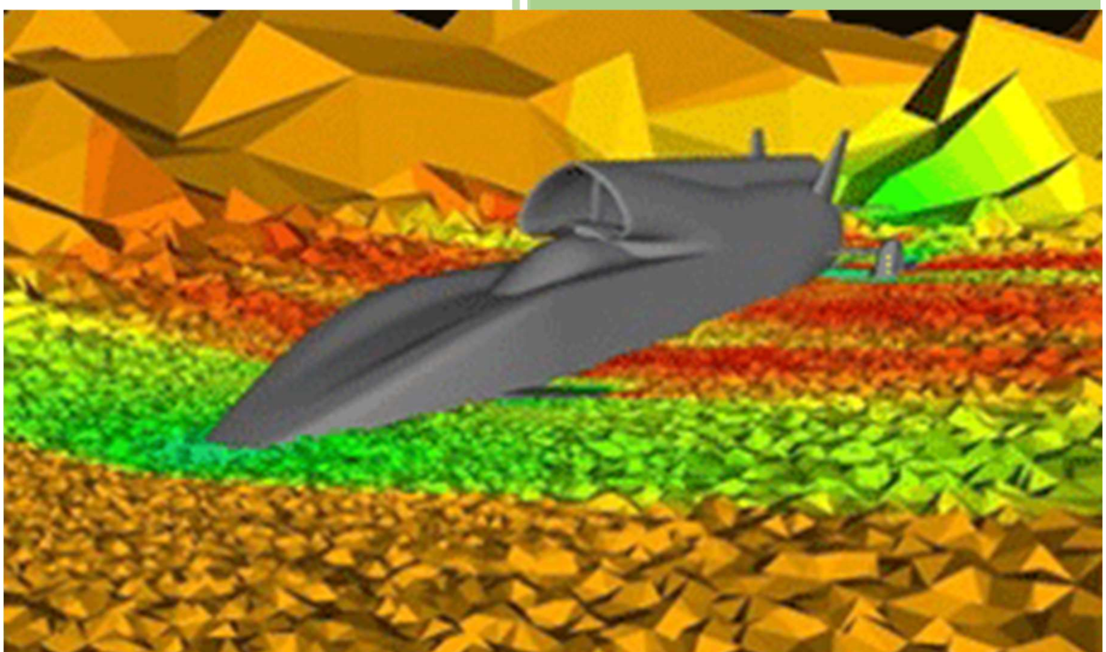
SwanSim FLITE Mesh Generators User Guide



Dr Jason W Jones

College of Engineering, Swansea University

May 2017

Contents

1 Introduction

This document is part of a series of short guides available on the SwanSim web site (http://www.swansim.org).

2 Obtaining the Software

There are three ways to obtain the software:

- Download the pre-compiled binaries
- Download a zip file of the source code
- Obtain the source code by cloning the Git repository

2.1 Pre-compiled binaries

On the Windows platform, there are two options – either you can download a zip file and extract it manually or download the installer program and run it.

For Linux platforms, there is only one choice – downloading the zip file and extracting it manually.

If this method is chosen then the next section describing how to build the software can be skipped.

2.2 Source Code

For either platform, you may download the source code as a zip file and extract it. If you choose this method then you must build the software before it can be used. This is covered in the next section.

2.3 Cloning the Git repository

This method of obtaining the source code relies on a small knowledge of the Git source code control system. There are graphical tools that can be used to interface with the Git repository as well as the basic command line tools. For this guide, we will be using the command line tools.

```
git clone git@git.swansim.org:swansim-public/flite-mesh-generators.git
git checkout master
```

This creates a new folder 'flite-mesh-generators' containing the source code. For the rest of this document this will be referred to as '*flite-mesh-generators*'.

3 Building the Software

The software is built using the CMake system (http://cmake.org). This must be downloaded and installed before building can commence.

CMake works by performing an out-of-source build. This means that the folder in which the object files and executables are created is separate from the folder containing the source code. This saves polluting the source code with any artifacts from the build process.

The FLITE software requires the OpenCascade CAD libraries in order to be built. These are available here. The correct zip file for your platform must be downloaded and extracted (the location is up to the user). For the rest of this guide, the top-level folder will be referred to as *'oce-dir'*.

In order to build the software, two folders must be created:

- The folder in which the software will be built (for now we will call this *'flite-build'*).
- The folder in which the software will be installed (for now we will call this *'flite-install'*).

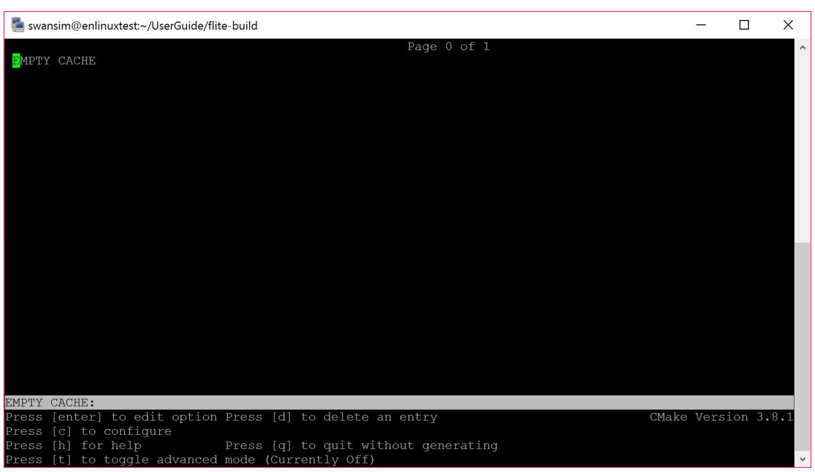
The locations of these folders are completely up to the user.

3.1 CMake on Linux

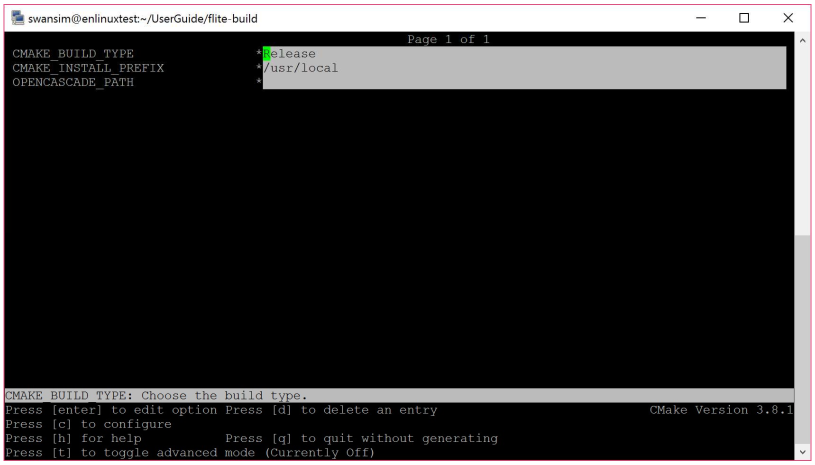
In order to build the software in Linux, run these commands:

```
cd flite-build
ccmake flite-mesh-generators
```

This displays a textual interface like the one below:



Press 'c' to start the configuration process. This then adds three variables that need to be checked/edited by the user – as shown below.



Use the cursor keys to move down to the CMAKE_INSTALL_PREFIX variable and press *RETURN* to change the value to the location of the '*flite-install*' folder. Press *RETURN* again to commit that change.

Repeat for the OPENCASCADE_PATH so that it contains the path to the *'oce-dir'* folder.

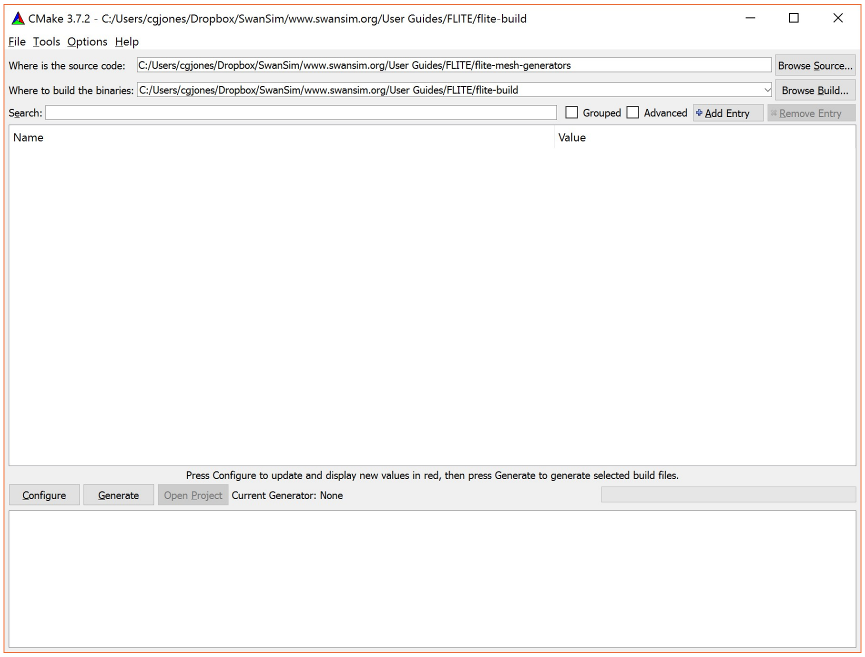
When this is done, press 'c' to re-configure then 'g' to generate the Makefiles. This exits the CMake tool.

Build the software using the following command:

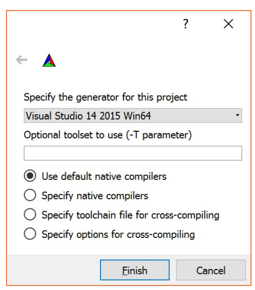
```
make
```

3.2 CMake on Windows

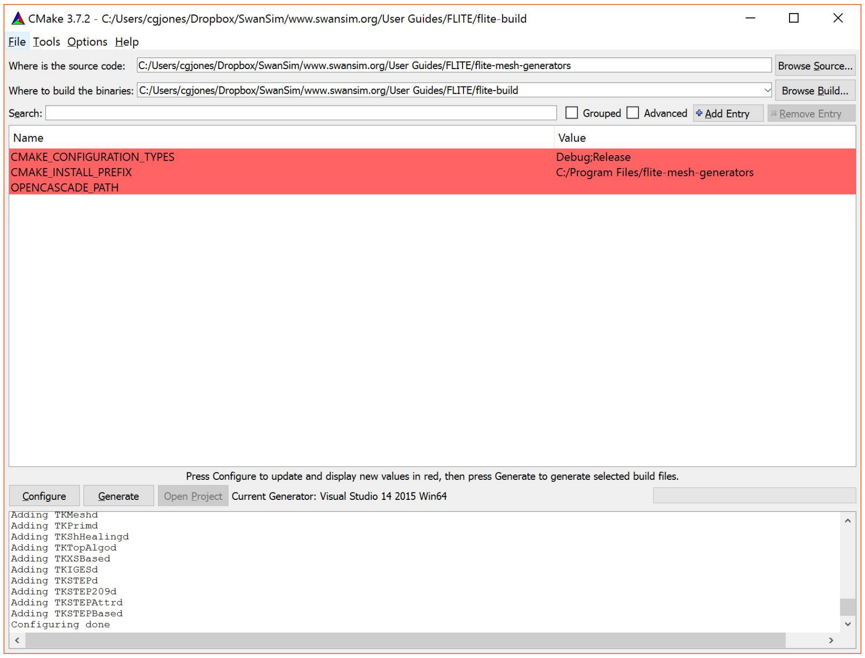
In order to build the software in Windows, run the 'CMake' desktop app. This will appear as below:



Enter the *'flite-mesh-generators'* folder into the Source code box and *'flite-build'* into the Build box – then press 'Configure to start the configuration. This opens a dialog in which the build tools are selected (in this case Visual Studio 14 2015 Win64 is chosen).

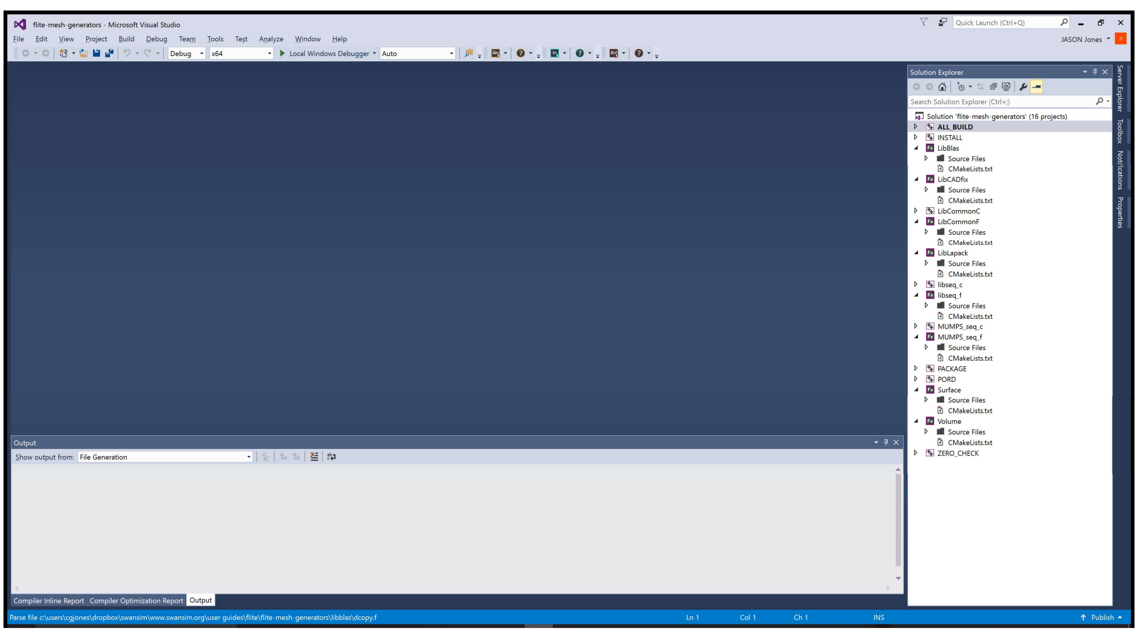


After this the configuration process starts and the user is asked for 3 variables.



Enter the *<u>'flite-install'</u>* folder into CMAKE_INSTALL_PREFIX and *<u>'oce-dir'</u>* folder into OPENCASCADE_PATH, then press 'Configure' again followed by 'Generate'.

This produces a Visual Studio project structure in the *<u>'flite-build'</u>* folder. This can be opened by double-clicking on the solution file (.sln).



The software can then be built by selecting 'Build'.

3.3 Installing the Software

Once built the software should be installed in the folder, '*<u>FLITE-Install</u>*', before being run. Installing ensures the correct executables and libraries are all copied to the correct folder structure.

Regardless of the platform, this is performed by building the target ‘INSTALL’.

In Visual Studio, right-clicking on this target and selecting Build.
In Linux, typing the command:

```
make install
```

4 Testing the Software

Within the '*flite-mesh-generators*' folder, there is a sub-folder called 'examples'. This contains a number of small test-cases that help to ensure the software has been built / installed correctly.

4.1 Running the Surface Mesher

In Windows, open a Command Prompt and go to the 'Examples' folder and enter the following command:

```
FLITE-Install\<Debug or Release>\bin\Surface.exe cube
```

In Linux, this can be achieved in a shell window by entering the following commands:

```
cd flite-mesh-generators/examples
FLITE-Install/Surface.exe cube
```

This will run the Surface Mesher to produce a triangulation on the 6 faces of the cube in the STEP file 'cube.stp'. If successful, a file 'cube.fro' will be created which contains the surface mesh (this is a text file and can be viewed in a text editor).

4.2 Running the Volume Mesher

In Windows, open a Command Prompt and go to the 'Examples' folder and enter the following command:

```
FLITE-Install\<Debug or Release>\bin\Volume.exe cube
```

In Linux, this can be achieved in a shell window by entering the following commands:

```
cd flite-mesh-generators/examples
FLITE-Install/Volume.exe cube
```

This will run the Volume Mesher to produce a volume mesh inside the cube. If successful, a file 'cube.plt' will be created which contains the volume mesh (this is a binary file that cannot be viewed in a text editor).

5 Surface Mesher User Guide

The purpose of this section is to provide more detail into how to run and control the operation of the Surface Mesher for your own models.

5.1 Input Files

The Surface Mesher reads a number of input files, most of which must differ in name only by their extension:

- **Geometry** – This can be either an IGES file (*.iges or *.igs), a STEP file (*.step or *.stp) or our internal geometry format (*.dat) (described in Section ###).
  Regardless of the input file format, the geometry must be water tight and manifold. This means that every intersection curve must join two faces and every face must be completely bounded by at least 2 curves. This last requirement means that if a face is bounded by a circle, for example, then that circle must be split into two curves.
- **Background Mesh** – This file controls the density of the mesh in user-defined regions (described in Section ###).
  This file is optional – if it does not exist then the options in the Control File will be used.
- **Control File** – This allows the user to define a number of global control parameters which alter the behaviour of the Surface Mesher. (This breaks the naming rule and must be named 'Surf3D_vx.ctl', where the 'x' is the version of the file (currently 25). This follows the Fortran 90 NAMELIST convention and is described later in this section.
  This file is optional – if it does not exist then the default options will be used.

5.2 Output File

Once the Surface Mesher has finished, a surface mesh is written out as a text file (*.fro). The format of this is described in Section ###.

5.3 Surface Mesher Control File

This section is devoted to describing the various control parameters for the Surface Mesher. A brief description of each of these can also be viewed by running the Surface Mesher with the '-hh' option.

Running the Surface Mesher with the '-o' option produces a default control file as shown below:

```
!--- please refer README for the control parameters reading...

 &CONTROLPARAMETERS
 DEBUG_DISPLAY       = 1,
 START_POINT         = 1,
 CURVATURE_TYPE      = 4,
 GENERATE_METHOD     = 1,
 ORIENTATION_DIRECT = 0
 /
 &BACKGROUNDPARAMETERS
 BACKGROUND_MODEL          = 5,
 GLOBE_GRIDSIZE            = 1000.00,
 CURVATURE_FACTORS         = 0.20, 0.10, 1.00, 10.00,
                             150.00, 120.00, 1.00,
 STRETCH_LIMIT             = 10.00,
 MAPPING_INTERP_MODEL      = 1,
 INTERPOLATE_OCT_MAPPING = F,
 GRADATION_FACTOR          = 0.50
 /
 &COSMETICPARAMETER
 LOOP_COSMETIC          = 1,
 LOOP_SMOOTH            = 1,
 SMOOTH_METHOD          = 3,
 COLLAPSE_ANGLE         = 12.50,
 SWAPPING_ANGLE         = 30.00,
 LOOP_SUPERCOSMETIC     = 3,
 SUPERCOSMETIC_METHOD = 0,
 SUPERCOSMETIC_QUAD     = -1,
 LOOP_SUPERFRQFILTER    = 0,
 SUPERFRQFILTER_POWER = 1.00
 /
```

5.3.1 DEBUG_DISPLAY

The debug display parameter determines how much information is output to the screen and how many internal checks are performed on the mesh. By default this is set to 1 and takes a value between 0 and 4. If you are having trouble meshing a geometry increasing this value is a good way to start diagnosing your problem.

5.3.2 START_POINT

If the start point is set to 1 then the mesher will read in a geometry definition and generate a new mesh from scratch. If the start point is set to 2 then the mesher reads in an existing mesh to modify. This second option may be useful if your original surface definition is a triangulation you wish to refine or smooth. For example some 3D scanning software will output .stl files in this way. Our mesher expects the input mesh to be in the .fro format and named [jobname]_0.fro, the final mesh will then be output as [jobname].fro.

5.3.3 CURVATURE_TYPE

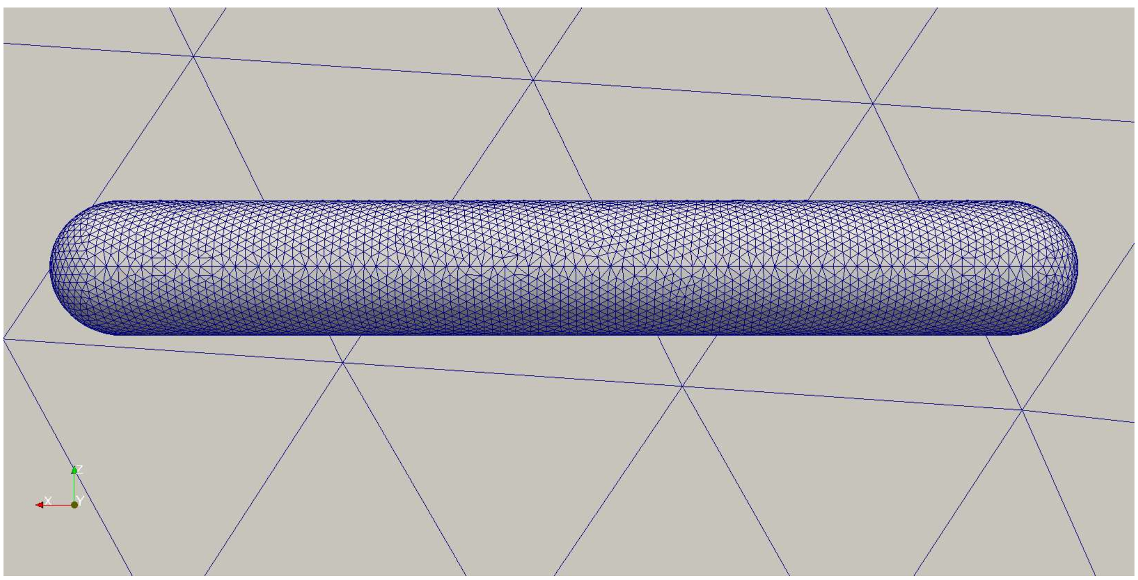
The curvature type parameter tells the mesher which geometry definition format you are using. There are four options 1-4. Setting the value to 1 will tell the mesher to read the geometry from the file [jobname].dat in the Swansea University file format. The default value, 4, makes the mesher read in either an IGES or STEP CAD file. When option 4 is selected the mesh will be generated on the exact CAD definition, this means the CAD definition must be water tight and topologically valid for mesh generation.

5.3.4 GENERATE_METHOD

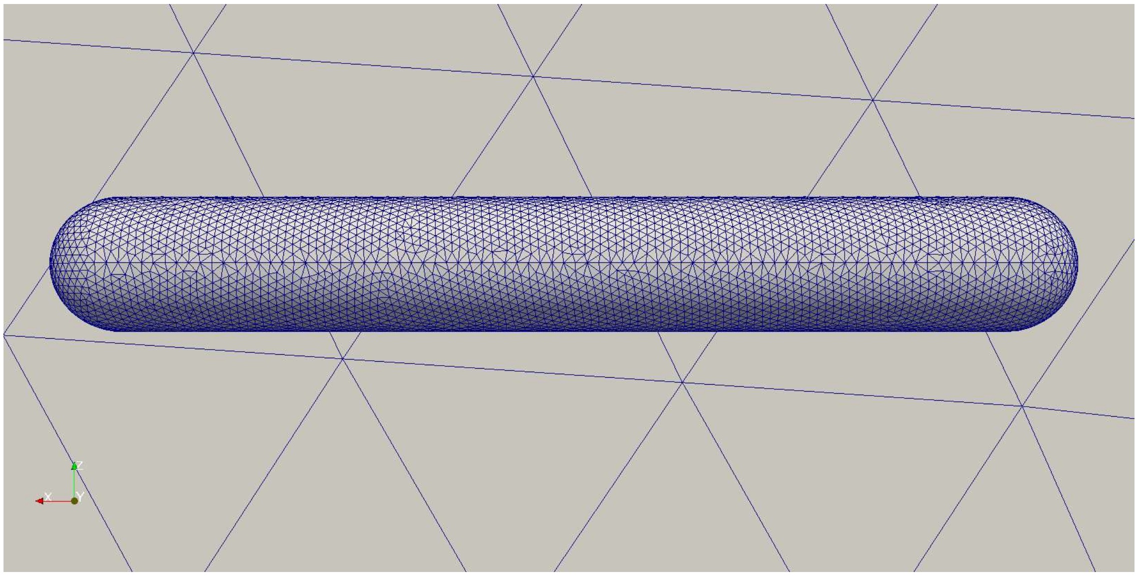
This parameter effects how the mesh is generated. There are three different techniques. Setting the parameter to 1 or 2 will generate the mesh using the advancing front technique. Method 1 is faster but less robust than 2. We suggest using 1 at first and only using 2 if mesh generation fails with 1. Setting the option to -1 generates the mesh using a Delaunay kernel and recursive element subdivision. This third method is the fastest and most fool proof, but the mesh quality will be lower. In the vast majority of cases you should use advancing front.

Examples:

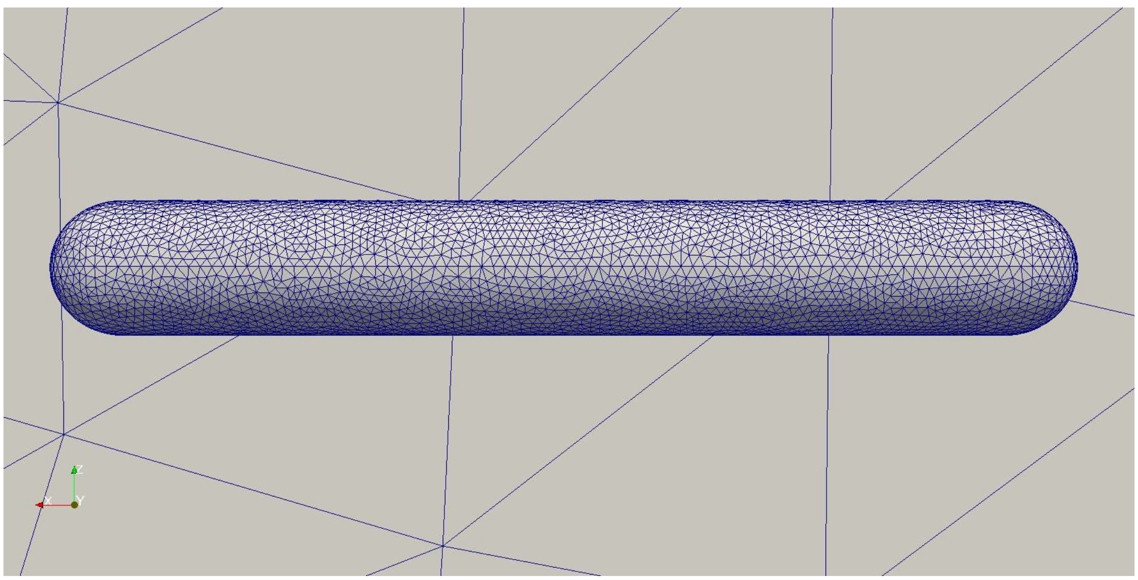
GENERATE_METHOD=1



GENERATE_METHOD=2



GENERATE_METHOD=-1



5.3.5 ORIENTATION_DIRECT

Ask OH

5.3.6 BACKGROUND_MODEL

This defines the method for determining element size as a function of location. We refer to this as the background spacing. Traditionally this is done by defining spacing values at the nodes of a number of large tetrahedra and interpolating these values. This is then refined by adding various sources, such as point and line sources. For complex geometries you may need a large number of sources, this can become computationally expensive as every time you interrogate the background spacing you need to check the distance to each source. For this reason we have also implemented an octree background spacing function which defines the spacing in the octants of an octree.

Positive values indicate you wish to generate an isotropic mesh and negative values indicate you wish to generate an anisotropic mesh. (More info here??)

Setting the background model to 2 allows the use of a traditional tetrahedral background spacing function. This spacing is input in the file [jobname].bac. Setting the background model to 4 reads in a .bac file the same as 2 but then generates an octree background function. When this option is selected the file [jobname].Obac is written which contains the generated octree.

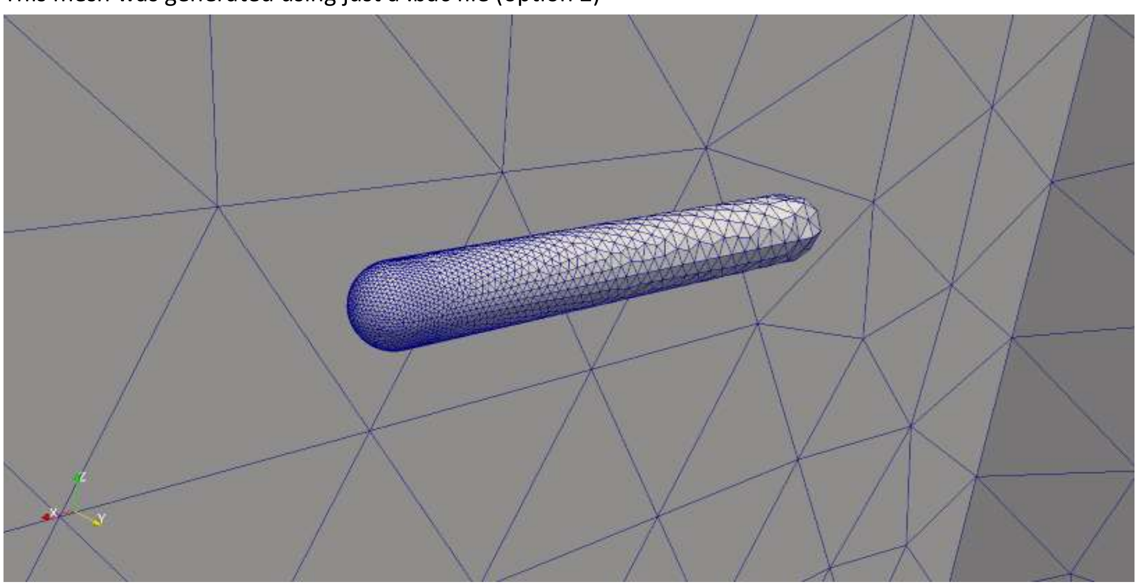
Option 5 is the same as 4 except the code also generates a curvature controlled background spacing function. This spacing function is dependent on the curvature of the geometry, this means features like trailing edges will be automatically refined. The magnitude of this refinement is controlled using CURVATURE_FACTORS. If the file [jobname].bac exists then when interrogating the spacing the minimum of the supplied function and the generated function is taken as the target spacing. If no .bac file is supplied then the value of GLOBE_GRIDSIZE needs to be set appropriately. We recommend that, in most cases, you try to generate a mesh with option 5 without supplying a .bac file then use the .bac file and adjustments to the CURVATURE_FACTORS to refine your mesh where necessary. Using the curvature control alone gives you a good starting point in your mesh generation journey. Setting option 5 always generates an octree background spacing function and saves an .Obac file.

Option 6 is similar to option 5 but also calculates the spacing of an existing triangular mesh (read from the file [jobname].fro) and interpolates this to generate a further spacing function. This might be useful for comparing our mesh generator to another, or if you have a surface mesh you wish to adapt but you do not have a spacing function for it.

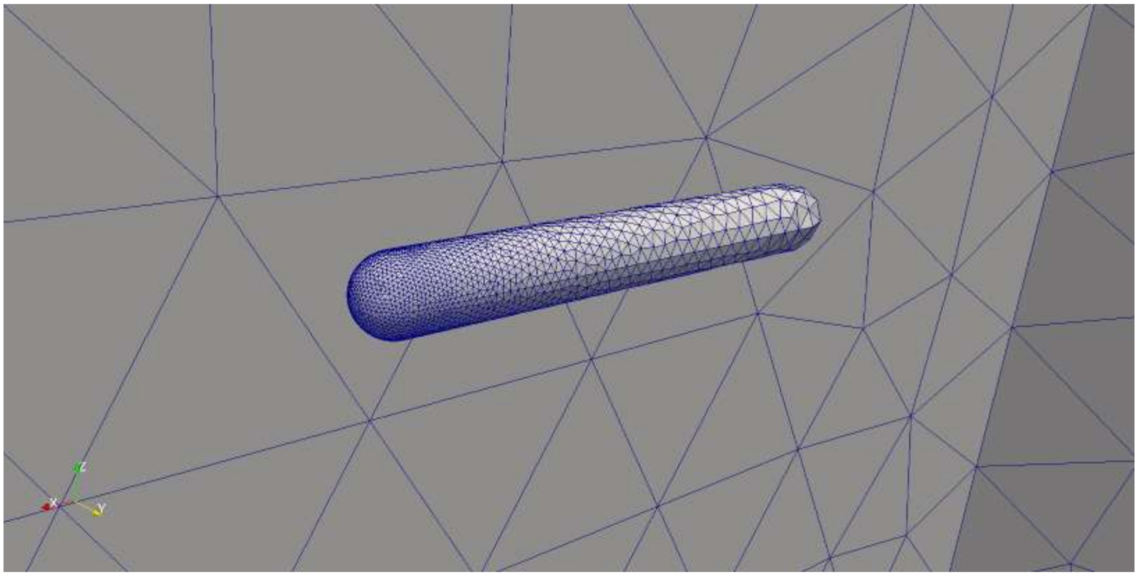
Option 7 simply reads in the file [jobname].Obac and uses this as the spacing function. For complex geometries the curvature control may take some time to generate, reading in the generated .Obac file means you can avoid re--generation.

Examples:

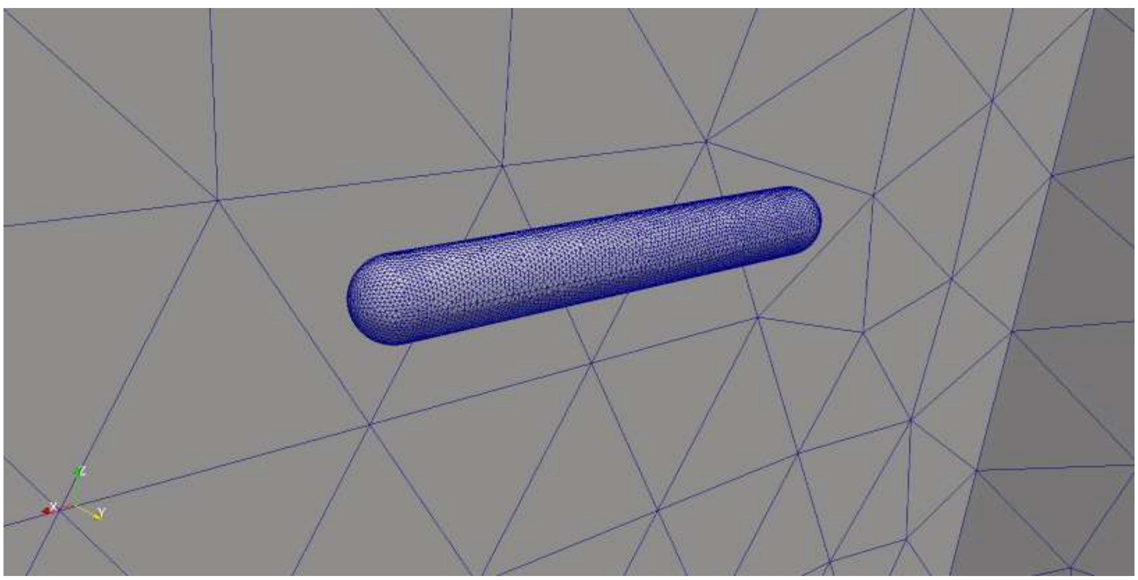
This mesh was generated using just a .bac file (option 2)



This mesh used the same .bac file but an octree spacing function was generated using it (option 4) you can see the gradation is slightly less smooth.



The final mesh was generated using the .bac file with curvature control enabled (option 5), the curvature control has correctly identified that one end of the object requires more refinement.



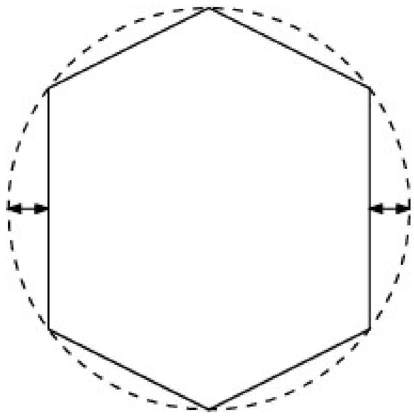
5.3.7 GLOBE_GRIDSIZE

In the case that the file [jobname].bac does not exist then the maximum element size in the mesh will be the value set for this parameter. The default is currently set to 1000.0 but you will need to change this depending on your geometries units. In FLITE we read in the raw values from your CAD file, so you need to set appropriate units in your CAD package.

5.3.8 CURVATURE_FACTORS

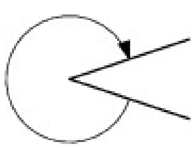
The curvature factors are a set of 7 values which allow you to modify the effect of the curvature control. It is important that you adjust these to suit your problem and the units of your geometry. The values are stored in a 7 element array in the following order:

1. This value is the target spacing as a ratio of the radius of curvature at a point on the surface. Since it is a ratio this value will not change based on your geometry units. If this value was set to 0.2 then the target spacing for a sphere radius 1 would be 0.2.
2. This value is the maximum distance allowed between the mesh and the true surface definition. This distance is indicated in two dimensions on the figure below. This value has units so it is important you consider this before meshing. The curvature control will refine the mesh in order to reduce this distance to the value specified.



3. This value is simply the minimum element size you wish to appear in your mesh, which has units. Setting this too large may restrict the curvature control from achieving the value set in (2), but setting it too small may result in ultra fine meshes when the value set in (2) is unrealistic. Think of this as a sanity check for the curvature control.
4. This value is simply the maximum element size you wish to appear on the surface of your mesh. This is different to GLOBE_GRIDSIZE which applies throughout the volume. It has units so take care to update it appropriately.

The final three values help identify and refine trailing edges. This is done based on the angle between two surfaces, the angle is measured as indicated in the following figure between two surfaces. Imagine this figure is a cut through the trailing edge of a wing.



5. If the angle between two surfaces, in the direction indicated above, is greater than this value then it is considered a trailing edge and will be refined.
6. If the angle between the two surfaces is also bigger than the value here then the refinement is locked to the value set in (7). This stops the spacing approaching zero as the trailing edge gets sharp.
7. The minimum spacing, with units, which will appear at a trailing edge.

5.3.9 STRETCH_LIMIT

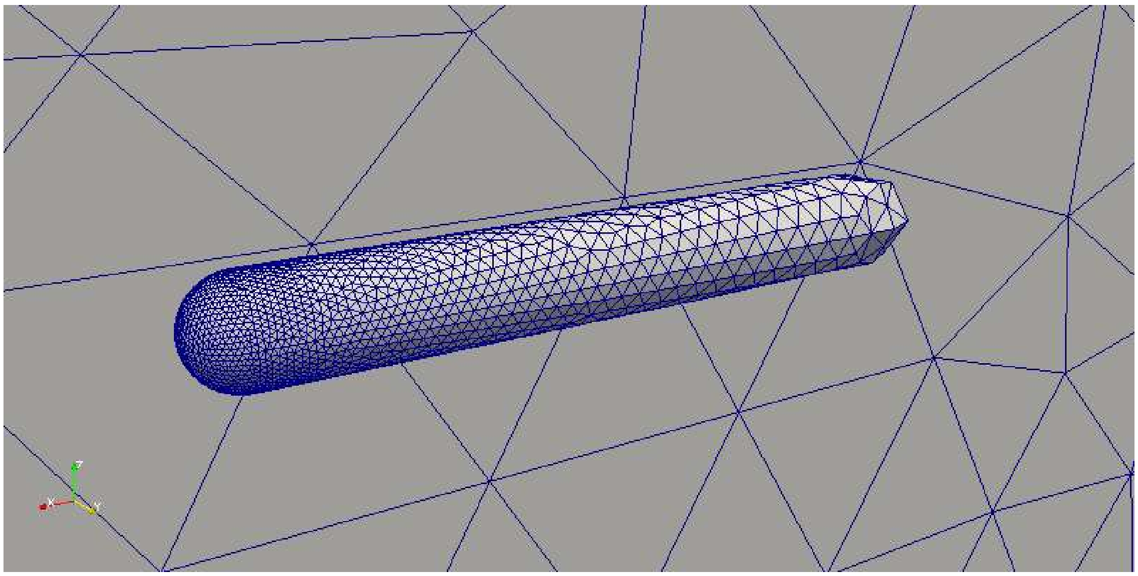
When generating anisotropic meshes this is the maximum stretching allowed.

5.3.10 INTERPOLATE_OCT_MAPPING

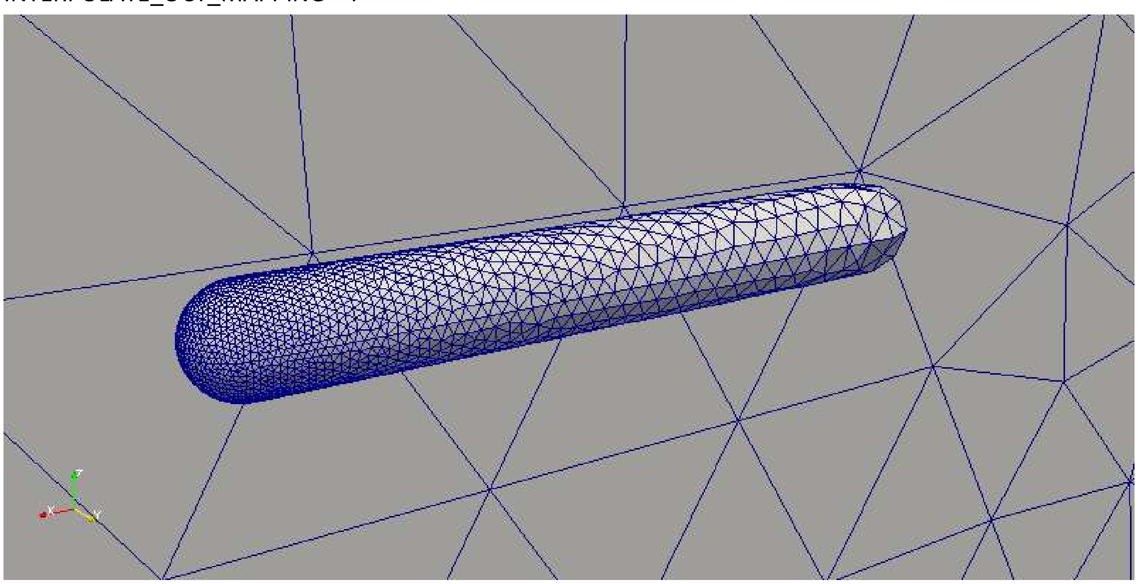
This parameter is either true or false, set with T or F. It only effects the mesh if an octree background function is used. If false (default) the spacing within an octant is constant, if true then the spacing is interpolated smoothly throughout the octant. Setting to true will result in a smoother but potentially much finer mesh.

The following example shows the subtle difference between the two spacing functions:

INTERPOLATE_OCT_MAPPING = T



INTERPOLATE_OCT_MAPPING = F

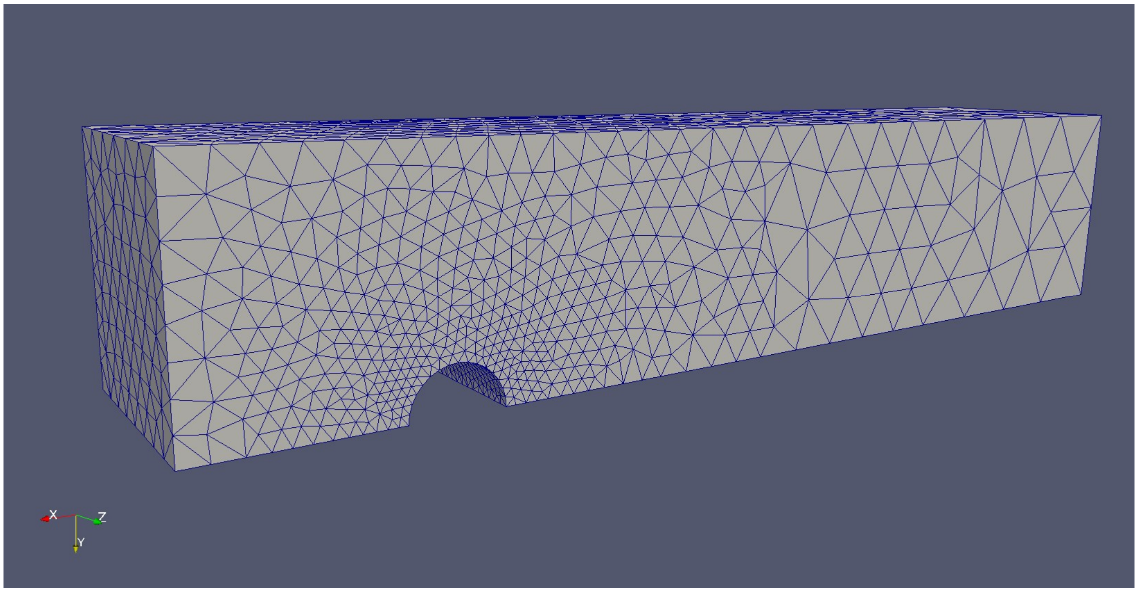


5.3.11 GRADATION_FACTOR

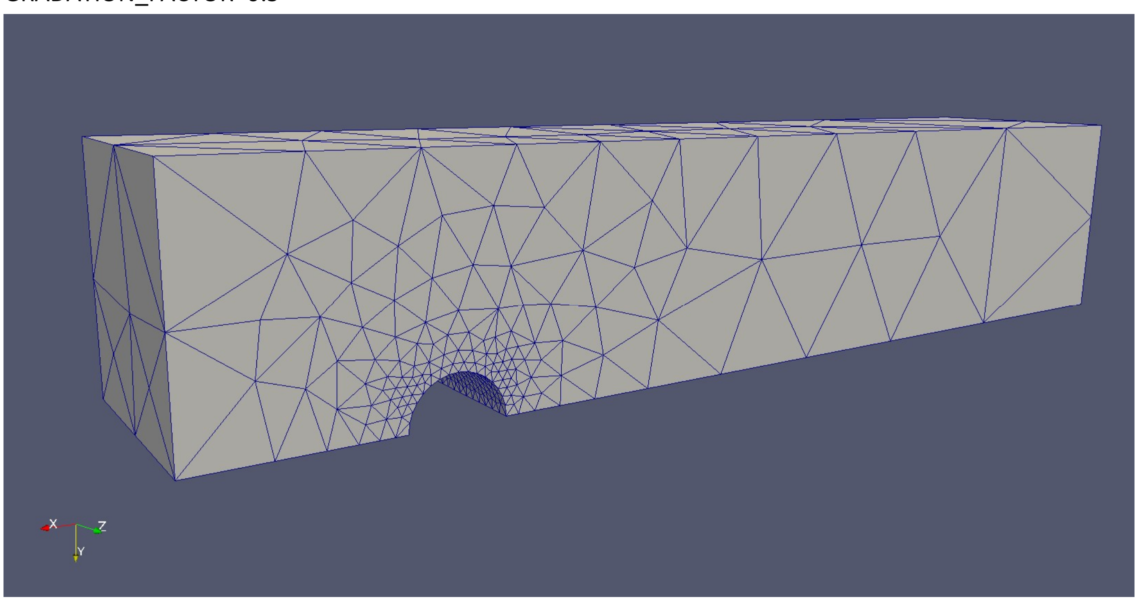
The gradation factor effects how quickly the spacing function decays away from the surface when using curvature control. Setting the value low increases the influence of the curvature control resulting in smoother more refined meshes.

Examples:

GRADATION_FACTOR=0.1



GRADATION_FACTOR=0.5



5.3.12 LOOP_COSMETIC

This is the total number of cosmetics loops to perform after generation. Each of these loops will include a number of smoothing loops, edge collapsing and edge swapping.

5.3.13 LOOP_SMOOTH

This is the number of smoothing loops performed each cosmetics loop. By default this is set to 1 which is fine for most geometries.

5.3.14 SMOOTH_METHOD

There are four different methods for smoothing the mesh, this flag allows you to select which one to use. Method 1 smooths the triangular nodes in the parameter space of the surface, methods 2 and 3 smooth the mesh in the physical space by measuring edge length from surrounding nodes or edge midpoints respectively. Method 4 is an experimental Centroidal Voronoi Tessellation algorithm, currently this is unstable but may be fixed in a later release. In most cases we suggest leaving this to the default of method 3.

5.3.15 COLLAPSE_ANGLE

If the maximum angle (in degrees) in an element is less than this value the code attempts to remove it through edge collapsing.

5.3.16 SWAPPING_ANGLE

If the maximum angle (in degrees) in an element is less than this value the code attempts perform an edge swap to increase the angle in this element.

5.3.17 LOOP_SUPERCOSMETIC

OH?

5.3.18 SUPERCOSMETIC_METHOD

OH?

5.3.19 SUPERCOSMETIC_QUAD

OH?

5.3.20 LOOP_SUPERFRQFILTER

OH?

5.3.21 SUPERFRQFILTER_POWER

OH?

6 Volume Mesher User Guide

The purpose of this section is to provide more detail into how to run and control the operation of the Volume Mesher for your own models.

6.1 Input Files

The Volume Mesher reads a number of input files, most of which must differ in name only by their extension:

- **Surface Mesh** – This is the file output by the Surface Mesher and has a *.fro extension (described in Section ###).
- **Background Mesh** – This file controls the density of the mesh in user-defined regions (described in Section ###).
  This file is optional – if it does not exist then the options in the Control File and the density of the surface mesh will be used.
- **Control File** – This allows the user to define a number of global control parameters which alter the behaviour of the Surface Mesher. (This breaks the naming rule and must be named 'Mesh3D_vx.ctl', where the 'x' is the version of the file (currently 25). This follows the Fortran 90 NAMELIST convention and is described later in this section.
  This file is optional – if it does not exist then the default options will be used.

6.2 Output File

Once the Volume Mesher has finished, a volume mesh is written out as a binary file (*.plt). The format of this is described in Section ###.

6.3 Volume Mesher Control File

This section is devoted to describing the various control parameters for the Volume Mesher. A brief description of each of these can also be viewed by running the Volume Mesher with the '-hh' option.

Running the Volume Mesher with the '-o' option produces a default control file as shown below:

```
!--- please refer README for the control parameters reading...

 &CONTROLPARAMETERS
 DEBUG_DISPLAY          = 1,
 START_POINT            = 1,
 CURVATURE_TYPE         = 1,
 ADTREE_SEARCH          = F,
 BOUND_INSERT_SWAP      = 0,
 BOUND_INSERT_SOFT      = 0,
 BOUND_INSERT_ORDER     = 0,
 RECOVERY_TIME          = 1,
 RECOVERY_METHOD        = 3,
 ELEMENT_BREAK_METHOD   = 8
 /
 &FRAMEPARAMETERS
 FRAME_TYPE           = 0,
 FRAME_STITCH         = 1,
 IDEAL_DIST_GAP       = 1.00,
 IDEAL_LAYER_GAP      = 2,
 IDEAL_LAYER_EXPAND   = 16
 /
 &BACKGROUNDPARAMETERS
 BACKGROUND_MODEL          = 7,
 STRETCH_LIMIT             = 10.00,
 STRETCHED_BOUND_INSERT    = F,
 MAPPING_INTERP_MODEL      = 1,
 MAPPING_CHOOSE_MODEL      = 1,
 INTERPOLATE_OCT_MAPPING   = F,
 GRADATION_FACTOR          = 0.50
 /
 &COSMETICPARAMETERS
 LOOP_CVT          = 0,
 CVT_TIME_LIMIT    = 0.50,
 LOOP_COSMETIC     = 1,
 LOOP_SMOOTH       = 5,
 SMOOTH_METHOD     = 2,
 BOUNDCELL_SPLIT   = 0,
 COLLAPSE_ANGLE    = 12.50,
 SWAPPING_ANGLE    = 30.00,
 HIGHORDER_MODEL   = 1
 /
 &BDLAYERPARAMETERS
 NORSMOOTH_METHOD   = 1,
 LAYER_HYBRID       = 0
 /
```

6.3.1 DEBUG_DISPLAY

The debug display parameter determines how much information is output to the screen and how many internal checks are performed on the mesh. By default this is set to 1 and takes a value between 0 and 4. If you are having trouble meshing a geometry increasing this value is a good way to start diagnosing your problem.

6.3.2 START_POINT

If START_POINT=1 then the surface mesh [jobname].fro is read and a volume mesh is generated. If START_POINT=2 the mesh generator reads in an existing volume mesh from the file [jobname]_0.plt and uses this as the starting point. This is useful if you wish to experiment with different cosmetics methods or you simply want to run more cosmetics on an existing mesh.

6.3.3 CURVATURE_TYPE

The curvature type parameter tells the mesher which geometry definition format you are using. There are four options 1-4. Setting the value to 1 will tell the mesher to read the geometry from the file [jobname].dat in the Swansea University file format. The default value, 4, makes the mesher read in either an IGES or STEP CAD file. When option 4 is selected the mesh will be generated on the exact CAD definition. The geometry file is rarely used in the volume mesh generation process, it is unlikely you will need to change this option.

6.3.4 ADTREE_SEARCH

This option is set to either true or false, T or F. If set to true then an AD-Tree is used whenever the code needs to determine which element a point lies within. In most normal mesh generation scenarios we usually have an accurate first guess to which element a point may lie, hence enabling this option will only have make a marginal improvement to performance. In most cases it is best to leave this option turned off to avoid the extra memory requirements it entails.

6.3.5 Boundary Recovery

The first step of the volume mesh generation is to generate a triangulation connecting all the nodes in the surface mesh. This triangulation (made up of tetrahedron) will not contain the exact triangulation that is present in the original surface mesh. At some point during the generation process this exact triangulation needs to be recovered. This is called boundary recovery and is the most likely place volume meshing will fail. A number of parameters control how boundary recovery takes place. Many of the defaults of these settings have been determined from years of experience with complex industrial geometries. It is unlikely that you will need to change these, with the possible exception of BOUND_INSERT_SOFT.

6.3.6 BOUND_INSERT_SWAP

This controls edge swapping directly after the boundary nodes have been triangulated. The default value is 3 which swaps edges based on the number of boundary edges, 2 swaps based on the dihedral angle of elements, 1 swaps based on the volume of elements and 0 does no swapping at all.

6.3.7 BOUND_INSERT_SOFT

The boundary nodes are triangulated by recursively inserting them into the Delaunay triangulation of a large cube. It is possible that some nodes fail to insert. This would usually happen when two nodes are very close to each other in the surface mesh. Occasionally changing the order nodes are inserted fixes this problem. By setting BOUND_INSERT_SOFT=1 the code will store any nodes which fail to insert and try inserting them again at the end of the process. The default value is 0 which will stop the process as soon as one node fails. If the boundary insertion fails when BOUND_INSERT_SOFT=0 then changing it to 1 may fix the problem, however be aware that this could indicate there is a problem with your surface mesh.

6.3.8 BOUND_INSERT_ORDER

This parameter controls the order at which the boundary nodes are inserted into the triangulation. Either boundary nodes are inserted in the order they are written in the surface mesh, BOUND_INSERT_ORDER=0, or the spacing function is calculated at each node and they are ordered in such a way that the nodes with largest spacing are inserted first, BOUND_INSERT_ORDER=1.

6.3.9 RECOVERY_TIME

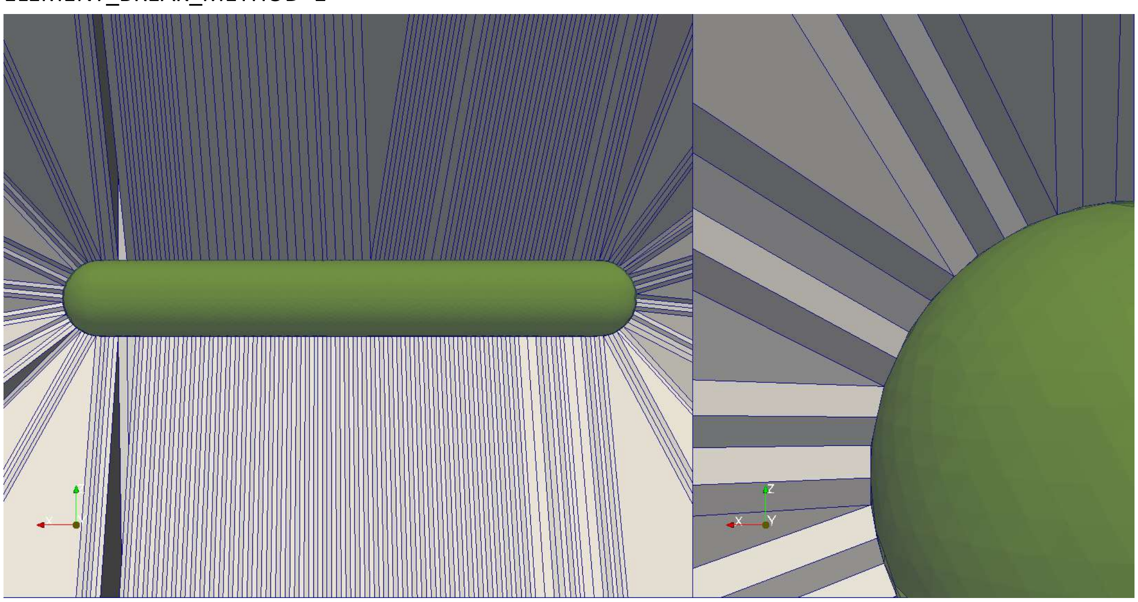
The RECOVERY_TIME parameter controls when the boundary triangulation is recovered. There are three settings, setting RECOVERY_TIME=0 does not recover the boundary. This allows you to find the Delaunay triangulation of a set of points, normally this is not used. RECOVERY_TIME=1 recovers the boundary before the volume elements are generated, this is the default setting and should be used in most cases. It is possible to recover the boundary after the volume elements are generated by setting RECOVERY_TIME=2, but this is computationally more expensive since the volume mesher will refine the mesh in and outs of the surface mesh.

6.3.10 ELEMENT_BREAK_METHOD

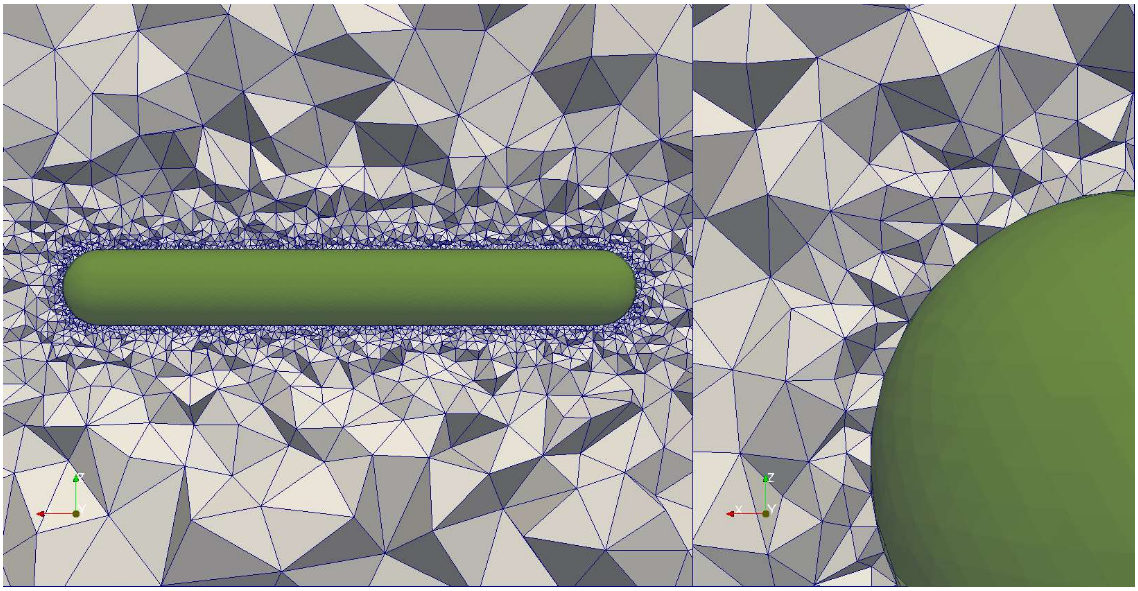
There are a number of different methods for generating the volume nodes. Depending on your application some techniques may be better suited than others. In total there are 10 different methods, the default method has been set since it produces the best quality meshes for most normal applications. The best way to learn the difference between these methods is through examples, the following figures show meshes generated using the different techniques available.

6.3.11 ELEMENT_BREAK_METHOD=0

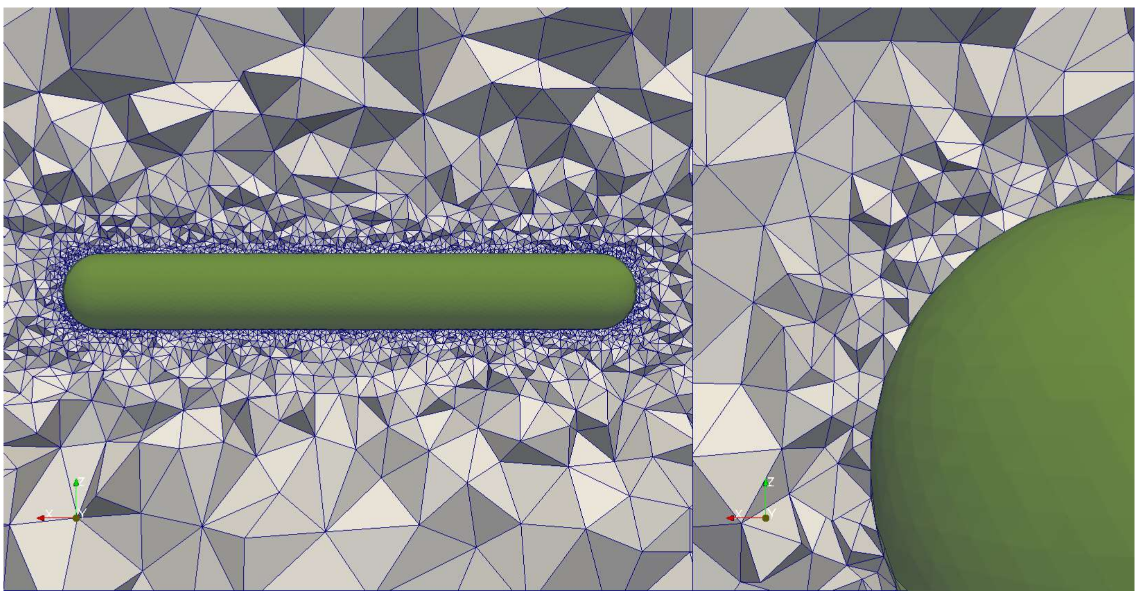
ELEMENT_BREAK_METHOD=1



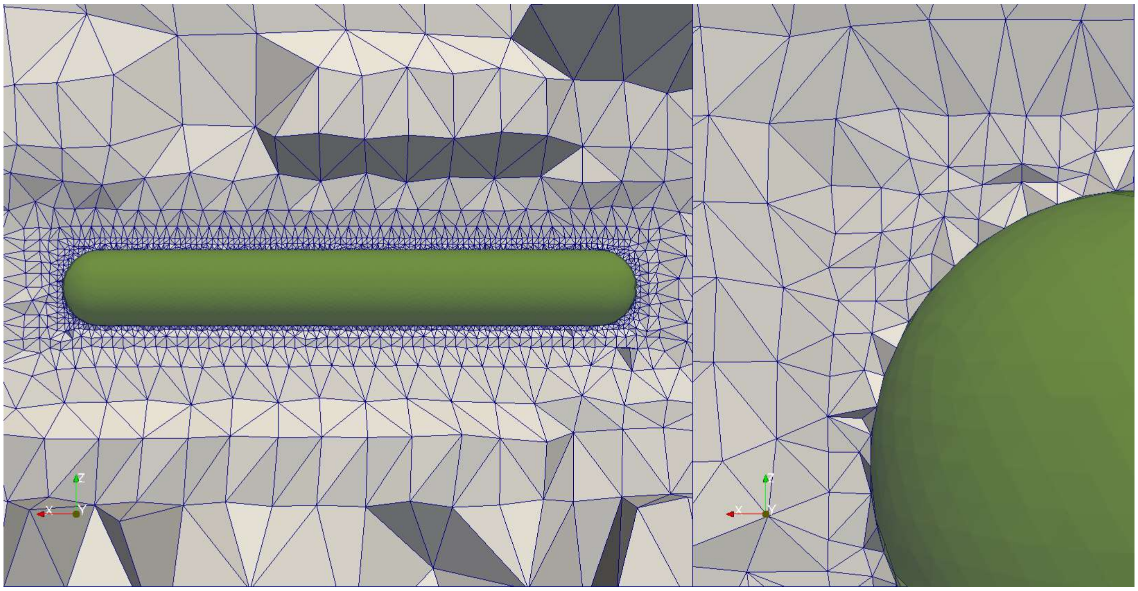
ELEMENT_BREAK_METHOD=2



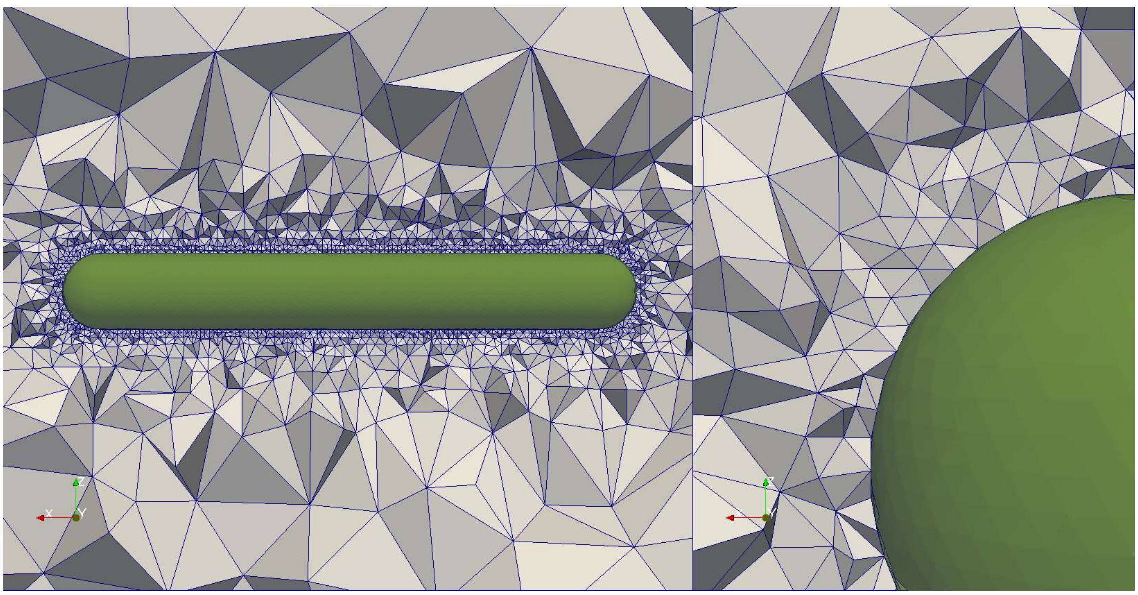
ELEMENT_BREAK_METHOD=3



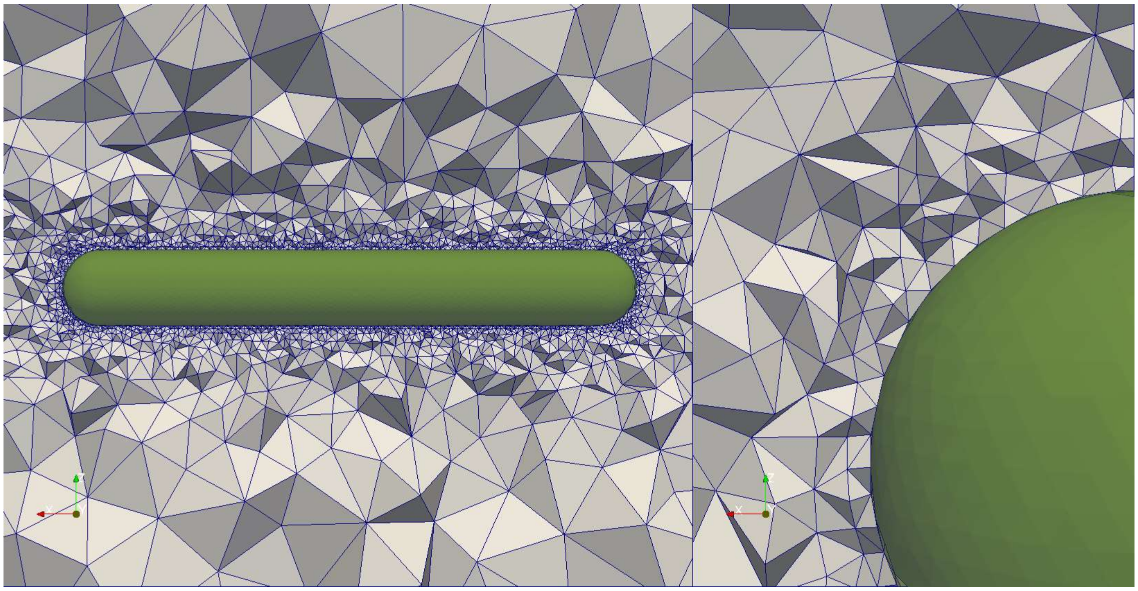
ELEMENT_BREAK_METHOD=6



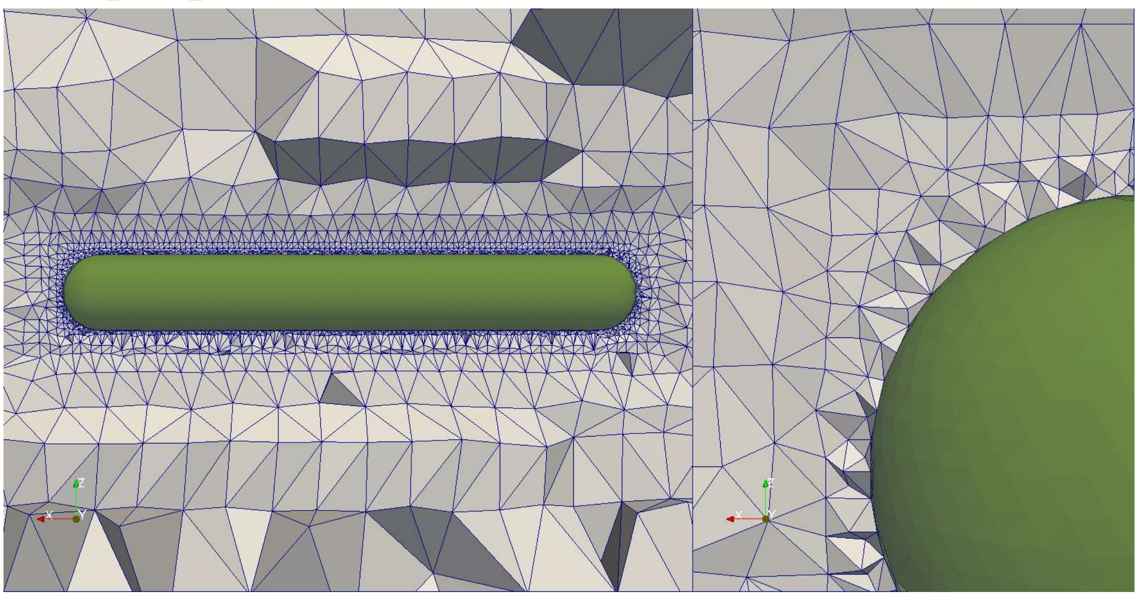
ELEMENT_BREAK_METHOD=7



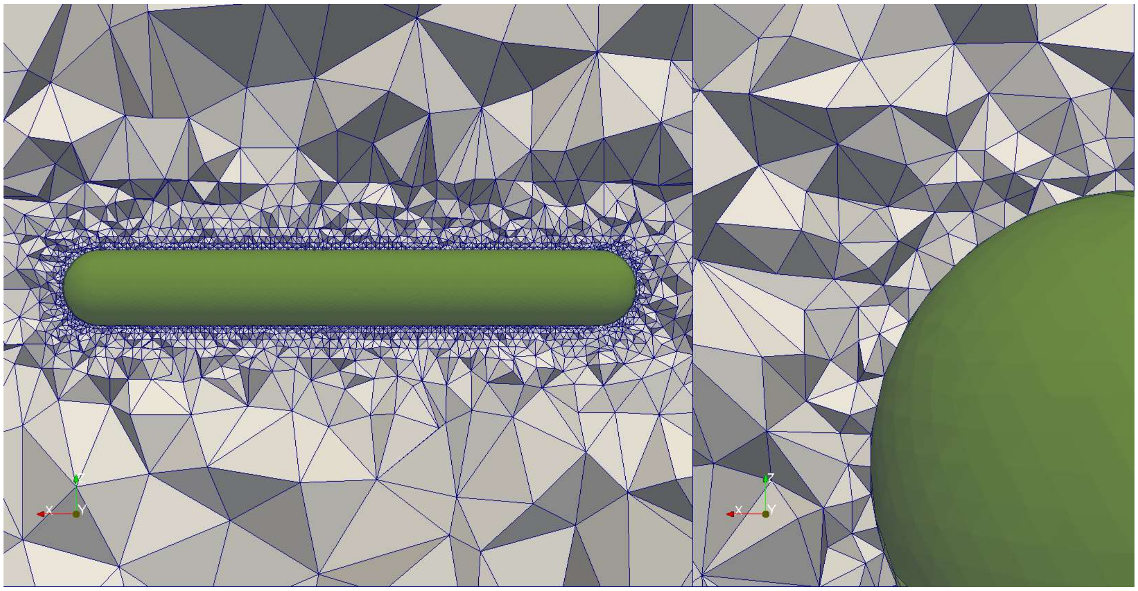
ELEMENT_BREAK_METHOD=8



ELEMENT_BREAK_METHOD=9



ELEMENT_BREAK_METHOD=10



6.3.12 BACKGROUND_MODEL

6.3.13 STRETCH_LIMIT

When generating anisotropic meshes this is the maximum stretching allowed.

6.3.14 STRETCHED_BOUND_INSERT

6.3.15 INTERPOLATE_OCT_MAPPING

This parameter is either true or false, set with T or F. It only effects the mesh if an octree background function is used. If false (default) the spacing within an octant is constant, if true then the spacing is interpolated smoothly throughout the octant. Setting to true will result in a smoother but potentially much finer mesh.

6.3.16 GRADATION_FACTOR

6.3.17 LOOP_CVT

Described in detail here

6.3.18 CVT_TIME_LIMIT

Described in detail here

6.3.19 LOOP_COSMETIC

This value specifies the number of global cosmetics loops. Each of these loops will include a number of smoothing loops, boundary element splitting, and element collapse/edge swapping.

6.3.20 LOOP_SMOOTH

This is the number of smoothing loops which take place every cosmetics loop.

6.3.21 SMOOTH_METHOD

There are 4 different smoothing methods which can be selected with this parameter. SMOOTH_METHOD=1 is a basic unweighted Laplace smoothing, each node moves towards the centroid of all the nodes it is connected too. SMOOTH_METHOD=2 attempts to move the position of a node to a position which sets the volume of each element it belongs to the same. SMOOTH_METHOD=3 is the same as SMOOTH_METHOD=1 except the centroid is weighted based on the volume of the surrounding elements, the node moves closer to nodes connected to larger elements. The final method SMOOTH_METHOD=4 uses Powell optimisation to find a location which makes each connected element the correct volume specified by the background spacing function.

6.3.22 BOUNDCELL_SPLIT

If this is set to 1 the mesh generator will split any elements which are made up of only boundary nodes before the mesh cosmetics loop. If it is set to 2 this split will be made before and after cosmetics which ensures that no boundary only cells exist. For no such treatment set the parameter to 0.

6.3.23 COLLAPSE_ANGLE

Attempts will be made to collapse elements whose maximum dihedral angle is lower than the value set for this parameter. This is measured in degrees.

6.3.24 SWAPPING_ANGLE

Attempts will be made to swap edges to eliminate elements whose maximum dihedral angles are lower than the values specified for this parameter. This is measured in degrees.

6.3.25 HIGHORDER_MODEL

If the surface mesh used to generate this volume mesh is a high order mesh then the order of this mesh should be specified here.

6.3.26 NORSMOOTH_METHOD

This parameter controls boundary layer generation which is discussed here.

6.3.27 LAYER_HYBRID

This parameter controls boundary layer generation which is discussed here.