ICEM Mesh for CFD Analysis
Index

1. Introduction to ICEM
2. Geometry Handling
3. Shell Meshing
4. Volume Meshing
5. Prism Meshing
6. Mesh Preparation Before Output to Solver
7. Output to Solver
8. ICEM CFD Hexa
1. Introduction to ICEM
What is a Mesh?

- **Mesh**
  - Volume comprised of elements used to discretize a domain for numerical solution
    - Structural
    - Fluid dynamics
    - Electromagnetics
    - Other
  - Elements
    - 0D – Node element
      - Point mass
      - Constraint, load location
    - 1D – Lines
      - Bars, beams, rods, springs
      - 2D mesh boundary
    - 2D – Surface/Shell
      - Quads
      - Tris
      - Thin sheet modeling
      - 2D volume
      - 3D mesh boundary
    - 3D - Volume
      - Tetra
      - Pyramid
      - Penta (prism)
      - Hexa
      - Solid modeling
      - 3D fluid modeling
    - Formats
      - Unstructured
      - Block Structured
    - Nodes
      - Point locations of element corners

- **Elements**
  - 0D – Node element
    - Point mass
    - Constraint, load location
  - 1D – Lines
    - Bars, beams, rods, springs
    - 2D mesh boundary
  - 2D – Surface/Shell
    - Quads
    - Tris
    - Thin sheet modeling
    - 2D volume
    - 3D mesh boundary
  - 3D - Volume
    - Tetra
    - Pyramid
    - Penta (prism)
    - Hexa
    - Solid modeling
    - 3D fluid modeling
  - Formats
    - Unstructured
    - Block Structured
  - Nodes
    - Point locations of element corners
Ansys ICEM CFD Features

- Geometry Creation/Repair/Simplification
  - Including Mid-Plane Extractions/Extensions
  - Most geometry intended to be imported
- Powerful Meshing tools
  - Tetra/Prism from CAD and/or existing surface mesh
  - Shell meshing: structured, unstructured
  - Hex-dominant, swept, Structured hexa, Extruded quads, Body-fitted and stair-step Cartesian
  - Hybrid meshing (merging, multi-zone)
- Advanced mesh editing
- Solver Setup
- Output to 100+ Solvers
- Scripting … and much more…
GUI and Layout

utility Menu

Utility Icons

Model Tree

Function Tabs

Selection Toolbar

Data Entry Panel

Message Window

Histogram Window

Display Triad
File and Directory Structure

- Use of many files
  - Not one large common database
  - For faster input/output
- All files can optionally be associated within a Project
  - Establishes working directory
  - Settings (*.prj) file contains associated file names
- Primary file types:
  - Tetin (.tin): Geometry
    - Geometry entities and material points
    - Part associations
    - Global and entity mesh sizes
    - Created in Ansys ICEM CFD or CAD Interface
  - Domain file (.uns)
    - Unstructured mesh
  - Blocking file (.blk)
    - Blocking topology
  - Attribute file (.fbc, .atr)
    - Boundary conditions, local parameters & element types
  - Parameter file (.par)
    - Solver parameters & element types
  - Journal and replay file (.jrf, .rpl)
    - Record of performed operations (echo file)
Mouse Usage

- ‘Dynamic’ viewing mode (click and drag)
  - left: rotate (about a point)
  - middle: translate
  - right: zoom (up-down)
    - screen Z-axis rotation (sideways)
  - Wheel zoom
- Selection mode (click)
  - left: select (click and drag for box select)
  - middle: apply operation
  - right: unselect last selection

- $F9$ toggles the mouse control to Dynamic mode while in Select mode
  - \textit{Toggle Dynamics} button also does this
- Spaceball allows for dynamic motion even while in select mode
Utility Menus

File Menu
(file i/o)

Edit Menu

View Menu

Info Menu
(preferences)

Help Menu

<table>
<thead>
<tr>
<th>New Project...</th>
<th>Undo</th>
<th>Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Project...</td>
<td>Redo</td>
<td>Box Zoom</td>
</tr>
<tr>
<td>Save Project...</td>
<td>Clear Undo</td>
<td>Top</td>
</tr>
<tr>
<td>Save Project As...</td>
<td>Shell</td>
<td>Bottom</td>
</tr>
<tr>
<td>Close Project...</td>
<td>Facets -&gt; Mesh</td>
<td>Left</td>
</tr>
<tr>
<td>Change Working Dir...</td>
<td>Mesh -&gt; Facets</td>
<td>Right</td>
</tr>
<tr>
<td>Geometry</td>
<td>Structure</td>
<td>Front</td>
</tr>
<tr>
<td>Mesh</td>
<td>Struct mesh -&gt; CAD Surfaces</td>
<td>Back</td>
</tr>
<tr>
<td>Blocking</td>
<td>Struct mesh -&gt; Unstruct Mesh</td>
<td>Isometric</td>
</tr>
<tr>
<td>Attributes</td>
<td>Shrink transient file</td>
<td>View Control</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
<td>Save Picture</td>
</tr>
<tr>
<td>Cartesian</td>
<td></td>
<td>Mirrors and Replicates</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Import Geometry</th>
<th>Geometry Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Mesh</td>
<td>Surface Area</td>
</tr>
<tr>
<td>Export Geometry</td>
<td>Frontal Area</td>
</tr>
<tr>
<td>Export Mesh</td>
<td>Curve Length</td>
</tr>
<tr>
<td>Workbench Readers</td>
<td>Curve Direction</td>
</tr>
<tr>
<td>Replay Scripts</td>
<td>Mesh Info</td>
</tr>
<tr>
<td>Exit</td>
<td>Element Info</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Annotation...</th>
<th>ToolBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Marker</td>
<td>Project File</td>
</tr>
<tr>
<td>Clear Markers</td>
<td>Domain File</td>
</tr>
<tr>
<td>Mesh Cut Plane</td>
<td>Mesh Report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General</th>
<th>Help Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Toolbar</td>
</tr>
<tr>
<td>Display</td>
<td>Model</td>
</tr>
<tr>
<td>Speed</td>
<td>Geometry Options</td>
</tr>
<tr>
<td>Memory</td>
<td>Meshing Options</td>
</tr>
<tr>
<td>Lighting</td>
<td>Solvers</td>
</tr>
<tr>
<td>Background Style</td>
<td>Restore</td>
</tr>
<tr>
<td>Mouse Bindings/Spaceball</td>
<td>Reset</td>
</tr>
<tr>
<td>Selection</td>
<td></td>
</tr>
<tr>
<td>Remote</td>
<td></td>
</tr>
</tbody>
</table>

Installation & Licensing Guide
What's New
Legal Notices
Show Customer Number
About ANSYS ICEM CFD
File menu

• To open/save/close
  – Projects
    • Will open/save/close all associated files including
      • Geometry (*.tin)
      • Mesh (*.uns)
      • Attributes… (*.fbc, *.atr)
  – All file types can be opened/saved/closed independently
• Also to
  – Import/Export Geometry/Mesh
  – Invoke scripting
• Exit

• Several common functions are duplicated as utility icons:

Save frequently!
Other Commonly Used Utilities

- **Edit > Undo/Redo**

- **View**
  - **Fit**
    - Fit visible entities into screen
  - **Box Zoom**
  - **Standard views**
    - Top, Bottom, Left, etc.
    - Can also select X, Y, Z axis of display triad in lower right hand corner of main view screen to orient to standard views, e.g. selecting “X” will orient “right”
    - Isometric – select blue dot within triad

- **Measure**
  - **Distance**
  - **Angle**
  - **Location**

- **Local Coordinate System**
  - Used by:
    - Select location
    - Measuring
    - Node/point movement/creation
    - Alignment
    - Loads
    - Transformation

- **Surface display**
  - Wireframe
  - Solid
  - Transparent

- **Other Commonly Used Utilities**
  - Also here
Help

- Menu Driven
  - Searchable
  - Includes tutorials
  - Programmers guide (for ICEM CFD/Tcl scripting procedures)
- Hyper-link to specific topic

- Bubble explanation with cursor positioning
**Common Function Tabs**

**Geometry**
- Create/Modify geometry

**Mesh**
- Set mesh sizes, types and methods
- Set global mesh options
- Auto create Shell, Volume, Prism meshes

**Blocking**
- Initialize blocking
- Split/modify blocks
- Generate structured hexa mesh

**Edit Mesh**
- Check errors/problems, Smooth, Refine/Coarsen, Merge, repair mesh, Transform, etc.

**Output**
- Set Boundary Conditions and Parameters
- Write mesh for 100+ solvers.
Structural Function Tabs

- Only available when solver is set to Abaqus, Ansys, Autodyne, LS-Dyna, or Nastran
- *Settings*->*Product* must also be set to an FEA version

Properties
- Create, read, write out material properties
- Apply to geometry/elements

Constraints
- Set constraints, displacements, define contacts, initial velocity, rigid walls

Loads
- Set force, pressure and temperature loads

Solve options
- Set parameters, attributes, create subcases, write out input file, run solver
Selection Toolbar

- During select mode, popup selection toolbar appears
  - Some tools are common to all, others are contextual
  - Linked to select mode hotkeys
  - Filtering of entities and mass-selection methods

![Selection Toolbar Diagram]

- Geometry
  - Polygon
  - Select all
  - Cancel Only visible
  - Flood fill to Angle

- Mesh
  - Circle
  - Entire/Partial toggle
  - Set Flood Fill angle
  - All Shells

- Blocks
  - Toggle Dynamic Mode (F9)
  - By Subset
  - From Corners

- Faceted Geometry
  - Segments

- Select By Part
  - Entity Filter
  - Mesh attached to Geometry

- Toggle between mesh and geometry
  - In between segments
Model Tree

- To toggle on/off various sections of the model
- Main Categories are:
  - Geometry, Mesh, Blocking, Parts
  - Local Coord Systems, Element Properties, Connectors, Displacements, Loads and Material Properties
- Toggle check boxes to blank/unblank
  - Blanked/inactive
  - Visible/active
  - Partially visible/active: some sub members turned on, some turned off
- Click on plus sign to expand tree
  - Expose sub members
- Right mouse click for display options
Model Tree: Parts

- **Parts**
  - Grouping of mesh, geometry, and blocking entities
    - Based on boundary condition/property
    - Based on mesh size (can set mesh size by part)
    - Based on material property
    - Just to partition large model
  - Select to blank/unblank all entities within part
  - Color coded: Part name matches entity screen display
    - **Right Mouse Button** on *Parts* to access:
      - *Create Part*
      - *Create Assembly*
      - *Delete Empty Parts*
      - *Etc.*
  - **RMB** on specific part names allows options to modify or delete these parts
  - **Properties** are shown as a sub branch of the part
    - Double Left Click or RMB > Modify to modify element properties
Typical ICEM CFD Workflow:
1. Create/open new project
2. Import/Create geometry
3. Build topology/Clean geometry/Create geometry
4. Mesh model (Possibly Hex Blocking)
5. Check/edit mesh
6. Output to Solver

General Order of Workflow
Accessing from Workbench

- Ansys ICEM CFD 14.0 is not fully linked inside Workbench
  - Export files from Mechanical Model (Simulation) or Meshing Application to open in ICEM CFD

- Some ICEM CFD capabilities have been integrated into the Meshing Application
  - Tetra octree (patch independent)
  - 3D blocking fill (Multizone)
  - Autoblock (2D, uniform quad)
  - Body fitted cartesian
Workbench Interactive Link

- Ansys ICEM CFD can be accessed from Workbench from certain mesh methods
  - Insert a meshing method
    - MultiZone
    - Patch Independent tetrahedrons
- Set Write ICEM CFD Files to Interactive
- Generate mesh
- Edit or remesh within ICEM CFD, save project, then exit ICEM CFD
  - Don’t edit geometry in ICEM CFD
2. Geometry Handling
Geometry handling

ANSYS ICEM CFD was designed to mainly import geometry, not create complicated geometries, although many geometry tools are provided.

An accurate solution reflects the underlying geometry. To get such, ICEM CFD provides:

- Geometry import
  - From CAD package
  - 3rd party formats (step, acis, etc…)
  - Via Workbench/ Design Modeler

- Surface geometry kernel
  - Imported solids are converted to surfaces

- Many internal CAD tools
  - Geometry creation
  - Geometry modification
  - Geometry fixing

This Jet engine model was built solely with ICEM CFD geometry tools.
CAD from just about any source

- **Workbench Readers** – for most CAD imports
  - Anything that Workbench can import can also be imported into ICEM CFD using Workbench readers
  - Requires a Workbench installation!

- **3rd-party import**
  - ACIS (.sat)  •  Parasolid
  - DWG/DXF  •  STEP/IGES  •  GEMS

- **Direct CAD Interfaces**
  - Legacy interfaces which are not updated. Use *Workbench Readers* instead for current CAD versions
  - Set up ICEMCFD/AI*E meshing requirements within CAD environment
    - Saved within CAD part for parametric geometry changes
  - Directly write out ICEM formatted geometry (tetin file)
    - No 3rd party exchange (clean!)
  - ProE, Unigraphics, Solidworks, Catia V4, IDEAS (IDI)
  - ProE, UG, and Solidworks imports require CAD libraries; CAD software and licensing must be available
When CAD is not available, an old legacy model or x-ray scan of the part can be imported as geometry. This input is a collection of facets (triangulated surfaces).

- Faceted Data
  - Nastran
  - Patran
  - STL (most common)
  - VRML
  - Other solver formats (indirectly from mesh conversion)

- Formatted Point Data
  - Auto curve/surface creation from regular table of points
Open Geometry

- Geometry saved as “tetin” (*.tin file)
  - Legacy name as an abbreviation of “tetra input.”
  - Surface geometry kernel
    - Any imported solid models are represented as a series of watertight surfaces
  - Surfaces are internally represented as triangulated data
    - Resolution or approximation of true bspline surface data set by *Triangulation Tolerance* in *settings>*model
      - Smaller value = better resolution
      - 0.001 works best for most models
      - Use a high tri tolerance to work with a large model, but lower the tolerance when it comes time to compute the mesh
      - Not used if surfaces are already facetized (e.g. STL, VRML)
Geometry Creation Tools

- Screen Select
- Explicit Coordinates
- Base Point and Delta
- Center of 3 Points/Arc
- Based on 2 Locations
- Curve Ends
- Curve-Curve Intersection
- Parameter along a Curve
- Project Point to Curve
- Project Point to Surface

- From Points
- Arc Through 3 Points
- Arc from Center Point/2 Points on Plane
- Surface Parameter
- Surface-Surface Intersection
- Project Curve on Surface
- Segment Curve
- Concatenate Curves
- Surface Boundary Extraction
- Modify Curves
- Create Midline
- Create Section Curves

- From Curves
- Curve Driven
- Sweep Surface
- Surface of Revolution
- Loft Surface Over Several Curves
- Offset Surface
- Midsurface
- Segment/Trim Surface
- Merge/Reapproximate Surface

- Untrim Surface
- Curtain Surface
- Extend Surface
- Geometry Simplification
  - Convex Hull
  - Cartesian Shrinkwrap
- Create Std Geometry
  - Sphere
  - Box
  - Cylinder
  - Plane
  - Disc
  - Trim normal to curve
Create Body

• 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
  – 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
Faceted Geometry Handling

- Convert from B-spline
- Create Curve
- Move nodes
- Merge nodes
- Create segment
- Delete segment
- Split segment
- Restrict segments
- Move to new curve
- Move to existing curve

- Convert from B-spline
- Coarsen Surface
- Create new Surface
- Merge Edges
- Split Edges
- Swap Edges
- Move Nodes
- Merge Nodes

- Create Triangles
- Delete Triangles
- Split Triangles
- Restrict Triangles
- Delete Triangles
- Move to new Surface
- Move to new Surface
- Merge Surfaces

- Align Edge to Curve
- Close Faceted Holes
- Trim by Screen Loop
- Trim by Surface Loop
- Repair Surface
- Create Curve

Facetted (triangulated) surfaces
Geometry Handling

Repair Geometry

- Build Diagnostic Topology
- Check Geometry
- Close Holes
- Remove Holes
- Stitch/Match Edges
- Split Folded Surfaces
- Adjust varying Thickness
- Modify surface normals
- Bolt hole detection
- Button detection
- Fillet detection

Transformation Tools

- Translate
- Rotate
- Mirror
- Scale
- Translate & Rotate
  - Three Points
  - Curve to Curve

Restore Dormant Entity
- Curves/points originally made inactive - ignored by meshing tools
- Restore to activate again - seen as constraints by meshing tools

- Delete
- Points
- Curves
- Surfaces
- Bodies
- Any Entity

Build topology with filtering
Building Topology – Determine Connectivity

- **Geometry -> Repair Geometry -> Build Diagnostic Topology**
- To diagnose potential geometry problems
  - Shows potential leakage (tetra octree) before meshing
  - Shows where surface mesh may not be connected
  - Patch dependent surface mesher requires build topology
- **Tolerance**
  - Specifies allowable gap between surfaces
  - Size should be set reasonably to ignore small gaps, but not ignore leakage (tetra octree) or remove important features
  - Default is $1/2500^{th}$ of the diagonal of the bounding box
  - Connectivity is set up between surface edges that meet within the tolerance
  - Filtering should be off when using to determine connectivity

![Diagram of Edge 1 and Edge 2 with Tolerance](image-url)
Building Topology – Color Coding

Color coding
- Topology curves are color coded to indicate their surface connection status
  - green = unconnected, yellow = single, red = double, blue = multiple, Grey = dormant (filtered out)
  - Turn color coding off/on in Model tree > Geometry > Curves > Color by count
  - Red curves indicate two surfaces meet within the tolerance, This is what you want for a solid model.
  - Yellow curves will usually indicate some repair is required

Can you spot the hole in the solid?

Now you can find the hole

Yellow curves indicate that the surface is probably missing or the gap is greater than the tolerance

Red curves indicate that surfaces meet within the tolerance setting
Build Topology – Extract Curves and Points

- Automatically extracts curves and points from the surfaces
  - **Filter by angle** (default 30 degrees)
    - **Filter Points**: Points between two curves whose tangency is below the feature angle will be “filtered out” (made dormant)
    - **Filter Curves**: Curves between two surfaces whose tangency is below the feature angle will be “filtered out” (made dormant)

Tetra octree and patch dependent surface mesher enforce nodes on the curves

Needs smaller mesh size at fillets
Build Topology – Segment Surfaces

- Automatically segments all surfaces where curves either make a complete loop on the surface or span across the surface.
- Turn *Split surface at T-connections* off to turn off segmenting.

You can then delete any surfaces you don’t want.

Check off to disable segmenting.

Build topology
Set adequate tolerance!

- Example: some multiple (blue) edges. This indicates that more than two surfaces meet within the tolerance setting.
- Turning on the surfaces reveals one surface is now missing.
- In this case, the tolerance (0.2) was set to greater than the thickness (0.1). One of the surfaces was seen as a “duplicate” within the tolerance and removed. UNDO

You will need to exercise care not to damage your model with build topology

- Too small is safer but indicates more gaps
- Too big can alter the model in bad ways

- Rule of thumb: tolerance should be about 1/10th smallest foreseen mesh size or smallest feature that you wish to capture
- Build topology will delete duplicate geometry because its tolerance is zero
Building Topology – Other Options

- **New Part Name**
  - **Inherit Part:** Default: new curves and points will inherit the part names from surfaces they are extracted from
    - Check off Inherit Part to type a new name or choose from the list

- **Single curve cleanup**
  - Merges single edge curves with a second tolerance while resolving sliver surfaces (normally larger than base tolerance)

- **Split Surface at T-connections**
  - Resulting mesh will conform to common edge even though the surface is not split into two separate surfaces
  - Will also split a surface into separate surfaces if the curves form a closed loop or span across the surface

- **Split Surface at Interior Curves**
  - Surfaces trimmed along curves that don’t span surface or form a closed loop
  - Resulting mesh will conform to curve
Building Topology – Other Options

– Method
  • All parts, default method
  • Only visible parts
    – Build topology is only run on active Parts in the model tree
    – Inactive Parts are not affected

– Selection
  – Build topology on one or more surface entities

– Part by part
  • Build topology is run on one part at a time
  • Use with assemblies to keep parts separate
  • Otherwise build topology may fix gaps, create T-connections or remove duplicates across Parts

– Delete unattached curves and points
  • Removes unattached curves (green) and points after running build topology
  • Easy clean-up of unwanted curves/points
  • Users may, however, wish to keep these curves/points for construction purposes (turn option off)
3. Shell Meshing
Introduction to Shell Meshing

• Usages of shell meshing:
  – Thin sheet solid modeling (FEA) – stamped parts
  – 2D cross sectional analysis (CFD)
  – Input for volume meshing (FEA/CFD) – Delaunay, Advancing Front, T-grid
    – Filling a surface mesh is faster than tetra octree but requires well-connected geometry

• Procedure
  – First need to decide mesh setup parameters
    • Mesh method
      – Algorithm used to create mesh
    • Mesh type
      – quad/tri/mix
    • Mesh sizes
      – Small enough to capture physics, important features
      – Large enough to reduce grid size (number of elements)
        » Memory limitations
        » Faster mesh/solver run
      – Set mesh sizes on parts, surfaces, and/or curves
      – Based on edge length
    – Can have different types/methods set on different surfaces
Global Mesh Setup

- Mesh Setup Icons

Global Mesh Setup
- To change defaults globally for size, method and type
- For entire model
- For Shells
- For Volume
- For Prism
- To set periodicity

Parameters relative to scale factor
- Max size
- Min size limit
- Max deviation

Global Mesh Size
- For entire model
- Scale factor
  - Global setting by which many local settings are multiplied
  - Good for scaling overall mesh
- Global Element Seed Size
  - Maximum possible element size in model
  - Default size if don’t wish to set local sizes
- Curvature/Proximity Based Refinement
  - Automatically creates smaller element size to better capture geometry
  - Only for Patch Independent method and tetra octree
Global Shell Meshing Parameters

- **Shell Mesh Setup**
  - From **Global Mesh Setup** tab
  - Set surface mesh parameters globally
    - Defaults for the selected mesh method
  - **Methods**
    - **Autoblock**
    - **Patch dependent**
    - **Patch independent**
    - **Shrinkwrap**
    - **Delaunay**
  - **Type**
    - All Tri, Quad w/one tri, Quad dominant, All quad
    - Options for different methods
    - Global types and methods can be overridden by
      - **Surface Mesh Setup**
        - Local settings
      - **Compute Mesh**
Quad layers grown from curves (e.g. rings around holes), use these 3 parameters:

- **Height**: First layer quad height on curves
- **Height ratio**: growth ratio which determines the heights of each subsequent layer
- **Num layers**: Number of rings/inflation layers

For quad layers, the minimum required to be set is **height** (for 1 layer) or **numlayers** (height = max. size)

If done in the Part Mesh Setup spreadsheet you must toggle on **Apply inflation parameters to curves**
Local Surface Mesh Setup

- **Surface Mesh Setup**
  - Same parameters as part mesh setup but also includes:
    - *Mesh type*
    - *Mesh method*
  - Select surfaces first from screen, set sizes/parameters and **Apply**
  - Mesh method/type will override global shell mesh settings for selected surface(s)
  - Will override **Part Mesh Setup** settings if set afterward
  - **Display**
    - Right mouse, select in Model tree on **Surfaces > Tetra/Hexa Sizes**
      - 1 Icon appears for each surface
      - Gives you a visual estimate of prescribed max. size
Local Curve Mesh Setup – General

- **Curve Mesh Setup**
  - **General**
    - Same as *Surface Mesh Setup*
    - But also can prescribe *Number of nodes*
      - Instead of element size
    - Also includes node biasing along curves
      - Initial spacing from either curve end
      - Bunching laws
      - Expansion ratios from either curve end
      - Matching of node spacing to adjacent curves
      - For a better description, refer to the *Hexa chapter – Edge Parameters*
    - Select curves first, middle mouse to accept selection, then type in parameters/sizes - *Apply*

Display
- Right mouse select in Model Tree, *Curves -> Curve Tetra/Hexa Sizes* or
- **Curve Node Spacing**

Arrow shows side 1 and side 2

Node spacing

Tetra sizes
Local Curve Mesh Setup – Dynamic and Copy

- **Curve Mesh Setup**
  - **Dynamic**
    - Adjust mesh parameters on screen
    - Interactively toggle displayed values near curve with left (to increase)/right mouse (to decrease) keys
  - **Copy Parameters**
    - Copy parameters set on one curve to others
    - e.g. parallel curves downstream
  - **Curve Mesh Setup** will override **Part Mesh Setup** parameters if set afterward
Mesh Methods

Algorithm used to create mesh

- **Patch Dependent**
  - Based on loops of curves surrounding patches
  - Best for capturing surface details and creating quad dominant mesh with good quality

- **Patch Independent**
  - Robust octree algorithm
  - Good for dirty geometry, ignoring small features, gaps, holes

- **Autoblock**
  - Based on 2D orthogonal blocks
  - Best for mapped meshing, mesh follows contours of geometry

- **Shrinkwrap**
  - Automatic defeaturing
  - Quick Cartesian algorithm
  - Allows ignoring of larger features, gaps and holes

- **Delauney** (beta options)
  - Allows for transition in mesh size
    - Coarser towards surface interior
  - Tri only

- **Set in Global Mesh Setup or locally using Surface Mesh Setup**
Patch Dependent Method

- Patch defined by a closed loop of curves
  - Typically each surface defines a patch
    - Loop defined by boundary curves
    - Curves automatically created by Build Diagnostic Topology - a must!
  - Can remove or filter out curves to define multi-surface patches
    - Delete curves
    - Turn on filter points/curves when building topology
- Only uses curve sizes (curve nodes seed loop perimeter)
- Paving algorithm used to fill interior of loop
  - Interior nodes typically projected to surface
  - Adjacent loops share nodes at common edge making mesh conformal throughout
- Default method, fastest method

Build topology MUST be done first to build surface connectivity and curves.
**Patch Dependent – Common Options**

- All method options set from *Global Mesh Setup -> Shell Meshing Parameters* section
- **General**
  - **Ignore Size**
    - Small features, such as sliver surfaces smaller than defined value are ignored. Merges loops behind the scenes
    - Will override max. size setting if smaller
  - **Respect line elements**
    - Line elements (bars) on existing mesh are respected
    - Maintains conformal mesh between newly created mesh and existing mesh on adjacent surfaces

Surrounding mesh done afterwards is conformal to existing mesh

Sliver Surface, 0.6 mm wide

Ignore size = 1, Sliver surface is ignored
### Patch Dependent Mesher - Boundary option

<table>
<thead>
<tr>
<th>Global Mesh Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Global Mesh Parameters" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shell Meshing Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2" alt="Shell Meshing Parameters" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore size: 0.1</td>
</tr>
<tr>
<td><img src="image3" alt="General" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4" alt="Boundary" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interior</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="Interior" /></td>
</tr>
</tbody>
</table>

- **Boundary**
  - **Protect given line elements**
    - Keeps existing line elements which are smaller than the *Ignore size*.
    - Grayed out unless *Respect line elements* is on.
  - **Smooth boundaries**
    - Smoothes the mesh boundaries after mesh generation. May not respect the initial node spacing set on curves.
- **Offset type**
- **Interior**
  - **Force mapping**
    - Forces mapped mesh on regular (4 sided) surfaces to desired degree (0-1).
    - Adjusts the number of nodes on opposite sides (0.2 = change number nodes by 20%).
  - **Project to surfaces**
    - Interior nodes project to surface rather than interpolate position.
  - **Adapt mesh interior**
    - Allows transition to larger element size in the interior of the surface (uses surface max size).

Would require too many nodes increased from original setting based on force mapping setting.
Patch Dependent Mesher - Repair option

- **Repair**
  - **Try harder**
    - For loops that fail with requested paving algorithm
    - Levels (0-3) to make further attempts to create grid
    - **0** - No further attempts, failed surface(s) marked and put into a subset
    - **1** - Simple triangulation of surface, converted to requested type
    - **2** - Same as 1, but dormant curves activated
    - **3** - Run octree, same as patch independent
  - **Improvement Level**
    - Levels (0-3) to improve mesh quality
    - **0** - Laplace smoothing only
    - **1** - STL tri mode, with conversion to quads (if requested)
    - **2** - tri to quad conversion, splitting of bad quads
    - **3** - allow nodes to move along boundary

- Other options and fuller descriptions may be found in the *Help* menu.
**Patch Independent**

- Uses robust Octree method
  - Volumetric tetra elements created around geometry
  - Faces mapped to surfaces
  - Only surface mesh is retained
  - Discussed in more detail in *Volume Mesh* lecture
- Mesh sizes defined on surfaces and curves
- Can walk over details, thin gaps, small holes
  - Relative to mesh size
- Nodes and edges don’t have to be lined up with surface edges
  - Only lined up where curves exist

Nearest nodes projected to surface and only surface mesh is left

Matches up with previously meshed surfaces

Volume around is first meshed
Autoblock

- Surface (2D) blocks are created automatically from each surface
  - Internal, blocks aren’t recognized or visible
  - For further description of blocking, refer to Hexa chapter
- Blocks structurally connected
  - Conformal mesh between blocks and surfaces
- Structured blocks result from 4-sided surfaces
  - For regular or four-sided blocks, structured (mapped) mesh follows contours of geometry
- Best for recognizing rounds or fillets
- Irregular (non-4 sided) or trimmed surface patches may be unstructured
Mesh sizes set on surfaces or curves
- Options
  - Ignore size
  - Mapped or free (unstructured as in patch dependent)
- **Build Topology** MUST be run beforehand
Shrinkwrap

- Cartesian (rectilinear) method
  - Can ignore larger features, gaps, holes
- Cube faces partially projected to geometry
- Quickest method for creating surface mesh
- Can’t recognize sharp features
  - Currently in development phase
- Best for “wrapping” geometry
  - Quick and dirty surface meshing of complex geometries
- For “solid” models
  - Not recommended for thin sheet solids
- Options
  - No. of smooth iterations
    - To improve grid quality
  - Surface projection factor
    - To fully project to original geometry (1.0), to not project at all (0.0), or partially (0.0 < factor < 1.0)
Mesh Types

- Set in **Global Mesh Setup > Shell Mesh Parameters** or **Surface Mesh Setup** (local upon selected surface entities)
  - Global defaults overridden by local settings or **Compute Mesh** options

- **All Tri**

- **Quad w/one Tri**
  - Almost all quad except with one tri per surface
  - Single tri allows transition between uneven mesh distribution on loop edges
  - Where pure quad will fail

- **Quad Dominant**
  - Allows for several transition triangles
  - Very useful in surface meshing complicated surfaces where a pure quad mesh may have poor quality

- **All Quad**

These mesh types will look different with the different mesh methods.
Compute Mesh

- Once sizes, methods and types are set – ready to compute!
- **Select** *Mesh > Compute Mesh > Surface Mesh Only*
  - Most of the time can just select *Compute* at bottom of panel which will create shell mesh for entire model (*Input = All*)
  - Other options
    - *Overwrite Surface Preset/Default Mesh Type/Method*
      - To quickly override global and local settings
      - Avoid going back to other *Mesh Setup* menus to change parameters
  - **Input**
    - Can mesh *All* (default – entire model)
    - *Visible* – only visibly displayed surfaces/geometry
    - *Part by Part*
      - Parts meshed separately
      - Mesh will be non-conformal between parts
    - *From Screen*
      - Select entities to mesh from screen
4. Volume Meshing
Introduction to Volume Meshing

- To automatically create 3D elements to fill volumetric domain
  - Generally termed “unstructured”
    - Mainly tetra
  - Full 3D analysis
    - Where 2D approximations don’t tell the full story
  - Internal/External flow simulation
  - Structural solid modeling
  - Thermal stress
  - Many more!
- Standard procedures
  - Start from just geometry
    - Octree tetra
      - Robust
      - Walk over features
    - Cartesian
      - Fastest
    - Have to set sizes
  - Start from existing shell mesh
    - Delauney/T-grid
      - Quick
    - Advancing Front
      - Smoother gradients, size transition
    - Hex Core
    - Hex Dominant
  - Both geometry and shell mesh
    - Octree tetra
      - Portions of model already meshed
      - Set sizes on rest
    - Prism layers
      - “Prism”
General Procedure

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

- Set mesh sizes
  - Globally
    - As in Shell Meshing
  - Locally
    - Part/Surface/Curve Mesh Setup
    - As in Shell Meshing
    - For From geometry:
      - Octree
      - Cartesian

- Define volumetric region
  - 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 following tetra creation
Define Volumetric Domain

- Optional
  - Recommended for complex geometries
  - Or multiple volumes

- **Geometry -> Create Body**

- **Material Point**
  - Centroid of 2 points
    - Select any two locations whose mid-point is within volume
    - Preferred, because more robust than By Topology method
  - At specified point
    - Define volume region at a “point” within volume

- **By Topology**
  - Defines volume region by set of closed surfaces
  - Must first *Build Diagnostic Topology* to determine connectivity
    - Will fail if gaps/holes in body
  - **Entire model**
    - Automatically define all volumes
  - **Selected surfaces**
    - User selects surfaces that form a closed volume
**Mesh Types**

- **Tetra/mixed**
  - Most used type
  - Pure tetra
  - With prism layers
    - Prisms from tri surface mesh
    - Hexas from quad surface mesh
    - Tetra and/or hex core filling interior
    - Pyramids to cap off any quad faces from prism sides, hex core, or hex prism layers
  - With hex core
    - Available in Cartesian type too
    - Hexa filling majority volume
    - Tetra (from Delauney algorithm) used to fill between surface or top of prism layers and hex core
    - Pyramids to make conformal between tetra and hex quad faces
  - Hybrid mesh can be created by merging with a structured hex mesh
**Mesh Types - Continued**

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

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

- Type - *Tetra/Mixed*
  - Method - *Robust (Octree)*
    - Same as Shell Meshing > Patch Independent
      - Retains volumetric tetras
    - Good choice for complex and/or dirty geometry
    - Good if you don’t want to spend too much time with geometry cleanup
    - Good if you don’t want to spend too much time with detailed shell meshing
    - Good if you don’t want to spend time defeaturing geometry
    - 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 Method Characteristics

- Octree process
  - Volume first generated independent of surface model
  - Tetras divided near regions where sizes are set smaller
  - Nodes are projected to model surfaces, curves and points
  - 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

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

The Octree process creates an initial mesh of "Maximum size" elements which fills a region around and through a bounding region completely encapsulating the geometry.
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 adjusted (projected) and edges are split/swapped to conform to the geometry
- 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
  - Multiple volumes are supported for multi-region or multi-material problems
  - Elements outside the domain are marked into a reserve part name called ORFN, then deleted

Flood fill

- User defined volumes kept
- ORFN region is discarded
Tetra Process, Cont’d

• Smooth
  – Octree mesh is initially composed of regular right angle tetras
  – Smoother can be set to run to improve quality
  – Or run afterwards:
    
    Edit Mesh -> Smooth Mesh Globally
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 (1/10th 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 create a material point if none exists (named *CREATED_MATERIAL#*)
- Set Global, Part, Surface, Curve Size Parameters
  - Similar to *Shell Meshing* section

Geometry Repair tools quickly locate and fix these problems.

Missing inlet surface

Hole highlighted by yellow single edge curve
Using Points and Curves with Tetra Octree

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

Mesh captures detail

Coarse mesh ‘walks over’ detail in surface model

- Curves and points included
- Mesh size specified on curves and 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
**Tetra Octree - Options**

- Setup options:
  - **Global Mesh Setup > Volume Meshing parameters**
    - **Run as batch process**
      - Runs as a separate process. GUI will stay interactive.
    - **Fast Transition**
      - Allows for a faster transition in element size from finer to coarser
      - Reduced element count
    - **Edge Criterion**
      - Split elements at a factor greater than set value to better capture geometry
    - **Define Thin cuts**
      - Tool for resolving thin gaps, sharp angles
      - User selects pairs of opposing parts
      - Resolves elements jumping from one side to another
    - **Smooth**
      - Automatically smoothes after grid generation process
    - **Coarsen**
    - **Fix Non-manifold**
      - Automatically tries to fix elements that jump from surface to another surface
  - For a more detailed description go to Help > Help Topics > Help Manual > Mesh > Global Mesh Setup > Volume Meshing Parameters > Tetra/Mixed > Robust (Octree)
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 have it loaded)
  - Use Existing Mesh Parts
    - Select Parts that are already surface meshed
    - Merges nodes to preexisting surface mesh
    - Uses Make Consistent to match octree volume mesh to existing surface mesh
Curvature/Proximity Based Refinement

- **Curvature/Proximity Based Refinement**
  - *Octree* only
  - 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

- **Global Mesh Setup**
  - **Global Mesh Parameters**
  - **Global Element Scale Factor**
    - Scale factor: 1
    - *Display*
  - **Global Element Seed Size**
    - Max element: 0
    - *Display*
  - **Curvature/Proximity Based Refinement**
    - **Enabled**
    - Min size limit: 1
    - *Display*
    - Elements in gap: 1
    - Refinement: 10
    - Ignore Wall Thickness

- **Prescribed element size**: Surface/Curve Max. Element Size times Scale Factor
- **Prescribed size is adequate here**
- **Min Size Limit**: multiplied by Scale Factor = global minimum
- **Auto subdivision at tighter radius of 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
  - 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
**Elements in Gap**

- Number of cells desired in narrow gaps
- To avoid subdivision always to global minimum which would otherwise result in too many elements
  - Subdivision will stop once number of cells in gap is reached
- Will not override global minimum (*Min size limit*)

**Example**

- Only one element in gap
- Can’t go smaller than *Min size limit*
- Have to set smaller *Min size limit*
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.
**Edge Criterion**

- **Edge criterion**
  - For *tetra octree* only
  - A number 0 – 1
  - 0.2 (the default) means if more than 20% of an edge crosses a surface or curve, then split the edge
  - Has an effect similar to globally applying a *thin cut*
  - Smaller numbers will cause more splitting. The closest node will be projected to the surface
  - Use prudently. Too small a number results in strange globs of refined mesh

![Diagram showing edge criterion](image)
**Mesh Methods - Delaunay**

- **Type** - Tetra/Mixed
  - Method - *Quick (Delauney)*
- Start from a good quality, closed surface mesh
  - Can be quad and tri elements
  - From Shell Mesh
  - From Octree
  - From imported surface mesh

**Setup Options:**
- **Delaunay Scheme**
  - *Standard*: Delaunay scheme with a skewness-based refinement
  - *TGrid*: 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

Initially distributes nodes so as the centroid of any tetra is outside the circumsphere of any neighboring tetra.
Mesh Methods – Advancing Front

- Type - **Tetra/Mixed**
  - Method - **Smooth (Advancing Front)**
    - Same as **Quick (Delauney)** 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**

- **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
**Compute Mesh – Delaunay, Adv. Front, TGrid**

- **Run Options:**
  - Similar options as octree except cannot mesh to part geometry and part mesh (option: *Use existing mesh parts*).
  - *Create Prism Layers* available for both.
  - *Hexa-Core* not available for *Advancing Front, ANSYS TGrid*.
  - **Volume Part Name**
    - For newly created tetras.
    - Can choose *Inherited* to use material.
  - **Input**
    - *All Geometry*
      - Will run shell mesh first with no user input/editing.
      - Using parameters from *Model/Part/Surface/Curve Mesh Setup*.
      - Review *Shell Meshing* chapter.
      - If doubtful as to shell mesh quality, run *Shell Mesh* first, then use *Existing Mesh*.
    - *Existing Mesh*
      - Most common method. Surfaces already meshed.
    - *Part by Part*
      - Meshes each part separately. Nodes are not connected.
    - *From File*
      - Saves memory. Surface mesh does not need to be loaded.
Comparison

Octree

Delauney

Adv.front

ANSYS TGrid
Density Region

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
Periodicity

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

Tip: Placing material point close to mid-plane makes tetra octree obey periodicity easier
5. Prism Meshing
**Prism Meshing**

- **Inflation layers**
  - To better simulate boundary layer effects
  - Mesh orthogonal to surface with faces perpendicular to boundary layer flow direction
- **Procedure**
  - Set *Global Prism Parameters*
  - Select *Parts* to grow layers from
    - Typically wall boundaries and holes
  - Set Local Parameters for each part
    - Local overrides global
    - Zero or blank entries will defer to global settings
  - Run mesher
    - From existing mesh
      - Extrude into tetra/hexa mesh
      - Extrude from surface tri mesh, then fill volumes
    - Run automatically during *Volume Mesh* creation
Prism - Global Parameters

- **Global Prism Parameters**
  - **Growth law**
    - *exponential*: \( \text{height} = h(r)^{(n-1)} \)  [\( n \) is layer #]
    - *linear*: \( \text{height} = h(1+(n-1)(r-1)) \)
    - *wb-exponential*: \( \text{height} = h \times \exp((r-1)(n-1)) \)
  - **Initial height** of first layer – \( h \) in formulae above
    - Auto calculated if not specified
      - Based on factor of edge length of base triangle/quad
      - Height determined so that top layer volume is slightly less than that of tetra/hex just above it
  - **Number of layers** \( n \)
  - **Height ratio** \( r \)
  - **Total height** - of all layers
  - Usually specify 3 of the above 4 parameters
    - *Compute params* will calculate the remaining parameter (*total height* usually left blank)
  - Or specify only **Height ratio** and **Number of layers** for auto calculation of initial height
  - Individual surface/curve height/ratio/layers will override these global defaults if set
  - Other global parameters explained later
Growth Law Comparison

- The growth rate of \textit{Wb-exponential} is greater than \textit{exponential}
- The growth rate of \textit{exponential} is greater than \textit{linear}
**Smooth Tetra/Prism Transition**

- Leave initial height as “0”
  - This causes the initial height to float in order to reduce the volume change between the last prism and adjacent tetra.
**Setting Prism Parameters on Parts**

- Prism extrusion areas defined by the parts
  - *Mesh > Part Mesh Setup*
  - Toggle on *Prism* for parts where inflation layers are desired
    - Surface mesh (tri/quad) gets extruded into prisms
    - Set *Height, Height Ratio, Num Layers*
      - Will use global defaults if not set or zero

Applying these settings causes these parameters to be applied to each individual surface within each part.

If *Apply inflation parameters to curves* is toggled on, they will also be set on each curve within each part.
Setting Prism Parameters on Volume Parts

- Normally toggle prism on only for parts that contain surfaces (becomes surface mesh)
- Can also toggle on prism for parts that contain material points (becomes volume mesh)
  - For interior surface mesh, this defines the allowable volumes for extrusion
  - Selecting no volume parts has the same result as selecting all volume parts
Setting Prism Parameters on Surfaces

- **Mesh > Surface Mesh Setup**
- You can specify different local *height* and *ratio* on any selected surface without moving the surface to a new part
- Usually set *height* and/or *ratio* smaller on specific surfaces to avoid collision

Collisions occurred when the height was 0.4 on all surfaces

No collisions after
**Setting Prism Parameters on Curve**

- **Mesh > Curve Mesh Setup**
  - You can get Prism to transition linearly across a surface by not setting a height (height = 0) on the surface, but instead set a different height on each curve on the opposite sides of the prism surface.
  - *Height ratio* and *Num. of layers* have no affect on prism for curve settings.

![Diagram showing prism setup parameters](image)

- **Height = 0.003**
- **Height = 0.01**
- **Height = 0 on surface**
- **Height = 0.003**

---

90
Run Prism

- Can run separately
  - Mesh > Compute Mesh > Prism Mesh
  - The Select Parts for Prism Layer button pops up the same menu as the Part Mesh Setup, except non-prism related columns aren’t displayed
  - Input
    - Existing Mesh
    - From File (saves memory by not loading mesh)

- Or run automatically linked into volume mesh
  - Toggle on Create Prism Layers when tetra meshing
  - Not advisable if this is the first mesh for a particular geometry
  - Must be confident about setup parameters and sizing
  - Running prism separately allows you to smooth and error-check the tri or tetra mesh first.
Input as Surface or Volume Mesh

- Input can be a surface mesh or volume mesh
  - Surface mesh
    - Must be a closed boundary mesh
    - Must specify a volume part
    - Use tetra fill methods after:
      - Delaunay
      - Advancing Front
      - Ansys TGrid
  - Volume mesh
    - Moves and reconnects tetras
Prism – Quality Control Options

- **Fix marching direction**
  - Maintains normal from surface
  - Can cause intersections with other mesh

- **Min prism quality**
  - Either re-smooth directionally or cap/replace with pyramids if quality not met (minimum allowed = 1x10^-6)

- **Ortho weight**
  - Weighting factor for node movement from 0 - improving triangle quality, to 1 - improving prism orthogonality

- **Fillet ratio**
- **Max prism angle**
- **Max height over base**
- **Prism height limit factor**
- **Ratio multiplier (m)**
  - For varying exponential growth: \[ \text{height} = h(r)^{(n-1)}(m)^{(n-1)} \]
Prism Options – Fillet Ratio

- Blends prism grid lines around sharp corners
  - 0 = no fillet
  - 1 = fillet ratio equals last prism height
- Improves angles further away from the corner
- Orient prisms more in direction of flow
- If meshing tight spaces with tight curves (less than 60°), may not have space for a fillet ratio

\[ \text{Fillet Ratio} = \frac{r}{h} \]

Fillet Ratio = 0.0  
Fillet Ratio = 0.5  
Fillet Ratio = 1.0
Prism options – Max Prism Angle

- Controls prism layer growth around bends or adhering to adjacent surfaces
- If the *Max* (internal) *Prism Angle* is not met, the prism layers will end and be capped off with pyramids in those locations
- Usually set in the 120° to 179° range
- Experience pays off here. If extruding from one part and not its neighbor, and the angle between the two surfaces is greater than the *Max Prism Angle*, the prisms will detach and be capped off with pyramids. This prevents bending the prisms that might create lower-quality internal angles. However, the pyramids are usually of lower quality, too.
- It’s usually better to run prism along adjacent surfaces until it can meet at a smaller angle, leaving quad faces. Pyramids will be avoided.

![Original mesh](image1.png)

160°

Original mesh

![Max prism angle = 180°](image2.png)

Max prism angle = 180°

![Max prism angle = 140°](image3.png)

Pyramids

Max prism angle = 140°
Prism Options – Max Prism Angle - Continued

- A high (up to 180°) \textit{Max Prism Angle} keeps the prism layers connected around tight bends.
  - Set this at 180 to prevent pyramids where possible.
Prism Options – Max Height Over Base

-restricts prism aspect ratio
-Prism layers stop growing in regions where prism aspect ratio would exceed specified value
  - Number of prism layers would not be preserved locally
-Mesh is made conformal with pyramids at prism boundaries
-Acceptable values vary widely (typically 0.5 – 8)

Max Height Over Base not set

Max Height Over Base = 1.0

Largest height over smallest base length

Pyramids

Height (h)

Base (b)

h/b
Prism Options – Prism Height Limit Factor

- Restricts prism aspect ratio
  - Prism height will not expand once this factor is met
- Uses the same height over base factor as the previous metric except prism layers are not capped off with pyramids
- Preserves the specified number of prism layers
- Will fail if sizes of adjacent elements differ by more than a factor of 2
- Acceptable values vary widely (typically 0.5 – 8)

\[
\frac{h}{b} \quad \text{Largest height over smallest base length}
\]
Prism Options - Part Control

- **New volume part**
  - Can specify new Part for prism elements
  - Must specify if extruding from surface-only mesh
  - If extruding into volume mesh, prism will inherit tetra volume Part if not specified

- **Side part**
  - For quad faces on side boundary

- **Top part**
  - For tri faces capping off top of last prism layer

- **Extrude into orphan region**
  - Extrude prisms away from existing volume, not into it
  - Must specify *new volume, side and top* Part, or the y’l’ll be in ORFN

Leaving these parts blank will inherit the names from the current mesh.
Prism Options - Smoothing

- Prepares tri/tetra for best prism quality
  - Set surface/volume steps to 0 if only extruding one layer or if tri/tetra mesh is already smoothed
    - Otherwise defaults adequate
    - Value depends on model/user experience
  - Set surface smoothing steps to zero for a tri/tetra mesh that is already smoothed
  - Triangle quality type
    - Laplace typically best for eventual prism quality
    - Other types may be better when marching directions condense at inside corners
  - Max directional smoothing steps
    - Redefines extrusion direction based on initial prism quality
    - Internally calculated for each layer

- Other Advanced Prism Meshing Parameters
  - Detailed in Help menu (usually left default)
**Prism Parameters File**

- **Read a Prism Parameters File**
  - To set all prism values from a prism settings file (*.prism_params)
  - Written to the working directory every time prism is run

```
surface_file temp_prism0.tin
input_file temp_prism0.uns
output_file prism.uns
law exponential
layers 6
height 0.30000001
ratio 1.2
total_height 2.9789801
n_processors 3
first_layer_smoothing_steps 1
fillet 0.3
max_prism_angle 180
triangle_quality_inscribed_ratio
family VALVE height 0.3 ratio 1.2
family ELBOW_SIDE1 height 0.3 ratio 1.2
family ELBOW_SIDE2 height 0.3 ratio 1.2
family ELBOW_SIDE2_PILFER height 0.3 ratio 1.2
family FLUID
```
Smoothing a Tetra/Prism Mesh

After generating prisms:

*Edit Mesh > Smooth Mesh Globally*

- Prisms are smoothed during prism generation
- If input mesh was a tetra mesh, the tetras adjacent to the last prism layer will be messed up
- First smooth only the tetras and tris
  - Set PENTA_6 to Freeze
  - Don’t want to modify the prism layers at this point
- Once tetra and tri elements are as smooth as possible, smooth all elements
  - Set PENTA_6 to Smooth
  - Decrease the Up to quality value so as not to distort prism elements too much

The prisms get compromised a bit when everything is on smooth.

1st step

2nd step
Splitting Prism Layers

- If many prism layers are desired, it is faster, but less robust – to create “fat” layers and then split them with mesh editing

  - *Edit Mesh > Split Mesh > Split Prisms*
    - *Fix ratio:* The layer is split such that its resulting layers employ the given growth ratio (height is free variable)
    - *Fix initial height:* The layer is split such that its first sub-layer is of the given height (ratio is free variable)

- Specify the number of layers to result from each existing layer

- Can split specified or all existing layers
Redistributing Prism Layers

Redistribute prism layers after splitting

- **Edit Mesh > Move Nodes > Redistribute Prism Edge**
  - *Fix ratio:* The initial height and subsequent layer heights will be adjusted to achieve this growth ratio
  - *Fix initial height:* The growth ratio is the variable that will be adjusted to achieve this initial height
  - The total prism thickness remains fixed and layers are adjusted within this thickness
6. Mesh Preparation Before Output to Solver
Mesh Preparation Before Output to Solver

What will you learn from this presentation:

- Checking and Improving the quality of the mesh
- Manipulating the elements
- Subsets

Usage of Edit Mesh tools

- To diagnose and fix any problems and improve mesh quality
- To convert element types
- Refine and/or coarsen mesh
- Manual and automatic tools
- For imported as well as internally created mesh
Mesh Checks

• To diagnose mesh connectivity problems
  – **Errors** – most likely to cause problems in:
    • Solver translation
    • Solver input
    • Solution convergence/run
  – **Possible Problems** – “Unclean surface mesh”
    • Unwanted elements
    • Unwanted holes/gaps
    • May result in incorrect solution
  – Can check any combination of errors/possible problems at any one time
    • Individually select
    • Clicking on **Error** or **Possible Problems** headings will select all options in column – selecting again will de-select all
    • **Set Defaults** will select the most common checks for the current mesh type (2D or 3D)
  – **Check Mode**
    • **Create Subsets** – creates a subset of elements for each problem found (will run through all selected checks)
    • **Check/Fix Each** – offers automatic fixing of indicated problem (needs user decision after each problem found)
Mesh Checks - Mesh Errors

- **Duplicate Elements**
  - Elements that share all nodes with other elements of the same type

- **Uncovered Faces**
  - Volumetric element faces that are neither attached to the face of another volumetric element nor to a surface element (boundary face)

- **Missing Internal Faces**
  - Volumetric elements that are adjacent to another of a different part with no surface element between them

- **Periodic Problems**
  - Inconsistency in the pattern of nodes/faces between periodic sides
  - Special check for rotating (sector) or translational periodic grids
  - Select pairs of parts to check

- **Volume Orientation**
  - Left handed elements due to incorrect connectivity (node numbering of cell)

- **Surface Orientations**
  - **Surface** elements whose attached volume elements share part of the same space

- **Hanging Elements**
  - Line (bar) elements with a free node (node not shared by any other element)

- **Penetrating Elements**
  - Surface element(s) that intersect or penetrate through other surface elements

- **Disconnected Bar Elements**
  - Bar elements where both nodes are unattached to any other elements
Mesh Checks - Possible Problems

- **Multiple Edges**
  - Surface elements with an edge that shares three or more elements
  - Can include legitimate T-junctions
- **Triangle Boxes**
  - Groups of 4 triangles that form a tetrahedron with no actual volume element inside
- **2-Single Edges**
  - Surface element with 2 free edges (not shared by another surface element)
- **Single-Multiple Edges**
  - Surface element with both free and multiple edges
- **Stand-Alone Surface Mesh**
  - Surface elements that don't share a face with a volumetric element
- **Single Edges**
  - Surface elements with a free edge
  - Can include legitimate hanging baffles
  - 2D-only mesh boundaries are single
- **Delaunay Violation**
  - Tri elements with nodes that are within the circumsphere of adjacent tri elements
  - legacy quality criteria
- **Overlapping Elements**
  - Continuous set of surface elements that occupy the same surface area (surface mesh that folds on to itself within a small angle)
- **Non-manifold vertices**
  - Vertices whose adjacent surface elements’ outer edges don't form a closed loop
  - Typically found in tent-like structures where surface elements jump from one surface to another across a narrow gap or sharp angle
- **Unconnected Vertices**
  - Vertices that are not connected to any elements
  - Can always be deleted
Mesh Checks – Check Mode

• If Create Subsets was selected
  – Will go through all checked criteria without interruption
  – Elements that have a particular error/problem are put into a subset with the same diagnostic type name
  – Subsets activated in Model Tree
    • Turn off all parts or shells to view subsets
• If Check/Fix Each was selected
  – Will be prompted with options one criteria at a time
    • Fix: Automatically fix the error/problem
      – Recommended only for Duplicate Elements, Uncovered Faces, Missing Internal Faces, Volume Orientations, Unconnected Vertices
    • Create Subset
    • Ignore
      – For example, multiple edges may be legitimate t-junctions; single edges may be legitimate free edges
Mesh Quality Display

- A diagnostic check of individual element quality
  - **Mesh types to check** – Allows you to select the mesh types to check
    - 1D (Line elements)
    - 2D (Tri and/or Quad)
    - 3D (Tetra, Penta, Hexa and/or Pyramid)
  - **Elements to check** – By part and subset
    - All
    - Active parts
    - Visible subsets
    - Visible subsets and active parts
  - **Refresh Histogram** – Refreshes the histogram displayed
  - **Quality type** – Specifies the Quality criterion for display
    - 48 quality criteria available
    - Some checks don’t apply to all element types

Selecting a histogram bar will display the elements in that range
Mesh Smoothing

- Automatically improve element quality
  - All element types
  - Necessary to have geometry loaded
  - Nodes are moved to improve the element quality
    - Automatic node movement constrained by node projection type to geometry type – e.g. curve nodes will be constrained to move only on curves
    - Histogram is automatically displayed/updated after smoothing

- User chooses:
  - **Criterion**
  - **Up to value**
  - **Smooth Mesh Type**
    - **Smooth**: Element types actively smoothed; quality of type appears as part of histogram.
    - **Freeze**: Nodes are held in place during the smoothing process. These elements not shown in histogram
    - **Float**: Nodes can be moved along with adjacent smoothed elements, but quality ignored; not shown in histogram
    - Example: Freeze Prisms and Pyramids while smoothing Tetra. Float surface elements
Mesh Smoothing

- **Advanced Options**
  - **Smooth Parts/Subsets**
    - Smooth all parts, visible parts (activated in model tree), or visible subsets
    - Quick, local smoothing
  - **Laplace smoothing**
    - Gives more uniform mesh size relative to neighboring elements and equal angles
    - Recommended for TRI only – smooth Tetra after with Laplace turned off and tri’s froze
    - Recommended prior to prism generation
  - **Not just worst 1%**
    - Factors in all elements instead of only the worst 1% of those beneath quality value and their neighbors
    - Can improve quality but takes much longer
  - **Violate geometry**
    - Unconstrain nodes slightly from geometry within user defined tolerance – absolute or relative to minimum edge length of mesh elements
7. Output to Solver
Selecting solver

- Use the red toolbox in the Output tab to select the solver format
- This same menu is accessed with Settings > Solver
- 123 solver formats available
- All output formats can be set with the Output Solver pulldown
- Help on each solver format can be found on the Ansys website:
  http://www.ansys.com/Products/Other-Products/ANSYS+ICEM+CFD/Output+Interfaces/Output+Interfaces+TOC

- Also found in Help > Output Interfaces
Mesh Formats

- There are 2 types of mesh formats that solvers read
  - **Unstructured**
    - Most solvers use unstructured formats
    - Some examples are Fluent, Ansys CFX, CFD++, Abaqus, Ansys
    - Nodes have an ID and location
    - The ICEM CFD unstructured mesh has a *.uns extension and can be made from any mesher in ICEM CFD
  - **Multiblock structured**
    - Some older CFD solvers require this format
    - Examples are Plot3D, CFX-TASCFLOW, KIVA-3V
      - CGNS supports both unstructured and multiblock structured
    - Nodes have IJK index designation and location
    - The ICEM CFD structured mesh has many files:
      - Project.1, project.2, project.3, etc… for each mesh domain
      - A topology file, topo_mulcad_out.top, that describes how each domain is connected to the other domain.
      - A dummy file, Project.multiblock, that enables selection of all the multiblock mesh files of that project name by selecting this one file
    - Only an ICEM CFD hexa blocking is capable of being written out in multiblock structured format
Boundary Conditions

- Both CFD and FEA solvers allow boundary conditions to be set.
  - **Output > Boundary Conditions**
    - BC’s are set on mesh elements grouped by part names.
    - Tree structure organizes part names by dimension (2D, 3D, etc) of geometry and mesh in the part.
    - Any part containing mixed dimensions (e.g., curves and tri’s) will be grouped into **mixed/unknown**.
    - Use **File > Attributes > Save Attributes As**… to save the BC file (*fbc* and *atr* extensions).

BC’s shown for solver Fluent_V6

![Selection](image1)

- Part highlights white when selected.
- Use **Create new** to select a BC type.
- Example: **velocity-inlet**.

![Family boundary conditions](image2)
Some structural solvers have global parameters

- **Output > Edit Parameters**

- If the solver type requires global parameters, you must bring up this menu before writing out and press **Accept**, even if not changing any parameters

- Then use **File > Parameters > Save Parameters As…** to save the file (**.par** extension)
• **Output > Write Input** to write the solver file
  – It will ask you to save a boundary condition file (*fbc) and the project
  – It’s always safe to save these when asked, but not necessary if you know you saved them and didn’t make any changes since then
  – It will then ask for the ICEM CFD mesh file to translate
    • *.uns file if the solver requires an unstructured mesh
    • *.multiblock file if the solver requires a structured mesh
  – A final menu will pop up that has specific options for the solver translation being used
    • The example on the left is for **Fluent_V6**
  – All translators will require a boundary condition file even if no BC’s are set
    • A empty BC file will contain just the part names
  – There will always be a name for the output file
Special Structural Solvers

- 5 Special structural solvers
  - There are 5 structural solvers that have extended boundary condition setup – called loads and constraints – which are set using the **Common Structural Solver** dropdown
  - These are *Nastran, Ansys, LS-Dyna, Abaqus, and Autodyne*.
  - These are set using the 4 extra tabs; **Properties, Constraints, Loads, and Solve Options**.

- Constraints tab
  - All functions are only available for *LS-Dyna* and *Autodyne*.
  - "Define contact" is allowable with *Ansys* and *Abaqus*.
  - *Nastran* only allows the first two functions.
Special Structural Solvers - Procedure

• Define which elements to write out
  – First define a **material** in the Properties tab
  – Then define an element property on the element types and parts you want to write out
    • 3D property for volume elements
    • 2D property on shells
    • 1D property on line elements
    • 0D property on node elements
• Define any **Loads** and **Constraints** in the other tabs (optional)
• Use the **Solve Options** tab to write the mesh to the solver
  – **Solve Options > Write/View Input File**
• Can submit solver run if the corresponding environment variable is set to find the solver executable
  – For Ansys, it is ANSYS_EXEC_PATH
If using the **Properties**, **Loads**, and **Constraints** tabs for one of the 5 special structural solvers, then use the **Solve Options** tab to write the file, not the **Output** tab.

- The *attribute* file (`.atr`) is the BC file used with these special structural solvers.
- The *attribute* file is replaced with the boundary condition file (`.fbc`) when using the **Output** tab.
- Any elements set to **All** or **None** will override any properties set in the **Properties** tab if not set to **Defined**.
- To communicate with the **Output** tab, click on **Advanced**, and then click **Create Attribute & Parameter Files**.
  - You can then **Edit Parameters** and **Edit Attributes** here or in the Output tab.
  - The attributes will then be the same as the boundary conditions.
- View input file here or in your own text editor.
8. ICEM CFD Hexa
What is Blocking?

- A hexa mesh is created by first making a “blocking”
  - A blocking breaks down a geometry into large brick-shapes and structures the direction of grid lines by the arrangement of the blocks
  - Each “block” is easily meshed with a pure Cartesian mesh
    - Some blocks can be defined as “swept” and be unstructured along one face
  - Block entities (faces, edges, and vertices) are projected onto the geometry
  - The blocking is saved to an independent file, and can be loaded onto a different geometry

![Geometry](image1)
![Blocking](image2)
![Mesh without projection](image3)
![Mesh with projection](image4)
Approach – Top Down and/or Bottom Up

Block structure is created independent of geometry
- “Top down” topology creation
  • The user as sculptor instead of brick layer
  • One-step creation of advanced topologies (O-grid)

- “Bottom up” topology creation
  • Blocking is built up like “laying bricks”
    – Create blocks
    – Extrude face
    – Copy topologies

- Combinations of top-down and bottom-up methods can be used
Geometry Requirements for Hexa

Use same geometry (tetin) as used with Tetra

- Does not necessarily need to be a completely enclosed volume

- Points and curves are not required but are very useful
  - Use Build Diagnostic Topology to quickly build all curves and points

- Block structure is projected to geometry
  - Surfaces – automatically with manual override
  - Curves and points – manually projected

Associating Face to Surface to a dummy point family or Interpolation can effectively mesh where geometry doesn’t exist
Geometry/Blocking Nomenclature

- **Geometry**
  - Point
  - Curve
  - Surface
  - Volume

- **Blocking**
  - Vertex
  - Edge
  - Face
  - Block

Note: “curve” refers to lines, arcs, and splines (1D geometry)
**Blocking process**

- Structure blocking to capture the shape of the geometry
  - Top down
    - split and discard unused blocks
  - Bottom up
    - create block by extrusion, creating, copying
- Associate blocking to geometry
  - Usually just edges to curves
- Move vertices onto geometry
  - Manual and automatic methods
- Assign mesh sizes
  - Quickly by setting sizes on surfaces and/or curves
  - Fine tune by setting edge distributions
- View mesh and check/improve quality
- Write out mesh
Initialize Blocking - 3D or 2D

- New Blocking
  - **3D Bounding Box**
    - One 3D volume block encompassing selected entities
  
- **2D Planar**
  - One 2D block on XY plane around entire 2D geometry
  - First rotate geometry to XY plane
  - No surfaces required

- **2D Surface Blocking**
  - Discussed next slide
2D Surface Blocking

- Each surface becomes one 2D block
- Free blocks – fully unstructured
  - Most robust
- Mapped – structured
  - Aligned mesh along 4 block boundary edges

- Must *Build Diagnostic Topology* first
  - It needs the connectivity information
  - Can always convert blocks between free and mapped afterward: *Edit Block > Convert Block Type*
Structuring Blocking to Fit Geometry

Top down approach

Start with one block which encloses the entire geometry

Delete unused blocks

Split the block to capture the underlying shape

Note: Deleted blocks are put into the part VORFN by default, so they can be re-used later if wanted
**Associate Blocking to Geometry**

- Associate blocking to geometry
  - Usually just Edges to Curves
  - In the final mesh, edges will take the shape (be projected to) these curves
  - Right click on *Edges > Show Association* in the model tree to display the association arrows
Move Vertices onto geometry

• Move vertices to better represent the shape
  – All visible or selected vertices can be projected to the geometry at once
  – Can be moved individually along the geometry
  – Single or multiple at a time
  – Along fixed plane or line/vector
Moving Vertices of Different Association

- Color indicates type of association and how a vertex will move (edges also follow these colors, except red)
  - **Red**
    - Constrained to a point
    - Can’t be moved unless association is changed
  - **Green**
    - Constrained to a curve
    - Vertex slides along that particular curve
  - **White (black on light background)**
    - Constrained to surfaces
    - Vertex will slide along any ACTIVE surface (surface parts which are turned on in the model tree)
    - If not on a surface, it will jump to the **NEAREST ACTIVE** surface when moved
  - **Blue**
    - Free (usually internal) vertex
    - Select NEAR (not on) the vertex on the edge to move along edge direction
Assign Mesh Size

• Assign mesh sizes
  – Hexa sizes can be assigned on surfaces and curves for a quick mesh
  – Need to *Update sizes* to apply to surface/curve sizes to edges
  – Or set edge-by-edge for fine tuning
    • Automatic *Copy to parallel edges*
**Edge Parameters**

Spacing 1 – distance between first two nodes on side 1
Ratio 1 – growth ratio from side 1 toward center
Spacing 2 – distance between first two nodes on side 2
Ratio 2 – growth ratio from side 2 toward center
Max Space – Maximum element length along edge
View Pre-Mesh

- Pre-Mesh
  - Create mesh at any stage of the process
  - Mesh with different projection methods
  - Use *Project faces* (default) to fully represent the geometry
  - View only certain surface mesh by turning on only that *Part* in the model tree
  - Use *Scan planes* to view internal mesh (covered later)
Checking Quality

- Using the Quality Histogram
  - Determinant
    - Measurement of element deformation (squareness)
    - Most solvers accept > 0.1
    - Shoot for > 0.2
  - Angle
    - Element minimum internal angles
    - Shoot for >18 degrees
  - Aspect ratio
  - Volume
  - Warpage
    - Shoot for < 45 degrees
  - Many more metrics

You can display elements in a given range by selecting the histogram bar.
Write Mesh

- Convert pre-mesh to a permanent mesh
  - Two formats depending on what your solver takes
    - Unstructured: Cells defined by node numbers (connectivity)
    - Structured: Multiblock – cells defined by i, j, k index
  - Blocking changes will no longer affect this mesh

Using the File > Blocking > Save… menu only writes the mesh to disk

Pre-Mesh > Convert to…” a mesh from the model tree saves and immediately loads the mesh
**Ogrid Definition**

- An O-grid is a series of blocks created in one step which arranges grid lines into an “O” shape or a wrapping nature
  - 3 basic types created through the same operation all referred to as “O-grids”
    - O-grid
    - C-grid (half O-grid)
    - L-grid (quarter O-grid)
  - Reduce skew where a block corner must lie on a continuous curve/surface
    - Cylinders
    - Complex geometries
  - Improves efficiency of node clustering near walls for CFD applications
Creating Ogrid

• Select blocks for O-grid
  – Can select by visible, all, part, around face, around edge, around vertex, 2 corner method, or individual selection

Select specific blocks or around face, edge, or vertex

Note: Internal block has all internal (blue) edges and vertices
Ogrid – Adding Edges/Faces

- Adding faces during O-grid creation
  - O-grid “passes through” the selected block faces
  - In general, add faces on the “flat parts”
  - Adding a face actually adds blocks on both sides of the face

- Examples of uses
  - Pipe ends
  - Symmetry planes
  - Complex geometries
**Ogrid – Adding Multiple Edges/Faces**

- Any number of faces can be added around a selected block
  - If all the faces are added around a block, the result is no change since the O-grid passes through all the faces

Quarter O-grid
(L-grid)

Quarter O-grids can be used to block triangular shapes

Seen as a C-grid in one direction and an L-grid in another direction
• Select *Around block(s)* to create the O-grid around the selected blocks
  – Useful for creating wrap-around grid around a solid object
  – Examples
    • Flow over a cylinder
    • Boundary layer resolution around an airplane or car body
O-grids can be re-sized after or during creation

- By default the O-grid size is set to minimize block distortion
- You are actually scaling all parallel O-grid (radial) edges to the selected edge
- The selected edge is given a factor of 1
- Numbers < 1 will shrink the edge and thus create a larger inner block
Why Create an Ogrid?

This mesh can be improved by using an O-grid
- An example of bad mesh in the block corners

Right mouse click on the histogram to access options like show, replot, or done
Index Scheme

- All blocks and vertices are defined with a global index scheme
  - Initial block has i,j,k indices aligned with Cartesian x,y,z, global coordinates
  - Subsequent blocks created by the split operation will maintain that orientation
  - O-grids will not conform to this orientation, so each O-grid creates a new index direction (O3, O4, etc…) (i,j,k correspond to dimension 0,1,2)
  - Vertex Indices can be displayed by right clicking on **Vertices > Indices** in the model tree
Using Index Control

- Blocks can be turned off and on based on indices
  - Use the index control to turn blocks off and on
  - Many operations can be applied to only the visible blocks
    - Split blocks
    - Rescale O-grid

Select corners is often faster than toggling the index arrows

Resets everything to fully visible
**VORFN Part**

- A region of blocks called VORFN surrounds the blocking
  - The reserved part name *VORFN* is created in the model tree when blocking is initialized
  - It is used to maintain the global index scheme
  - Indices begin at 0 (in *VORFN* region)
  - Deleting blocks normally (non-permanently) just changes the part to *VORFN*
    - Blocks can be moved back out of *VORFN* to another part
    - Deleting blocks *permanently* gets rid of the block and rebuilds *VORFN* as an O-grid

Selected blocks are actually removed and *VORFN* is rebuilt (as an O-grid)

Turn on to see *VORFN*
Removing Splits and O-grids

• Splits and O-grids can be deleted by using *Merge vertices > propagate*
  – Select two vertices at the ends of an edge running in the direction you want to delete
  – Middle click, then *Confirm*
  – The second vertex merges to the first vertex when *Merge to average* is off, and the merge will be propagated
  – If deleting an O-grid, only select the vertices of one radial edge
**Extrude Face**

- Select block face (select face or use two corner method)
  - Interactive
    - Drag with the pointer
  - Extrude a Fixed distance
    - Enter distance
  - Extrude along curve
    - Select curve
Set Location

Set X, Y, and/or Z of selected vertices

- Select **Ref. Vertex**, then select vertices to move or use the index control and select all visible vertices (“v”)
- Select directions to move (Modify X, Y, Z)
  - Can use a local coordinate system (cylindrical, Cartesian)
  - If cylindrical coor. System, (x, y, z) becomes (r, θ, z)
- Enter values or use the values from the **Ref. vertex** or screen location
- Method can be **Set Position** or **Increment**
- **Apply**
Align Vertices

Align multiple vertices in one plane with vertices in another plane (or split dimension)

- Select *Along edge direction*, then select the edge that connects the two planes, or runs approximately normal to the split plane that you want to move inside
- Select *Reference vertex*, then select any vertex in the plane you want to align to. These vertices will remain fixed
- Select the plane to allow vertex movement within (XY, YZ, XZ, or User Defined). The *User Defined* plane must be specified with a normal vector, such as (1 1 0)
- **Apply**

All visible vertices not sharing the reference index will be moved
A 2D blocking can be extruded into a 3D blocking by three different methods:

- **Multizone Fill**
  - Auto creation of 3D blocking from enclosed 2D surface blocks
- **Translate**
- **Rotate**

Number of nodes in circumferential direction

Extrudes points into curves and curves into surfaces where 3D geometry doesn’t exist

Surface mesh and scan planes
Create Block – Wedge

Degenerate
- Select 6 vertices or locations
- Order is important (see picture)
- Grid lines converge at vertices 1 and 4
- Results in penta6 (prism) elements along one edge

Quarter-O-Grid
- Results in all Hexa “wedge” (Y configuration)
Create Block – Swept Block

- **Swept** (Unstructured)
  - Select 6 vertices or locations
  - Different order than degenerate, quarter o-grid
  - Results in unstructured mesh
  - Some tris/prisms
- Can also **Convert Block Type**
  - To **Structured**
  - To **Unstructured** (2D)
  - To **Swept** (3D)
  - Hex blocks as well as degenerate wedges
Collapse Blocks

Collapsing blocks

- Select edge to define collapse direction
- Select blocks to collapse
- Results in a degenerate block (converges to penta6 (prism) elements)
- Degenerate block is often deleted

Example: meshing around knife-edge wings
**Split Vertex**

Separates or undoes merged vertices, including vertices merged due to a collapsed block

- Select any numbers of vertices and middle click
- **Apply**
- If you have deleted any blocks permanently after vertices were merged, this operation may not work to undo the merge. This is because a new index scheme is configured when blocks are permanently deleted.
Periodicity in Geometry

First must be specified in tetin file

- **Global Mesh Size** -> **Set up Periodicity**
  - **Translational**
    - Enter vector which specifies magnitude and direction
  - **Rotational**
    - Enter **Axis** vector – only specifies direction
    - Enter **Base** point that axis goes through
    - Enter **Angle** in degrees

![Diagram showing periodicity setup](image)
Periodicity in Blocking

Vertices then made periodic in blocking

- Select **Edit Block -> Periodic Vertices -> Create**
- Select pairs of vertices at a time
- The second vertex of each pair will move to the periodic position of the first vertex
- When you move one vertex, its pair will move with it
- Subsequent splits will also be periodic
- Visually verify with **RMB on Vertices -> Periodic, and Faces -> Periodic Faces** in model tree
- A face becomes periodic only if its 4 corner vertices are periodic

Subsequent splits are also periodic
2 Ways of Blocking the Same Geometry

- Creating a fork by Merge vertices

  - Delete block
  - Merge 2 vertex pairs
  - Merge 2 more vertex pairs
2 Ways of Blocking the Same Geometry

• Creating a fork by Extrude faces
Multiple Ways of Blocking the Same Geometry

- Creating a fork with Top-down methods

- Split

- Two quarter O-grids

- Delete blocks

- Move vertices

- One quarter O-grid

- One quarter O-grid
What is common in these?
  - Their block topology
• All these parts have the same basic topology
  – Blocking strategy for all pipes are similar.
  – Single block with O-grid
  – The only difference is the number of splits added to help control the blocking
  – Create the one block, then split, and add O-grid last

This helical blocking is quickly created with extrude along curve.
**Extend Split**

- Think of a split as a plane (even though it does not have to be planar)
- Select an edge at the outside of this “plane” and it will extend in all directions
  - Split only goes through the displayed blocks
  - Use the index control to limit the displayed blocks

Select *Project vertices* to have it automatically project new visible vertices to the nearest place on their associated geometric entities.
Shaping Edges – Split Edge

- Edges are by default linear before projection
- Edges can be shaped using split edge
  - Spline
  - Linear
  - Control point
- Use Move vertex to move splits after splitting

- Split edge will also override the automatic interpolation of edges during mesh computation
Shaping Edges – Split Edge Example

• Using split edge to shape block faces to make a hole where it doesn’t exist in the geometry
**Shaping Edges – Link Edge**

- Use one edge to control the shape of another
  - Select source edge then target edge(s)
  - Enter factor (higher number = greater curvature)

![Edit Edge(s) window](image)

- Source edge
- Target edge

- Factor = 1.0
- Factor = 0.9
- Factor = 1.3
- Factor = 0.5
**Bottom-Up Meshing Methods**

- Top-down is, generally, more robust

- Bottom-up methods improve flexibility
  - *Transform Blocks*
    - Translate
    - Rotate
    - Mirror
    - Scale
  - *2D to 3D* extrusion
    - Translate
    - Rotate
  - Block independently and merge
  - *Extrude face*
  - *Create block*
Transforming Blocks

- Simply transform selected blocks or make a copy and merge the transformed copy with the previous blocking
  - Select blocks to transform
  - Select **Method**
    - **Translate**
    - **Rotate**
    - **Mirror**
    - **Scale**
  - Enter parameters necessary for method
Create Block – Hexa (Vertex Locations)

- Create block types
  - Hexa
    - 8 vertices or locations (selection order is important)
      (Choose a vector direction and do the same order on opposite faces)
  - 2 faces
    - Quarter O-grid (Y-grid)
    - Degenerate

Select 8 vertices
Select 2 faces
Create Block – Hexa (Geometry Locations)

- What if I don’t have 8 vertices to select?
  - Select the vertices that you do have
  - Press middle mouse button
  - Select the rest of the locations on the screen
  - The same order must be maintained as before
Merge Vertices

- One by one

- Multiple within tolerance
  - When merging individually, select two vertices at a time
  - With *Merge to average* off, the second vertex will merge to the first
  - With *Merge to average* on, both vertices merge to the middle of the two
  - Use to join separate topologies together
  - Use to make degenerate blocks
Delete Blocks – Permanently

- Delete blocks will, by default, move blocks to the part VORFN, which is more stable than permanently deleting blocks (doesn’t recompute indices)

- However, there are some situations where deleting permanently is useful
  - Deleting all of VORFN can serve as a repair tool in complex topologies (indices get reconfigured)
  - Deleting individual blocks will free up node connectivity across VORFN blocks

Equal number of nodes across hole

Delete block permanently

Number of nodes can be unequal
Three Basic uses of O-Grids

• Three basic uses of O-grids
  – Capture the shape of the geometry
    • Usually done early in the blocking process
  – Improve element quality in block corners where surfaces do not make a corner (smooth transition at block corner)
  – Improve efficiency of node clustering near walls
    • Boundary layer resolution
  – These last two are usually accomplished with the same O-grid, and done late in the blocking process

Example of using an O-grid to capture the basic shape as the first step in the process
Refinement

- Defines integer multipliers of elements across block interfaces
  - Can only be used with certain solvers
  - Refine: Factor > 1 (enter integer)
  - Coarsen: Factor < 1 (enter fraction – 1/2, 1/3, etc.)

Select refinement edge direction or select “All”

Select block(s) to refine within

Factor = 1/3
Resolve Refinements

- Creates 1-to-1 node connections for refinements done on the blocking
  - Only works on refinement ratio multiples of 3
  - Operates on the unstructured mesh only

Multiple steps of refinement and resolving refinements can be done (3, 9, 27, 1/3, 1/9, 1/27, etc...)
**Edge Parameters – Linked Bunching**

- Link node counts and distribution law to another edge
  - Can link to one master edge or a series of edges with the same end-bounds
  - Select the large edge first
  - Select the small edge on side 1 when selecting the Link Edge

The long edge gets linked to all these shorter edges
**Match Edges**

- Matches the end spacing to another edge
  - Reference edge and target edge must meet at the same vertex
  - Does not link spacing
  - The effect is usually only noticed when the target edge has an end spacing larger than the reference edge.

In this example, the side 1 node spacing of the target edge is set to the same node spacing as side 2 of the reference edge.
Split Face

• *Split Face* is actually a split block operation

• *Split Face* splits the adjacent blocks in a non-active part (usually VORFN), and what is left visible is the end of the split on the visible faces
  - Select Face to split
  - Left click on edge and drag split
  - Split will be normal to the selected edge

Select face

Select edge

Face split is normal to selected edge

VORFN blocks are what gets split
Merge Blocks

- **Merge blocks**
  - Select blocks to merge, then middle click
  - **Apply**
  - You cannot merge blocks of different parts unless you first change them to the same part
Merge Faces

- **Merge Faces**
  - Select 2 corners diagonally across faces to merge
  - **Apply**
  - You cannot select across O-grids because this is a different index direction
  - This actually merges blocks on both sides of the selected faces

Select diagonally across vertices
Output Blocks

- Reduces number of blocks in a multiblock mesh, which reduces solver time
  - Three steps (in order)
    1. Initialize output blocks
    2. Turn on Output blocks
    3. Merge blocks (automatic or manual)

Before 29 blocks

After 8 blocks

Initializing output blocks to the full blocking BEFORE merging blocks will prevent the mesh from being altered when merging blocks

Output blocks can be toggled on and off between the merged blocking and the full blocking
감사합니다.