating released materials.

In compliance calculations, NUTS is exercised first using the five equivalent radionuclides. If releases do not approach EPA limits, results are permitted to stand. If releases approach EPA limits, NUTS will be re-exercised with actual inventories. The physical properties of the 5 "equivalent" radionuclides are computed by the same ALGEBRA/PANEL run that combines maximum concentrations due to dissolution and colloidal mobilization (see Section 4.5.1)

4.5.3 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:

$$D_m^*(T) = D_{m\,ref}^* \frac{T}{T_{ref}} \frac{\mu_{ref}}{\mu(T)} \qquad \text{Eq. 4-91}$$

where

$D_{m\,ref}^*$ 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},

$\mu(T)$ is solvent viscosity at temperature T, and

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

Information Only

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.

$$k_{d2} = k_{d1}\, exp\left[\frac{\xi(T_2 - T_1)}{T_2 T_1}\right], \quad \text{where } \xi = \frac{\Delta H^0}{R} \qquad \text{Eq. 4-92}$$

In the above formulation, k_{d1} and k_{d2} are equilibrium partition constants at temperatures T_1 and T_2, respectively. H^0 is the enthalpy of sorption at reference temperature T^0, and R is the universal gas constant.

4.5.4 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 (Volatile Organic Compounds) + H_2O, 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

$$K_{LG} = \frac{C_L}{C_G} \qquad \text{Eq. 4-93}$$

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:

Information Only

$$\frac{C_L}{C_G} = A - BT \qquad \text{Eq. 4-94}$$

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:

$$\begin{gathered}\Delta_x\left[\phi_f S_{Gf} K_{Gf} \Delta_x C_{nGf}\right] + \Delta_y\left[\phi_f S_{Gf} K_{Gf} \Delta_y C_{nGf}\right] + \Delta_z\left[\phi_f S_{Gf} K_{Gf} \Delta_z C_{nGf}\right] \\ +\Delta_x q_{Gf} C_{nGf} + \Delta_y q_{Gf} C_{nGf} + \Delta_z q_{Gf} C_{nGf} + C^*_{nGf} Q_{Gf} \\ - \tau_{Gm/f} C_{nGm/f} - \phi_m S_{Gm} K_{Gm/f}\left(C_{nGf} - C_{nGm}\right) = \\ \frac{V_R}{\Delta t}\Delta_t\left[\phi_f S_{Gf} C_{nGf} + (1-\phi_t)\rho_s C_{nsf} + \phi_f (1-S_{Gf}) K_{LG} C_{nGf}\right] \\ +V_R\left[\phi_f S_{Gf} C_{nGf} + (1-\phi_t)\rho_s C_{nsf} + \phi_f (1-S_{Gf}) K_{LG} C_{nGf}\right]\lambda_n \\ -V_R \sum_{j=1}^{J}\left[\left\{\phi_f S_{Gf} C_{nGfpj} + (1-\phi_t)\rho_s C_{nsfpj} + \phi_f (1-S_{Gf}) K_{LG} C_{nGfpj}\right\}\lambda_{npj}\right] \\ n = 1,2,\cdots,N\end{gathered} \qquad \text{Eq. 4-95}$$

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

$$\begin{gathered}\Delta_x[\phi_m S_{Gm} K_{Gm} \Delta_x C_{nGm}] + \Delta_y[\phi_m S_{Gm} K_{Gm} \Delta_y C_{nGm}] + \Delta_z[\phi_m S_{Gm} K_{Gm} \Delta_z C_{nGm}] \\ +\Delta_x q_{Gm} C_{nGm} + \Delta_y q_{Gm} C_{nGm} + \Delta_z q_{Gm} C_{nGm} + C^*_{nGm} Q_{Gm} \\ + \tau_{Gm/f} C_{nGm/f} + \phi_m S_{Gm} K_{Gm/f}\left(C_{nGf} - C_{nGm}\right) = \\ \frac{V_R}{\Delta t}\Delta_t[\phi_m S_{Gm} C_{nGm} + (1-\phi_t)\rho_s C_{nsm} + \phi_m (1-S_{Gm}) K_{LG} C_{nGm}] \\ +V_R[\phi_m S_{Gm} C_{nGm} + (1-\phi_t)\rho_s C_{nsm} + \phi_m (1-S_{Gm}) K_{LG} C_{nGm}]\lambda_n \\ -V_R \sum_{j=1}^{J}[\{\phi_m S_{Gm} C_{nGmpj} + (1-\phi_t)\rho_s C_{nsmpj} + \phi_m (1-S_{Gm}) K_{LG} C_{nGmpj}\}\lambda_{npj}] \\ n = 1,2,\cdots,N\end{gathered} \qquad \text{Eq. 4-96}$$

where

C_{nG} = solute concentration of component n in the gas phase[kg/m^3],

C^*_{nG} = injected or produced solute concentration of component n [kg/m^3],

C_{ns} = sorbate concentration of component n [kg/kg],

K = dispersion coefficient [m^3/s],

K_{LG} = gas-liquid equilibrium constant [dimensionless],

Information Only

N = total number of radionuclides,

Q = gas injection/production rate [m^3/s],

q = gas interfacial volumetric rate [m^3/s],

S = gas saturation, fraction,

t = time [s],

V_R = grid block volume [m^3],

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

4.5.5 Precipitate models

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

Information Only

$$-\nabla\bullet v_w C_l + Q^* + q = \frac{\partial}{\partial t}(\phi S_w C_l) + (\phi S_w C_l)\lambda_l - \phi S_w \sum_{j=1}^{J_l} C_{l,j}\lambda_j \qquad \text{Eq. 4-97}$$

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 Eq. 4-97, 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 WIPP 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:

$$C_l(x, y, 0) = f_l(x, y) \qquad 0 < x < X, 0 < y < Y \qquad \text{Eq. 4-98}$$

where $f_l(x, y)$ represents the initial concentration of the l^{th} radionuclide and is understood to be known at every point in the grid domain. For all calculations, the initial concentrations outside the waste region 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:

$$\left.\begin{aligned} C_l(x, y, 0) &= C_{ls}^0 \\ P_l^0 &= T_l^0 - \phi S_w C_{ls}^0 \end{aligned}\right\} \quad if\ T_l^0 \geq M_l^0 \qquad \text{Eq. 4-99}$$

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_{ls}^0 (kg/m^3 brine) and where, P_l^0 (kg/m^3 bulk

Information Only

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,

$$\left.\begin{aligned} C_l(x,y,0) &= \frac{T_l^0}{\phi S_w} \\ P_l^0 &= 0 \end{aligned}\right\} \quad if\ T_l^0 < M_l^0 \qquad \text{Eq. 4-100}$$

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 radionuclides are consumed. The capacity of the brines to dissolve radionuclides 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 isotope 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 radionuclide concentrations should exceed their solubility limits, NUTS will precipitate the excess 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 the solute or until the precipitate is exhausted. The precipitate calculations can be done with two different numerical implementations[1]:

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 Eq. 4-97 (with $q = 0$) and the solid phase equation

$$\sum_{j=1}^{J_l} \lambda_{lj} P_{lj} - \lambda_l P_l = \frac{\partial P_l}{\partial t} \qquad \text{Eq. 4-101}$$

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

$$(C_l^{n+1})_{adjusted} = C_{ls}^{n+1} \qquad \text{and,} \qquad \text{Eq. 4-102}$$

$$(P_l^{n+1})_{adjusted} = P_l^{n+1}\{\text{Eq. } 4-101\} + (\phi S_w)^{n+1}\left|C_l^{n+1} - C_{ls}^{n+1}\right| \qquad \text{Eq. 4-103}$$

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

$$(C_l^{n+1})_{adjusted} = C_l^{n+1}\{\text{Eq. } 4-97\} + \frac{P_l^{n+1}\{\text{Eq. } 4-101\}}{(\phi S_w)^{n+1}} \qquad \text{Eq. 4-104}$$

where $n + 1$ refer to the new time step. From Eq. 4-102 through Eq. 4-104, 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.

[1] Even though dispersion and adsorption are neglected in the current use (Eq. 4-97), the numerical implementation of the precipitation model is taking into consideration the NUTS capabilities described in Eq. 4-40 and Eq. 4-41.

Information Only

Even though the explicit method is not-iterative and precipitate calculation requires almost no additional computational time, it may suffer from two drawbacks:

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 Eq. 4-97 subjected to the constraint,

$$C_l^{n+1} \leq C_{ls}^{n+1} \qquad \text{Eq. 4-105}$$

and the solid phase equation

$$-q + \sum_{j=1}^{J_l} \lambda_{lj} P_{lj} - \lambda_l P_l = \frac{\partial P_l}{\partial t} \qquad \text{Eq. 4-106}$$

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

The system of Eq. 4-97, Eq. 4-105, and Eq. 4-106 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:

$$P_l^k = P_l^n \qquad \text{Eq. 4-107}$$

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

2) For each grid block i with $P_l^k > 0$, and assuming that enough precipitate is available to maintain the solubility limit, set Dirichlet condition by the penalty method (Becker et al. 1981) at a concentration equal to the solubility limit. Then solve Eq. 4-97 for the concentration, C_l^{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 $P_l^k > 0$) by solving for the residual of Eq. 4-97 as follows:

$$q_l^{k+1} = \frac{\partial}{\partial t}(\phi S_w C_l) + \nabla \bullet v_w C_l - Q^* + (\phi S_w C_l)\lambda_l - \phi S_w \sum_{p=1}^{J_l} C_{l,p}\lambda_p \qquad \text{Eq. 4-108}$$

Eq. 4-108 is solved in the discretized form. Since all the entries for this equation are available after the solution of Eq. 4-97, the value of the residual (q_l^{k+1}) can be directly solved for by substituting of the solubility limit for the concentration of the grid block i and the computed

Information Only

concentration (from Eq. 4-97) for the surrounding grid blocks (five-point stencil in two-dimensional grid).

4) Evaluate the new precipitate, P_l^{k+1} from Eq. 4-106, using q_l^{k+1} from step 3.

5) Check the sign of the P_l^{k+1}.

   a) If $P_l^{k+1} < 0$, then the precipitate available is not enough to maintain the solubility limit. Therefore, set $P_l^{k+1} = 0$, $q_l^{k+1} = P_l^k/\Delta t$, and turn Dirichlet condition off in that grid block.

   b) If $P_l^{k+1} \geq 0$, 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 ($P_l^n = 0$) may acquire precipitate due to dissolved daughter isotopes in growth, compare the computed concentration, C_l^{k+1} in each grid block with the solubility limit. If $C_l^{k+1} > C_{ls}^{n+1}$ then,

$$P_l^{k+1} = \phi S_w \left(C_l^{k+1} - C_{ls}^{n+1}\right) \qquad \text{Eq. 4-109}$$

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

6) Evaluate the change in the precipitate as follows:

$$\Delta P_l = P_l^{k+1} - P_l^k \qquad \text{Eq. 4-110}$$

7) If $|\Delta P_l| \leq \varepsilon$, 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:

$$C_{ls}^{n+1} = F^{n+1} C_e \qquad \text{Eq. 4-111}$$

where, F is the mole fraction of the isotope and C_e is the element solubility limit (kg/m^3 brine)

Knowing that F is a function of C_l ($F = f(C_l)$), the system of Eq. 4-97 and Eq. 4-105 becomes highly nonlinear[2] and may not converge in the implicit treatment (in the explicit treatment, F^{n+1} is known from C_l^{n+1}). One recommended way to overcome this problem is by linearization of Eq. 4-108 through assigning a solubility limit for each isotope. In the compliance calculations, both isotopes ^{238}Pu and ^{239}Pu share the solubility of the plutonium and may require such a kind of treatment

[2] 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.

Information Only

5.0 INPUT AND OUTPUT FILES

In compliance calculations, NUTS isotope transport simulation relies upon multiple files as input: one binary CDB output and one ASCII input files from BRAGFLO; one binary CDB output from PANEL; and one binary CDB output for undisturbed scenario from NUTS. One NUTS input ASCII file is used for NUTS' input data as input. The "input" CDB files contain static parameters related to unit conversions, atomic properties, radionuclide inventory, and concentration limits, and historic parameters related to brine flow field. These files are typically created by the WIPP PA codes MATSET, PANEL, and BRAGFLO. For NUTS intrusion (INT) mode transport simulations (at intrusion times except 350 and 1000 years), the undisturbed transport simulation CDB files by NUTS are used for input data.

5.1 NUTS input ASCII file

This file consists of pieces of information provided by different procedures and controlled by the flags listed during NUTS's initial interrogation. 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 are a sequential description of the ASCII input file that can be used to calculate radionuclide transport through porous media (no fracture) with single porosity (this is WIPP PA transport calculation condition). Sample ASCII input files for SCN, ISO, and INT run modes are given in Appendix A.

5.1.1 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 , but < the total number of the grid blocks.*

Information Only

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 , but < 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 ^{238}U 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, but < 8.*

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[3]. *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

[3] Throughout this manual, isotope, radioactive isotope, radionuclide, 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.

Information Only

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 elements, 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 , but < 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.*

END DO

Note: A classical definition of the element is a substance that cannot 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.

Information Only

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 , but < 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 5.1.1. *Accepts an integer >0 , but < 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 5.1.1. *Accepts an integer* >0 , but < 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

Information Only

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

Line 22. MORDER:

MORDER: Order of the method. The following options are allowed. *Accepts an integer >0, but ≤ 4.*

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.

Line 23. Descriptor: Degree of implicitness.

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. *Accepts a real number ≥ 0, but ≤ 1.*

BETA = 1 , for fully implicit.

BETA = 0 , for fully explicit.

BETA = 1/2 , for Crank-Nicolson.

5.1.2 Single porosity matrix controls and flags

Matrix Sorption

Line 25. Descriptor: Is matrix sorption required (Y/N). *Accepts a character string up to 100 characters in length as a descriptive line*

Line 26. 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 27. MAT_ISOTHERM

MAT_ISOTHERM: one of three types of matrix sorption isotherm, namely:

Linear isotherm. *Accepts L or l.*

Information Only

Freundlich isotherm. *Accepts F, or f.*

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

DO I = 1, NUCLIDE

Line 28. 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 29. 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 30. MDISPREQ

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

Line 31. 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 32. 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 33. 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 34. MSTATUSINJ

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

Information Only

Line 35. 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 36. 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 37. Descriptor: Is manual initialization present in the matrix (T/F) *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 38. 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 compliance 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 39. 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 40. 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 41. Descriptor: Variables to be printed in an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 42. (IPRNTMA(I), I = 1, NMVARA)

Information Only

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, kg

IPRNTMA(2): Matrix total mass of precipitated constituents in each grid block, kg

IPRNTMA(3): Matrix total mass of dissolved, precipitated and sorbed constituents in each grid block, kg

IPRNTMA(4): Matrix total dissolved curies of all constituents in each grid block, curies

IPRNTMA(5): Matrix total precipitated curies of all constituents in each grid block, curies

IPRNTMA(6): Matrix total dissolved, precipitated, and sorbed curies of all constituents in each grid block, curies

IPRNTMA(7): Matrix volumetric concentration of the dissolved isotope, kg/(m^3 brine).

IPRNTMA(8): Matrix dissolved mass in the brine of a certain constituent, kg.

IPRNTMA(9): Matrix precipitated mass of a certain constituent, kg.

IPRNTMA(10): Matrix soil base concentration of the isotope, mg/(kg soil).

IPRNTMA(11): Matrix curies of volumetric concentration of a certain constituent, curies/(m^3 brine).

IPRNTMA(12): Matrix curies of dissolved mass in the brine of a certain constituent, curies.

IPRNTMA(13): Matrix curies of precipitated mass of a certain constituent, curies.

IPRNTMA(14): Matrix curies of dissolved, precipitated and sorbed mass of a certain constituent, curies.

Line 43. 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 44. 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 45. Descriptor: Matrix variables to be printed in a BINARY file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Information Only

Line 46. (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 44 is "Y" or "y'), then:

Line 47. Descriptor: Ranges to be printed in an ASCII file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 48. NO_ASC_MATRIX_RANGES

NO_ASC_MATRIX_RANGES: Number of ranges to be printed. *Accepts an integer > 0 , but < the total number of grid blocks.*

DO I = 1, NO_ASC_MATRIX_RANGES

Line 49. IARANGE(I) ,MASTRTI(I) ,MASTRTJ(I) ,MASTRTK(I), MAENDI(I), MAENDJ(I), MAENDK(I)

IARANGE: Range number. *Accepts an integer > 0, but <* NO_ASC_MATRIX_RANGES

MASTRTI,J,KI,J,K: Indices of the grid block starting the range. *Accepts an integer > 0 and I < NX, J < NY, K < NZ*

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

Line 50. (IPRNSMA(I,J), J = 1, NMVARA)

IPRNSMA: an integer array similar to IPRNTMA

END DO

5.1.3 k_d temperature dependency information

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

Line 51. Descriptor: Component name, reference temperature, and sorption exponential coefficient. *Accepts a character string up to 100 characters in length as a descriptive line.*

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

DO I = 1, NUCLIDE

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

Line 52. NAME(I), REFTEMPKD(I), ADSEXPCOEFF(I)

Information Only

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.*

ADSEXPCOEFF: Sorption exponential coefficient. *Upper and lower bounds are not provided.*

END DO

5.1.4 Molecular diffusion temperature dependency flags

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

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

Line 53. 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 54. 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

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

Line 56. IPRFRQA, TIMEASCMAX

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.*

TIMEASCMAX: 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 , but < the total simulation time.*

Information Only

If a BINARY file is to be output, then:

Line 57. Descriptor: Printing frequency to a BINARY file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 58. IPRFRQB, TIMEBINMAX

IPRFRQB: Frequency of printing to a BINARY file. *Accepts an integer > 0; n means every nth timestep will be printed.*

TIMEBINMAX: Maximum time allowed between printing interval. *Accept real number >0 , but < the total simulation time.*

If a CDB file is to be output, then:

Line 59. Descriptor: Printing frequency to a CDB file. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 60. IPRFRQC, TIMECDBMAX

IPRFRQC: Frequency of printing to a CDB file. *Accepts an integer > 0; n means every nth timestep will be printed.*

TIMECDBMAX: Maximum duration allowed between printings. *Accepts real numbers >0 , but < the total simulation time.*

5.1.6 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 compliance 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 sources in many of the test cases for code evaluation (see the QA documentation package NUTS (Kim, 2022a-c)).

Line 61. Descriptor: Is external source interfacing present (T/F). *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 62. STOCKMAN

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

Information Only

5.1.7 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} seconds or less is used whenever zero time step is encountered.

Line 63. Descriptor: Input timestep lower limits. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 64. TIMELIMIT

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

5.1.8 Intrusion time interpolation and extrapolation

Line 65. Descriptor: Actual time of intrusion, Interpolated/Extrapolated time, 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 66. TIME_INTRUSION, RESET_TIME_INTRUSION, TIMETOL

TIME_INTRUSION: The actual intrusion time specified in BRAGFLO run (350 and 1000 yrs for the compliance calculations).

RESET_TIME_INTRUSION: Interpolated/Extrapolated intrusion time.

TIMETOL[4]: Accepted tolerance to initialize at closest time to the RESET_TIME_INTRUSION.

5.1.9 Material map input

NUTS's 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. If the user specified NUTS as a source for the material map, NUTS will proceed reading the input file in the following sequence:

[4] NUTS will read the time in the CDB file and add the tolerance. If the total of time + TIMETOL ≥ RESET_TIME_INTRUSION, this time will be picked up for initialization.

Information Only

Line 67. Descriptor: Number of times for specifying material map. *Accepts a character string up to*[5] *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.**

Line 68. NMATTIMES

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

DO I = 1, NMATTIMES

Line 69. Descriptor(I): Start time for material map. *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 70. TIMEMAST(I)

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

Line 71. Descriptor(I): Material-type grid map *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 72. (((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. *Accepts an integer > 0.*

END DO

Line 73. Descriptor: Material name. *Accepts a character string up to 17 characters in length as a descriptive line.*

DO I = 1, NMAT

Line 74. 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 75. Descriptor: Number of waste regions. *Accepts a character string up to 17 characters in length as a descriptive line.*

[5] NUTS will read the time in the CDB file and add the tolerance. If the total of time + TIMETOL ≥ RESET_TIME_INTRUSION, this time will be picked up for initialization.

Information Only

Line 76. 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 77. Descriptor: Names of waste regions. *Accepts a character string up to 17 characters in length as a descriptive line.*

Line 78. (MAT_WASTE(I), I = 1,NWST)

Line 79. (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 80. RTEMP_BRAG, RPRES_BRAG

RTEMP_BRAG: Reference temperature. *Accepts 273.15<any real number < 473.15*

RPRES_BRAG: Reference pressure. *Accepts 0 < any real, positive number < 1.E10*

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

Line 81. 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 82, IA1, A2, BRESD_MAT(I)

IA1, A2: BRAGFLO parameters irrelevant to NUTS applications.

BRESD_MAT: Brine residual saturation of the material.

END DO

Information Only

5.1.10 Properties input

5.1.10.1 Physical properties

Line 83. 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 84. Descriptor: Element solubility input. *Accepts a character string up to 100 characters in length as a descriptive line.*

If the solubility is not temperature dependent and not table look-up (ELTEMP_SOLB(I) = .FALSE. and SOLB_TABLE(I) = .FALSE.) then:

DO I = 1, NOELEMENT

Line 85. NAME(I), ALOG_ELEMNT_SOLB_LIMIT(I)

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

ALOG_ELEMNT_SOLB_LIMIT: Log_{10} of the element solubility limit (mol/liter). *Accepts any real number for which Log_{10} inverse exists.*

END DO

If the solubility is table look-up (SOLB_TABLE(I) = .TRUE.) then:

DO I = 1, NOELEMENT

Line 86. NAME(I),EQSPACED(I),ID_INT

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

EQSPACED: Logical flag for equally spaced 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 then:

Line 87. NROW1D

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

DO K = 1, NROW1D

Line 88. ROW1D, FX

ROW1D: Row entry in the table. *Accepts any real number.*

FX: Table entries. *Accepts any real number.*

END DO

If 2D table then:

Information Only

Line 89. 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 90. (COLUMN(J,I), J = 1,NCOLUMN(I)

COLUMN: Column entry in two-dimensional table. *Accepts any real number.*

DO K = 1, NROW(I)

Line 91. ROW(K,I),(FXY(K,NNC,I), NNC = 1, NCOLUMN(I))

ROW: Row entry in two-dimensional table. *Accepts any real number.*

FXY: Table entries. *Accepts any real number.*

END DO

END DO

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

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

DO I = 1, NOELEMENT

Line 92. 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 93. Descriptor: Name, molecular (or atomic) weight, inventory, half-life. *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NUCLIDE

Line 94. 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 and < 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.*

Note: 0 is used for HALF-LIFE as a flag to identify stable components.

END DO

Information Only

If phase type is gas, then:

Line 95. 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 96. NAME(I), EQCI(I), EQCS(I)

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

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

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

END DO

Line 96. 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:

Line 97. (RPH(I), I = 1, NMAT)

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

Line 98. Descriptor: Standard conditions brine density if a test run. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 99. RBR_DEN

RBR_DEN: Standard condition brine density. *Accepts 0 < real number < 2000.*

5.1.10.2 Component molecular diffusion input

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

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

DO I = 1, NUCLIDE

Line 101. NAME(I), DMOL(I)

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

Information Only

DMOL: Molecular diffusion of the component (m^2/s). *Accepts a real number > 0 and < a number that gives a molecular diffusional velocity[6] of 0.6 m/s*

END DO

Line 102. Descriptor: Reference viscosity and temperature. *Accepts a character string up to 100 characters in length as a descriptive line.*

DO I = 1, NUCLIDE

If DMOLTEMDEP(I) is .TRUE. then:

Line 103. VISREF(I), TREF(I)

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

5.1.10.3 Rock density input

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

Line 104. Descriptor: Rock density. *Accepts a character string up to 100 characters in length as a descriptive line.*

If soil base concentration is required to be output (option IPRNTMA(10) or IPRNTMB(10) or IPRNTMC(10)) is 1 or sorption is to be calculated (ADSTYPEM is "Y" or "y"),

Line 105. (RROCK_DENS(I), I = 1, NMAT)

RROCK_DENS: Rock grain density for the range specified by the material map index (kg/m^3). *Accepts a real number > 0 and < 5000.*

5.1.10.4 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 106. Descriptor: Waste matrix input. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 107. NWSTCOMP_TOTAL

[6] 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).

Information Only

NWSTCOMP_TOTAL: Number of Component placed in the waste. *Accepts an integer > 0 and < the total number of components.*

DO I = 1, NWSTCOMP_TOTAL

Line 108. 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 and < the total number of components.*

NWASTREGION: Waste site number. *Accepts an integer > 0 and < NSITES.*

END DO

Line 109. 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 110. 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 and < the total number of grid blocks.*

Line 111. (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 integer > 0 and IS ≤ NX, JS ≤ NY, and KS ≤ NZ.*

5.1.10.5 Single porosity matrix input

Matrix Sorption Input

This module inputs the constituent sorption properties of the matrix. Three equilibrium isotherms are considered, depending on the value of MAT_ISOTHERM.

Line 112. Descriptor: Matrix sorption. *Accepts a character string up to 100 characters in length as a descriptive line.*

If ADSTYPEM is 'Y' or 'y', and MAT_ISOTHERM IS 'L' or 'l', then:

DO I = 1, NUCLIDE

Line 113. NAME(I)

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

Line 114. (RXLM(J), J = 1, NMAT)

Information Only

RXLM: Linear sorption coefficient (k_d) of the matrix for the range specified by the material map index (m^3/kg). *Accepts a number > 0*

END DO

If ADSTYPEM is 'Y' or 'y', and MAT_ISOTHERM is 'F' or 'f' then:

DO I = 1, NUCLIDE

Line 115. NAME(I)

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

Line 116. (RXFDCM(J), J = 1, NMAT)

Line 117. (RXFCM(J), J = 1, NMAT)

RXFDCM: Freundlich distribution coefficient of the matrix for the range specified by the material map index (m^3/kg). *Upper and lower bounds are not specified.*

RXFCM: Freundlich coefficient of the matrix for the range specified by the material map index (dimensionless). *Upper and lower bounds are not specified.*

END DO

If ADSTYPEM is 'Y' or 'y', and MAT_ISOTHERM is 'LA' or 'la' or 'La', then:

DO I = 1, NUCLIDE

Line 118. NAME(I)

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

Line 119. (RXLDCM(J), J = 1, NMAT)

Line 120. (RXLCM(J), J = 1, NMAT)

RXLDCM: Langmuir distribution coefficient of the Matrix for the range specified by the material map index (m^3/kg). *Upper and lower bounds are not specified.*

RXLCM: Langmuir coefficient of the matrix for the range 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 121. Descriptor: Matrix dispersion. *Accepts a character string up to 100 characters in length as a descriptive line.*

If MDISPREQ is 'Y' or 'y', then,

Information Only

Line 122. Descriptor: Longitudinal dispersivity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 123. (RALPHALM(I), I = 1, NMAT)

RALPHALM: Longitudinal dispersivity of the matrix for the range specified by the material map index (m). *Accepts 0 < any real number < the smallest hydrological dimension of the spatial domain.*

Line 124. Descriptor: Transverse dispersivity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 125. (RALPHATM(I), I = 1, NMAT)

RALPHATM: Transverse dispersivity of the matrix for the range specified by the material map index (m). *Accepts a real number > 0 and < longitudinal dispersivity.*

Line 126. Descriptor: Tortuosity in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 127. (RTORM(I), I = 1, NMAT)

RTORM: Matrix tortuosity for the range specified by the material map index (dimensionless). *Accepts a real number > 1.*

Matrix Source Input

Line 128. 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 'Y' or 'y', then

Line 129. NTMNINJCOMP

NTMNINJCOMP: Total number of injected components. *Accepts 0 < NTMNINJCOMP < total number of components.*

DO J = 1, NTMNINJCOMP

Line 130. 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 0 < LOCINPM < total number of components.*

NMTBLOKINJ: Total number of grid blocks at which this component is injected. *Accepts an integer 0 <* NMTBLOKINJ *< total number of grid blocks.*

Line 131. (INJBIM(I,J), INJBJM(I,J), INJBKM(I,J), I = 1, NMTBLOKINJ(J))

INJBIM: I index of the injection block. *Accepts an integer 0 <* INJBIM *< total number of grid blocks in x direction.*

Information Only

INJBJM: J index of the injection block. *Accepts an integer 0 < INJBJM < total number of grid blocks in y direction.*

INJBKM: K index of the injection block. *Accepts an integer 0 < INJBKM < total number of grid blocks in z direction.*

END DO

DO I = 1, NTMNINJCOMP

Line 132. NAME(I)

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

Line 133. (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 134. Descriptor: Injection production rates. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 135. NMBLOCKSOURCE, TIMEMSTRT, TIMEMEND

NMBLOCKSOURCE: Number of grid blocks having the source. *Accepts an integer 0 <* NMBLOCKSOURCE *< total number of grid blocks.*

TIMEFSTRT: Starting time of injection/production in the matrix (s). *Accepts a real number 0 < TIMEFSTRT < TIMEFEND.*

TIMEFEND: Ending time of injection/production in the matrix (s). *Accepts a real number 0 < TIMEFEND.*

DO I = 1, NMBLOCKSOURCE

Line 136. IQIM(I), IQJM(I), IQKM(I), RATE(I)

IQIM, IQJM, IQKM: I,J,K indices of the injection/production block. *Accepts an integer 0<*IQIM, IQJM, IQKM *< total number of grid blocks in x ,y, z direction, respectively.*

RATE: Injection/production rate (m^3/s). *Accepts a real number > 0.*

END DO

Line 137: Descriptor: Dirichlet boundary condition in the matrix. *Accepts a character string up to 100 characters in length as a descriptive line.*

Information Only

If MDIRICHLET, then:

Line 138. NTMDIRCOMP,TYPEDIR

NTMDIRCOMP: Number of components with Dirichlet boundary condition (DBC). *Accepts 0 < NTMDIRCOMP < 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 139. 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 0 < LOCMDIRINP < total number of components.*

GCONDIR: Dirichlet boundary condition on concentration (kg/m^3). *Accepts a real number > 0.*

If TYPEDIR is "NOT-GENERAL" or "not-general", then:

Line 140. 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 0 < LOCMDIRINP < total number of components.*

NMTDIRBLOK: Total number of grid blocks with D.B.C. *Accepts an integer 0 < NMTDIRBLOK < total number of grid blocks.*

Line 141. (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 0 < IDRBIM, IDRBJM, IDRBKM < total number of grid blocks in x ,y, z direction, respectively.*

END DO

DO I = 1, NTMDIRCOMP

Line 142. NAME(I)

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

Line 143. (DIRCONCMIJK(IDRBIM(J,I), IDRBJM(J,I), IDRBKM(J,I)),J = 1, NMTDIRBLOK(I))

Information Only

DIRCONCMIJK: Concentration of the component having DBC (kg/m^3). *Accepts a real number > 0.*

END DO

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 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 144. Descriptor: Testing Dynamic source input.

If STOCKMAN = .TRUE., then

Line 145. NTMDEPSRC

NTMDEPSRC: Total number of the radioactive isotopes that have time dependent source. *Accepts a real number 0 < NTMDEPSRC < total number of components.*

If NTMDEPSRC is not zero, then

DO J = 1, NTMDEPSRC

Line 146. 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 0 < LOCMDIRINP < total number of components.*

NMTBLOKSRC: Total number of grid blocks with dynamic source. *Accepts an integer 0 < NMTBLOKSRC < total number of grid blocks.*

Line 147. (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 0 <ISRCI, ISRCJ, ISRCK < total number of grid blocks in x, ,y, z direction, respectively.*

END DO

DO J = 1, NTMDEPSRC

Line 148. NAME(J)

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

Information Only

Line 149. (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 150.

((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.

Line 151.

((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.

END DO

Matrix Concentration Initialization Input

Line 152. Descriptor: Initial concentration in the matrix if any. *Accepts a character string up to 100 characters in length as a descriptive line.*

Information Only

If MMAN_CONC_INIT = .TRUE., then:

Line 153. NINITCOMP

NINITCOMP: Number of components to be initialized. *Accepts 0 < integer < total number of components.*

DO J = 1, NINITCOMP

Line 154. 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 0 <* LOCINP *< total number of components.*

NBLOKINI: Total number of grid blocks initialized. *Accepts an integer 0 <* NBLOKINI *< total number of grid blocks.*

Line 155. (INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

INII, INIJ, INIK: I,J,K indices of the initialized grid blocks. *Accepts an integer 0 <* INII, INIJ, INIK *< total number of grid blocks in x ,y, z direction, respectively.*

END DO

DO J = 1, NINITCOMP

Line 156. NAME(J)

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

Line 157. CONCIJKINI(INII(I,J), IINIJ(I,J), INIK(I,J), I = 1, NBLOKINI(J))

CONCIJKINI: Initialized concentration, kg/s

Matrix Colloid Transport Scaling Factors Input

Line 158. 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 159. Descriptor: Scaling factor for velocities in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 160. (SCALEX(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEX: Scaling factors of x-direction velocities. *Accepts an integer ≥ 1.*

Information Only

Line 161. Descriptor: Scaling factor for velocities in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 162. (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 163. Descriptor: Scaling factor for velocities in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 164. (SCALEZ(I,J,K),I=1,NX),J=1,NY),K=1,NZ)

SCALEZ: Scaling factors of z-direction velocities. *Accepts an integer ≥ 1.*

5.2 ASCII flux field input file (homogeneous)

An example of NUTS's ASCII flux-field input file for homogeneous properties is given in Section 11.2.1.

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.*

Line 3. Descriptor: Test run title. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 4. TEST_TETLE

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.*

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 > 0.*

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.*

Information Only

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.*

Line 11. Descriptor: Matrix porosity and saturation. *Accept character string up to 100 chars as a descriptive line*

Line 12. 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 13. Descriptor: Matrix constant velocity field. *Accept character string up to 100 chars as a descriptive line*

Line 14. TVELXM, TVELYM, TVELZM

TVELXM: Matrix velocity of the phase in x-direction (m/s). *Accepts a real number TVELXM < 0.6*

TVELYM: Matrix velocity of the phase in y-direction (m/s). *Accepts a real number TVELYM < 0.6*

TVELZM: Matrix velocity of the phase in z-direction (m/s). *Accepts a real number TVELZM < 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_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.*

5.3 ASCII flux field input file (heterogeneous)

An example of NUTS's ASCII flux-field input file for heterogeneous properties is given in Section 11.2.2.

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.*

Information Only

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.*

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 > 0.*

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.*

Line 15. Descriptor: Matrix porosities. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 16. ((PHIMINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

PHIMINT3D: Porosity of the matrix (fraction). *Accepts a real number 0 < PHIMINT < 1*

Information Only

Line 17. Descriptor: Matrix saturations. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 18. ((SWMINT3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

SWMINT3D: Saturation of the matrix (fraction). *Accepts a real number 0 < SWMINT < 1.*

Line 19. Descriptor: Matrix velocity field in x-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 20. ((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 TVELXM3D < 0.6.*

Line 21. Descriptor: Matrix velocity field in y-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 22. ((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 TVELYM3D < 0.6.*

Line 23. Descriptor: Matrix velocity field in z-direction. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 24. ((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 TVELZM3D < 0.6*

Line 25. Descriptor: Matrix Block temperature. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 26. ((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 27. Descriptor: Matrix Block brine viscosity. *Accepts a character string up to 100 characters in length as a descriptive line.*

Line 28. ((XMVIS3D(I,J,K),I=1,NX),J=1,NY),Z=1,NZ)

XMVIS3D: Matrix brine viscosity (Pa/s). *Accepts a real number > 0.*

5.4 NUTS output files

NUTS produces an output ASCII file or output CDB file. When test run command line argument option is "Y" or "y", the output ASCII file is produced. When test run command line argument option is "N" or "n" for compliance calculation, the output CDB file is produced by NUTS. These output files include the same information contained in the input files along with NUTS's calculation results. NUTS creates "History Variables" that are specified by IPRNTMA variables in NUTS input ASCII file (Section 5.1.2).

Information Only

In the output ASCII file, component's history variable data are printed with component name and variable name, that is, that is listed in Table 5-1. For compliance calculations (in the output CDB file), NUTS input ASCII file sets to create data of 4 element variables – CMn for volumetric concentration (kg/m^3 brine) of the dissolved radionuclide, BMDISMn for the dissolved mass (kg) in the brine, BMPRCMn for the precipitated mass (kg), and TOTMMCn for total mass (Ci) of dissolved, precipitated and sorbed (Table 5-2). Because radionuclide components are listed in the order of AM241L, PU239L, PU238L, U234L, and TH230L, n numbers directly correspond to the components by $n = 1$ for AM241L, $n = 2$ for PU239L, $n = 3$ for PU238L, $n = 4$ for U234L, and $n = 5$ for TH230L.

Specific variables like CMn, BMDISMn, BMPRCMn, and TOTMMCn represent the concentrations or masses of all components at each grid block, that IPRNTMA variables in NUTS input ASCII fill determine whether any variable will be printed to the NUTS output CDB file. IPRNTMA variables can have 0 (not print) and 1 (print) value (see Section 5.1.2). When all IPRNTMA = 0, NUTS still prints transport related variable data – History Variables, Global Variables, and Element Variables – at each time step to the output CDB file for compliance calculations. These variables are described in Table 5-3 through Table 5-5.

Information Only

Table 5-1. Variables' names to be printed in the output ASCII file

IPRNTMA No	Variable name	Description
1	'MTMPG OF THE D-N IN THE BRINE'	Matrix total mass of dissolved constituents in the brine in each grid block, kg
2	'MTMPG OF THE P-N'	Matrix total mass of precipitated constituents in each grid block, kg
3	'MTMPG OF THE D-N, P-N, AND A-N'	Matrix total mass of dissolved, precipitated and sorbed constituents in each grid block, kg
4	'MTCPG OF THE D-N IN THE BRINE'	Matrix total dissolved curies of all constituents in each grid block, curies
5	'MTCPG OF THE P-N'	Matrix total precipitated curies of all constituents in each grid block, curies
6	'MTCPG OF THE D-N, P-N, AND A-N'	Matrix total dissolved, precipitated, and sorbed curies of all constituents in each grid block, curies
7	'MVCPG OF THE D-N IN THE BRINE'	Matrix volumetric concentration of the dissolved isotope, kg/(m^3 brine)
8	'MMPG OF THE D-N IN THE BRINE'	Matrix dissolved mass in the brine of a certain constituent, kg
9	'MMPG OF THE P-N '	Matrix precipitated mass of a certain constituent, kg
10	'MMPG OF THE D-N, P-N, AND A-N'	Matrix soil base concentration of the isotope, mg/(kg soil)
11	'MCPG OF NUCLIDE VC'	Matrix curies of volumetric concentration of a certain constituent, curies/(m^3 brine)
12	'MCPG OF THE D-N'	Matrix curies of dissolved mass in the brine of a certain constituent, curies
13	'MCPG OF THE P-N'	Matrix curies of precipitated mass of a certain constituent, curies
14	'MCPG OF THE D-N, P-N, AND A-N'	Matrix curies of dissolved, precipitated and sorbed mass of a certain constituent, curies

Table 5-2. Variables' names to be printed in the output CDB file when a NUTS input ASCII file sets any IPRNTMA variable to 1 for the compliance calculations, where $n = 1$ for AM241L, $n = 2$ for PU239L, $n = 3$ for PU238L, $n = 4$ for U234L, and $n = 5$ for TH230L

IPRNTMA No	Variable name	Description
7	CMn	Matrix volumetric concentration of the dissolved isotope, kg/(m^3 brine)
8	BMDISMn	Matrix dissolved mass in the brine of a certain constituent, kg
9	BMPRCMn	Matrix precipitated mass of a certain constituent, kg
14	TOTMMCn	Matrix curies of dissolved, precipitated and sorbed mass of a certain constituent, curies

Information Only

Table 5-3. History variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0

Variables name	Description
TIMEDAY	Elapsed time (day)
TIMEYR	Elapsed time (yr)
DELT	Time step (s)
DELTDAY	Time step (day)
DELTYR	Time step (yr)
ITERAVG	Average Newton-Raphson iterations per time step (iterations/step)
CPUTIME	Time step CPU time (s)
CPUTOT	Total CPU time (hr)
STEPNUM	Time step number (dimensionless)
ITERTOT	Total Newton-Raphson iterations (iterations)

Table 5-4. Global variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0

Variables name	Description
BRNMBCUM	Cumulative brine mass balance (dimensionless)
GASMBCUM	Cumulative gas mass balance (dimensionless)
BRNMBMAX	Maximum brine mass balance (dimensionless)
GASMBMAX	Maximum gas mass balance (dimensionless)
IBRNMBMX	I-index, location of maximum brine mass balance (dimensionless)
JBRNMBMX	J-index, location of maximum brine mass balance (dimensionless)
KBRNMBMX	K-index, location of maximum brine mass balance (dimensionless)
IGASMBMX	I-index, location of maximum gas mass balance (dimensionless)
JGASMBMX	J-index, location of maximum gas mass balance (dimensionless)
KGASMBMX	K-index, location of maximum gas mass balance (dimensionless)

Information Only

Table 5-5. Element variables' names to be printed in the output CDB file for the compliance calculation when all IPRTMA = 0, where $n = 1$ for AM241L, $n = 2$ for PU239L, $n = 3$ for PU238L, $n = 4$ for U234L, and $n = 5$ for TH230L

Variables name	Description
PRESBRIN	Brine pressure (Pa)
PRESGAS	Gas pressure (Pa)
POROS	Porosity (dimensionless)
DENGAS	Gas density (kg/m^3)
PERMBRX	Permeability to brine, x-direction (m^2)
PERMGASX	Permeability to gas, x-direction (m^2)
SATGAS	Gas saturation (fraction of void volume)
FLOWBRX	Time-avg interblock brine flow, x-direction (m^3/s at reference conditions)
FLOWBRY	Time-avg interblock brine flow, y-direction (m^3/s at reference conditions)
FLOWGASX	Time-avg interblock gas flow, x-direction (m^3/s at reference conditions)
FLOWGASY	Time-avg interblock gas flow, y-direction (m^3/s at reference conditions)
H2RATE	H_2 generation rate - simple model (kg/(s m^3))
BRINRATE	Brine consumption rate - simple model (kg/(s m^3)
FECONC	Fe concentration - simple model (kg/m^3)
CELLCONC	C6-H10-O5 concentration - simple model (kg/m^3)
MGOC	MgO concentration - simple model (kg/m^3)
H2_RAD_R	H_2 generation rate - radiolysis (kg/(s m^3))
SM1*iso*	Amount of isotope *iso* from Waste Region 1 (mol). *iso*=AM241, PU238, PU239, PU240, PU242.
SM2*iso*	Amount of isotope *iso* from Waste Region 2 (mol). *iso*=AM241, PU238, PU239, PU240, PU242.
CNCNJMn	Concentration of the injected component n (kg/m^3 brine (or fluid))
FLUXIMn	Mass flux of component n at the interface I (the convention is to put the flux at the left interface) of each grid block (Ci/s)
FLUXJMn	Mass flux of component n at the interface I (the convention is to put the flux at the lower interface) of each grid block (Ci/s)
CNDMMSn	Mass of the condensed component n (has a non-zero value only in gas transport)

Information Only

6.0 REQUIRED USER TRAINING AND/OR BACKGROUND

To exercise the code NUTS, users should have a working knowledge of Linux operating systems, including how to execute commands via a system shell (such as BASH), and of Sandia's CAMDAT database (CDB) file format. The input/output CDB files can be manipulated with PA codes ALGEBRACDB, GROPECDB, and SUMMARIZE, thus having a working knowledge of these codes is helpful when preparing inputs and analyzing outputs from NUTS.

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 PANEL, and those that exercise immediately after NUTS, principally CCDFGF.

Information Only

7.0 CAPABILITIES AND LIMITATIONS

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 radionuclides transported by NUTS are assumed to be thermodynamically stable within their solubility limits. No gas phases or radionuclide-rich liquid phases are allowed. However, precipitation is possible and is calculated when and if solute concentrations exceed their solubility limits.
4. Dissolved radionuclides 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:

   [Isotopic Solubility] = [Mole Fraction of that particular isotope] × [the elemental solubility]
8. Gas-liquid equilibrium is assumed to obey the linear relationship represented by Eq. 4-93, with temperature dependence given by Eq. 4-94.
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 8.0.

Information Only

8.0 USER INTERACTIONS WITH THE SOFTWARE

NUTS Version 2.08 runs in the CAMCON configuration and executes on a Linux platform. To execute NUTS, the program executable name (nuts) must be in the user's current PATH or referenced by its path name.

NUTS expects a number of command line arguments. The program arguments may either be positional (provided in order) or flagged (preceded by the appropriate flag). Flags always start with "–"; they may be abbreviated. All positional arguments must be listed before any flagged arguments. The only difference between using positional and flagged arguments is that optional arguments may be omitted for flagged arguments.

NUTS expects the following command line arguments:

1	-input	The file containing the flux data. This file must be a CAMDAT file for compliance calculations. If the -test argument is "Y", this file is the NUTS ASCII flux-field input file, described in Sections 5.2 and 5.3.
2	–test	This text option determines if this is a test run. This option must be "N" for compliance calculations. It should be "Y" only for certain verification tests that use test input files. Case is unimportant and only the first character is checked.
3	–ascii	This text option must be "BIN" for compliance calculations. If the -test argument is "Y", this option must be "ASC". Case is unimportant, but at least three characters must be specified.
4	–user	The input ASCII radionuclide input file, described in Section 5.1.
5	–output	The output NUTS file. This must be a CAMDAT file for compliance calculations. A secondary text file is also output. The text file has the same name as the CAMDAT file, but with a ".dat" extension. If the -test argument is "Y", the NUTS output file is an ASCII file; no secondary text file is output.
6	–debug	The output diagnostics file.
7	–liquid	This optional text option determines the type of phase in the transporting medium. This option must be "L" (liquid) for compliance calculations. Option "G" (gas) for modeling transport in the gas phase is not tested. Case is unimportant and only the first character is checked.
8	–bragflo	The BRAGFLO input ASCII file with the material map. This file is required for compliance calculations. If the -test argument is "Y", enter "CANCEL" to indicate that no file is specified, and the material map will be read from the NUTS input file specified in the -user argument.

Information Only

9	–nuclide	This text option should be "CDB" to read the nuclide related material properties from the property CAMDAT file specified in the -inventory argument, or "N" to read them from the NUTS input file specified in the -user argument.
10	–inventory	The optional input properties CAMDAT file that contains the chemical constituent properties and rock properties. This file should only be specified if the -nuclide argument is "CDB". In the WIPP PA, the PANEL (CONcentration mode) output CDB file is used (Table 8-2 and Table 8-3). Enter "CANCEL" if no file is specified.
11	–undist	The optional input CAMDAT file for the undisturbed scenario used to initialize well bore intrusion calculations. This file is only required for intrusion times other than the one used in BRAGFLO. Enter "CANCEL" if no file is specified.

Table 8-1 through Table 8-3 list example of executing NUTS for different run modes. Each table shows two command line arguments but are executing the same runs. The first command line uses the positional arguments. The second command line uses the flagged arguments.

Table 8-1 lists an example to execute NUTS screening (SCN) mode for the CRA19 (Compliance Recertification Application 2019) analysis; Table 8-2 lists an example to execute NUTS isotope (ISO) mode; and Table 8-3 lists an example to execute NUTS intrusion (INT) mode.

Information Only

Table 8-1. Example to execute NUTS in SCN (screening) mode

```
nuts ./Inputs/bf3_CRA19_r1_s1_v001.cdb N BIN ./Inputs/nut_scn_CRA19_s1.inp
./Output/nut_scn_CRA19_r1_s1_v001.cdb ./Output/nut_scn_CRA19_r1_s1_v001.xdbg L
./Inputs/bf2_CRA19_r1_s1_v001.inp N CANCEL CANCEL

nuts -input ./Inputs/bf3_CRA19_r1_s1_v001.cdb -test N -ascii BIN -user ./Inputs/nut_scn_CRA19_s1.inp
-output ./Output/nut_scn_CRA19_r1_s1_v001.cdb -debug ./Output/nut_scn_CRA19_r1_s1_v001.xdbg
-liquid L -bragflo ./Inputs/bf2_CRA19_r1_s1_v001.inp -nuclide N -inventory CANCEL -undist CANCEL
```

Table 8-2. Example to execute NUTS in ISO (isotope) mode

```
nuts ./MATSET/Output/ms_nut_CRA19_r1_s1_v001.cdb N BIN ./Inputs/nut_iso_CRA19_s1.inp
./Output/nut_iso_CRA19_r1_s1_v001.cdb ./Output/nut_iso_CRA19_r1_s1_v001.xdbg L
./Inputs/bf2_CRA19_r1_s1_v001.inp CDB ./Inputs/panel_con_CRA19_b1_r1_s1_v001.cdb CANCEL

nuts -input ./MATSET/Output/ms_nut_CRA19_r1_s1_v001.cdb -test N -ascii BIN -user
./Inputs/nut_iso_CRA19_s1.inp -output ./Output/nut_iso_CRA19_r1_s1_v001.cdb -debug
./Output/nut_iso_CRA19_r1_s1_v001.xdbg -liquid L -bragflo ./Inputs/bf2_CRA19_r1_s1_v001.inp -nuclide
CDB -inventory ./Inputs/panel_con_CRA19_b1_r1_s1_v001.cdb -undist CANCEL
```

Table 8-3. Example to execute NUTS in INT (intrusion) mode

```
nuts ./MATSET/Output/ms_nut_CRA19_r1_s2_v001.cdb N BIN ./Inputs/nut_int_CRA19_s2_t00100.inp
./Output/nut_int_CRA19_r1_s2_t00100_v001.cdb ./Output/nut_int_CRA19_r1_s2_t00100_v001.xdbg L
./Inputs/bf2_CRA19_r1_s2_v001.inp CDB ./Inputs/panel_con_CRA19_b1_r1_s2_v001.cdb
./Output/nut_iso_CRA19_r1_s1_v001.cdb

nuts -input ./MATSET/Output/ms_nut_CRA19_r1_s2_v001.cdb -test N -ascii BIN -user
./Inputs/nut_int_CRA19_s2_t00100.inp -output ./Output/nut_int_CRA19_r1_s2_t00100_v001.cdb -debug
./Output/nut_int_CRA19_r1_s2_t00100_v001.xdbg -liquid L -bragflo ./Inputs/bf2_CRA19_r1_s2_v001.inp
-nuclide CDB -inventory ./Inputs/panel_con_CRA19_b1_r1_s2_v001.cdb -undist
./Output/nut_iso_CRA19_r1_s1_v001.cdb
```

Information Only

9.0 ERROR MESSAGES

A complete listing of NUTS's error and warning messages is given in this section. 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 date. Error Messages are categorized according to the files in which they arise.

Here "*xxx,*" "*yyy*" and "*zzz*" represent a integer value, "*xxxExx*" and "*yyyEyy*" represent a real value, , and "*i,j,k* " represents a grid block number.

Aborting Messages:

1. 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.
2. Number of colloids used as *xxx* is > the total number of the isotopes *yyy*. Please set NCOLLOID <= *yyy*. NUTS will be terminated.
3. Number of colloids used as *xxx* cannot be 0 or negative. Please use a number <= *yyy*. NUTS will be terminated.
4. Location of the colloid used as *xxx* is > the total number of the isotopes *yyy*. Please set the location <= *yyy*. NUTS will be terminated.
5. Location of the colloid used as *xxx* cannot be negative. Please use a positive number <= *yyy*. NUTS will be terminated.
6. (BRAGFLO INPUT) NO DATA AVAILABLE TO READ AFTER *xxx* TIME STEPS OF BRAGFLO. THE PROGRAM IS TERMINATED.
7. (BRAGFLO INPUT) Number of grid blocks in x cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.
8. (BRAGFLO INPUT) Number of grid blocks in y cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.
9. (BRAGFLO INPUT) Number of grid blocks in z cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.
10. (BRAGFLO INPUT) Grid block length in x cannot be *xxxExx* for block number *i,j,k*. Use positive number > 0. NUTS will be aborted.
11. (BRAGFLO INPUT) Grid block length in y cannot be *xxxExx* for block number *i,j,k*. Use positive number > 0. NUTS will be aborted
12. (BRAGFLO INPUT) Grid block length in z cannot be *xxxExx* for block number *i,j,k*. Use positive number > 0. NUTS will be aborted.
13. (BRAGFLO INPUT) Viscosity of the brine cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.

Information Only

14. (BRAGFLO INPUT) Grid block porosity cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
15. (BRAGFLO INPUT) Grid block saturation can not be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
16. (BRAGFLO INPUT) Darcian Velocity in x cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 are appropriate. NUTS will be aborted.
17. (BRAGFLO INPUT) Darcian Velocity in y can not be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 are appropriate. NUTS will be aborted.
18. (BRAGFLO INPUT) Darcian Velocity in z cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 are appropriate. NUTS will be aborted.
19. (BRAGFLO INPUT) Grid block brine pH cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
20. (BRAGFLO INPUT) Flux in x cannot be *xxxExx* for block number *i,j,k*. Numbers based on v <= 0.6 m/s are appropriate. NUTS will be aborted.
21. (BRAGFLO INPUT) Flux in y cannot be *xxxExx* for block number *i,j,k*. Numbers based on v <= 0.6 m/s are appropriate. NUTS will be aborted.
22. (BRAGFLO INPUT) Flux in z cannot be *xxxExx* for block number *i,j,k*. Numbers based on v <= 0.6 m/s are appropriate. NUTS will be aborted.
23. (BRAGFLO INPUT) Density of brine cannot be *xxxExx* for block number *i,j,k*. Use a positive number. NUTS will be aborted.
24. (BRAGFLO INPUT) Grid block brine viscosity as *xxxExx* is high. This value is not for a liquid. NUTS will be aborted.
25. (BRAGFLO INPUT) Grid block saturation cannot be *xxxExx* for block number *i,j,k*. Use numbers <= 1. NUTS will be aborted.
26. (BRAGFLO INPUT) Grid block porosity cannot be *xxxExx* for block number *i,j,k*. Use numbers <= 1. NUTS will be aborted.
27. (BRAGFLO INPUT) MATERIAL MAP OF TIME ZERO IS NOT IN BRAGFLO INPUT. EXECUTOPM TERMINATED IN SURBOUTINE MAT_MAP.
28. (BRAGFLO INPUT) Unable to read parameters from BRAGFLO input. Execution stopped.
29. (BRAGFLO INPUT)(In the debug file) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE SEE THE ABOVE WRITTEN MESSAGES
30. (BRAGFLO INPUT)(On the screen) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS debug file) TO IDENTIFY THE PROBLEM.
31. (TEST FILE INPUT) Number of grid blocks in x cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.

Information Only

32. (TEST FILE INPUT) Number of grid blocks in y cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.
33. (TEST FILE INPUT) Number of grid blocks in z cannot be *xxx*. Use at least "1" grid block. NUTS will be aborted.
34. (TEST FILE INPUT) Grid block length in x cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
35. (TEST FILE INPUT) Grid block length in y cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
36. (TEST FILE INPUT) Grid block length in z cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
37. (TEST FILE INPUT) Viscosity of the brine cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
38. (TEST FILE INPUT) Grid block porosity cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
39. (TEST FILE INPUT) Grid block saturation cannot be *xxxExx* for block number *i,j,k*. Use positive number. NUTS will be aborted.
40. (TEST FILE INPUT) Darcian Velocity in x cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
41. (TEST FILE INPUT) Darcian Velocity in y cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
42. (TEST FILE INPUT) Darcian Velocity in z cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
43. (TEST FILE INPUT) Grid block brine viscosity as *xxxExx* is high. This value is not for a liquid. NUTS will be aborted.
44. (TEST FILE INPUT) Grid block saturation cannot be *xxxExx* for block number *i,j,k*. Use numbers <= 1. NUTS will be aborted.
45. (TEST FILE INPUT) Grid block porosity cannot be *xxxExx* for block number *i,j,k*. Use numbers <= 1. NUTS will be aborted.
46. (TEST FILE INPUT) Time step size cannot be *xxxExx*. Use positive number. NUTS will be aborted.
47. (TEST FILE INPUT) Number of time steps cannot be *xxx*. Use positive number. NUTS will be aborted.
48. (TEST FILE INPUT)(In the debug file) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE SEE THE ABOVE WRITTEN MESSAGES
49. (TEST FILE INPUT)(On the screen) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS debug file) TO IDENTIFY THE PROBLEM.

Information Only

50. (NUTS INPUT) ZERO TIME MATERIAL MAP IS NOT IN NUTS INPUT. EXECUTION TERMINATED IN SUBROUTINE MAT_MAP.
51. (NUTS INPUT) Number of disposal sites cannot be *xxx*. Use at least "1" site. NUTS will be aborted.
52. (NUTS INPUT) Number of sites *xxx* cannot be > number of grid blocks *yyy*. They can be equal. NUTS will be aborted.
53. (NUTS INPUT) Number of materials cannot be *xxx*. Use at least "1" material. NUTS will be aborted.
54. (NUTS INPUT) Number of materials *xxx* cannot be > # of grid blocks *yyy*. They can be equal. NUTS will be aborted.
55. (NUTS INPUT) Number of components in site # *xxx* in the input cannot be *yyy*. Use a number >= 1. NUTS will be aborted.
56. (NUTS INPUT) Number of elements cannot be *xxx*. Use at least "1" element. NUTS will be aborted.
57. (NUTS INPUT) Number of elements *xxx* cannot be > number of nuclides *yyy*. They can be equal. NUTS will be aborted.
58. (NUTS INPUT) Number of NUCLIDEs cannot be *xxx*. Use at least "1" nuclide. NUTS will be aborted.
59. (NUTS INPUT) Solubility of element number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
60. (NUTS INPUT) Molecular weight of component number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
61. (NUTS INPUT) Molecular weight of component number *yyy* in the input cannot be > 1. as *xxxExx*. The conversion factors are based on "Kg Mol" and not "gm Mol". NUTS will be aborted.
62. (NUTS INPUT) Contact handled inventory of component number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
63. (NUTS INPUT) Remote handled inventory of component number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
64. (NUTS INPUT) Half-Life of component number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
65. (NUTS INPUT) Brine pH cannot be *xxxExx* for material number *yyy*. Use positive number. NUTS will be aborted.
66. (NUTS INPUT) Molecular diffusion of component number *yyy* in the input cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
67. (NUTS INPUT) Diffusion Velocity in x cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.

Information Only

68. (NUTS INPUT) Diffusion Velocity in y cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
69. (NUTS INPUT) Diffusion Velocity in z cannot be *xxxExx* for block number *i,j,k*. Numbers <= 0.6 m/s are appropriate. NUTS will be aborted.
70. (NUTS INPUT) Rock density cannot be *xxxExx* for material number *yyy*. Use a positive number. NUTS will be aborted.
71. (NUTS INPUT) Number of waste components cannot be *xxx*. Use a positive number. NUTS will be aborted.
72. (NUTS INPUT) Number of waste components cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
73. (NUTS INPUT) Location of waste component in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
74. (NUTS INPUT) Location of waste component cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
75. (NUTS INPUT) The # of the waste region in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
76. (NUTS INPUT) The # of the waste region in the input cannot be *xxx*. Use a number < *yyy* which is total # of sites. NUTS will be aborted.
77. (NUTS INPUT) The # of grid block in each site cannot be *xxx*. Use a number < *yyy* which is total # of grid blocks. NUTS will be aborted.
78. (NUTS INPUT) The # of grid block in each site cannot be *xxx*. Use a number >= 1. NUTS will be aborted.
79. (NUTS INPUT) Linear adsorption coefficient of component number *yyy* in the material number *zzz* cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
80. (NUTS INPUT) Rock longitudinal dispersivity cannot be *xxxExx* for material number *yyy*. Use a positive number. NUTS will be aborted.
81. (NUTS INPUT) Rock transverse dispersivity cannot be *xxxExx* for material number *yyy*. Use a positive number. NUTS will be aborted.
82. (NUTS INPUT) Number of injected nuclides cannot be *xxx*. Use a positive number. NUTS will be aborted.
83. (NUTS INPUT) Number of injected nuclides cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
84. (NUTS INPUT) Location of injected nuclide in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
85. (NUTS INPUT) Location of injected nuclide cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.

Information Only

86. (NUTS INPUT) Total # of injection grids in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
87. (NUTS INPUT) Total # of injection grids in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids. NUTS will be aborted.
88. (NUTS INPUT) Injection grid I-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
89. (NUTS INPUT) Injection grid I-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in I-direction. NUTS will be aborted.
90. (NUTS INPUT) Injection grid J-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
91. (NUTS INPUT) Injection grid J-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in J-direction. NUTS will be aborted.
92. (NUTS INPUT) Injection grid K-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
93. (NUTS INPUT) Injection grid K-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in K-direction. NUTS will be aborted.
94. (NUTS INPUT) Number of the grids with a source cannot be *xxx*. Use a positive number. NUTS will be aborted.
95. (NUTS INPUT) Number of the grids with a source cannot be *xxx*. Use a number < *yyy* which is total # of grids. NUTS will be aborted.
96. (NUTS INPUT) Source grid I-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
97. (NUTS INPUT) Source grid I-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in I-direction. NUTS will be aborted.
98. (NUTS INPUT) Source grid J-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
99. (NUTS INPUT) Source grid J-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in J-direction. NUTS will be aborted.
100. (NUTS INPUT) Source grid K-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
101. (NUTS INPUT) Source grid K-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in K-direction. NUTS will be aborted.
102. (NUTS INPUT) Source rate is zero in source block # *xxx*. If you have a source, you should have a rate. NUTS will be aborted.
103. (NUTS INPUT) Source starting time cannot be *xxxExx*. Use a positive number or 0. NUTS will be aborted.

Information Only

104. (NUTS INPUT) Source starting time cannot be *xxxExx* which is > ending time *yyyEyy*. NUTS will be aborted.
105. (NUTS INPUT) Source ending time cannot be *xxxExx*. Use a positive number or 0. NUTS will be aborted.
106. (NUTS INPUT) Number nuclides having Dirichlet B.Cs cannot be *xxx*. Use a positive number. NUTS will be aborted.
107. (NUTS INPUT) Number nuclides having Dirichlet B.Cs cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
108. (NUTS INPUT) Location of nuclides having Dirichlet B.Cs in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
109. (NUTS INPUT) Location of nuclide having Dirichlet B.Cs cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
110. (NUTS INPUT) Total # of grids having Dirichlet B.Cs cannot be *xxx*. Use a positive number. NUTS will be aborted.
111. (NUTS INPUT) Total # of grids having Dirichlet B.Cs cannot be *xxx*. Use a number < *yyy* which is total # of grids. NUTS will be aborted.
112. (NUTS INPUT) Dirichlet grid I-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
113. (NUTS INPUT) Dirichlet grid I-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in I-direction. NUTS will be aborted.
114. (NUTS INPUT) Dirichlet grid J-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
115. (NUTS INPUT) Dirichlet grid J-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in J-direction. NUTS will be aborted.
116. (NUTS INPUT) Dirichlet grid K-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
117. (NUTS INPUT) Dirichlet grid K-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in K-direction. NUTS will be aborted.
118. (NUTS INPUT) Injected concentration of component *yyy* in grid *zzz* cannot be *xxxExx*. Please use a positive number. NUTS will be aborted.
119. (NUTS INPUT) Dirichlet B.C. concentration of component *yyy* in grid *zzz* cannot be *xxxExx*. Please use a positive number. NUTS will be aborted.
120. (NUTS INPUT) Number nuclides initialized cannot be *xxx*. Use a positive number. NUTS will be aborted.
121. (NUTS INPUT) Number of initialized nuclides cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.

Information Only

122. (NUTS INPUT) Location of initialized nuclides in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
123. (NUTS INPUT) Location of initialized nuclide cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
124. (NUTS INPUT) Total # of initialized grids cannot be *xxx*. Use a positive number. NUTS will be aborted.
125. (NUTS INPUT) Total # of initialized grids cannot be *xxx*. Use a number < *yyy* which is total # of grids. NUTS will be aborted.
126. (NUTS INPUT) Initialized grid I-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
127. (NUTS INPUT) Initialized grid I-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in I-direction. NUTS will be aborted.
128. (NUTS INPUT) Initialized grid J-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
129. (NUTS INPUT) Initialized grid J-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in J-direction. NUTS will be aborted.
130. (NUTS INPUT) Initialized grid K-index in the input cannot be *xxx*. Use a positive number. NUTS will be aborted.
131. (NUTS INPUT) Initialized grid K-index in the input cannot be *xxx*. Use a number < *yyy* which is total # of grids in K-direction. NUTS will be aborted.
132. (NUTS INPUT) Initialized concentration of component *yyy* in grid *zzz* cannot be *xxxExx*. Please use a positive number. NUTS will be aborted.
133. (NUTS INPUT) Scaling factor for colloid transport in x-direction for grid block *i,j,k* cannot be *xxxExx*. Use a number >= 1. NUTS will be aborted.
134. (NUTS INPUT) Scaling factor for colloid transport in y-direction for grid block *i,j,k* cannot be *xxxExx*. Use a number >= 1. NUTS will be aborted.
135. (NUTS INPUT) Scaling factor for colloid transport in z-direction for grid block *i,j,k* cannot be *xxxExx*. Use a number >= 1. NUTS will be aborted.
136. (NUTS INPUT) Linear adsorption coefficient of component number *yyy* in the material number *zzz* cannot be *xxxExx*. Use a positive number. NUTS will be aborted.
137. (NUTS INPUT) Linear adsorption coefficient of component number *yyy* in the material number *zzz* cannot be *xxxExx*. Use a positive number <=1. NUTS will be aborted.
138. (NUTS INPUT) Number of time dependent source nuclides cannot be *xxx*. Use a positive number. NUTS will be aborted.
139. (NUTS INPUT) Number of time dependent source nuclides cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.

Information Only

140. (NUTS INPUT) Location of time dependent source nuclides cannot be *xxx*. Use a positive number. NUTS will be aborted.
141. (NUTS INPUT) Location of time dependent source nuclide cannot be *xxx*. Use a number < *yyy* which is total # of the nuclides. NUTS will be aborted.
142. (NUTS INPUT) Total # of time dependent source grids cannot be *xxx*. Use a positive number. NUTS will be aborted.
143. (NUTS INPUT) Total # of time dependent source grids cannot be *xxx*. Use a number < *yyy* which is total # of grids. NUTS will be aborted.
144. (NUTS INPUT) Time dependent source grid I-index cannot be *xxx*. Use a positive number. NUTS will be aborted.
145. (NUTS INPUT) Time dependent source grid I-index cannot be *xxx*. Use a number < *yyy* which is total # of grids in I-direction. NUTS will be aborted.
146. (NUTS INPUT) Time dependent source grid J-index cannot be *xxx*. Use a positive number. NUTS will be aborted.
147. (NUTS INPUT) Time dependent source grid J-index cannot be *xxx*. Use a number < *yyy* which is total # of grids in J-direction. NUTS will be aborted.
148. (NUTS INPUT) Time dependent source grid K-index cannot be *xxx*. Use a positive number. NUTS will be aborted.
149. (NUTS INPUT) Time dependent source grid K-index cannot be *xxx*. Use a number < *yyy* which is total # of grids in K-direction. NUTS will be aborted.
150. (NUTS INPUT)(In the debug file) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE SEE THE ABOVE WRITTEN MESSAGES
151. (NUTS INPUT)(On the screen) SERIOUS ERROR IS DETECTED IN THE INPUT FILE NUTS RUN IS TERMINATED. PLEASE CONSULT WITH (NUTS debug file) TO IDENTIFY THE PROBLEM.

Warning Messages:

1. Not enough memory allocated for band width Parameter declaration of MBW is *xxx* while the number memory required is *yyy*. Please set MBW >= *yyy* in NUT_PARAM.INC and rerun
2. Not enough memory allocated for x dimension Parameter declaration of MX is *xxx* while the number of grid blocks is *yyy*. Please set MX >= *yyy* in NUT_PARAM.INC and rerun.
3. Not enough memory allocated for y dimension Parameter declaration of MY is *xxx* while the number of grid blocks is *yyy*. Please set MY >= *yyy* in NUT_PARAM.INC and rerun.
4. Not enough memory allocated for z dimension Parameter declaration of MZ is *xxx* while the number of grid blocks is *yyy*. Please set MZ >= *yyy* in NUT_PARAM.INC and rerun.

Information Only

5. Not enough memory allocated for total number of grid blocks, Parameter declaration of NB is *xxx* while the total number of grid blocks is *yyy* Please set NB >= *yyy* in NUT_PARAM.INC and rerun.
6. Only 1 for 1st order or 2 for 2nd order, 3 for 2nd order with split operator, and 4 for central difference are allowed. The order of the method requested as *xxx* is not available in NUTS.
7. Only BETA2 of [0,1] interval is accepted. 0 is explicit solution, 1/2 is Crank Nicolson, and 1 is fully implicit. The value requested as *xxx* is not available in NUTS.
8. Not enough memory allocated for the nuclide. Parameter declaration of NC is *xxx* while the number of radionuclides is *yyy*. Please set NC >= *yyy* in NUT_PARAM.INC and rerun.
9. Number of iterations can be *xxx*. Iteration is set to "1", and explicit precipitate is used.
10. Grid block PCO2 # is too high for block number i,j,k.
11. Grid block Brine Pressure # is too high for block number i,j,k.
12. Reference temperature as # is low. Use > 273.15 K.
13. Reference temperature as # is high. This may need thermal simulator.
14. Reference brine pressure as # is too high.
15. Temperature of the grid block is # for block number i,j,k. Use > 273.15 K.
16. Grid block temperature as # is high for grid block i,j,k. This may need thermal simulator.
17. Grid block brine pH as # is high for grid block i,j,k. Use practical value.
18. (BRAGFLO INPUT) Material Map is not found in BRAGFLO input.
19. (BRAGFLO INPUT) Ref. Temp.Press. is not found in BRAGFLO input.
20. (BRAGFLO INPUT) Brine Compres. is not found in BRAGFLO input.
21. (BRAGFLO INPUT) Brine Residual Saturation is not found in BRAGFLO input.
22. (BRAGFLO INPUT) Grid block brine density as *xxxExx* is high for grid block i,j,k. Use practical value
23. (BRAGFLO INPUT)(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).
24. (BRAGFLO INPUT)(In debug file) Undesirable value(s) is detected in the input file (BRAGFLO binary file name) For more information, please check above written messages.
25. (TEST FILE INPUT) Grid block dimension in x as *xxxExx* is high for grid block *i,j,k*. This may lead to high numerical dispersion.
26. (TEST FILE INPUT) Grid block dimension in y as *xxxExx* is high for grid block *i,j,k*. This may lead to high numerical dispersion.
27. (TEST FILE INPUT) Grid block dimension in z as *xxxExx* is high for grid block *i,j,k*. This may lead to high numerical dispersion.

Information Only

28. (TEST FILE INPUT) Grid block temperature as *xxxExx* is high. This may need thermal simulator.
29. (TEST FILE INPUT) Temperature of the grid block is *xxxExx* for block number *i,j,k*. Use > 273.15.
30. (TEST FILE INPUT) You have *xxx* characters in title. Title should not exceed 100 characters.
31. (TEST FILE INPUT)(On the screen) Undesirable value(s) is detected in the input file (TEST FILE INPUT file name) For further information, please check (NUTS debug file name).
32. (TEST FILE INPUT)(In debug file) Undesirable value(s) is detected in the input file (TEST FILE INPUT file name) For more information, please check above written messages.
33. (NUTS INPUT) Brine pH as *xxxExx* for material number *yyy* is high!! Check it please.
34. (NUTS INPUT) Rock density as *xxxExx* for material number *yyy* is high. Please check !!.
35. (NUTS INPUT) Linear adsorption coefficient of component number *yyy* in the material number *zzz* is very high as *xxxExx*. Use a positive number <=1.
36. (NUTS INPUT) Rock longitudinal dispersivity input as *xxxExx* for material number *zzz* is larger than the maximum hydrological distance *yyyEyy*. Please check the value.
37. (NUTS INPUT) Rock transverse dispersivity input as *xxxExx* for material number *zzz* is larger than longitudinal dispersivity *yyyEyy*. Please check the value.
38. (NUTS INPUT)(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).
39. (NUTS INPUT)(In debug file) Undesirable value(s) is detected in the input file (NUTS INPUT 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.

1. NORMAL COMPLETION READING THE INPUT FILE FROM BRAGFLO INPUT
2. NORMAL COMPLETION READING THE INPUT FILE FROM TEST FILE INPUT.
3. NORMAL COMPLETION READING THE INPUT FILE FROM NUTS INPUT.

Information Only

10.0 REFERENCES

Amter, S., E. Behl and B. Ross 1988. "Carbon-14 Travel Time at Yucca Mountain", Disposal Safety Inc., Washington, DC.

Barenblatt, G. J. and Y. P. Zheltov 1960. "Basic flow equations for homogeneous fluids in naturally fractured rocks." *Doclady Akademii Nauk* USSR 132(3) pp.545-548.

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.

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

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.

Doctor, P.G. et al. (1992). "An Example Postclosure Risk Assessment Using the Potential Yucca Mountain Site", Pacific Northwest Laboratory, Washington.

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.

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.

Kim, S. 2022a. WIPP PA Requirements Document, Verification and Validation Plan, and Validation Document for NUTS Version 2.08. Sandia National Laboratories, Carlsbad, NM. ERMS #573823.

Kim, S. 2022b. WIPP PA Design Document and User's Manual for NUTS Version 2.08. Sandia National Laboratories, Carlsbad, NM. ERMS #573824.

Kim, S. 2022c. WIPP PA Implementation Document for NUTS Version 2.08. Sandia National Laboratories, Carlsbad, NM. ERMS #573825.

Litvak, B.L. 1985. "Simulation and Characterization of Naturally Fractured Reservoirs", Proc. Res. Characterization Technical Conf., Dallas Academic Press, NY.

Long, J. J. 2020. Nuclear Waste Management Procedure NP 19-1 Software Requirements, Revision 19. Sandia National Laboratories, Carlsbad, NM. ERMS #573707.

van Golf-Racht, T. D. 1982. Fundamentals of Fractured Reservoir Engineering, 1st Ed., Elsevier Science.

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.

Information Only

11.0 APPENDICES

11.1 Appendix A – ASCII input files for the CRA19 analysis

11.1.1 ASCII input file for NUTS SCN run mode for scenario 2

```
** NUTS TITLE (Thomas
Lowry,5/13/05,A.E.Ismail,12/10/07,6/22/09,S.Kim,2/6/13,B.Day,3/1/16,6/29/16,S.Kim,11/9/16,B.Day,0
6/29/17,R.Sarathi,03/06/19) **
'CRA19, NA, SCN_CRA19_S2, Screening Run'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,41
'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.D0
**  PRECIPITATE IMPLICITNESS; 1.T/F,IF IMPLICIT 2.# OF ITERATION,TOL. *
.FALSE.
** IS MATRIX ADSORPTION 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 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 **
'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
```

Information Only

```
'TWASTE'  1   1
*** (1.SITE NAME, NUMBER OF GRIDS IN THE SITE 2.INDECES)1...NSITES ***
'WIPP_SITE' 33
  23,10,1   24,10,1   25,10,1   26,10,1   27,10,1   28,10,1   29,10,1
 32,10,1   33,10,1   36,10,1   37,10,1
  23,11,1   24,11,1   25,11,1   26,11,1   27,11,1   28,11,1   29,11,1
 32,11,1   33,11,1   36,11,1   37,11,1
  23,12,1   24,12,1   25,12,1   26,12,1   27,12,1   28,12,1   29,12,1
 32,12,1   33,12,1   36,12,1   37,12,1
** MATRIX ADSORPTION 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     33
  23,10,1   24,10,1   25,10,1   26,10,1   27,10,1   28,10,1   29,10,1
 32,10,1   33,10,1   36,10,1   37,10,1
  23,11,1   24,11,1   25,11,1   26,11,1   27,11,1   28,11,1   29,11,1
 32,11,1   33,11,1   36,11,1   37,11,1
  23,12,1   24,12,1   25,12,1   26,12,1   27,12,1   28,12,1   29,12,1
 32,12,1   33,12,1   36,12,1   37,12,1
'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  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 **
```

Information Only

11.1.2 ASCII input file for NUTS ISO run mode for scenario 2

```
** NUTS TITLE (Joseph
Kanney,7/9/05,A.E.Ismail,12/10/07,6/22/09,S.Kim&C.Camphouse,2/6/13,B.Day,3/1/16,6/29/2016,S.Kim,1
1/9/16,B.Day,06/29/17,R.Sarathi,03/06/19) **
'CRA19, NA, ISO_CRA19_S2, Isotope Run'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,41
'WIPP_SITE'   5
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'WIPP_SITE'
'AM241L'  'NONE'    'NONE'    'AM241L'
'PU239L'  'NONE'    'NONE'    'PU239L'
'PU238L'  'U234L'   'NONE'    'PU238L'
'U234L'   'TH230L'  'PU238L'  'U234L'
'TH230L'  'TH230D'  'U234L'   'TH230L'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
5
'AM241L'   .FALSE.  .FALSE.
'PU239L'   .FALSE.  .FALSE.
'PU238L'   .FALSE.  .FALSE.
'U234L'    .FALSE.  .FALSE.
'TH230L'   .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
**  PRECIPITATE IMPLICITNESS; 1.T/F,IF IMPLICIT 2.# OF ITERATION,TOL. *
.FALSE.
** IS MATRIX ADSORPTION 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
```

Information Only

```
'PU238L'    3     1
'U234L'     4     1
'TH230L'    5     1
*** (1.SITE NAME, NUMBER OF GRIDS IN THE SITE 2.INDECES)1...NSITES ***
'WIPP_SITE' 33
  23,10,1   24,10,1   25,10,1   26,10,1   27,10,1   28,10,1   29,10,1
 32,10,1   33,10,1   36,10,1   37,10,1
  23,11,1   24,11,1   25,11,1   26,11,1   27,11,1   28,11,1   29,11,1
 32,11,1   33,11,1   36,11,1   37,11,1
  23,12,1   24,12,1   25,12,1   26,12,1   27,12,1   28,12,1   29,12,1
 32,12,1   33,12,1   36,12,1   37,12,1
** MATRIX ADSORPTION 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 **
```

Information Only

11.1.3 ASCII input file for NUTS INT run mode for scenario 2 at 100 year intrusion time

```
** NUTS TITLE (Thomas Lowry,5/21/05,Joseph
Kanney,7/11/05,A.E.Ismail,12/10/07,6/22/09,S.Kim&C.Camphouse,2/6/13,B.Day,3/1/16,6/29/16,S.Kim,11
/9/16,B.Day,06/29/17,R.Sarathi,03/06/19) **
'CRA19, NA, INT_CRA19_S2_T0100, Intrusion Run'
** 1.# OF SITES,# OF MATERIAL,(2.SITE NAME,# COMP. TO BE MODELED)1,..,NSITES **
1,41
'WIPP_SITE'   5
**(1. SITE, 2.COMP., DAUGHTER, PARENT, GROUP NAMES)1,..,NSITES **
'WIPP_SITE'
'AM241L'  'NONE'    'NONE'   'AM241L'
'PU239L'  'NONE'    'NONE'    'PU239L'
'PU238L'  'U234L'   'NONE'    'PU238L'
'U234L'   'TH230L'  'PU238L'  'U234L'
'TH230L'  'TH230D'  'U234L'   'TH230L'
** 1.# OF ELEMENT,(2.ELEM. NAME, TEMP. DEPEND., TABLE LOOK-UP)1,..,NELEMENT **
5
'AM241L'   .FALSE.  .FALSE.
'PU239L'   .FALSE.  .FALSE.
'PU238L'   .FALSE.  .FALSE.
'U234L'    .FALSE.  .FALSE.
'TH230L'   .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
**  PRECIPITATE IMPLICITNESS; 1.T/F,IF IMPLICIT 2.# OF ITERATION,TOL. *
.FALSE.
** IS MATRIX ADSORPTION 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 **
0.11044926D11    0.3155693D10    0.8D4
*** 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) **
```

Information Only

```
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' 33
  23,10,1   24,10,1   25,10,1   26,10,1   27,10,1   28,10,1   29,10,1
 32,10,1   33,10,1   36,10,1   37,10,1
  23,11,1   24,11,1   25,11,1   26,11,1   27,11,1   28,11,1   29,11,1
 32,11,1   33,11,1   36,11,1   37,11,1
  23,12,1   24,12,1   25,12,1   26,12,1   27,12,1   28,12,1   29,12,1
 32,12,1   33,12,1   36,12,1   37,12,1
** MATRIX ADSORPTION 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 **
```

Information Only

11.2 Appendix B – NUTS's ASCII flux field input file

11.2.1 Example of an ASCII flux 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
```

Information Only

11.2.2 Example of an ASCII flux 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
** 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
```

Information Only

```
** 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
** 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
** 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  1.E-2  1.E-2  1.E-2  1.E-2
1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2
1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2
1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2
1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2
1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2  1.E-2
```

Information Only