TetGen Analysis Interface Module (AIM)

Ryan Durscher
AFRL/RQVC

January 23, 2020

Contents

1 Introduction

1.1 TetGen AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact with the open-source volume mesh generator, TetGen [1]. TetGen is capable of generating exact constrained Delaunay tetrahedralizations, boundary conforming Delaunay meshes, and Voronoi partitions.

An outline of the AIM's inputs and outputs are provided in AIM Inputs and AIM Outputs, respectively.

The accepted and expected geometric representation and analysis intentions are detailed in geomRepIntentTetGen.

Details of the AIM's shareable data structures are outlined in AIM Shareable Data if connecting this AIM to other AIMs in a parent-child like manner.

Current issues include:

- The holes or seed points provided to TetGen by creating an taking the cnetroid of a tetrahedron from an 'empty' mesh. This is guaranteed to work with solid bodies, but sheet bodies with multiple segregated regions where some regions are holes require manual seed points to indicate the hole.
- (**Important**) If Tetgen is allowed to added Steiner points (see "Preserve_Surf_Mesh" in AIM Inputs) discrete data transfer will **NOT** be possible.

1.2 TetGen Interface

In order to use TetGen, CAPS will automatically build the TetGen source code supplied in "library" mode. The directory in which the source code exists is set in the ESP configuration script. The C++ API is interfaced within the AIM through an interface function that takes the body tessellation and transfers the data to a "tetgenio" object in PLCs format (Piecewise Linear Complexes). After volume meshing is complete the mesh can be output in various mesh formats (see AIM Inputs for additional details).

2 AIM Inputs

The following list outlines the TetGen meshing options along with their default value available through the AIM interface.

- **Proj_Name = NULL**
  This corresponds to the output name of the mesh. If left NULL, the mesh is not written to a file.
- **Tess_Params = [0.025, 0.001, 15.0]**
  Body tessellation parameters. Tess_Params[0] and Tess_Params[1] get scaled by the bounding box of the body. (From the EGADS manual) A set of 3 parameters that drive the EDGE discretization and the FACE triangulation. The first is the maximum length of an EDGE segment or triangle side (in physical space). A zero is flag that allows for any length. The second is a curvature-based value that looks locally at the deviation between the centroid of the discrete object and the underlying geometry. Any deviation larger than the input value will cause the tessellation to be enhanced in those regions. The third is the maximum interior dihedral angle (in degrees) between triangle facets (or Edge segment tangents for a WIREBODY tessellation), note that a zero ignores this phase
- **Preserve_Surf_Mesh = True**
  Tells TetGen to preserve the surface mesh provided (i.e. do not add Steiner points on the surface). Discrete data transfer will **NOT** be possible if Steiner points are added.
- **Mesh_Verbose_Flag = False**
  Verbose output from TetGen.
- **Mesh_Quiet_Flag = False**
  Complete suppression of all TetGen output messages (not including errors).
- **Quality_Rad_Edge = 1.5**
  TetGen maximum radius-edge ratio.
- **Quality_Angle = 0.0**
  TetGen minimum dihedral angle (in degrees).
- **Mesh_Format = "AFLR3"**
  Mesh output format. Available format names include: "AFLR3", "TECPLOT", "SU2", "VTK", and "NASTRAN".
- **Mesh_ASCII_Flag = True**
  Output mesh in ASCII format, otherwise write a binary file if applicable.
- **Mesh_Gen_Input_String = NULL**
  Meshing program command line string (as if called in bash mode). Use this to specify more complicated options/use features of the mesher not currently exposed through other AIM input variables. Note that this is the exact string that will be provided to the volume mesher; no modifications will be made. If left NULL an input string will be created based on default values of the relevant AIM input variables. See TetGen Command Line Inputs for options to include in the input string.
- **Ignore_Surface_Mesh_Extraction = True**
  If TetGen doesn't preserve the surface mesh provided (i.e. Steiner points are added) a simple search algorithm may be used to reconstruct a separate (not dependent on the volume mesh node numbering) representation of the surface mesh. In general, this has little use and can add a significant computational penalty. The default value of "True" is recommended.
- **Mesh_Tolerance = 1E-16**
  Sets the tolerance for coplanar test in TetGen.
- **Multiple_Mesh = False**
  If set to True a volume will be generated for each body. When set to False (default value) only a single volume mesh will be created.
- **Mesh_Sizing = NULL**
  See Mesh Sizing for additional details.

- **Regions = NULL**
  If this input is set, the volume mesh will be divided into regions, each bounded by surface mesh and identified by an interior `seed` point. If a seed appears in the `Regions` input, then its region will be meshed and the markers for all cells in the region will be set to the region's `id`. The markers for cells that fall outside of any user-defined region or hole will be numbered automatically. The input is a vector of tuples. The tuple keys are ignored and the tuple values are dictionaries; each requires an integer `id` entry and a 3-vector `seed` point. For example, from within a pyCAPS script,

```
tetgen.setAnalysisVal('Regions', [
  ( 'A', { 'id': 10, 'seed': [0, 0,  1] } ),
  ( 'B', { 'id': 20, 'seed': [0, 0, -1] } )
])
```

Automatic hole detection will be disabled if one or both of the `Regions` and `Holes` inputs is not NULL.

- **Holes = NULL**
  If this input is set, the volume mesh will be divided into regions, each bounded by surface mesh and identified by an interior `seed` point. If a seed appears in the `Holes` input, then its region will not be meshed. The input is a vector of tuples. The tuple keys are ignored and the tuple values are dictionaries; each requires a 3-vector `seed` point. For example, from within a pyCAPS script,

```
tetgen.setAnalysisVal('Holes', [
  ( 'A', { 'seed': [ 1, 0, 0] } ),
  ( 'B', { 'seed': [-1, 0, 0] } )
])
```

Automatic hole detection will be disabled if one or both of the `Regions` and `Holes` inputs is not NULL.

3 aimInputsAFLR4

- **Edge_Point_Min = 2**
  Minimum number of points along an edge to use when creating a surface mesh.

- **Edge_Point_Max = NULL**
  Maximum number of points along an edge to use when creating a surface mesh.

4 AIM Shareable Data

The TetGen AIM has the following shareable data types/values with its children AIMs if they are so inclined.

- **Volume_Mesh**
  The returned volume mesh after TetGen execution is complete in meshStruct (see meshTypes.h) format.

- **Attribute_Map**
  An index mapping between capsGroups found on the geometry in mapAttrToIndexStruct (see miscTypes.h) format.

5 AIM Outputs

The following list outlines the TetGen AIM outputs available through the AIM interface.

- **Done** = True if a volume mesh(es) was created, False if not.

6 Mesh Sizing

NOTE: Available mesh sizing parameters differ between mesh generators.

Structure for the mesh sizing tuple = ("CAPS Group Name", "Value"). "CAPS Group Name" defines the capsGroup on which the sizing information should be applied. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword string (see Section Single Value String)

6.1 JSON String Dictionary

If "Value" is a JSON string dictionary (e.g. "Value" = {"edgeDistribution": "Even", "numEdgePoints": 100}) the following keywords (= default values) may be used:

- **edgeDistribution = "Even"**
  Edge Distribution types. Options: Even (even distribution), Tanh (hyperbolic tangent distribution).

- **numEdgePoints = 0**
  Number of points along an edge.

- **initialNodeSpacing = [0.0, 0.0]**
  Initial (scaled) node spacing along an edge. [first node, last node] consistent with the orientation of the edge.

- **tessParams = (no default)**
  Face tessellation parameters, example [0.1, 0.01, 20.0]. (From the EGADS manual) A set of 3 parameters that drive the EDGE discretization and the FACE triangulation. The first is the maximum length of an EDGE segment or triangle side (in physical space). A zero is flag that allows for any length. The second is a curvature-based value that looks locally at the deviation between the centroid of the discrete object and the underlying geometry. Any deviation larger than the input value will cause the tessellation to be enhanced in those regions. The third is the maximum interior dihedral angle (in degrees) between triangle facets (or Edge segment tangents for a WIREBODY tessellation), note that a zero ignores this phase.

6.2 Single Value String

If "Value" is a single string, the following options maybe used:

- (NONE Currently)

7 TetGen Command Line Inputs

This is the output from executing "tetgen -h" on command line.

Parameter	Description
-p	Tetrahedralizes a piecewise linear complex (PLC).
-Y	Preserves the input surface mesh (does not modify it).
-r	Reconstructs a previously generated mesh.
-q	Refines mesh (to improve mesh quality).
-R	Mesh coarsening (to reduce the mesh elements).
-A	Assigns attributes to tetrahedra in different regions.
-a	Applies a maximum tetrahedron volume constraint.
-m	Applies a mesh sizing function.
-i	Inserts a list of additional points.

Parameter	Description
-O	Specifies the level of mesh optimization.
-S	Specifies maximum number of added points.
-T	Sets a tolerance for coplanar test (default 1e-8).
-X	Suppresses use of exact arithmetic.
-M	No merge of coplanar facets or very close vertices.
-w	Generates weighted Delaunay (regular) triangulation.
-c	Retains the convex hull of the PLC.
-d	Detects self-intersections of facets of the PLC.
-z	Numbers all output items starting from zero.
-f	Outputs all faces to .face file.
-e	Outputs all edges to .edge file.
-n	Outputs tetrahedra neighbors to .neigh file.
-v	Outputs Voronoi diagram to files.
-g	Outputs mesh to .mesh file for viewing by Medit.
-k	Outputs mesh to .vtk file for viewing by Paraview.
-J	No jettison of unused vertices from output .node file.
-B	Suppresses output of boundary information.
-N	Suppresses output of .node file.
-E	Suppresses output of .ele file.
-F	Suppresses output of .face and .edge file.
-I	Suppresses mesh iteration numbers.
-C	Checks the consistency of the final mesh.
-Q	Quiet: No terminal output except errors.
-V	Verbose: Detailed information, more terminal output.
-h	Help: A brief instruction for using TetGen.

Examples of TetGen input strings:

Input String	Description
-pq1.414a.1	generates a mesh whose tetrahedra have radius-edge ratio smaller than 1.414 and have volume of 0.1 or less

References

[1] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2):11:1–11:36, Feb. 2015. 1