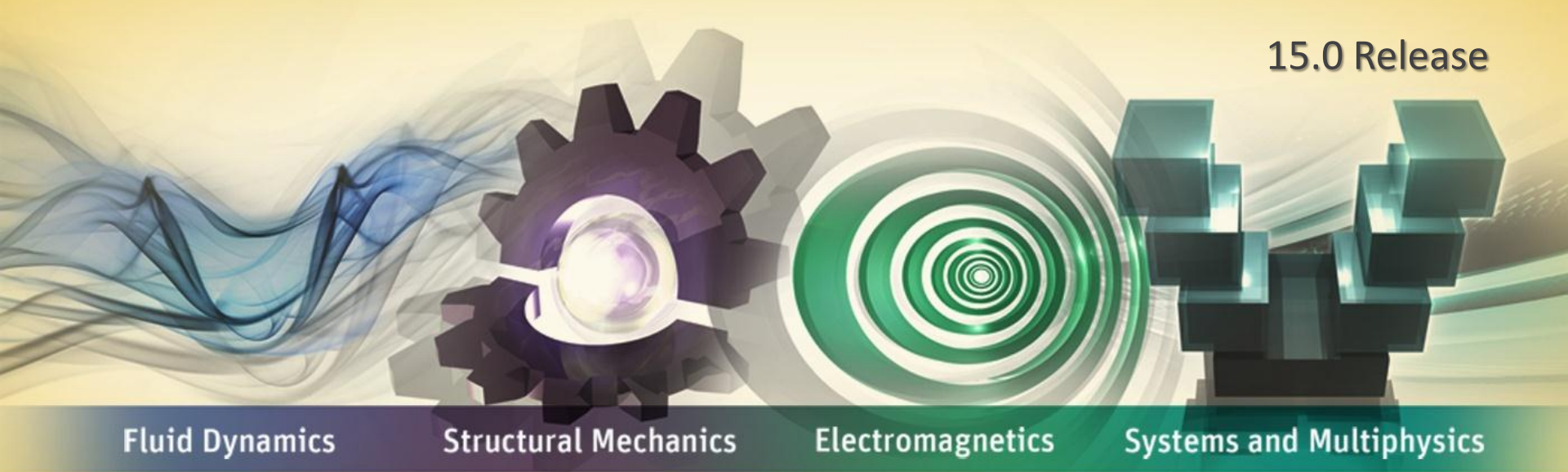


Lecture 4

Unstructured Volume Meshing

15.0 Release

A horizontal banner with a yellow-to-orange gradient background. It features four distinct 3D visualizations: blue fluid flow streamlines, a purple gear with a glowing center, a green concentric circle pattern, and a stack of teal cubes. Below these images is a dark teal bar containing four white text labels.

Fluid Dynamics

Structural Mechanics

Electromagnetics

Systems and Multiphysics

Introduction to ANSYS

ICEM CFD

General Procedure

- First decide global volume mesh parameters
 - *Global Mesh Setup > Volume Meshing Parameters*
 - Select *Mesh Type*
 - Select *Mesh Method* for chosen *Type*
 - Set options for specific *Methods*



- Set mesh sizes
 - Globally 
 - Locally 
 - *Part/Surface/Curve Mesh Setup*
 - For *From geometry*:
 - Octree
 - Cartesian

• Define volumetric region

- Material points 
 - Typically for octree on complex models
 - Multiple volumes possible

• Define density regions (optional)

- Applying mesh size within volume where geometry doesn't exist 

- Load/create surface mesh
 - As in *Shell Meshing* chapter
 - For *Delauney, Advancing Front, ANSYS TGrid, Hex-Dominant*
 - Either of these types run from geometry will automatically create surface mesh using global and local *Shell Mesh* settings without any user input/editing
 - If in doubt, run *Shell Mesh* first, then from “existing mesh”



• Compute Mesh

- *Mesh > Compute*
- *Mesh > Volume Mesh*



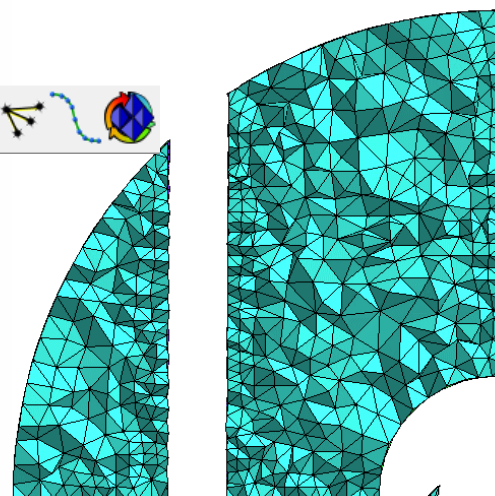
• Compute Prism (optional)

- As separate process
- Also option to run automatically with tetra creation





Pure tetra

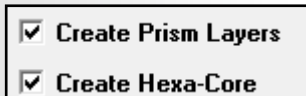


– Most used type

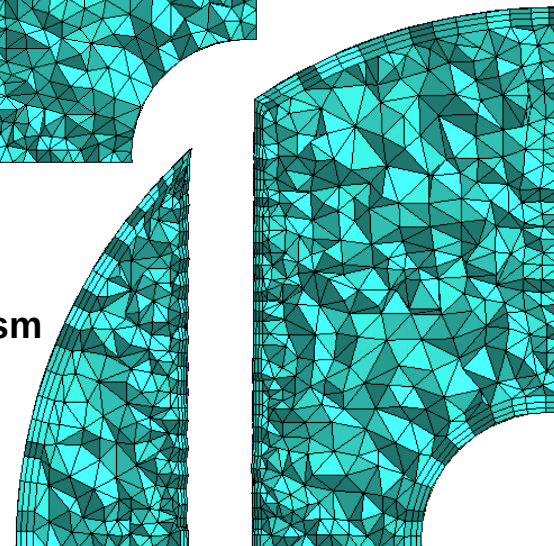
1. Pure tetra

2. Tetra with prism layers

- Prism is discussed in a later presentation
- Prisms from tri surface mesh
- Hexas from quad surface mesh
- Pyramids to cap off any quad faces from prism sides or hex prism layers

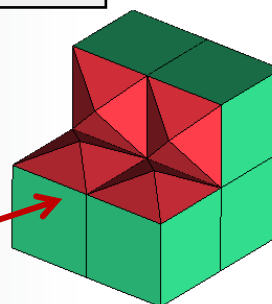


Tetra/Prism

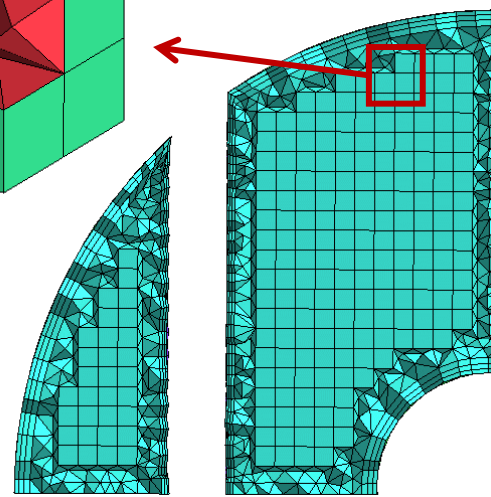


3. Tetra with hex core

- Can also have prisms
- Hexa filling majority of volume
- Tetra (from Delaunay algorithm) used to fill between surface mesh or top of prism layers and hex core
- Pyramids to make conformal between tetra and hexa quad faces



Tetra/Prism/
Hexcore

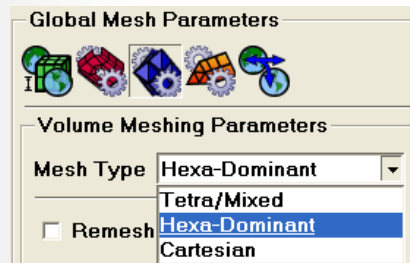


– A Hybrid mesh can be created by merging a tetra mesh with a structured hex mesh

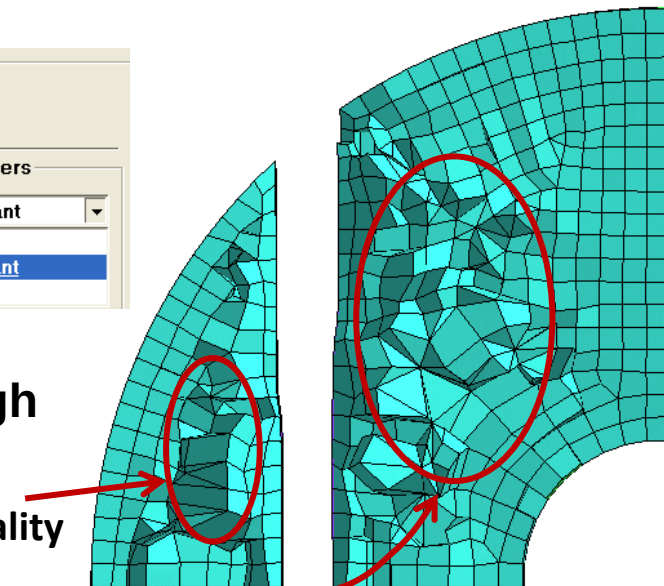
Mesh Types - Continued

- **Hexa-Dominant**

- Uses existing quad mesh
- Good quality hex near surface
- Typically poor interior quality
- Typically not suited for CFD, but good enough for static structural analysis
- Not covered in detail here

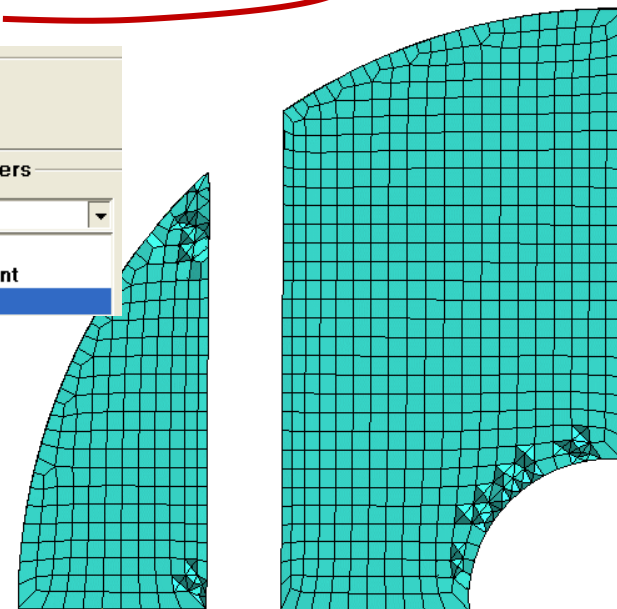
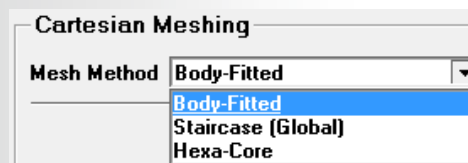
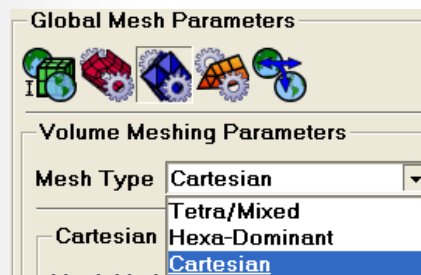


Poor quality interior



- **Cartesian**

- Methods available in Cartesian
 - *Body fitted*
 - *Staircase*
 - *Hexa-Core*
- Automatic pure Hexa
- Rectilinear mesh
- Fastest method for creating volume mesh
- Not covered in detail here



Mesh Input Sources

- Input sources for tetra/mixed meshing modules

1. Start from just geometry

- Octree tetra
 - Robust
 - Can “walk” over small features and gaps/holes
- Cartesian
 - Stair step
 - Body Fitted Cartesian
- Mesh sizes set on geometry or globally

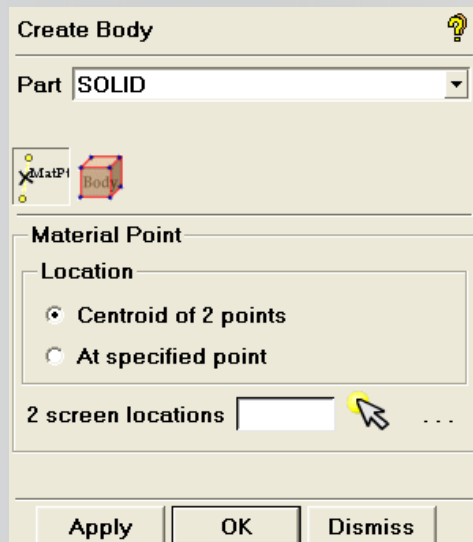
2. Start from a closed, watertight shell mesh

Volume
filling
methods

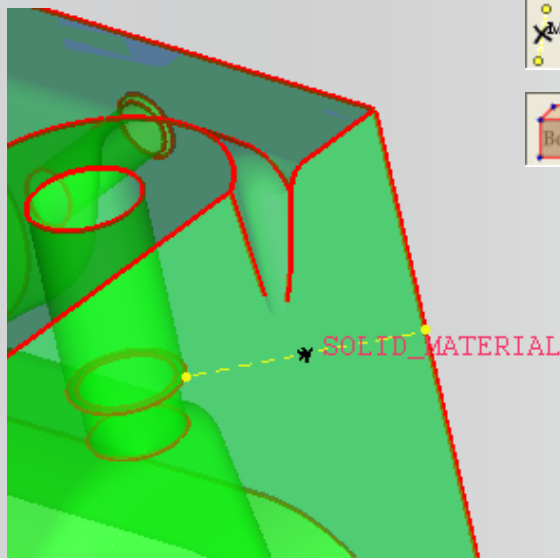
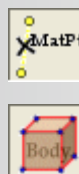
- Delaunay and T-Grid
 - Quick
- Advancing Front
 - Slow mesh size growth for several layers (6 or so) from surface, then fast growth after that into volume
- Hexa Core
- Hex Dominant

3. Start from both geometry and shell mesh

- Octree tetra only
 - Portions of model already shell meshed
 - Shell meshed section may not be 100% respected (may adjust shell mesh some)
 - Set mesh sizes on geometry of unmesh portions



- **Material point and body**
 - Material point used by tetra octree to instruct which volume regions to keep
 - Volume elements will be in the same part as the material point
 - Used in hexa blocking as a part for placing blocks
 - Delaunay, Adv Front, and Tgrid also use this if **Flood fill after completion** is on in global settings
 - Material point method is most robust
 - **By Topology** method automatically creates a material point in every closed volume
 - Requires **build diagnostic topology** first to determine connectivity
 - Can save you the work of creating a lot of material points for each region
 - Any regions not completely closed (yellow curves indicating gaps/holes) will not get a material point so this is less robust



Mesh Methods - Octree

- Type - **Tetra/Mixed**

- Method - **Robust (Octree)**

- Same as **Shell Meshing > Patch Independent**

- Except volumetric tetras are kept

- Advantages

- Good choice for complex and/or dirty geometry
 - Little or no geometry cleanup necessary
 - No time wasted with detailed shell meshing
 - Don't want to spend time defeaturing geometry

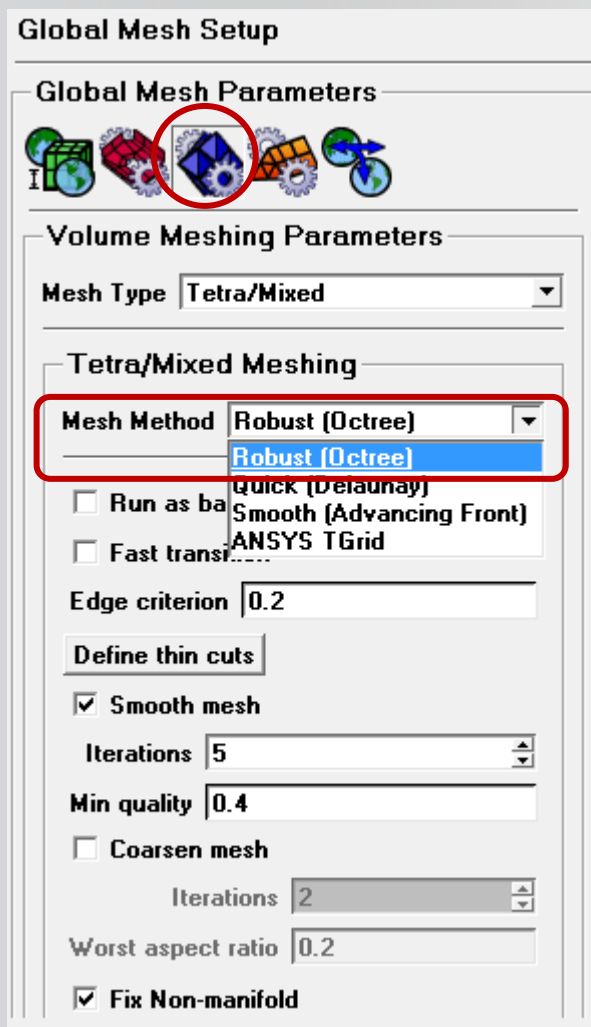
- Disadvantage

- Intensive on CPU usage and memory

- Just set appropriate mesh sizes on geometry

- Global sizes (max size, curvature/proximity based)
 - By parts (**Part mesh setup** spreadsheet)
 - Surfaces
 - Curves
 - Review **Shell Meshing** chapter

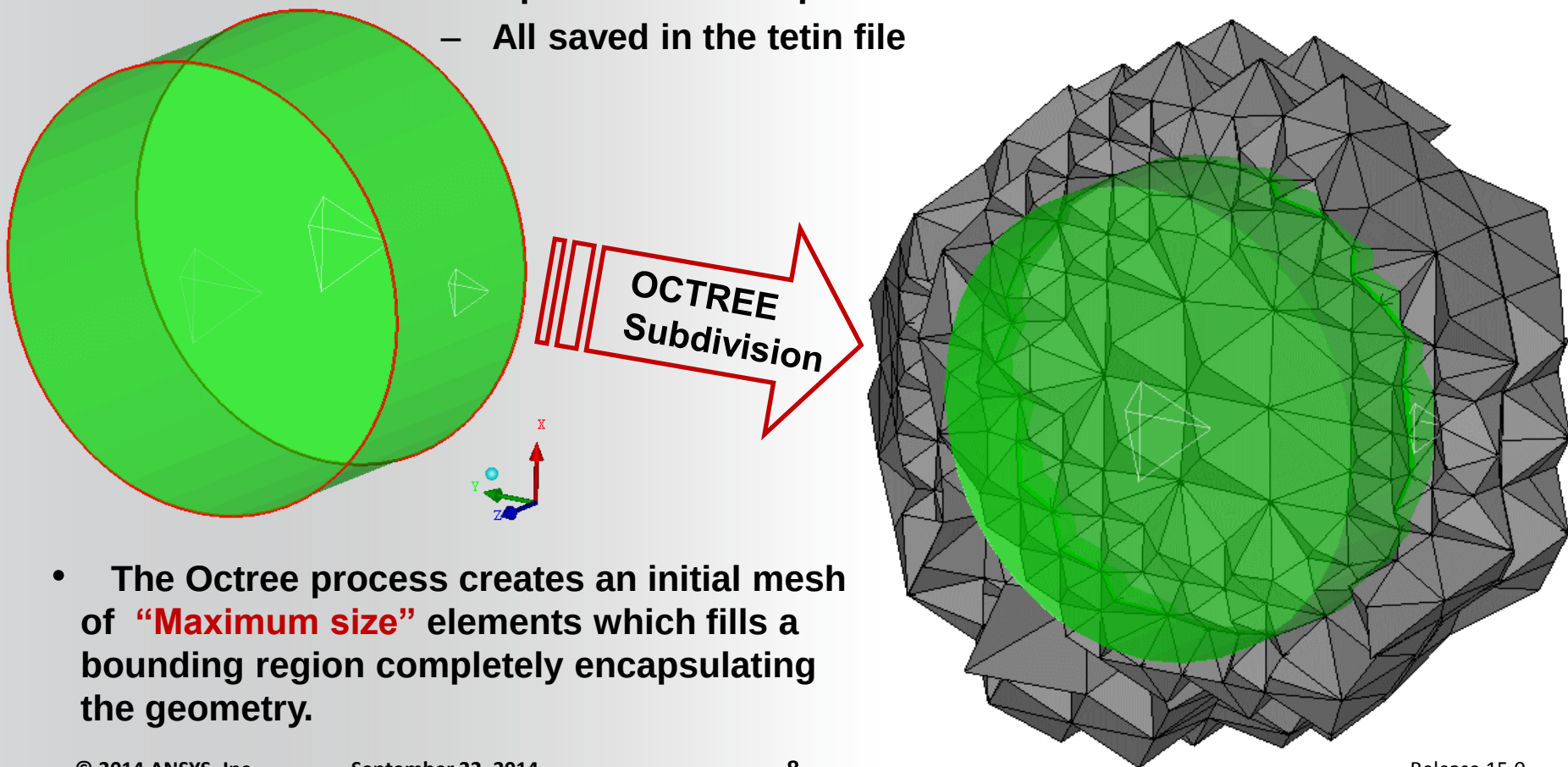
- **Part/Surface/Curve Mesh Setup**



Octree Tetra Process

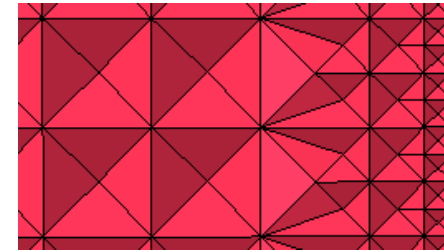
Initial conditions

- Geometry including surfaces, curves and points (from **Build Topology**)
- Mesh size set globally and/or on surfaces/curves/densities
- Optional material point could also be created
- All saved in the tetin file

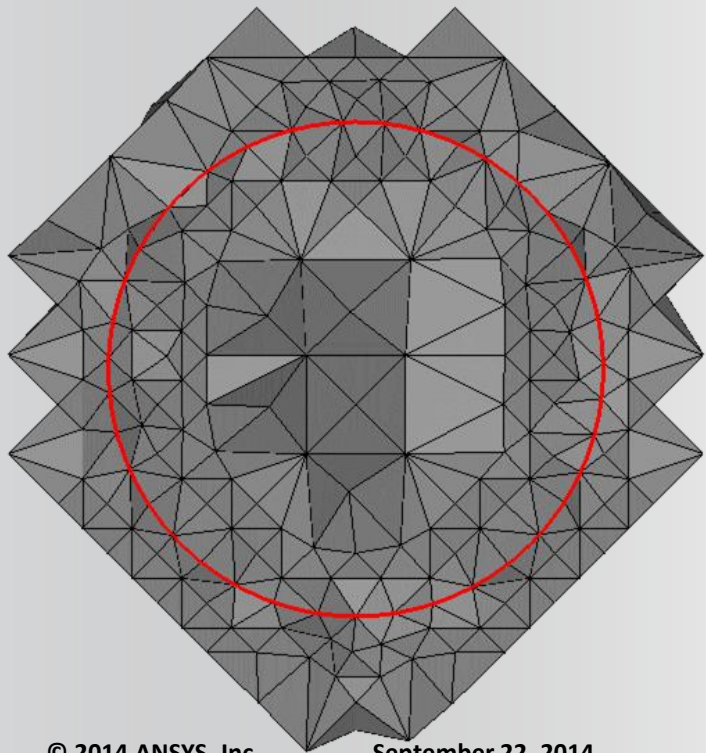


Tetra Process, Cont'd

- Mesh then subdivided to meet the entity size parameters
 - Factor of 2 in 3-dimensions, hence the name Octree
- Nodes are projected to model surfaces, curves, and points
- Automatic “flood fill” process finds volume boundaries
 - Initial element assigned to part name of material point
 - Adjacent layers added to same part until boundary surfaces are reached
 - Elements outside the domain are marked into a reserve part name called **ORFN**, then deleted

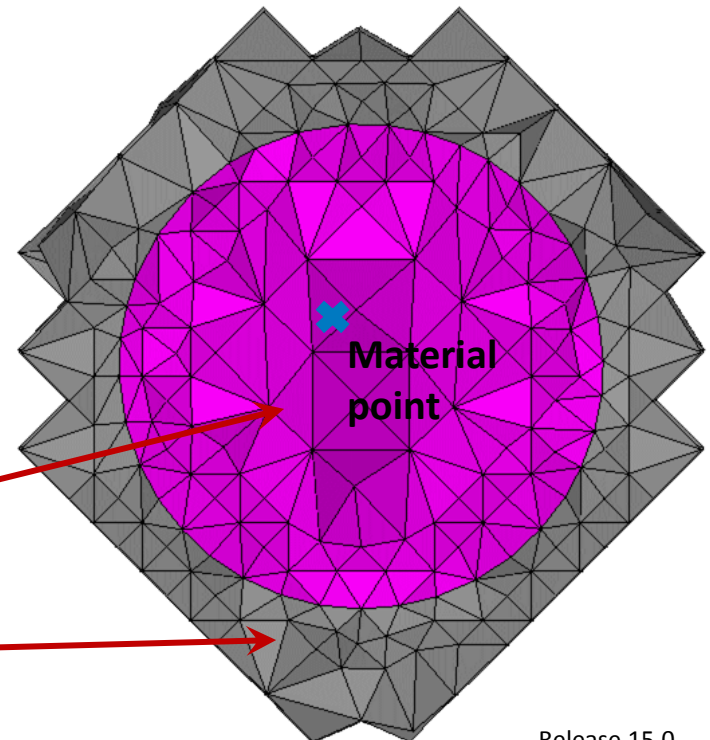


2:1 growth
ratio because
edges get split
in half



• User defined
volumes kept

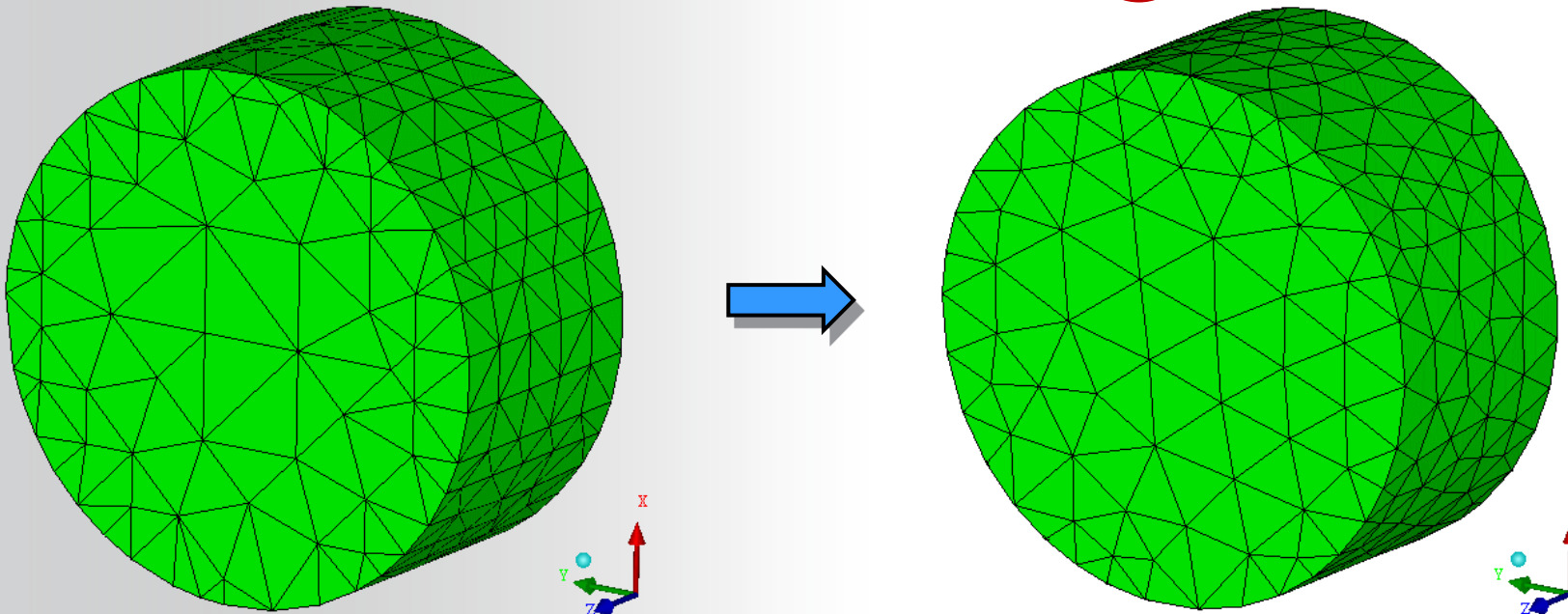
• **ORFN** region
is discarded



Tetra Octree Process, Cont'd

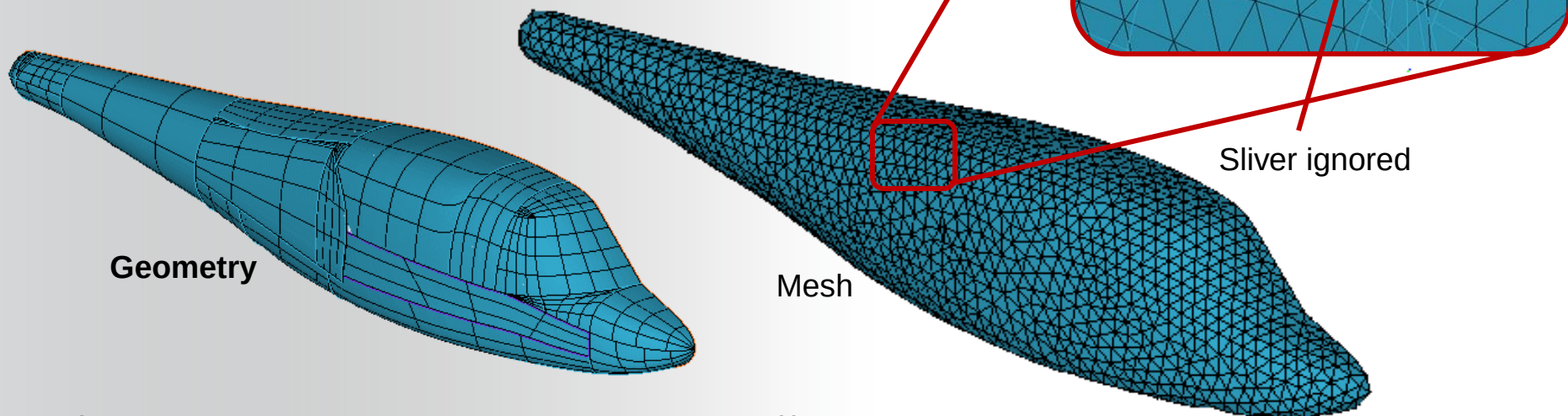
- Smooth
 - Octree mesh is initially composed of regular right angle tetras
 - Since nodes are simply moved to nearest geometry, smoothing is necessary
 - Smoother can be set to run automatically to improve quality in *Global Mesh Setup*
 - Or run afterwards:

Edit Mesh -> Smooth Mesh Globally



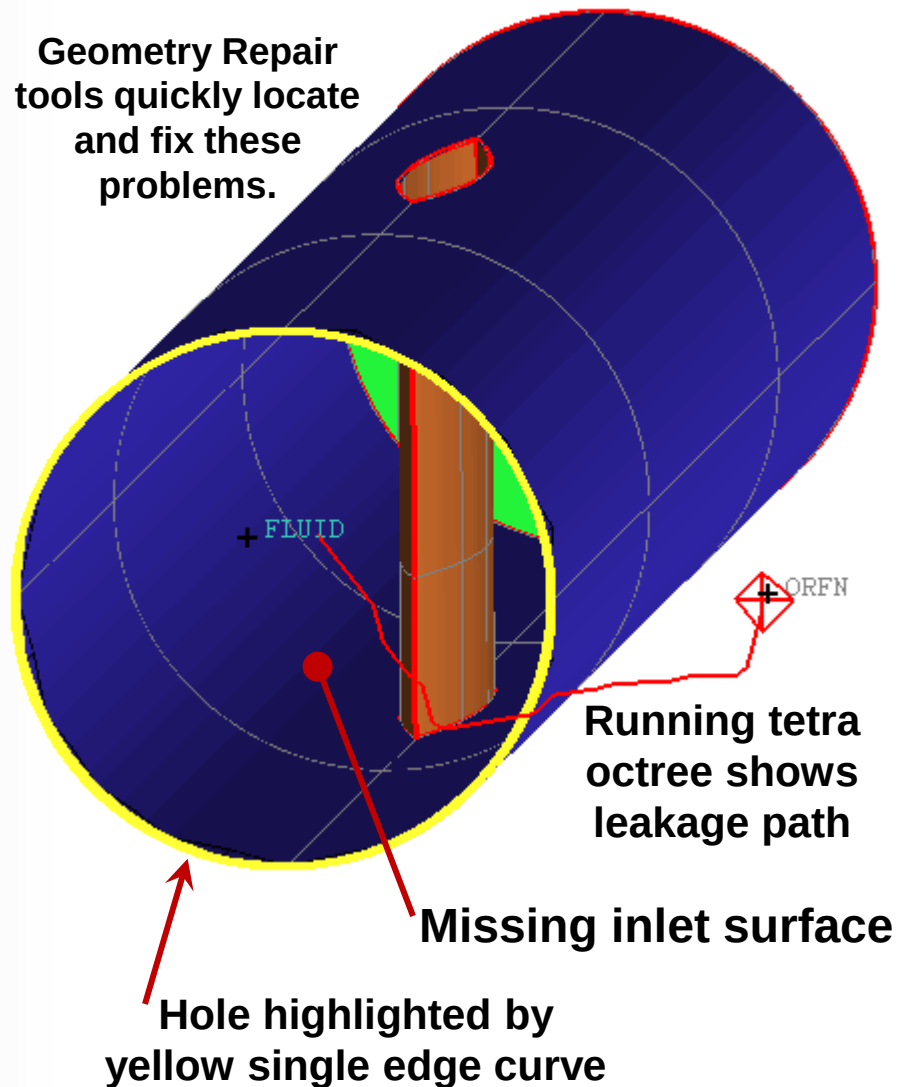
Octree Method Characteristics

- Volume first generated independent of surface model and nodes projected onto surfaces
- Surface mesh is created when outside tetras are cut away
- Resulting mesh is independent of the underlying arrangement of surfaces
 - Not all surface edges need to be captured!
 - Surfaces edges only captured if curve exists there
 - Delete curves to ignore hard edges
 - Or “filter” points/curves under *Build Diagnostic Topology*

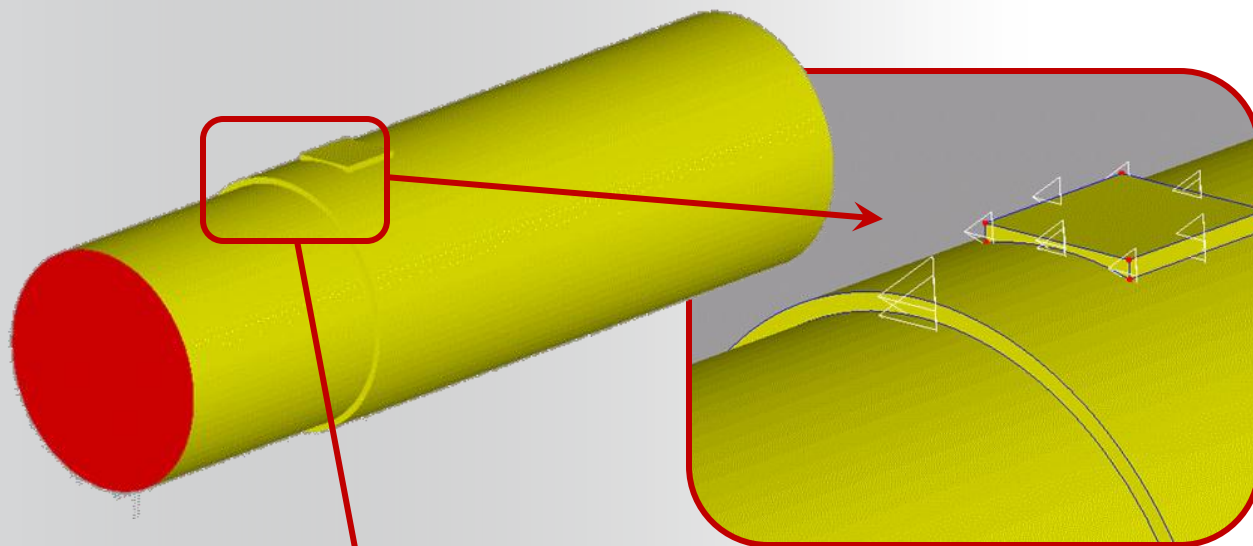


Geometry Requirements for Octree Tetra

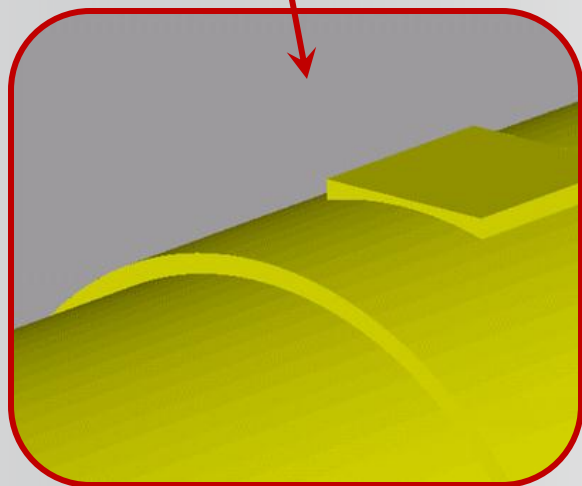
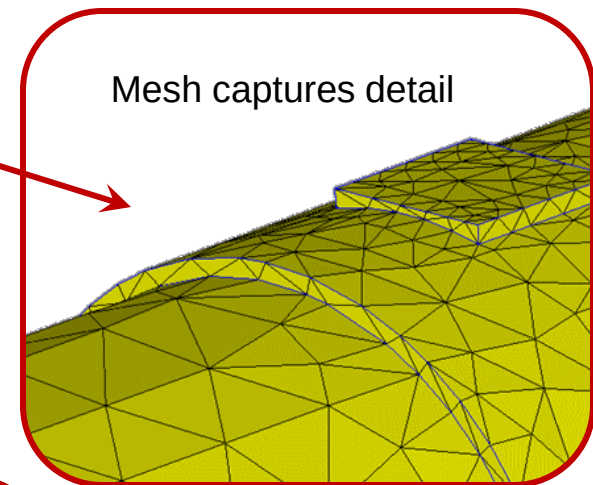
- Tetra requires a reasonably enclosed surface model
 - Run **Build Diagnostic Topology** to find gaps/holes
 - Octree can tolerate gaps smaller than the local element size (approximately $1/10^{\text{th}}$ the element size or less)
- Keep points and curves at key features and hard edges
 - **Filter curves and points** by angle with **Build Diagnostic Topology**
- Create Material points to define volumes
 - Will material point(s) if none exists (named **CREATED_MATERIAL#**)
- Set Global, Part, Surface, Curve Size Parameters
 - Similar to **Shell Meshing** section



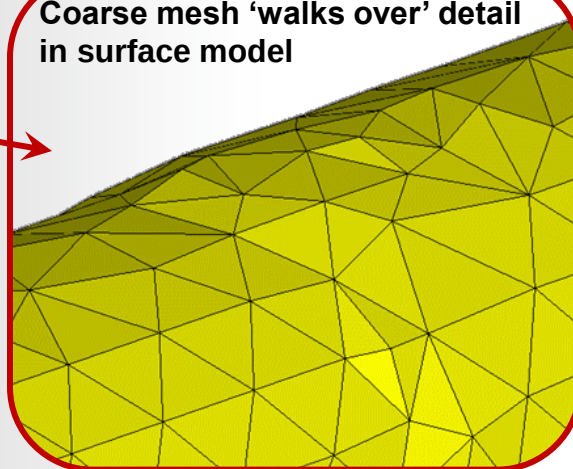
Using Points and Curves with Tetra Octree



- Curves and points included
- Mesh size specified on curves and surfaces



Coarse mesh 'walks over' detail
in surface model



- Curves and points not included
- Mesh size specified only on surfaces

- Curves and points affect which features are captured by the mesh!
- **Build Topology** easily creates the necessary points and curves easily with **filter** by angle

Compute Mesh – Tetra Octree



• Run options: *Compute Mesh > Volume Meshing Parameters*

– *Create Prism Layers*

- Will create prisms marked under *Part Mesh Setup*
- Immediately after tetra calculation
- Prism layers grown into existing tetra mesh

– *Create Hexa-Core*

- Will retain tri surface mesh (or tri and prisms), throw away tetra mesh and regenerate volume
- Fill volume interior with Cartesian hexas
- Cap off hexas with pyramids
- Map tetra to tri or top prism face with Delaunay filling algorithm

– *Input*

• Select Geometry

– *All, Visible*

– *Part by Part*

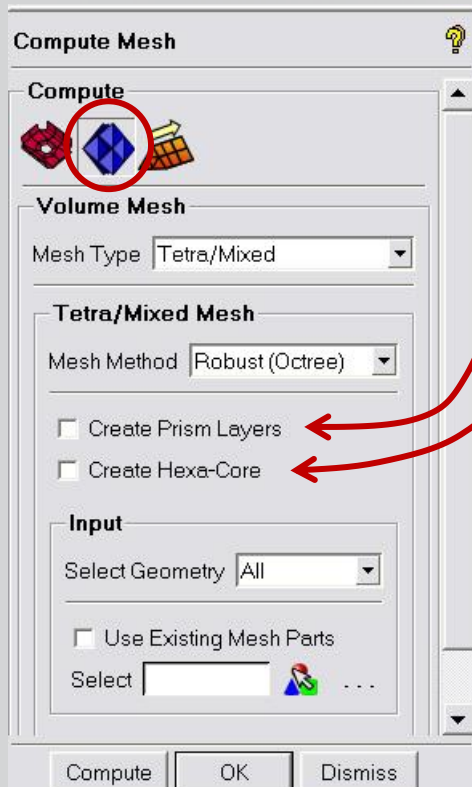
- Meshes each part separately
- Mesh not conformal between parts

– *From file*

- Select tetin file (save memory by not loading it)

– *Use Existing Mesh Parts*

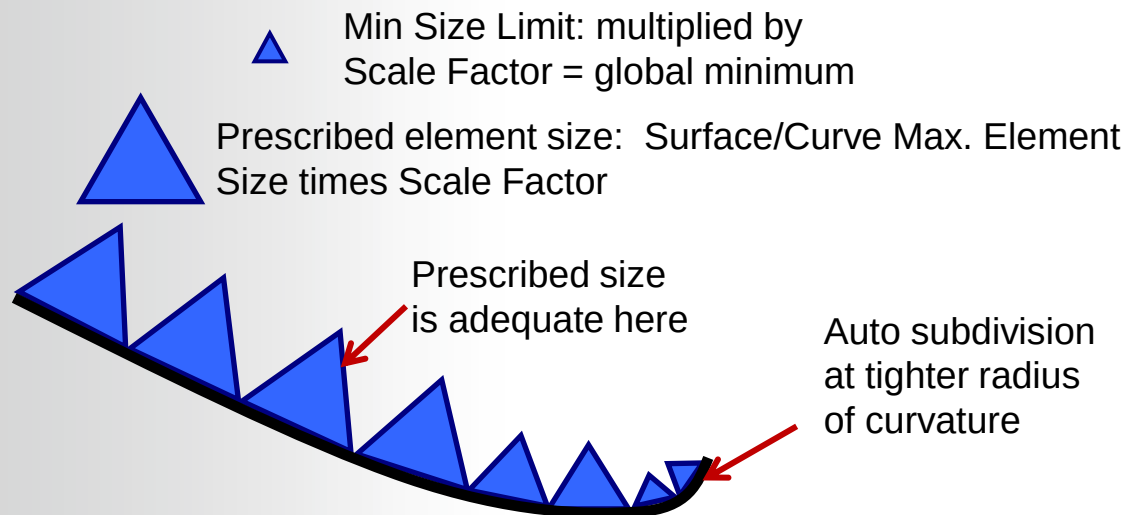
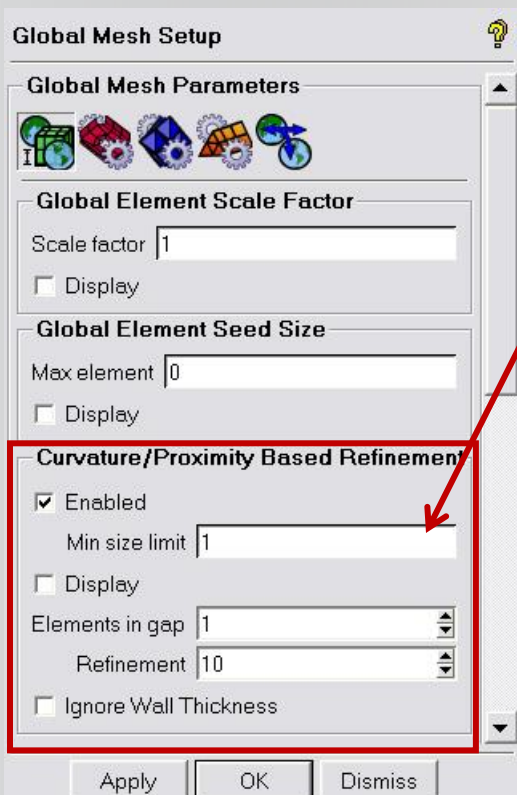
- Select *Parts* that are already surface meshed
- Uses *Make Consistent* to match octree volume mesh to existing surface mesh



Curvature/Proximity Based Refinement

- **Curvature/Proximity Based Refinement**

- **Octree** and **Patch Independent Surface Mesher**
- Automatically subdivides to create elements that are smaller than the prescribed entity size in order to capture finer features
- **Min size limit** value entered is multiplied by the global **Scale Factor** and is the minimum size allowed for the automatic subdivision
- Used primarily to avoid setting up meshing parameters specifically for individual entities thus allowing the geometry to determine the mesh size
- Convenient for geometry with many fillets of varying curvature



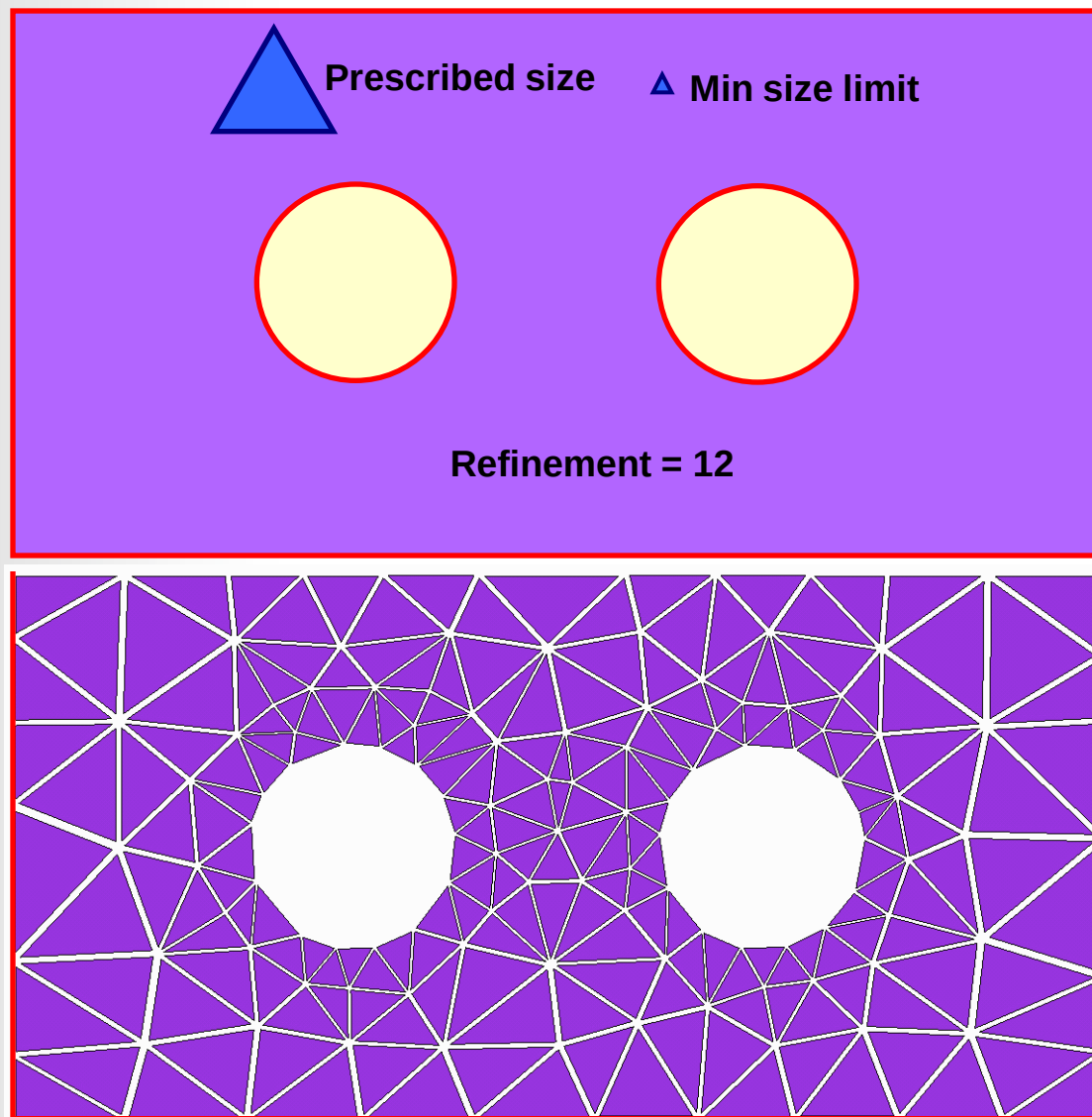
Curvature Based Refinement

- **Refinement**

- Approximate number of elements along curvature if extrapolated to 360°
- To avoid subdivision always to global minimum which would otherwise result in too many elements
 - Subdivision will stop once number of elements along curvature is reached
 - Approx max number of edges along 360°
- Won't exceed global minimum set by **min size limit** value

- **Example**

- Specified refinement achieved with larger elements
- Global minimum (**min size limit**) not realized, not necessary to capture curvature



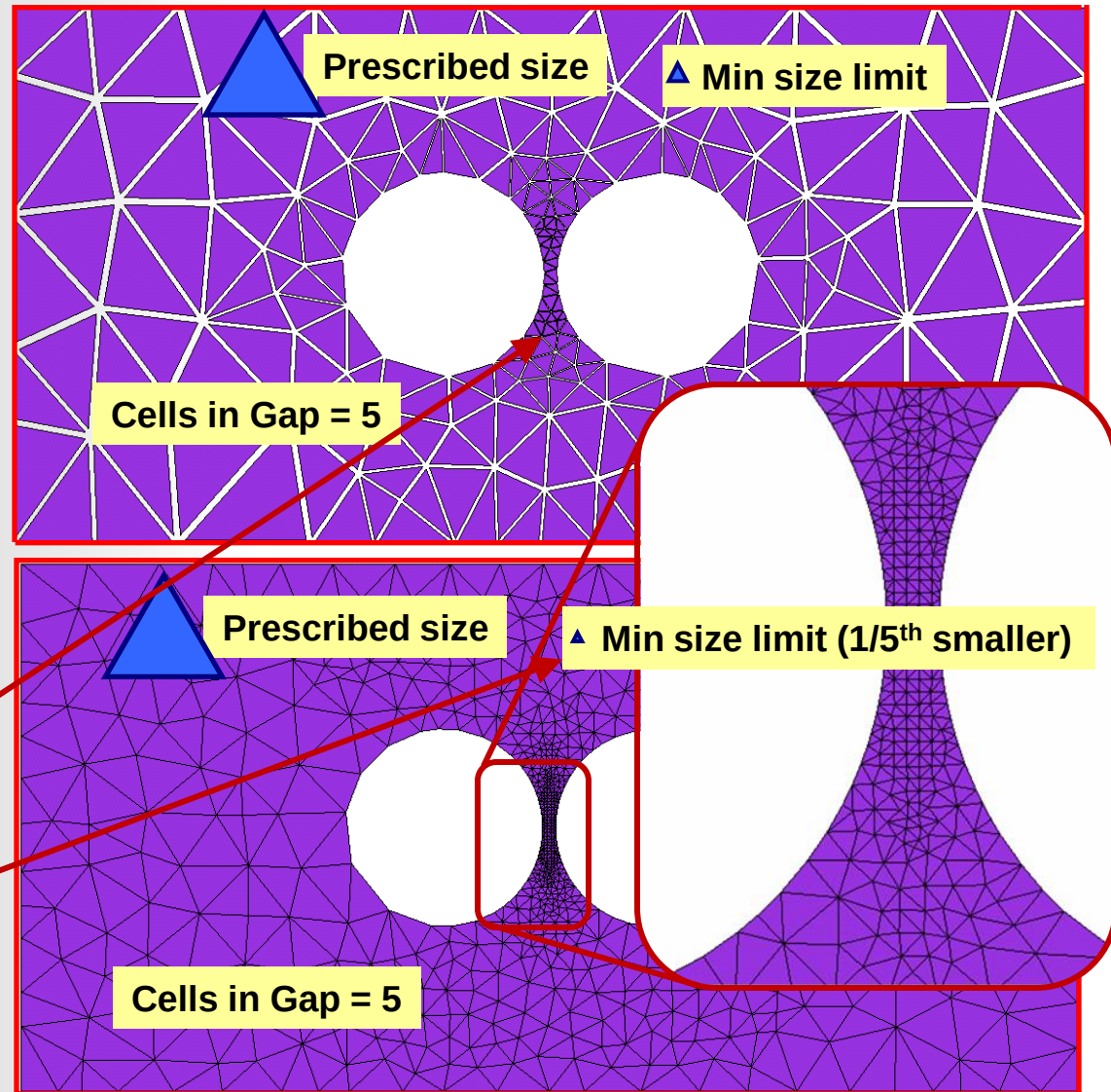
Proximity Based Refinement

- **Elements in Gap**

- Number of cells desired in narrow gaps
- The default is one element in the gap to just model the feature
 - Can specify more than one element in gaps
- Will not make elements smaller than the **Min size limit**

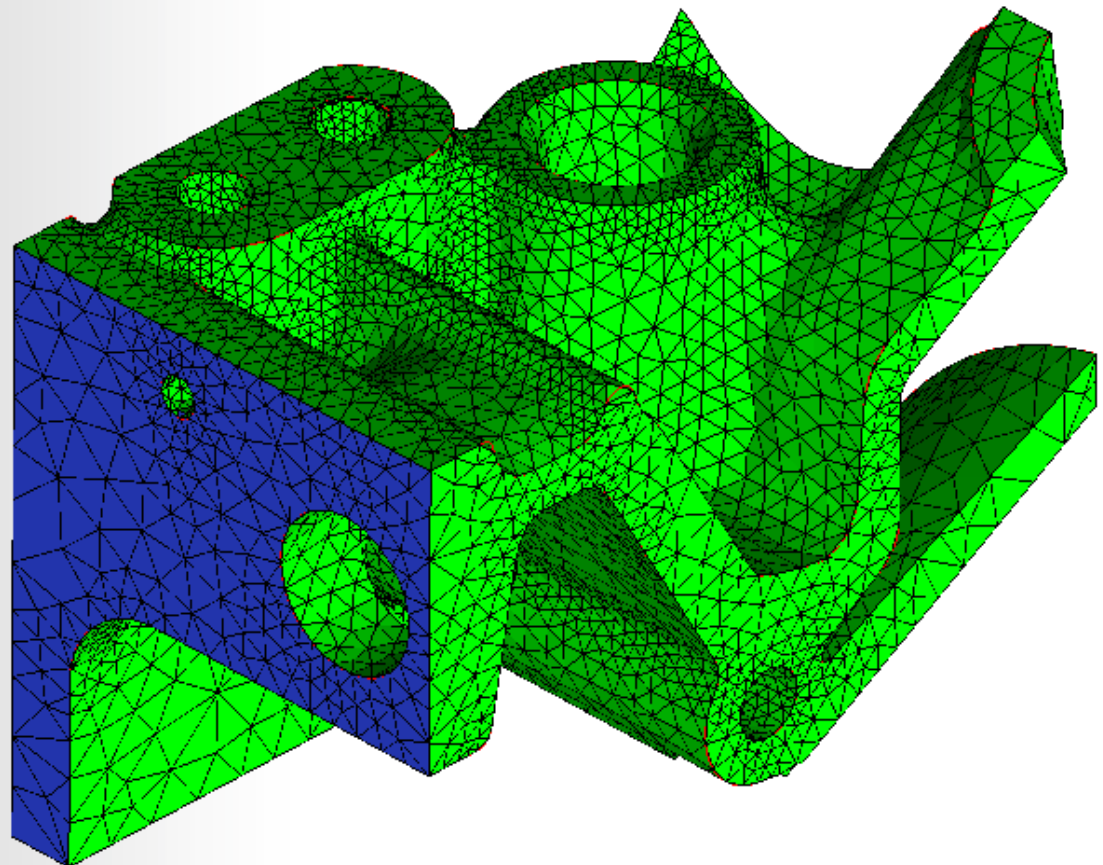
- **Example**

- Only one element in gap
- Can't go smaller than **Min size limit**
- Have to set smaller **Min size limit** to allow the 5 elements in gap



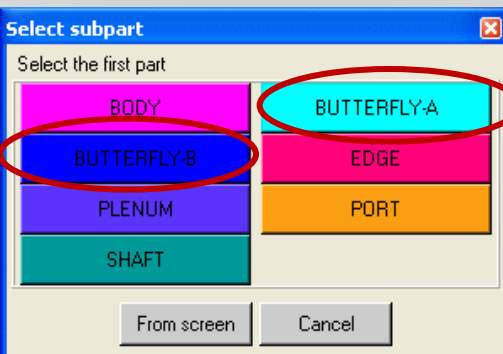
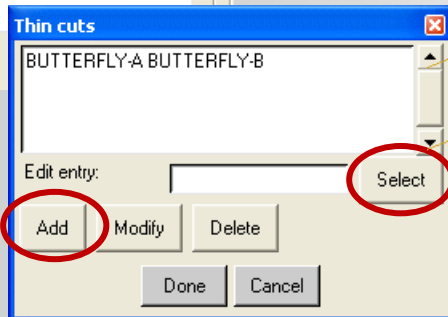
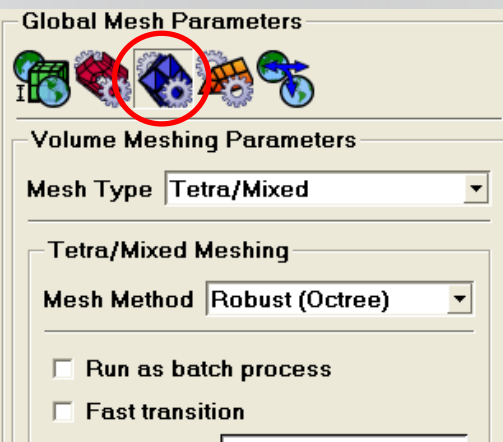
Workshop 4.1 – Engine Block Model

- Build diagnostic topology
- Octree mesh
- Mesh with surface sizes
- Smooth mesh
- View cut plane
- Curvature/proximity refinement
- Delaunay mesh

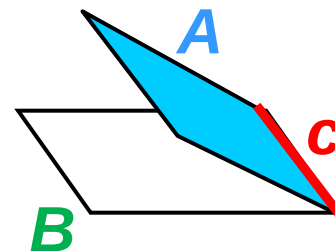
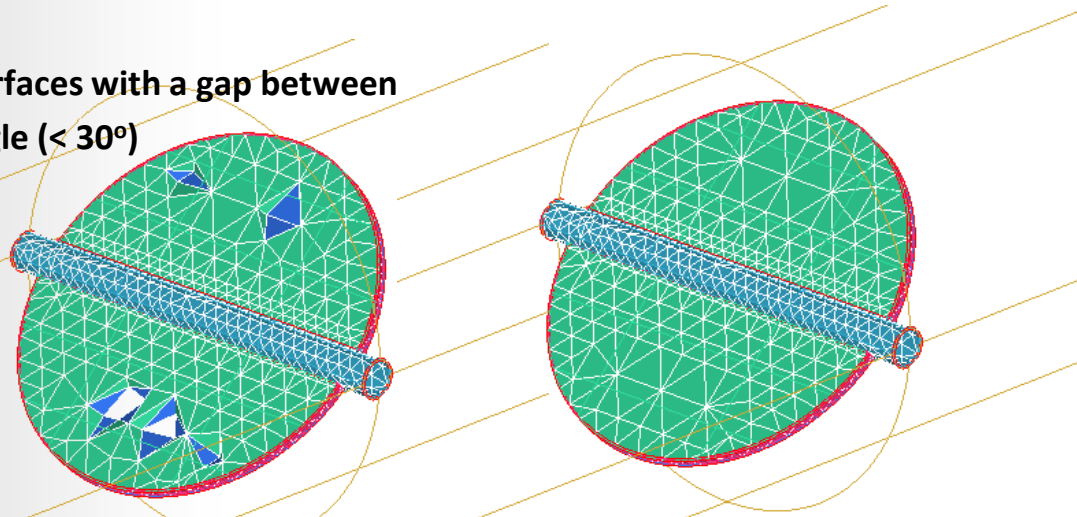


Thin Cuts

- Only works with Tetra Octree
- To avoid 'holes' in thin solids/narrow gaps when mesh size is much larger than gap (can handle up to 10 times gap distance)
- **Define thin cuts** by selecting two parts and then **Add** them to the list of defined Thin cuts
 - The two sides of the thin cut must be in different parts
- **2 types:**
 1. Offset surfaces with a gap between
 2. Small angle ($< 30^\circ$)



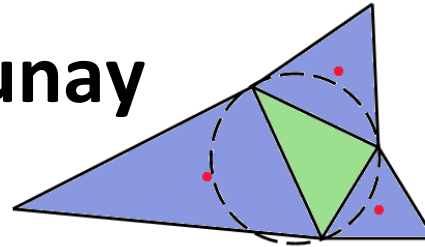
If the face of a tetra element has a surface/line node on part "A" then it may not have a surface/line node on part "B"



Note: If the surfaces of the two parts, **A** and **B**, meet, then the contact curve must be in a third part, **C**, or the thin cut will fail.

Mesh Methods - Delaunay

- Type - **Tetra/Mixed**
 - Method - **Quick (Delaunay)**
 - Start from a good quality, closed surface mesh



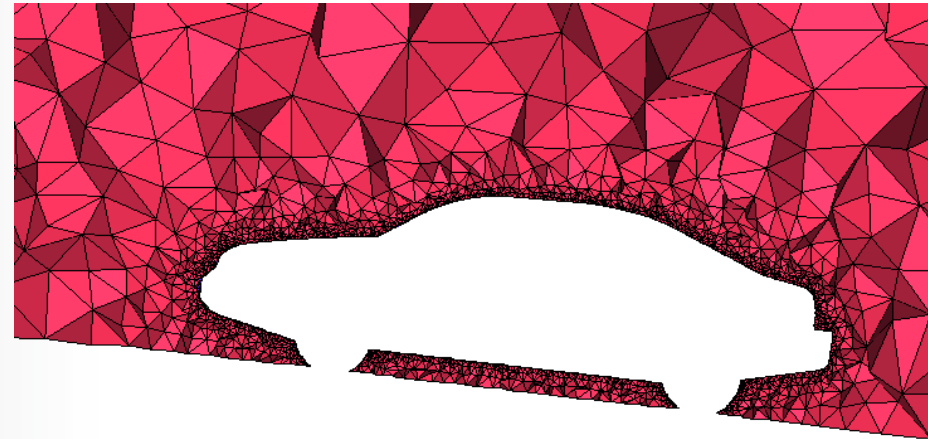
Initially distributes nodes so as the centroid of any tetra is outside the circumsphere of any neighboring tetra

Global Mesh Setup

Global Mesh Parameters



- Can be quad and tri elements
- From Shell Mesh
- From Octree
- From imported surface mesh



Tetra/Mixed Meshing

Mesh Method Quick (Delaunay)

Delaunay Scheme

Select

☐ Standard ☒ TGLib

☒ Use AF

Memory scaling factor 1

Spacing Scaling Factor 1

☐ Fill holes in volume mesh

☐ Mesh internal domains

☐ Flood fill after completion

☐ Verbose output

☒ Smooth mesh

Iterations 5

Min quality 0.4

• Setup Options:

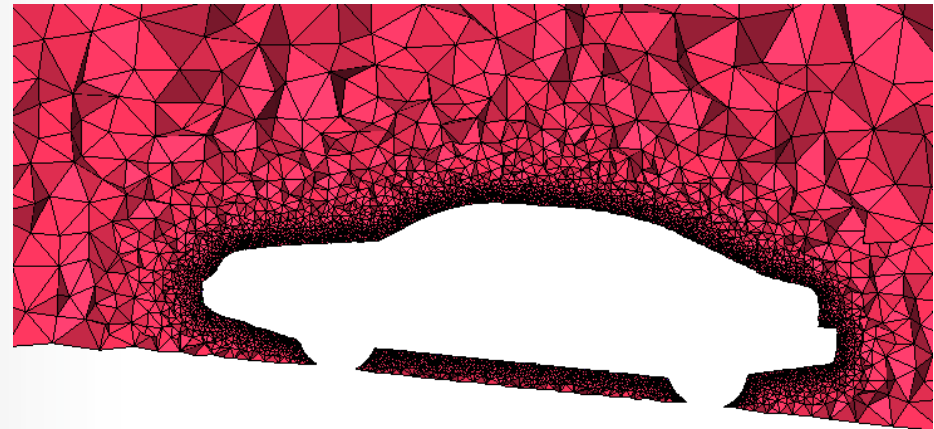
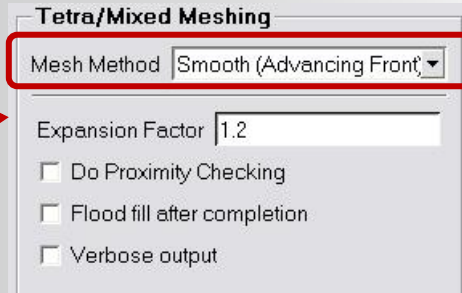
– **Delaunay Scheme**

- **Standard:** Delaunay scheme with a skewness-based refinement
- **TGLib:** TGrid Delaunay volume grid generation algorithm that utilizes a more gradual transition rate near the surface and faster towards the interior
- **Use AF:** TGrid Advancing Front Delaunay algorithm which has smoother transitions than the pure Delaunay algorithm.
- **Memory Scaling Factor:** To allocate more memory than originally
- **Spacing Scaling Factor:** Growth ratio from surface (1 – 1.5 typically)
- **Fill holes in volume mesh:** Use to fill holes/voids in existing volume mesh . E.g. if bad quality region is deleted
- **Mesh internal domains:** For multiple sets of closed volumes in one model
- **Flood fill after completion:** For multiple volumes – Will assign tetras within closed volume to **Part** designated by **Body** or **Material Point**
- **Verbose output:** For troubleshooting

Mesh Methods – Advancing Front

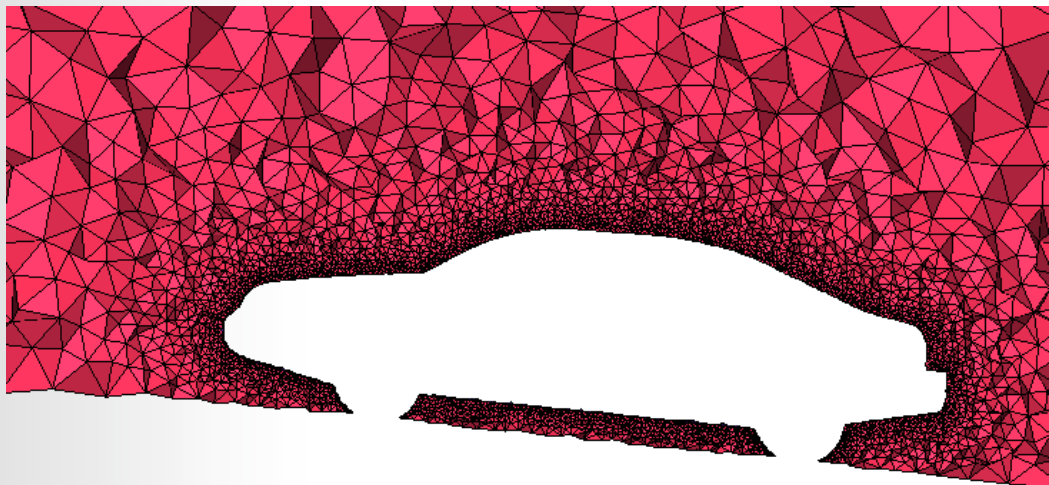
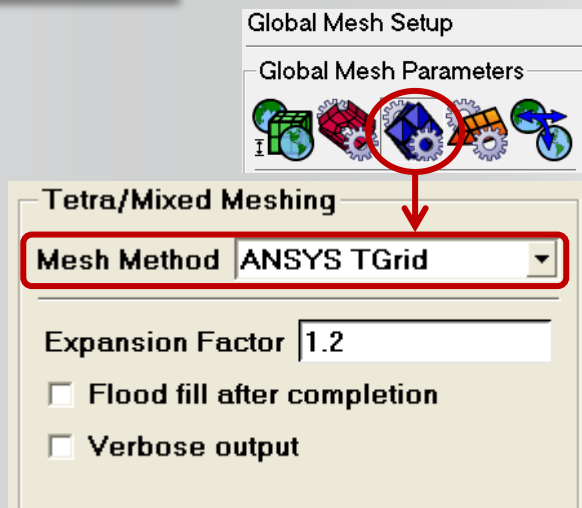
Global Mesh Setup

Global Mesh Parameters



- Type - **Tetra/Mixed**
 - Method - **Smooth (Advancing Front)**
 - Same as **Quick (Delaunay)** but
 - Uses advancing front method that marches tetras from surface into interior
 - Algorithm from GE/CFX
 - Results in more gradual change in element size
 - “Better” but finer mesh, more elements than Delaunay
 - Elements grow slowly for first few layers from surface, then growth rate increases into volume more
 - Input surface mesh has to be of fairly high quality
 - Setup Options: **Do Proximity Checking**
 - Check to properly fill small gaps
 - Longer run time
 - Can create pyramids from quads
 - Quads need to be a 10 aspect ratio or less
 - Delaunay can handle much higher quad aspect ratios
 - Respects densities

Mesh Methods – ANSYS TGrid

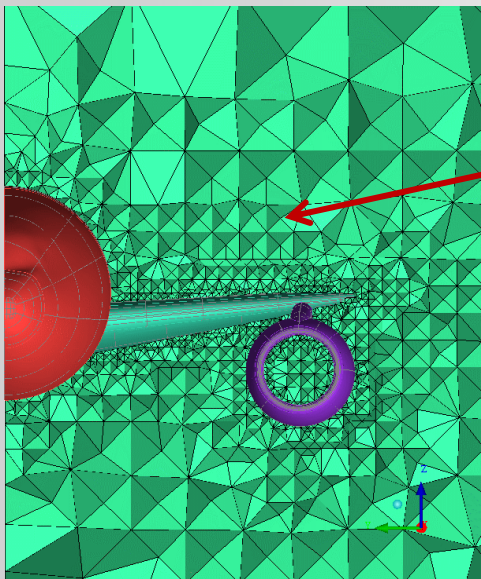


- Type - ***Tetra/Mixed***
 - ***ANSYS TGrid***

- Runs Tgrid through an extension module
- Good mesh quality
- Fast mesh generation
- Setup Options:
 - ***Flood fill after completion***: Same as octree Flood fill
 - ***Verbose output***: This option writes more messages to help in debugging any potential problem
- Will not respect densities
- Will not mesh to quads. It converts them to triangles
- Similar to Advancing Front, but does not group elements as close near surface

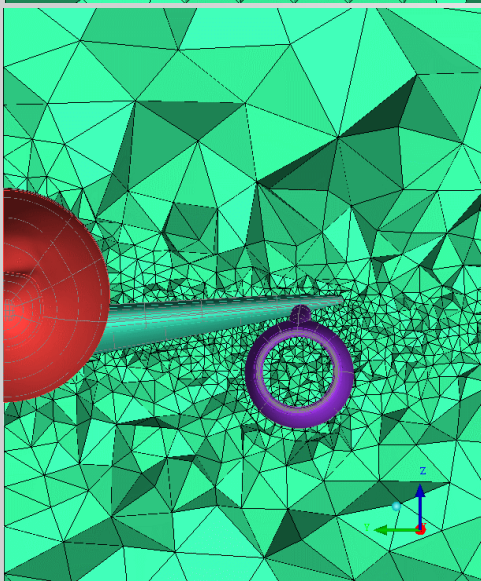
Comparison

Octree



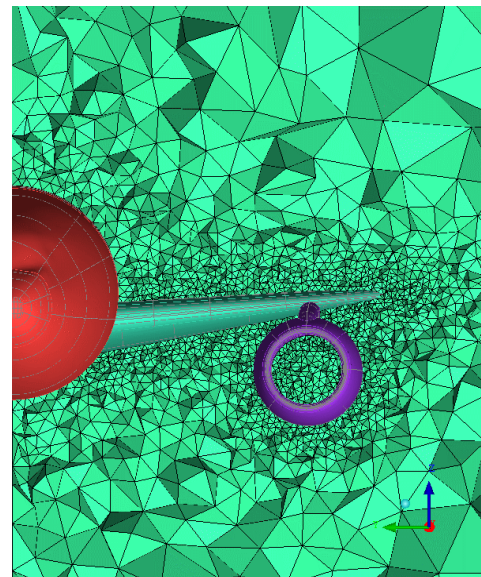
Note the 2:1 growth ratio and square volume elements of the octree method

Delaunay

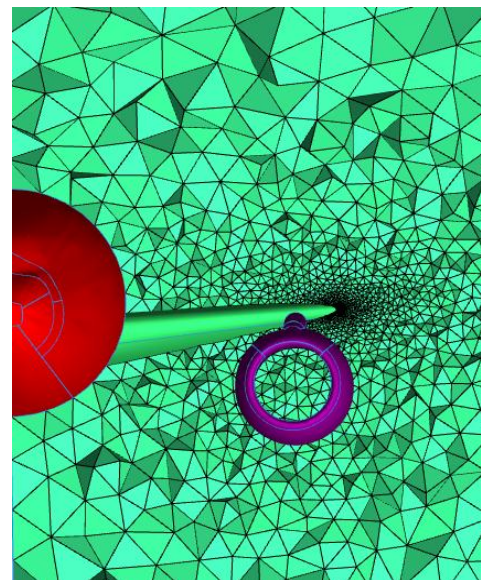


The 3 filling methods (Delaunay, Adv Front, Tgrid) allow a smoother growth (less than 2:1) than octree

Adv.front



ANSYS TGrid



Density Region



Model

- Geometry
- Subsets
- Points
- Curves
- Surfaces
- Parts

Create Density

Name: density.0

Size: 0.5

Ratio: 1.2

Width: 10

Density Location

From

☒ Points ☐ Entity bounds

Points: VERT162 VERT175

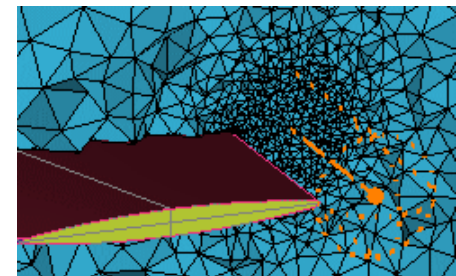
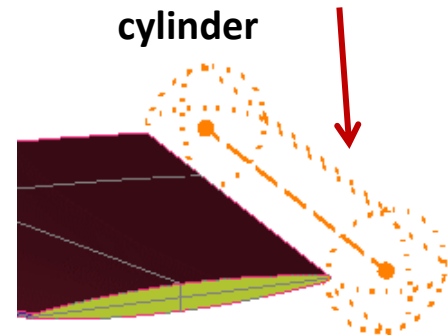
Apply OK Dismiss

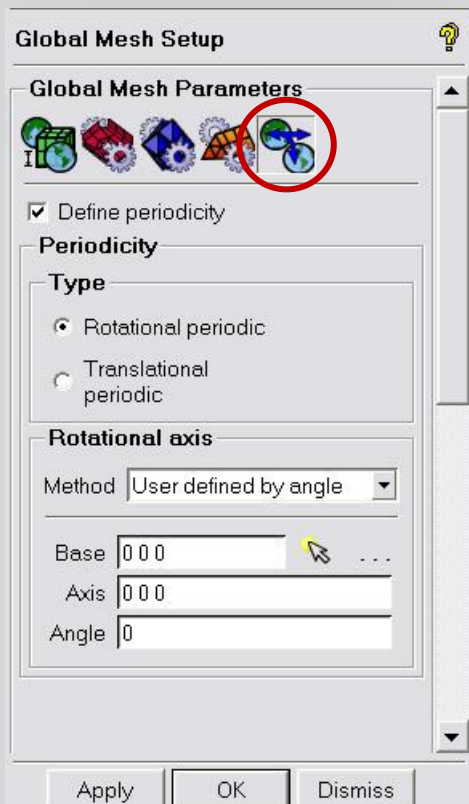
Densities are only used by tetra octree, Delaunay, and Advancing Front

- **Create Mesh Density**

- Define volumetric region with smaller mesh size where no geometry exists, e.g. wake region behind a wing
- Not actual geometry!
 - Mesh nodes not constrained to density object
 - Can intersect geometry
- Can create densities within densities
 - Always subdivides to SMALLEST set size
- Set **Size**
 - Max size within – multiplied by global **Scale Factor**
 - **Ratio** – expansion ratio away from density object
 - **Width** – Number of layers from object before mesh size is allowed to growth
- **Type**
 - **Points** – Select any number of points
 - **Size** and **Width** (number of layers) will determine “thickness” of volume if number of points selected is 1-3
 - 4-8 creates polyhedral volume
 - **Entity bounds** – define region by bounding box of selected entities

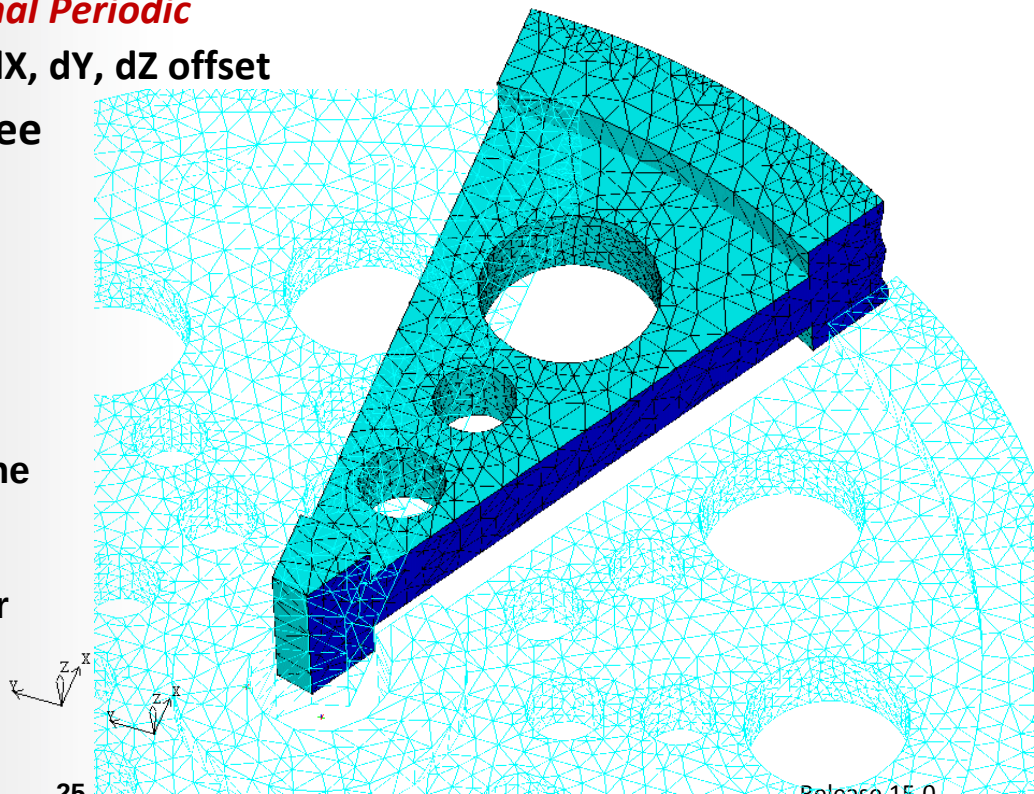
Density from 2 points makes a line. The **width** defines the radius of the cylinder





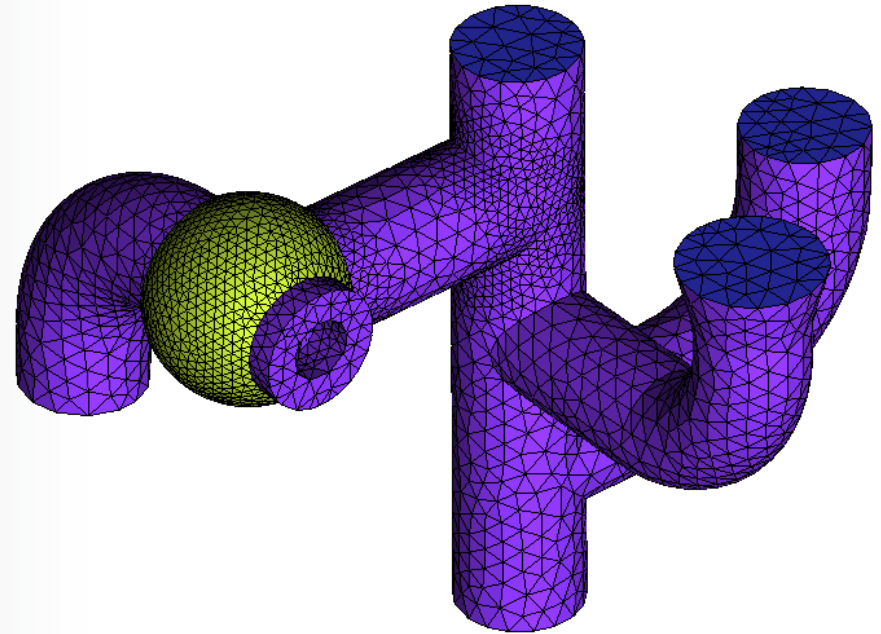
- **Define Periodicity**
- Forces mesh alignment across periodic sides
- For meshing only one section of symmetrically repeatable geometry
 - **Rotational Periodic**
 - Enter **Base**, **Axis**, and **Angle**
 - **Translational Periodic**
 - Enter dX, dY, dZ offset
- For tetra octree and hexa blocking only

Tip: Placing material point close to mid-plane makes tetra octree obey periodicity easier



Workshop 4.2 – Valve Model

- Build diagnostic topology
- Create parts, Create material point
- Set sizes – Global, Surface, Curves, density
- Octree mesh
- Compare density to width setting



Workshop 4.3 – Body Fitted Cartesian

- Import Design Modeler file into ICEM
- Create Parts from Subsets
- Global, Part mesh set up
- Using ANSYS ICEM CFD Hexa to create a Cartesian initial grid with biasing
- Computing the mesh
- Viewing cut-planes

