

4 INSTALLING AND RUNNING MARFA

MARFA is written in Fortran 95. A Fortran compiler that supports the allocatable component extension [ISO/IEC, 1998] is required to compile the source code. MARFA is otherwise platform independent. A generic *makefile* is included with MARFA. This *makefile* will work unmodified with gfortran on most systems. To compile with another compiler, edit the *makefile* and replace the Fortran 95 compile/link commands with the one appropriate for the local system. Compiler flags should also be replaced with appropriate values.

MARFA reads several files that contain data required for efficient sampling of retention time distributions. These are included in the directory named *data*. The required data files must be placed in the same directory, which can be placed in any accessible location on the local file system. The local file system path to the directory is part of the required input (See Appendix A).

Several example simulations are included with the MARFA installation package.

MARFA does not have a graphical interface. It is executed from the command line with one command line argument. All other input comes from a series of ASCII files, as described in Appendix A. Warning, error and other diagnostic messages are written to the screen and may be redirected to a file, if desired. Three output files are created in the run directory. One file contains particle arrival times and the other two contain reconstructed breakthrough curves on a mass or activity basis. Details of the output are described in Appendix B.

MARFA runs in two modes controlled by the command line argument. If an integer is specified as the command line argument, then it is interpreted as the number of particles used in the simulations. That situation results in the complete execution of the MARFA code, including particle transport calculations and breakthrough curve calculations. MARFA can also skip the particle transport calculation and reconstruct breakthrough curves from existing particle arrival times stored in the file *results.dat*. This option is activated using the command line argument *ppo* (post-processor only) by typing “marfa ppo.”

MARFA NR has an option to trace a subset of the particles and output their positions versus time. This capability is activated by including two integers on the command line. The first represents the number of particles simulated. The second is the number of particles to be traced.

APPENDIX A: MARFA SP INPUT

The units for all pathway and radionuclide property input parameters are various combinations of meters and years. The radionuclide source rate should be specified in mol/yr or Bq/yr.

Conventions used in this Appendix are as follows. Input parameter names are shown in **bold** font. File names are shown in *italics*. The `courier` font is used when specifying input file formats. A line consisting of a single colon in the input block means that lines are skipped. An ellipsis (...) indicates that the item is to be repeated. Any text following an exclamation point in an input block is to be regarded as a comment or explanation.

A minimum of 5 input files are required for MARFA runs. Users must specify radionuclide decay chains in the *nuclides.dat* file, the set of rock types in the *rocktypes.dat* file, equilibrium distribution coefficients in the *kdbins.dat* file, radionuclide sources in *sources.dat*, and the pathway properties in *trajectories.dat*. These five files must exist and must reside in the run directory. Symbolic links may be used in the input file if that is more convenient. If time-dependent distribution coefficients are used, a sixth file *kdhistory.dat* is also required. If only one set of distributions coefficients is specified per rock type, the Kd's are then assumed to be static and the *kdhistory.dat* file is not read.

MARFA uses the system random number generator in its Monte Carlo algorithms. An option exists to supply the seed for the random number generator. This is useful in testing when it is desirable to use the same sequence of random numbers. If the file *seedfile.dat* exists, then the seed is read from that file. Otherwise the random seed is taken from the system clock. Regardless of how the seed is specified, the seed used in the calculation is always written in the header of the *results.dat* file. The seed may be copied manually from the *results.dat* file to the *seedfile.dat* if reproducing the same run is of interest.

If the downscaling algorithm is to be used (i.e., at least one rock type is specified as stochastic), then a set of subgrid trajectories must also be provided. In contrast to the other input files, the subgrid trajectories data need not reside in the run directory. The location (path) of the subgrid trajectories directory is specified in the *rocktypes.dat* file.

In addition to the user-supplied input, MARFA also requires access to a set of files that contain data for lookup tables used in sampling the retention times. The required data files are included as part of MARFA software repository. The location (file path) of the required data files on the local file system must be specified in the *rocktypes.dat* file.

Any input file may start with an optional header section. The header section contains an arbitrary number of header lines. A header line is identified by an exclamation point "!" as the first non-black character. Comment lines are not permitted after the input starts.

Partial lines of comment may follow the required data fields on a line, however. The following subsections give the required data format after the header section.

A.1 Specifying the Decay Chains

The format for the *nuclides.dat* file is

```
nelem
elem_name !one line for each element(nelem times)
:
nnuc
nuc_name lambda associated_elem nextnuc imprtnce nsplit
!above line repeated nnuc times
```

Here **nelem** is the number of elements in the decay chain, and **elem_name** is the name of the element. The list of element names is used only to associate each nuclide with the appropriate element. Retention properties are entered for each element, as described in the input for the *rocktypes.dat* file.

The input **nnuc** is the number of nuclides to be included in the transport calculation. Each nuclide is assumed to have, at most, one daughter nuclide (branching in a decay chain is not supported). The inputs **nuc_name** and **lambda** are the nuclide name and decay constant [T⁻¹] for the nuclide, and **associated_elem** is the name of the associated element. The next nuclide in the decay chain is specified by the **nextnuc** parameter. The value NULL for a **nextnuc** parameter indicates that the nuclide is at the end of the chain. Each specified **associated_elem** must be in the list of element names, and each **nextnuc** must be in the list of nuclide names.

The input **imprtnce** is a user-specified relative importance factor that may be used to force some nuclides to be sampled more frequently at the source. This option is useful, for example, for obtaining higher accuracy for risk significant nuclides. The probability of selecting a given nuclide is proportional to the total source strength for that particle multiplied by its **imprtnce** value. MARFA assigns each particle of a given nuclide type a statistical weight inversely proportional to the nuclide's **imprtnce** value, thus ensuring that the resulting estimate is unbiased. As with any importance-weighted Monte Carlo sampling scheme, an informed choice for the importance value may decrease the variance in the result, but a poor choice will increase the variance.

The input **nsplit** is an optional parameter controlling splitting at each decay event. If it is present and greater than 0, the progeny particle will be split into **nsplit**+1 particles upon a decay event. This option is useful when modeling transport of decay chains when one of the radionuclides has a very long half-life.

Note the nuclide list may include multiple chains and nuclides that do not belong to any chain. The user simply needs to specify the decay product for each radionuclide. Converging chains are supported, but branching decay chains are not.

An example *nuclides.dat* file is shown in Figure A-1. This example has 7 radionuclides and two chains: $^{241}\text{Pu} \rightarrow ^{241}\text{Am} \rightarrow ^{237}\text{Np} \rightarrow ^{233}\text{U}$ and $^{246}\text{Cm} \rightarrow ^{238}\text{U}$. The radionuclide ^{99}Tc is not modeled as part of a chain.

```

6
Pu
Am
Np
Cm
U
Tc
7
Pu241      4.8e-2   Pu      Am241  1.0
Am241      1.6e-3   Am      Np237  1.0
Np237      3.2e-7   Np      U233   1.0
U233      4.37e-6  U       NULL   1.0
Cm246     1.46e-4  Cm      U238   1.0
U238     1.55e-10 U       NULL   1.0
Tc99      3.25e-6  Tc      NULL   1.0

```

Figure A-1. Example *nuclides.dat* file.

A.2 Specifying the Radionuclide Source

The radionuclide source is specified in the *source.dat* file. The source is to be specified as a source rate in mol/yr or Bq/yr. The format is

```
units          ! mol/yr or Bq/yr
samplemethod   ! keyword controlling source sampling
nsources       ! number of sources
sourceID       ! beginning of source block
traj1 traj2    ! Range of trajectories associated with source
ndpts
  time RN1 RN2 RN3 ...          ! one for each nuclide
  :      ! repeat above line for a total of ndpts lines
  :      ! repeat above source block for a total of nsources blocks
```

Prior to versions SP, the total number of particles to be used was specified in the *source.dat* file. The number of particles is now specified on the command line as a command line argument. *x*. The **units** parameter is used to specify the source units. The options are “mol/yr” or “Bq/yr”. The optional keyword **samplemethod** controls the sampling of the source. The default method is to sample the source history with uniform statistical weights for each particle of a given radionuclide. If the **samplemethod** keyword is present and specified as “UNIFORM IN TIME”, sampling of the source will be uniform in time with appropriate statistical weighting applied to each particle, as described in Section 3.3 of this document. The parameter **nsources** is the number of source locations (e.g., failed canisters). For each source the **sourceID** is a 10character identifier for that source. The parameters **traj1** and **traj2** identify the upper and lower indices in the trajectory list for those trajectories that will be associated with each source. The number of time points in a source history is **ndpts**. For each value of **time**, **nnuc** values of the source release rate are read, where **nnuc** is the number of nuclides (see Section A.1).

An example *source.dat* file is shown in Figure A-2. This example has two source locations and a dirac- δ function input, which is approximated as a square pulse of duration 0.1 years. The source strength is 0.001 mol/yr for the first radionuclide and 0.005 mol/yr for the second nuclide in the two-member chain, where the net cumulative source strength is computed using a linear interpolation of the defined mass release rates.

```

mol/yr      ! Units
2           ! Two source locations
Can1       ! Canister 1
1 20       ! Canister 1 is associated with trajectories 1 through 20
2          ! Two times are used to specify source
0.0 0.001 0.005 ! Source strength for RN1 and RN2 at 0 years
0.1 0.001 0.005 ! Cumulative source for RN1 and RN2 at 0.1 years
Can2       ! Canister 2
21-40     ! Canister 2 is associated with trajectories 21-40
2          ! Number of times used to specify source
0.0 0.001 0.005 ! Source strength for RN1 and RN2 at 0 years
1. 0.001 0.005 ! Cumulative source for RN1 and RN2 at 0.1 years

```

Figure A-2. Example *source.dat* file.

A.3 Defining Rock Types

The *rocktypes.dat* file specifies the retention model and retention properties for each rock type. In addition, the file specifies stochastic or deterministic mode for each. The format is

```

dpath !path to directory for data files
sbtrjdir !path to directory for subgrid trajectory files
numtypes ! number of rock types defined
typeID!length 5 character name for the rocktype
alpha stochasticflag numdir numvc
ret_model
phys_params ! element-independent parameters
ret_params ! retention parameters for each element
: ! above line is repeated for each element
: ! repeat block for a total of numtypes

```

Here **numtypes** is the number of rock types modeled. The **dpath** parameter provides the file system path to the lookup tables used to define retention time distributions, which are provided as part of MARFA. The **sbtrjdir** parameter provides the file system path to the subgrid trajectories data files for stochastic rock types. The **sbtrjdir** parameter is read but not used if all rock types are deterministic. The input parameter **typeID** is a 5-character string identifying the rock type. The input block headed by a **typeID** is repeated for a total of **numtype**. If retention parameters are modeled as constant in the entire domain, then only one rock type needs to be defined.

The parameter **alpha** is longitudinal dispersivity [L]. If **alpha** is entered as “0,” the longitudinal dispersion calculation is skipped. Note that the longitudinal dispersion calculation may increase the run time significantly.

The single-character input field **stochasticflag** indicates whether the rock type is stochastic or deterministic. Allowed values are S for stochastic or D for deterministic. For deterministic rock types, no other information is read. For stochastic rock types, the **numdir** and **numv** parameters are read from the same line. The **numdir** parameter specifies the number of directions for the subgrid trajectories (see Section 2). A separate data file with subgrid trajectories is required for each of the **numdir** directions, as explained in Section A.5. The **numv** parameter is a numerical parameter used in the stochastic downscaling algorithms. Details of the **numv** parameter are provided by Painter and Cvetkovic [2005].

The retention model is specified by the **ret_model** parameter. Allowed values are MD (unlimited matrix diffusion), LD (limited matrix diffusion), ES (equilibrium sorption), and TAB (tabular). The TAB option allows the user to specify a retention time distribution for each element as a lookup table. The format is different from the other retention models in that the TAB option redirects input to a user-specified directory. The structure of the lookup tables for the tabular option is described in Appendix A-4.

The next line **phys_params** contains physical parameters that are independent of chemistry and element types. The **phys_params** line differs depending on which retention model is specified. If ES or MD is specified, then bulk density [kg/m^3] and colloid mass density [kg/m^3] are read. If LD is specified, those two parameters are read and a third parameter Δ [m], the size of the matrix region accessible by diffusion, is also read. If TAB is specified, then the only parameter read is the size of the accessible matrix region in meters. For a multilayer matrix, this is the sum of all matrix thicknesses.

The next **nelem** lines contain the retention parameter values **ret_params** that differ by element but are independent of chemistry. Retention parameters for the ES, MD and LD options are read in the same order as the element list in *nuclides.dat*, with one set of retention parameters for an element on each **ret_params** line. The input read from these lines depend on which retention model is specified. If ES is specified, then the only parameter read for each element is the porosity. Note that MARFA has the option to make the porosity different for each element to accommodate anion exclusion, which is often represented by lower the effective porosity relative to that of cations. If MD or LD is specified, then matrix effective diffusion coefficient in units of m^2/yr is also specified after the matrix porosity. Care should be taken not to confuse the effective matrix diffusion coefficient with the pore diffusion coefficient. If TAB is specified as the retention model, then the **nelem** lines with **ret_params** are not read.

It is important to note that chemistry-dependent parameters, specifically K_d 's, are not specified in the *rocktypes.dat* file. Those parameters are specified in the *kdbins.dat* file, which is described in Section A.x below.

An example *rocktypes.dat* files is shown in Figure A-3. Three rock types are defined in this example: TUN1, DFN1, and CPM1. The TUN1 rock type uses the equilibrium sorption model, has a dispersivity of 2 m, and is deterministic. A rock type defined this

way might represent, for example, a backfilled repository tunnel. The DFN1 rock type is deterministic and uses the limited diffusion model (e.g., an explicitly represented fracture network). The CPM1 rock type also uses the limited diffusion retention model, but is stochastic (e.g., a continuous porous medium region). The downscaling algorithm is used for this rock type. The subgrid trajectories for the downscaling algorithm is read from the directory SubgridTrajectories/CPM1/ in the working directory. One set of subgrid trajectories will be used (numdir = 1), which presumes an isotropic network. A total of eight velocity groups (numvc = 8) is used in the downscaling algorithm.

```
/home/marfadir/data
./SubgridTrajectories/
3      ! Number of rock types
TUN1   ! Rock type name
2. D   ! Dispersivity, stoch/det flag (S/D)
ES     ! Retention model (ES/LD/MD)
2700. 0.0 ! rock density colloid mass
0.2    ! porosity for element one
0.2    ! porosity for element 2
DFN1 !name for second rock type
LD
0. D
2700. 0. 0.1 ! rock density, colloid mass, size of matrix
0.01 1.0e-7
0.01 1.0e-7
CPM1
LD
0. S 1 8
2700. 0. 0.1 ! rock density, colloid mass, size of matrix
0.01 1.0e-7
0.01 1.0e-7
```

Figure A-3. Example *rocktypes.dat* file with three rock types and 2 elements.

A.4 Defining sorption parameters

The file *kdbins.dat* defines the sorption parameters (Kd's) for each geochemical states. A geochemical state is defined as a combination of Kd bins (one bin for each radioelement if the retention model is ES; two bins for each radioelement if the retentioS1n model is LD or MD). Each rock type defined in the *rocktype.dat* file must have at least one state defined. The format is:

```
numtypes ! number of rock types.
  typeID ! 5-character identifier for the first rock type
  nbins ! number of geochemical states
    binlabel! 5-character identifier geochemical state 1
    Rc ! Retardation factor for colloids in state 1
    kdset1 !set of Kds for radioelement1 in state 1
    : ! repeat above line for all radioelements
    : !repeat above block for all states
    : ! repeat block for a total of numtypes
```

The *kdbins.dat* file must be consistent with the *rocktypes.dat* file. Specifically the number of rock types **numtypes** and the **numtype** values for **typeID** must be the identical in the two files.

The parameter **Rc** is retardation factor for colloids.

What is read in the **Kdset** arrays depends on the retention model specified. If ES is specified, the equilibrium distribution coefficient for sorption onto immobile mineral grains [m^3/kg] is read followed by the distribution coefficient for sorption onto colloids [m^3/kg]. If MD or LD is specified, the **Kdset** arrays have three values each: the matrix Kd [m^3/kg], the distribution coefficient for sorption onto colloids [m^3/kg], and the fracture surface sorption coefficient k_a [m^{-1}].

A.5 Transient sorption parameters

The *kdhistory.dat* file is used to associate a sequence of geochemical states (kd bins, see section A.4) to each trajectory segment. The *kdhistory.dat* file is not read if there is at most 1 Kd set for each rocktype. Otherwise, an integer sequence representing indices into the array of geochemical states (see previous section) is specified for each segment in the *kdhistory.dat* file. In addition, the times at which the geochemical states change are specified. The format is as follows.

```
ntraj          ! number of trajectories
  TrajID ! trajectory ID
    nstates
    statelist
    timelist
    : ! above line is repeated for each segment
  END          ! end of trajectory delimiter
  : ! repeat block for each trajectory
```

The first line read is an integer **ntraj** representing the number of trajectories. This value must match the *trajectories.dat* file. The next line read is a trajectory ID and should match those in the *trajectories.dat* file. After that line, $3 \times \mathbf{nsegs}$ lines are read, where **nsegs** is the number of segments in the trajectory (comes from the *trajectories.dat* file). For each segment, the first line read contains the number **nstates** of chemical states (Kd bins) associated with each segment. The second line contains the list **statelist** of those states in order of their occurrence. The set of Kd bins defined in the *kdbins.dat* file provide the allowed values for the entries in the **statelist** array. The third line contains the real list **timelist** when the states change. The integer sequence **statelist** is of length **nstates** and represents labels corresponding to the possible chemical states for the rocktype. The real array **timelist** is of length **nstates**-1. The convention in MARFA is that the first chemical state for each segment begins at time zero, and the last chemical state persists until the end of the simulation. Thus, the number of time values to specify is one less than the number of states. In the event there are only one states for a segment, the a blank line is to be entered for the **timelist**.

A.6 Defining Rock Types using the TABular option

A tabular retention model is also available (TAB) option. This option allows the user to specify a retention time distribution for each element as a lookup table.

Each specified directory should contain one file for each element. The file will be named **retdist.dat1** for element 1, **retdist.dat2** for element 2, etc. Each of these files contains the retention time distribution conditional on β . The format is

```

numbeta !number of tabulated beta values
beta    ! beta value
          nquantiles      ! number of quantiles used to define the
distribution
          Q1 T1 !quantile, retention time
          : !repeat above line for a each quantile-time combination
          : !repeat above blocks for a total of numbeta vslues

```

In addition, the directory should contain one file named *auxdata.dat*. This format for this file is

```

ka Dbar !
: ! repeat above line for each element

```

The parameter **ka** is k_a [1/L], a sorption coefficient for equilibrium sorption on fracture surfaces. **Dbar** is a weighted diffusion coefficient [m^2/yr] across the matrix layers. **Dbar** and **Delta** are used only when there are flow changes.

Although a user may specify any retention time distribution through the tabular option, the primary intended use is for multilayer matrix diffusion models. Included with MARFA is a helper application in the form of two Mathematica scripts *ThreeLayerRetention.m* and *3LayerNoInterface.m*. These scripts would typically be called by a third user-customized script to create the retention time distributions for a 3-layer matrix model. The helper applications are currently configured to produce tables ranging from $\beta=1$ to 10^7 yr/m. Included with Verification Tests 9a is an example of such a script that can serve as a template for using the helper application.

A.7 The trajectories.dat File

Trajectories, as calculated from CONNECTFLOW nested DFN/CPM regional scale models, are read from a file named *trajectories.dat*. Two possible formats of this file are available. With the first format, the trajectory data are fully contained within the *trajectories.dat* file. With the second format, the *trajectories.dat* file contains the location of a CONNECTFLOW data file containing the trajectory information.

Format 1 is described as follows.

```

ntraj ! number of trajectories
TrajID xloc yloc zloc! Trajectory ID and location
  xe ye ze typeID tau beta cpmgradh ! segment data
  : ! repeat above line for each segment
  END ! end of trajectory delimiter
  : ! above block repeated for each trajectory

```

The parameter **ntraj** is the number of trajectories; **TrajID** is a 10-character label identifying the trajectory; and **xloc**, **yloc**, and **zloc** are the x-, y-, and z-positions for the trajectory head. The **TrajID** is employed by MARFA to identify specific trajectories, such that each **TrajID** needs to be unique. Sources are associated to trajectories by their sequential order, matched to the trajectory range defined in *source.dat*. The data read for each segment include the location of the end of the segment (**xe,ye,ze**) and a 5-character identifier for the rock type (**typeID**). The advective travel time (**tau**) and the F-quotient (**beta**) are also read. If the rock type is a stochastic one, then **tau** and **beta** represent CPM (upscaled) values; otherwise, they are DFN values for the segment. If the particular rock type is a stochastic one, then the magnitude of the hydraulic gradient (**cpmgradh**) at that location is also read. If the particular rock type is a deterministic one, then **cpmgradh** is not used. The hydraulic gradient is used to adjust (scale) the stochastically generated segment properties to the local macroscopic conditions. The scaling is necessary because the subgrid trajectories will, in general, be generated using an imposed hydraulic gradient different from the macroscopic gradient resulting from the CPM flow simulation.

Format 2 is described as follows.

filename ! filename of data file containing trajectory data.
first character of filename needs to be nonnumeric

The parameter **filename** is the name of the data file that contains the trajectory data. The required format of the file named in *trajectories.dat* is the CONNECTFLOW modified verbose format for pathline calculations. That format allows for a single comment line. The format following the comment line is described as follows.

```
ntraj REAL TIMESTEP
PART TRAJ_NUM REALISATION_NO START TIME flag
xloc yloc zloc XN XN XN UX UY UZ TOT_TIME TOT_LENGTH
TRAJ_BETA
STEPS num_seg
CUM_TIME xe ye ze HYD_AP TRANS_AP tau beta cpmgradh typeID
: !repeat until the end of the trajectory is reached
: !repeat until all of the trajectories have been entered
```

Several of the parameters in the input file are not employed by MARFA, but have been retained for consistency between MARFA and CONNECTFLOW. These unused data fields are labeled in uppercase lettering in the previous description; the unused data fields are expected as placeholders. The parameter **ntraj** is the number of trajectories. The parameter **flag** indicates whether the trajectory reaches a transport boundary; **num_seg** is the number of segments on the trajectory; and **xloc**, **yloc**, and **zloc** are the x-, y-, and z-positions for the trajectory head. The data read for each segment include the location of the end of the segment (**xe, ye, ze**); a five-character identifier for the rock type (**typeID**); the advective travel time (**tau**); the magnitude of the hydraulic gradient (**cpmgradh**), and the F-quotient (**beta**).

Three optional sections exist for the file *trajectories.dat*, independent of the format: "FLOW CHANGES", "VALID TRAJECTORIES" and "CHANNELING FACTOR". These three sections are key-phrase driven and may come in any order and are not required to be present. The optional keywords and associated input format are as follows.

```
CHANNELING FACTOR chanfac
FLOW CHANGES nchanges
time_of_change vscale
: !above block repeated for each flow change to be analyzed
VALID TRAJECTORIES
Trj_num terminal_segment
: !above block repeated for each trajectory to be analyzed
END
```

The optional key phrase "CHANNELING FACTOR" allows the user to specify a global factor **chanfac** that multiplies the segment beta values read from a PTV file (Format 2).

This keyword has no effect in Format 1. The channeling factor also has no effect for rock types that are defined as having stochastic pathway variability (see Section A.3)

The key phrase “FLOW CHANGES” initiates an optional section that allows the user to specify velocity changes. The parameter **nchanges** is user defined, establishing the number of flow changes to be evaluated. The parameter **time_of_change** sets the time of a flow change (defining end of a flow period). The parameter **vscale** established the velocity change during the flow period. If the optional section “FLOW CHANGES” is not present or if **nchanges** is equal to zero, no flow changes are evaluated. Note under all conditions the number of retention parameter sets defined for each hydrofacies in *rocktypes.dat* must match the number of flow periods (**nchanges** + 1).

The key phrase “VALID TRAJECTORIES” initiates a section that allows the user to specify a subset of trajectories for particle transport analysis. The parameter **trj_num** identifies the index in the trajectory list of the trajectory to be evaluated. The parameter **terminal_segment** allows the setting of the monitoring location for each selected trajectory to a userdefined trajectory segment number. If the trajectory has fewer segments than the userspecified **terminal_segment** or if the user enters 0, the final segment of the trajectory will be used as the monitoring location. If the optional section “VALID TRAJECTORIES” is not present, all trajectories are evaluated, and the final segment of each trajectory is used as the monitoring location.

A word of caution regarding the specification of the terminal segment: if the trajectory input contains a segment with zero value for τ , the internal indexing of segments in MARFA will be different from the input. If MARFA encounters a zero- τ segment, then that segment is ignored, and indexing for subsequent segments in the trajectory is adjusted.

An example *trajectories.dat* file is shown in Figure A-5. This example uses Format 1 and has one trajectory with eight segments. The trajectory starts at the location (0,0,-6) and terminates at (11,0,250) after passing through the three rock types defined in Figure A-5. Note that the trajectory begins in an explicit DFN region, enters a repository tunnel, and then returns to an explicit DFN region before entering a CPM region. In the section FLOW CHANGES, one flow change has been specified, which occurs at 10,000 yrs with a relative change in velocity of 1.5. In the section VALID TRAJECTORIES, trajectory one has been included for analysis with breakthrough monitoring at segment seven.

An example *trajectories.dat* file for format 2 is shown in Figure A-6. The file containing the trajectory data is *nro.ptv*. In the section “FLOW CHANGES”, no flow changes have been defined. In the section “VALID TRAJECTORIES”, three trajectories will be used in the particle transport calculations. Trajectories 1, 15, and 31 terminate at segments 100, 150, and 120, respectively.

```
1
CAN1 0. 0. -6.0
  1 0. -5.0 DFN1 3.0 2.0d4 0
 11 0. 0. TUN1 100.100. 0.0
 11. 0. 0.57073 DFN1 1.1105 48031. 0
 11. 0. 2.72963 DFN1 0.00095129 36.061 0
 11. 0. 3.62113 DFN1 3.0657 19465. 0
 11. 0. 4.97553 DFN1 3.8401 519210. 0
 11. 0. 10.0 DFN1 3.7925 88610. 0
 11. 0. 250. CPM1 100.1.0d6 0.001
END
FLOW CHANGES 1
10000.0 1.5
VALID TRAJECTORIES
  1. 7
END
```

Figure A-4. Example trajectories file (Format 1).

```
nro.ptv
FLOW CHANGES 0
CHANNELING FACTOR 0.1
VALID TRAJECTORIES
  1. 100
 15. 150
 31 120
END
```

Figure A-5 Example trajectories file with trajectories contained in an external CONNECTFLOW file (Format 2).

A.8 The Subgrid Trajectories

For each stochastic rock type, the user must supply sets of subgrid trajectories, one set for each direction.

As discussed in Section 2, the subgrid trajectories are used to recover the effects of subgrid scale variations in velocity, which were lost in the CPM representation of flow. Specifically, the subgrid trajectories provide the raw data for the Painter and Cvetkovic [2005] upscaling/downscaling algorithm. The full path name for the directory containing the subgrid trajectories is read from the *rocktypes.dat* file. Contained in that directory are several subdirectories, one for each stochastic rock type. Each subdirectory must be named as the corresponding rock type identifier (**typeID**). Each subdirectory must contain several files, one for each direction in the direction set. The file name for *i*-th direction must be formed by appending *_i* to the **typeID**. Thus, the subgrid trajectory file for direction 2 of a rock type named CPM4 must be named CPM4_2.

The format for each subgrid trajectory file is

```
!header information
:
gradh dip strike vcpm ! imposed gradient and upscaled transport
velocity
nsgt ! number of subgrid trajectories in this set
  tau beta length ! segment data
  : ! repeat above line for each segment
  END ! end-of-trajectory flag
: ! above block repeated for each trajectory
```

Here **dip** and **strike** are the dip and strike angles (in degrees) characterizing the direction of the imposed hydraulic gradient used in constructing the given set of trajectories; **gradh** is the magnitude of the imposed gradient. The magnitude of the upscaled transport velocity is denoted **vcpm**. The parameter **nsgt** is the number of trajectories in the set. For each segment the parameters **tau**, **beta**, and **length** are read. These represent the advective travel time, F-quotient, and length for each segment, respectively. The END keyword denotes the end of a trajectory.

MARFA pre-processes the subgrid trajectory information to a form convenient for use by the downscaling algorithm. Because the pre-processing step may take several minutes for large sets of subgrid trajectories, MARFA saves the results of the preprocessing in intermediate files. Previously processed subgrid data may then be used in subsequent runs without re-running the preprocessor.

The existence of pre-processor output in the subgrid trajectory directories controls whether the pre-processor runs. If the subgrid trajectories directory contains

pre-processed data files and information extracted from the headers of these files is consistent with certain input from the *rocktypes.dat* file, then the pre-processor is skipped. Otherwise, it is called by MARFA to process the raw subgrid trajectories.

The naming convention for the processed subgrid data is identical to that of the subgrid trajectories, but without the rock type name. For example, the file *_1* in the *CPM1* directory is the processed subgrid data for direction 1 of rock type CPM1. The file *CPM1_1* in the same directory would contain the raw subgrid trajectories.

A.9 Postprocessor Control

The postprocessor calculates breakthrough curves based on the particle arrival times. The user may specify which trajectories are to be included in the breakthrough curve calculation. In addition, the times at which breakthrough values are needed may also be specified. The postprocessor options are read from the *postprocessor.dat* file. If the file *postprocessor.dat* is not present, the postprocessor will include all trajectories and calculate discharge at 200 times, spread uniformly between the maximum and minimum arrival times for each nuclide. In this mode of operation, postprocessor operation is fully automated and requires no user input.

The user may opt to include all trajectories or restrict the breakthrough curve to a subset of the trajectories. If a subset is required, the user must specify which trajectories are of interest by defining one or more batches of trajectories, where a batch is defined by a continuous range of indices in the trajectory list. A batch is specified by entering the lower and upper indices (inclusive) in the trajectory list.

Three options are provided for determining discharge (breakthrough) calculation times. The times may be determined automatically based on a uniform or logarithmic spacing, or specified manually.

The format for the *postprocessor.dat* file is as follows.

```
gamma t0 ! sensitivity parameter and time of first release
trajset ! Enter A for all or P for a subset
nbatches ! number of batches if Partial list selected
i1 i2 ! upper and lower indices defining a subset
: ! repeat for total of nbatches

disttype ! distribution type uniform, logarithmic, manual (U/
L/M)
nbrktimes ! number of times at which breakthrough values are
needed
time1 ! if manual is selected first time
: ! repeat for a total of nbrktimes
```

Gamma is a sensitivity parameter used by the postprocessor when reconstructing the instantaneous breakthrough curves. The default value for gamma is 0.2, which under most conditions results in an accurate and robust breakthrough curve. Under conditions that may result in prompt or rapid changes in breakthrough, however, larger values of gamma (e.g., 0.3 or 0.4) more accurately track the rapid transitions. Increasing the sensitivity will result in more statistical noise throughout the entire breakthrough curve.

The optional parameter **t0** is the first time at which releases into the far field are possible. This parameter allows the user to redefine the time “origin” and enforce a zero-release condition at that time. In a pinhole release case, for example, t0 would be set to the time that the hole appears.

The **trajset** parameter indicates whether the breakthrough will be based on all trajectories or a subset. Allowed values are A for all trajectories or P for a partial reconstruction based on a subset of the trajectories. If a partial reconstruction is specified, **nbatches** is the number of batches; **i1** and **i2** are the lower and upper indices for the batch.

The **disttype** parameter controls the times at which breakthrough curves are calculated. Allowed values are U for uniformly spaced between minimum and maximum arrival time, L for logarithmically spaced, and M for manually specified. The parameter **nrbktimes** is the number of times at which discharge is required. If times are manually specified, a list of length of **nrbktimes** must be given, one per line.

A.10 Resident Concentration Calculation

MARFA in static pathway mode includes an option to calculate resident concentration at specific points along the trajectory. Specifically, resident concentration is calculated for at most one segment along each trajectory. The monitored segment is controlled by specifying a set of rock types that are to be monitored. MARFA then selects the first segment of each trajectory that falls in one of the specified rock types. If a trajectory does not pass through any of the specified rock types, resident concentration is not calculated. If a trajectory has more than segment in each rocktype, only the first one along the trajectory is monitored. The capability is controlled through an input file called *rc_control.dat*.

Three options are provided for specifying the times at which resident concentration is to be calculated. The times may be determined automatically based on a uniform or logarithmic spacing, or specified manually.

The format for the *rc_control.dat* file is as follows.

```
num_monitored_rt    ! Number of rock types to be monitored  
rocktype1         ! name of monitored rocktype  
:                   ! repeat for total of nbatches  
disttype          !distribution type uniform, logarithmic, manual (U/L/M)  
nbrktimes         !number of times at which concentration is needed
```

The next set of input is dependent on the value of **disttype**. If manual is selected a list of times is read

```
time1            ! if manual is selected - first time  
:                 ! repeat for a total of nbrktimes
```

If U or L is specified one line of input is read containing first and last times that resident concentration will be calculated.

```
firsttime lasttime      ! first and last time
```

APPENDIX B: MARFA NR INPUT

MARFA NR requires several input data files. Files that define the radionuclide decay chains, the rock properties, equilibrium distribution coefficients, and the postprocessor control are *nuclides.dat*, *rocktypes.dat*, *kdbins.dat* and *postprocessor.dat*, respectively. Those files are the same as the corresponding files for MARFA SP. The *source.dat* file contains radionuclide source histories and is only slightly different from the corresponding version SP input. Similarly, the *kdhistory.dat* file is nearly identical between the two versions. In addition to those files, MARFA NR requires files that describe the node networks, the time of flow changes, and the number of visits a particle can make to a node before it is terminated.

B.1 The *flowdata.dat* File

Information about the flow periods is contained in the *flowdata.dat* file. This file is required and must have the name *flowdata.dat*. The format of the *flowdata.dat* is

datadirectory

VISITS ALLOWED **maxvisits**

FLOW PERIODS **nfp**

Time iufp ! one time needed for each flow period

NODE DATA FILES **nufp**

nodedatafile ! one file needed for each unique fp

The character string **datadirectory** is the path to the directory containing the lookup tables, as in Version SP. This character string is required and must be on the first line. The remainder of the **flowdata.dat** file is key-phrase driven. The valid key phrases are "VISITS ALLOWED," "FLOW CHANGES," "NODE DATA FILES," "TRANSLATION DATA FILES," and "CHANNELING FACTOR." The key phrases may come in any order after the first line. Each key phrase is followed immediately by corresponding input.

The integer **maxvisits** following the optional key phrase "VISITS ALLOWED" is the number of times a particle may visit a node. If a particle visits a node more than **maxvisits** times, the particle is terminated as described in Section 2.2. If the "VISITS ALLOWED" key phrase is not present in the file, **maxvisits** defaults to 1.

The integer **nufp** immediately following the required key phrase "NODE DATA FILES" is the number of unique flow periods. Following the key phrase, a total of **nufp** values of the character string **nodedatafiles** are read. These strings specify the locations of the node network data files in the MARFA NR format (see Section 3.2). A total of **nufp** files are required, one for each flow period.

The integer **nfp** following "FLOW PERIODS" is the number of flow periods during the simulation. After this keyword is read, **nfp** values of the **time** and **iufp** fields are then read, one value of each per line. Each value of the **time** field is the time in years at the end of a flow period. The first period is presumed to start at time 0. The **iufp** are indices

into the list of node data files. For example, a value of 2 for **iufp** means the flow period in question is associated to the second node data file. Note that each **iufp** must be less than or equal to **nufp**. Note that this format allows more flow periods than **nodedatafiles**. This allows the node data files to be cycled, if for example, a repeated glacial cycle is being modeled.

The optional key phrase “CHANNELING FACTOR” allows the user to specify a global factor **chanfac** that multiplies the segment beta values in the **nodedatafiles**.

The optional key phrase “ASCII” indicates that the node data files are to be given in ASCII format. If this keyword is not present, they are written as unformatted data. The unformatted files load faster.

B.2 The Node Network Data Files

Information about the groundwater velocity field in each flow period is provided to MARFA NR in the form of a directed graph, which provides node-to-node connectivity. The format of a node network data file is

```

num_nodes
  node_num xloc yloc zloc rocktypeID !single node data
  num_neigh
  neighi      ...      !repeated a total of num_nodes times
  probi       ...
  taui        ...
  lengthi    ...
  betai       ...
  : ! Repeat node data structure for each node

```

The field **num_nodes** is the total number of nodes contained in the data file. The integer field **node_num** and the real fields **x**, **y**, **z** and **rocktypeID** are the node number, its spatial coordinates and its rocktype. The integer **num_neigh** is the number of nodes immediately downstream of the current node. The node numbers for downstream nodes are given in the **neigh** fields. A total of **num_nodes** must be given, all on one line. The **prob** fields are the probabilities of advancing to each downstream node. The probabilities are input as a cumulative probability for neighbors 1 through **num_nodes**. The pathway to each downstream node is defined by a transport time, length, retention parameter, and rock type, which are **tau**, **length**, and **beta**, respectively. The definitions of these parameters are the same as in Version SP.

A helper application named *extractnodedata* is included with MARFA NR. The *extractnodedata* application prompts for a NAPSAC extended ASCII format (NEF) file describing the fracture system and a CONNECTFLOW binary pathline library (PTH) file. The application writes a file named *nodedata.dat* in the correct format for MARFA NR.

B.3 The *source.dat* File

The radionuclide source is specified in the *source.dat* file. The source is to be specified as a source rate in mol/yr or Bq/yr. The format is nearly identical to Version SP.

```
npart
units ! mol/yr or Bq/yr
samplemethod ! keyword controlling source sampling
nsources ! number of sources
sourceID ! beginning of source block
nnodes ! number of nodes associated with this source
nodenum ! node number for node associated with source
: ! repeat for a total of nnodes lines
ndpts
  time RN1 RN2 RN3 ... ! one for each nuclide
: ! repeat above line for a total of ndpts lines
:! repeat above source block for a total of nsources blocks
```

The parameter **npart** is the total number of particles to be used. The **units** parameter is used to specify the source units. The options are “mol/yr” or “Bq/yr.” The optional keyword **samplemethod** controls the sampling of the source. The default method is to sample the source history with uniform statistical weights for each particle of a given radionuclide. If the optional **samplemethod** keyword is present and specified as “UNIFORM IN TIME,” sampling of the source will be uniform in time with appropriate statistical weighting applied to each particle, as described in Section 3.3 of this document. The parameter **nsources** is the number of source locations (e.g., failed canisters). For each source, the **sourceID** is a 10character identifier for that source. The parameter **nnodes** is the number of nodes associated with the specified source. The node numbers **nodenum** associated with the source are then read, one per line. The number of time points in a source history is **ndpts**. For each value of **time**, **nnuc** values of the source release rate are read, where **nnuc** is the number of nuclides.

B.4 Transient sorption parameters

The *kdhistory.dat* file is used to associate a sequence of geochemical states (kd bins, see section A.4) to each trajectory segment. As with version SP, the *kdhistory.dat* file is not read if there is at most 1 Kd-set for each rocktype. Otherwise, an integer sequence representing indices into the array of geochemical states (see previous section) is specified for each node in the *kdhistory.dat* file. In addition, the times at which the geochemical states change are specified. The format is identical to that of the SP version except that trajectory names and the delimiter END is not needed. The format is as follows:

```
nodes           ! number of nodes
  nstates
  statelist
  timelist
  : ! above 3 lines are repeated for each node
```

The first line read is an integer **nodes** representing the number of nodes. This value must match the *nodedata.dat* file. After that line, $3 \times \mathbf{nsegs}$ lines are read, where **nsegs** is the number of nodes (comes from the *nodedata.dat* file). For each node, the first line read contains the number **nstates** of chemical states (Kd bins) associated with each segment. The second line contains the list **statelist** of those states in order of their occurrence. The set of Kd bins defined in the *kdbins.dat* file provide the allowed values for the entries in the **statelist** array. The third line contains the real list **timelist** when the states change. The integer sequence **statelist** is of length **nstates** and represents labels corresponding to the possible chemical states for the rocktype. The real array **timelist** is of length **nstates**-1. The convention in MARFA is that the first chemical state for each segment begins at time zero, and the last chemical state persists until the end of the simulation. Thus, the number of time values to specify is one less than the number of states. In the event there are only one states for a segment, the a blank line is to be entered for the **timelist**.

APPENDIX C: MARFA OUTPUT

MARFA does not have a graphical user interface. Error and warning messages are written to the screen. MARFA also reports progress in initializing the calculation. Once the main Monte Carlo calculation starts, no output is provided until after all particles are processed.

Raw results in the form of particle arrival times are written to the file *results.dat* in the run directory. If the file does not exist, it will be created. If it does exist, it will be overwritten. The *results.dat* file starts with several lines of header information. The number of particles used and the random seed used are written in the header. Following the header, the output format is

```
npart !number of particles used
nnuc ! number of nuclides
nuc_name lambda ! nuclide name, decay constant
: ! repeat a total of nnuc times
atime inuc mass sourceID trajID ! data for each particle
: ! repeat for each particle arriving
```

Here **npart** is the number of particles used, **nnuc** is the number of nuclides modeled, **nuc_name** and **lambda** are the name and decay constant for each nuclide. One line of data is written for each particle that makes it to a trajectory terminus; **atime** is the arrival time and **mass** is the mass in moles of the particle. The integer **inuc** is an index into the nuclide list, and thus specifies the state of the particle upon arrival. The character string **sourceID** uniquely identifies the source location for the particle from the lists in Sections 5.2 and 5.4 and the **trajID** uniquely identifies the trajectory for the particle.

A post-processor calculates instantaneous and cumulative breakthrough curves from the arrival times. The file *Breakthrough_Mo.rlt* contains instantaneous mass discharge in units of mol/yr and cumulative mass discharge in units of mol. The file contains a global header section followed by several results sections, one for each nuclide type. The number of particles used in the simulation is written into the global header along with information about the version used, etc.

The format following the global header section is

```
!ID =1      Name= nname1 Number of particles = nscore1
!Time Cumulative Breakthrough Instantaneous Breakthrough
!      (Moles) (Moles/year)
!
time1          cbrkval1          ibrkval1
:              ! repeat above line for each estimation time
:              ! repeat above block for each nuclide modeled
```

Where **nname1** is the name of nuclide 1, and **nscore1** is the number of particle “scoring” (arriving at a pathway endpoint) as nuclide 1. The cumulative and instantaneous breakthrough curves are printed following that four line header, with **cbrkval** and **ibrkval** representing cumulative and instantaneous breakthrough at a given time (**time**).

The file *Breakthrough_Bq.rlt* is identical to *Breakthrough_Mo.rlt* except that results are in units of Becquerel per year.