Recent Developments in OpenFOAM

Henrik Rusche

Wikki GmbH, Germany

`henrik.rusche@wikki-gmbh.de`

III International Conference ”Cloud computing. Education. Research. Development.

ISP RAS, Moscow, 5.-6. December 2013

Outline

Objective

- OpenFOAM?
- Quick overview of recent developments in OpenFOAM
- Update on OpenFOAM community and activities

Topics

- OpenFOAM Executive Summary
- Complex Physics Topics and New Capabilities
  - Multiphase-Flows New from Darmstadt
  - Immersed boundary method
  - Gradient-based optimization
- OpenFOAM community & Summerschool
- Summary

OpenFOAM: Executive Overview

What is OpenFOAM?

- **OpenFOAM** is a free-to-use Open Source numerical simulation software library with extensive CFD and multi-physics capabilities
- Free-to-use means using the software without paying for license and support, including **massively parallel computers**: free 1000-CPU CFD license!
- Substantial installed user base in industry, academia and research labs
- All components implemented in library form for easy re-use
- Possibility of extension to non-traditional, complex or coupled physics: Fluid-Structure Interaction, complex heat/mass transfer, internal combustion engines, nuclear

Main Components

- Discretisation: Polyhedral Finite Volume Method, second order in space and time
- Lagrangian particle tracking, Finite Area Method (2-D FVM on curved surface)
- Massive parallelism in domain decomposition mode
- Automatic mesh motion (FEM), support for topological changes
- Physics model implementation through **equation mimicking**

Implementing Continuum Models

Equation Mimicking

- Natural language of continuum mechanics: partial differential equations
- Example: turbulence kinetic energy equation

$$\frac{\partial k}{\partial t} + \nabla\bullet(\mathbf{u}k) - \nabla\bullet[(\nu + \nu_t)\nabla k] = \nu_t \left[\frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\right]^2 - \frac{\epsilon_o}{k_o}\,k$$

- Objective: **represent differential equations in their natural language**

```
solve
(
    fvm::ddt(k)
  + fvm::div(phi, k)
  - fvm::laplacian(nu() + nut, k)
== nut*magSqr(symm(fvc::grad(U)))
  - fvm::Sp(epsilon/k, k)
);
```

- Correspondence between the implementation and the original equation is clear

Recent Development

Example of Capabilities of OpenFOAM in Complex Physics and Industrial CFD

- This is only a part of the OpenFOAM capabilities!
- Chosen for relevance and illustration
- In some cases, simplified geometry is used due to confidentiality
- Regularly, the work resulted in a new solver; in many cases, it is developed as an extension or combination of existing capabilities
- Not all of the work is done by Wikki – Work is acknowleged on the slides and at the end of this presentation

Multi-Scale Two-Phase Flow Modeling

Hybrid Interface-Resolving Two-Fluid Model

- **Model Framework**: Two-Fluid Model
- **Conceptual Approach**: Conditional Volume Averaging & Immersed Interface Concept
- **Features**:
  - derived from first principle & general closure
  - resolution-consistent capturing of multiple interfacial scales (flow regime transitions)
- **Future Work**
  - enhancement of dispersed flow modeling (capturing polydispersity)
  - algorithm and solution technique (incorporation of phase compressibility)

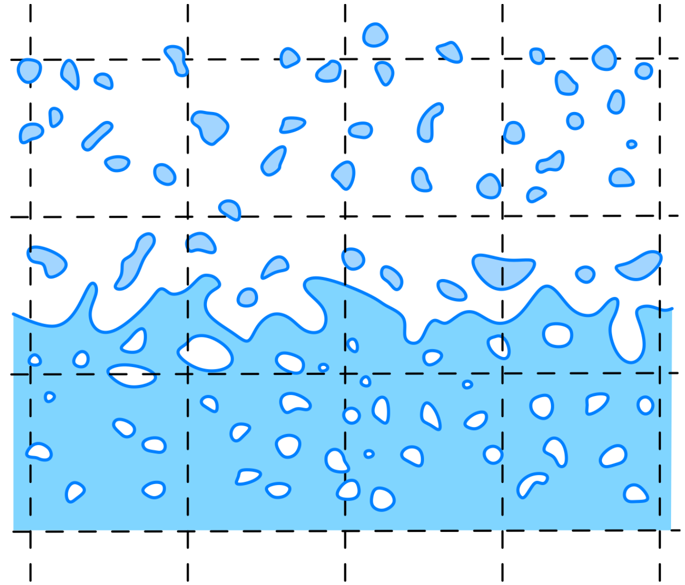


Figure 1: Mixed flow regime.

Multi-Scale Two-Phase Flow Modeling

Example: Industrial-Scale Simulation

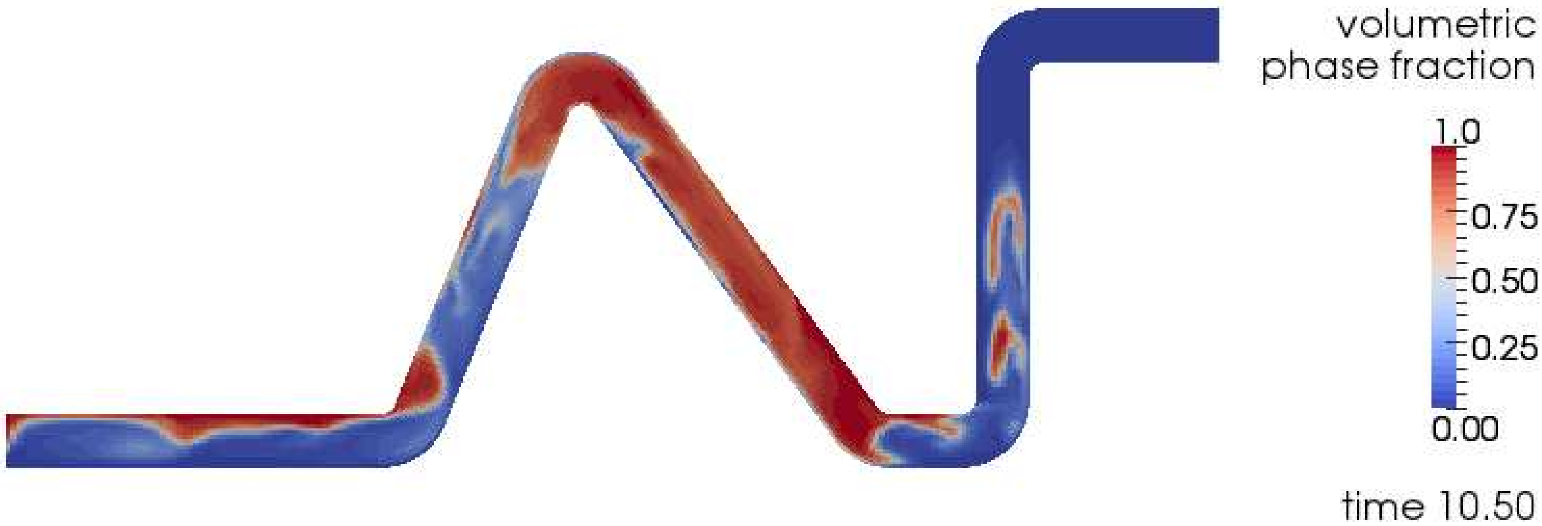


Figure 2: Flow Regime Transition in Large-Scale Pipeline

domain bounding box: height 3 m, width 9 m; pipeline length: 14.5 m

Geometrical Volume of Fluid method

Method properties

- **geometrically computed volumetric phase fluxes**
- interface represented as a set of polygons
- single layer of interface cells: a **consistent** numerical method
- **fully parallelized**

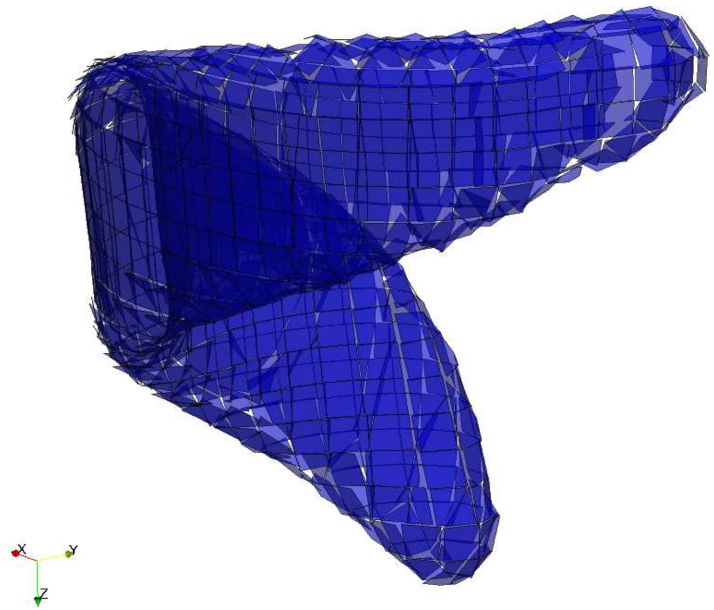


Figure 3: Interface in a shear flow.

Geometrical Volume of Fluid Method

Geometrical transport algorithm for the volume fraction:

1. reconstruct a plane in the interface cell
2. sweep a face backward with point interpolated velocities
3. intersect the swept polyhedron with the cell and the plane
4. update the volume fraction field

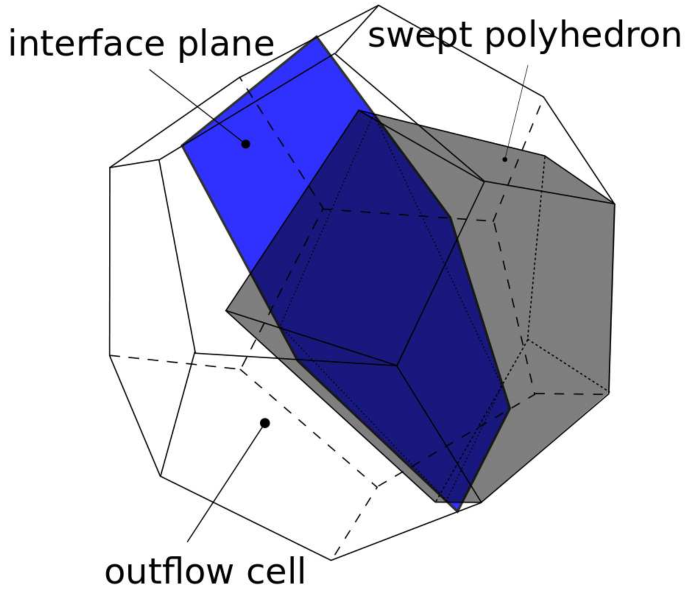


Figure 4: Geometrical flux calculation.

Geometrical Volume of Fluid method

Volume fraction update:

$$\alpha^n_{1P} = \alpha^o_{1P} - \frac{1}{V_P} \sum_f V_f$$

V_f = volumetric phase flux

Ongoing work:

- investigation of wisps : artificial interface cells (method inherent)
- flow solver coupling
- local mesh adaptivity

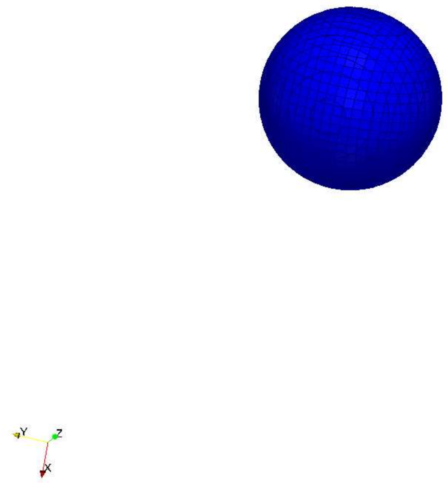


Figure 5: Interface deformation.

Direct Numerical Simulation of Surfactant Mixtures

1. **Transport of the surfactant in the bulk**

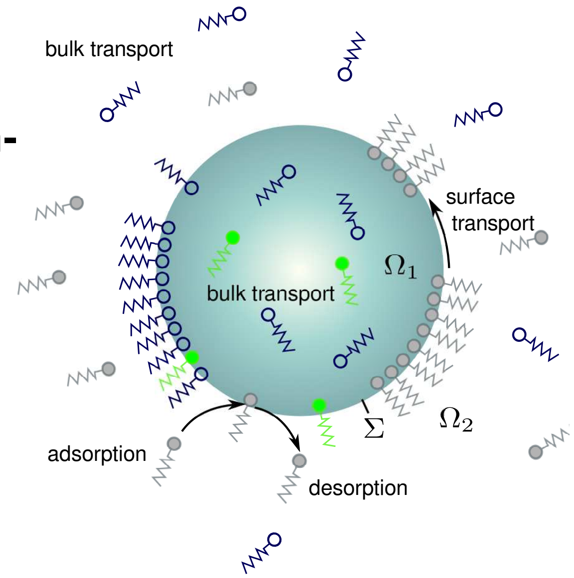
$$\partial_t \rho_i + \nabla \cdot (\rho_i \mathbf{v}) + \nabla \cdot \mathbf{j}_i = r_i$$

2. **Transport of the surfactant on the fluidic interface**

$$\partial_t^\Sigma \rho_i^\Sigma + \nabla_\Sigma \cdot \left(\rho_i^\Sigma \mathbf{v}\right) + \nabla_\Sigma \cdot \mathbf{j}_i^\Sigma = s_i^\Sigma + r_i^\Sigma$$

3. **Multiregion coupling (ad-/desorption)**

$$-\left[\!\left[\mathbf{j}_i \cdot \mathbf{n}^\Sigma\right]\!\right] = s_i^\Sigma$$



Features of the Solver

- Sharp representation of the Interface
- Block-coupled Finite Area based solution of the interfacial transport equations
- Library for sorption processes

Diffusive Flux on the Interface – Maxwell-Stefan Equations

Maxwell-Stefan equations for adsorped species:

$$-\sum_{j\neq i}^{N}\frac{\rho_j^\Sigma \mathbf{j}_i^\Sigma - \rho_i^\Sigma \mathbf{j}_j^\Sigma}{\rho^\Sigma D_{ij}^\Sigma} = \frac{\rho_i^\Sigma}{RT}\nabla\mu_i^\Sigma \qquad (1)$$

matrix representation

$$\mathbf{B}^\Sigma \mathbf{j}^\Sigma = \mathbf{d}^\Sigma \qquad (2)$$

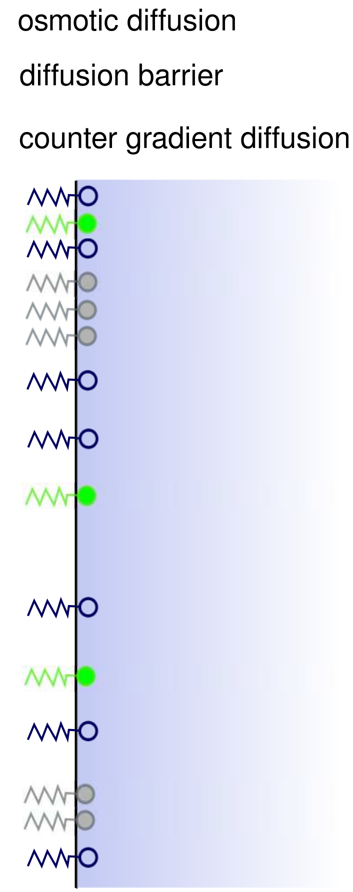
constraint: $\sum_i \mathbf{j}_i^\Sigma = 0$

$\Rightarrow$ aim is to find a representation of the system in the form

$$\mathbf{\Gamma}^\Sigma \cdot \mathbf{B}^\Sigma \mathbf{j}^\Sigma = \mathbf{\Gamma}^\Sigma \mathbf{d}^\Sigma \qquad (3)$$

where $\mathbf{\Gamma}^\Sigma$ is the group inverse $\mathbf{B}^{\Sigma,\#}$ of $\mathbf{B}^\Sigma$. $\rightarrow$ Giovangigli
Inserting into the interfacial species transport equations:

$$\partial_t^\Sigma\,\rho_i^\Sigma + \nabla_\Sigma\cdot\left(\rho_i^\Sigma \mathbf{v}\right) - \nabla_\Sigma\cdot\sum_{j=1}^{N}\Gamma_{ij}^\Sigma\nabla\rho_i^\Sigma = \mathbf{j}_i^\Sigma \qquad (4)$$



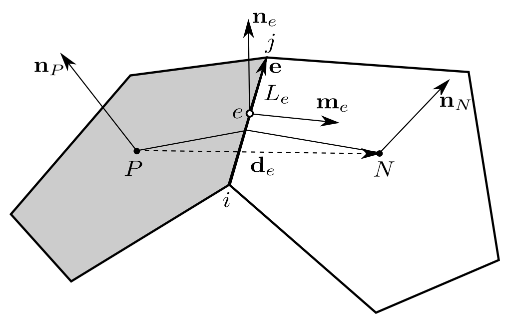
$\Rightarrow$ Strong coupling effects among the species!
$\Rightarrow$ Needs to be accounted during numerical solution

Spatial Discretization

Finite-Area Method (on a surface)

arbitrary interfacial transport equation

$$\partial_t^\Sigma \phi^\Sigma + \nabla_\Sigma \cdot \left(\mathbf{v}\phi^\Sigma\right) = \nabla_\Sigma \cdot \left(\Gamma^\Sigma \nabla_\Sigma \phi^\Sigma\right) + s_\phi^\Sigma \qquad (5)$$



equation discretisation:

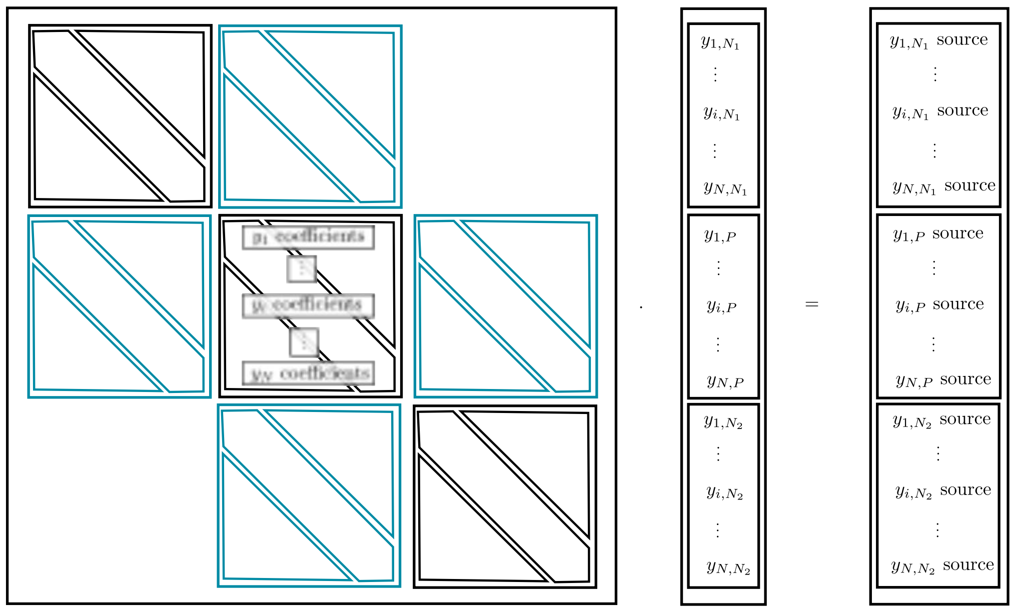
$$\frac{d\left(\phi_P^\Sigma S_P\right)}{dt} + \sum_e \mathbf{m}_e \cdot (\mathbf{u}_t)_e \, \phi_e^\Sigma L_e = \sum_e \Gamma_e^\Sigma \mathbf{m}_e \cdot \left(\nabla_S \phi^\Sigma\right)_e L_e + \left(s_\phi^\Sigma(\kappa)\right)_P S_P \qquad (6)$$

Block Coupled Solution

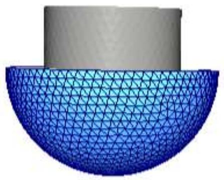
Arbitrary number of equations can be coupled. $\mathbf{a}_P$ and $\mathbf{a}_N$ may be $n \times n$ tensors The discretised species equations for each cell are put into the form

$$\mathbf{a}_P \mathbf{y}_{i,P}^{new} + \sum_N \mathbf{a}_N \mathbf{y}_{i,N}^{new} = \mathbf{R}_P \tag{7}$$

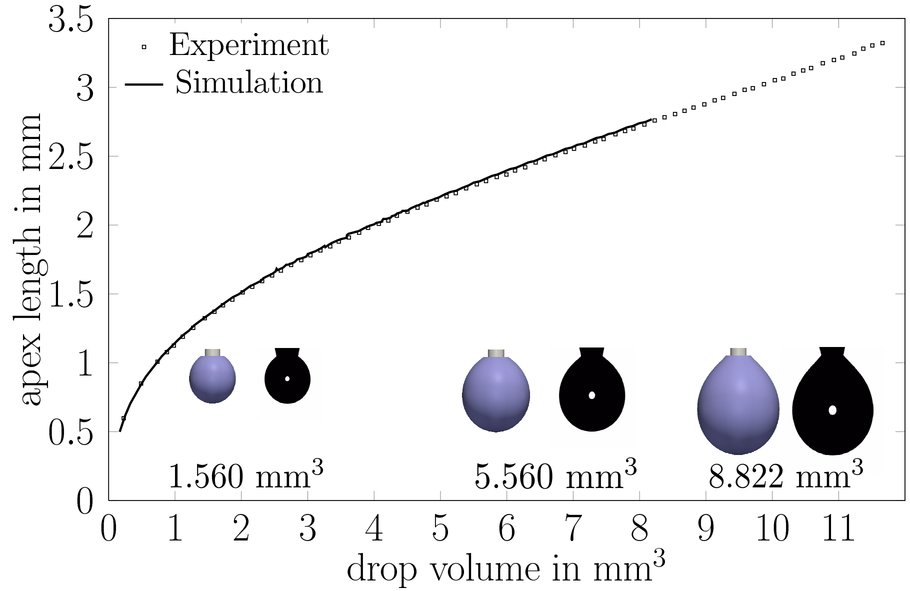
$\rightarrow$ Solution: block-stuctured GMRES with incomplete Cholesky preconditioner



Examples

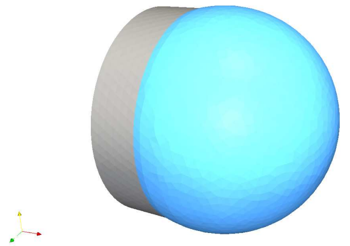


σ	0.07 N/m
ν	10^{-6}m^2/s
d_i	0.45 mm
d_i	0.7 mm
$\dot{V}$	10 mm^3/s

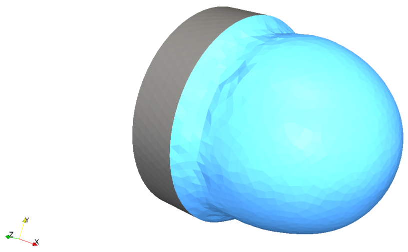


Examples

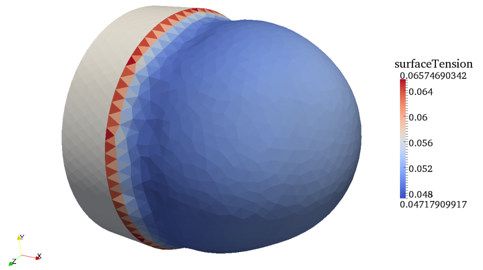
Formation without Surfactant

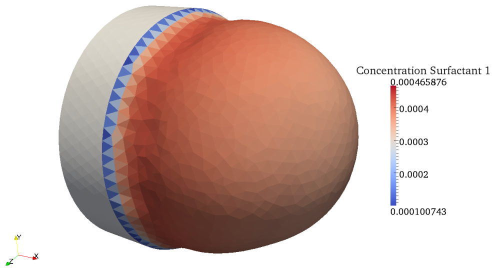


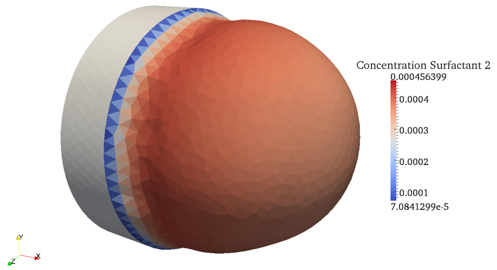
Formation with Surfactant



Analysis of Surfactant Influence on Droplet Formation



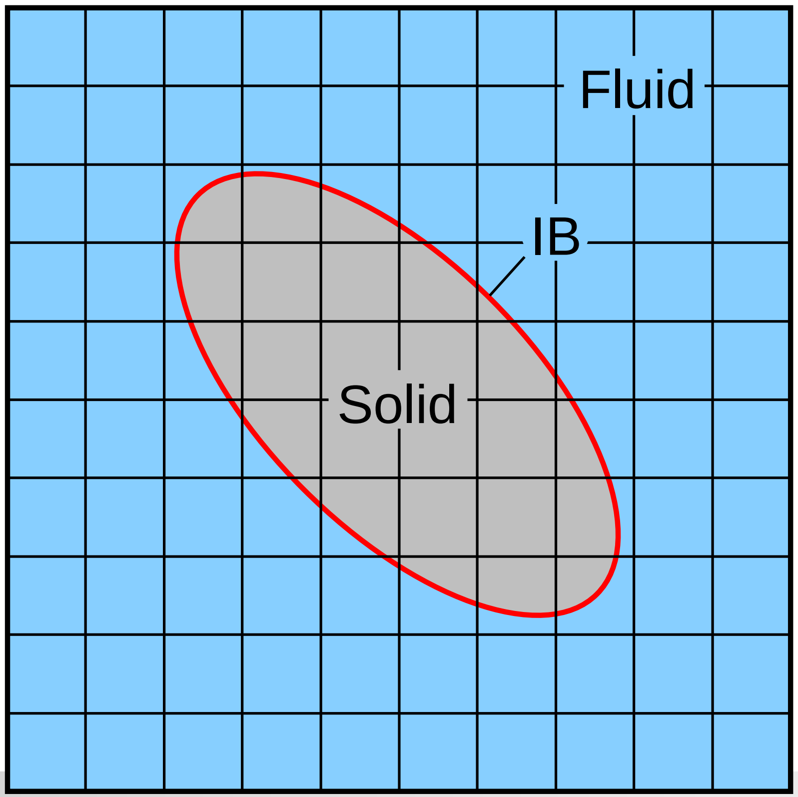




Immersed Boundary Methods (IBM)

Immersed Boundary Method: Non-Conformal Boundary Surfaces

- Simulation of the flow around immersed boundary is carried out on a grid (usually Cartesian) which does not conform to the boundary shape
- Immersed boundary (IB) is represented by surface grid
- Imposition of boundary conditions at IB requires modification of governing equations in cells which interact with the immersed boundary



IBM: Pros and Cons

Advantages of IBM Over Body-Fitter Mesh Methods

- Substantially simplified grid generation for complex geometry
- Inclusion of body motion is relatively simple due to the use of stationary, non-deforming background grids

Disadvantages of IBM Over Body-Fitter Mesh Methods

- Imposition of boundary conditions at IB is not straightforward: special techniques are developed and implemented
- Problem with grid resolution control in boundary layers: effective near-wall mesh size is approx 50% larger than in equivalent body-fitted mesh
- Limited to low and moderate Reynolds number flows: this is resolved using the Immersed Boundary wall function implementation

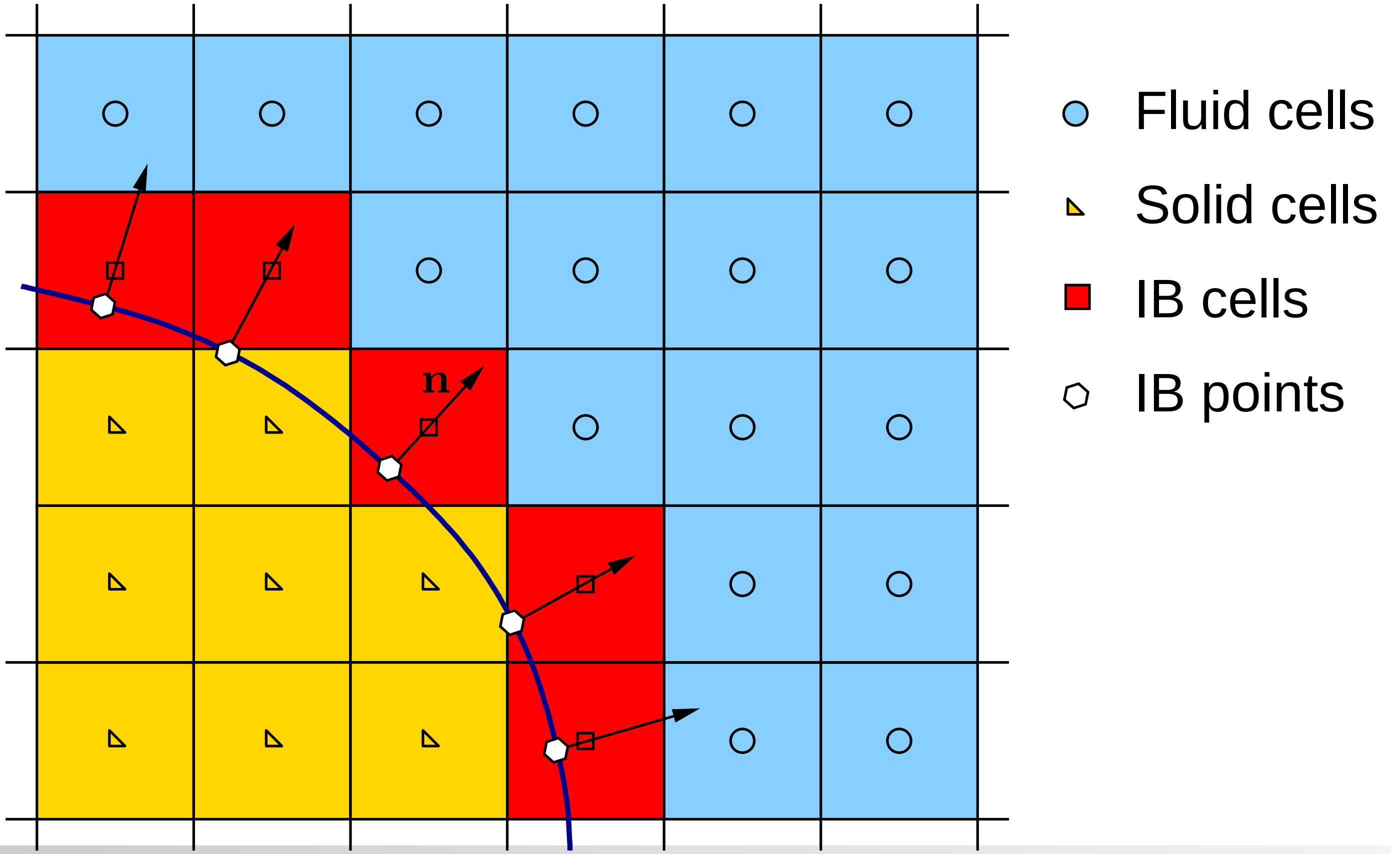
Wish List

1. IBM solution MUST mimic the equivalent body-fitted mesh solution
2. Minimal interaction in top-level code: flow solvers and auxiliary models to be used without coding changes
3. Remove limitations on background mesh structure: must work with polyhedra
4. Automate mesh refinement under the IB surface

IBM In OpenFOAM: Selected Approach

Implementation of IBM In OpenFOAM

- Discrete forcing approach with direct imposition of boundary conditions
- Basic principle: Value of dependent variable in the IB cell centres is calculated by interpolation using neighbouring cells values and boundary condition at the corresponding IB point



IBM: User Perspective

Including `immersedBoundaryPolyPatch` into boundary mesh of a `polyMesh` by modifying `constant/polyMesh/boundary` dictionary

```
6
(
    ibCylinder // constant/triSurface/ibCylinder.ftr
    {
        type                immersedBoundary;
        nFaces              0;
        startFace           3650;

        internalFlow        no;
    }
    in
    {
        type                patch;
        nFaces              25;
        startFace           3650;
    }

    ...
)
```

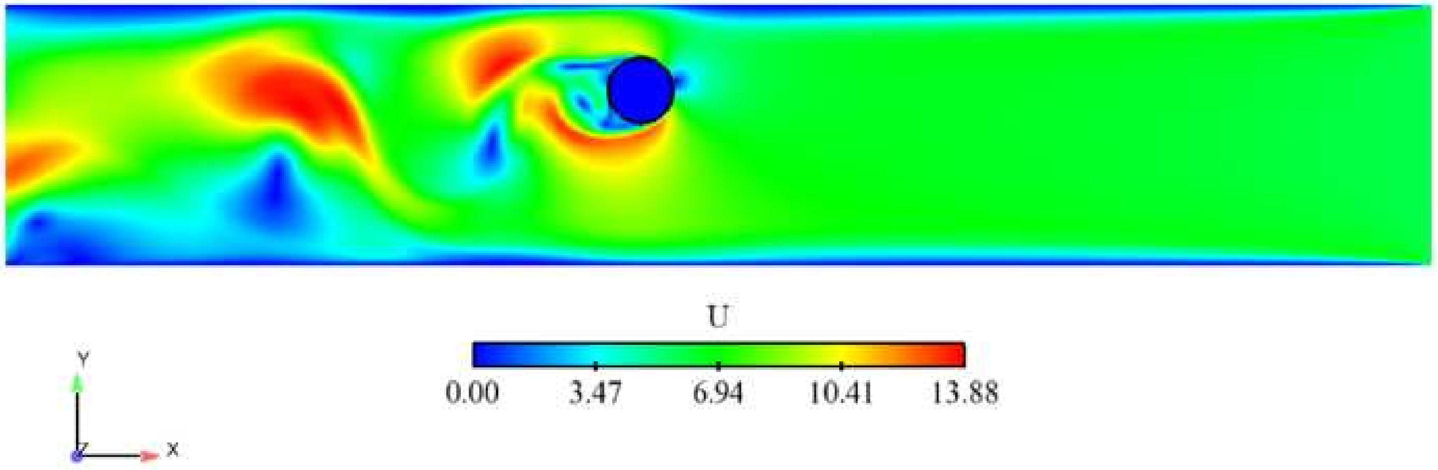
IBM: User Perspective

Including `immersedBoundaryFvPatchField` into boundary of a `volVectorField`

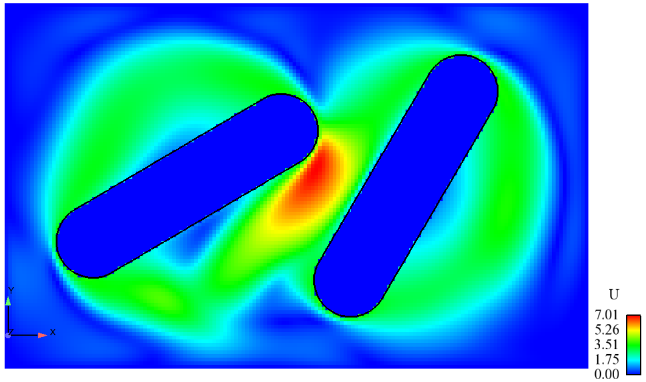
```
boundaryField
{
    ibCylinder
    {
        type immersedBoundary;
        refValue uniform (0 0 0);
        refGradient  uniform (0 0 0);
        fixesValue yes;

        setDeadCellValue   yes;
        deadCellValue      (0 0 0);

        value uniform (0 0 0);
    }
    movingWall
    {
        type parabolicVelocity;
        maxValue 0.375;
        n (1 0 0);
        y (1 0 0);
        value uniform (0.375 0 0);
    }
    ...
}
```

IBM: Examples

- Laminar flow around a 2-D moving circular cylinder in a channel



- Laminar flow around two counter-rotating elements in a cavity



Introduction Optimisation

WIKKI

- Find

$$min(J(p))$$

  where J is the objective functional and p are free parameter
- J is calculated by executing a computational workflow (pre-processing, solver, post-processing) which takes into account the parameters p
- Classification:
  - Single vs. Multi-Objective problem (J is a vector)
  - Unconstraint vs. constraint problem (additional restrictions on J and/or p)
  - Parameter vs. Shape or Topology optimisation (geometry is changed)
  - Gradient vs. non-gradient based algorithms (uses $\frac{\partial J}{\partial p}$)
- Optimization as a discipline is very rich in methods, approaches and algorithms
- Many external tools exists (Dakota, modeFrontier, LMS Optimus)

Non-gradient Based optimization

- Non-gradient based optimization is based on the evaluation of gradients of objective functional in which OpenFOAM is responsible for providing the current state
- Usually several states are saved in order to deduce the behavior of the function being minimized
- This knowledge is used to retain and discard the states in an effort to find the optimal point
- Typical problem with non-gradient based algorithms is so called "curse of dimensionality" - too many parameters require excessive number of non-linear function evaluations thus making the algorithms very expensive
- In shape optimization effective geometrical parameterization is crucial in controlling the cost of the optimization algorithm

Gradient Based optimization

- Gradient based optimization algorithms rely on the analytical and semi-analytical tools of gradient evaluation in order to guide optimization algorithms
- Gradients can be obtained in several ways: Numerically, Automatic differentiation, Analytically
- If the original PDEs are differentiated with respect to the optimization parameter, the resulting equations form a continuous approach to gradient evaluation
- Two approaches in analytical differentiation of PDEs and gradient evaluations:
  - Continuous tangent equations
  - Continuous adjoint equations
- Continuous approach is ideal for OpenFOAM as software framework allows rapid implementation of PDEs

Continuous Adjoint

WIKKI

- Construct the Lagrange functional from the weak form of Navier-Stokes equations and the cost function:

$$L(x,u,p) = J(x,u,p) + \int_t \int_\Omega F(x,u,p) U^*(x,u) d\Omega dt \tag{8}$$

- Continuous adjoint equations are obtained by using the duality principles on continuous tangent equation
- In the case of the incompressible laminar Navier-Stokes equations, continuous adjoint equations are as follows:

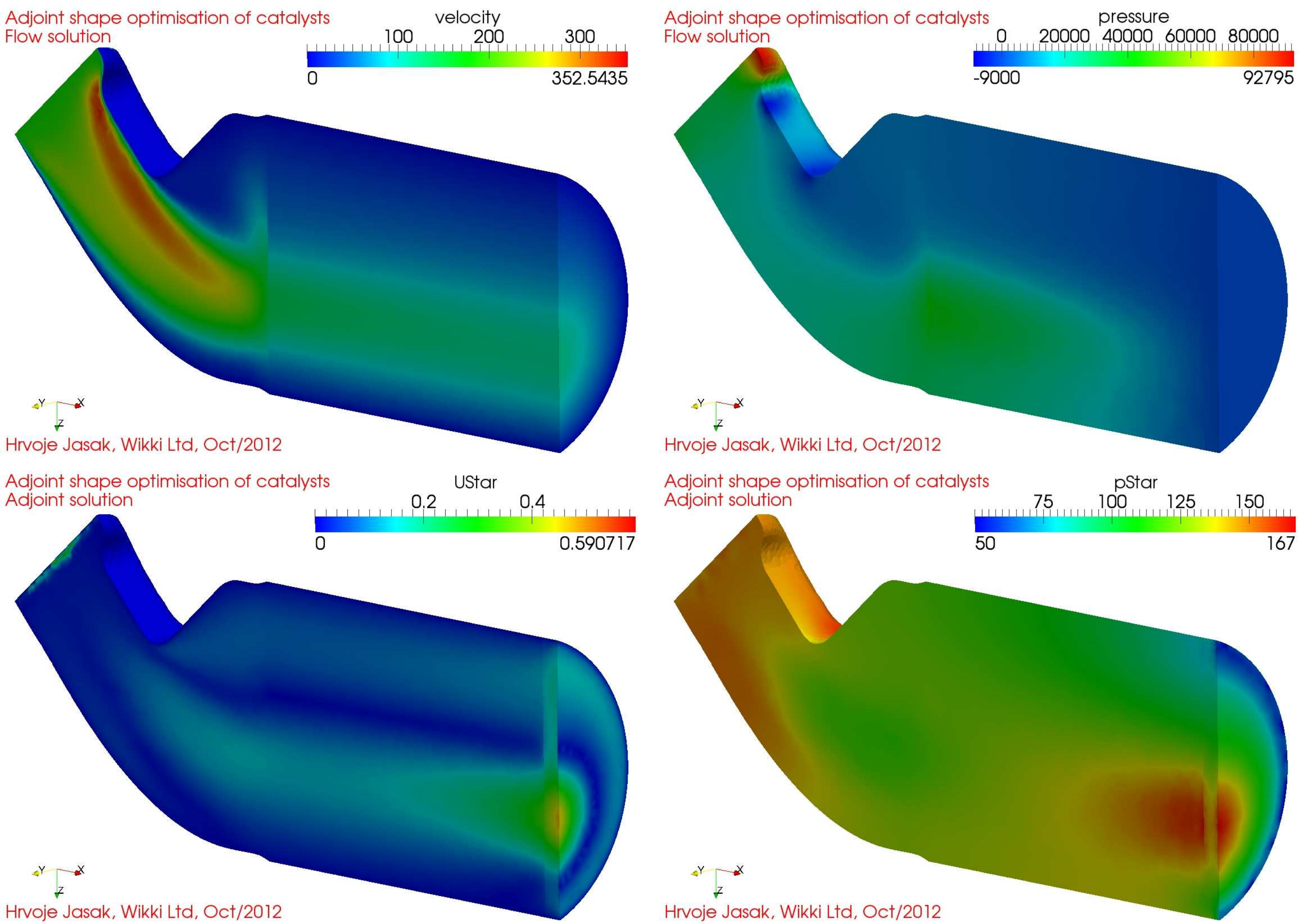
$$\begin{aligned} -\nabla \bullet \mathbf{u}^* &= 0 \\ -\mathbf{u}\nabla\bullet\mathbf{u}^* - \nabla(\mathbf{u}^*)\bullet\mathbf{u} - \nu\nabla\bullet\nabla\mathbf{u}^* &= -\nabla p^* \end{aligned}$$

- This system of equations describes the propagation of the derivatives from outputs to inputs in the system - It finds the source of a specific anomaly
- It does NOT model physical quantities, but the sensitivity of a property to these quantities
- Efficiently computes sensitivity of a model with a large number of sensitivity parameters and few objective/cost functions

Catalyst System

Optimisation of a Catalyst (Inlet Part) with Porous Media: Body-Fitted Mesh

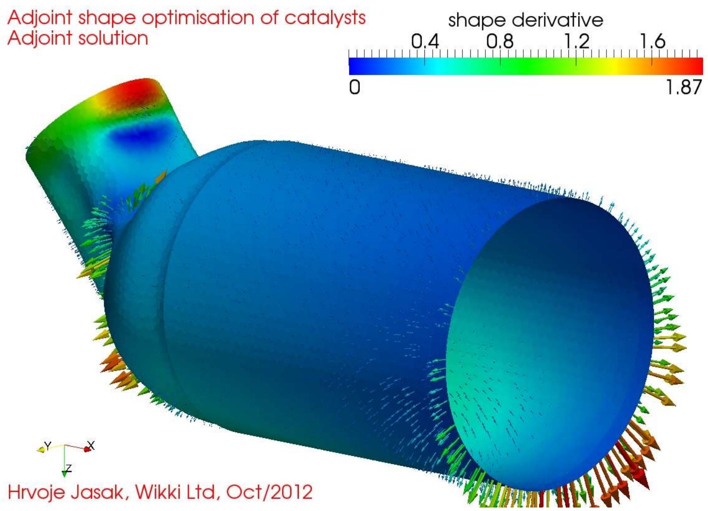
- Forward and adjoint solution: pressure and velocity

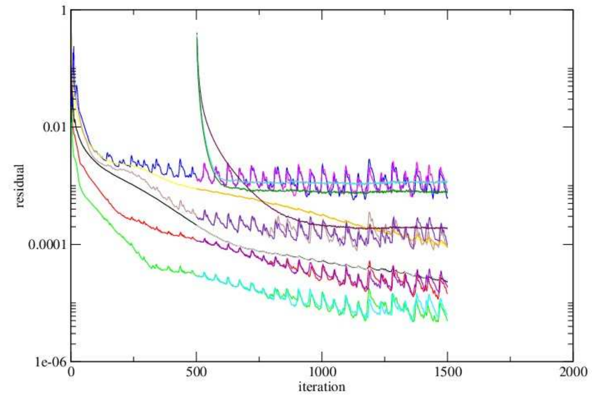


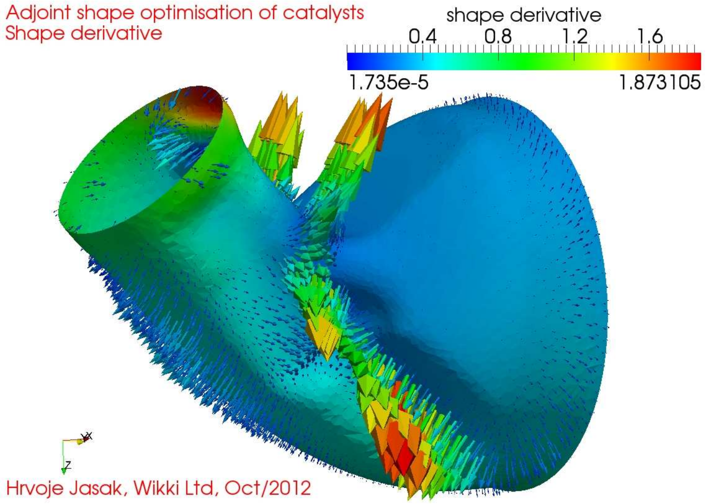
Catalyst System

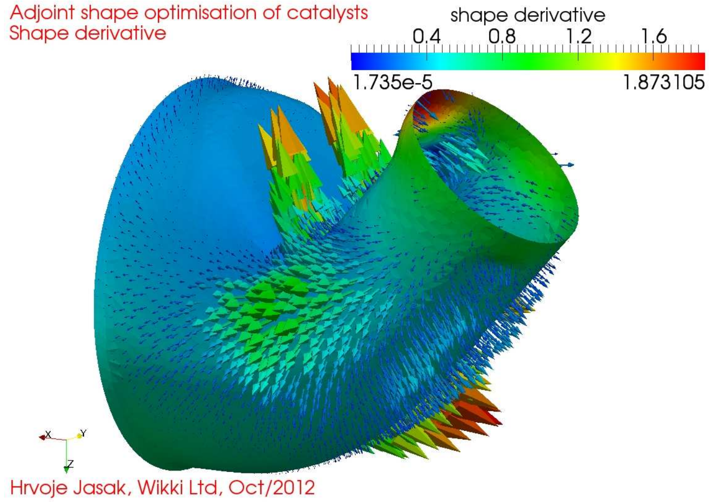
Optimisation of a Catalyst (Inlet Part) with Porous Media: Body-Fitted Mesh

- Shape derivative and convergence history





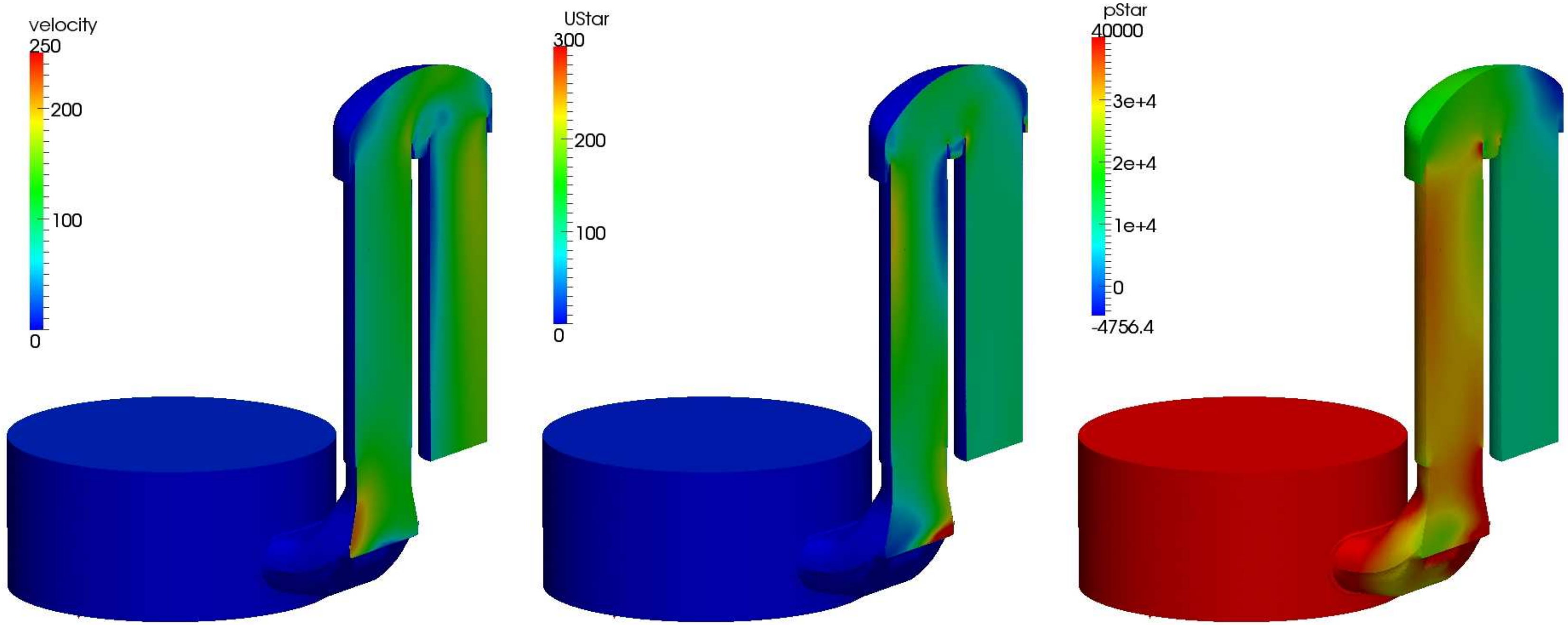




Piping System

Optimisation of Connecting Piping

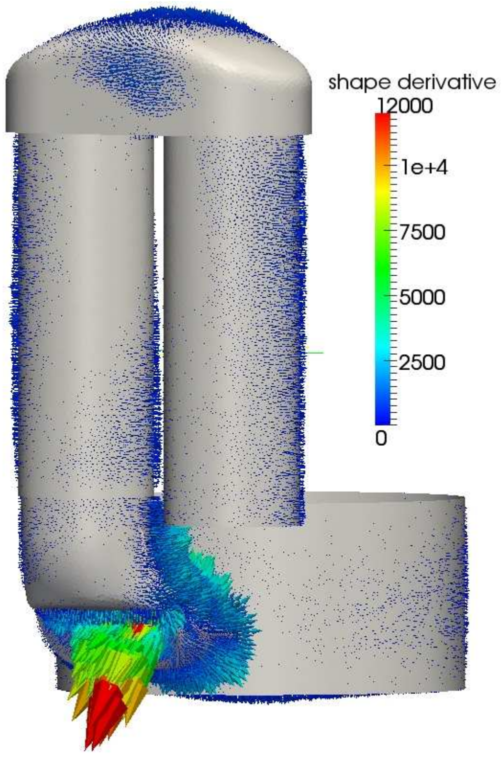
- Forward and adjoint solution: pressure and velocity

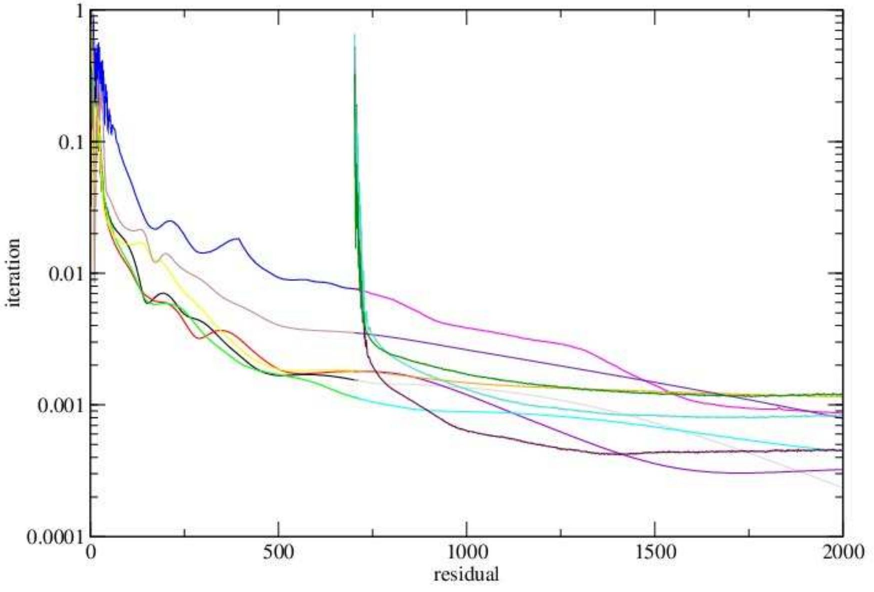


Piping System

Optimisation of Connecting Piping

- Forward and adjoint solution: shape derivative and convergence history

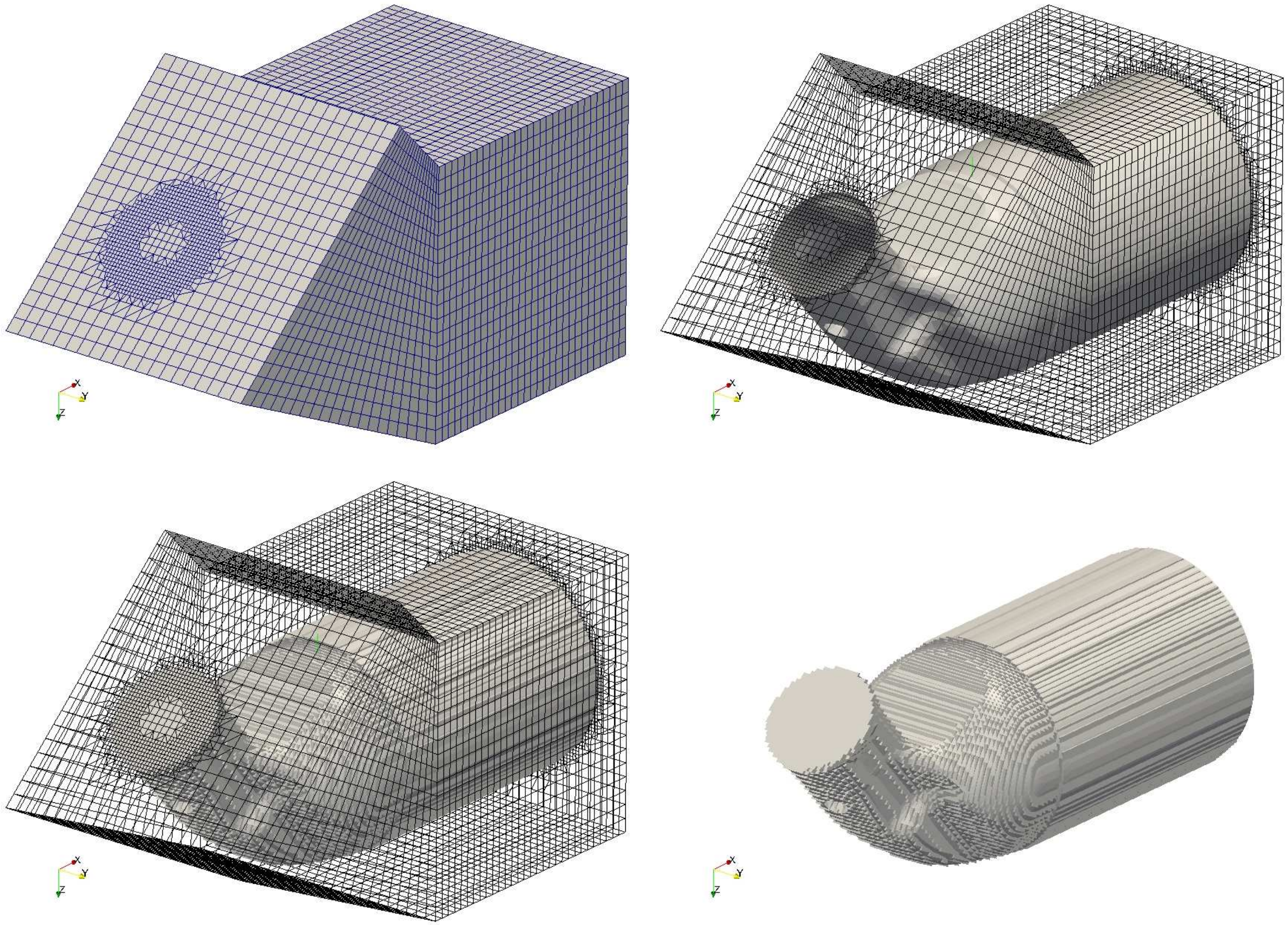




Immersed Boundary Catalyst System

Optimisation of a Catalyst (Inlet Part) with Porous Media: Immersed Boundary

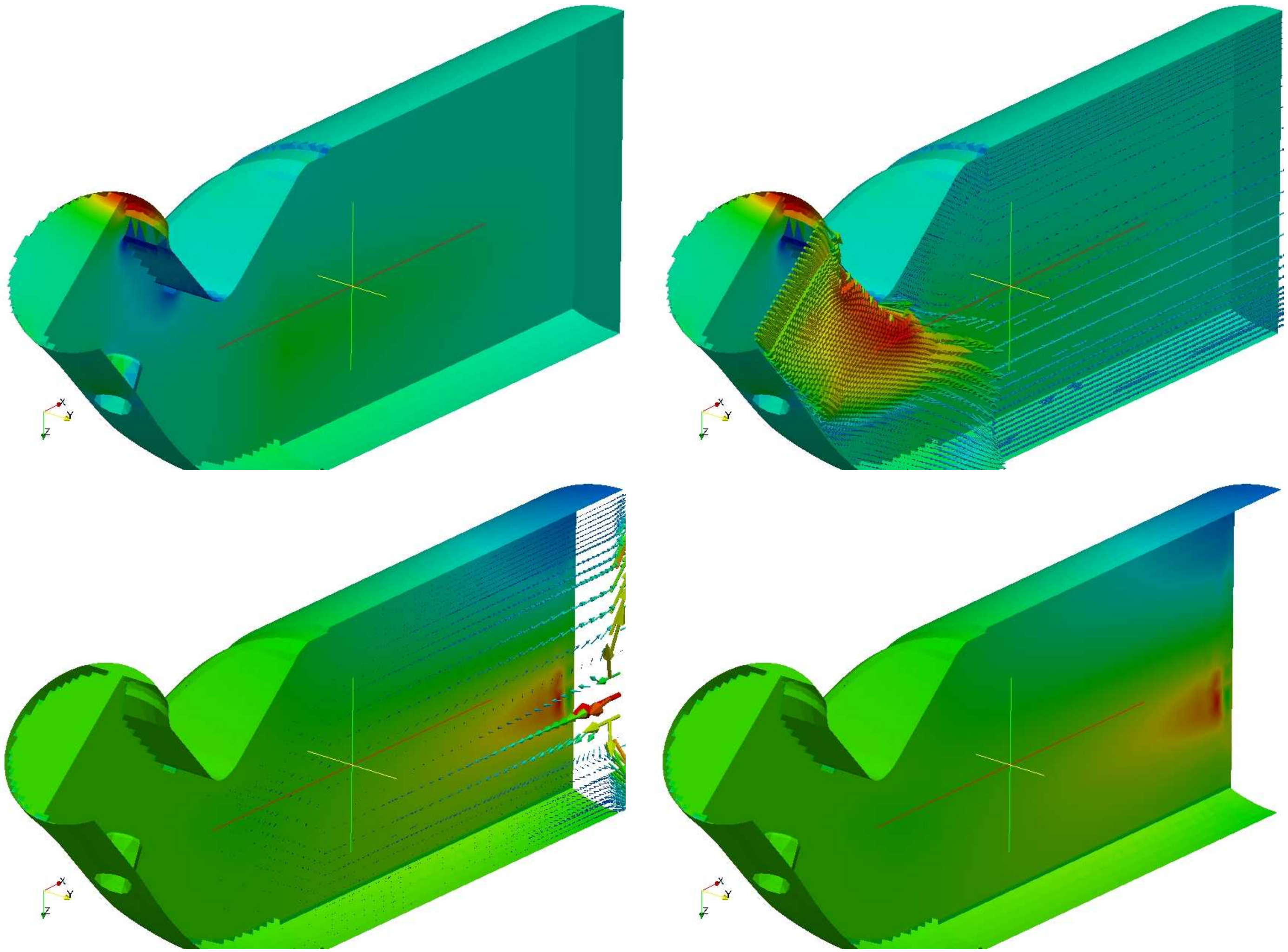
- Mesh setup, immersed boundary and live subset



Immersed Boundary Catalyst System

Optimisation of a Catalyst (Inlet Part) with Porous Media: Immersed Boundary

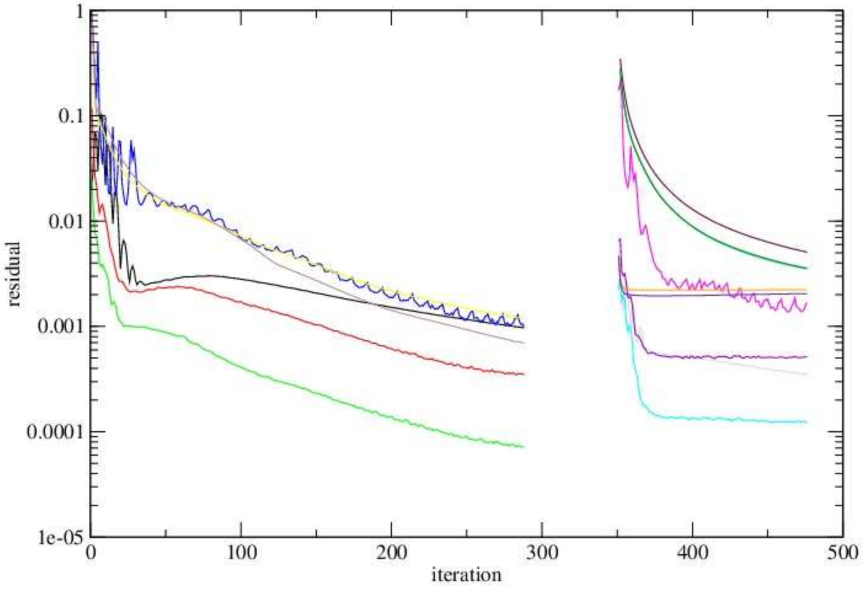
- Forward and adjoint solution: pressure and velocity



Immersed Boundary Catalyst System

Optimisation of a Catalyst (Inlet Part) with Porous Media: Immersed Boundary

- Convergence history

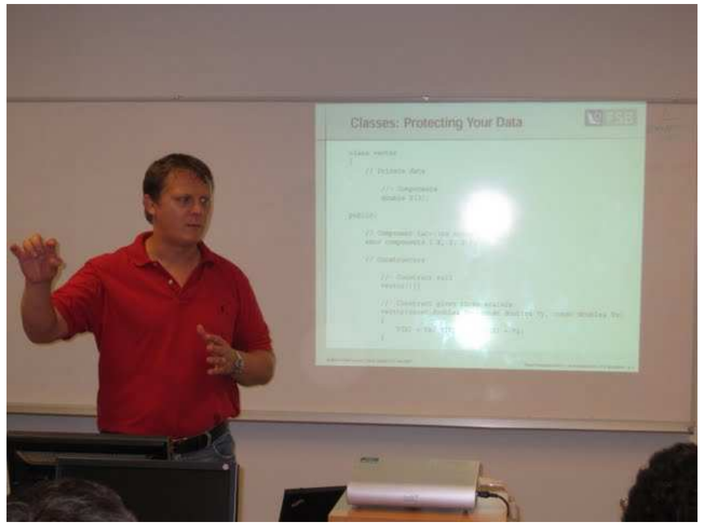


OpenFOAM Summerschool

- expand the physical modelling knowledge, numerics and programming skills of attendees
- providing extended tuition (14 days) for a small and selected group
- students and researchers from academia and industry
- Organisers: Prof. Hrvoje Jasak & Prof. Zdravko Terze

Activities

- Each student works on his project
- direct supervision and one-to-one work
- Lectures on chosen topics
- Having fun together



General Information

- When? → Once a year, the first two weeks in September
- Where? → At the University of Zagreb, Croatia
- How to apply?
  → Application with a description of your project, current problems and goals
- Who can apply?
  → All students on MSc and PhD university courses
  → Young researchers in commercial companies
  OpenFOAM experience is pre-requisite. It is **NOT** an introductory course
- More information? Checkout the web-site. Ask the Alumni!

Summary

WIKKI

- OpenFOAM is an object-oriented approach to solve continuum mechanics problems
- It focuses on fluid dynamics and complex coupled physics
- It is still gaining momentum around the world
- Leading research centres and major industrial companies adopted OpenFOAM
- Healthy commercial support infrastructure offering training, support, (core) development, graphical user interfaces, acceleration
- This ecosystem provides all ingredients for a successful future: Teaching (academic & commercial), Development (application & core level), Funding (industrial & public)
- Open Source model dramatically changes the industrial CFD landscape: new business model
- Come and find out more at the **Eighth OpenFOAM Workshop**: Organised by Seoul National University in Jeju, South Korea 11-14 June 2013 `http://www.openfoamworkshop.org`

Acknowledgements

- Carsten Thorenz; Bundesanstalt für Wasserbau, Karlsruhe, Germany
- Kathrin Kissling; Holger Marschall; Tomislav Maric; Centre for Smart Interfaces, Darmstadt, Germany
- Zeljko Tukovic; FSB Zagreb, Croatia
- Participants and tutors of OpenFOAM Summerschool 2012