

How to use pentagrow

Overview

pentagrow is a command-line interface to the hybrid mesh generation features in libsurf. This interface allows to generate hybrid pentahedral/tetrahedral meshes around existing surfaces meshes. The program is controlled by means of a configuration file where some execution options can be set. The final mesh can be viewed (and converted to other formats) using the program dwfscope.

The program is called as in

```
pentagrow wallmesh.cgns config.cfg
```

where “wallmesh” is the surface mesh, no including any farfield boundaries, in one of the supported file formats, and “config” is a plain text file containing program settings.

Internally, pentagrow makes use of the exact same functions also called by sumo when starting a hybrid mesh generation pass on a surface mesh in the graphical user interface. The difference is that pentagrow can be used to generate a hybrid volume mesh around a surface mesh which does not originate from sumo.

pentagrow operates in three phases:

1. Generate envelope: As a first step, the envelope surface of the prismatic layer is determined from the wall mesh and some of the parameters given in the configuration file. This step usually completes in a few seconds.
2. Tetrahedral region: Envelope and farfield boundaries are passed to the external program tetgen, which generates the tetrahedral mesh between these two surfaces and removes elements from any marked cavities. In most cases, this step is the most time consuming and may take as long as 20 minutes for envelope meshes of a million triangles. Note that the combination of a surface mesh of extremely low quality and very high tet quality requirements can result in excessively long refinement times, because tetgen is forced to repeatedly split envelope surface triangles in order to achieve the desired tet shape quality measure.
3. Prismatic layer: Once the tetrahedral region has been filled with elements, the envelope surface and wall mesh is adapted to the inner boundary of the tet mesh and pentahedra are created starting from the wall. Although this step usually results in most of the mesh nodes, it tends to complete rather more quickly than the more complicated tetgen stage.

Output files

The final hybrid mesh is stored in the file ‘hybrid.bmsh’ (or hybrid.cgns when that format is selected) and the file penta.zml for the native format. Moreover, some additional files are written to aid in mesh quality checking. One file, named outermost.zml, contains the prismatic layer envelope mesh along with distance and normal fields. If necessary, a file named ‘diagnose.txt’ contains the locations of irregular pentahedral elements. This file also documents whether these elements are just geometrically unsound or whether they will actually fail to pass the EDGE preprocessor. In many cases, bad pentahedral elements which pass the preprocessor will only affect the quality of the flow solution near the bad elements, but not lead to failure of the solver. However, for best solver reliability, it is recommended to avoid tangled pentahedral elements by improving surface mesh quality sufficiently.

Configuration file

Lines starting with a hash character (#) are treated as comments and have no influence on the program. Hence, settings can be quickly disabled by prefixing them with a hash sign.

Global settings

The first group of settings concern the global mesh topology and the automatic creation of farfield boundaries.

FarfieldRadius: Radius of the spherical farfield boundary.

FarfieldCenter: Center point (x y z) of the farfield boundary.

TetgenPath: This optional value can be used to give the location of the tetgen executable, which can be useful when multiple different versions are installed.

TetgenOptions: When set, this options string is passed to the external program tetgen.

HolePosition: In most cases, the surface mesh passed to pentagrow only contains a single cavity. The point passed (x y z) as the value here is used to start the process of erasing tetrahedral elements so as to create the cavity. When more than one disjoint cavity is present, for instance with deployed flaps or eternal store bodies, multiple hole points can be specified by concatenating their coordinates.

Second tetgen pass

When using tetgen version 1.5 (or later?), the quality of the tetrahedral region mesh can be improved by performing a second mesh refinement pass on the

volume mesh. This second tetgen run is initiated automatically when the growth factor option below is set to a non-zero value.

TetGrowthFactor: Desired growth factor between edge lengths of coincident tetrahedra, which the second refinement pass attempts to achieve. Quite often, a value above 1.6 has very little effect. For subsonic or benign transonic cases, a value of 1.4 is recommended for an initial mesh. When high shock resolution is desired or the nearfield region should be relatively fine to resolve suspected vortices, this value may be chosen between 1.1 and 1.3 instead. (Default value is 0.0, which disables the second pass.)

TetEdgeSmoothing: Number of smoothing iterations which diffuse the desired edge length distribution throughout the volume. The larger the number of iterations, the more widely the edge growth rate criterion is spread. With a small number of iterations, say 32 or less, element edge lengths are equalized only in a fairly close range. In this way, it is possible to achieve a low edge length growth ratio (perhaps 1.2) in the close vicinity of fine surface mesh resolution only. With a very large value, e.g. 256 or more, then the growth rate criterion is enforced globally. (Default value is 128.)

Nearfield refinement

The following options allow to define an ellipsoidal region enclosing the surface mesh, in which a certain constant volume constraint for tetrahedral elements can be enforced. In general, it is recommended to use the nearfield refinement by means of a second tetgen pass (see above) instead of this possibility.

NearfieldEdgeLength: Desired maximum length of tetrahedron edges in the refined near-field region. Set this value to 0.0 in order to disable near-field refinement.

NearfieldSize: A scalar (default 3.5) which determines the size of an ellipsoidal refinement region around the surface mesh in terms of the wall surface bounding box size. The tet mesh inside this ellipsoidal is refined such that the tet edge length is below 'NearfieldEdgeLength'. Setting this option is easier than specifying the nearfield radii explicitly.

NearfieldCenter: Optional setting which can be used to shift the center point of the refinement ellipsoidal. Default is to use the wall surface bounding box center.

NearfieldSemiAxes: Optional explicit specification of semi-axes of the nearfield refinement volume. This option can be used to control the size of the refinement region in a very detailed manner. pentagrow will terminate with an error message if the ellipsoidal resulting from NearfieldCenter and NearfieldSemiAxes does not fully enclose the wall surfaces.

Basic settings for the prismatic layer

The following settings control properties of the prismatic layer and are most likely necessary to define when a new case is handled.

InputFormat: is a hint used to determine the format of the input surface mesh. It can take the values msh, stl, cgns or zml. For large meshes, CGNS or the native zml (LZ4-compressed hierarchical hybrid binary/XML) is recommended.

NLayers: Number of prismatic element layers.

InitialHeight: is the height of the first prismatic cell, touching the wall, in mesh length units. This value will usually need to be of the order of a few micrometers in order to ensure sufficient resolution of boundary layer features.

OutputFormat: The format of the hybrid mesh output. Any combination of bmesh, cgns and zml. Other formats can be written by importing the mesh into scope.

Advanced settings

A few options described here can be employed to control certain aspects of the mesh generation procedure. Usually, it is not necessary to change these settings.

FeatureAngle: Whenever the dihedral angle of two triangle is smaller than this limit, the resulting edge is understood to represent an actual geometrical feature. Larger angles are treated as resulting from approximation of curved surfaces by linear triangles. (Default value is 20 deg).

MaxGrowthRatio: is the largest allowed ratio between the wall-normal edge lengths of consecutive cells - default is 1.5; note, however, that the average growth ratio in most cases is very significantly less than this.

MaxLayerThickness: is the maximum allowed absolute thickness of the prismatic layer. This is an upper limit; the actual layer thickness will be adapted locally but never exceed this value.

MaxRelativeHeight: is the upper limit of the prismatic layer height in terms of local triangle edge length. This value can usually be set relatively high, say above 5.

SplineNormals: is named a little inaccurately; when set to ‘true’, pentahedral element edges are enforced to be exactly normal to the wall for the first element and are then gradually aligned to the envelope normal. Default is ‘false’, since this option puts slightly higher demands on surface mesh quality to avoid tangled pentahedra.

HeightIterations: Minimum number of local envelope height smoothing iterations.

NormalIterations: Minimum number of local growth direction smoothing iterations.

MaxCritIterations: Maximum permitted number of iterations when resolving geometric collisions. Do not reduce this number unless absolutely necessary.

LaplaceIterations: Minimum number of global envelope smoothing iterations.

Calling tetgen

As one step of the processing sequence, the external program [tetgen](#), available for use under the conditions of the AGPL, is called from within pentagrow. Normally, pentagrow is packaged with a version of tetgen in the same directory.

Details of how tetgen generates the tetrahedral mesh filling the space between the prismatic layer envelope and the farfield boundary can be controlled by setting the value

```
TetgenOptions = -pq1.2Ya
```

in the configuration file. The detailed meaning of these flags can be found in the [tetgen documentation](#). One important setting is the ‘Y’ flag, which controls whether tetgen is forbidden to split triangles on the input surfaces. When set, tet quality cannot be improved when nearby surface triangles prevent it; in this case, tetgen often finishes much faster and generates fewer elements. Without the ‘Y’ option, the program will enforce the tet quality (1.2 in this case) by all means, including subdividing surface triangles. In the rather common case of surface meshes which do not themselves comply with the radius-to-edge criterion usually specified, this will result in a possibly very large number of additional surface triangles (and hence volume elements). Tetgen version 1.5 is much more aggressive in this regard than version 1.4 was.

When using the nearfield refinement option of pentagrow, then the flag ‘a’ must be added to the tetgen call as shown in the example above. This ensures that tetgen will use the regional element size distribution file created by pentagrow.

Visualization

When visualizing very large meshes with scope, it should be noted that the graphical display requires much more data than what is stored to file. Therefore, scope needs roughly 2 Gigabytes of memory for every 10 million elements in the mesh. This may severely limit the size of meshes which can be viewed on low-memory machines. Very large meshes may hence be more suitable for inspection with paraview or other software which allows to selectively load parts of the mesh (for instance, only surfaces).

Example Configuration

```
InputFormat = msh
NLayers = 32
FeatureAngle = 31.0
InitialHeight = 2e-6
MaxLayerThickness = 0.5

FarfieldRadius = 120.0
# three cavities marked
HolePosition = 5.0 0.0 0.0 8.0 1.3 0.46 8.0 -1.3 0.46
TetgenOptions = -pq1.2VY
TetGrowthFactor = 1.35

HeightIterations = 8
NormalIterations = 8
MaxCritIterations = 128
LaplaceIterations = 8
```