Dynamic Mesh Support in the Finite Volume Method

Hrvoje Jasak

hrvoje.jasak@fsb.hr

FSB, University of Zagreb, Croatia

Introduction

Objective

- Present the derivation and discretisation support for moving mesh simulation
- Present the layout and use of dynamic mesh features in OpenFOAM

Topics

- Definition of a mesh motion problem
- Geometry handling: polyhedral cell support
- Reynolds Transport Theorem and extension to arbitrary volume
- Conservation of space
- Numerical methodology: FVM for scalar transport on a moving mesh
- Deforming mesh strategies
- FE-based automatic mesh motion solver for the FVM
- Topological mesh changes
- Algorithmic implications
- Dynamic mesh classes with examples: 6-DOF bodies, IC engines, FSI
- New features: RBF mesh motion, immersed boundary, Overset grid
- Summary

Background

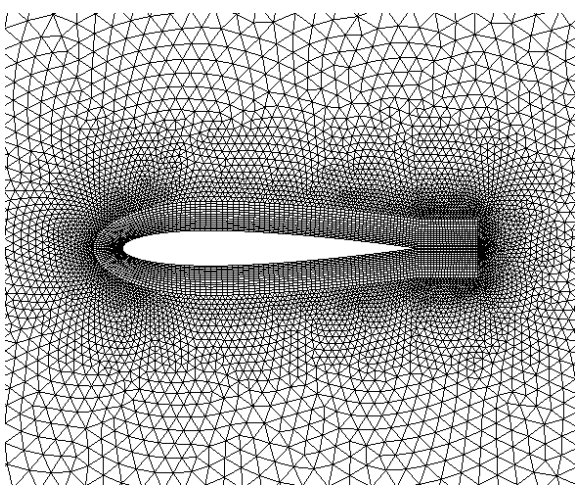
Simulations with Time-Varying Geometry in CFD

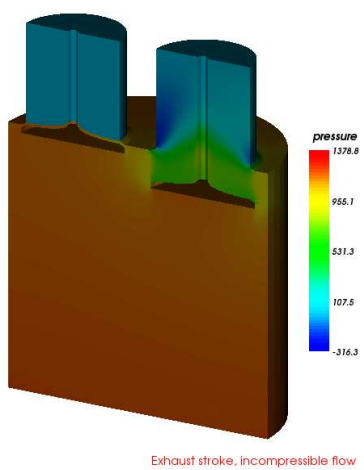
- In many simulations, shape of computational domain changes during the solution, either in a prescribed manner or as a part of the solution
- In cases of **prescribed motion**, it is possible to pre-define a sequence of meshes or mesh changes which accommodate the motion
- . . . but mesh generation now becomes substantially more complex
- **Solution-dependent motion** implies the shape of computational domain is a part of the solution itself: depends on solution parameters
- Examples of solution-dependent motion
  - Contact stress analysis: shape and location of contact region is unknown
  - Free surface tracking simulation: unknown shape of free surface
  - Fluid-structure interaction

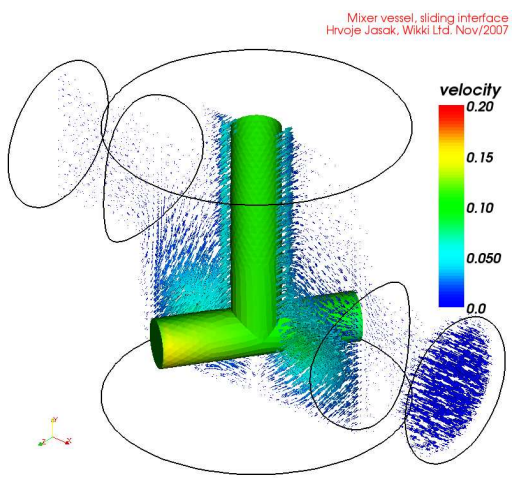
Prescribed Mesh Motion

Example: Prescribed Mesh Motion

- Domain shape is changing during the simulation in a prescribed manner
- Motion is known and independent of the solution
- . . . but it is usually only prescribed at boundaries
- Definition of moving mesh involves point position and mesh connectivity for every point and cell for every time-step of the simulation. This is typically defined with reference to a pre-processor or parametrically in terms of motion parameters (crank angle, valve lift curve, etc.)
- Solution-dependent mesh changes are out of the question: eg. mesh refinement
- Can we simplify mesh generation for dynamic mesh cases?



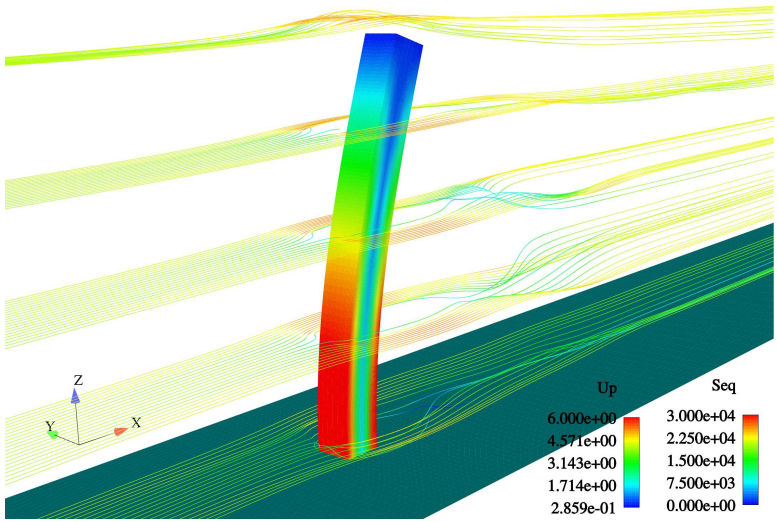


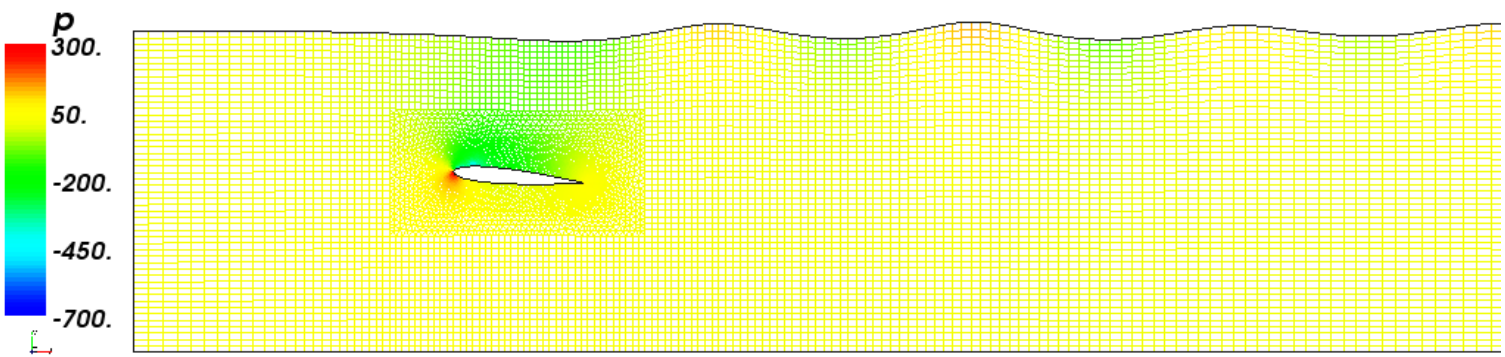


Solution-Dependent Motion

Example: Solution-Dependent Motion

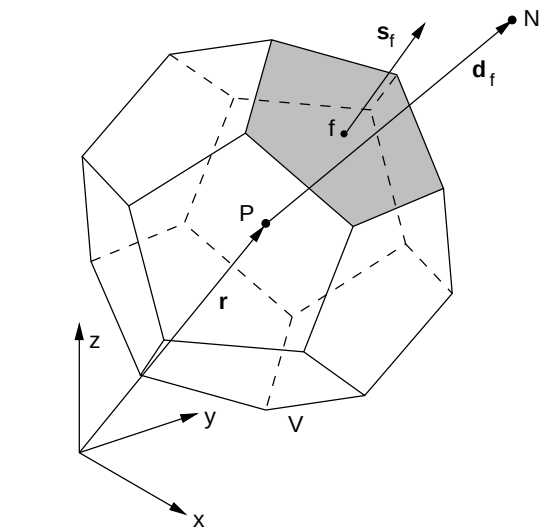
- External shape of the domain is unknown and a part of the solution
- By definition, it is impossible to pre-define mesh motion a-priori
- In all cases, it is the **motion of the boundary** that is known or calculated
- Automatic mesh motion determines the position of internal points based on boundary motion

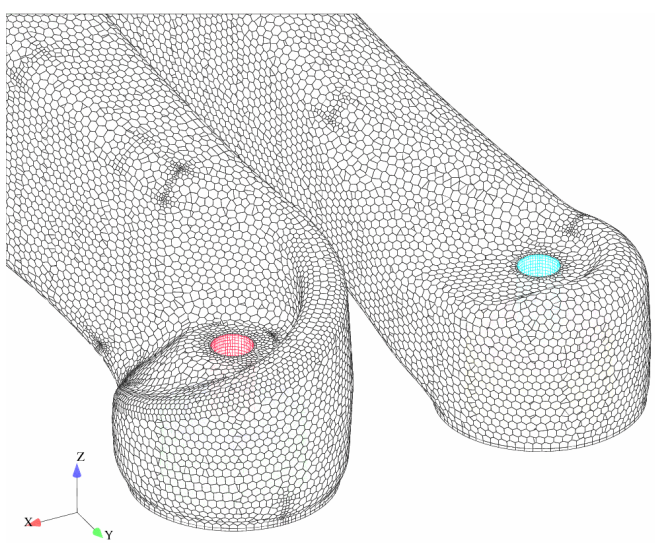




Surface tracking: hydrofoil

Geometry Handling





Handling Complex Geometry in OpenFOAM

- Complex geometry is a rule, not exception
- Polyhedral cell support
  - Polyhedral cell with polygonal faces
  - Consistent handling of all cell types
  - More freedom in mesh generation
- Interfaces to mesh generation packages

Native Automatic Mesh Generation

- Two techniques under active development, based on STL surface geometry description
- Polyhedral dual mesh; surface-adjusted octree hexahedral mesh

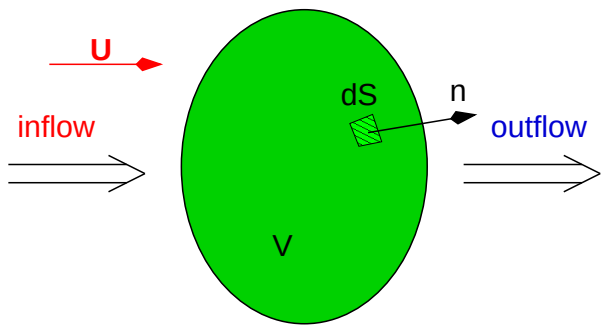
Dynamic Mesh Handling

- Supporting cases of deforming geometry using automatic mesh motion solvers
- For extreme mesh deformation, mesh topology is modified during the simulation using the topology change engine

Reynolds Transport Theorem

Handling Convective Transport

- Reynolds transport theorem is a first step to assembling the standard transport equation
- Examine a region of space: a Control Volume (CV)



The rate of change of a general property ϕ in the system is equal to the rate of change of ϕ in the control volume plus the rate of net outflow of ϕ through the surface of the control volume.

$$\frac{d}{dt}\int_{V_m}\phi\,dV = \int_{V_m}\frac{\partial\phi}{\partial t}\,dV + \oint_{S_m}\phi(\mathbf{n}\bullet\mathbf{u})dS$$

$$\frac{d}{dt}\int_{V}\phi\,dV = \int_{V}\left[\frac{\partial\phi}{\partial t} + \nabla\bullet(\phi\mathbf{u})\right]\,dV$$

Reynolds Transport Theorem

Handling Convective Transport

- Transformation from the surface integral into the volume integral used above is called the Gauss' Theorem

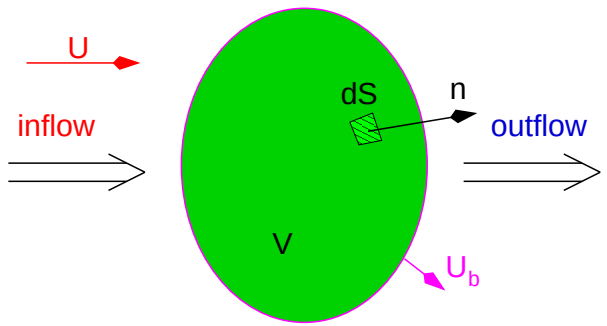
$$\int_{V_P} \nabla \bullet \mathbf{a}\, dV = \oint_{\partial V_P} d\mathbf{s} \bullet \mathbf{a} = \oint_{\partial V_P} d\mathbf{n} \bullet \mathbf{a}\, dS$$

- $\mathbf{u}$ in the equation above represents the **convective velocity**: flux going in is negative ($\mathbf{u} \bullet \mathbf{n} < 0$). The convective velocity in general terms can be considered as a coordinate transformation.
- $\mathbf{u}$ is also a function of space and time: our coordinate transformation is not trivial. Examples: "solid body motion", solid rotation, cases where $\mathbf{u}$ is not divergence-free

Arbitrary Moving Volume

Generalisation: Moving Volume in place of Control Volume

- Arbitrary volume, moving in space, where $\mathbf{u}_b$ is **boundary velocity**



- While deriving equation, we account for net flux through the surface

$$\frac{d}{dt}\int_V \phi\, dV = \int_V \frac{\partial \phi}{\partial t}\, dV + \oint_S \phi\, [\mathbf{n}\bullet(\mathbf{u}-\mathbf{u}_b)]\; dS$$

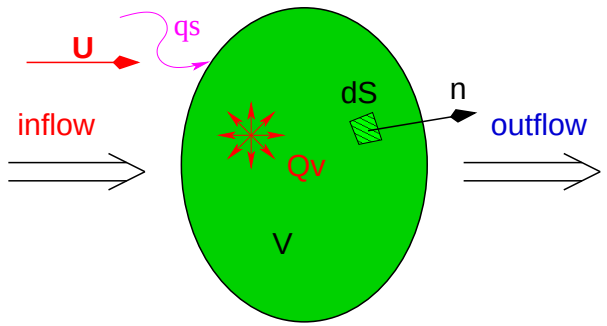
$$\frac{d}{dt}\int_V \phi\, dV = \int_V \left[\frac{\partial \phi}{\partial t} + \nabla\bullet[\phi(\mathbf{u}-\mathbf{u}_b)]\right] dV$$

- In conclusion, all convection flux terms appear with relative velocity $\mathbf{u}_r = \mathbf{u} - \mathbf{u}_b$

Sources and Sinks

Volumetric Source/Sink Terms

- Volume source: distributed through the volume, *e.g.* gravity



$$\frac{d}{dt}\int_V \phi\, dV = \int_V q_v\, dV - \oint_S (\mathbf{n}\bullet\mathbf{q}_s) dS$$

Surface Sources: Diffusive Transport

- Surface source: act on external surface S, *e.g.* heating
- Gradient-based transport is a model for surface source/sink terms

$$\mathbf{q}_s = -\gamma\, \nabla\phi$$

Generic Transport Equation

Generic Transport Equation on a Static Mesh

- Here, $V = const.$ and volumetric integrals can be collapsed

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{temporal derivative}} + \underbrace{\nabla \bullet (\phi \mathbf{u})}_{\text{convection term}} - \underbrace{\nabla \bullet (\gamma \nabla \phi)}_{\text{diffusion term}} = \underbrace{q_v}_{\text{source term}}$$

- Temporal derivative = inertia of the system
- Convection term represents a coordinate transformation
- Diffusion term represents gradient transport
- Sources and sinks account for local production and destruction of ϕ

Generic Transport Equation

Moving Mesh Generic Transport Equation

- On a moving mesh, collapsing the equation into differential form is ugly
- . . . but we plan to work with the integral form

$$\int_V \frac{\partial \phi}{\partial t}\, dV + \oint_S \phi \left[\mathbf{n}\bullet(\mathbf{u} - \mathbf{u}_b)\right]\, dS - \oint_S \gamma(\mathbf{n}\bullet\nabla\phi)\, dS = \int_V q_v\, dV$$

Conservation of Space

- Following the generic transport equation, an auxiliary law is derived: Conservation of Space

$$\int_V \frac{\partial V}{\partial t} - \oint_S (\mathbf{n}\bullet\mathbf{u}_b)\, dS = 0$$

- Mathematically, this is nonsensical: rate of change of volume equals the volume swept by its surface
- . . . but a good check of numerical consistency in discretisation and mesh motion

Discretisation

Temporal Discretisation on a Moving Mesh

- Rate-of-change terms takes into account changing volume

$$\int_V \frac{\partial \phi}{\partial t}\, dV \approx \frac{\phi^n V^n - \phi^o V^o}{\Delta t}$$

- Convection term operates with relative velocity: $\mathbf{u}_r = \mathbf{u} - \mathbf{u}_b$

$$\oint_S \phi \left[\mathbf{n}\bullet(\mathbf{u} - \mathbf{u}_b)\right]\, dS = \sum_f \int_S \phi_f \left[\mathbf{n}\bullet(\mathbf{u} - \mathbf{u}_b)\right]\, dS_f = \sum_f \phi_f F_r$$

where F_r is the **relative flux**

$$F_r = \mathbf{s}_f \bullet (\mathbf{u} - \mathbf{u}_b) = F - F_b$$

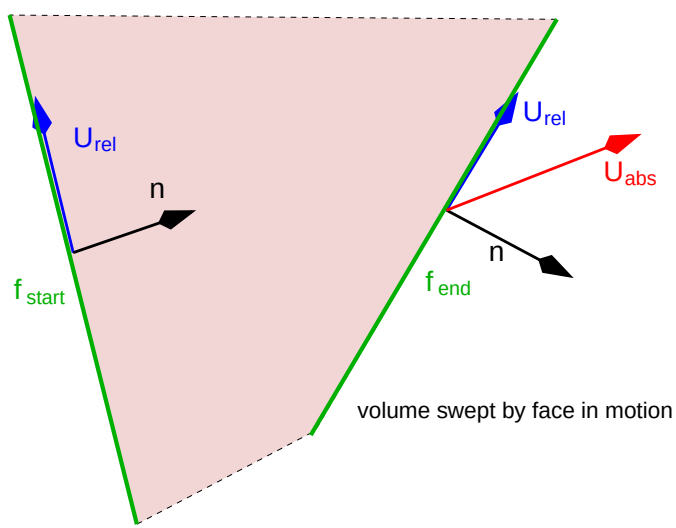
- In practice, $F_b = \mathbf{s}_f \bullet \mathbf{u}_b$ is calculated from geometric considerations: using $\mathbf{u}_b$ would violate the discrete form of space conservation:

$$\frac{V^n - V^o}{\Delta t} - \sum_f F_b = 0$$

Moving Wall Boundary

Prescribing Conditions on a Moving Wall

- All velocity boundary conditions are specified in **absolute velocity**. For static mesh, this presents no problem
- In moving mesh simulations, a fixed value boundary condition should account for **velocity of the wall** $\mathbf{u}_b$
- Knowing $\mathbf{u}_b$ is not always easy: consistency with swept volume
- **Moving Wall Velocity** boundary condition
  1. Specifies boundary velocity in the local (moving) coordinate system
  2. Normal velocity component is adjusted by mesh motion flux



Automatic Mesh Motion Solver

Definition of Automatic Mesh Motion

- Automatic mesh motion will determine the position of mesh points based on the prescribed boundary motion
- Motion will be obtained by solving a **mesh motion equation**, where boundary motion acts as a boundary condition
- Properties of motion equation: preserve spatial consistency
- The “correct” space-preserving equation is a large deformation formulation of linear elasticity . . . but it is too expensive to solve
- Choices for a simplified mesh motion equation:
  - Spring analogy: insufficiently robust
  - Linear + torsional spring analogy: complex, expensive and non-linear
  - Laplace equation with constant and variable diffusivity

$$\nabla \bullet (k \nabla \mathbf{u}) = 0$$

  - Linear pseudo-solid equation for small deformations

$$\nabla \bullet [\mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \lambda \mathbf{I} \, \nabla \bullet \mathbf{u}] = 0$$

Automatic Mesh Motion Solver

Control of Mesh Spacing and Discretisation Error

- Mesh spacing and quality control by **variable diffusivity** (Tuković, 2005)
- Changing diffusivity implies re-distribution of the boundary motion through the volume of the mesh: it is not necessarily good to absorb deformation next to the moving boundary
  - Distance-based methods: $1/L$, quadratic or exponential distance from a moving boundary
  - Quality- and distortion-based methods

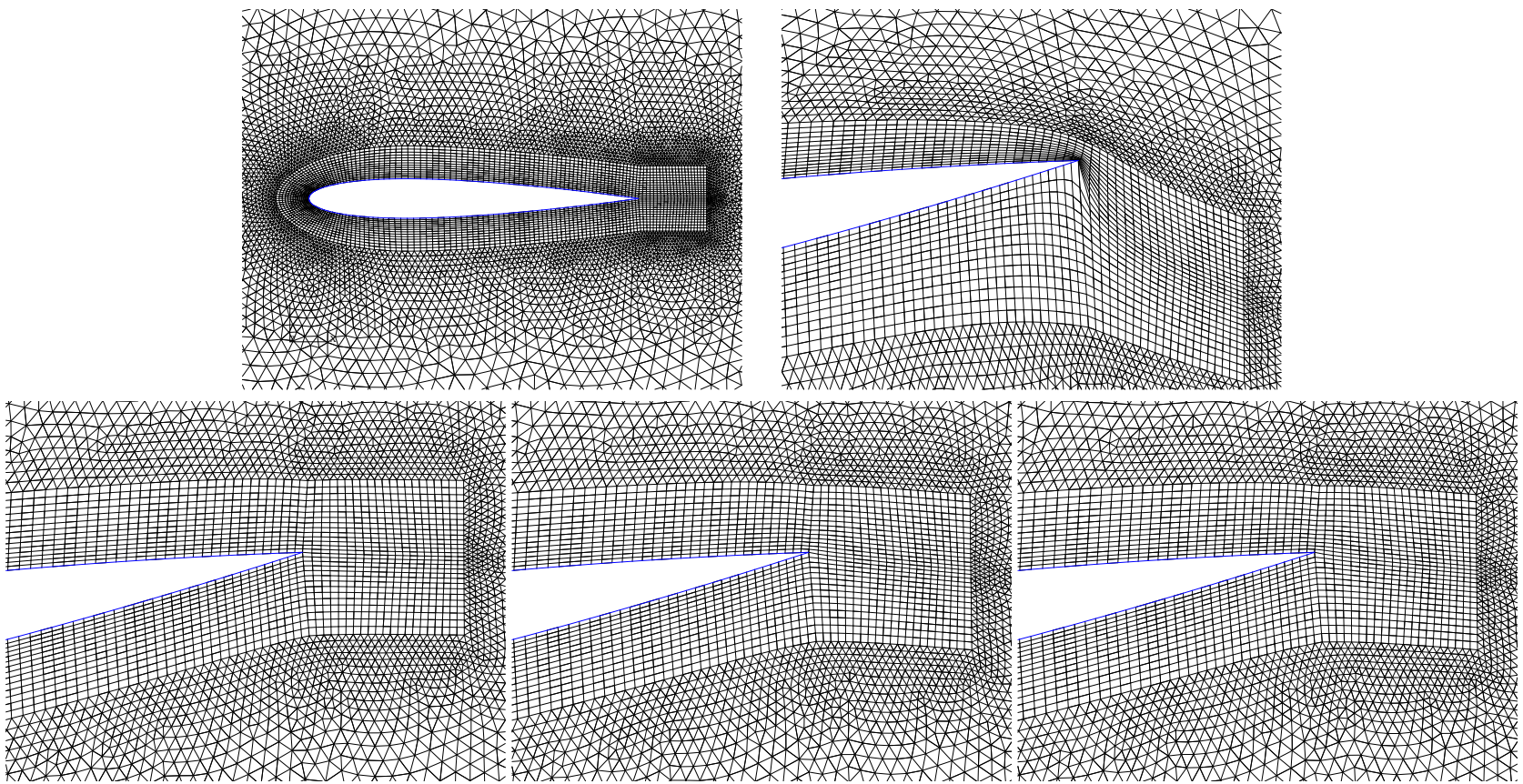
Preserving Mesh Quality

- Definition of valid motion from an initially valid mesh implies that no faces or cells are inverted during motion
- Face area and cell volume on polyhedral meshes is calculated using triangular/tetrahedral decomposition of the cell of face. Motion validity is guaranteed if no triangles or tetrahedra in the cell and face decomposition are inverted during motion
- The rest reduces to controlling the discretisation error in the motion equation

Automatic Mesh Motion

Effect of Variable Diffusivity: Oscillating Airfoil Simulation

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



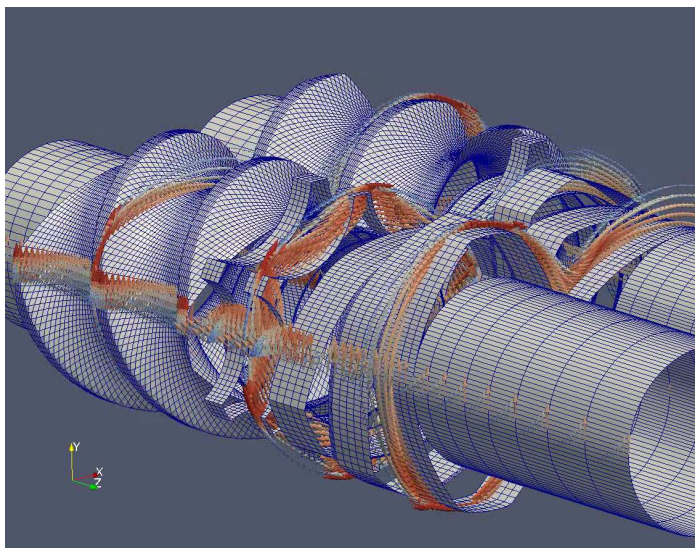
Choice of Motion Solvers

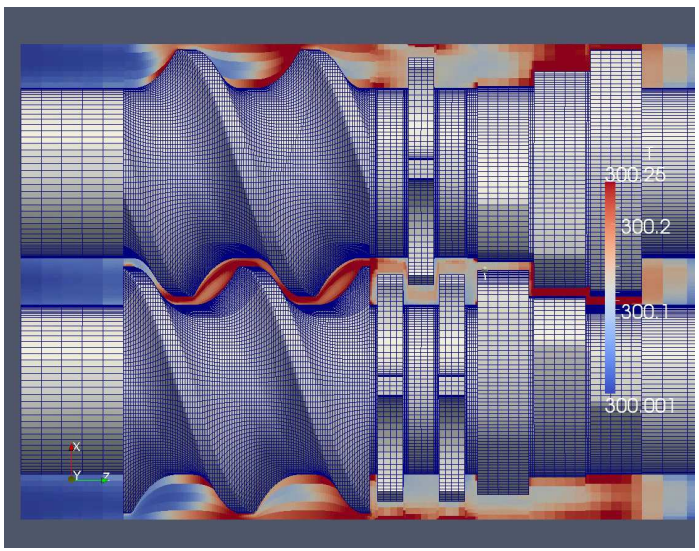
Motion Solver Equations in OpenFOAM

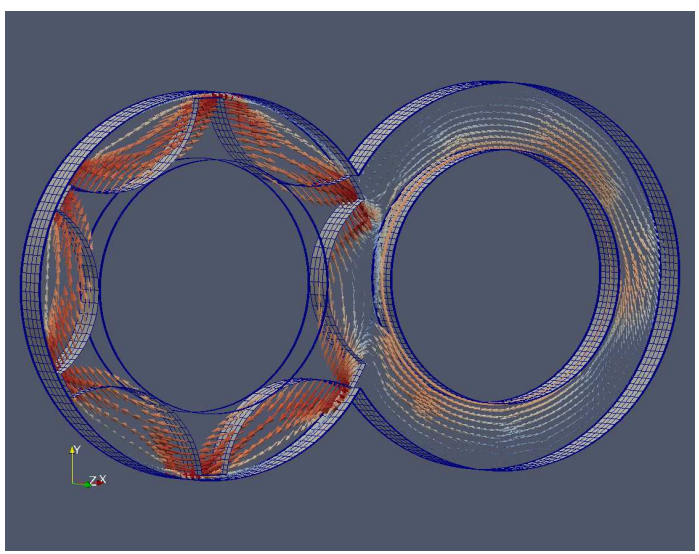
- Second-order Finite Element method with polyhedral support
  - Vertex-based method: no interpolation required
  - FE shape function does not allow tetrahedral or triangular flip: perfect for large boundary deformation
  - Mini-element technique involves enriching the point set: more equations but lower discretisation error
  - Choice of element decomposition
    - **Cell decomposition**: additional point in every cell centroid
    - **Cell-and-face decomposition**: additional point in cell and face centroid
  - Validated and efficient for large deformation, eg. internal combustion engines
- Cell-based methods
  - Solving motion equation on cell centres, interpolating motion into points
  - Special corrections and extrapolation on corners and patch intersections
  - Smaller equation set, but problems in cell-to-point interpolation

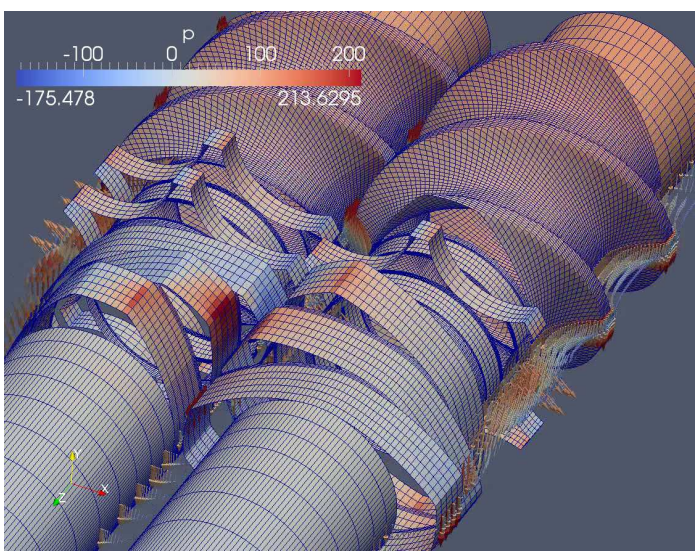
Complex Mesh Motion

Dynamic Mesh Examples of Complex Combination of Motion and Sliding



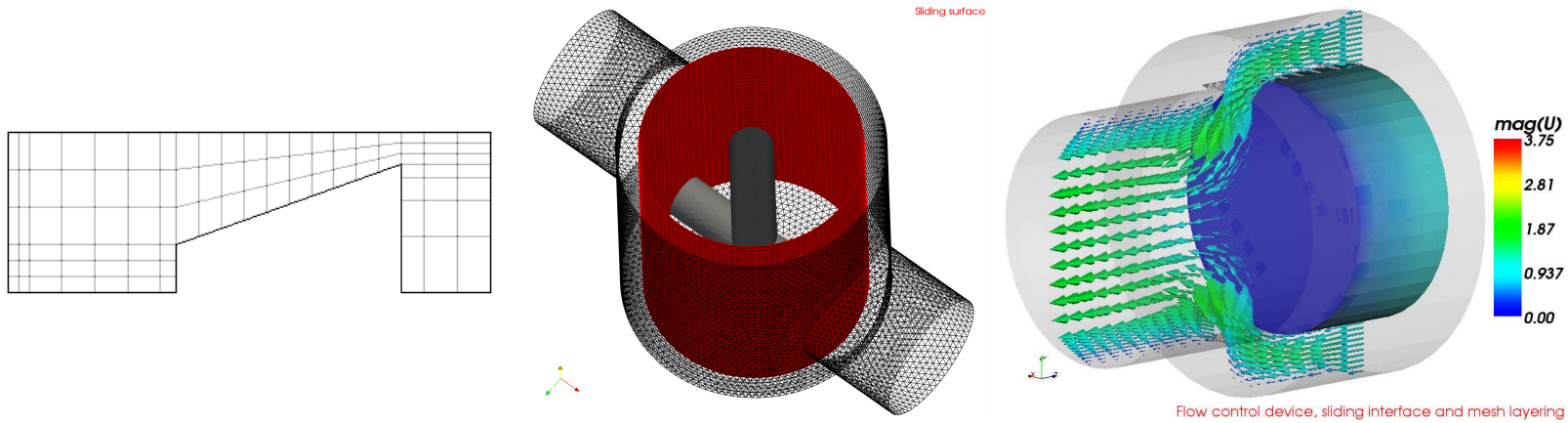






Topological Mesh Changes

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

Topological Mesh Changes

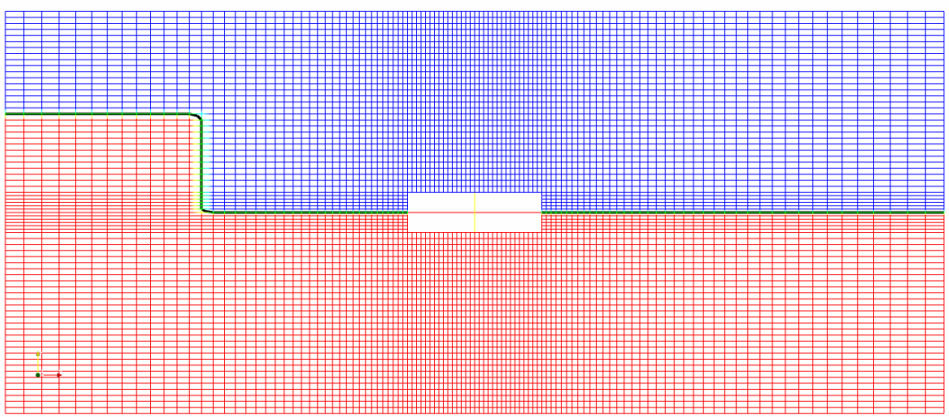
Implementation of Topological Changes in OpenFOAM

- **Primitive mesh operations**
  - Add/modify/remove a point, a face or a cell
  - This is sufficient to describe all cases, even to 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
  - Some implemented topology modifiers
    - Attach-detach boundary
    - Cell layer additional-removal interface
    - Sliding interface
    - Error-driven adaptive mesh refinement
- **Dynamic meshes**
  - Combining topology modifiers and user-friendly mesh definition, create dynamic mesh types for typical situations
  - Examples: mixer mesh, 6-DOF motion, IC engine mesh (valves + piston), solution-dependent crack propagation in solid mechanics

Dynamic Mesh: Floating Body

Example: Single Floating Body in Free Surface Flow (VOF)

- Single phase VOF free surface flow model with accurate pressure reconstruction
- 6-DOF force balance for solid body motion: solving an ODE
- Variable diffusivity Laplacian motion solver with 6-DOF boundary motion as the boundary condition condition



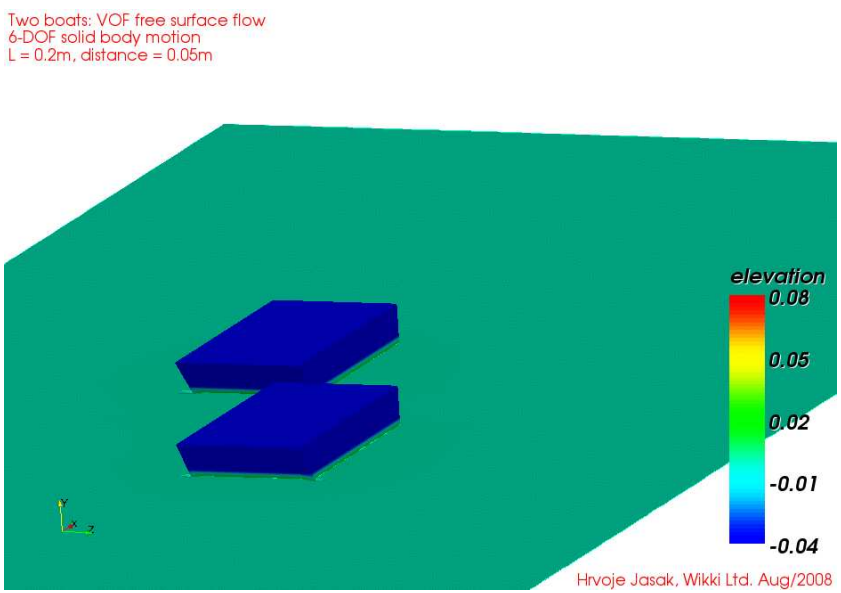
Problem Setup

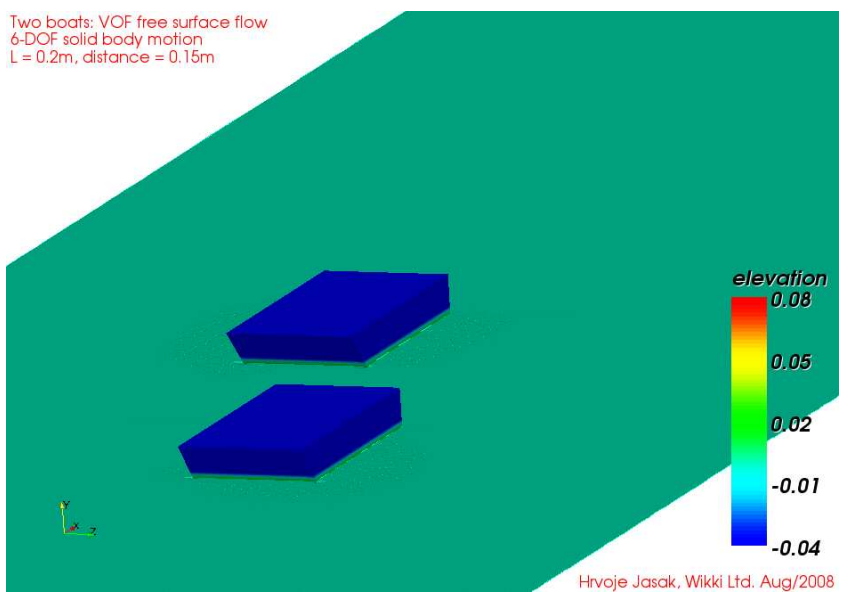
1. Specify mesh, material properties and initial + boundary flow conditions
2. Dynamic mesh type: `sixDofMotion`. Mesh holds `floatingBody` objects
3. A floating body holds 6-DOF parameters: mass, moment of inertia, support, forces
4. Flow solver only sees a `dynamicMesh`: encapsulated motion

Floating Body Simulations

Multiple Floating Bodies

- Problem setup: as above, but with multiple bodies ⌣
- Example: simulation of two bodies in close proximity with different distance
- Elastic support for each boat in the x-direction with linear spring and damping; minor elastic support in the y-direction
- Automatic mesh motion shows its use: adding constrained components is trivial
- Extensive validation effort under way in collaboration with clients and University research groups

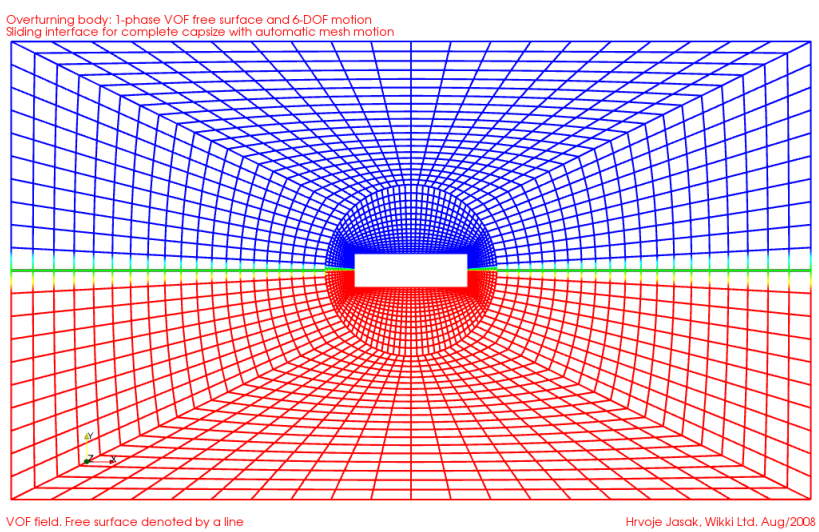


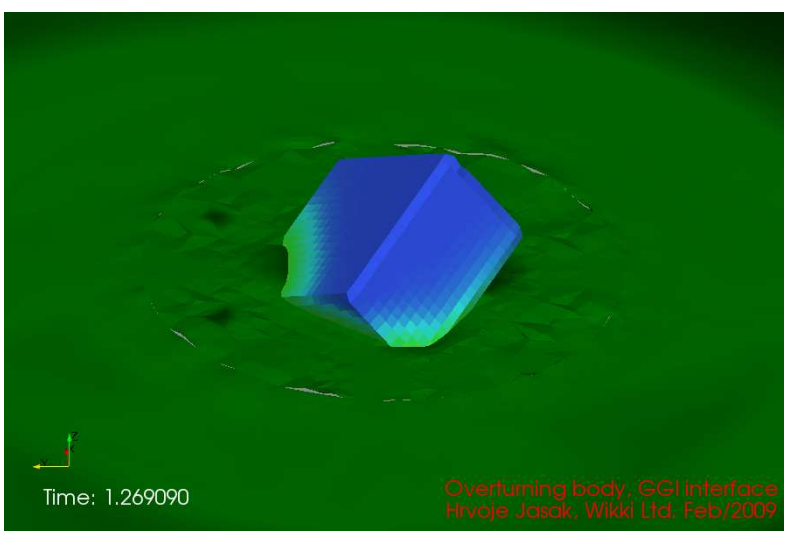


Floating Body Simulations

Capsizing Body with Topological Changes or GGI

- Full capsize of a floating body cannot be handled without topology change
- Mesh motion is decomposed into translational and rotational component
  - External mesh performs only translational motion
  - Rotation on capsize accommodated by a GGI interface
- Automatic motion solver handles the decomposition, based on 6-DOF solution
- Mesh inside of the sphere is preserved: boundary layer resolution
- Precise handling of GGI interface is essential: boundedness and mass conservation for the VOF variable must be preserved

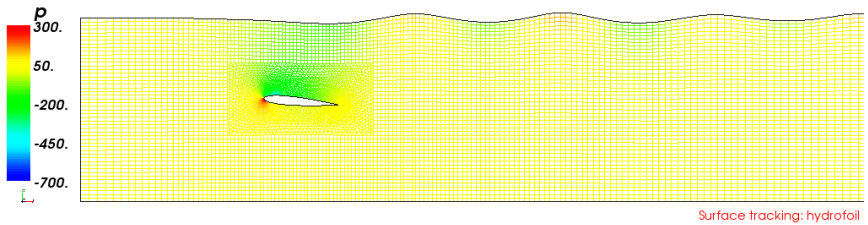




Automatic Motion: Surface Tracking

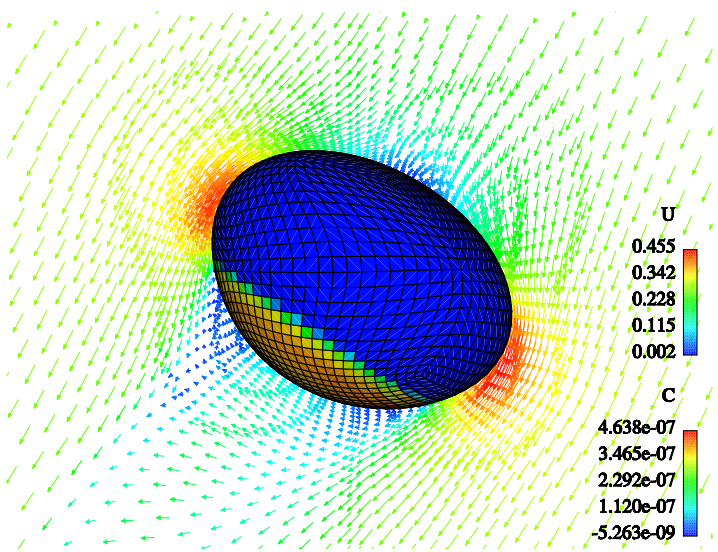
Hydrofoil Under A Free Surface

- Flow solver gives surface displacement
- Mesh adjusted to free surface position



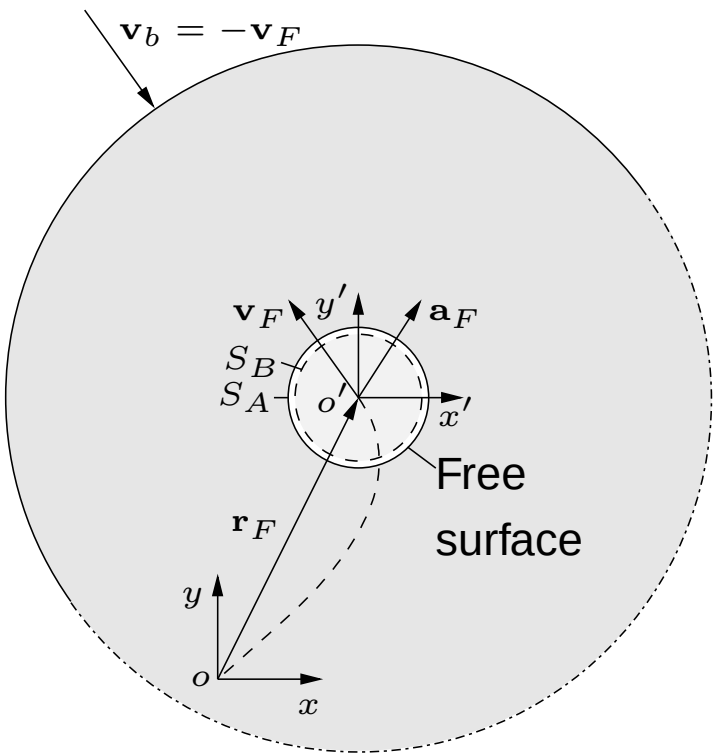
Free-Rising Air Bubble with Surfactants

- Two meshes coupled on free surface



Single Solver, Complex Coupling

- FVM on moving meshes: segregated $p - \mathbf{u}$ solver
- Automatic mesh motion
- Finite Area Method (FAM)



Dynamic Mesh: IC Engine

Example: Multi-Valve Engine Mesh

- For complex topological changes, multiple interacting topology modifiers are used and need to be synchronised and used in unison with mesh motion
- Case setup requires the mesh class to adhere to the “language of the problem”
- Example: Engine valve
  - Definition identifies valve stem, top and bottom surface in geometry
  - 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
  - **Result:** topology modifiers are added automatically for each valve
- 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, piston bowl etc.
- The user builds the mesh and associates various surfaces to engine components: easy setup after initial static mesh generation

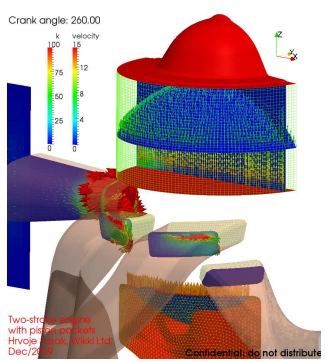
More Dynamic Mesh Examples

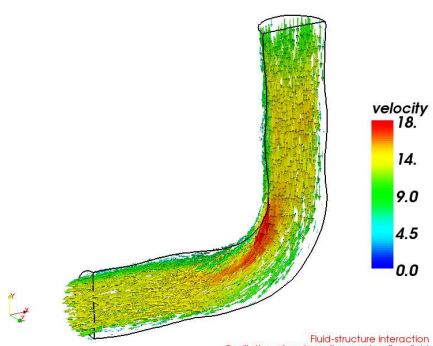
Example: In-Cylinder Flow with Moving Piston and Valves

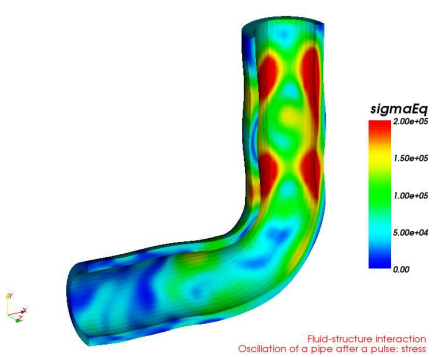
- Exhaust and intake stroke in a 2- and 4-stroke engine
- Moving piston and operating valves using topological changes
- Interacting topological modifiers and mesh motion handled by the mesh class
- Politecnico di Milano: combining mesh deformation, topological changes and re-meshing to achieve optimum resolution and quality (SAE 2007-01-0170)

Example: Fluid-Structure Interaction

- Formally, considerably more complex than floating body cases
- . . . but due to automatic mesh motion only limited changes are needed
- Dynamic mesh class transfers data on surface and uses automatic motion
- (Close coupling: shared matrix format or Reduced Rank Extrapolation)



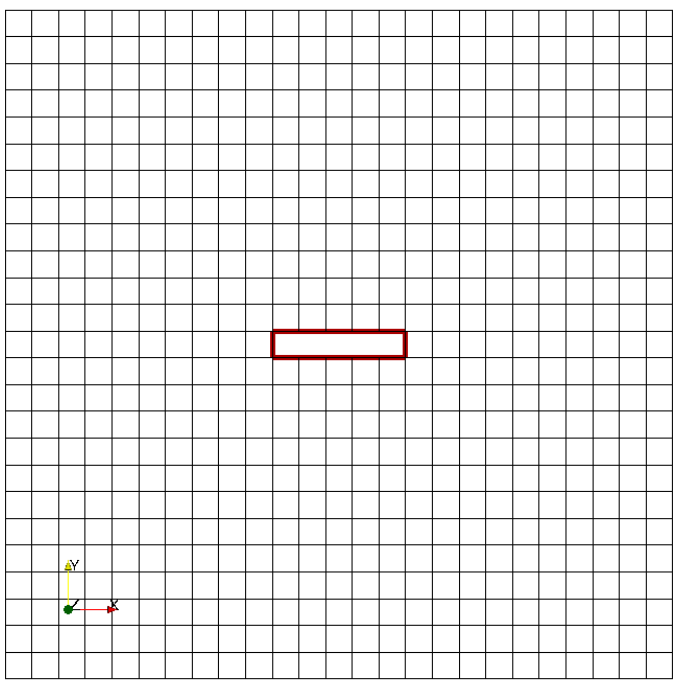




New Interpolation Tools

Radial Basis Function Interpolation

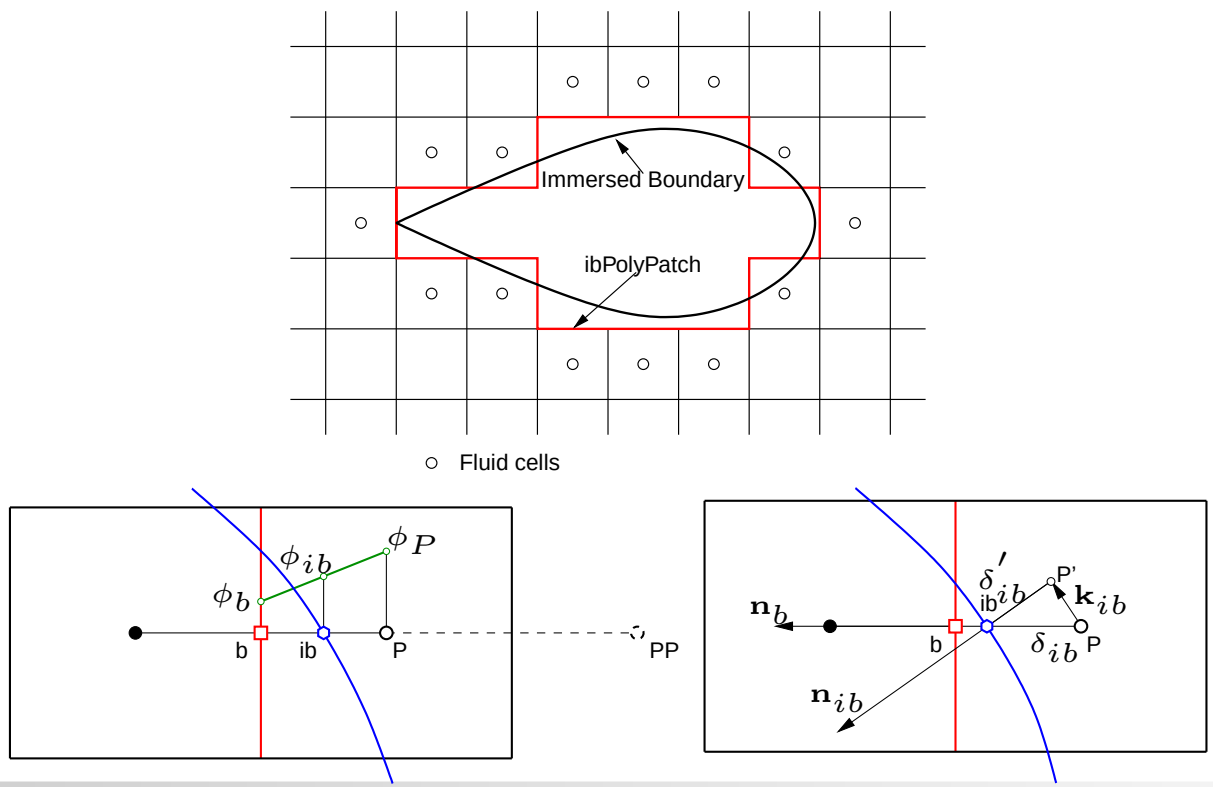
- General interpolation for clouds of points
- Mathematical tool which allows data interpolation from a small set of control points to space **with smoothness criteria**
- Used for mesh motion in cases of large deformation: no cross-over
- Implemented by Frank Bos, TU Delft and Dubravko Matijašević, FSB Zagreb



Immersed Boundary Method

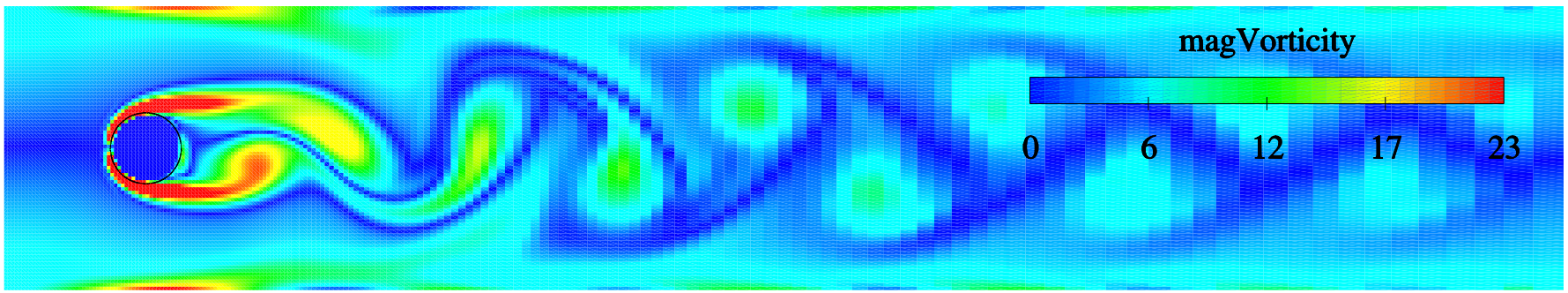
Immersed Boundary Method

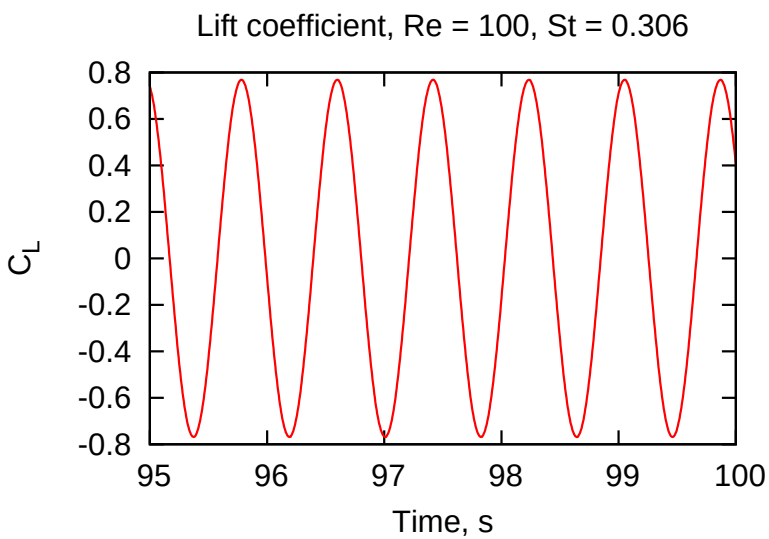
- Handling of moving obstacles in the flow domain whose size is larger than mesh resolution: covering multiple cells
- Mesh topology and connectivity does not change: **immersed STL surface**
- Presence of boundary implicitly accounted for in discretisation, with appropriate handling of boundary conditions

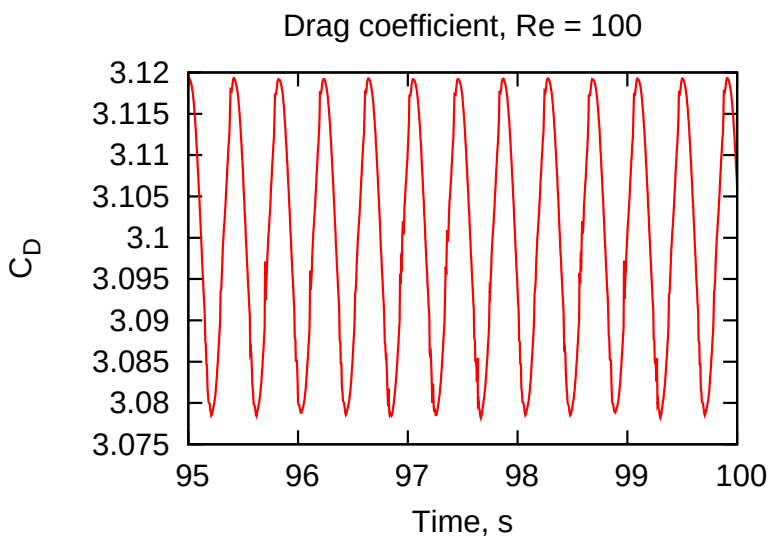


Immersed Boundary Method

Immersed Boundary Method: Vortex Shedding in a channel (Ferziger, Perić)





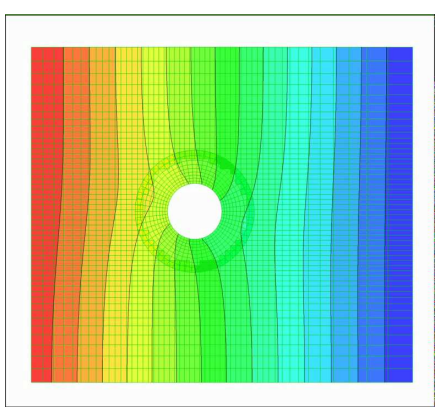


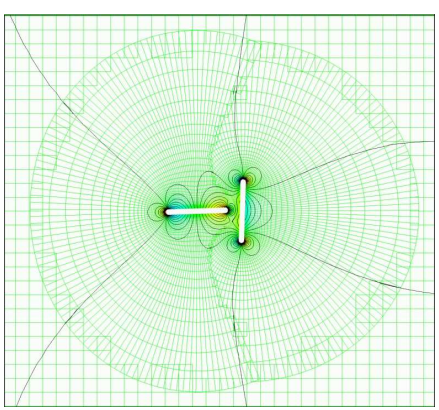
How about a moving cylinder? Željko Tuković, FSB Zagreb

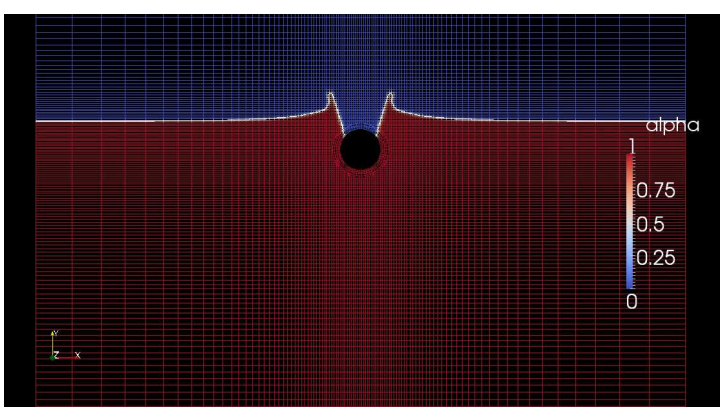
Overset Grid

`foamedOver`: Overset Grid Technology

- Work by David Boger, Penn State University using SUGGAR and DirtLib libraries developed by Ralph Noack, Penn State
- Overset Grid Technology
  - Multiple components meshed individually, with overlap
  - Hole cutting algorithm to remove excess overlap cells
  - Mesh-to-mesh interpolation with implicit updates built into patch field updates and linear solver out-of-core operations
- Body-fitted component meshes: preserving quality and near-wall resolution
- Simple mesh motion and geometrical studies (replacing individual components)
- Overset grid is physics-neutral: ideal candidate for library implementation flows







Summary

Summary

- OpenFOAM implements polyhedral mesh handling in library form
- Mesh analysis classes separated from discretisation support classes
- Built-in support for mesh motion and topological changes
- Simple handling of moving boundary problems: automatic mesh motion solver
- Topological changes support
  - Basic operations: add/modify/remove point/face/cell
  - Mesh modifier classes operate in terms of basic operations. Triggering of topological changes is a part of the class (automatic)
  - Pre-defined mesh modifiers available for standard operations
- Dynamic mesh classes encapsulate topology modifiers and (automatic) mesh motion for easy handling of classes of dynamic mesh problems
- Top-level physics solver is separated from dynamic mesh handling for inter-operability of components
- Alternative techniques are needed for extreme cases: immersed boundary method, Overset grid