

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- This content was made in 2014 and may contain incorrect or outdated information.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.

Content

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

Comparison: MRF versus AMI

	MRF	AMI
Rotation	Virtual	Real
Mesh	Stationary	Moving
Solver	Any solver	Specific ...DyM... Solvers only
Time	Steady & unsteady state	Unsteady state only
Web	http://www.openfoam.org/version2.2.0/fvOptions.php	http://www.openfoam.org/version2.3.0/ami.php

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

```
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!

```
{
  //- Master side patch
  name          AMI1;
  patchInfo
  {
    type          cyclicAMI;
    matchTolerance 0.0001;
    neighbourPatch AMI2;
    transform      noOrdering;
  }
  constructFrom patches;
  patches (rotatingRegion);
}

{
  //- Slave side patch
  name          AMI2;
  patchInfo
  {
    type          cyclicAMI;
    matchTolerance 0.0001;
    neighbourPatch AMI1;
    transform      noOrdering;
  }
  constructFrom patches;
  patches (rotatingRegion_slave);
}
```

Model properties

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

```
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

```
application    simpleFoam;  
startFrom      startTime;  
startTime      0;  
stopAt         endTime;  
|  
endTime        500;  
deltaT         1;
```


Solver properties

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

```
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)

```
MRF1
{
    type            MRFSource;
    active          true;
    selectionMode   cellZone;
    cellZone        rotatingRegion;

    MRFSourceCoeffs
    {
        origin      (0 0 0);
        axis         (0 0 1);
        omega        151.784; // 1450 rpm equals 151.84 rad/s
    }
}
```

Field entries

- MRF:
 - Rotor: movingWallVelocity
- AMI:
 - extra entries for the AMI interfaces
 - Required for all fields

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

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