# particleDriver – Amanzi-Walkabout interface

particleDriver is a C++ program for a particle advection system using Amanzi, LaGriT, and Walkabout. It automates the process of producing input and output files for LaGrit and Walkabout from an Amanzi checkpoint file and creates data for visualization that can be used in Paraview, PlumeCalc, and Meshlab.

The workflow of this program is as follows:

RUN AMANZI to create h5 checkpoint file.

RUN particleDriver

If pre, preprocess files. Read Amanzi h5 file and write Walkabout and LaGriT input files. These include mesh cell center and edge points, properties for the points, cell velocities, and control files.

If runLagrit, use LaGriT command file to connect the Amanzi points into a tet mesh. Write the element adjacency file and FEHM format files for the tet mesh. These files include mesh geometry; mesh node zone files, and the sparse matrix connectivity and area interface file.

If runWalkabout, run Walkabout with the files written by particleDriver.

If plumecalc.files, create and write rock, and sim files for PlumeCalc

If post, process mesh and velocity data into viewable formats.

# Known Issues

1) Material (block) information not passed, so the mesh nodes and elements have a single material id of 1. This means that node zones cannot be defined for Walkabout without using LaGriT interpolation from the Amanzi mesh to the Walkabout nodes.

Additional complication, Amanzi block materials are consecutive, FEHM zone are not so that zone id numbers may need to be correlated. *(Note that block id may be passed in newer versions of Amanzi)*

2) The connected tet mesh does not directly correlate to the Amanzi hex or prism mesh elements. The Amanzi mesh is represented by nodes at the cell centers and nodes at cell vertices on the boundary. These mesh nodes are connected by LaGriT into tetrahedral elements, each node at a tet vertex.

3) By converting to tets there can be a large number of negative voronoi volumes and ccoefs along boundaries, and bad elements can be created across non-convex mesh shapes. The mesh size can be much larger than original if converting from Amanzi hex or prism to a tetrahedral mesh.

4) The Walkabout User Manual is V1.0 and seems incomplete with regard to usage and only mentions the FEHM style files required, not Amanzi files.

5) The Amanzi facesets for boundary faces are lost in the conversion to Walkabout node zones.Default mesh boundary zones are written based on the 6 normal directions of the mesh.

# Syntax

particleDriver <checkpoint file> [ pre | runLagrit | runWalkabout | post ]

pre Runs preprocessing commands for LaGriT and Walkabout

runLagrit Runs LaGriT to connect points into tets and write FEHM files

runWalkabout Runs Walkabout

post Runs post processing commands for data visualization

# Input Files for particleDriver

*walk####.h5* – Amanzi checkpoint file. Required

This file is required by particleDriver and is the first command line parameter. It is in HDF5 format and is produced by Amanzi. Configure Amanzi to produce this file by specifying the parameter **walkabout** to **true** in the xml file used to run Amanzi:

<ParameterList name="Checkpoint Data">

<ParameterList name="Cycle Data">

<Parameter name="End" type="int" value="-1"/>

<Parameter name="Interval" type="int" value="1000"/>

<Parameter name="Start" type="int" value="0"/>

</ParameterList>

<Parameter name="File Name Base" type="string" value="chkpoint"/>

<Parameter name="File Name Digit" type="int" value="5"/>

**<Parameter name="walkabout" type="bool" value="true"/>**

</ParameterList>

This file contains DATASET arrays with values representing the mesh cell center points and the cell vertex points on the mesh boundaries. The data arrays includes **x, y, z, pore velocity x, pore velocity y, pore velocity z, porosity, pressure,** and **saturation**.

*config.ini* – to specify Walkabout and LaGrit executables. Required.

This file must be present in the directory in order to run LaGriT and Walkabout. The label for the application is specified by the string of characters which come before the first colon (:), and the associated location is assumed to be everything after the first colon (:)

lagrit:/n/local\_linux/lagrit

walkabout: ~/bin/walkabout

*traj.out* – Walkabout output of particle trajectories. Optional

# Preprocess Files written by particleDriver

*control.dat* – specifies parameters. Required for Walkabout.

*.files* – specifies input/output files. Required for Walkabout.

*.ama* – cell velocity file. Required for Walkabout.

*.avs* – AVS file with point properties from .h5 file. Required for Walkabout

*.lgi* – command file to connect tets and write files. Required for LaGriT.

*.inp* – AVS file with point x,y,z from .h5 file. Required for LaGriT

*.rock* – density, Kd, porosity properties. Optional.

*.ply* – view file showing domain points colored by Velocity

# Output Files written by particleDriver/LaGriT

*.fehmn* – FEHM geometry file for the tet mesh. Required for Walkabout.

*.stor* – FEHM tet connectivity and interface areas. Required for Walkabout.

*.graph* – Mesh element adjacency list. Required for Walkabout.

*\_outside.zone* – FEHM boundary zones for mesh points. Optional.

*\_outside\_vor.area* – FEHM Voronoi areas for boundary points. Optional.

*\_material.zone* – FEHM material zones for mesh points. Optional.

*.tet.inp* – AVS file for viewing the tet mesh.

*.gmv* – binary GMV file for viewing the tet mesh.

*logx3dgen, outx3dgen* – LaGriT log and screen output files.

# Post process Files written by particleDriver

*.obj* – Wavefront OBJ file for viewing tetrahedral mesh.

*.csv* – text file for viewing cell center point positions and velocities.

*.vtk* – binary VTK file for viewing particle path lines from Walkabout.

Details for each file type are given in the Walkabout UM appendix.

This Section is copied from the Appendix in the V1 2011 User Manual.

It does not include the Amanzi style files used specifically in the Amanzi-Walkabout workflow as used and written by particleDriver program.

# Appendix A. Input and Output Formats

### The control file

The control file contains two keyword blocks that allow the user to specify dispersion tensors and particle start locations. In addition, it contains numerical parameters that control the functioning of the particle tracking simulation.

* + 1. *INITIAL* keyword block

Particle starting positions are specified in the *INITIAL* keyword block. The keyword block may be placed anywhere in the **control** file. The form of the *INITIAL* keyword block is as follows

*INITIAL* ! case sensitive

### dist\_type

**num\_part** ! conditional input depends on **dist\_type location\_info** ! conditional input depends on **dist\_type**

: ! repeat as needed

The dist\_type keyword specifies how particles are to be distributed. Allowed values are *MANUAL*, *UNIFORM*, and *RANDOM* (case sensitive).

If *MANUAL* is specified then a particle location must be specified for each particle. In this case, the number of particles should be entered in the **num\_part** field and start coordinates for each particle should be entered (**location\_info** field, one x,y,z triplet on each line).

If *RANDOM* is specified, the particles are to be distributed randomly in a user- specified rectangular box. In this case, the number of particles should be entered in the **num\_part** field and the **location\_info** has two lines representing locations of minimum (first line) and maximum (second line) coordinates of the bounding box for the start region.

The *UNIFORM* keyword is similar to the *RANDOM* keyword except that particles will be distributed uniformly in the region of interest. In this case, three integers should be entered in the **num\_part** representing spacing in the x, y, and z directions. The **location\_info** lines are identical to the *RANDOM* case.

* + 1. *DTENSOR* keyword block

Dispersion tensor information is given in the *DTENSOR* keyword block. The keyword block may be placed anywhere in the **control** file. The form of the *INITIAL* keyword block is as follows

DTENSOR

### region\_specifier !

**type** ! tensor type

**dispersivities** ! conditional on **type**

: ! repeat above three lines as many times as needed END ! terminates the keyword block

The **region\_specifier** comes in two forms. If three integers (**min max stride)** are given, then the region comprises the nodes between **min** and **max** (inclusive) with the specified stride. The value 0 for **max** is interpreted as the last node in the grid. This format is identical to FEHM’s format for specifying nodes in a region. If **region\_specifier** starts with a nonnumeric character, then it is interpreted as a filename that contains zone information. In this case, the first field of the **region\_specifier** is the filename and the second field of the **region\_specifier** is the zone number in the specified zone file. The zone file should be in LaGrit’s format.

The **type** keyword is specifies the type of the dispersion tensor. In Version 1.0, the only allowed value is *BF* denoting the Burnett and Frind tensor. The dispersivity\_values required for the Burnett and Frind tensor are (in order) longitudinal dispersivity, horizontal transverse dispersivity, vertical transverse dispersivity, and molecular diffusion coefficient. The dispersivities have units of

m. The molecular diffusion coefficient has units of m2/day.

* + 1. Numerical control parameters

Several numerical control parameters may also be specified in the control file. Defaults exist for each parameter. The parameters take scalar values may come in any order. The form is

### parameter\_name parameter\_value

where **parameter\_name** is one of the following

*dtmax* – maximum allowed value of time step in days. Optional. Defaults to 1000

*dt0* – initial time step in days. Optional. Defaults to 0.01

*maxstretch* – maximum allowed time step relative to previous time step (dimensionless). Must be greater than 1. Optional. Defaults to 1.2

*maxsteps* – maximum number of time steps in the simulation. A particle is terminated if maxstep timesteps are taken. Optional defaults to 100000

*dxtarget* – Courant factor, maximum allowed time step relative to time required to advect across the cell. Optional. Defaults to 0.1

*dttarget* – maximum allowed time step as fraction of characteristic time to disperse across cell. Optional. Defaults to 0.1

*toutfreq* – number of time steps between output of trajectory data. If 0 is specified, no trajectory output is written. Optional. Defaults to 0

* + 1. Example **control** file

Examples of control files are given in Figures A-x and A-y. In the example in Figure A-x, 10000 particles are released randomly in the region between (10,-50,-50) and (10,50,50). The dispersion coefficient in this example is

Title – the title goes here

dtmax 100 !maximum step size days dxtarget 0.1 !relative to grid size

dttarget 0.1 ! relative to characteristic dispersion time maxstretch 1.3

maxsteps 100000 ! maximum number of steps allowed dt0 0.1 ! initial step days

toutfreq 0 !no trajectory output

INITIAL RANDOM 10000

10.0 -50.0 -50.0

10.0 50.0 50.0

DTENSOR 1 0 0

BF !burnett and frind tensor

40. 0.0 0.0 0.0 !dispersivity values END

Figure A-1. Example **control** file.

Title Goes Here

dtmax 100 !maximum step size days dxtarget 0.1 !relative to grid size

dttarget 0.1 ! relative to characteristic dispersion time maxstretch 1.3

maxsteps 100000 ! maximum number of steps allowed dt0 0.1 ! initial step days

toutfreq 0 !no trajectory output

INITIAL RANDOM 10000

10.0 -50.0 -50.0

10.0 50.0 50.0

DTENSOR 1 0 0

BF !burnett and frind tensor

40. 0.0 0.0 0.0 !dispersivity values END

Figure A-2. Example **control** file.

### Files describing geometry of the mesh

Three files describing geometry of the computational grid are required, and a fourth is optional. It is anticipated that these files will be produced by the LaGrit software, although any grid generation software could be used provided the output is converted to the required format.

* + 1. The **fehmn** file

The **fehmn** file provides geometry information (location of nodes and lists of nodes that compose each element). See the LaGrit and FEHM manuals for details. In Walkabout Version 1.0, the **fehmn** file must be provided in ASCII format.

* + 1. The **stor** file

The **stor** file provides information about nodal connectivity and interface areas. See the LaGrit and FEHM manuals for details. LaGrit options exist to produce vector areas, scalar areas, or ratio of scalar area to distance for each node-to- node connection. The latter option is required by Walkabout. In Walkabout Version 1.0, the **stor** file must be provided in ASCII format.

* + 1. The **ealist** file

The **ealist** file provides information about element adjacency, which should not be confused with nodal connectivity. Element adjacency is not needed by FEHM,

but is required by Walkabout. See the LaGrit manual for details on how to produce the element adjacency lists.

* + 1. The **cbound** file

The **cbound** (closed boundary) file provides a list of nodes on boundaries that are closed to transport. It has the same format as the LaGrit outside zone file, but excludes outflow boundaries. A simple strategy for producing the **cbound** file is to use LaGrit to produce a list of all outside nodes, and then remove those nodes associated with outflow boundaries.

In Version 1, external faces of cells on a boundary must aligned with the principal directions in the coordinate system. That is, boundary faces must be top, bottom, left, right, back or front. See the LaGrit manual. It is important to recognize, that this restriction only applies to cell faces on boundaries. Cell faces internal to the model have no such restrictions. A node/cell may have more than one no- transport boundary face associated with it, in which case it would appear more than once in the **cbound** file.

For nodes on a no-transport boundary, Walkabout first attempts to reconstruct the nodal velocity using the unconstrained algorithm, Eq. 5. If this procedure results in inflow into the domain at the boundary node, the reconstructed velocity is used as is. If the unconstrained procedure results in outflow on a no-transport boundary, then the velocity reconstruction is repeated using the constrained procedure Eq. 7 to enforce the no-flow condition on the cell’s boundary face.

Particles are not allowed to disperse across boundaries that are closed to transport.

### Files containing FEHM results

Two files containing required FEHM results are required.

Internodal liquid fluxes are read from the **fin** file, an FEHM restart file in ASCII format. If the liquid fluxes are not found in the fin file, Walkabout will terminate.

Porosity, liquid saturation and liquid density are read from an ASCII **avs** file produced by FEHM. The **avs** file is produced by the FEHM contour macro. See the FEHM manual for details.

### Output files

Two output files are always produced, a log file that echoes back input parameters and a reduced SPTR2 file for PLUMECALC. The latter may be read directly by PLUMECALC.

In addition, to the two files always produced, a trajectory file (**trajout**) may be produced if the *toutfreq* parameter is set greater than 0 (see section A.1). The **trajout** file has format

!header line

!header line

**npart** !number particles

**ntimes** !number time steps reported for this particle

**t1 x1 y1 z1** ! spatial position at time **t1**

: ! repeat above line for a total of **ntimes**

: ! repeat each particle block for a total for **npart** particles