Rotation – MRF versus AMI

Advanced course

Legal notes:
- This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OpenFOAM® and OpenCFD® trade marks. OpenFOAM® is a registered trade mark of OpenCFD Limited, a wholly owned subsidiary of the ESI Group.
- This content was made in 2014 and may contain incorrect or outdated information. The reader is solely responsible for his or her use of this information and AirShaper cannot be held liable for any damages.

www.airshaper.com
Content

• Comparison: MRF versus AMI
• Mesh
• Model properties
• Boundaries
• Solver properties
• Field entries
## Comparison: MRF versus AMI

<table>
<thead>
<tr>
<th></th>
<th>MRF</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation</td>
<td>Virtual</td>
<td>Real</td>
</tr>
<tr>
<td>Mesh</td>
<td>Stationary</td>
<td>Moving</td>
</tr>
<tr>
<td>Solver</td>
<td>Any solver</td>
<td>Specific ...DyM... Solvers only</td>
</tr>
<tr>
<td>Time</td>
<td>Steady &amp; unsteady state</td>
<td>Unsteady state only</td>
</tr>
<tr>
<td>Web</td>
<td><a href="http://www.openfoam.org/version2.2.0/fvOptions.php">Link</a></td>
<td><a href="http://www.openfoam.org/version2.3.0/ami.php">Link</a></td>
</tr>
</tbody>
</table>
Mesh

- MRF: create a cellZone for the rotating region
- AMI:
  - snappyHexMesh: add boundary patch for the rotating region cellZone
  - This patch will be used for the mesh interface generation

```plaintext
rotatingRegion
{
    level (4 4);
    faceType boundary;
    cellZone rotatingRegion;
    faceZone rotatingRegion;
    cellZoneInside inside;
}
```

// Whether any faceZones (as specified in the refinementSurfaces) are only on the boundary of corresponding cellZones or also allow free-standing zone faces. Not used if there are no faceZones.
allowFreeStandingZoneFaces true;
Mesh

• AMI:
  – Extra dictionary “createPatchDict”: convert the boundary of the rotating region into both sides of the AMI
  – AMI mesh must be of high quality!
Model properties

- MRF: no modified properties
- AMI:
  - extra dynamicMeshDict file under \constant
  - Specify cellZone, rotation axis and speed (rad/s)

```plaintext
dynamicFvMesh solidBodyMotionFvMesh;
motionSolverLibs ( "libfvMotionSolvers.so" );
solidBodyMotionFvMeshCoeffs
{
    cellZone rotatingRegion;
    solidBodyMotionFunction rotatingMotion;
    rotatingMotionCoeffs
    {
        origin (0 0 0);
        axis (0 0 1);
        omega 151.84; // rad/s --> komt van 1450 rpm
    }
}
```
Boundaries

• MRF: no modified properties
• AMI: extra boundaries defined under O/U...P
Solver properties

• controlDict
  – MRF:
    • simpleFoam (others possible): pseudo time
Solver properties

- controlDict
  - AMI:
    - pimpleDyMFoam: real time – automatic time step possible, based on maximum Courant number

```plaintext
application pimpleDyMFoam;
startFrom startTime;
startTime 0;
stopAt endTime;
endTime 0.2;
deltaT 1e-6;
writeControl adjustableRunTime;
writeInterval 0.01;
maxCo 2;
```
Solver properties

• fvSolutions, fvSchemes: depend on solver
• MRF:
  – fvOptions: extra file under \system to describe the rotation
  – Specify cellZone, rotation axis and speed (rad/s)
Field entries

• MRF:
  – Rotor: movingWallVelocity

• AMI:
  – extra entries for the AMI interfaces
  – Required for all fields

```plaintext
AMI1
{
    type    cyclicAMI;
    value   $internalField;
}
AMI2
{
    type    cyclicAMI;
    value   $internalField;
}
```