Automated moving mesh techniques in CFD
Application to fluid-structure interactions and rigid motions problems

MANUELA PROFIR
manuela@crs4.it

April, 11 2012
Approach to Mesh Morphing in CFD framework

Morphing in Computer Graphics

- gradually changes a source shape into a target shape
- medical imaging, scientific visualization, special effects in movies

- Volume mesh = mathematical description of the geometry: vertices, faces, cells
- Surface deformations in moving fluids/solids simulations: warped, self-intersected faces lead to negative volume cells

Research’s objectives

- control mesh deformations
- develop re-meshing strategies
- use Java programming for automation
Research project work-flow

THEORETICAL BACKGROUND

DEFORMABLE MODELS

MESH MORPHING

IMPLEMENTATION CODE

PROGRAMMING LANGUAGE

TECHNICAL SKILLS

ANALYSIS CFD CODE
IMPLEMENTATION TECHNIQUES STARCCM+

CFD SIMULATIONS
FLUID STRUCTURE INTERACTIONS
IMPOSED MOVEMENTS

MESH MORPHING
COMPATIBLE WITH
SIMULATED PHYSICAL FIELDS
Implementation code: CD-adapco’s Star-ccm+

Tools and Novel Concepts

- Numerical algorithms: Cell-based discretization (arbitrary polyhedra), AMG solver, Convergence
- Physical models: Motions (Rigid, Morphing, Solid displacement), Turbulence, Solid Stress
- Flexible mesh manipulation
- Multi-physics, continuum-based modelling
- Separation of physics, geometry and mesh
- Generalized interfaces
- 3D-CAD modeller - design parameters
- Client-Server Architecture - Java Scripting
Mesh Morphing procedure in Star-ccm+

- **Morphing Motion** model (deforming mesh) → **Mesh Morpher**
  - Morpher collects control points and specified displacements from **boundaries** → morpher boundary conditions:
    - Displacement, Grid Velocity, Solid Stress, Floating.
  - Control points $x_i$ and displacements $d(x_i)$ are used to generate an interpolation field:
    \[
    d(x_i) = \sum_{j=1}^{n} \lambda_j \sqrt{\|x_i - x_j\|^2 + c_j^2} + \alpha,
    \sum_{j=1}^{n} \lambda_j = 0 \Rightarrow \lambda_j, \alpha
    \]
  - The interpolation field applies to all the mesh vertices:
    \[
    d(x) = \sum_{j=1}^{n} \lambda_j \sqrt{\|x - x_j\|^2 + c_j^2 + \alpha}
    \]
  - Final adjustments to mesh vertices on or near boundaries
Morpher boundary conditions

Displacement

\[ \delta_r = \begin{cases} 
0.00625 x_z, & x_z < 3 \\
0.00625(6 - x_z), & x_z \geq 3 
\end{cases} \]

Grid Velocity

\[ v_r = \begin{cases} 
-0.003 x_z, & x_z < 3 \\
-0.003(6 - x_z), & x_z \geq 3 
\end{cases} \]
Morpher boundary conditions: Solid Stress in FSI

Pipe deformation/oscillation

- **Morphing**: fluid mesh deforms under external load
- **Solid displacement**: body load applied to the solid pipe
Morpher Boundary Conditions: Solid Stress in FSI

Imposed solid displacement

\[ \delta(t) = [0.0, ((0 < z < l) \cdot (2 \sin t \sin^2 \left( \frac{z}{l} \pi \right) : 0), 0.0] \]

<table>
<thead>
<tr>
<th>Pipe/water system</th>
<th>Cells</th>
<th>Solid stress</th>
<th>Imposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured-trimmed</td>
<td>30 K</td>
<td>( \delta_y = 0.2 mm )</td>
<td>( \delta_y = 1 cm )</td>
</tr>
<tr>
<td>Unstructured-poly</td>
<td>15 K</td>
<td>( \delta_y = 0.15 mm )</td>
<td>( \delta_y = 0.5 cm )</td>
</tr>
</tbody>
</table>
Morphing and rigid motions

- "Piston" subject to a periodical translation motion
- Behaviour of surrounding fluid mesh when the piston is moving?
- Keep under control the expected deformations of the fluid mesh!

\[ \delta y = 0.01 \sin t \sin^2 \left( \frac{z}{l} \pi \right), \quad 0 \leq z \leq l \]

\[ v_z(t) = A \omega \sin(\omega t) \]
Bad mesh deformations - Negative volume cells

- Large motions determine cells degeneracy! Re-mesh needed!
Mesh quality metrics: **Face Validity**

- Face normals must point away from cell centroid.
- Perfect cells: \( FV = 1 \). Minimum acceptable value: \( FV = 0.51 \).
- Face Validity metric description (reprinted from User Guide [?]):

![Good Cell](image1)

![Bad Cell](image2)
Negative volume cell with low face validity
Problems arising: Non-planar faces loose feature curves!
Geometry is not preserved!
Re-meshing strategy: surface mesh extraction
Re-meshing strategy: update CAD geometry

Solution Time 0.1 (s)

Cell Quality

0.0000 0.20000 0.40000 0.60000 0.80000 1.00000

X Z

STAR-CCM+
Re-meshing strategy: CAD geometry update

Automation of re-meshing with external loops in a Java macro:

1. Run the simulation for one time step.
2. Control the condition about the mesh quality criterion.
3. If the criterion is satisfied, take one time step forward; otherwise:
4. Regenerate the volume mesh, including:
   I. prepare a report measuring the exact performed displacement;
   II. in the CAD construction, apply a translation to the piston body and expose as design parameter the translation vector;
   III. the macro reads the displacement’s value and assigns it to the translation vector, updating the initial position of the piston;
   IV. regenerate the surface and volume meshes, taking the most recent boundaries as starting point (i.e. the updated piston position).
Mesh quality metrics: **Cell Quality**

- Depends on the relative geometric distribution of the centroids of cells neighbours of a face and also on the cell faces orientation.
- Perfect cells: $CQ = 1.0$. Degenerate cells: $CQ = 0$.
- Cell Quality metric description (reprinted from User Guide [?]):

![Good Cell](image1)

![Bad Cell](image2)
Translational periodical motion

Grid Velocity for Morphing motion

\[ v(t) = A \omega \sin(\omega t) \]
Improved strategy: optimised re-meshing

- Divide domain into three regions, by subtraction, intersection and imprinting boolean operations
- Central region: **Translation motion**
- Lateral regions: **Morphing motion**
- MBC: **Floating** for Wall boundaries, **Grid Velocity** for interfaces
- Update geometry for re-meshing
- Re-mesh only lateral regions

- faster mesh generation
- less re-meshing
Update lateral bodies through translations

Translate lateral bodies:
\[ \Delta z = p_1 - p_0 \]

- \( p_0 \) = Init. Position
- \( p_1 \) = Max. Position of central region

Solution Time 0.1 (s)
Central body: 2-way asymmetric extrusion distances

\[ d = d_0 + \Delta z = p_1 \]
\[ d' = d_0 - \Delta z = 2p_0 - p_1 \]

Java implementation

```java
ExtrusionMerge extrusionMerge_0 = ((ExtrusionMerge) cadModel_0.getFeatureManager().getObject("Central");
extrusionMerge_0.getDistance().setValue(d);
extrusionMerge_0.getDistanceAsymmetric().setValue(d');
```
Update lateral bodies with extrusion distances
**Floating** - Sliding interfaces

**Grid Velocity** - Interfaces with Central region

**Fixed** - Interfaces with external region

**Translation velocity/Grid velocity**

\[ v(t) = [0.0, A \frac{\pi}{2} \sin(\frac{\pi}{2} t + \frac{\pi}{2}), 0.0], \]

\[ A \leq \Delta z. \]
Solution Time 10 (s)
Solution Time 34 (s)
Re-meshing strategies on complex geometries
Surface extraction strategy on complex geometries
double d = maxReport_0.getReportMonitorValue();
ExtrusionMerge extrusionMerge_0 = ((ExtrusionMerge)
  cadModel_0.getFeatureManager().getObject("B");

extrusionMerge_0.getDistance().setValue(d);
extrusionMerge_0.getDistanceAsymmetric().setValue(0.34-d);

MoveBodyFeature moveBodyFeature_0 =
  ((MoveBodyFeature) cadModel_0.getFeatureManager().
    getObject("MoveBody 5");
CadModelCoordinate cadModelCoordinate_0 =
moveBodyFeature_0.getTranslationVector();
cadModelCoordinate_0.setCoordinate(new DoubleVector(new double[] {0.0, 0.0, d-0.33}));
Pressure (Pa)
4245.1
3388.9
2532.7
1676.6
820.39
-35.779

Solution Time 5 (s)
Conclusions and further work

Results

- Control of the mesh deformations coherently with the imposed displacements.
- Optimised computation/mesh regeneration time, through a selective (region-wise) re-meshing.
- Extend the software’s capabilities, by the coupling with the Java scripts.
- Automation of the re-meshing procedures.
- Regular development of the fluid flow, even in presence of morphing and re-meshing operations.
- Combine sliding interfaces with moving/deforming domains, maintaining or quickly re-creating good quality mesh.
Further work

- Integration of the moving parts in a larger closed loop
- Coupling with the physics of nuclear facilities
- Reproduce the movement of the control/safety rods system in relation to applied physical forces (fluid drag, buoyancy, gravity).
Acknowledgments & References

**RAS**

This project is supported by RAS (Regione Autonoma della Sardegna) through a grant co-funded by PO Sardegna FSE 2007-2013, L.R.7/2007 Promozione della ricerca scientifica e dell’innovazione tecnologica in Sardegna.

**CRS4**

- *Energy & Environment Program*
- Process Engineering and Combustion
- Vincent Moreau