Dynamic Mesh Handling in OpenFOAM

Hrvoje Jasak

h.jasak@wikki.co.uk

Wikki Ltd, United Kingdom and
FSB, University of Zagreb, Croatia

7-9th June 2007
Handling Complex Geometry in OpenFOAM

- Complex geometry is a rule, not exception
- Polyhedral cell support
  - A cell is a polyhedron bounded by polygons
  - Consistent handling of all cell types
  - More freedom in mesh generation
- Interfaces to all major mesh generators

Automatic Mesh Generation

- Two techniques under active development, based on STL surface geometry description
- Polyhedral dual mesh from Delaunay triangulation and cut cell technique

Dynamic Mesh Handling

- Supporting cases of deforming geometry using mesh motion solvers
- For extreme mesh deformation, mesh topology is modified during the simulation
Handling Shape Change: Problem Specification

- Initial valid mesh is available
- Time-varying boundary motion
  - Prescribed in advance: e.g. IC engines
  - Part of the solution: surface tracking
- Need to determine internal point motion based on prescribed boundary motion
- Mesh in motion must remain valid: face and cell flip must be prevented by the solution algorithm and control of discretisation error

Solution Technique

- Point position provided by solving an equation where motion of the boundary acts as the boundary condition for the motion equation
- Choice of motion equation: Laplace or pseudo-solid equation
- Details of mesh grading controlled by variable diffusivity, based on distance to the moving boundary, cell distortion or similar criteria
- Current implementation allows multiple solver techniques. Experience shows cell-based methods fail in interpolation; spring analogy technique is unreliable
Solving Mesh Motion Equation

- Vertex-based (FEM) mini-element discretisation taken out by OpenCFD Ltd. without appropriate replacement
  - Well tested, parallelised and in active use: essential for IC engines simulation
  - Replaced by inferior cell-based technique with interpolation issues
  - It will remain present in the public version of OpenFOAM

- Cell-based motion solver
  - Faster than the FEM technique: segregated cell-based FVM solver with interpolation
  - For simple meshes and smooth boundary motion, it is useful; it will fail for more complex motion cases

- Variable diffusivity operates in the same manner in both techniques
Automatic Mesh Motion

Effect of Variable Diffusivity: Oscillating Airfoil Simulation

- Initial mesh; constant diffusivity
- Distance-based diffusivity $1/l^2$; deformation energy; distortion energy
Topological Mesh Changes

Topological Changes on Polyhedral Meshes

- For extreme cases of mesh motion, changing point positions is not sufficient to accommodate boundary motion and preserve mesh quality.
- Definition of a **topological change**: number or connectivity of points, faces or cells in the mesh changes during the simulation.
- Motion can be handled by the FVM with no error (moving volume), while a topological change requires additional algorithmic steps.
- Cell insertion and deletion will formally be handled as a combination of mesh motion (collapsing cells and faces to zero volume/area) and a change in connectivity after the face and cell collapse.
Implementation of Topological Changes in OpenFOAM

- **Primitive mesh operations**
  - Add/modify/remove a point, a face or a cell
  - This is sufficient to describe all cases, even to build a mesh from scratch
  - ... but using it directly is very inconvenient

- **Topology modifiers**
  - Typical dynamic mesh operations can be described in terms of primitive operations. Adding a user-friendly definition and triggering logic creates a “topology modifier” class for typical operations
  - Implemented topology modifiers
    - Attach-detach boundary
    - Cell layer additional-removal interface
    - Sliding interface
    - Error-driven mesh refinement

- **Dynamic meshes**
  - Combining topology modifiers and user-friendly mesh definition, create dynamic mesh types for typical situations
  - Examples: mixer mesh, IC engine mesh (valves + piston) etc.

- User define own modifiers and dynamic meshes. Example: crack propagation
Example: Layer Addition-Removal

- Layer addition-removal mesh modifier removes cell layers when the mesh is compressed and adds cells when the mesh is expanding.

- Definition consists of:
  - Face zone, defining oriented active surface for cell addition and removal.
  - Minimum and maximum mesh thickness in front of the surface.

- Triggering condition derived from mesh motion (handled separately).

- Example definition: `constant/polyMesh/meshModifiers`

```plaintext
right
{
    type layerAdditionRemoval;
    faceZoneName rightExtrusionFaces;
    minLayerThickness 0.0002;
    maxLayerThickness 0.0005;
    oldLayerThickness 0.0;
    active on;
}
```
Example: Sliding Interface

- Definition consists of a master and slave surface as zones and patches
- Additional (empty) point and face zones required for operation
- Allows uncovered master and slave faces to remain as boundaries, with master side dictating the shape in case of imperfect match

```plaintext
mixerSlider
{
  type slidingInterface;
  masterFaceZoneName outsideSliderZone;
  slaveFaceZoneName insideSliderZone;
  cutPointZoneName cutPointZone;
  cutFaceZoneName cutFaceZone;
  masterPatchName outsideSlider;
  slavePatchName insideSlider;
  typeOfMatch integral;
  coupleDecouple off;
  projection visible;
  attached off;
  active on;
}
```
Example: Multi-Valve Engine Mesh

- Even for simple cases, it is easier to speak about problem classes (mixer vessels, engines, moving bodies) rather than working out individual topology modifiers.
- For complex topological changes, multiple interacting topology modifiers are used and need to be synchronised and used in unison with mesh motion.
- Example: engine valve
  - Definition identifies valve stem, top and bottom surface.
  - Topology modifiers: layer addition-removal on top and bottom surface; sliding interface along valve curtain.
  - Valve motion defined in terms of valve lift curves and crank angle degree.

- Engine mesh components
  - Piston class, with motion defined in terms of crank angle degree and cell layering thickness.
  - List of valves, each with own lift curve.
  - Identification of intake and exhaust ducts (possibly allowing removal when valves are closed).

- The user builds the mesh and associates various surfaces to engine components: easy setup after mesh generation.
Example: In-Cylinder Flow with Moving Piston and Valves

- Exhaust and intake stroke in a 2- and 4-stroke engine
- Moving piston and operating valves using topological changes
- Examples by Tommaso Lucchini, Politecnico di Milano