.smesh File Format

Fei Zhu

06/10/2014

1 Introduction

We introduce the file format for simulation mesh in Physika, .smesh. "smesh" stands for "simulation mesh", it's not the ".smesh" format used by TetGen [Si06]. The .smesh format is heavily built on the .veg format developed by Jernej Barbič in VegaFEM [BSS12]. Unlike VegaFEM, we do not include material properties in simulation mesh files and .smesh files also support simulation mesh in 2D.

2 .smesh File Format

Basics: The smesh file format is ASCII. Lines starting with * denote a command. Lines starting with # are comments. Empty lines are ignored. Files can be nested using the *INCLUDE command. The effect of *INCLUDE is to include, at that point in the .smesh file, the entire contents of the included file. Include files can include other files; they can nest arbitrarily.

Vertices: Vertices are specified using the *VERTICES keyword. The second line gives the number of vertices, followed by integer "2" (two-dimensional simulation) or "3" (three-dimensional simulation), optionally followed by more parameters, which are ignored. The subsequent lines give one vertex per line, in the format:

<vertex index> <x> <y> [z]

where the vertex index starts either at 0 or 1, increments by 1 for every vertex. The unit for vertex positions is meters.

2D example:

```
*VERTICES
5 2 0 0
1 0.5 0.5
2 1.0 -0.5
3 -1.0 0.0
4 0.25 -0.25
5 0.6 0.2
```

3D example:

```
*VERTICES
5 3 0 0
1 0.5 0.5 0.5
2 1.0 -0.5 0.5
3 -1.0 0.0 1.0
4 0.25 -0.25 0.5
```

```
5 0.6 0.2 0.3
```

Elements: Elements are specified using the *ELEMENTS keyword. The second line gives the element type, namely “TRI”, “QUAD”, “TET” or “CUBIC”. “TET” and “CUBIC” stand for tetrahedral and cubic mesh for 3D simulation, “TRI” and “QUAD” are their counterparts in 2D. The third line gives the number of mesh elements, followed by the number of vertices in each element (3 for triangles, 4 for quads and tets, 8 for cubes), followed by some optional integers that are ignored. Subsequent lines give one element per line, in the format:

```
<element index> <vertex 1> ··· <vertex n>
```

where the element index starts either at 0 or 1 and increments by 1 for every element. Example:

```
*ELEMENTS
TETS
2 4 0
1 2 3 4 1
2 5 3 4 1
```

specifies a mesh with two tets. The first tet has vertices 2, 3, 4, 1 (in that order), and the second tet has vertices 5, 3, 4, 1. The ordering of vertices within the element matters. Incorrect order give wrong results.

Sets: Sets store a set of integer indices. They are used to store indices of elements that share the same property, for example material property. Sets are specified using the *SET keyword, followed by the name of the set. The next lines are comma-separated list of set elements. Elements should be stored and identified by the indices given in the *ELEMENTS section, i.e., either 0-indexed or 1-indexed, depending on whether *ELEMENTS are given 0-indexed or 1-indexed. Example:

```
*SET set1
11,17,21,37,113,310,555,
556,557,570,601,991,1013,1210,1225
```

Easy re-use of Stellar/TetGen meshes: The .smesh file format supports easy re-use of Stellar/TetGen meshes since it’s derived from .veg file. Suppose that the mesh vertices and elements are stored in myMesh.node and myMesh.ele files, in the standard Stellar/TetGen format. The following “template” .smesh file is the shortest way to import those meshes into a .smesh file:

```
*VERTICES
*INCLUDE myMesh.node
*ELEMENTS
TETS
*INCLUDE myMesh.ele
```

References

[BSS12] Barbič J., Sin F. S., Schroeder D.: Vega FEM Library, 2012. http://www.jernejbarbic.com/vega.

[Si06] Si H.: *TetGen A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. 2006.