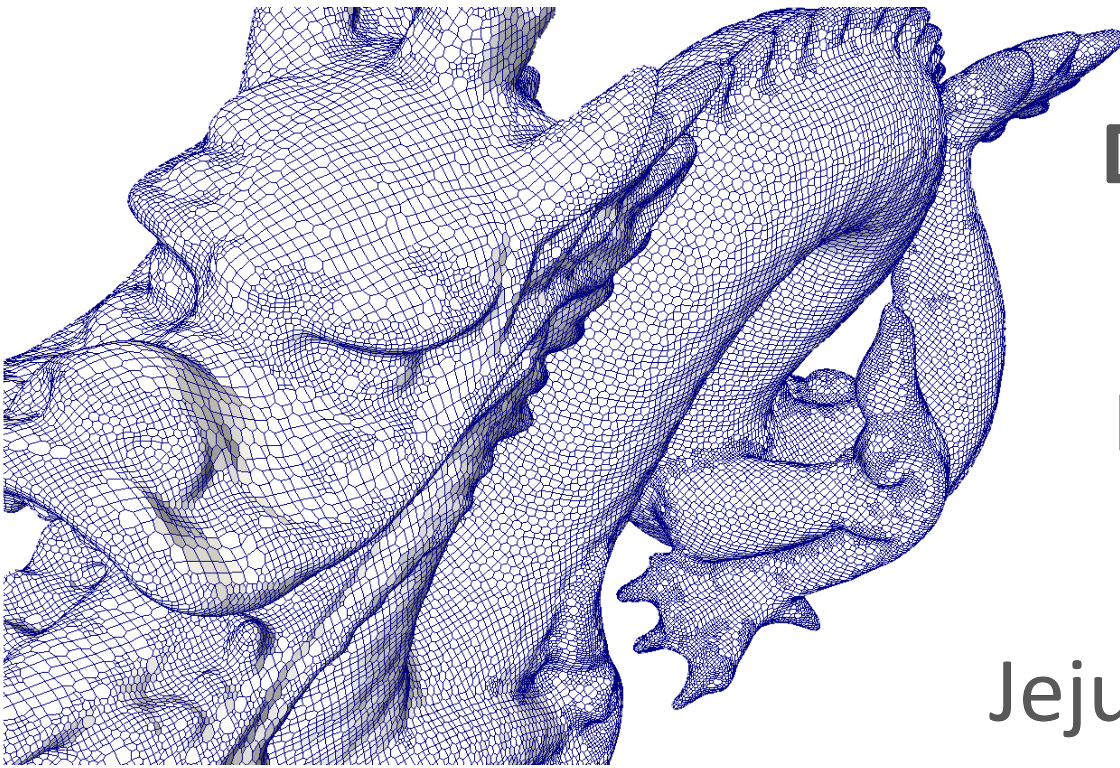
Advanced Pre-Processing and Meshing with `snappyHexMesh`



Daniel P Combest

Andrew Jackson

Eugene de Villiers

8th OF Workshop

Jeju Korea June 2013

Disclaimer

OPENFOAM® is a trademark of OpenCFD (ESI Group). This presentation is not approved or endorsed by OpenCFD or ESI Group, the owner of the OPENFOAM® trademark.

Contents

- `snappyHexMesh` (40 Minutes)
  - Description
  - Background
  - Methodology Overview
  - HELYX-OS GUI
- Hands-On Meshing Examples (40 Minutes)
  - Initial Setup
  - Snapping
  - Layer Addition
- Closing Remarks (5 Minutes)

Before we Begin

To take part in hands-on portion of training

- Must have HELYX-OS installed (Already on workshot virtual Machine)
- Download patch and training material from https://sourceforge.net/projects/helyx-os/files
  - Material includes presentations "training_material.zip"
  - Install patch where directory "Engys/HelyxOS/v1.0.2" is located with

```
tar xvjf 20130610-Engys-HelyxOS-1.0.2-linux-x86_64-PATCH.tar.bz2
```

Enjoy!

Description

- Utility **`snappyHexMesh`** is used to create high quality hex-dominant meshes based on arbitrary geometry
- Controlled by dictionary *system/snappyHexMeshDict*
- This utility has the following key features:
  - Fully parallel execution
  - STL and Nastran (.nas) files support for geometry data
  - Preservation of feature edges
  - Addition of wall layers
  - Zonal meshing for support of porous media and MRF
  - Quality guaranteed final mesh that will run in OPENFOAM®

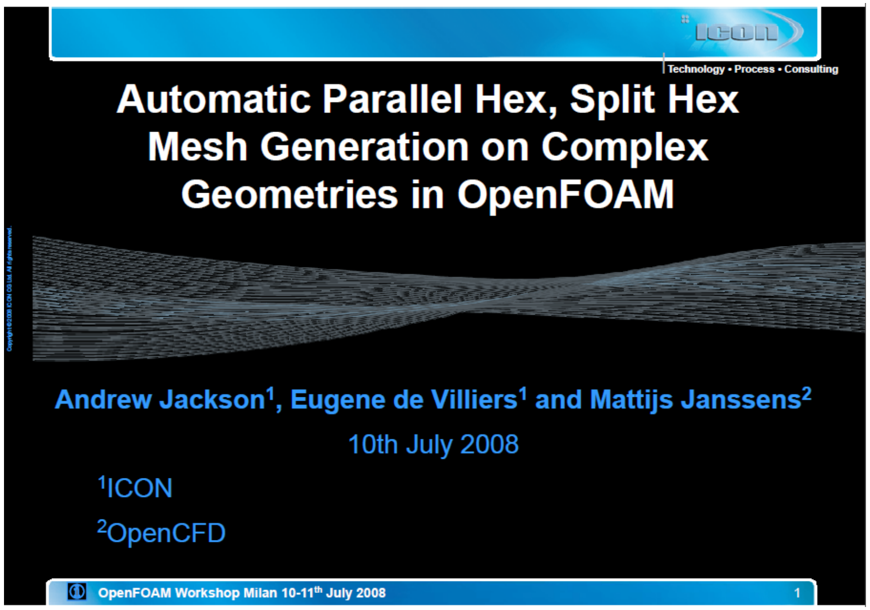
Contents

- `snappyHexMesh` (40 Minutes)
  - Description
  - Background
  - Methodology Overview
  - HELYX-OS GUI
- Hands-On Meshing Examples (40 Minutes)
  - Initial Setup
  - Snapping
  - Layer Addition
- Closing Remarks (5 Minutes)

Background | Original and Current Dev.

Debut in 2008 at 3rd OpenFOAM Workshop in Milan by original developers

- Andrew Jackson
- Eugene de Villiers
- Mattijs Janssens



New Improvements in snappyHexMesh in Standard OpenFOAM 2.2.x

- Additional layer control functionality (slipFeatureAngle)
- Additional feature capturing options (implicitFeatureSnap)
- faceType for internal face definition (baffle, boundary, internal)
- Distance refinement from eMesh files

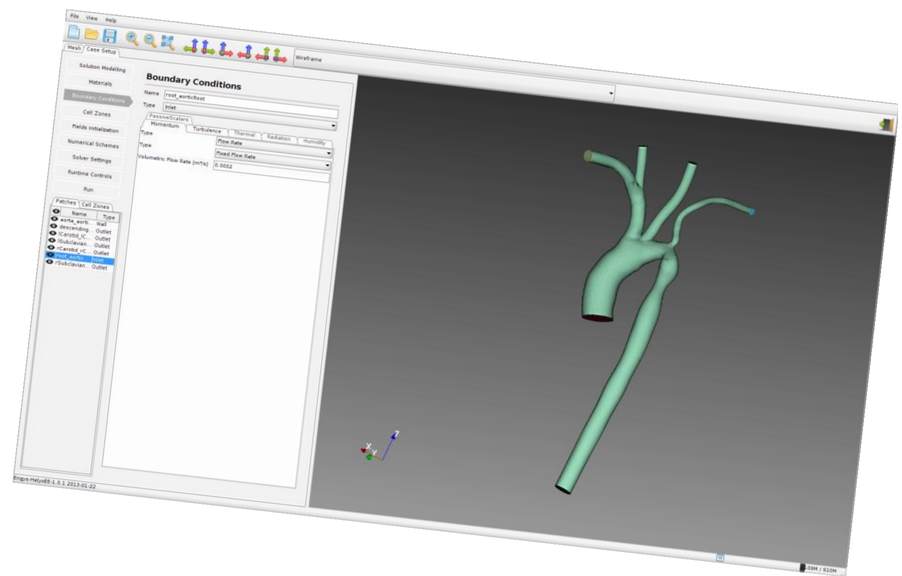
engys

Background | Original and Current Dev.

Engys' Continued Developments in an In-House Version

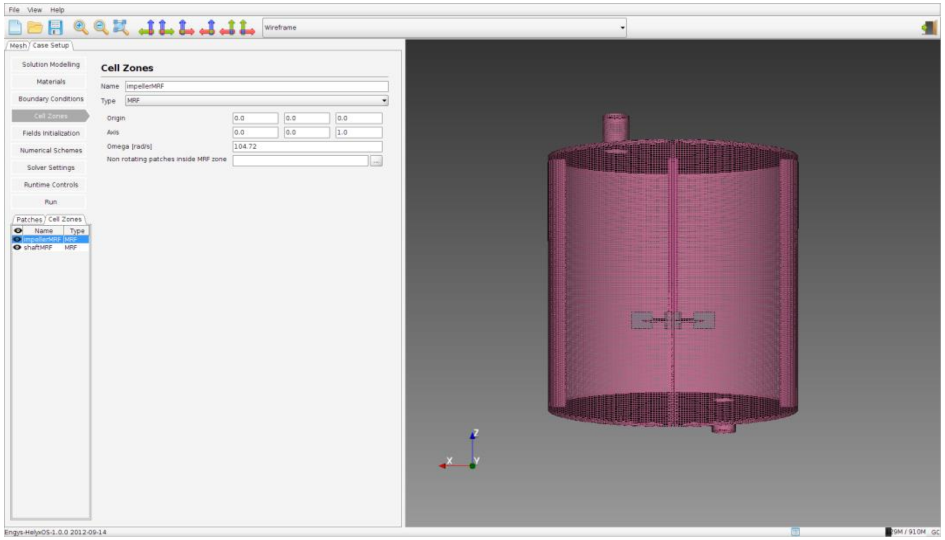
- Enhanced feature capturing and automation
- Improved layers and layer specification methods
- Layers growing up patches
- Generation of Internal layers
- Proximity based refinement
- Multi layer addition
- Automatic block mesh creation and decomposition
- Mesh wrapping and small leak closure
- and many other extensions

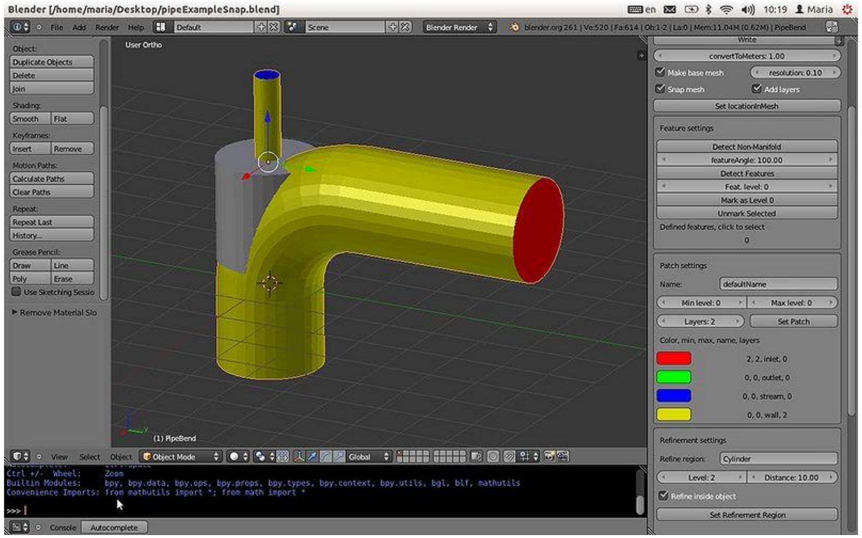
- Enhanced parallel performance
  - Reduced overall memory usage
  - Improved scaling for 32+ processors
- Graphical User interface Integration
  - HELYX
  - HELYX-OS
  - ELEMENTS



Background | Evolution

Open Source Preprocessors that Drive **`snappyHexMesh`**





HelyxOS

Developer: **Engys**
Released: 2012
Version: 1.0.2 for OpenFOAM-2.2.x
Language: JAVA + VTK
http://engys.com/products

swiftSnap

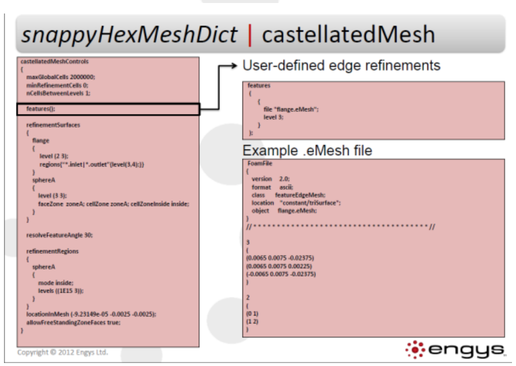
Developer: **Karl-Johan Nogenmyr**
Released: 2012
Version: OpenFOAM-2.1.x
Language: Blender 2.67 Plugin
http://openfoamwiki.net/index.php/Contrib/SwiftSnap

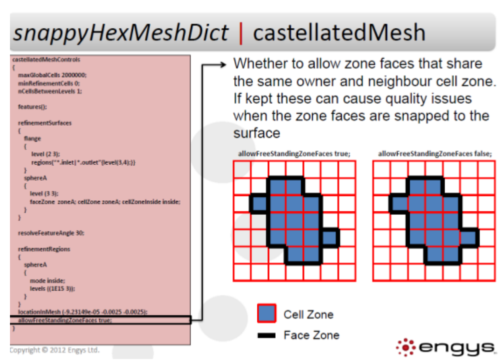
engys

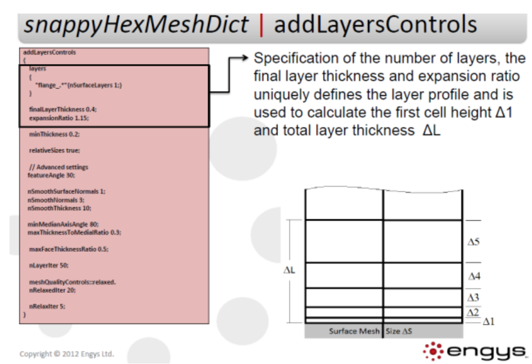
Background | Useful References

Recommended Source

- ✓ Covers some of today's topics
- ✓ Straight from developer
- ✓ See local copy in Appendix A1 of this presentation

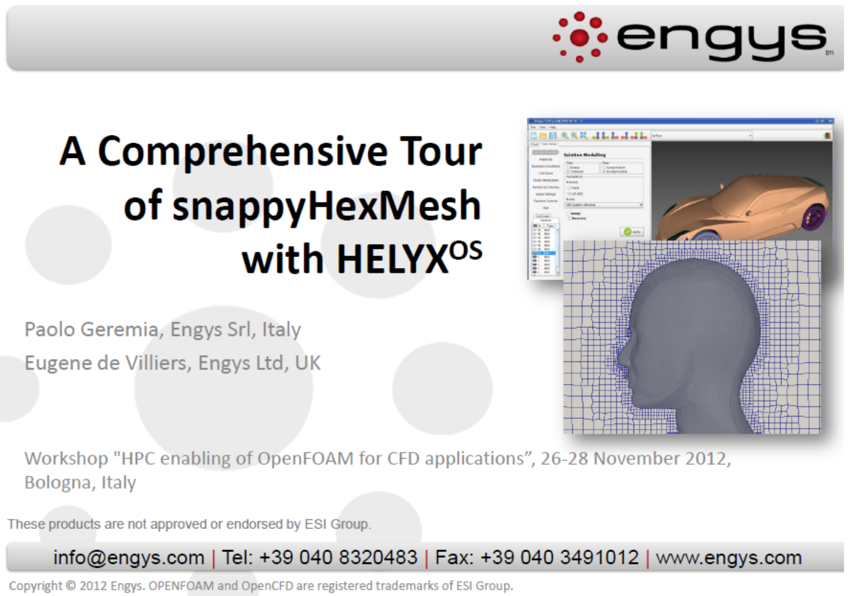






engys

Background | Useful References

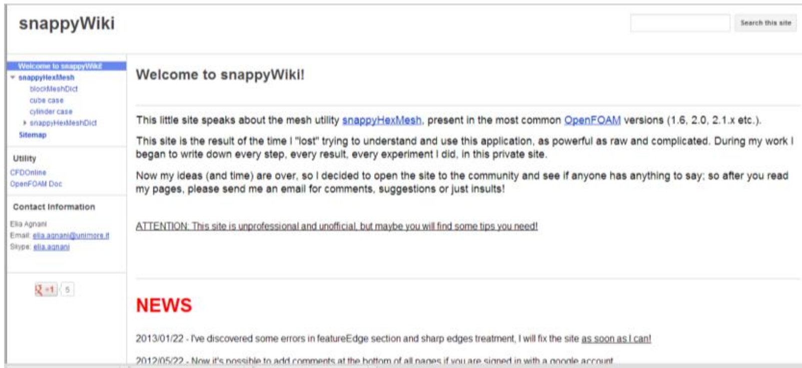


✓Other presentations

✓CFD-Online Forums

✓snappyWiki

✓Standard Tutorials



Contents

- `snappyHexMesh` (40 Minutes)
  - Description
  - Background
  - Methodology Overview
  - HELYX-OS GUI
- Hands-On Meshing Examples (40 Minutes)
  - Initial Setup
  - Snapping
  - Layer Addition
- Closing Remarks (5 Minutes)

Methodology Overview | Usage

- Define *snappyHexMeshDict* → Execute `snappyHexMesh`
- Execution:

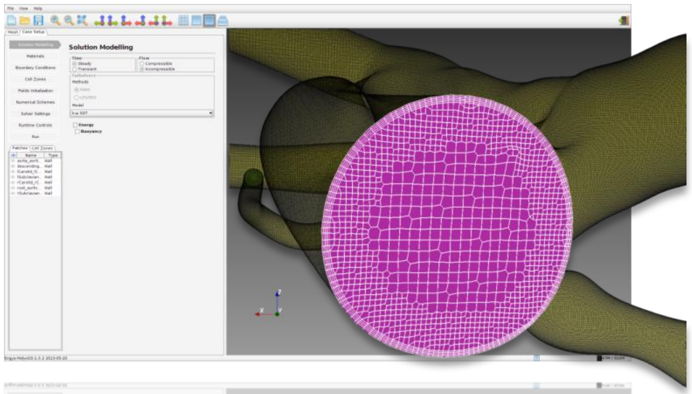
  snappyHexMesh [-noFunctionObjects][-overwrite] [-parallel] [-case dir] [-roots <(dir1 .. dirN)>] [-help]

  - Parallel execution available using **mpirun**
- Requirements:
  - Dictionary file *system/snappyHexMeshDict*
  - Geometry data (stl, nas, obj) in *constant/triSurface*
  - Hexahedral base mesh (decomposed if running in parallel)
  - Dictionary file *system/decomposeParDict* for parallel runs
  - All *system* dictionaries (e.g *controlDict, fvSchemes, fvSolutions*)

Methodology Overview | Workflow

- Define blockMeshDict
- Define snappyHexMeshDict
- Define decomposeParDict
- Execute `blockMesh`
- Execute `decomposePar`
- Execute `snappyHexMesh`

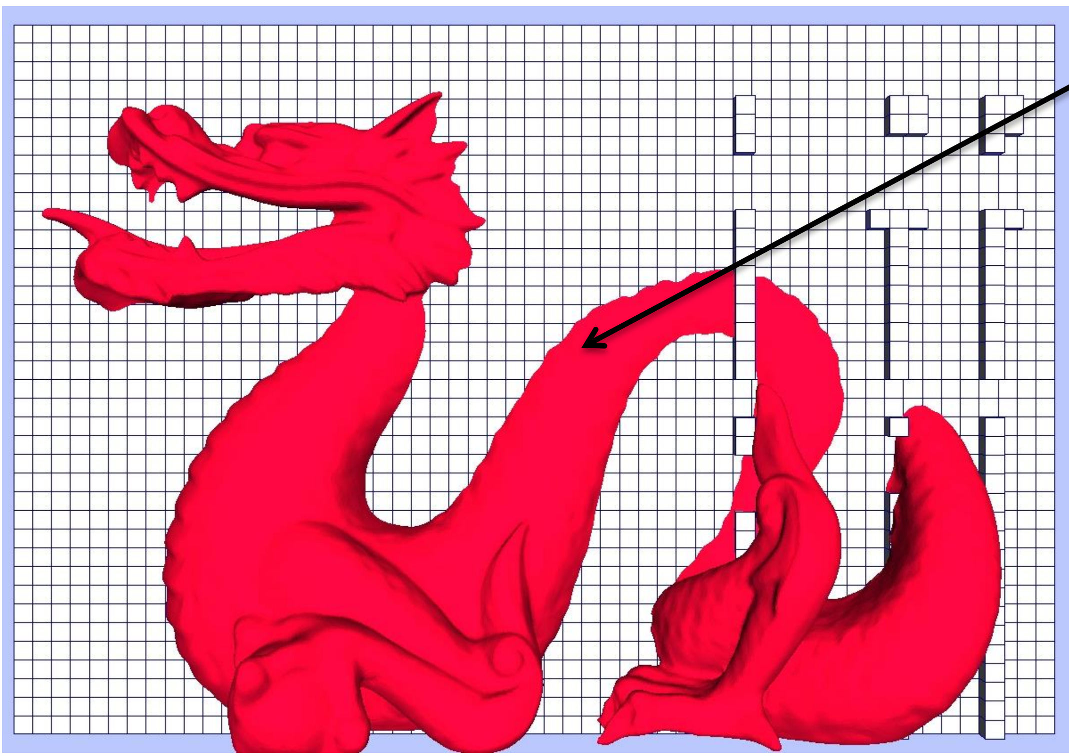
- Manually
- Scripted
- Graphical User Interface



Methodology Overview | Base Mesh

- Step 1: Create base mesh
  - Custom made → Using utility `blockMesh`

Note: Cells should be close to unit Aspect Ratio for optimum behaviour



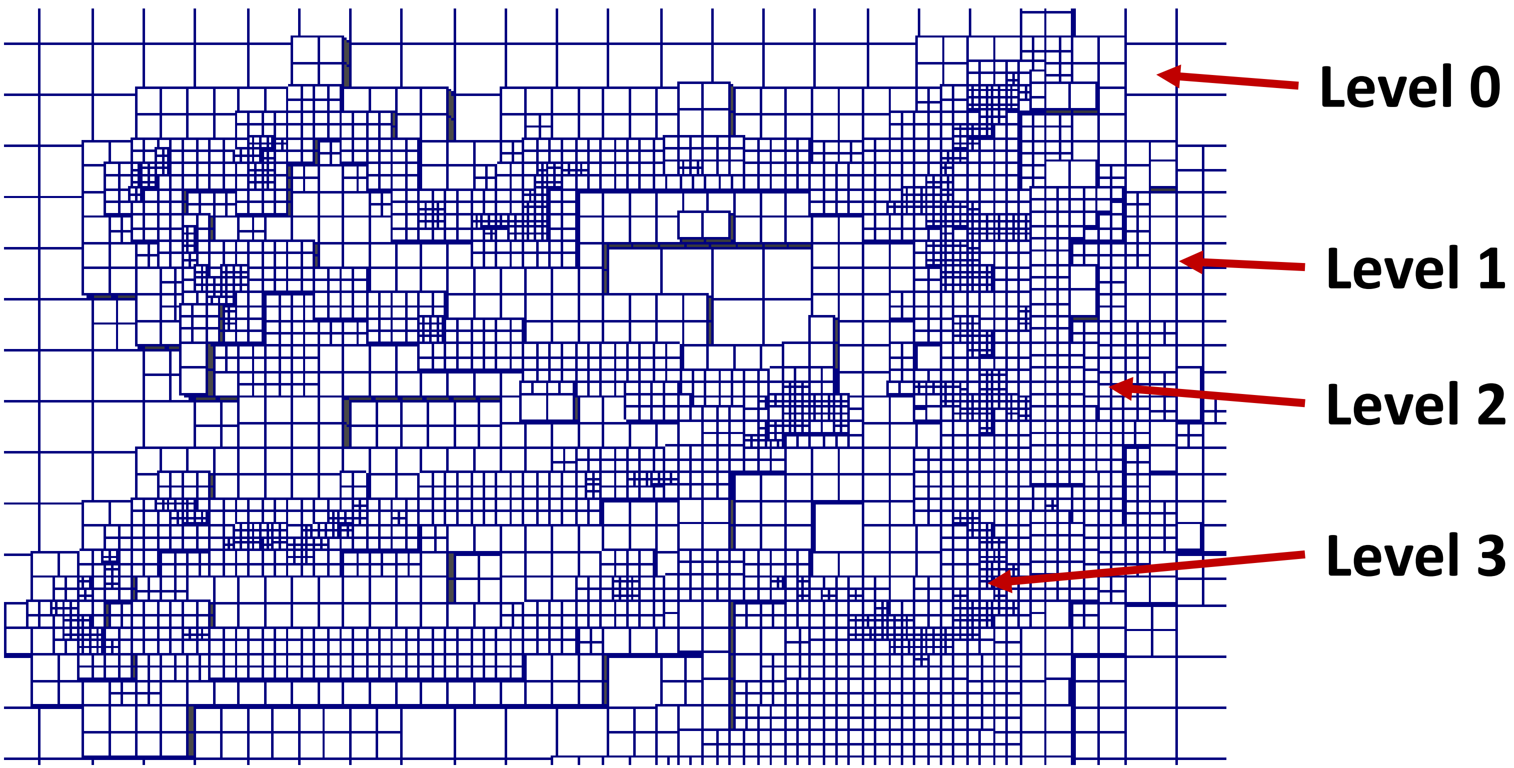
STL or Nastran (*constant/triSurface*)

NOTE: File names must not contain any spaces, unusual characters or begin with a number. The same applies to the names of the parts and regions defined within the geometry files.

Hexahedral base mesh → level 0 size

Methodology Overview | Base Refine

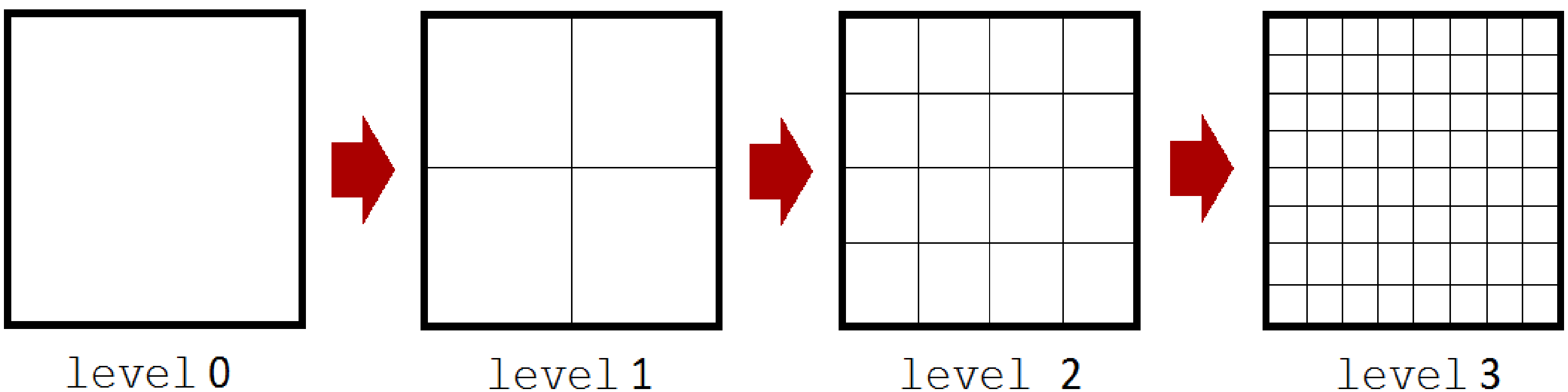
- Step 2: Refine base mesh
  - Surface refinement → feature lines, proximity & curvature
  - Volume refinement → closed surfaces, geometric shapes



engys

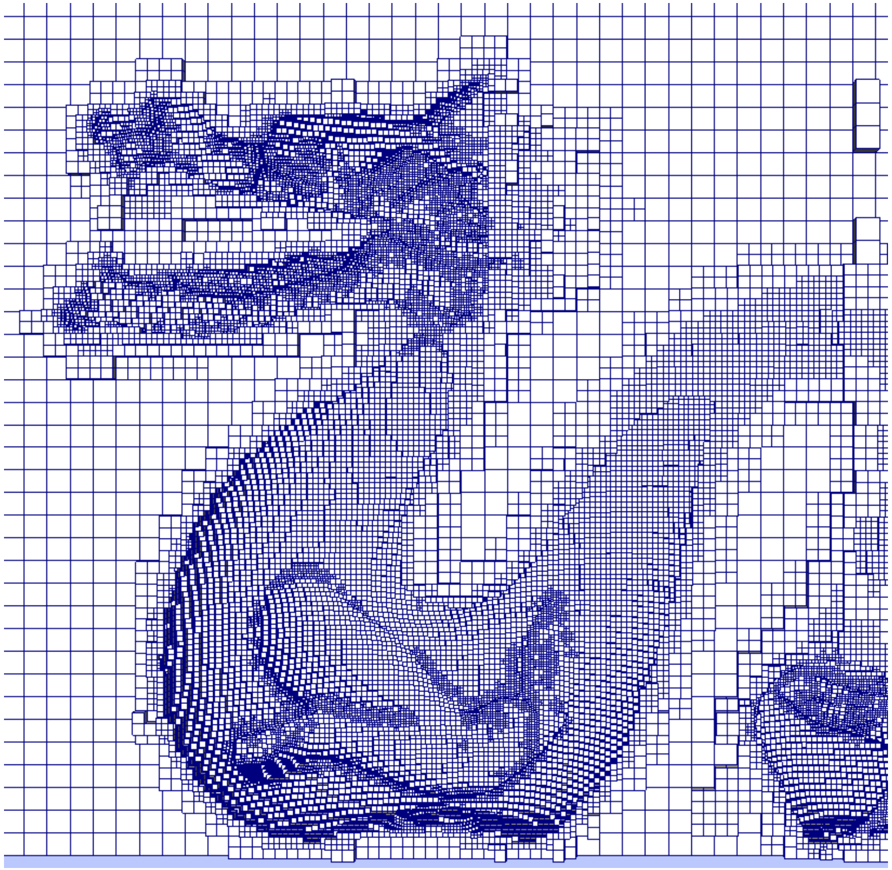
Methodology Overview | Base Refine

- Step 2: Refine base mesh
  - Surface refinement → feature lines, proximity & curvature
  - Volume refinement → closed surfaces, geometric shapes

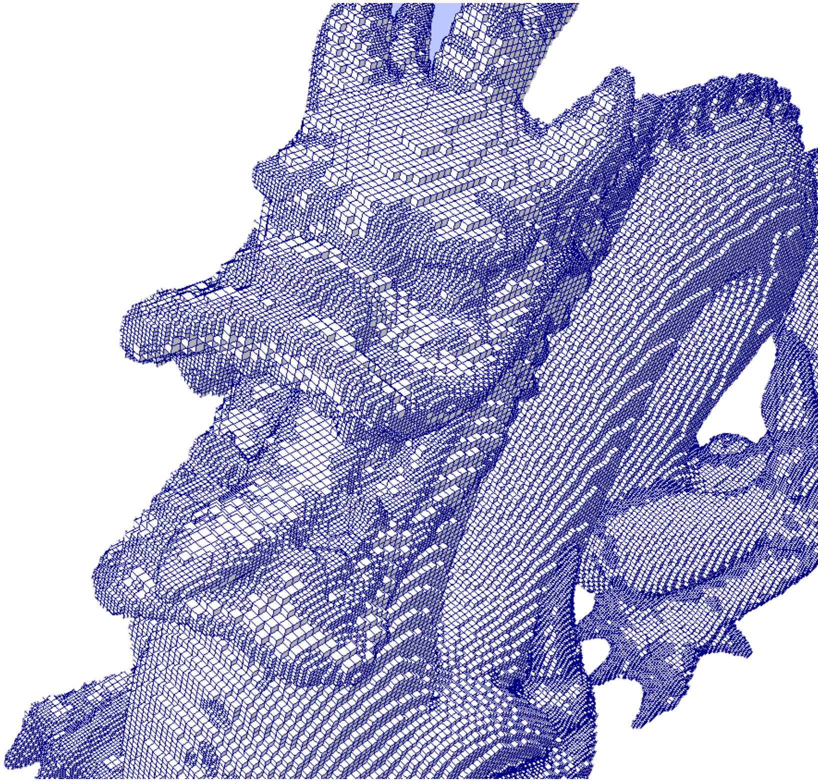


Methodology Overview | Remove Cells

- Step 3: Remove unused cells
  - User defines keep point

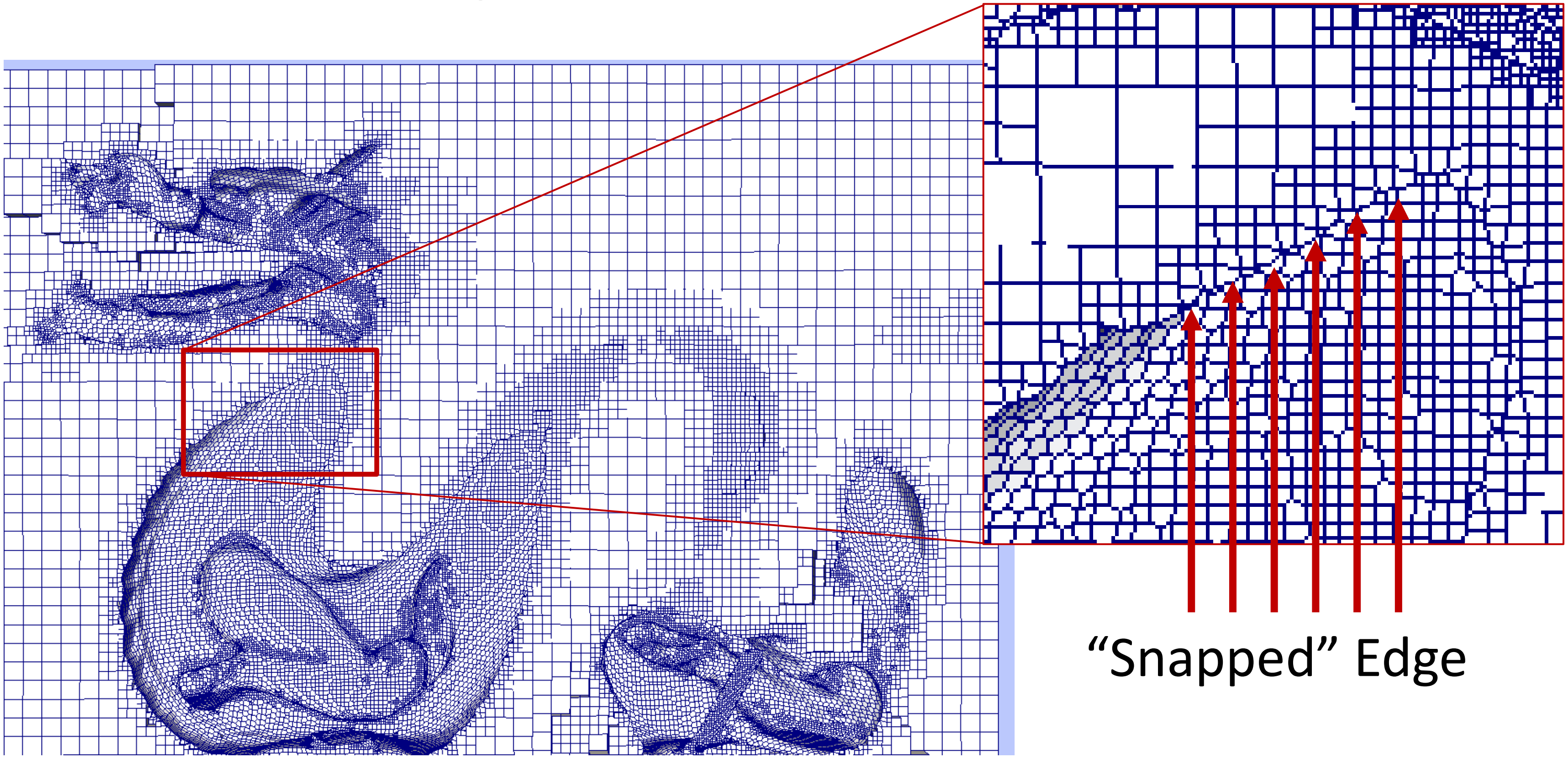


Castellated mesh



Methodology Overview | Snapping

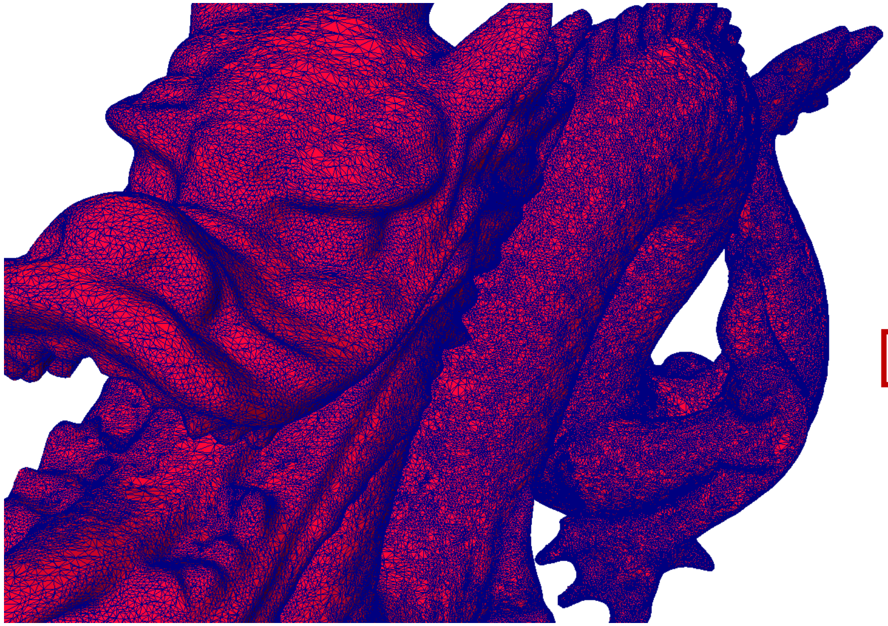
- Step 4: Snap mesh to surface
  - Implicit wrapping → Preserve features
  - Smooth & Merge faces



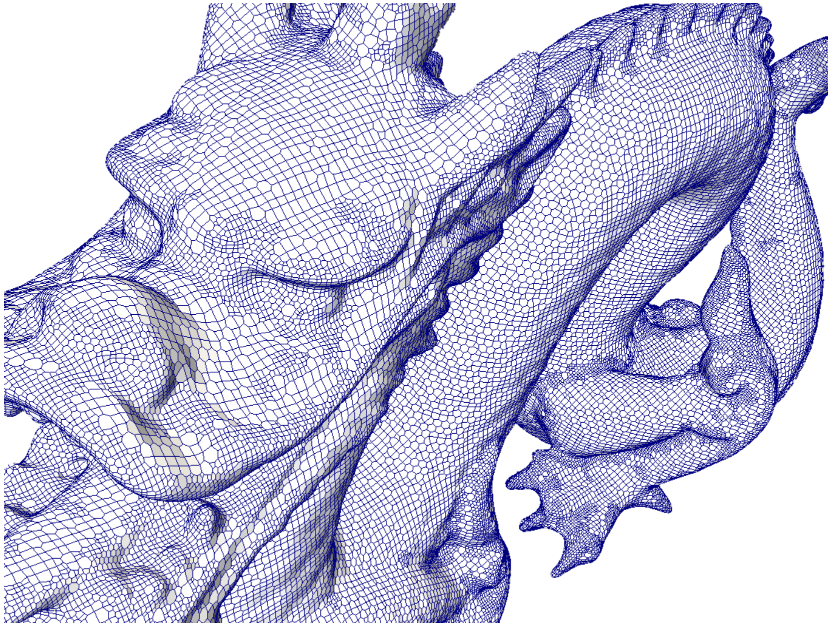
engys

Methodology Overview | Snapping

- Step 4: Snap mesh to surface
  - Implicit wrapping → Preserve features
  - Smooth & Merge faces



Original STL Surface

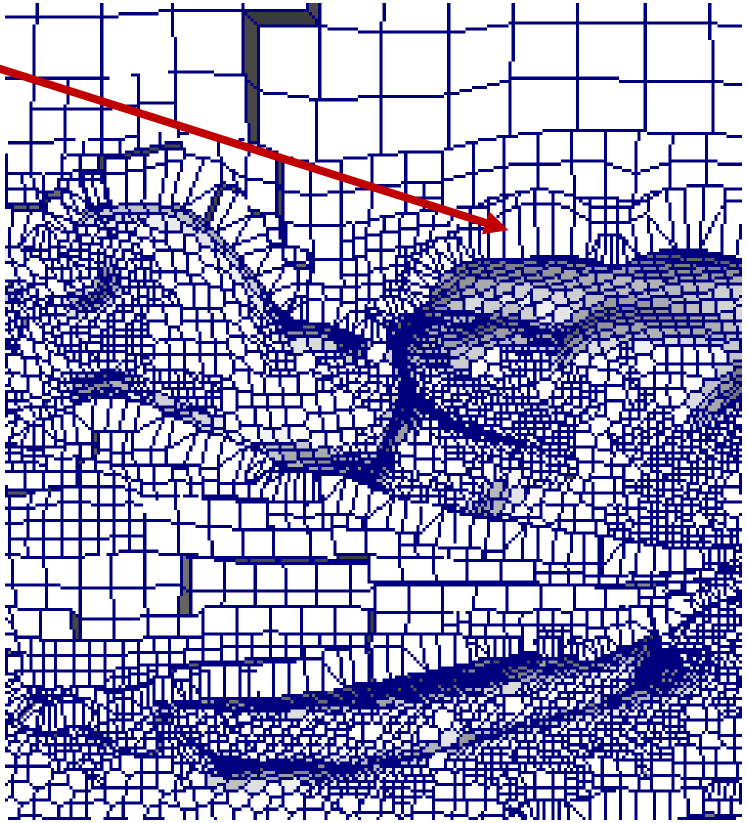


Snapped Surface

engys

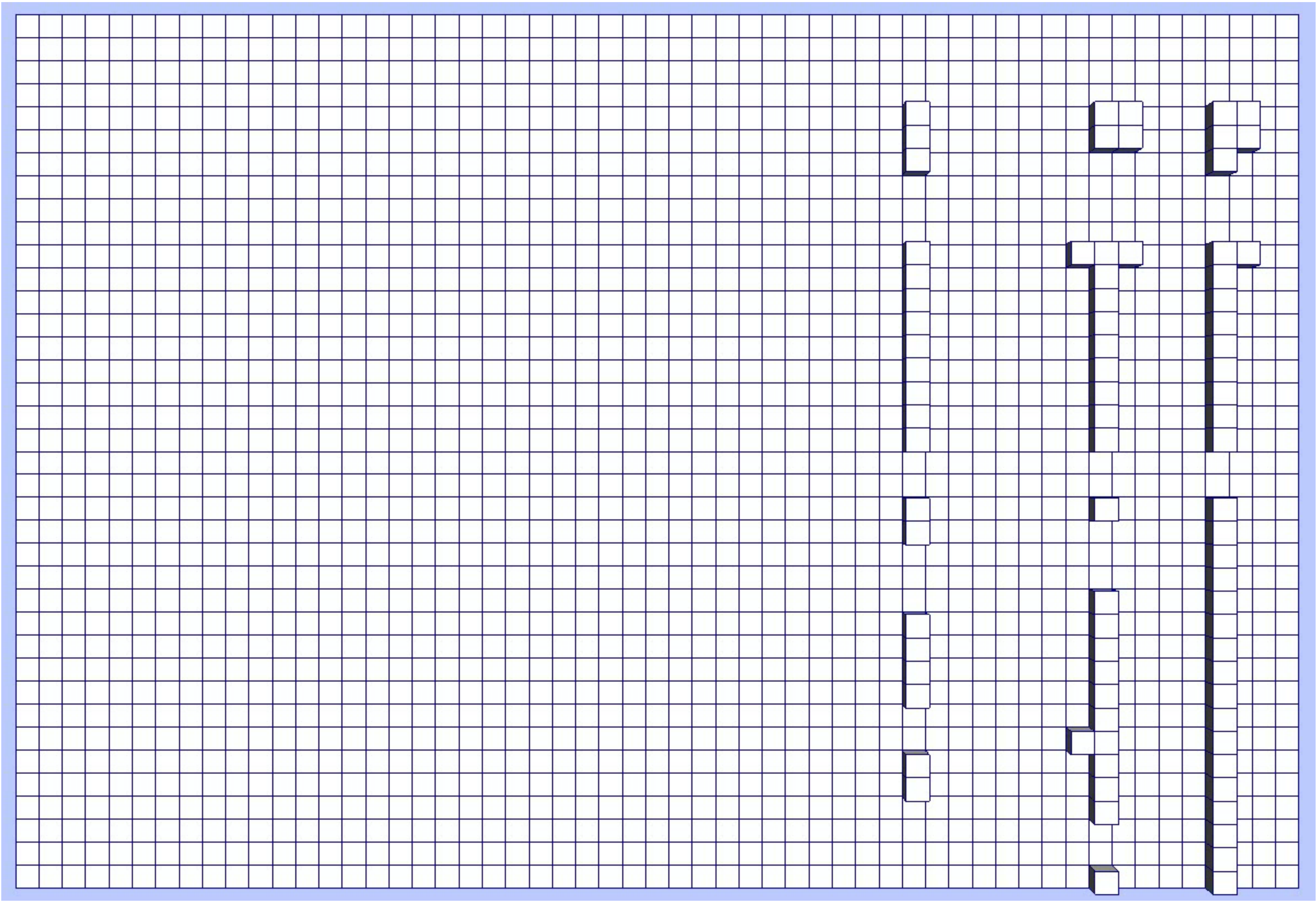
Methodology Overview | Layers + Final

- Step 5: Add layers to the surface
  - Push mesh away from surface
  - Add layers
  - Check quality
  - Scale back displacement if errors
  - Repeat until all quality checks pass



- Step 6: Final load balance
  - Output to file

Methodology Overview | Overall



Contents

- `snappyHexMesh` (40 Minutes)
  - Description
  - Background
  - Methodology Overview
  - HELYX-OS GUI
- Hands-On Meshing Examples (40 Minutes)
  - Initial Setup
  - Snapping
  - Layer Addition
- Closing Remarks (5 Minutes)

engys

HELYX-OS | Capability Overview

What is HELYX-OS?

An Open Source pre-processor for:

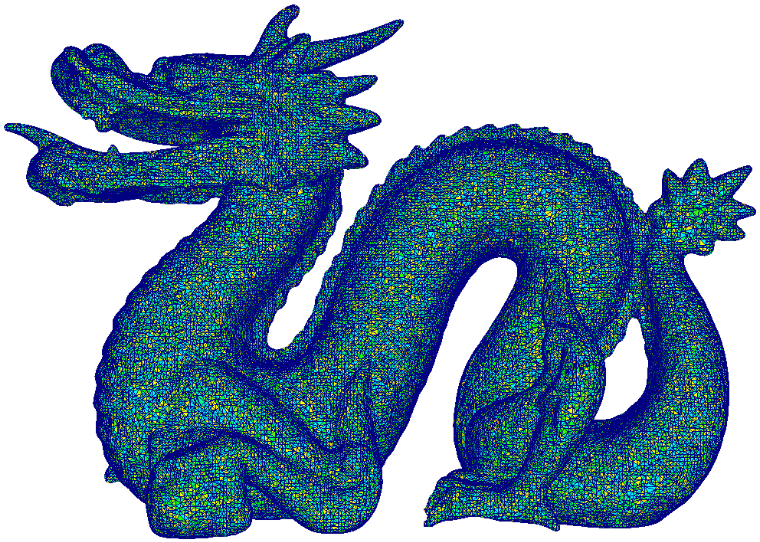
HELYX-OS | Capability Overview

What is HELYX-OS?

An Open Source pre-processor for:

- **Importing Geometries**
- Creating Meshes
- Configuring Cases
- Running Solvers

Geometries in stereolithography format (STL)

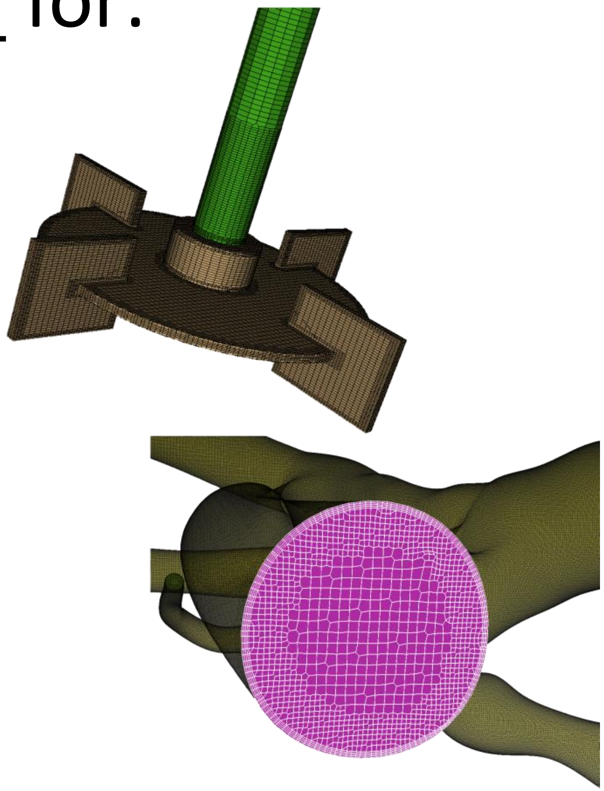


HELYX-OS | Capability Overview

What is HELYX-OS?

An Open Source pre-processor for:

- Importing Geometries
- **Creating Meshes**
- Configuring Cases
- Running Solvers



Using the `snappyHexMesh` utility

engys

HELYX-OS | Capability Overview

What is HELYX-OS?

An Open Source pre-processor for:

- Importing Geometries
- Creating Meshes
- **Configuring Cases**
- Running Solvers

Setup file structure and dictionaries

- Case Project Folder
  - log
  - constant
    - triSurface
  - system

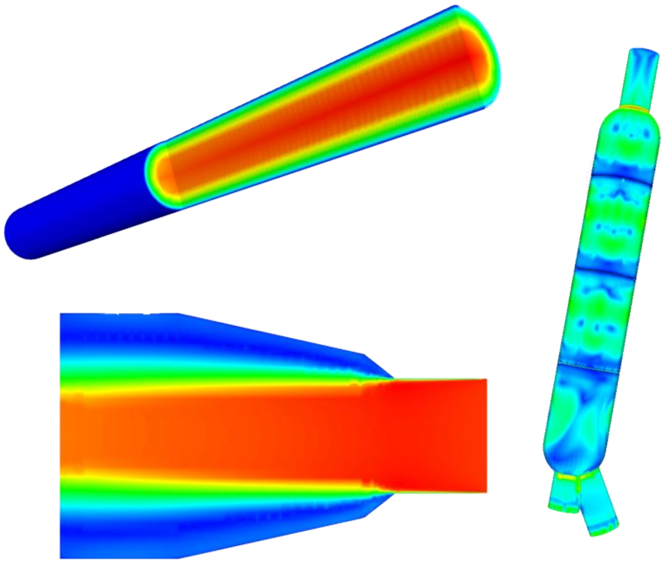
engys

HELYX-OS | Capability Overview

What is HELYX-OS?

An Open Source <u>pre-processor</u> for:

- Importing Geometries
- Creating Meshes
- Configuring Cases
- **Running Solvers**



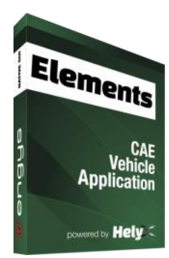
RANS, LES, Thermal, MRF, Porous…

HELYX-OS | Capability Overview

-Details and Compatibility-

- Written in Java, leveraging VTK
  - Provided in compiled 64 bit versions
- Currently in v1.0.2
  - Support for OpenFOAM® version 2.2.x
  - Available for free download on Sourceforge
- Derived from HELYX™ and Related to ELEMENTS™

- ✓ Enhanced Solvers + BCs
- ✓ Enhanced Meshing
- ✓ More Functionality
- ✓ Full User Support
- ✓ Documentation



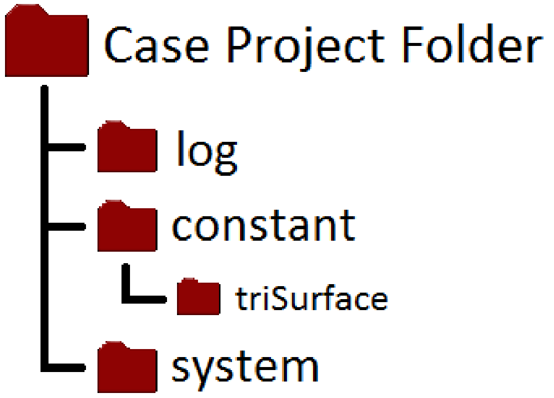
- ✓ Specifically for Automotive
- ✓ Meshing + simulation best practices
- ✓ Full Suite of tools
- ✓ Full User Support

HELYX-OS | Workflow Walkthrough

Create the Case

Meshing

- Parent folder definition and case directories are setup.
- Dictionaries are defined



HELYX-OS | Workflow Walkthrough

Create the Case

Define a Base Mesh

Meshing

- A blockMeshDict is created to provide a base mesh for `snappyHexMesh`

Create the Case

Define a Base Mesh

Advanced Settings

Meshing

- Geometries are imported
- Refinement surface and layer addition controls defined
- `snappyHexMesh` is setup and run

Contents

- `snappyHexMesh` (40 Minutes)
  - Description
  - Background
  - Methodology Overview
  - HELYX-OS GUI
- Hands-On Meshing Examples (40 Minutes)
  - Initial Setup
  - Snapping
  - Layer Addition
- Closing Remarks (5 Minutes)

Hands-On Session | Initial Setup

Overall Goal of Session

- To familiarize users with configuring and meshing a given STL surface in HELYX-OS
- Learn some best best-practices when using **`snappyHexMesh`**

Skills Obtained

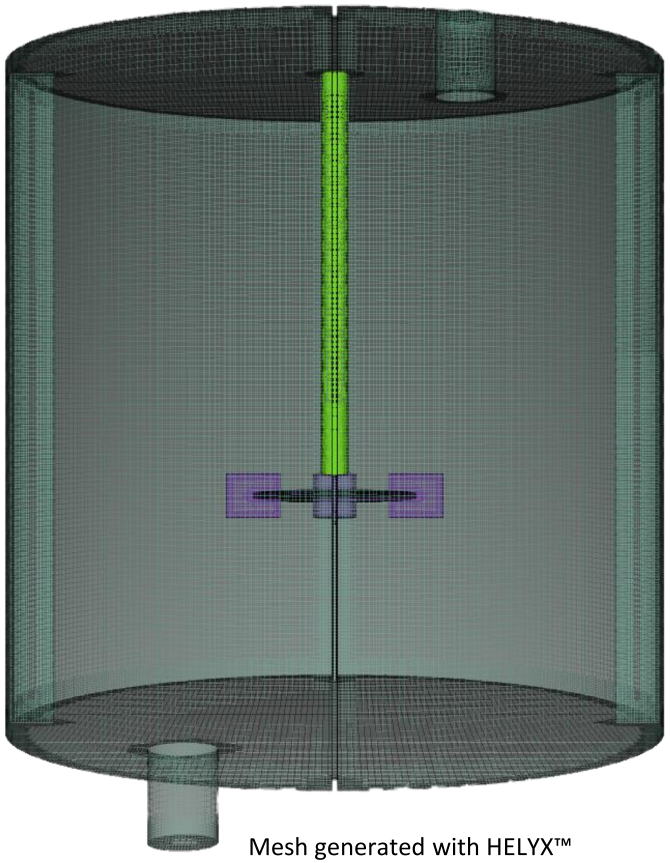
✓ Importing STL surfaces into HELYX-OS

✓ Defining and meshing a specific region of the domain

Hands-On Session | Initial Setup

Continuously Stirred Tank Reactor (CSTR)

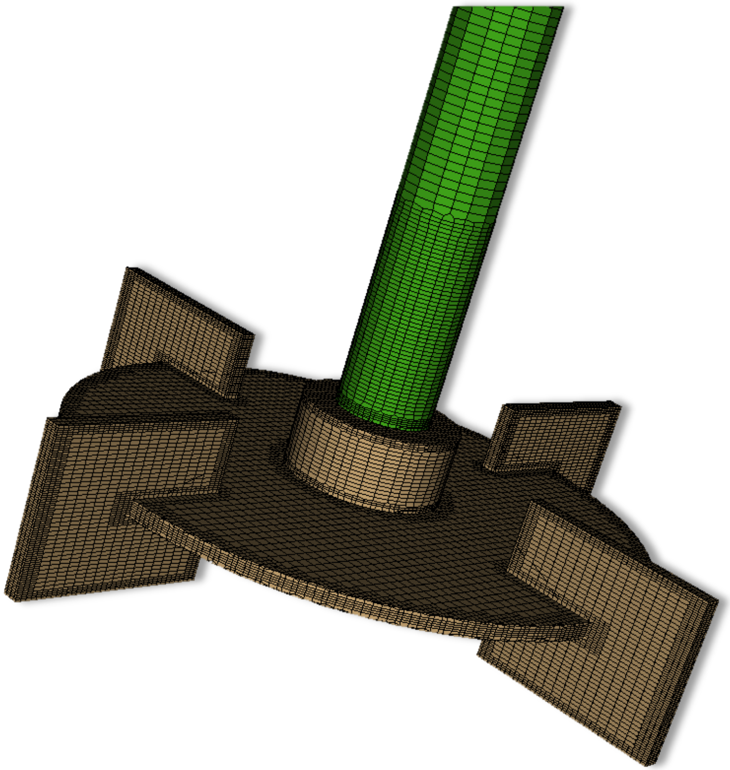
- Comprised of vessel with baffles, impeller, shaft, inlet, and outlet
- Classical chemical engineering problem
- Can be run in batch mode by setting a zero inflow velocity
- Steady-states single phase systems can be modeled with a multiple reference frame solver (MRF)
- Example could use **`simpleFoam`** with MRF fvOptions functionality

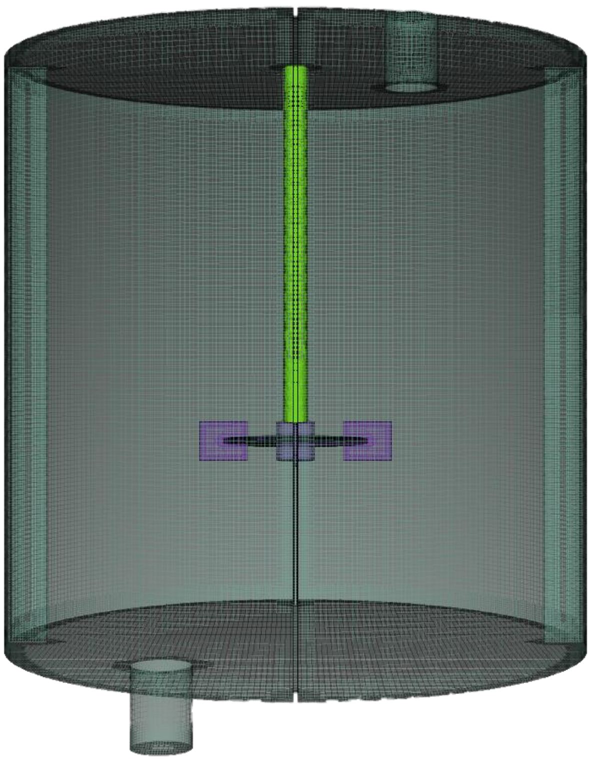


Mesh generated with HELYX™

engys

Hands-On Session | Initial Setup



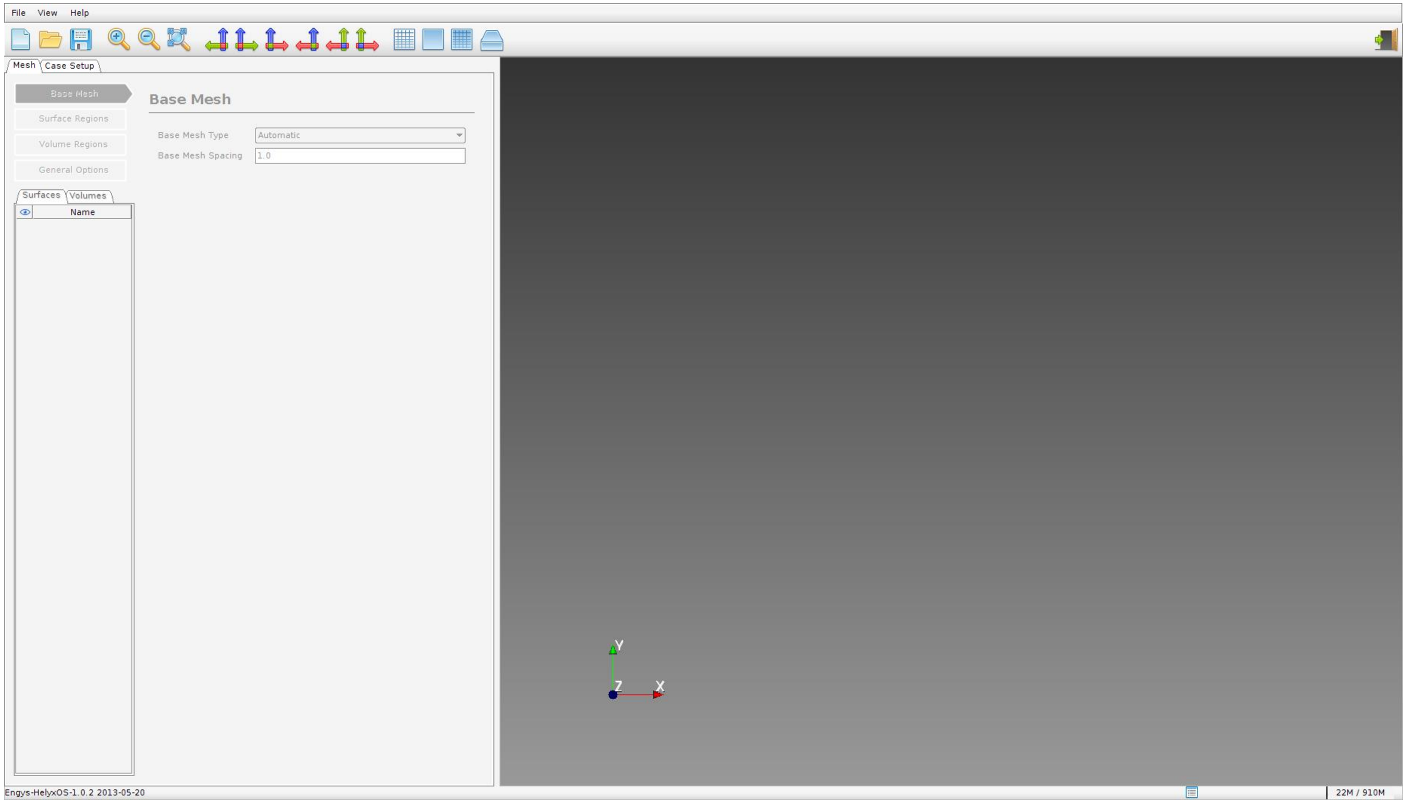


- Underlying geometries were constructed in blender
- Each patch was separated, and exported as an STL file

engys

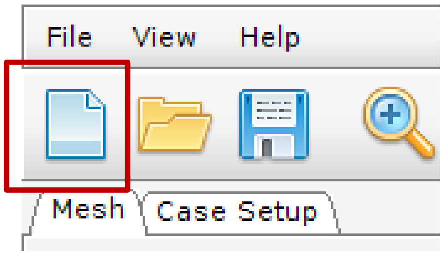
Hands-On Session | Initial Setup

1. **Open a terminal (control + alt + t)**
2. **Type `helyxOS.sh` in the shell command line**



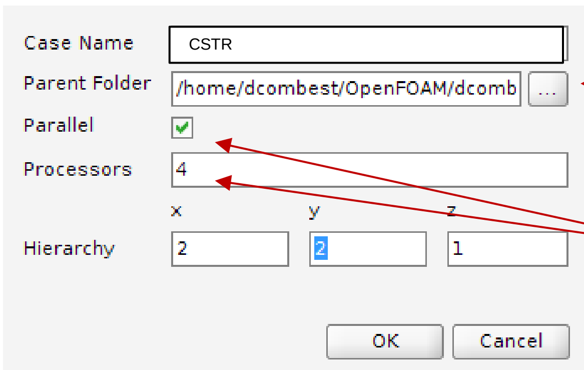
engys

Hands-On Session | Initial Setup



3. Click the "blank paper" icon to create a new case

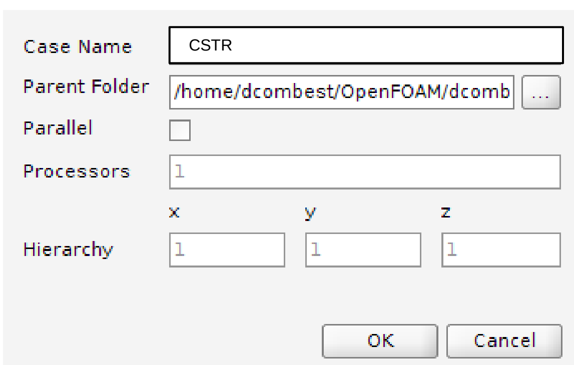
Parallel Cases



Select the parent folder for the cases.

Possible to select "parallel", number of processors, and simple decomposition parameters

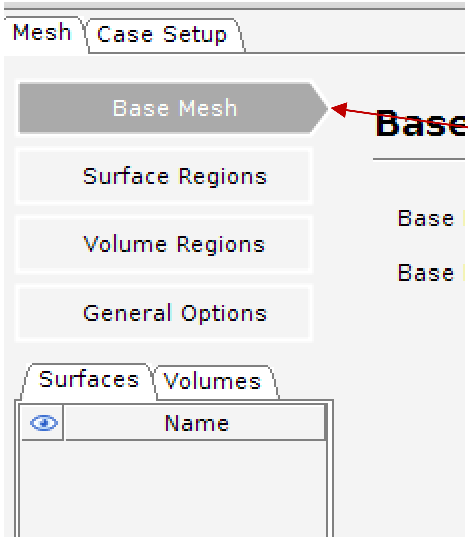
Serial Cases



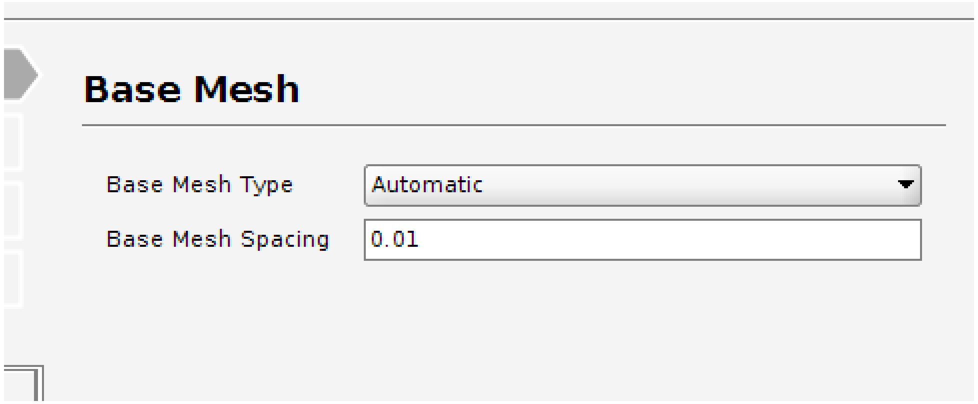
4. Call the case "CSTR" and select a Parent folder.

5. Leave the case as serial and Click "OK"

Hands-On Session | Initial Setup

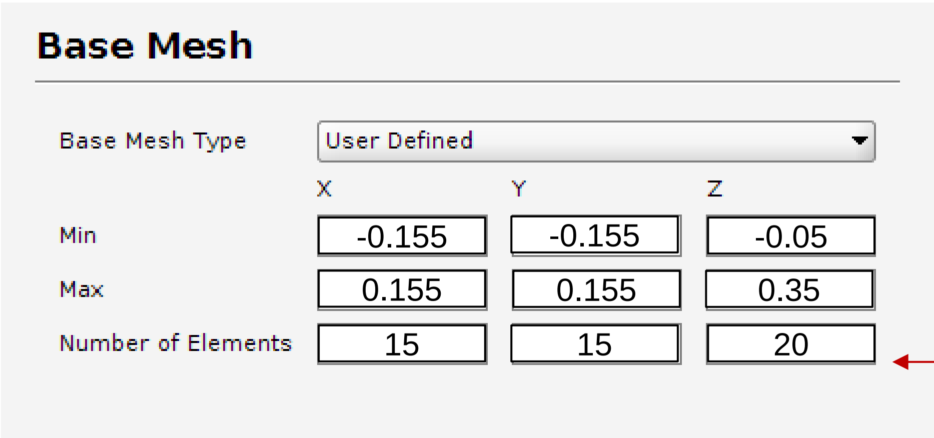


The "Base Mesh" tab is where we define the base mesh used by snappyHexMesh to create a geometry and is selected by default when a case is defined



By default, the method is set to automatic and HELYX-OS generates a blockMeshDict for you.

Hands-On Session | Initial Setup



You may also define a base mesh bounding box with intervals in the x, y, and z-direction.

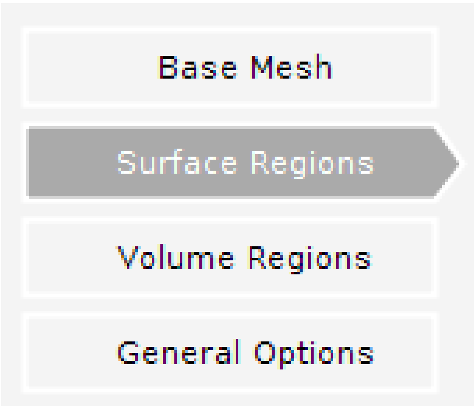
6. Select "User Defined" and enter these setting for this case

- Ideally, create a mesh where cells are perfect cubes i.e.

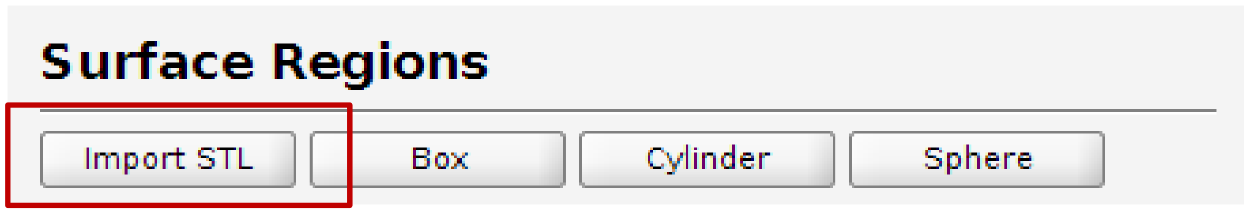
$$\frac{|maxX - minX|}{n_x} = \frac{|maxY - minY|}{n_y} = \frac{|maxZ - minZ|}{n_z}$$

snappyHexMesh will yield better meshes

Hands-On Session | Initial Setup



7. Select the "Surface Regions" tab

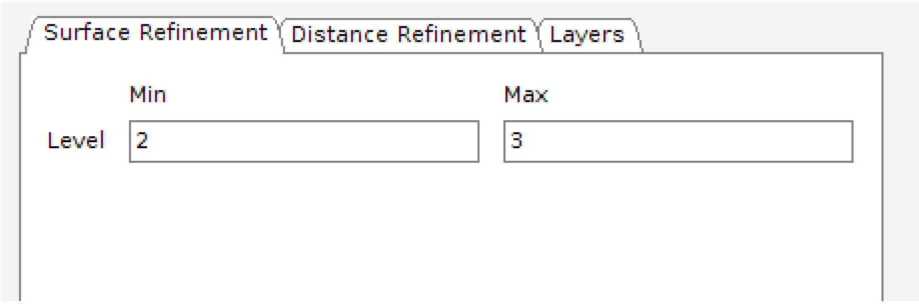


We are interested in creating a pipe geometry and internal mesh. **8. Select the "Import STL" primitive shape**

9. Navigate to the geometry directory and load the CSTR.stl geometry

The geometry should appear in the window.

Hands-On Session | Initial Setup

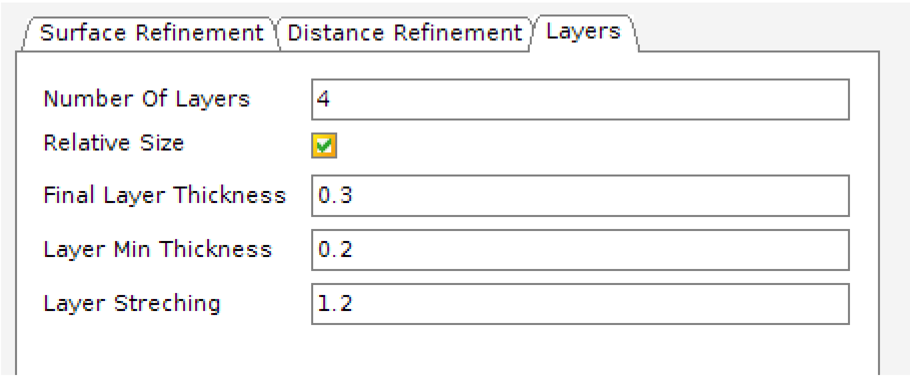


Below the "Surface Regions" definition for the CSTR, is the "surface refinement" tab. This defines how much refinement is performed with respect to the base mesh and "feature angle". A higher number results in a finer mesh around a feature.

10. Enter min and max level refinement for the patches

Patches	Min Level	Max Level
Impeller	2	3
Inlet	1	2
Outlet	1	2
Shaft	1	2
Vessel	2	3

Hands-On Session | Initial Setup



11. Select the "Layers" subtab in surface region

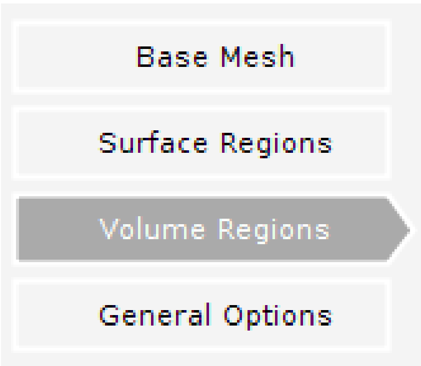
This will allow us to define the number of surface layers to add and some characteristics of the layers.

12. Enter exactly what is shown in the table below.

Surface	# of Layers	Relative?	Final Layer Thickness	Min Thickness	Layer Stretching
Impeller	4	Yes	1.6	0.2	1.2
Vessel	4	yes	1.6	0.2	1.2

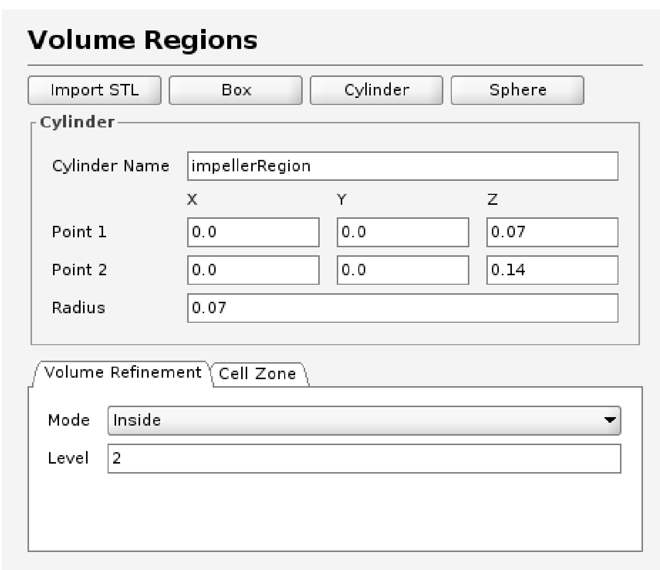
Layer addition is **exaggerated in this example**, the default values are good values to use for first approximations

Hands-On Session | Initial Setup



The "Volume Regions" tab is where we can define refinement regions and cellZones

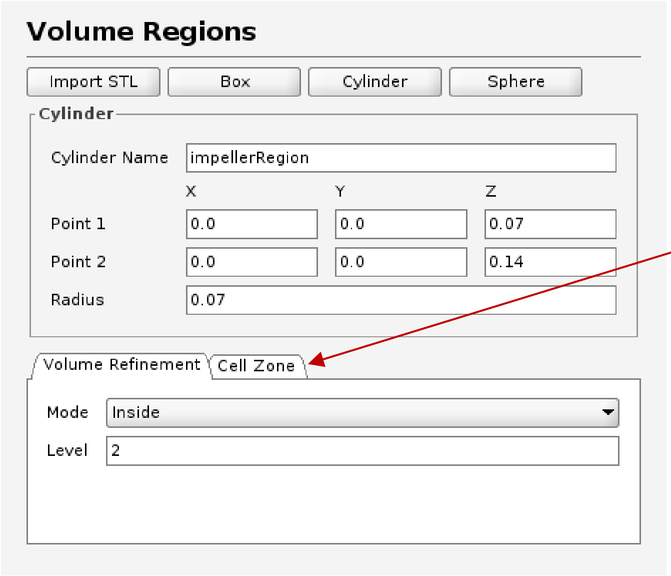
13. Select Volume Regions



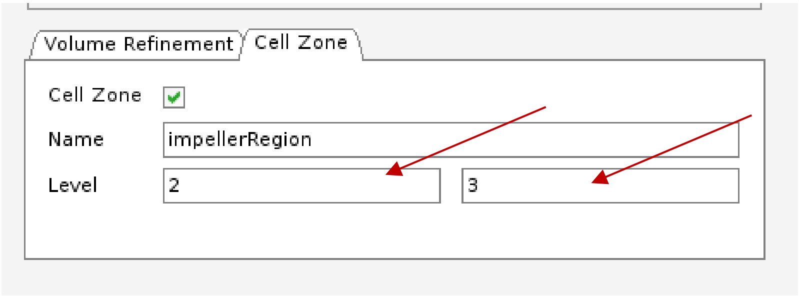
We may want to create a cellZone to be used for MRF and porous flows. This can easily be done with a volume region

14. Select "cylinder" volume region and name the cylinder "impellerRegion"
15. Enter in size and orientation information
16. In the volume refinement subtab, set mode to "inside" and the level to 2

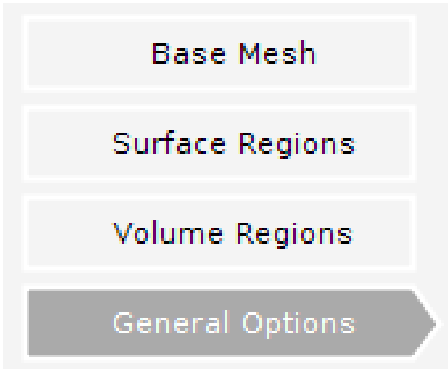
Hands-On Session | Initial Setup



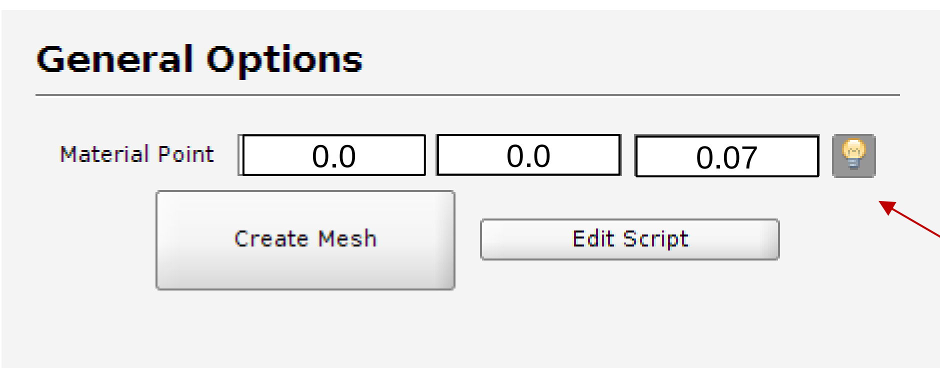
17. Select the "cell zone" subtab
18. Check "Cell Zone" box and name it "impellerRegion"
19. Set minimum and maximum levels to 2 and 3



Hands-On Session | Initial Setup



20. Select the "General Options" tab



We must define a point where a mesh cell will exist. To do this, we need to select a point in space inside the pipe.

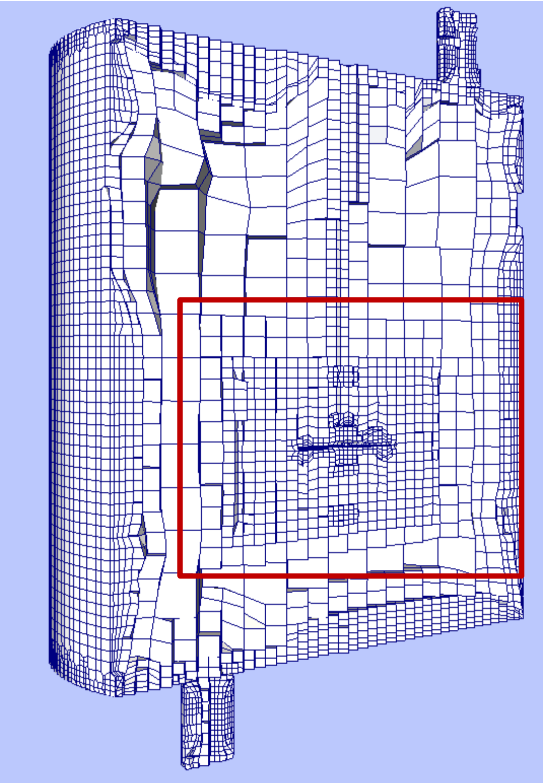
Use the light bulb icon to visualize the current point

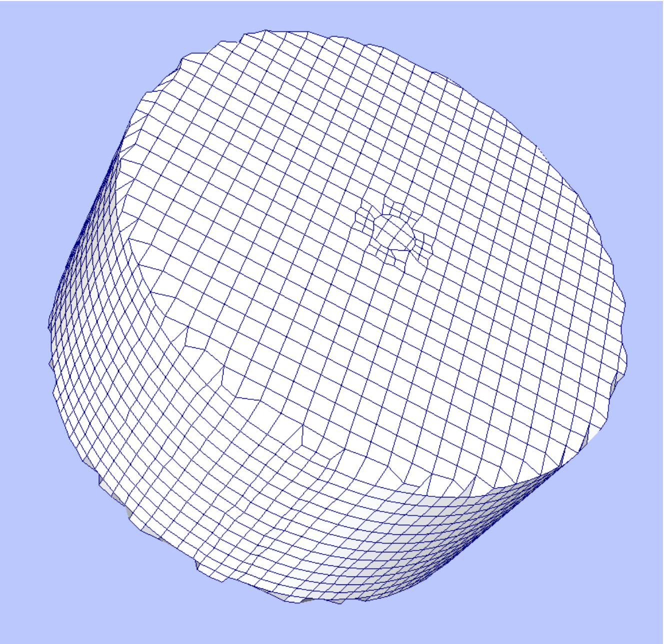
21. Enter (0, 0, 0.07) and hit "Create Mesh"

engys

Hands-On Session | Initial Setup

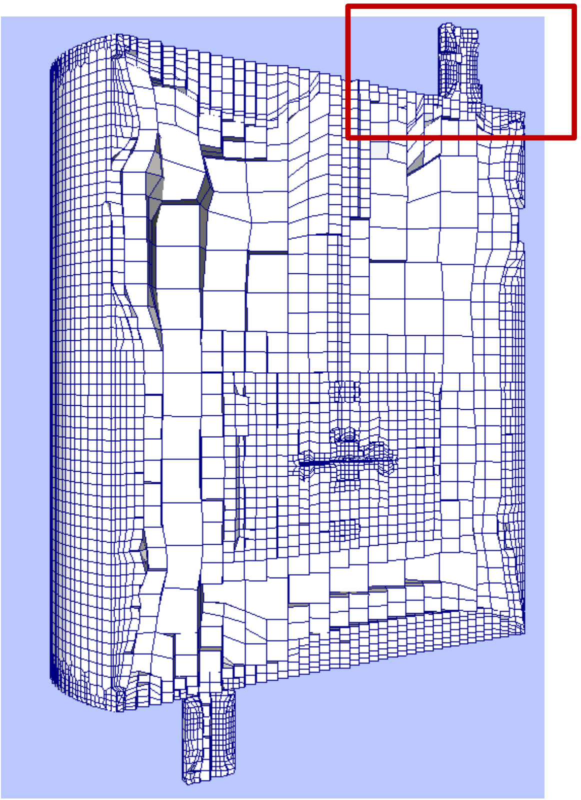
23. **While the mesher is running, navigate to the case directory. When it is finished launch Paraview with `ParaFoam`**
24. **Select the internal mesh and then Filters>Alphabetical>Extract Cells by Region with the following settings.**

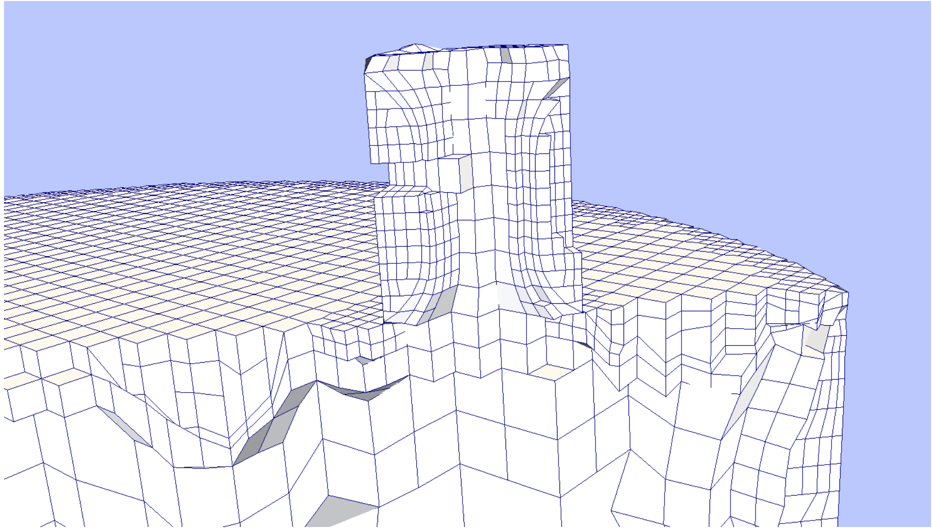




Hands-On Session | Initial Setup

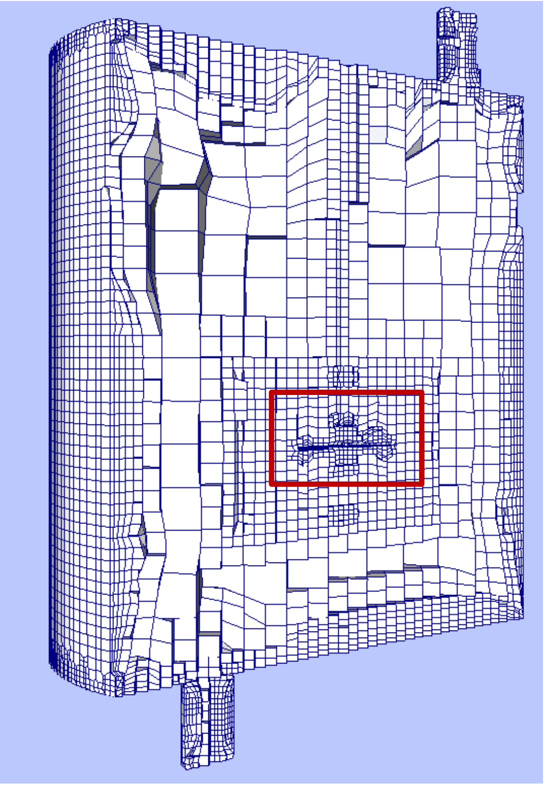
23. While the mesher is running, navigate to the case directory. When it is finished launch Paraview with `ParaFoam`
24. Select the internal mesh and then Filters>Alphabetical>Extract Cells by Region with the following settings.

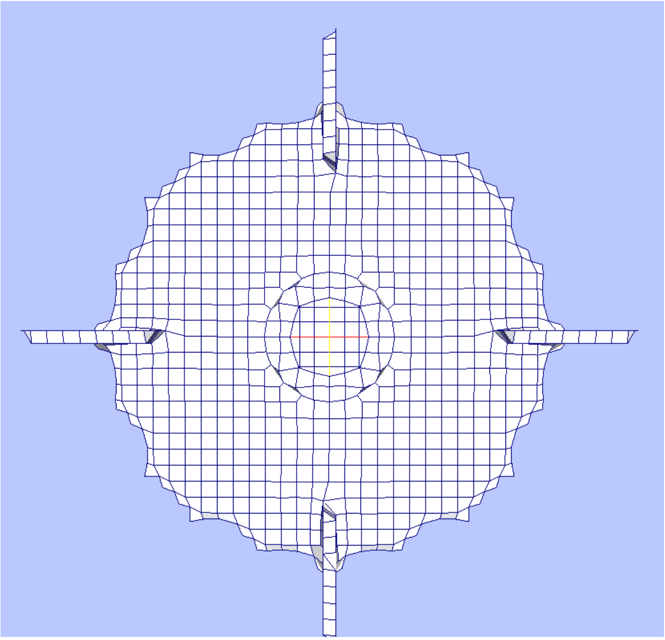




Hands-On Session | Initial Setup

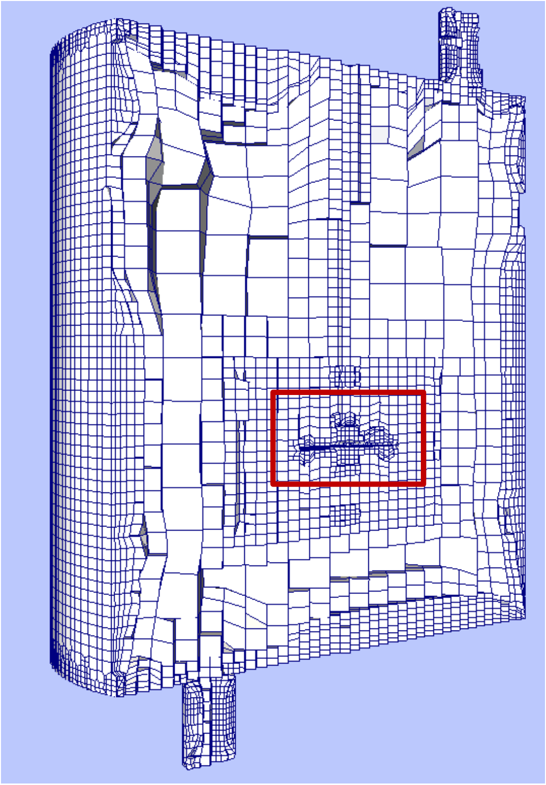
23. **While the mesher is running, navigate to the case directory. When it is finished launch Paraview with `ParaFoam`**
24. **Select the internal mesh and then Filters>Alphabetical>Extract Cells by Region with the following settings.**

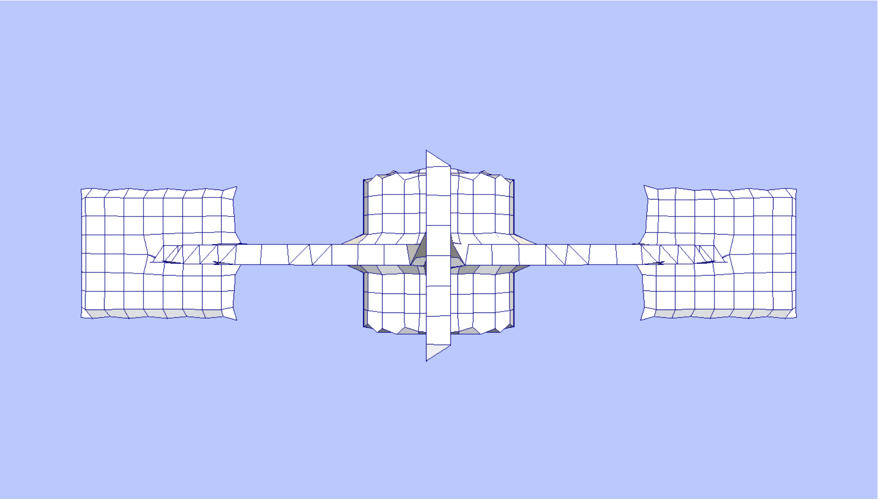




Hands-On Session | Initial Setup

23. While the mesher is running, navigate to the case directory. When it is finished launch Paraview with `ParaFoam`
24. Select the internal mesh and then Filters>Alphabetical>Extract Cells by Region with the following settings.





engys

Hands-On Session | Initial Setup

Goals by the end of training

- Snap to the impeller more
- Capture surfaces in the CSTR
- Create better boundary layer meshes

Important Points to Remember

- This domain is well behaved with relatively smooth surfaces
- Domain is aligned with Cartesian coordinate system
- Mesh is very coarse compared to what is needed for a solution
- We are going to study each part separately to save effort

Hands-On Session | Snapping

Overall Goal of Session

- Make some adjustments to further capture feature edged
- Learn strategies to improve snapping

Skills Obtained

✓ Learning how to set a base mesh around portions

✓ Extracting eMesh files and using them

✓ Troubleshooting the snapping step

Hands-On Session | Snapping

How do we improve snapping?

✓ Lower resolveFeatureAngles?
- Already set to 30, and will resolve angles over 30

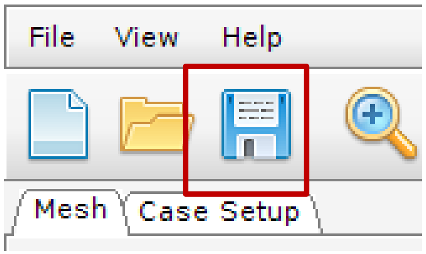
✓ Use eMesh Files?
- Will not have much of an affect at this size of level 0

✓ Increase min and max levels on surfaces?
- May work if you are satisfied with the base mesh level 0 value

✓ Smaller base mesh?
- If there is room to decrease level 0 cell size
- May help boundary layer addition

✓ Use a refinement box to create smaller cells near surface?

Hands-On Session | Snapping



Clone the existing CSTR case

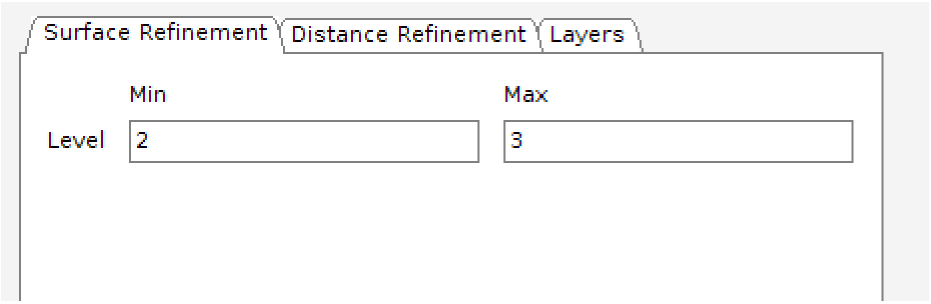
1. Click the "disk" icon to save the CSTR case, but create a new folder outside CSTR named "impeller" and save the current case in that folder

2. Create a user defined base mesh with the same level 0 spacing as our original CSTR mesh using the values below.

	X- Direction	Y-Direction	Z-Direction
Min Value	-0.07	-0.07	0.07
Max Value	0.07	0.07	0.14
# of Elements	7	7	3

Narrower base mesh will isolate the impeller

Hands-On Session | Snapping



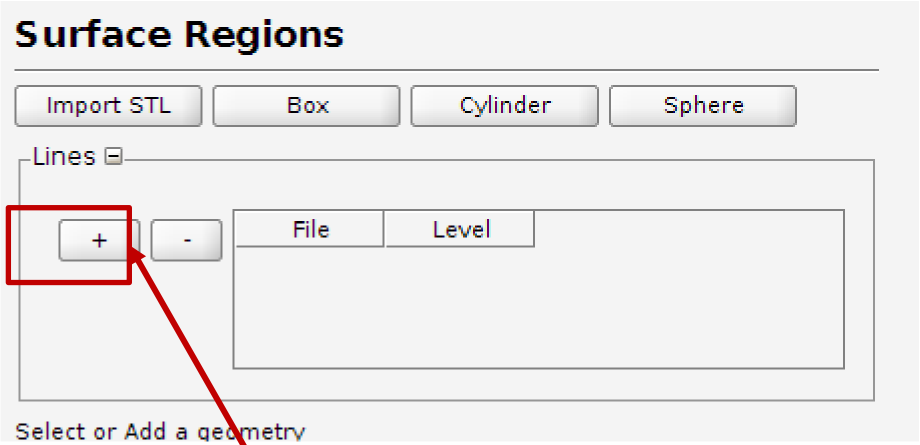
Below the "Surface Regions" definition for the CSTR, is the "surface refinement" tab. This defines how much refinement is performed with respect to the base mesh and "feature angle". A higher number results in a finer mesh around a feature.

3. Enter min and max level refinement for the patches

Patches	Min Level	Max Level
Impeller	2	3
Shaft	1	2

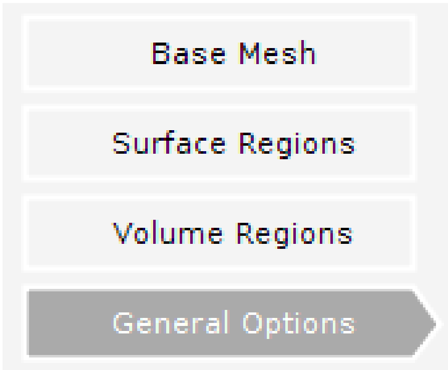
Hands-On Session | Snapping

4. Make sure you have patched your installation of HELYX-OS
5. Double check you have patched your installation of HELYX-OS

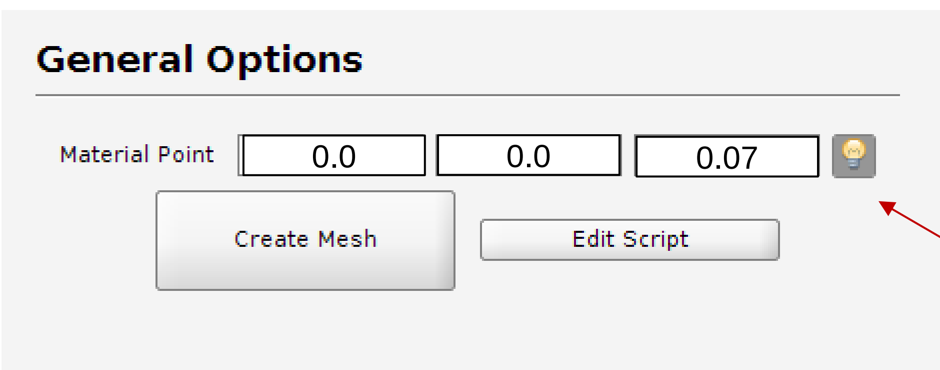


6. Click the "+" button
7. Navigate to geometry folder
8. Load the CSTR.eMesh file and click "Open"
9. Set the refinement level to 2 for edge

Hands-On Session | Snapping



10. Select the "General Options" tab



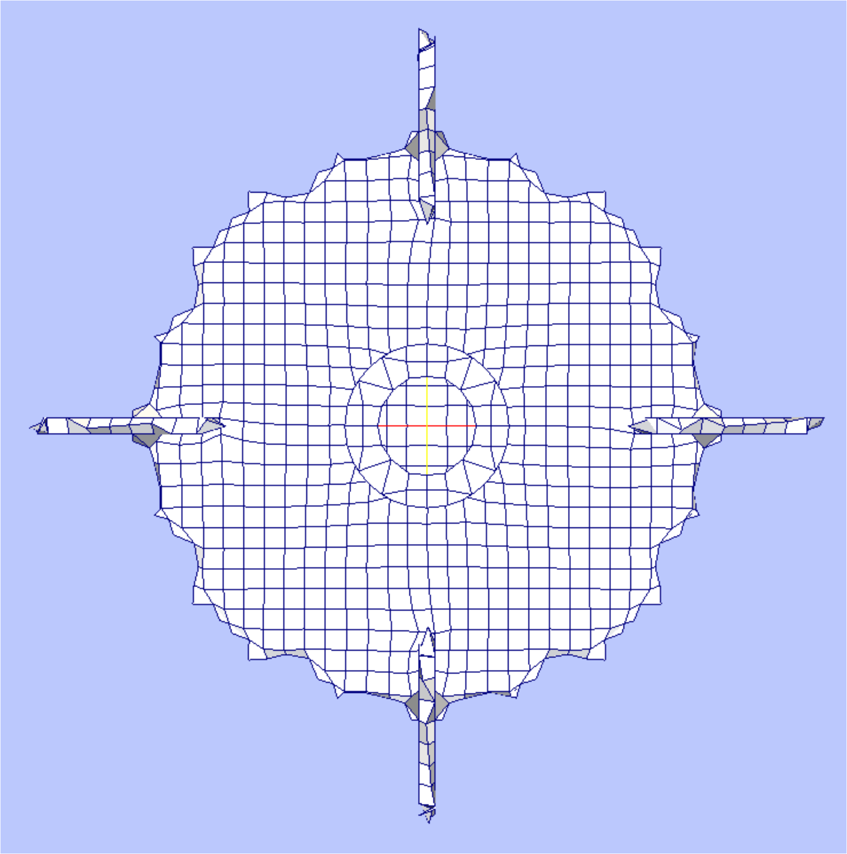
Use the light bulb icon to visualize the current point

11. Enter (0, 0.0, 0.07) and hit "Create Mesh"

engys

Hands-On Session | Snapping

12. Open paraFoam and select the impeller patch only.



The eMesh alone will not resolve the surface

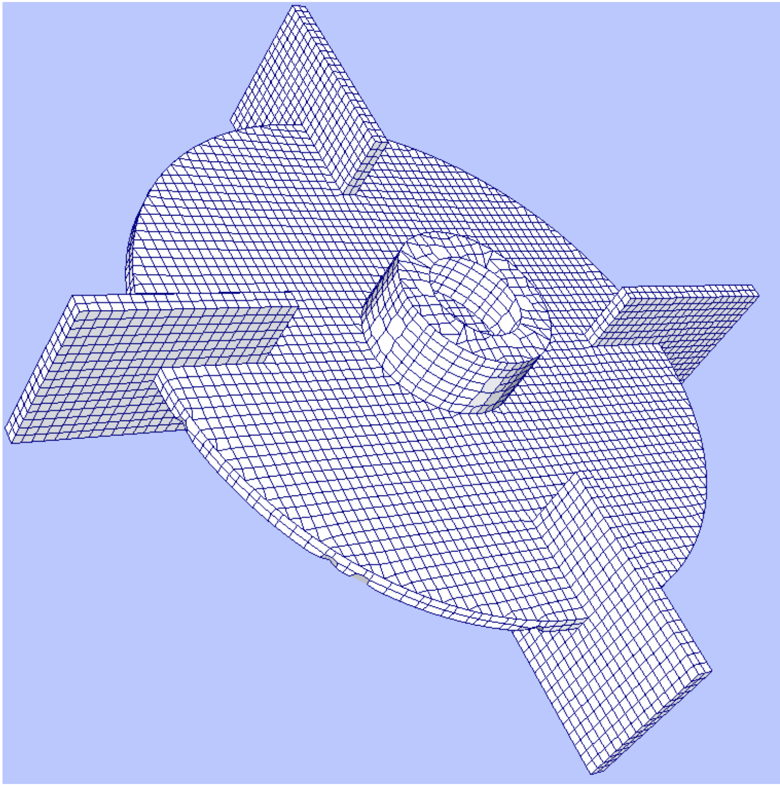
Hands-On Session | Snapping

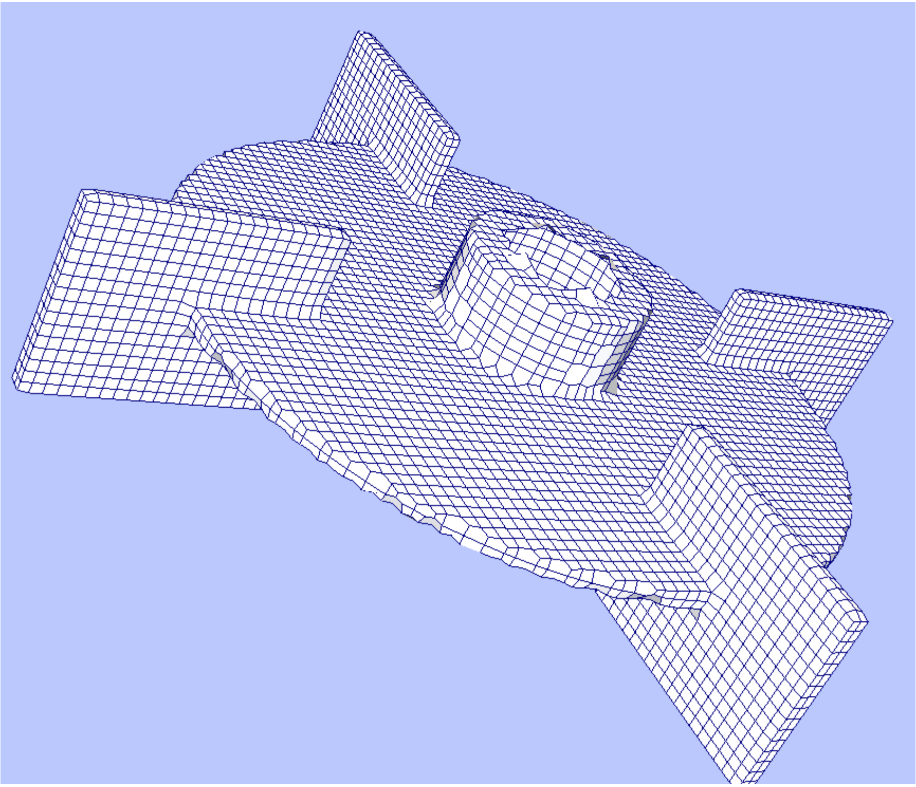
13. Increase the base mesh # of elements to

	X- Direction	Y-Direction	Z-Direction
Min Value	-0.07	-0.07	0.07
Max Value	0.07	0.07	0.14
# of Elements	70	70	30

14. Remesh the domain
15. Open paraFoam when the case is finished meshing
16. Look only at the impeller patch with “surface and edges” selected

Hands-On Session | Snapping





eMesh and Decreased 0 level
- Sharp edges near fully resolved

Decreased 0 level Only
- Sharp edges are rounded off

Hands-On Session | Layer Addition

Overall Goal of Session

- Improve understanding of layer addition process
- Learn strategies to improve layer addition

Skills Obtained

✓ Generate layers more reliably

✓ More fine-tuning of layer additions

✓ Troubleshooting the layer addition

engys

Hands-On Session | Layer Addition

How can we improve layer addition?

✓Decrease base mesh size?

- Will need to capture surfaces

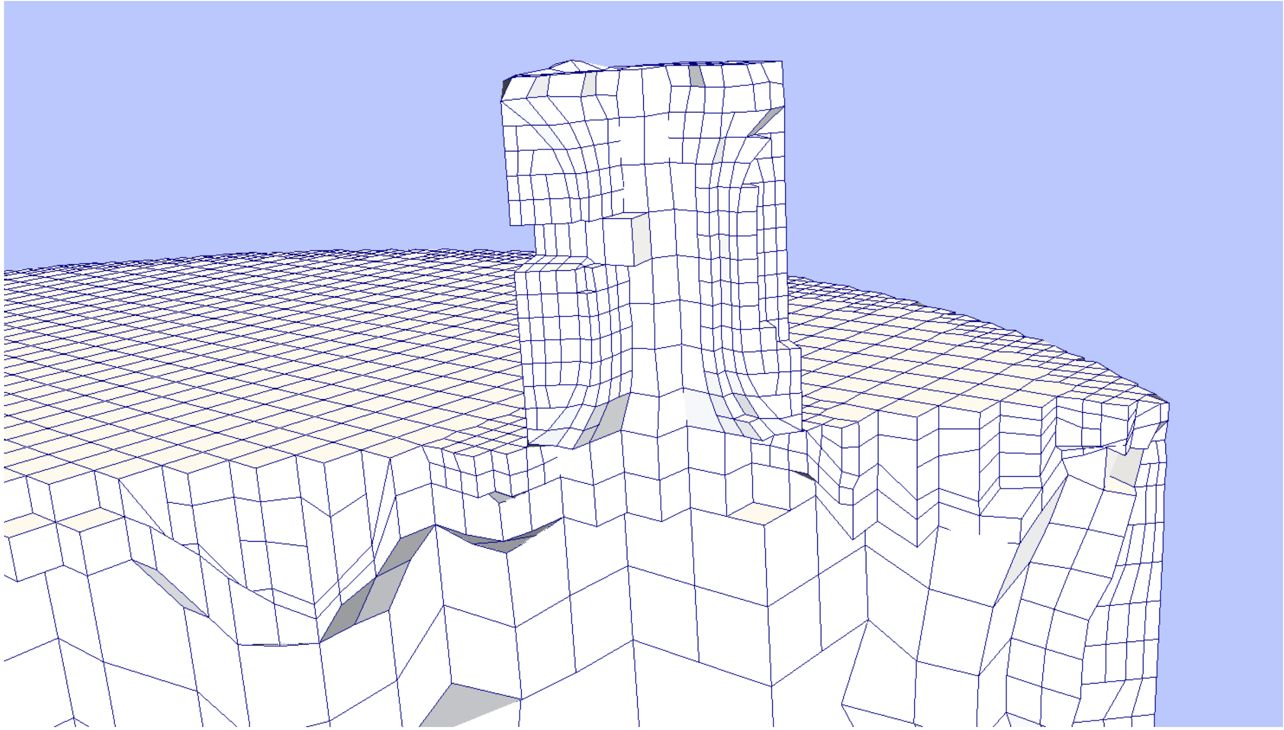
✓Change layer addition options?

- Create smaller BL mesh near surface
- Use relative instead of absolute distances
- Slowly increase number or size to what you want

✓Create more uniform cell size near patches?

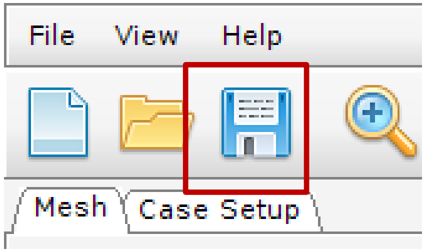
- Use distance refinement to create uniform cell-level near patches

Hands-On Session | Layer Addition



- Layer addition settings were too aggressive
- Base mesh was too coarse

Hands-On Session | Layer Addition

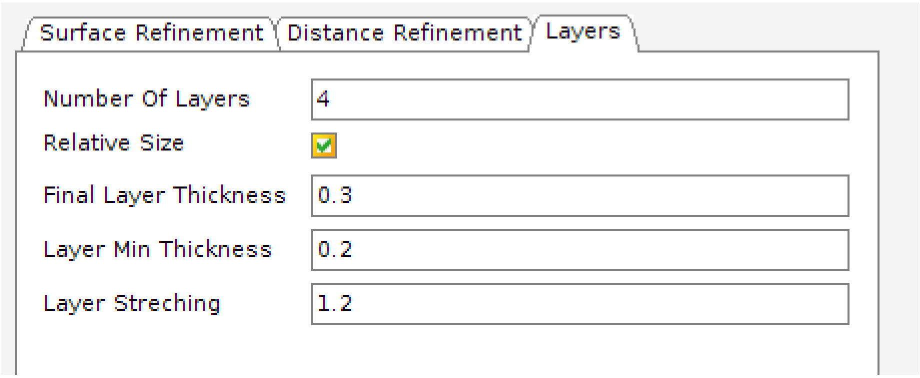


1. **Click the "disk" icon to save the CSTR case, but create a new folder outside CSTR and save the current case**

2. **Create a user defined base mesh with the same level 0 spacing as our original CSTR mesh using the values below.**

	X- Direction	Y-Direction	Z-Direction
Min Value	0.03	0.03	0.27
Max Value	0.1	0.1	0.35
# of Elements	30	30	40

Hands-On Session | Layer Addition

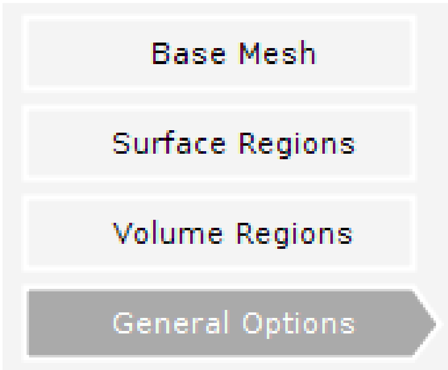


3. Select the "Layers" subtab

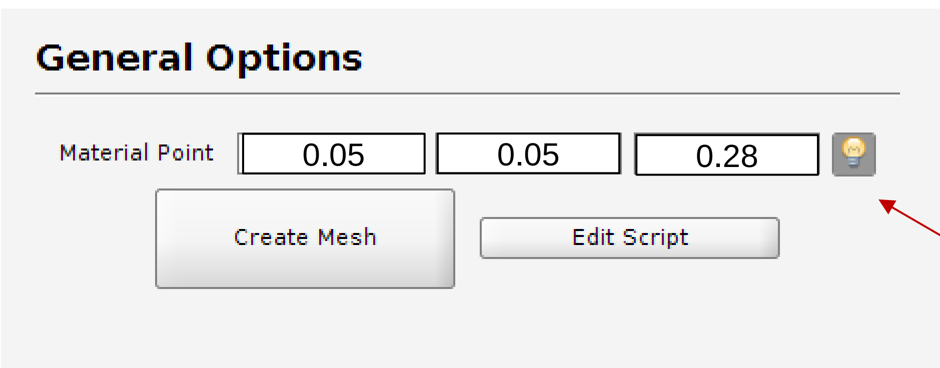
This will allow us to define the number of surface layers to add and some characteristics of the layers.

4. Enter exactly what is shown on the left for the "vessel" patch

Hands-On Session | Layer Addition



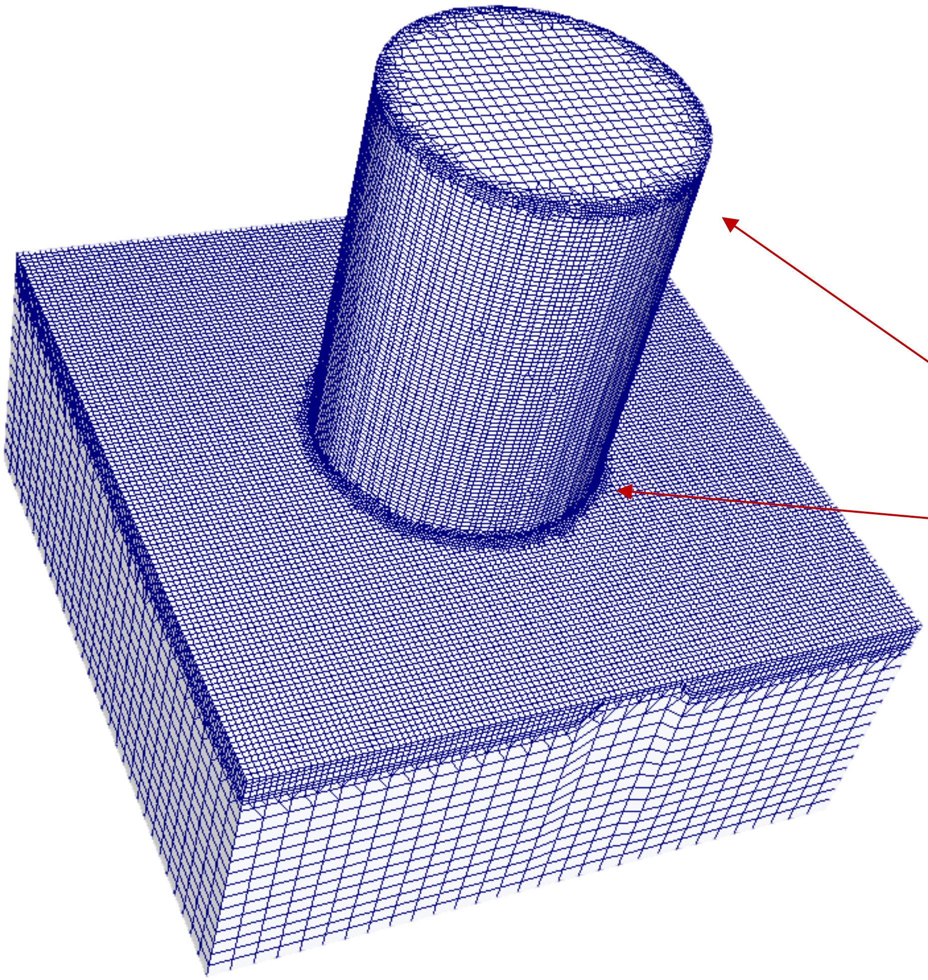
5. Select the "General Options" tab



Use the light bulb icon to visualize the current point

6. Enter (0.05, 0.05, 0.28) and hit "Create Mesh"
7. Open paraFoam when then mesh is finished

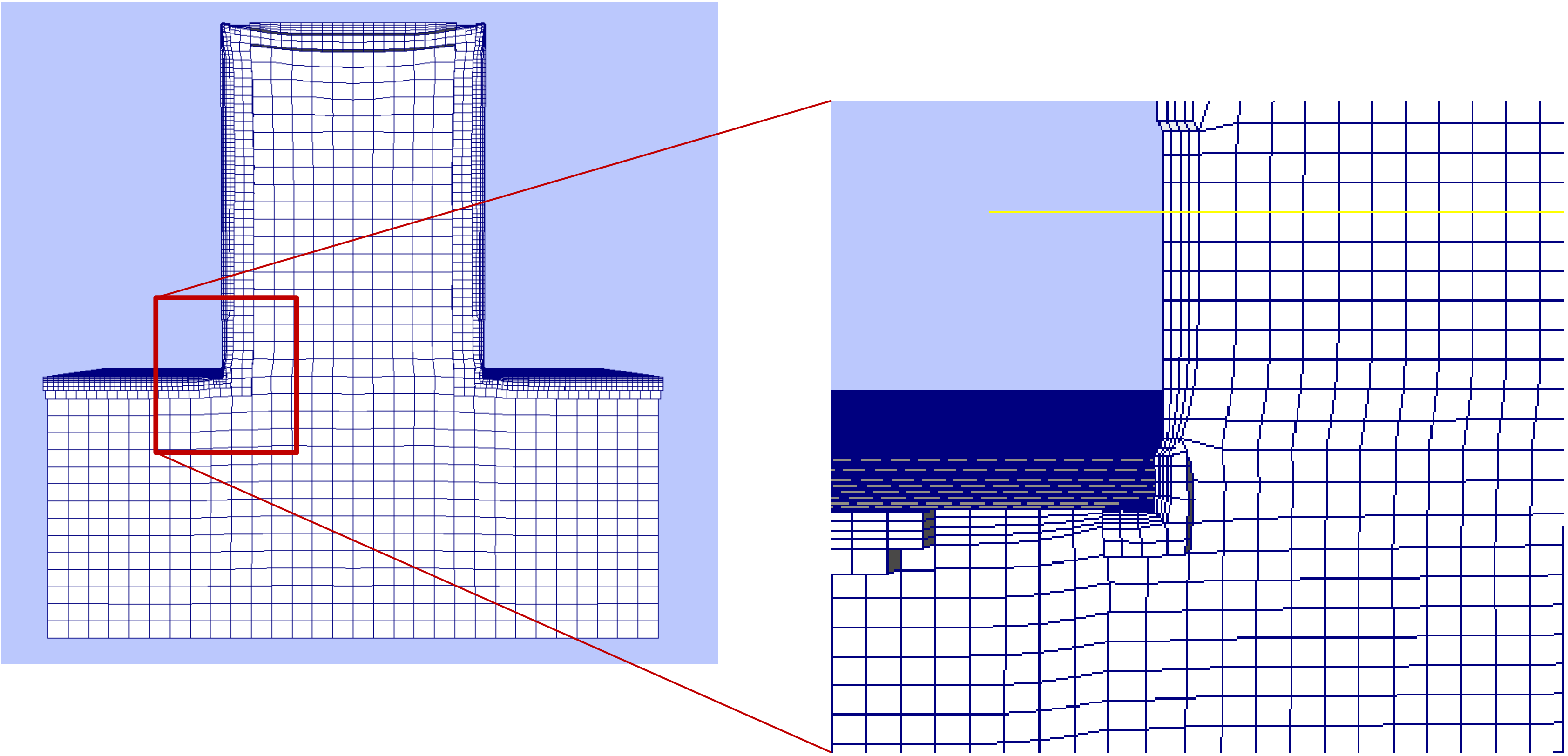
Hands-On Session | Layer Addition



Nice resolution of the edges and

8. Use the “extract cells by region” filter and “surface and edges” selected

Hands-On Session | Layer Addition



We have better boundary layers, but because we used relative layer measures and there are differences in cell level around edges, we have various sizes of layer cells
We want more uniform BL cells for a more uniform y+

Hands-On Session | Layer Addition

9. Go back to HELYX-OS and navigate to the “Surfaces Regions” tab

10. With CSTR selected, Select the “Distance Refinement” subtab

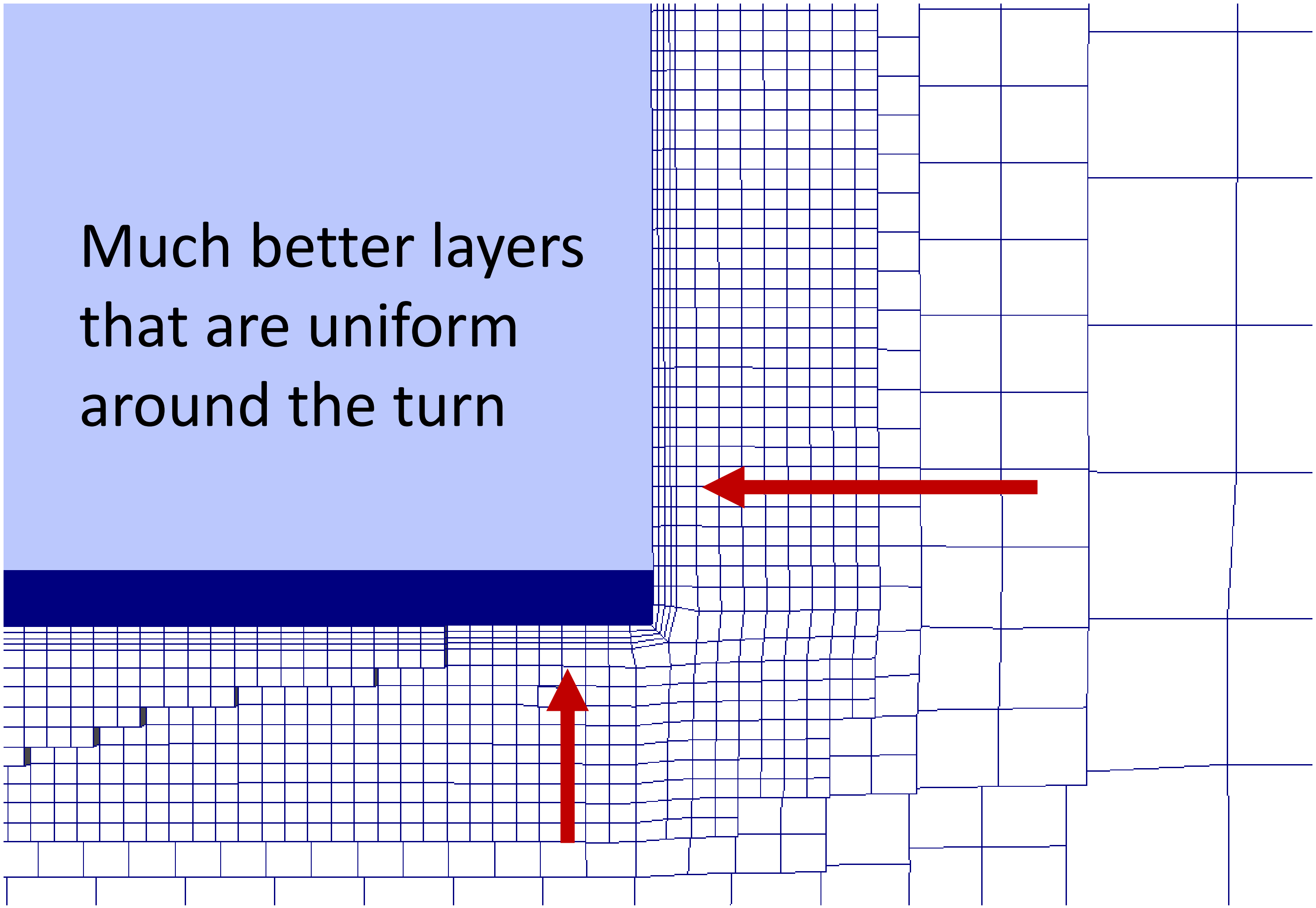
This will allow us to define the a refinement level with regard to a distance from a surface

11. Enter the values 0.002 and 3, setting a refinement level of 3 within 0.002 meters from the surface.

12. Go to the “General Options” tab and create the mesh again

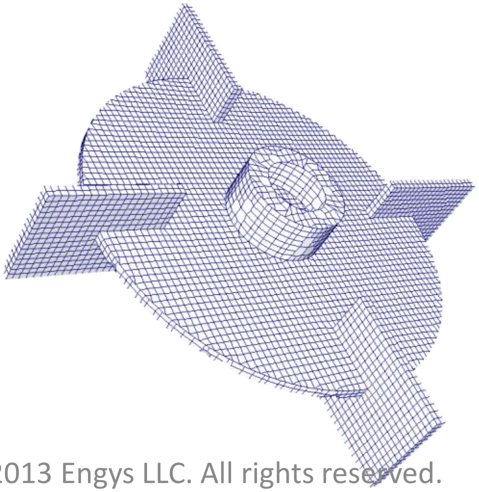
11. Open paraFoam when the meshing finishes and extract the cells by region for the last time

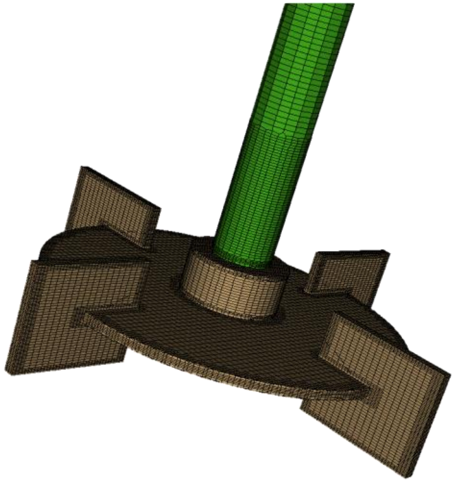
Hands-On Session | Layer Addition



Closure

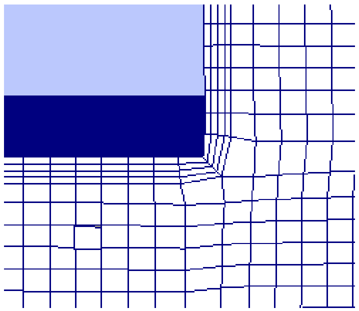
- Overall
  - ✓ Coordinate system aligned surface are easier to mesh
  - ✓ It is best not to change the meshQuality settings too much
  - ✓ Wedges can be difficult since cells are collapsed in narrow regions
  - ✓ Distance refinement is a per STL surface feature and not a per patch feature.
    - Get around this by importing separate STL files for each patch





Closure

- Snapping
  - ✓ Increasing min and max for a particular surface has the same affect as decreasing level 0 size
  - ✓ eMeshes can help fully resolve edges, only if base mesh is sufficiently fine
  - ✓ We can further increase snapping if we increase snapping iterations see appendix
- Layer Addition
  - ✓ Thinner layers are easier to insert
  - ✓ Use relative size rather than absolute layer size
  - ✓ More uniform cells near a surface will have a more uniform boundary layer meshes



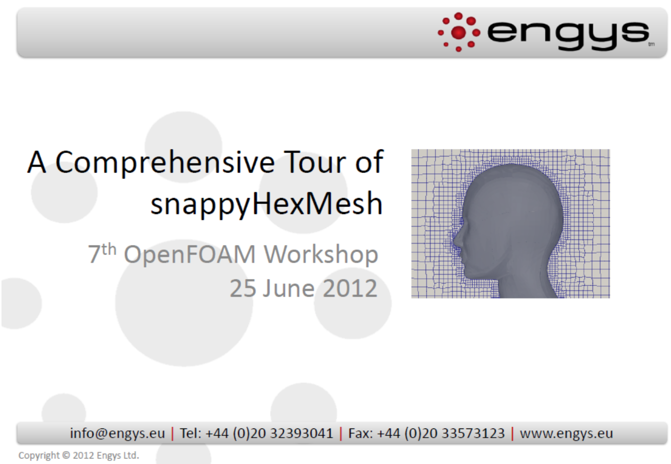
Questions?

Thank You

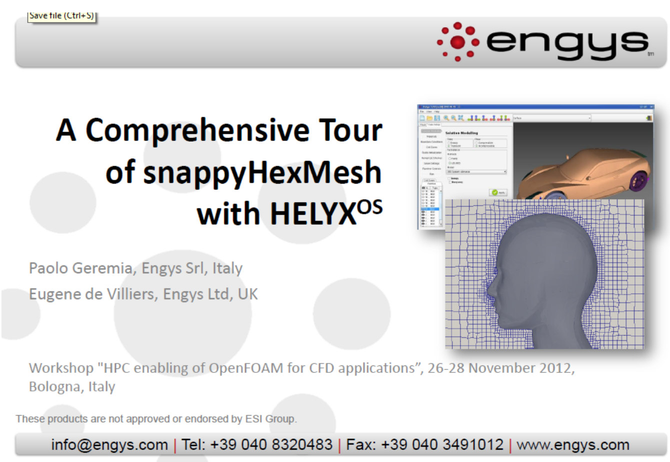
감사합니다

www.engys.com

Appendix A1: References for snappyHexMeshDict



Andrew Jackson. *A Comprehensive Tour of snappyHexMesh*. 7th OpenFOAM Workshop. June 25 2013. Darmstadt Germany.



Paolo Geremia and Eugene de Villiers. *A Comprehensive Tour of snappyHexMesh with HELYX-OS.* Workshop "HPC enabling of OpenFOAM for CFD applications", 26-28 November 2012, Bologna, Italy

A1: snappyHexMeshDict

Dictionary file consists of five main sections:

`geometry`

- Prescribe geometry entities for meshing

`castellatedMeshControls`

- Prescribe feature, surface and volume mesh refinements

`snapControls`

- Control mesh surface snapping

`addLayersControls`

- Control boundary layer mesh growth

`meshQualityControls`

- Control mesh quality metrics

A1: Basic Controls

```
FoamFile
{
   version    2.0;
   format     ascii;
   class      dictionary;
   object     autoHexMeshDict;
}
castellatedMesh true;
snap            true;
addLayers       false;

geometry
{
   flange.stl
   {
      type triSurfaceMesh;
      name flange;
   }
   sphereA
   {
      type searchableSphere;
      centre (0 0 -0.012);
      radius 0.003;
   }
}
```

File header

Keywords

- Switch on/off mesh steps

A1: geometry

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      autoHexMeshDict;
}
castellatedMesh true;
snap            true;
addLayers       false;
```

```
geometry
{
    flange.stl
    {
        type triSurfaceMesh;
        name flange;
    }
    sphereA
    {
        type searchableSphere;
        centre (0 0 -0.012);
        radius 0.003;
    }
}
```

Definition of geometry types

- STL and Nastran files → serial or distributed
- Basic shapes → box, cylinder, sphere...

A1: Supported Types

```
geomA.stl
{
    type          triSurfaceMesh;
    name          geomA;
}
```

```
geomB.stl
{
    type                distributedTriSurfaceMesh;
    distributionType    follow;
    name                geomB;
}
```

Triangulated (e.g. Nastran, STL, OBJ)

- The standard type "**triSurfaceMesh**" reads a copy of each surface on to each processor when running in parallel.

- A distributed surface type exists "**distributedTriSurfaceMesh**" which can reduce the memory overhead for large surfaces
- Utility **surfaceRedistributePar** is used to initially decompose the surface
- Three distribution methods available
  independent: distribution independent of mesh to produce best memory balance
  follow: distribution based on mesh bounding box to reduce communication
  frozen: distribution remains unchanged

A1: Supported Types

```
box
{
    type searchableBox;
    min (-0.2 -0.2 -0.02);
    max (0.44 0.2 0.32);
}

sphere
{
    type searchableSphere;
    centre (3 3 0);
    radius 4;
}

 cylinder
 {
    type          searchableCylinder;
    point1         (0 0 0);
    point2         (1 0 0);
    radius        0.1;
 }
```

User defined shapes

- Basic shapes → box, cylinder and sphere

A1: Supported Types

```
plane
{
    type          searchablePlane;

    planeType     pointAndNormal;
    pointAndNormalDict
    {
        basePoint      (0 0 0);
        normalVector   (0 1 0);
    }
}

plate
{
    type        searchablePlate;
    origin      (0 0 0);
    span        (0.5 0.5 0);
}
```

User defined shapes

- Basic shapes → plane and plate

A1: Supported Types

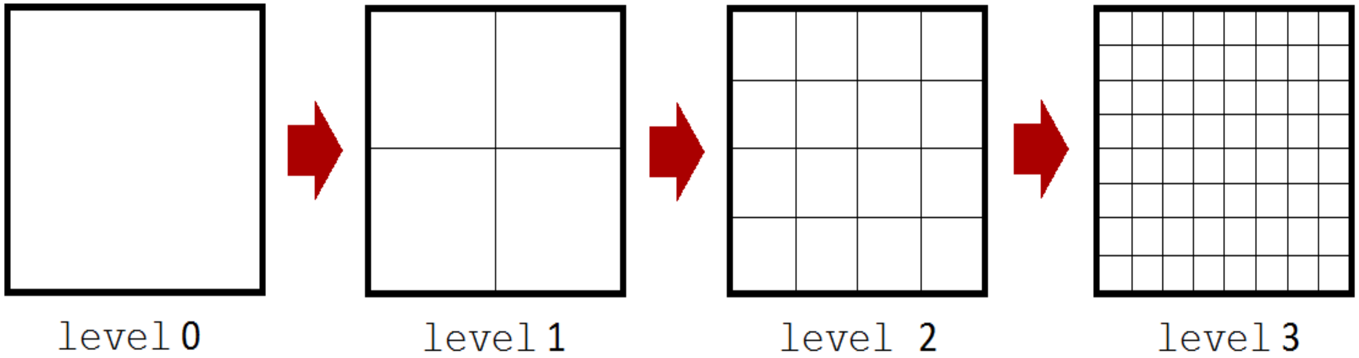
```
twoBoxes
{
    type searchableSurfaceCollection;
    mergeSubRegions true;

    boxA
    {
        surface box;
        scale (1.0 1.0 2.1);
        transform
        {
            type    cartesian;
            origin  (2 2 0);
            e1      (1 0 0);
            e3      (0 0 1);
        }
    }
    boxB
    {
        surface box;
        scale (1.0 1.0 2.1);
        transform
        {
            type    cartesian;
            origin  (3.5 3 0);
            e1      (1 0 0);
            e3      (0 0 1);
        }
    }
}
```

User defined shapes

- Complex shapes → Collection of basic shapes scaled and transformed

A1: Refinement

The first meshing stage is called "Refinement". This is where the initial block mesh is refined based on surface and volumetric refinement settings in the **castellatedMeshControls** sub-dictionary

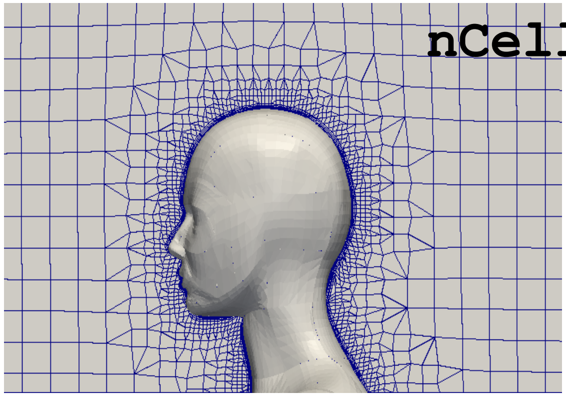


A1: castellatedMesh

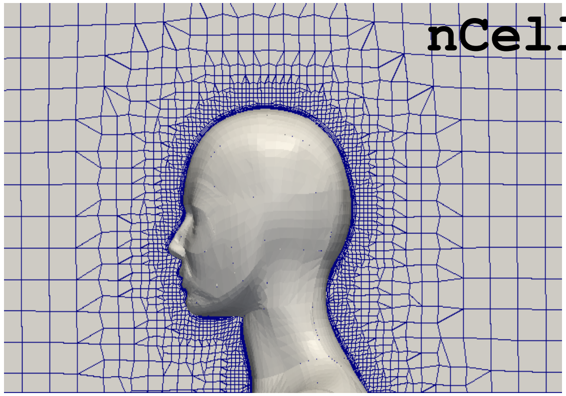
```
castellatedMeshControls
{
    maxGlobalCells 2000000;
    minRefinementCells 0;
    nCellsBetweenLevels 1;

    features();

    refinementSurfaces
    {
      flange
      {
          level (2 3);
          regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
        level (3 3);
        faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
    }

    resolveFeatureAngle 30;

    refinementRegions
    {
      sphereA
      {
        mode inside;
        levels ((1E15 3));
      }
    }
    locationInMesh (-9.23149e-05 -0.0025 -0.0025);
    allowFreeStandingZoneFaces true;
}
```

Mesh control keywords:

- Global mesh size controls
- Buffer layers



nCellsBetweenLevels 1



nCellsBetweenLevels 3

engys

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
          level (2 3);
          regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

User-defined edge refinements

```
features
 (
      {
         file "flange.eMesh";
         level 3;
         //or levels ( (0.1 3) (0.33 2) ) //distance refinement, new in 2.2.x
      }
 );
```

Example .eMesh file

```
FoamFile
{
   version     2.0;
   format      ascii;
   class       featureEdgeMesh;
   location    "constant/triSurface";
   object      flange.eMesh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

3
(
(0.0065 0.0075 -0.02375)
(0.0065 0.0075 0.00225)
(-0.0065 0.0075 -0.02375)
)

2
(
(0 1)
(1 2)
)
```

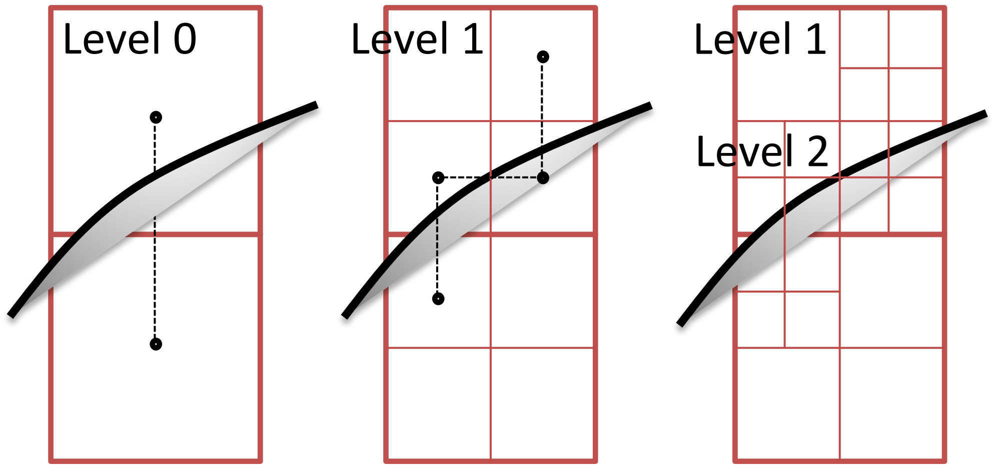

engys

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
          level (2 3);
          regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Surface based refinements:

- Global min. and max. refinements
- Refinement by patch (region)



Surface Mesh Refinements

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
         level (2 3);
         regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Surface based refinements:

- POSIX regular expresssions supported
- patchInfo keyword can be used to set the boundary type on a per surface basis

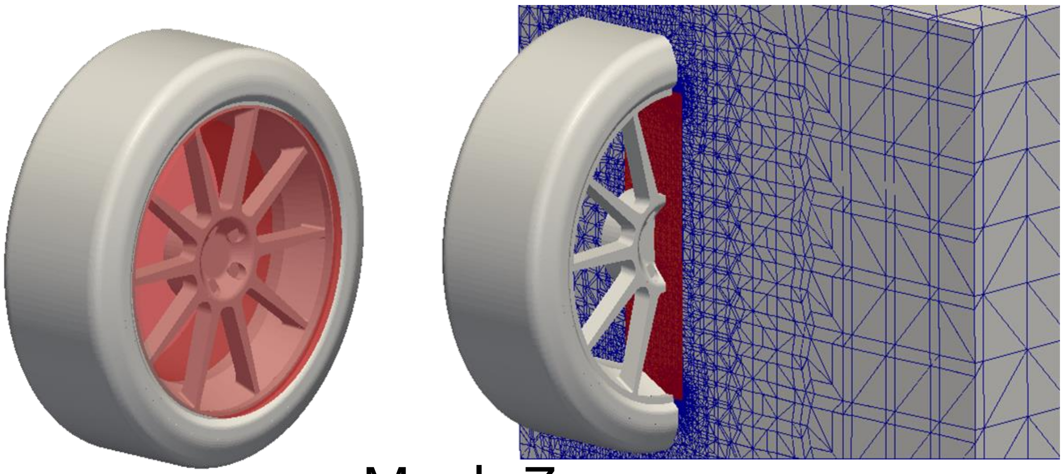
```
refinementSurfaces
   {
      flange
      {
         level (2 3);
         patchInfo
         {
            type wall;
         }
         regions
         {
            "*.inlet|*.outlet"
            {
               level(3,4);
            }
         }
      }
}
```

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
         level (2 3);
         regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Definition of mesh zones:

- Min. and max. refinement levels
- Cell zone name
- Face zone name
- Area selection: inside, outside or insidepoint



Mesh Zones

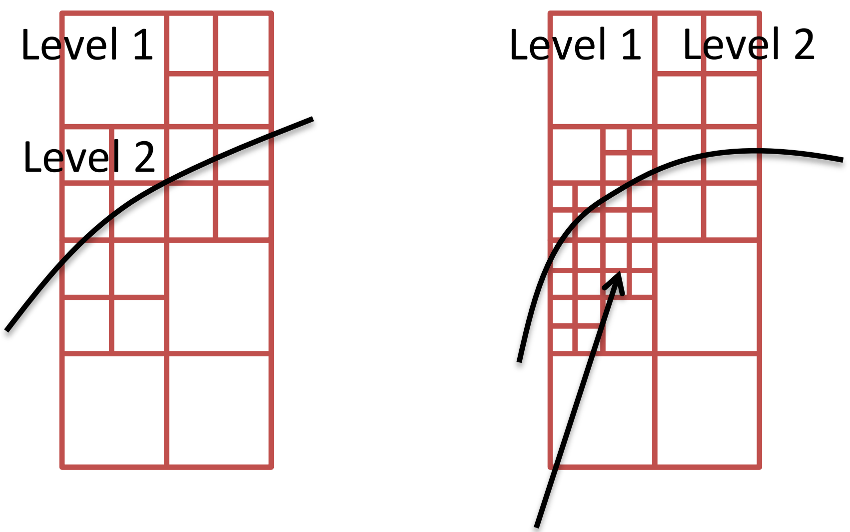
engys

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
         level (2 3);
         regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Additional feature refinements:

- Local curvature
- Feature angle refinement



Level 3 → Local curvature refinement

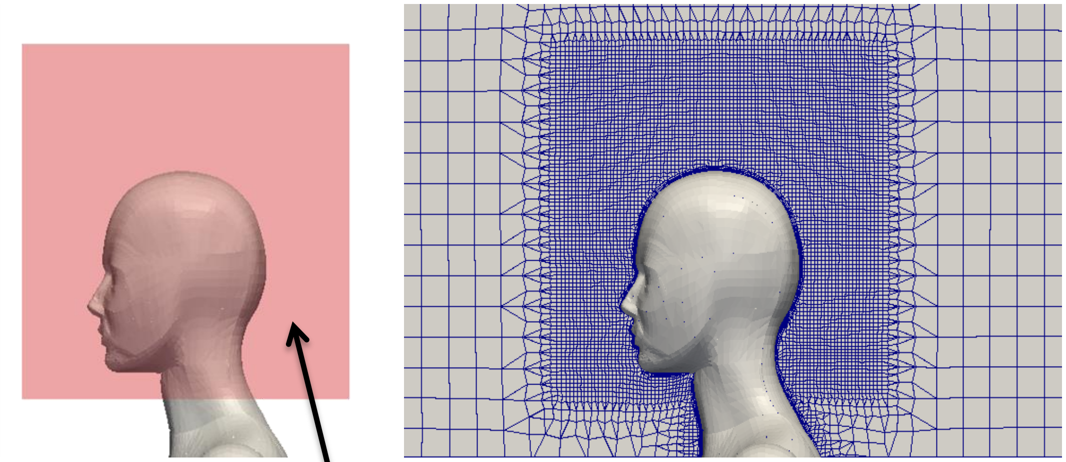
engys

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
          level (2 3);
          regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Volume refinements

- inside (outside)
- distance

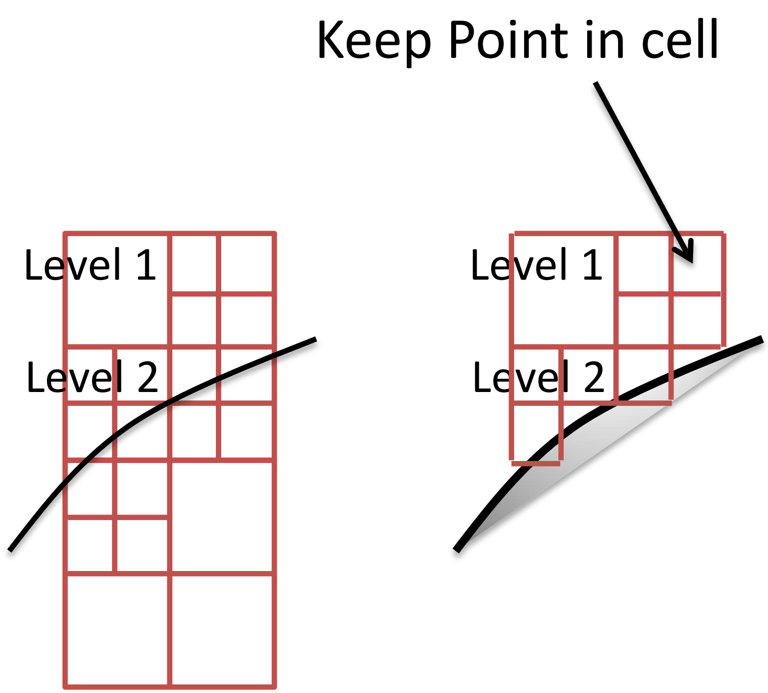


mode inside;
levels ((1E15 3));

A1: castellatedMesh

```
castellatedMeshControls
{
   maxGlobalCells 2000000;
   minRefinementCells 0;
   nCellsBetweenLevels 1;

   features();

   refinementSurfaces
   {
      flange
      {
          level (2 3);
          regions{"*.inlet|*.outlet"{level(3,4);}}
      }
      sphereA
      {
         level (3 3);
         faceZone  zoneA; cellZone zoneA; cellZoneInside
inside;
      }
   }

   resolveFeatureAngle 30;

   refinementRegions
   {
      sphereA
      {
         mode inside;
         levels ((1E15 3));
      }
   }
   locationInMesh (-9.23149e-05 -0.0025 -0.0025);
   allowFreeStandingZoneFaces true;
}
```

Cartesian point (x, y, z) to retain required volume mesh



engys

A1: castellatedMesh

```
castellatedMeshControls
{
  ...
  refinementSurfaces
  {
     flange
     {
        level (2 3);

        faceZone flange;

        faceType boundary;

        cellZone flange;

        cellZoneInside inside;

     }
}
```

New functionality in 2.2.x

Used to define either

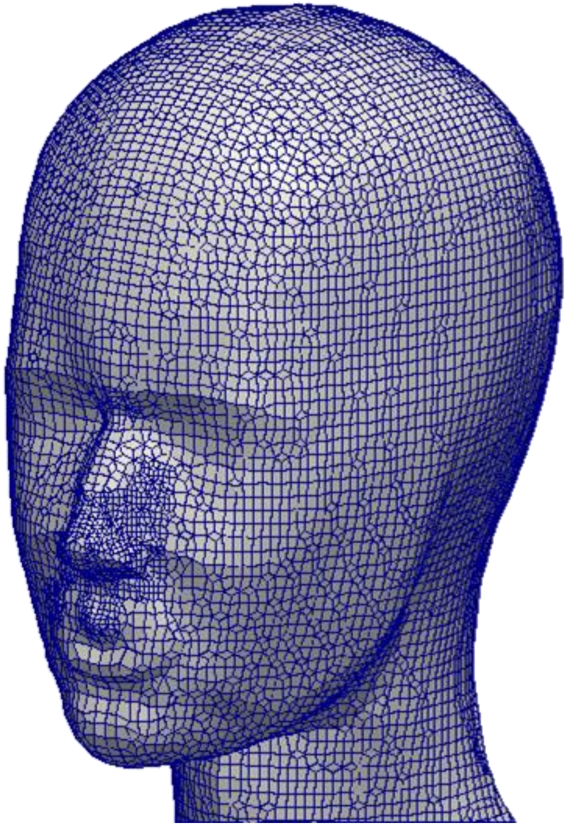
- "baffle"
  creates a pair of faces which match one-to-one.

- "boundary"
  Creates a pair of faces that do not match one-to-one. Less constraint on meshing to create more high-quality meshes.

- "internal"
  keeps faces of faceZone as internal faces.

A1: Surface Snapping

The second meshing stage is called "Snapping" where patch faces are projected down onto the surface geometry. This stage is controlled by settings in the **snapControls** sub-dictionary



A1: snapControls

```
snapControls
{
    nSmoothPatch 3;

    tolerance 1.0;

    nSolverIter 300;

    nRelaxIter 5;

    nFeatureSnapIter 10;

    implicitFeatureSnap false;

    explicitFeatureSnap true;

    multiRegionFeatureSnap false;
}
```

Number of pre smoothing iterations of patch points before projection to the surface is performed

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

Scaling of the maximum edge length for attraction to the surface

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

Number of interior smoothing iterations applied to snapped displacement field

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

Controls number of scaling back iterations for error reduction stage

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

Number of feature snapping iterations to perform. Features edges to attract to are defined by an .eMesh file setup in **castellatedMeshControls** which can also be used for feature refinement.

To extract an eMesh file containing the feature edge information about a particular surface the utility **surfaceFeatureExtract** can be used e.g.

surfaceFeatureExtract -includedAngle 150 <surface> <output set>

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

New functionality in 2.2.x

Uses resolveFeatureAngle to detect changes in the features to find “creases”. Snapping is then performed on a “representation” of the feature from the local topology. (default = false)

A1: snapControls

```
snapControls
{
   nSmoothPatch 3;

   tolerance 1.0;

   nSolverIter 300;

   nRelaxIter 5;

   nFeatureSnapIter 10;

   implicitFeatureSnap false;

   explicitFeatureSnap true;

   multiRegionFeatureSnap false;
}
```

Indicates that an eMesh file is to be used to approximate features within the mesh i.e. uses features defined in castellatedMeshControls
(default = true;

engys

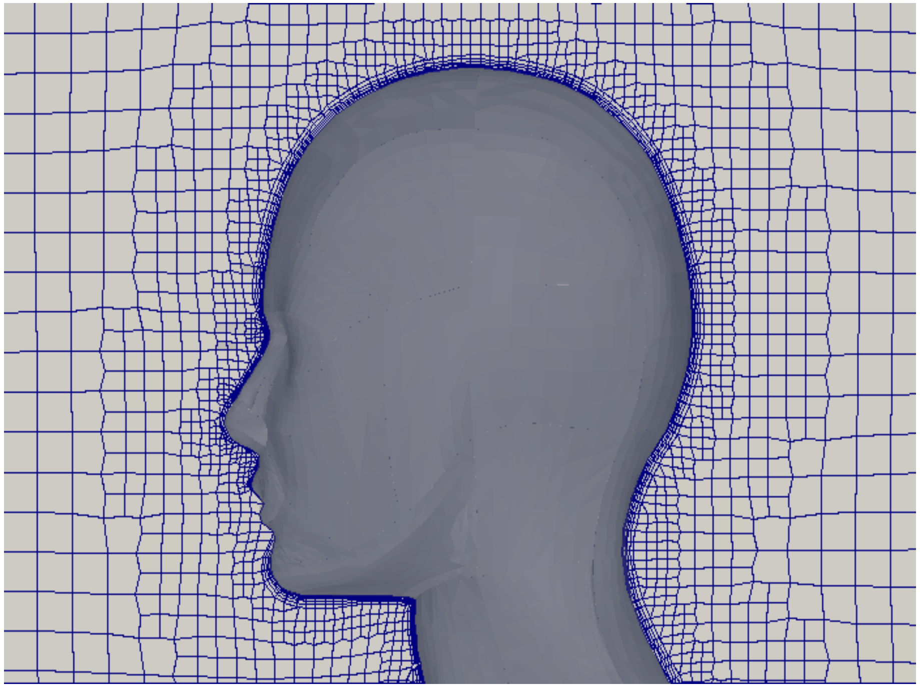
A1: snapControls

```
snapControls
{
    nSmoothPatch 3;

    tolerance 1.0;

    nSolverIter 300;

    nRelaxIter 5;

    nFeatureSnapIter 10;

    implicitFeatureSnap false;

    explicitFeatureSnap true;

    multiRegionFeatureSnap false;
}
```

In conjunction with explicitFeatureSnap, this is used to detect the features between multiple surfaces.

A1: Layers

The final meshing stage is called “Layer addition” where a layer of cells is added to a specified set of boundary patches. This stage is controlled by the settings in the **addLayersControls** sub-dictionary



A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

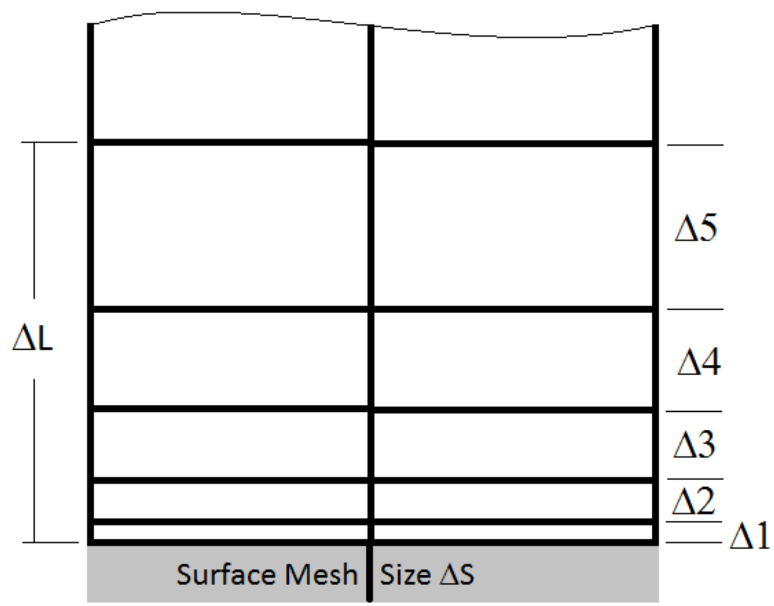
Specification of the number of layers to be grown on each patch. Supports regular expression syntax

A1: addLayersControls

```
addLayersControls
{
  layers
  {
    "flange_.*"{nSurfaceLayers 1;}
  }

  finalLayerThickness 0.4;
  expansionRatio 1.15;

  minThickness 0.2;

  relativeSizes true;

  // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

  maxFaceThicknessRatio 0.5;

  nLayerIter 50;

  meshQualityControls::relaxed.
  nRelaxedIter 20;

  nRelaxIter 5;
}
```

finalLayerThickness is the ratio of the final layer height relative to the adjacent surface mesh size, i.e. $\frac{\Delta 5}{\Delta S}$

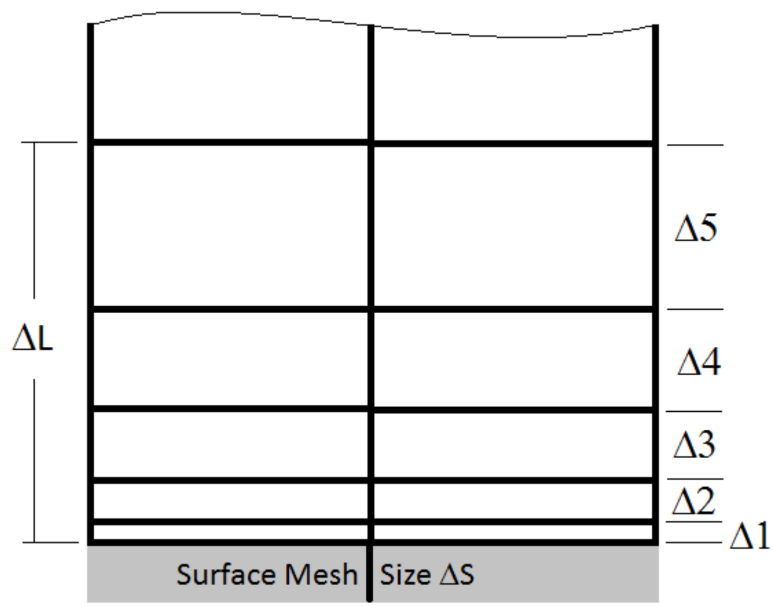
expansionRatio is the ratio of heights from one layer to the next consecutive layer in the direction away from the surface, i.e. $\frac{\Delta 2}{\Delta 1} = \frac{\Delta 3}{\Delta 2} = \frac{\Delta 4}{\Delta 3} = \frac{\Delta 5}{\Delta 4}$



A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Specification of the number of layers, the final layer thickness and expansion ratio uniquely defines the layer profile and is used to calculate the first cell height Δ1 and total layer thickness ΔL



A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

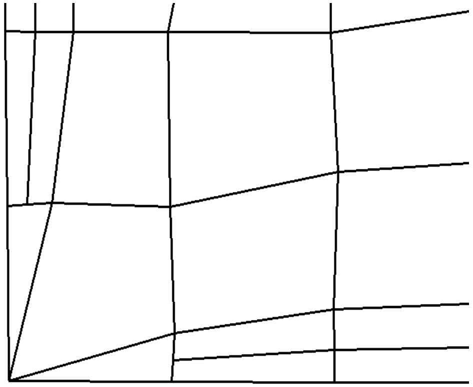
Specification of a minimum layer thickness below which height layers will automatically be collapsed.

The final layer thickness and minimum thickness can be defined as either being relative (true) to the background spacing ΔS or defined as an absolute (false) length.

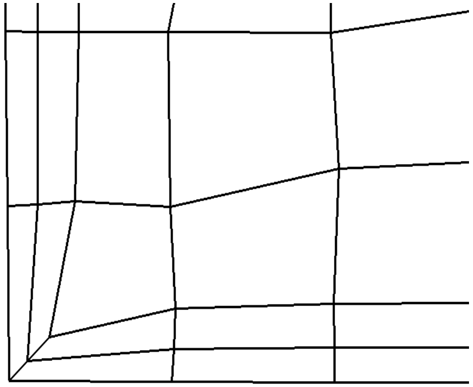
A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Specification of feature angle above which layers are collapsed automatically



featureAngle 45;



featureAngle 180;

engys

A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Specifies what feature angle to allow layers to slip “perpendicularly” up a patch

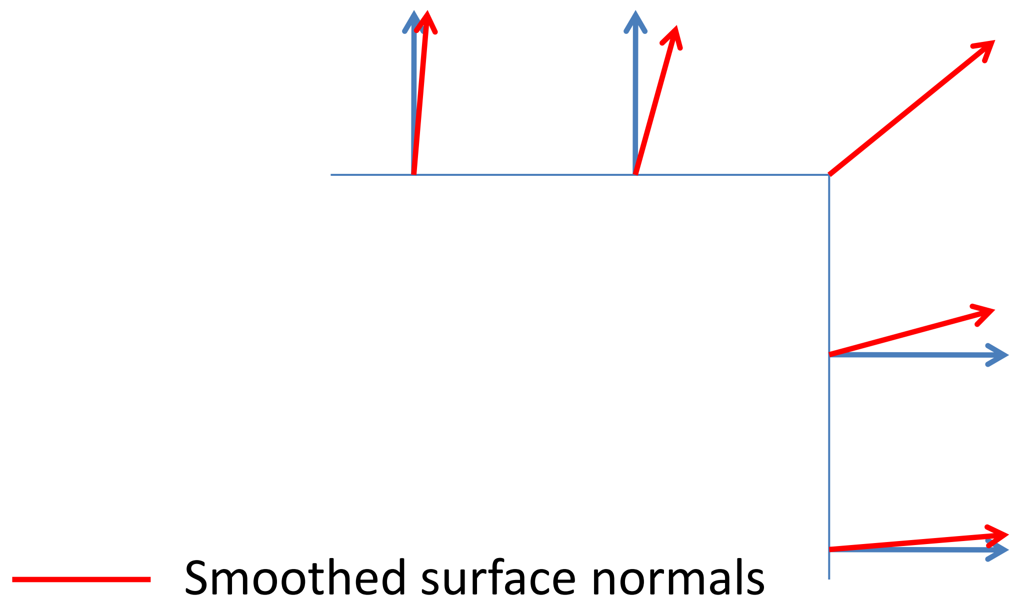
i.e. “at non-patch sides, allow mesh to slip if extrusion direction makes an angle larger than slipFeatureAngle”

engys

A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Smoothing can be performed on the surface point normals (nSmoothSurfaceNormals), layer thickness (nSmoothThickness) and the interior displacement field (nSmoothNormals) e.g.

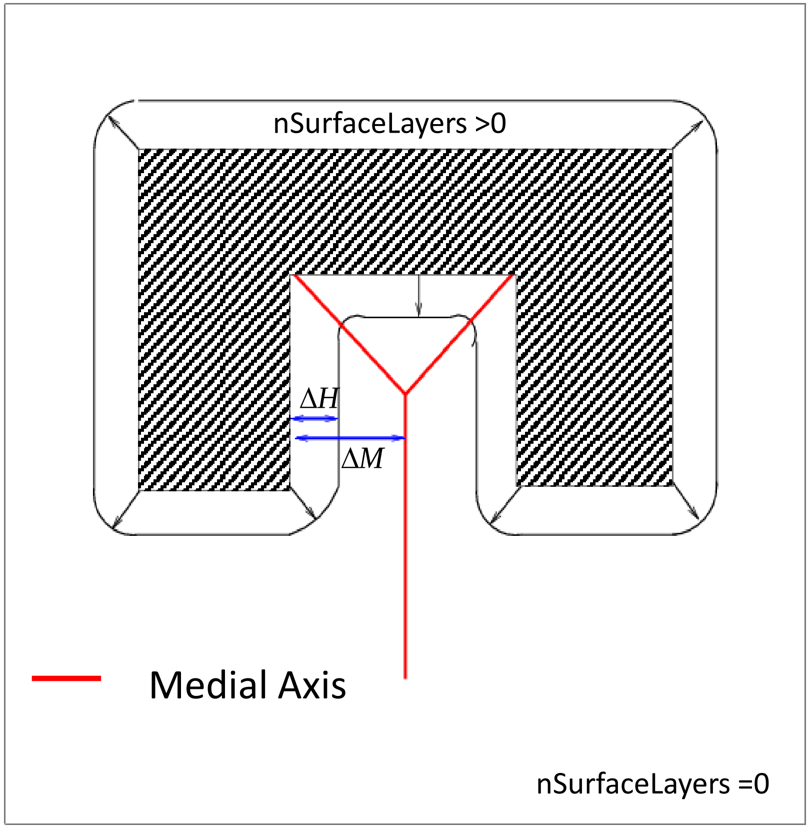


engys

A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

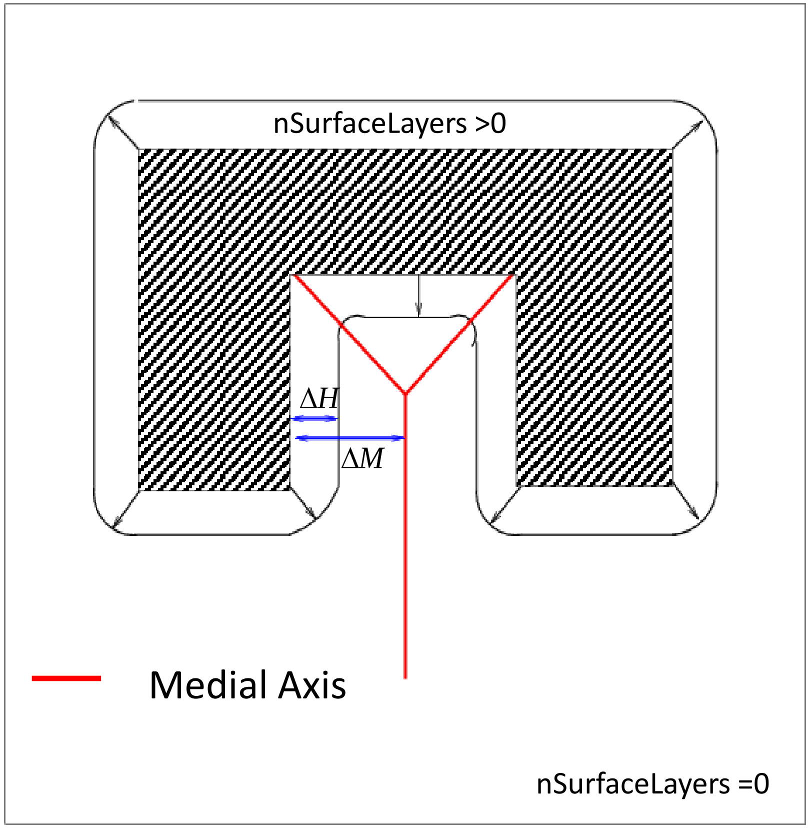
This angle is used to define a medial axis which is used when moving the mesh away from the surface



A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Used to reduce the layer thickness where the ratio of layer thickness to distance to medial axis (ΔH/ΔM) becomes too large



A1: addLayersControls

```
addLayersControls
{
   layers
   {
      "flange_.*"{nSurfaceLayers 1;}
   }

   finalLayerThickness 0.4;
   expansionRatio 1.15;

   minThickness 0.2;

   relativeSizes true;

   // Advanced settings
  featureAngle 30;
  slipFeatureAngle 30;
  nSmoothSurfaceNormals 1;
  nSmoothNormals 3;
  nSmoothThickness 10;

  minMedianAxisAngle 80;
  maxThicknessToMedialRatio 0.3;

   maxFaceThicknessRatio 0.5;

   nLayerIter 50;

   meshQualityControls::relaxed.
   nRelaxedIter 20;

   nRelaxIter 5;
}
```

Used to identify warped faces and terminate layers on these faces

If the layer iteration has not converged after a certain number of iterations exit the layer addition loop early with the currently generated layer

If layer iteration has not converged after a specified number of iterations then use a set of relaxed mesh quality metrics, set in **meshQualityControls,** to achieve convergence

Controls number of scaling back iterations during error reduction stage