ILLINOIS ROCSTAR

GridConversion Users Guide Version 0.1.0

September 1, 2016

www.illinoisrocstar.com

License

GridConversion sources, executables, and this document are the property of Illinois Rocstar LLC. Licensing and support of the software package, including full source access for government, industrial, and academic partners, are arranged on an individual basis. Please contact Illinois Rocstar at

- **tech@illinoisrocstar.com**
- **sales@illinoisrocstar.com**

for support and licensing.

Contents

1 Introduction

GridConversion is a tool that aids in creating input meshes for *Rocfrac*, which is a component of the multiphysics software *Rocstar*. Currently, *GridConversion* takes a mesh input in Stanford format along with a brief text file describing the boundary conditions. Then, an output mesh is given in the correct format for use with *Rocfrac*. The format is essentially *Patran neutral* format, except all lines that are unused by *Rocfrac* are given dummy values. The final goal for the *GridConversion* tool is to become a utility that can take in a variety of mesh formats and likewise output a variety of mesh formats.

2 Installation

2.1 Obtain Source

GridConversion is distributed as part of *Rocstar*, located with *Rocfrac*, for which it is used as a preprocessing mesh tool. *Rocstar* and *GridConversion* can be obtained by downloading or cloning the source code from the *GitHub* at https://github.com/IllinoisRocstar/Rocstar. The following command can be used on a *Linux* terminal with git to obtain the *Rocstar* source code:

```
> git clone https://github.com/IllinoisRocstar/Rocstar
```

Note that *GridConversion* is not built along with *Rocstar* since it is a separate tool, with different dependencies. Therefore, it is recommended that you build and compile *GridConversion* outside of the *Rocstar* build. After downloading *Rocstar*, copy the directory GridConversion from Rocstar/Rocfrac/utilites to desired location.

```
> cp -r $DOWNLOAD_LOCATION/Rocstar/Rocfrac/utilites/GridConversion $GRIDHOME
```

Note that `$GRIDHOME` will be used throughout this guide to refer to the path of the top level directory for the *GridConversion* source.

2.2 Obtain Dependency

GridConversion has one main dependency that is not found on most *Linux* environments. This dependency is *IMPACT*, Illinois Rocstar's Multiphysics Application Coupling Toolkist. *IMPACT* can also be found on *GitHub* at https://github.com/IllinoisRocstar/IMPACT. Please see the instructions provided with *IMPACT* to build and install it.

2.3 Build

GridConversion uses *CMake* for constructing a Makefile. *CMake* should be able to locate most required files and dependencies. In the case that it cannot find something you can configure *CMake* using command line options or the command `ccmake`, which provides an interface. For more information about using *CMake* please see https://cmake.org. Listed below are the general commands for compiling *GridConversion* using *CMake* in a *Linux* terminal. Words in all caps beginning with a `$`, like `$GRIDHOME`, are used to signify your specific paths to the file or directory indicated.

```
> cd $GRIDHOME
> mkdir build
> cd build
> cmake .. -DCMAKE_PREFIX_PATH=$IMPACT_INSTALL
> make
```

Optional:

```
> make install
```

3 Preprocessing

GridConversion is a conversion tool created to take mesh input in Stanford format as output by *Pointwise*, which is a commerical mesh generation tool. This user guide does not cover mesh generation or the use of *Pointwise* beyond general grid exporting instructions. For information about purchasing, installing, and utilizing *Pointwise* see http://www.pointwise.com/.

3.1 Mesh Exporting

Beginning with a 3D mesh in *Pointwise*, set the solver to Standford by doing the following:

- Select CAE from the top menu
- Select Solver
- Choose Stanford ADL/SU2
- Select OK

Then, it is recommended that you set the boundary conditions for your mesh to help you when generating the boundary condition (BC) input file later. For usage with *GridConversion* and the Stanford format, you should set a distinct boundary condition for every domain in the mesh. These are not the actual boundary conditions used in the simulation, but are simply names to help you identify the mesh domains later. To set the boundary conditions do the following:

- Select CAE from the top menu

- Select Set Boundary Conditions

A box should appear where you can interactively set the boundary conditions for your mesh. To do so select a domain in the viewing screen, then select the New button to generate a new boundary condition. You can then name this domain anything you wish. This name will be exported in the mesh file and referenced later. For more information on how to set boundary conditions see the *Pointwise* User Manual.

After setting the boundary conditions, select all components of the mesh. Then do the following to export the mesh:

- Select File from the top menu
- Select Export
- Select CAE
- Follow the export prompts.

You now should have the necessary grid file for input into *GridConversion*. The next preprocessing step is to generate a BC file to work with the mesh.

3.2 Boundary Condition File

The boundary condition file is a simple text file written by hand. The format of this file is to have one line for every domain and edge in the mesh on which you wish to enforce a boundary condition. The boundary conditions relate to those set in the *Rocfrac* input file. The meaning of the boundary conditions will be explained briefly here in the context of a fluid-structure interaction problem. For more information about the *Rocfrac* boundary conditions please see the documentation provided with *Rocfrac*.

An example boundary condition file is shown below. This file will be referenced in order to explain the creation of a BC file for *GridConversion*.

```
domain 1  2   8 1 0 1 1 0 2 10   6 1
domain 2  1   8 1 1 0 2 3 0 3
domain 3  1   8 1 1 1 1 2 3 4
domain 5  1   8 1 1 0 1 1 0 6
domain 6  2   6 2   8 1 0 0 2 0 0 2
domain 7  2   8 0 0 1 0 0 3 8    6 1
domain 8  1   8 0 1 1 0 2 1 7
domain 4  2   8 1 0 1 1 0 2 5    6 7
domain 9  1   8 1 0 1 3 0 2 9
domain 10 1   6 30
```

```
edges 2
edge 2  9 4 1   8 1 1 0 2  2  0 1
edge 1  3 2 1   8 1 1 0 2  2  0 11
```

The BC file above shows one line for every domain in this particular mesh and one line for two of the twelve edges in the mesh. Note that only the entities that require boundary conditions need have a line in the file. The domains and edges can be in any order, but the edges MUST be listed AFTER a line containing the keyword “edges” followed by a space and the total number of edges with boundary conditions.

The content of a line containing domain information is as follows, with each component being separated by a space:

- The keyword “domain” is given.
- The number of the domain is given. This domain number corresponds to the order in which the domains are listed in the Stanford mesh file output from *Pointwse*. In order to determine the order of the domains, look at the end of your Stanford mesh file. Note that the word following MARKER_TAG in the file is the name of the boundary condition for the domain you provided in *Pointwise*.
- 1 or 2 is given, indicating the number of boundary conditions for the domain.
- An integer for the BC is given. This can be an 8 or a 6.

  For type 8, seven numbers follow, explained below.

  - Three integers 1 or 0. A 1 indicates that this boundary condition type is applied, and 0 indicates that it is NOT. The boundary condition types are structural, thermal, and mesh motion, in that order.
  - Three integers. These numbers indicate which boundary condition of the type should be applied. This number corresponds to the BC values in the *Rocfrac* input file.
  - One integer which signifies the priority of the boundary condition. A node may only have one type 8 boundary condition specified. Therefore, if a node lies on two domains with different type 8 BCs, one must take priority over the other. The BC with the lowest number in this last column takes priorty.

    For example, domain 1 in the BC file above has `8 1 0 1 1 0 2 10`. This indicates that domain 1 has a structural boundary condition with value 1 and a mesh motion boundary condition with value 2. In the *Rocfrac* input file there are boundary condition cards for each BC type. An example of these is shown below. A structural BC with value 1 indicates that this domain should have the BC defined in the `*BOUNDARY` section with a `1` as the first entry on the line. Similarly, a mesh motion BC with value 2 indicates that this domain should have the BC defined in the *BOUNDARYMM section with a 2 as the first entry on the line. For an explanation of the definitons of the BCs in the *Rocfrac* input file see the *Rocfrac* User Guide or Section **??**. Finally, the 10 at the end of the line for domain 1 indicates that this BC is priority 10 (i.e., lower priority than 1-9).

For type 6, one integer follows. This indicates the interaction type of the domain as follows

– 0: No FSI
– 1: FSI with burning
– 2: FSI without burning
– 10: No FSI, non-matching mesh

```
*BOUNDARY
4
1 0 0 0  0. 0. 0.
2 0 1 1  0. 0. 0.
3 1 1 0  0. 0. 0.
4 0 1 0  0. 0. 0.
*BOUNDARYMM
4
1 0 0 0  0. 0. 0.
2 0 1 1  0. 0. 0.
3 1 1 0  0. 0. 0.
4 0 1 0  0. 0. 0.
```

3.3 *Rocfrac* Boundary Conditions

For detailed information about how to specify boundary conditions in the *Rocfrac* input file please see the *Rocfrac* and *Rocstar* User Manuals. A brief description will be given here for enabling boundary conditions for an FSI problem without burning. Please read Section **??** first in order to better understand this section.

The mesh generated for *Rocfrac* should represent the structures part of the problem domain. First, make sure that the domains which will interact with the fluids have boundary type 6 values of 1 in the BC file discussed in Section **??**.

The next step is to determine what type 8 boundary conditions to enforce on the mesh domains. For an FSI problem without burning the boundary specifications relate to the displacement and movement of the mesh. In the legacy example cases, the FSI problems with *Rocfrac* use the exact same boundary conditions in the *BOUNDARY and *BOUNDARYMM sections. It is understood that keeping these the same is to enforce the proper mesh motion due to the force of the fluid displacing the solid. (Please note that *Rocfrac* is a legacy software, and the authors of this User Guide are doing their best to glean what information they can about how to properly use it.)

The `*BOUNDARY` and `*BOUNDARYMM` sections contain first a line with the total number of boundary conditions listedin the section. Then, there is one line for each boundary condition. These lines are formatted as follows:

- An integer identifying the boundary condition. These are mapped to from the mesh file as discussed in Section **??**.
- Three integers 1 or 0. These represent the x, y, and z axes. A 1 indicates that the mesh is free to move or be displaced in this coordinate direction. A 0 indicates that the mesh must be fixed here.
- Three "0."'s follow. The authors of this User Guide have not seen these values changed in any example FSI problems using *Rocfrac*, and they are unsure about their usage or meaning.

For example, third listing in the `*BOUNDARY` secion shows `3 1 1 0 0. 0. 0.` Therefore, all nodes in the mesh file with type 8 BC of value 3 will be allowed to move in the x and y directions but not the z.

4 Runtime

GridConversion is run from a *Linux* command line. It requires two input arguments in order: the name of the mesh file followed by the name of the BC input file. The output grid will either be written to the screen or to a file named after the command line flag "-o." See the example below:

```
> ./gridConversion mesh.input bcs.input -o output.grd
```

5 Example Cases

Due to budgeting and time constraints there are no current example cases. Additionally, the unit tests are currently broken.

6 Troubleshooting

If users encounter problems or have difficulties, please contact us at Illinois Rocstar.

specific_code@illinoisrocstar.com