Multi-Block Structured Meshing and Pre-Processing for OpenFOAM Turbomachinery Analysis

Dr. Chris Sideroff, Pointwise
TOC

• Axial turbine meshing demo
• Advanced feature meshing demo
• Mesh topology
• Mesh quality
• Cyclic setup
• Transforms
• Flow Initialization
• Miscellaneous
Meshing Demo

- Aachen axial rotor
- High-stagger angle
- 1-1 periodic
- No fillets
- No tip gap
More Advanced Applications

• GAMM Francis Runner
  – Highly-3D blade
  – High solidity
  – Turning topology

• DARPA Hi-Re Pump
  – Tip gap w/ round
  – Swept blade
  – Spanwise twisting
  – Fillets
Fillet Topologies

BLADE

fillet

O

H

H

OK

Better

POINTWISE®
Reliable CFD Meshing
Meshing Demo

• DARPA HIREP: Fillet
Why are structured meshes so desirable?
Mesh Topology

For a given # points, hex cells are more efficient

- 5381 points
  - 5200 cells

- 6136 points
  - 11,910 cells
Mesh Topology

Body fitted structured mesh is more accurate
Mesh Topology

Less truncation error when faces aligned with flow gradients

\[ u_{\text{face}} = u_{\text{cell}} + \frac{\partial u}{\partial y} \Delta y + O(\Delta y^2) \]
Mesh Topology

More consistent character across mesh resolutions and machine families.
Mesh Topology
Are structured meshes really that great?

• For turbomachinery flows, they can’t be beat
  – Axial compressor polytropic efficiencies \( \geq 95\% \)
• For high-accuracy, they can’t be beat
  – Aircraft drag predictions < 1 count
• Long-time turbo customer: “We’ve been trying unstructured methods for 10 years and have given up.”
Mesh Topology

When can I get away without using structured meshes?

- My boss told me I can only spend 1 day meshing
- When the flow path is unknown or changes rapidly, structured meshes lose their benefit
- Unstructured meshes provide more flexible resolution control, i.e. localized clustering
- Hybrid meshes are a good compromise between effort and accuracy
  - Prism/hex in important regions (BL’s, wakes, shocks, free-surface)
  - Unstructured elsewhere
What is a good mesh?
Mesh Quality

*How do we determine if we have a good mesh?*

- “A standard rule of thumb is that the element shape should be ‘pleasing to the eye’ ...” *Desktop Engineering, March 2011*
- No single standard, benchmark or metric exists that can succinctly assess the “quality” of a mesh
- Rely on suggested best practices and our own experiences
- Grid dependency studies are good but not trivial
checkMesh

*aspect ratio*

- AR = longest side / shortest side
- Large AR OK if gradients in long direction are small
checkMesh

*skewness*

- Skewness = deviation of CC vector from FC
- Mid-point rule 2\textsuperscript{nd} order accurate if interpolated face values are at face center
checkMesh

*orthogonality*

- Ortho = angular deviation of CC vector from face vector
- Affects interpolation of gradient to FC
Quality Metrics
Are aspect ratio, skewness and ortho sufficient?

AR = 1, Skew = 0, Ortho = 0
AR = 1, Skew = 0, Ortho = 0
My mesh is good, what next?
Cyclic (aka Periodic) BCs

- **cyclic**
  - 1-to-1 point matching
  - Consistent face ordering
  - Constant transformation matrix

- **cyclicGgi**
  - Non-point matching
  - Face ordering unnecessary
  - Non-constant transformation matrix

Constant xfrm

Non-constant xfrm
cyclicGgi
polyMesh/boundary

perPres
{
  type cyclicGgi;
nFaces 4200;
startFace 1266360;
shadowPatch perSuc;
zone perPresZone;
rotationAngle 8.7805;
rotationAxis (1 0 0);
separationOffset (0 0 0);
bridgeOverlap false;
}

perSuc
{
  type cyclicGgi;
nFaces 4200;
startFace 1270560;
shadowPatch perPres;
zone preSucZone;
rotationAngle 8.7805;
rotationAxis (1 0 0);
separationOffset (0 0 0);
bridgeOverlap false;
}

Need to create these
Sets

Groups of points, faces or cells

- General form:

  cellSet|faceSet|pointSet <setName> <action> <source>

where <action> is one of

- list : prints the contents of the set
- clear : clears the set
- invert : inverts the set
- remove : remove the set
- new <source> : sets to set to the source set
- add <source> : adds all elements from the source set
- delete <source> : deletes
- subset <source> : combines current set with the source set
Sets cont.

Many types of sources

- Points
  - boxToPoint
  - cellToPoint
  - faceToPoint
  - labelToPoint
  - nearestToPoint
  - pointToPoint
  - setToPoint
  - surfaceToPoint
  - zoneToPoint

- Faces
  - boundaryToFace
  - boxToFace
  - cellToFace
  - faceToFace
  - labelToFace
  - normalToFace
  - patchToFace
  - pointToFace
  - setToFace

- Cells
  - boxToCell
  - cellToCell
  - cylinderToCell
  - faceToCell
  - faceZoneToCell
  - fieldToCell
  - labelToCell
  - nbrToCell
  - nearestToCell
  - pointToCell
  - regionToCell
  - rotatedBoxToCell
  - setToCell
  - shapeToCell
  - sphereToCell
Sets and Zones

*Can be used 3 ways*

1. Interactively: `setSets`
2. In batch: `setSets -batch <my_batch_file>`
3. Using `pointSet/faceSet/cellSet` from CLI with appropriate `pointSetDict/faceSetDict/cellSetDict` in system/

- Refer to:
  ```
  $FOAM_APP/utilities/mesh/manipulation/{point,face,cell}Set/
  {point,face,cell}SetDict
  ```

- Last step, convert sets to zones:
  ```
  setsToZones -noFlipMap
  ```
Mesh Transformations

- `transformPoints` <-<type> -<options>

  Types:
  - scale “(xScaleFactor yScaleFactor zScaleFactor)”
  - translate “(dX dY dZ)”
  - rotate “((x0 y0 z0) (x1 y1 z1))”
  - Plus a couple more

- E.g. change from inches to meters
  `transformPoints` -scale “(0.0254 0.0254 0.0254)”

- E.g. change from X rotation axis to Z
  `transformPoints` -rotate “((0 0 1) (0 1 0))”
  `transformPoints` -rotate “((1 0 0) (0 0 1))”
Measured Data as BC
1D data from .csv file as boundary condition

rotor_inlet
{
    type profile1DfixedValue;
    fileName inletData.csv;
    fileFormat "turboCSV";
    interpolateCoord "R";
    fieldName "Velocity";
}

• fieldNames:
  – Velocity
  – K
  – Epsilon
  – Pressure
• Z-axis only
• Interpolation only

Sample file:

[Data]
R, Velocity Axial, Velocity Radial, Velocity Circumferential
0.265,0,0,0,297853.515
0.294639972,10.77936539,-0.34820352,10.35051285,294904.4082
...

http://openfoamwiki.net/index.php/Sig_Turbomachinery_Library_OpenFoamTurbo
# Initialization Tools

<table>
<thead>
<tr>
<th>simpleSRFFoam</th>
<th>MRFSimpleFoam, etc</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addRotation</strong></td>
<td><strong>addSwirlAndRotation</strong></td>
</tr>
<tr>
<td>- Add rotational velocity to 0/Urel</td>
<td>- Add rotational velocity to 0/U</td>
</tr>
<tr>
<td>- Specify values in constant/swirlAndRotationProperties</td>
<td>- Specify values in constant/swirlAndRotationProperties</td>
</tr>
<tr>
<td><strong>potentialTurbFoam</strong></td>
<td><strong>potentialFoam</strong></td>
</tr>
<tr>
<td>- Potential flow solver for U_{rel}</td>
<td>- Potential flow solver for U</td>
</tr>
</tbody>
</table>

* from Brent Craven, PSU ARL
Miscellaneous

Rotating solver settings and parallel processing

• constant/SRFProperties
  – Single rotating reference frame
• constant/MRFZones
  – Multiple reference frames (stationary + moving)
• constant/dynamicMeshDict
  – Physically moving (i.e. rotating mesh)
  – Cell zone *must* be called movingCells
• system/decomposeParDict
  globalFaceZones( perPresZone perSucZone );
  – Keeps a copy of cells adjacent to faces zones on each CPU
  – Use for cyclicGgi zones
Happy TurboFOAM’ing!

Contact me with Q’s
cnsidero@pointwise.com