# Review of Recent and Ongoing Developments of the OpenFOAM library

**Henrik Rusche**

`henrik.rusche@wikki-gmbh.de`

**Wikki GmbH, Germany**

**Extended Cloud Services Workshop, Moscow, 3 June 2011**

# Agenda

- Complex Physics Topics and New Capabilities
- Industrial CFD Topics
- OpenFOAM community
- Summary

# OpenFOAM: Executive Overview

What is OpenFOAM?

- **OpenFOAM** is a free-to-use Open Source numerical simulation software 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!
- Software under active development, capabilites mirror those of commercial CFD
- Substantial installed user base in industry, academia and research labs
- 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
- All components implemented in library form for easy re-use
- 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

# Examples of Simulation

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 of the range of capabilities rather than exhaustive illustration of range of capabilities
- 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

Description of Simulation and Setup

- Physics and numerical method setup
- Standard or customised solver; details of mesh resolution and customisation

# Viscoelastic Flow Model

MSc Thesis: Jovani Favero, Universidade Federal de Rio Grande del Sul, Brazil

- Viscoelastic flow model:

$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial(\rho\, \mathbf{u})}{\partial t} + \nabla \bullet (\rho\, \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \bullet \boldsymbol{\tau}_\mathbf{s} + \nabla \bullet \boldsymbol{\tau}_\mathbf{p}$$

  where $\boldsymbol{\tau}_\mathbf{s} = 2\eta_s \mathbf{D}$ is the solvent stress contribution and $\boldsymbol{\tau}_\mathbf{p}$ is the **polymeric part of the stress**, non-Newtonian in nature

- Depending on the model, $\boldsymbol{\tau}_\mathbf{p}$ is solved for: saddle-point problem
- Models introduce “upper”, “lower” or Gordon-Schowalter derivatives, but we shall consider a general form: standard transport equation in relaxation form

$$\frac{\partial \boldsymbol{\tau}_\mathbf{p}}{\partial t} + \nabla \bullet (\mathbf{u}\boldsymbol{\tau}_\mathbf{p}) = \frac{\boldsymbol{\tau}^* - \boldsymbol{\tau}_\mathbf{p}}{\delta}$$

  where $\delta$ is the relaxation time-scale

- Problem: $\boldsymbol{\tau}_\mathbf{p}$ dominates the behaviour and is explicit in the momentum equation

# Viscoelastic Flow Model

WIKKI

Model Implementation Recipe

- Recognise $\boldsymbol{\tau}^*$ as the equilibrium stress value: make it implicit!

$$\nabla\bullet\boldsymbol{\tau}^* = \nabla\bullet\left[\boldsymbol{\kappa}\bullet\frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right]$$

- Calculate implicit viscoelastic viscosity:

$$\boldsymbol{\kappa} = \boldsymbol{\tau}^*\bullet\left[\frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\right]^{-1}$$

- Split complete stress into implicit and explicit component

$$\begin{aligned}\nabla\bullet\boldsymbol{\tau}_{\mathbf{p}} &= \nabla\bullet\boldsymbol{\tau}^* + \nabla\bullet\boldsymbol{\tau}_{\mathbf{corr}} \\ &= \nabla\bullet(\boldsymbol{\kappa}\bullet\nabla\mathbf{u}) + \nabla\bullet\boldsymbol{\tau}_{\mathbf{corr}}\end{aligned}$$

# Viscoelastic Flow Model

Implemented Viscoelastic Models

- **Kinetic Theory Models**: Maxwell linear; UCM and Oldroyd-B; White-Metzner; Larson; Cross; Carreau-Yasuda; Giesekus; FENE-P; FENE-CR
- **Network Theory of Concentrated Solutions and Melts Models**: Phan-Thien-Tanner linear (LPTT); Phan-Thien-Tanner exponential (EPTT); Feta-PTT
- **Reptation Theory / Tube Models**: Pom-Pom model; Double-equation eXtended Pom-Pom (DXPP); Single-equation eXtended Pom-Pom (SXPP); Double Convected Pom-Pom (DCPP)
- **Multi-Mode Form**: The value of $\boldsymbol{\tau}_{\mathbf{p}}$ is obtained by the sum of the K modes

$$\boldsymbol{\tau}_{\mathbf{p}} = \sum_{K=1}^{n} \boldsymbol{\tau}_{\mathbf{p}K}$$

Flow Solvers Implemented by Jovani Favero: **Example Simulation**

- Single-phase non-Newtonian solver based on transient SIMPLE
- Multi-phase free surface VOF solver: viscoelastic in each phase
- Support for topological changes: syringe ejection

# Multi-Phase Compressibility

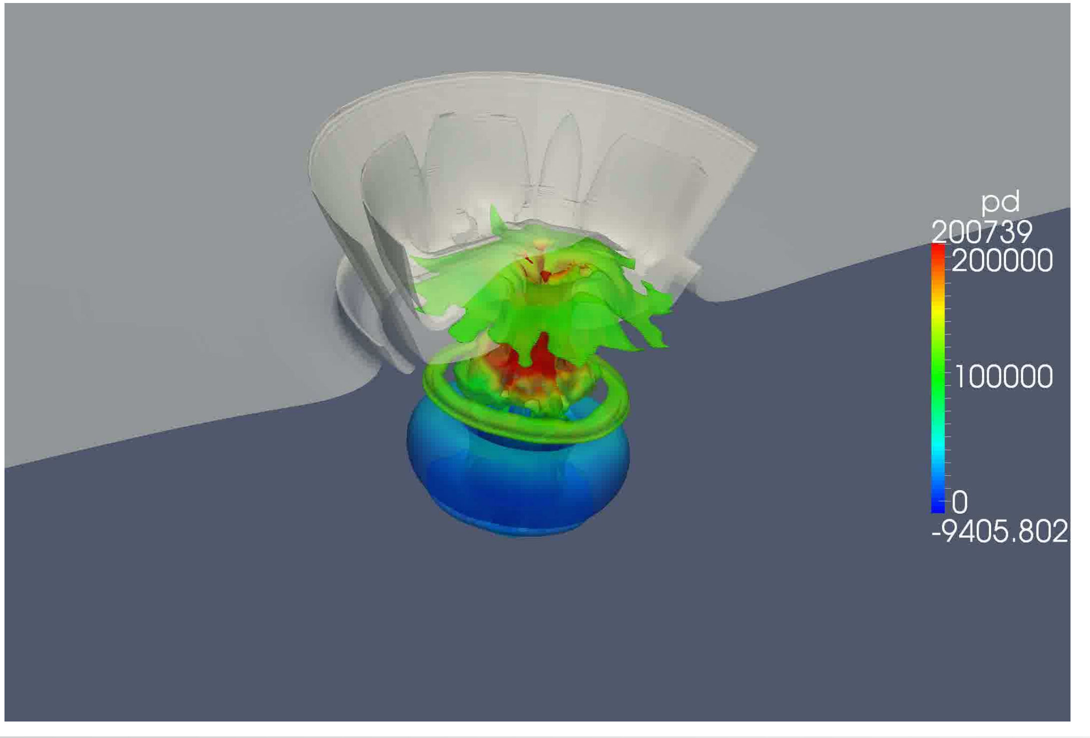
Simulation of Under-Water Explosions

- This is ongoing collaboration with Johns Hopkins APL, Penn State University and Wikki: working hard for almost a year
- Dominating effects of compressibility in air and water: massive change in density, with propagating pressure waves
- Pressure ranges from 500 bar to 20 Pa
- Stiff numerics: collapse of over-expanded bubble due to combined compressibility of both phases are the basis of the phenomenon
- Test cases: Rayleigh-Plesset oscillation, undex under a plate, explosion
- Eric Paterson, Scott Miller, David Boger, Penn State; Ashish Nedungadi, JHU-APL

# Multi-Phase Compressibility

First Simulations of Under-Water Explosions: Eric Paterson, Penn State

- Bubble of high initial pressure expands after explosion
- Initial pressure pulse is very fast - with little effect
- Bubble collapse creates re-entrant jer which pierces the free surface
- Stability problems resolved in segregated solver
- Next phase: **block-coupled** $p - \alpha$ **solution algorithm**



# Flash-Boiling Simulations

Flash-Boiling Flows: **Shiva Gopalakrishnan, David P. Schmidt, UMass Amherst**

- The fundamental difference between flash boiling and cavitation is that the process has a higher saturation pressure and temperature: higher density
- Enthalpy required for phase change is provided by inter$-$phase heat transfer
- **Jakob number**: ratio of sensible heat available to amount of energy required for phase change

$$Ja = \frac{\rho_l c_p \Delta T}{\rho_v h_{fg}}$$

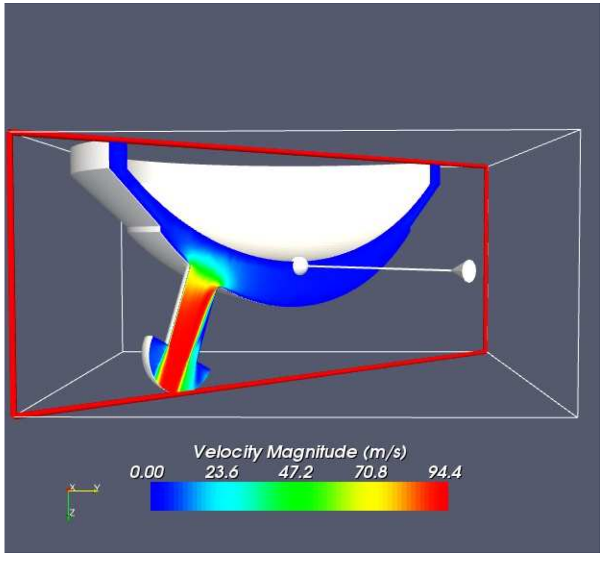
- Equilibrium models are successful for cavitation since Ja is large and timescale of heat transfer is small. Flash boiling represents a finite rate heat transfer process: **Homogeneous Relaxation Model (HRM)**

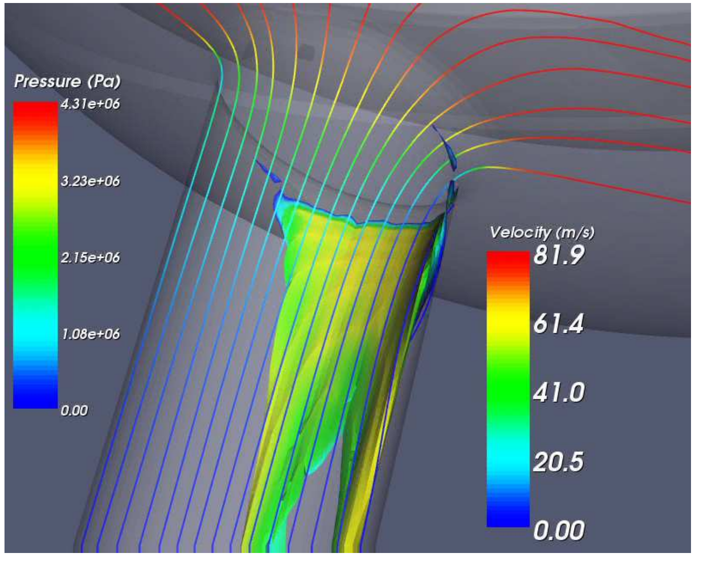
$$\frac{Dx}{Dt} = \frac{\bar{x} - x}{\Theta}; \qquad \Theta = \Theta_0 \epsilon^{-0.54} \phi^{1.76}$$

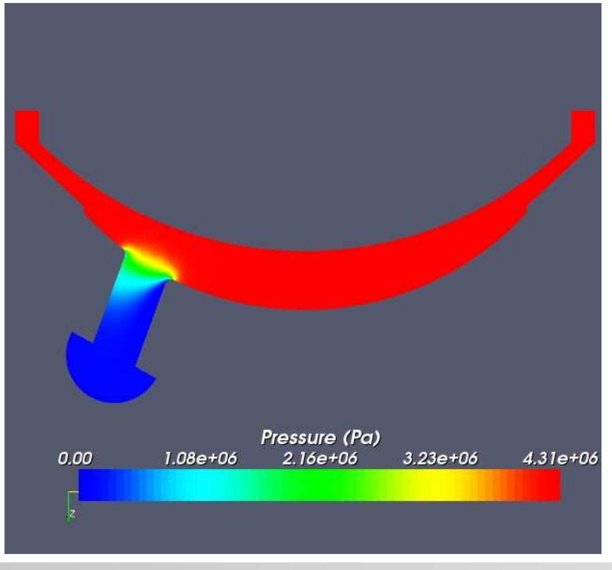
  $x$ is the quality (mass fraction), relaxing to the equilibrium $\bar{x}$ over a time scale $\Theta$
- The timescale $\Theta$ is obtained from empirical relationship: Downar–Zapolski [1996]. $\epsilon$ is the void fraction and $\phi$ is the non$-$dimensional pressure.

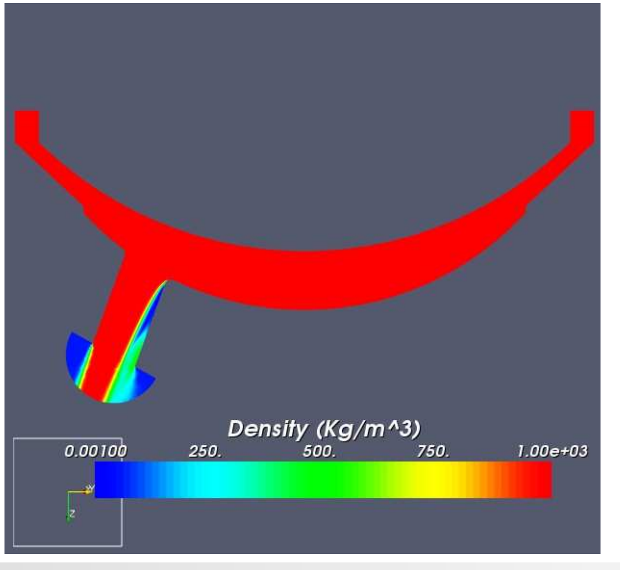
# Flash-Boiling Simulations

Asymmetric Fuel Injector Nozzle-Design from Bosch GmbH.



# Block Matrix in Use: Multi-Phase VOF

Multi-Phase Volume-of-Fluid Solver

- System of equations contains multiple VOF equations and global continuity handled by a pressure equation in standard form
- The phase for which VOF is solved first dominates the other phases: this is not acceptable; flipping the order of solution moves the problem around
- Aim: **Coupled pressure based approach to achieve physical behaviour**
- Solution strategy: solve $\alpha_i$ transport equations, volumetric continuity and closure equation in a strongly coupled manner, making the coupling terms implicit!
- To make this run, we shall use the block matrix and block solver (Jasak and Clifford, 2009)
- Credit goes to **Kathrin Kissling and Julia Springer, NUMAP-FOAM 2009**

# Block Matrix in Use: Multi-Phase VOF

Multi-Phase Volume-of-Fluid Solver: Equation Set and Coupling

- Volume fraction transport equation, with separated pressure-driven flux terms

$$\left[\frac{\partial\,[\alpha_i]}{\partial t}\right] + \left[\nabla\bullet\left(\left(\left(\frac{A_H}{A_D}\right)_f \bullet \mathbf{s}_f + \frac{1}{A_D}(\sigma\kappa)_f \nabla_f^{\perp}\alpha + \frac{1}{A_D}(\mathbf{g}\bullet\mathbf{x})_f |\mathbf{s}_f| \nabla_f^{\perp}\rho\right)[\alpha_i]\right)\right]$$

$$- \left[\nabla\bullet\left(\left(\frac{1}{A_D}\right)\nabla p^*\,[\alpha_i]\right)\right] + \left[\nabla\bullet\left([\alpha_i]\sum_{k=1,k\neq i}^{N}\alpha_k\,\phi_{r,ij}\right)\right] = 0$$

- Pressure equation, with separated phase fluxes (coupling terms)

$$\left[\nabla\bullet\left[\left(\frac{1}{A_D}\right)_f \nabla\,[p^*]\right]\right] = \nabla\bullet\left(\sum_{i=1}^{N}\alpha_i\phi^*\right)$$

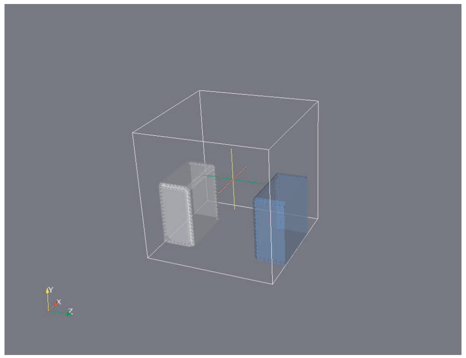
- Closure equation and definition of fluxes

$$\sum_i \alpha_i = 1 \qquad \phi^* = \sum_i \alpha_i^N \phi^*$$

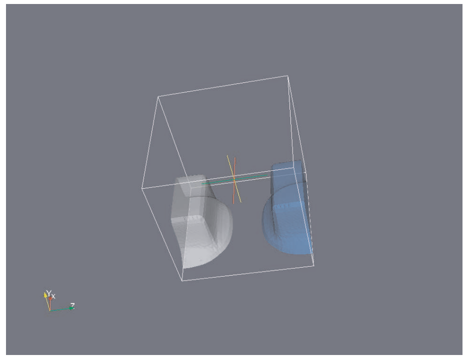
# Block Matrix in Use: Multi-Phase VOF

Multi-Phase Volume-of-Fluid Solver: Solution Strategy

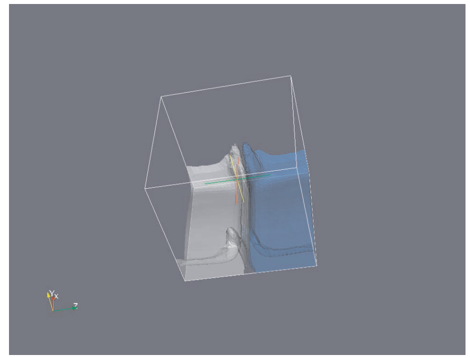
- Above equations are dumped into a block matrix format, with coefficient size N + 1: $(p^*, \alpha_i)$ and solved in a block-coupled manner
- Result: strong coupling between $\alpha$s and $p$: no dominant phase



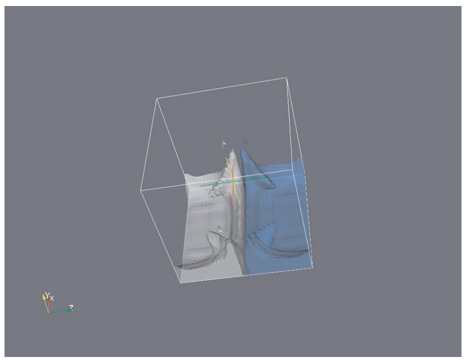
t = 0



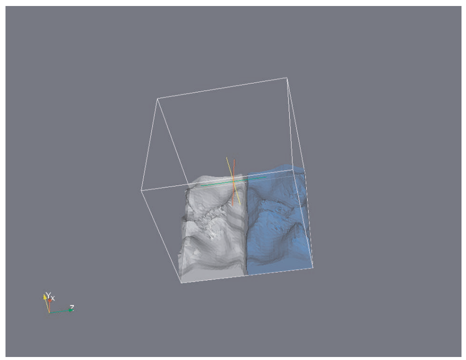
t = 1



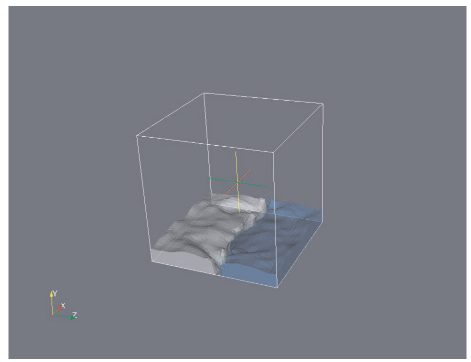
t = 2



t = 3



t = 4

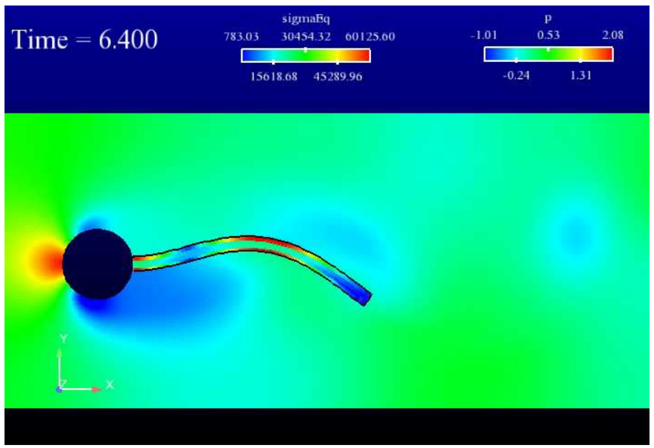


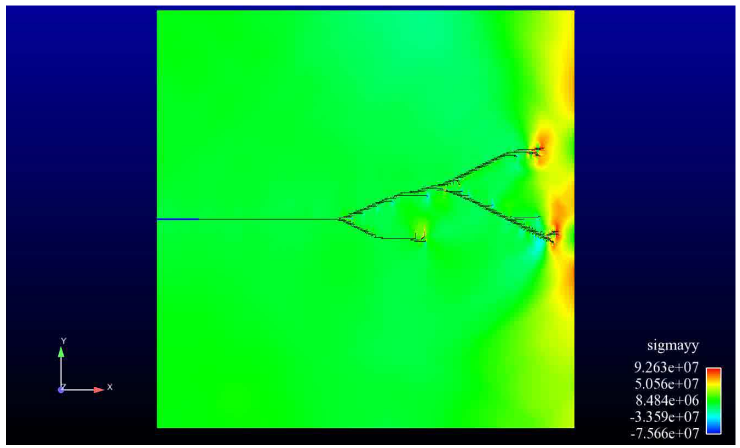
t = 5

# Fluid-Structure Interaction

Fluid-Structure Coupling Capabilities in OpenFOAM

- As a Continuum Mechanics solver, OpenFOAM can deal with both fluid and structure components: easier setup of coupling
- (Parallelised) surface coupling tools implemented in library form: facilitate coupling to external solvers without “coupling libraries” using proxy surface mesh
- Structural mechanics in OpenFOAM targeted to non-linear phenomena: consider best combination of tools
  - Large deformation formulation in absolute Lagrangian formulation
  - Independent parallelisation in the fluid and solid domain
  - Parallelised data transfer in FSI coupling
- Dynamic mesh tools and boundary handling used to manipulate the fluid mesh

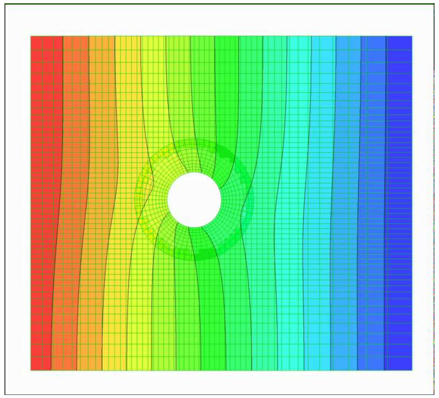


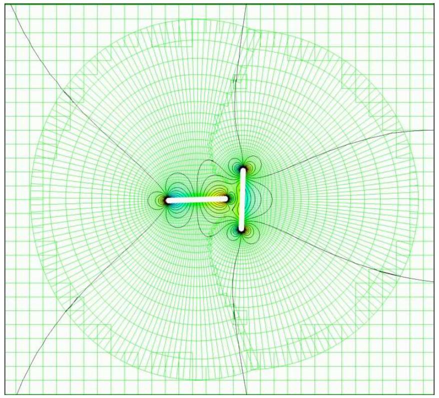



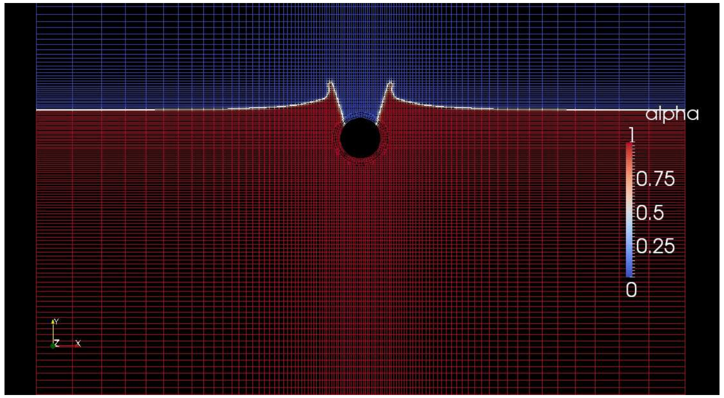
# Overset Grid in OpenFOAM

`foamedOver`: Overset Grid Technology in OpenFOAM

- Work by David Boger, Penn State University using SUGGAR and DirtLib libraries developed by Ralph Noack, Penn State (must mention Eric Paterson!)
- 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! Currently testing for free surface flows



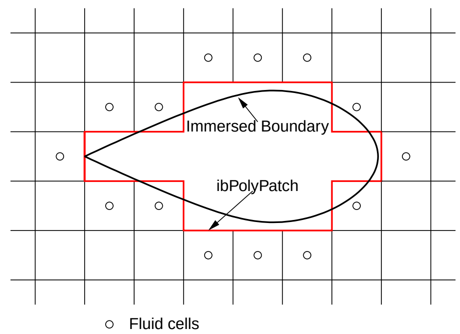


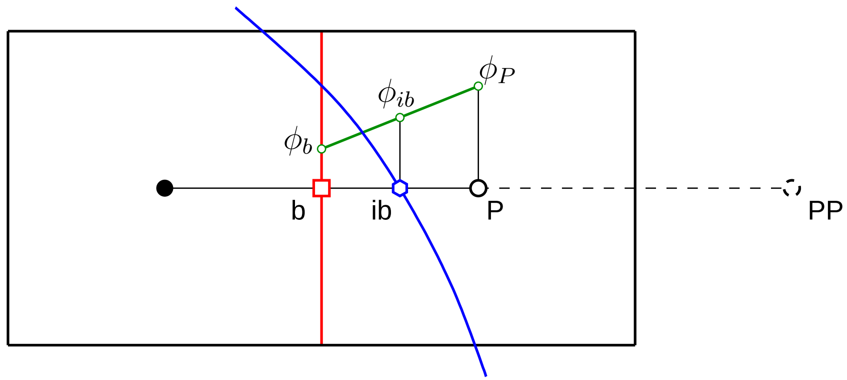


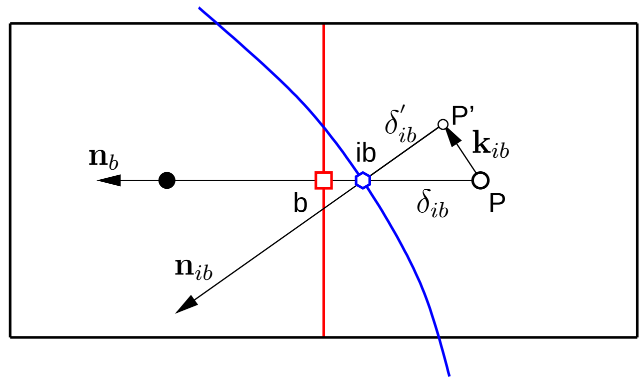
# Immersed Boundary Method

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



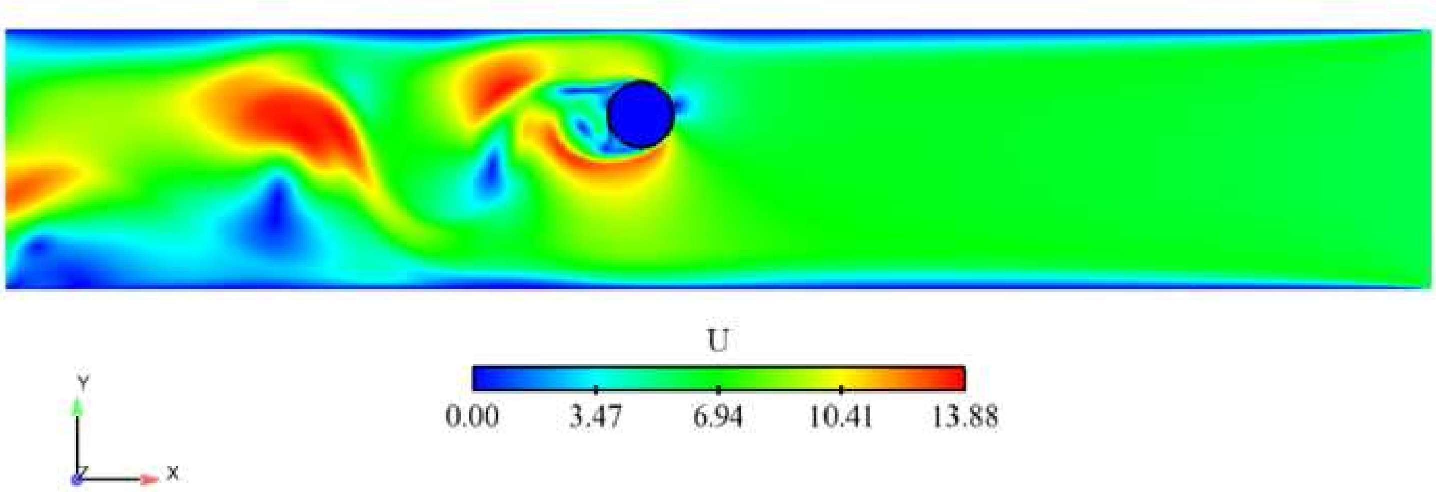



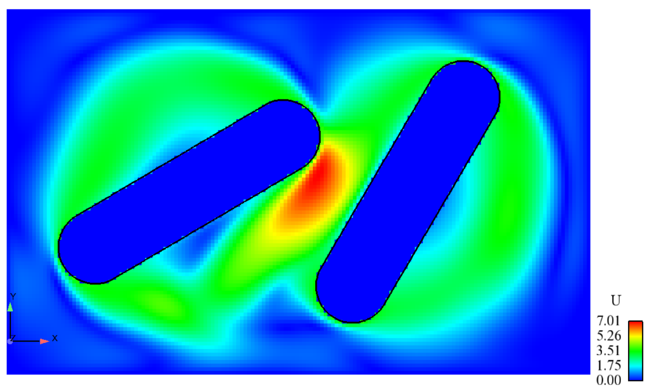
# Immersed Boundary Method

Immersed Boundary Method: Examples

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



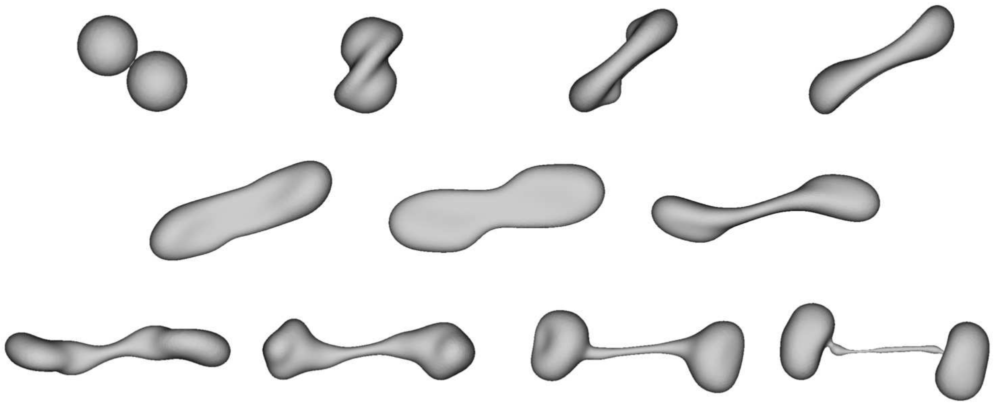

- Laminar flow around two counter-rotating elements in a cavity

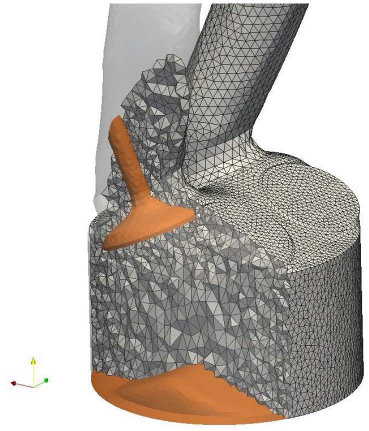


# Tetrahedral Edge Swapping

Re-Meshing with Tetrahedral Edge Swapping

- In cases where mesh motion involves topological change at the boundary or unpredictable mesh deformation, topological change machinery is impractical: cannot decide a-priori where to place topology modifiers
- **Dynamic remeshing using tetrahedral edge swapping**
  - Motion is prescribed on external boundaries
  - Tetrahedral cell quality examined continuously: bad cells trigger automatic remeshing without user interaction: answers to `dynamicMesh` interface
  - Implemented by **Sandeep Menon, UMass Amherst** as a ready-to-use library
- Example: viscoelastic droplet collision using free surface tracking
- Can be used for all dynamic mesh cases: ultimate ease of mesh setup!

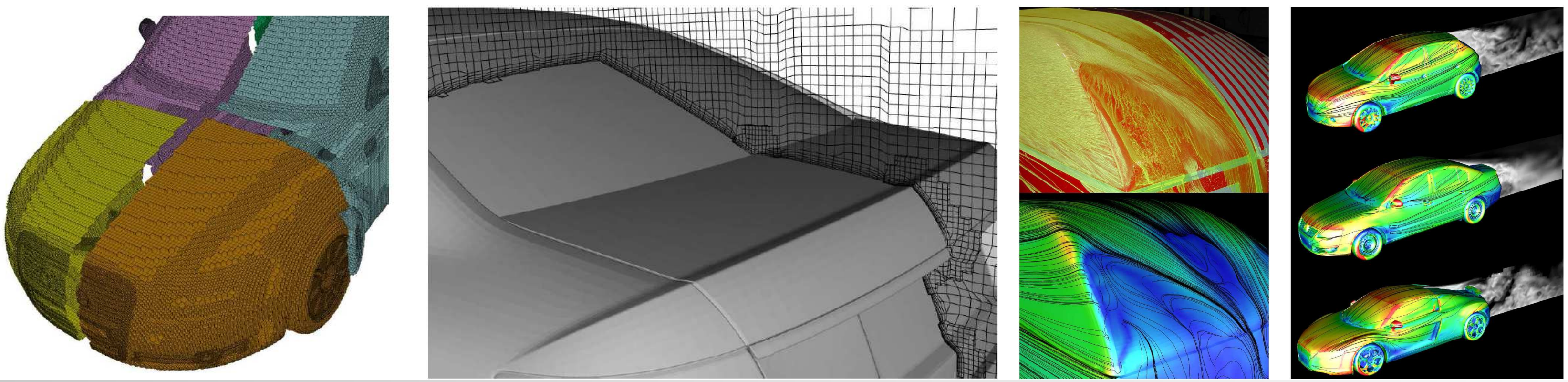




# External Aerodynamics with LES

Detached Eddy Simulation for External Aerodynamics

- Pushing state-of-the-art by applying Detached Eddy Simulation (DES) to full car body external aerodynamics simulations: native solver and mesher, no change
- Increase in simulation cost over transient RANS is over 1 order of magnitude!
- **Controlling the Cost of Full Car DES**:
  - Automated meshing and simulation environment, from STL surface of the car body to averaged DES results and forces
  - Hex-core mesher with near-wall layers and local refinement: mesh is designed to make it good for second-order LES numerics with minimal cost
  - No parallel license cost of CFD solver: simulations run on approx. 200 CPUs
- Improvement in $C_D, C_L$ and force-per-component predictions due to better capturing of turbulence and transient flow features



Reproduced with permission SAE 2009-01-0333, Islam et.al.

# CFD in Metallurgical Applications

Assembling a Matrix for Conjugate Heat Transfer Problems

- OpenFOAM supports multi-region simulations, with possibility of separate addressing and physics for each mesh: multiple meshes, with local fields
- Some equations present only locally, while others span multiple meshes

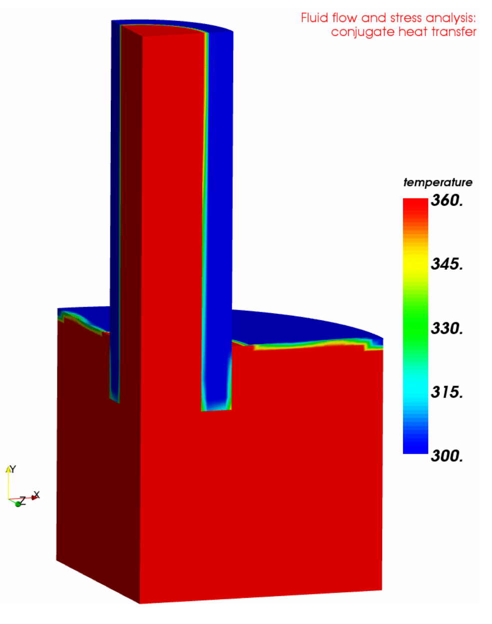
```
coupledFvScalarMatrix TEqns(2);

TEqns.hook
(
    fvm::ddt(T) + fvm::div(phi, T)
  - fvm::laplacian(DT, T)
);

TEqns.hook
(
    fvm::ddt(Tsolid) - fvm::laplacian(DTsolid, Tsolid)
);

TEqns.solve();
```

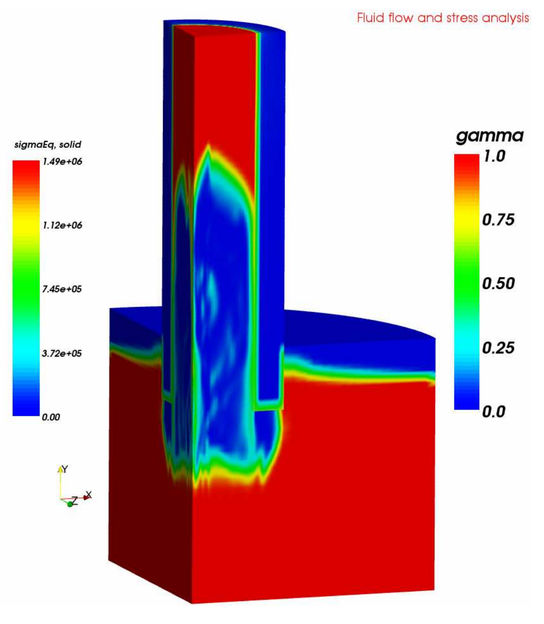
- Coupled solver handles multiple matrices together in internal solver sweeps: arbitrary matrix-to-matrix and domain-to-domain coupling

# CFD in Metallurgical Applications

Conjugate Heat Transfer and Thermal Shock

- Coupling may be established geometrically: adjacent surface pairs
- Each variable is stored only on a mesh where it is active: (U, p, T)
- Choice of conjugate variables is completely arbitrary: e.g. catalytic reactions
- Coupling is established only per-variable: handling a general coupled complex physics problem rather than conjugate heat transfer problem specifically
- Allows additional models to be solved on each region without overhead: structural stress analysis, turbulence or LES

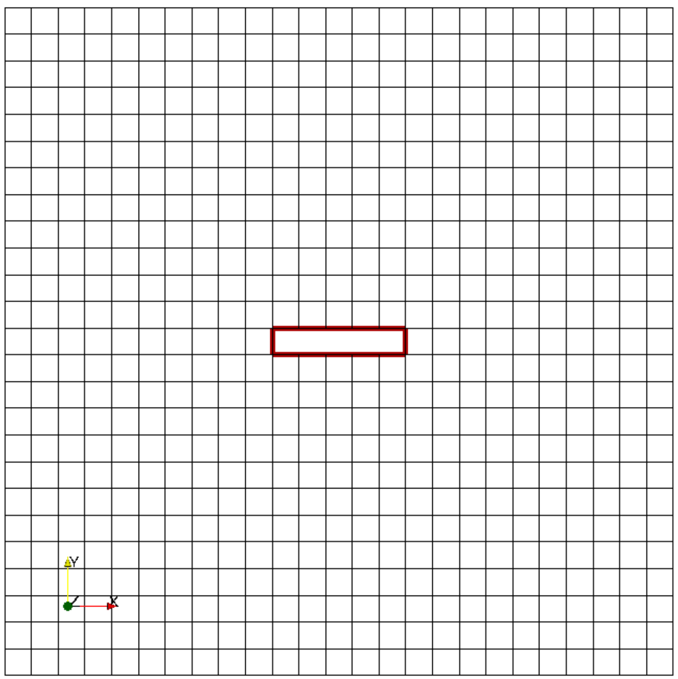


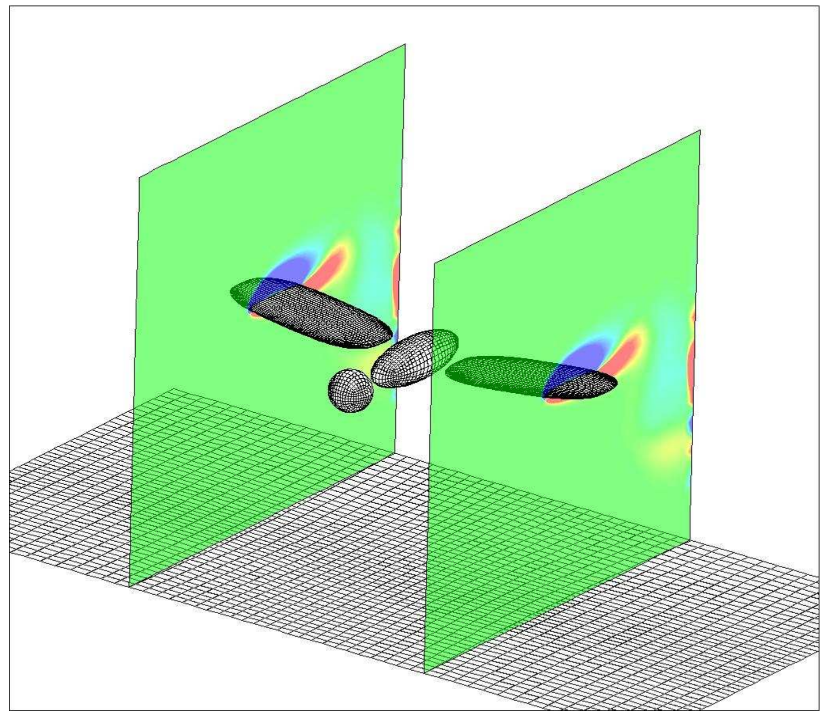



# Radial Basis Function

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** built into the derivation
- Used for mesh motion in cases of large deformation: no inverted faces or cells
- Implemented by Frank Bos, TU Delft and Dubravko Matijašević, FSB Zagreb





# Radial Basis Function

RBF Mesh Morphing Object

- RBF morphing object defines the parametrisation of geometry (space):
    1. Control points in space, where the parametrised control motion is defined
    2. Static points in space, whose motion is blocked
    3. Range of motion at each control point: $(\mathbf{d}_0, \mathbf{d}_1)$
    4. Set of scalar parameters $\delta$ for control points, defining current motion as

$$\mathbf{d}(\delta) = \mathbf{d}_0 + \delta(\mathbf{d}_1 - \mathbf{d}_0), \quad \text{where} \quad 0 \leq \delta \leq 1$$

- For each set of $\delta$ parameters, mesh deformation is achieved by interpolating motion of control points $\mathbf{d}$ over all vertices of the mesh: new deformed state of the geometry
- Mesh in motion remains valid since RBF satisfies smoothness criteria

Using RBF in Optimisation

- Control points may be moved individually or share $\delta$ values: further reduction in dimension of parametrisation of space
- Mesh morphing state is defined in terms of $\delta$ parameters: to be controlled by the optimisation algorithm

# Geometric Shape Optimisation

Geometric Shape Optimisation with Parametrised Geometry

- Specify a desired object of optimisation and use the parametrisation of geometry to explore the allowed solution space in order to find the **minimum of the optimisation objective**

$$objective = f(\mathbf{shape})$$

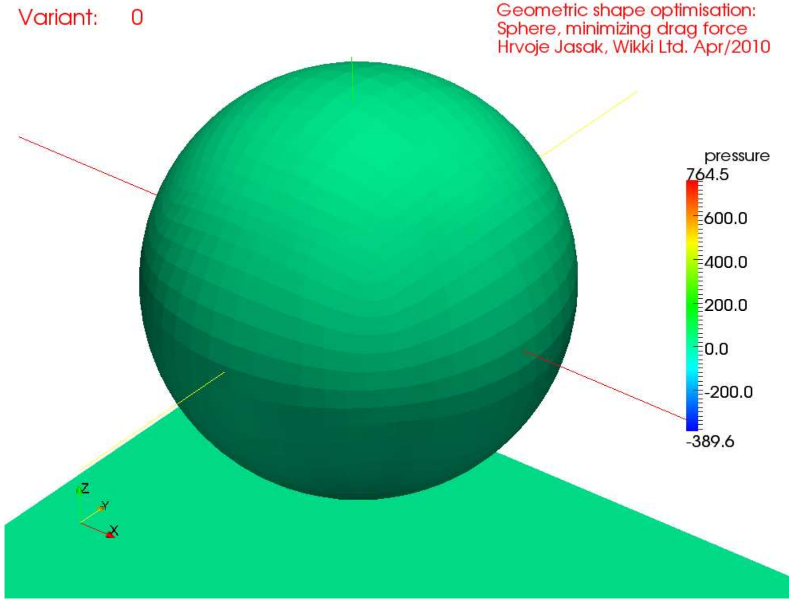
1. **Parametrisation of Geometry**
   - Computational geometry is complex and usually available as the computational mesh: a large amount of data
   - Parametrisation tool: **RBF mesh morphing**, defining deformation at a small number of mesh-independent points in space
2. **CFD Flow Solver** is used to provide the flow solution on the current geometry, in preparation for objective evaluation
3. **Evaluation of Objective**: usually a derived property of the flow solution
4. **Optimiser Algorithm**: explores the solution space by providing sets of $\mathbf{shape}$ coordinates and receiving the value of $objective$. The search algorithm iteratively limits the space of solutions in search of a minimum value of $objective$

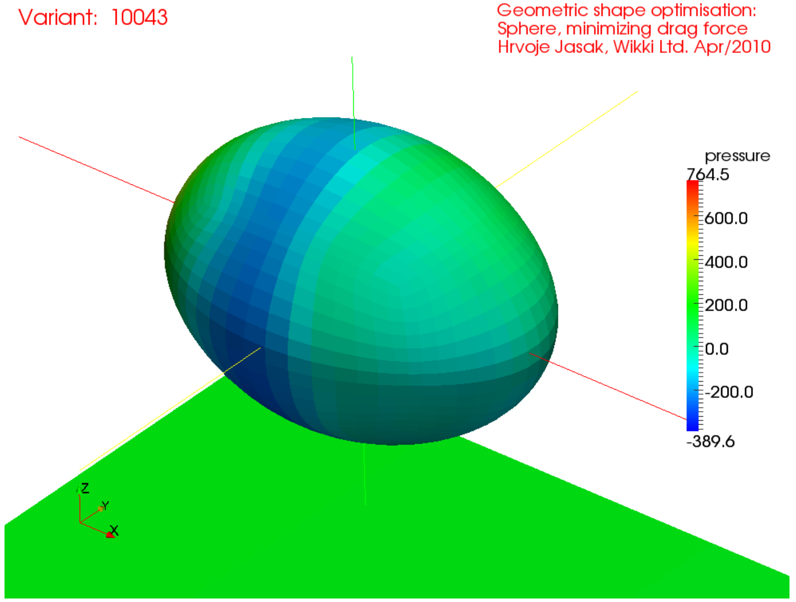
# Geometric Shape Optimisation

3-D Sphere: Minimising Drag Force

- Using 9 control points in motion, with symmetry constraints: 4 points in front square, radial motion; 4 points in back square, radial and axial motion; 1 tail point, axial motion only
- Optimisation is performed with 3 parameters:

```
iter = 1 pos = (0.2 0.7 0.2) v = 147.96 size = 0.2997
iter = 5 pos = (0.06111 0.7092 0.7092) v = 106.26 size = 0.2153
iter = 12 pos = (0.03727 0.9354 0.3830) v = 77.934 size = 0.0793
iter = 22 pos = (0.04095 0.9458 0.3413) v = 75.821 size = 0.006610
```



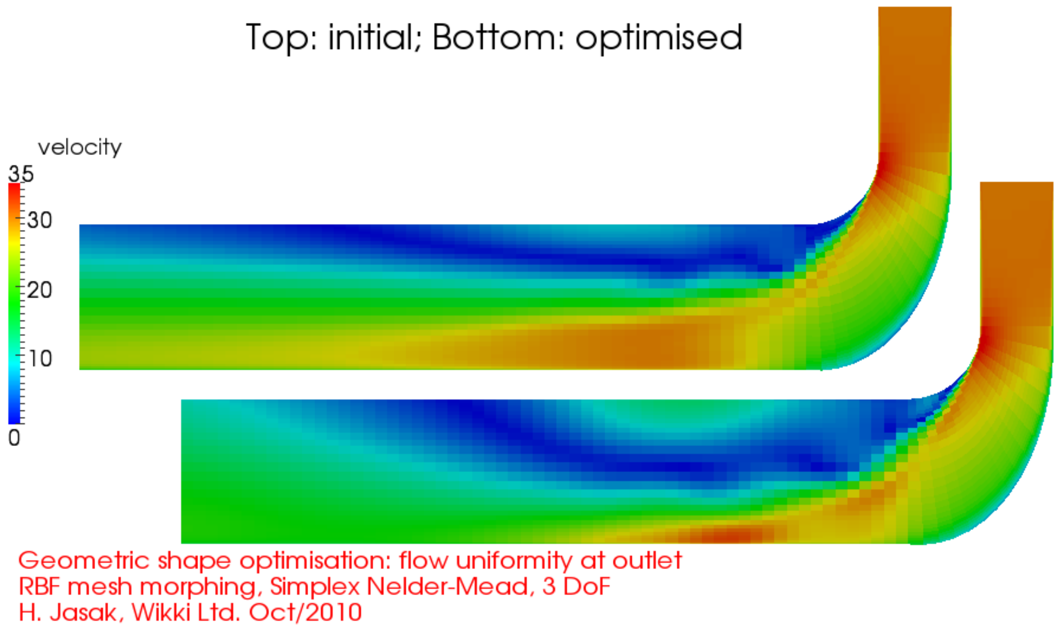



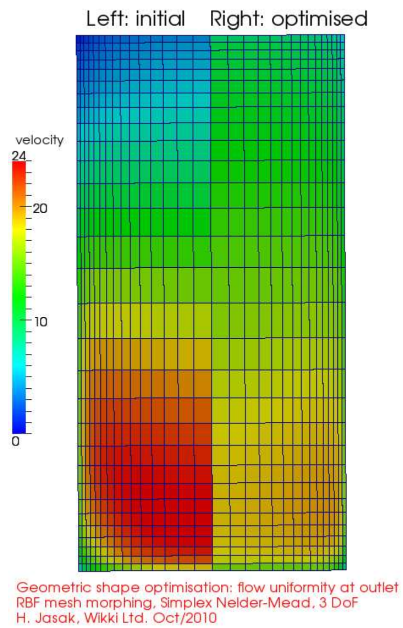
# Geometric Shape Optimisation

HVAC 90 deg Bend: Flow Uniformity at Outlet

- Flow solver: incompressible steady-turbulent flow, RANS $k - \epsilon$ model; coarse mesh: 40 000 cells; 87 evaluations of objective with CFD restart
- RBF morphing: 3 control points in motion, symmetry constraints; 34 in total
- Objective: flow uniformity at outlet plane

```
iter = 0 pos = (0.9 0.1 0.1) v = 22.914 size = 0.69282
iter = 5 pos = (0.1 0.1 0.1) v = 23.0088 size = 0.584096
iter = 61 pos = ((0.990164 0.992598 0.996147) v = 13.5433 size = 0.00095
```

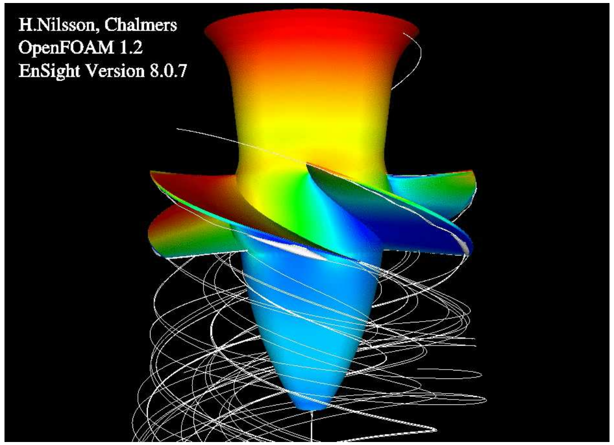


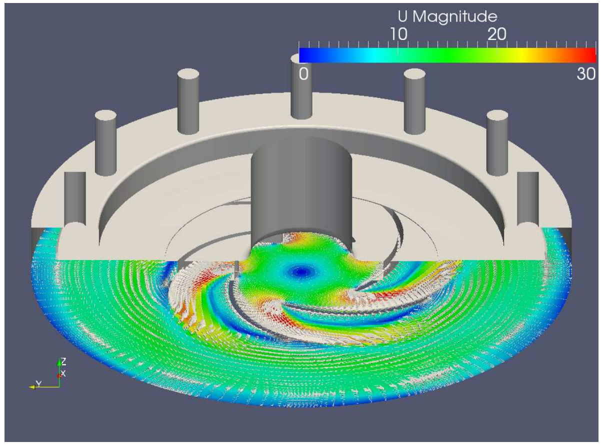



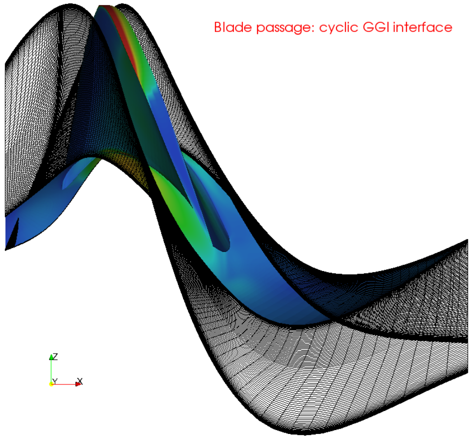
# OpenFOAM Turbo Tools

General Grid Interface

- Turbomachinery CFD requires additional features: implemented in library form
- **General Grid Interface (GGI)** and its derived forms
  - Cyclic GGI
  - Partial overlap GGI
  - Mixing plane interface (under testing)
- Implementation and parallelisation is complete: currently running validation cases in collaboration with commercial clients and Turbomachinery Working Group
- Other turbo-related components in pipeline: harmonic balance solver
- Library-level implementation allows re-use of GGI beyond turbomachinery



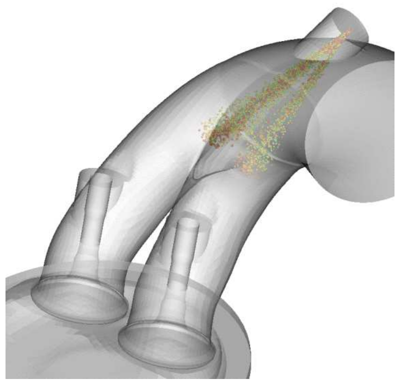



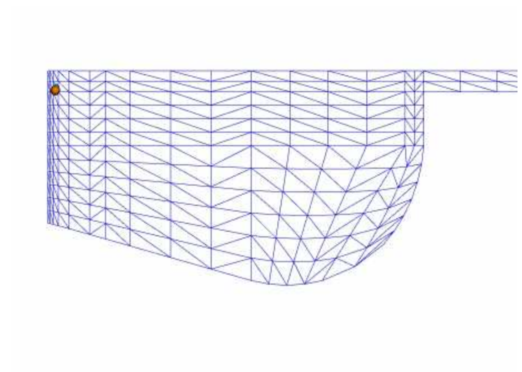



# Automotive Applications

Spray, Wall Film and Combustion Simulations in Internal Combustion Engines

- Complete simulation of spray injection, evaporation, wall film and combustion in a GDI engine. Mesh motion and topological changes as shown before
- Basic flow solver, **automatic mesh motion**, topological changes used in standard form. Simulation includes intake stroke (moving piston + valves): **reverse tumble**
- Full suite of Diesel spray modelling using Lagrangian modelling framework
- Implementation of wall film and spray-film interaction: Željko Tuković, FSB
- Mesh sensitivity of **spray penetration**: solved with adaptive refinement!
- Authors of engine simulations: Tommaso Lucchini, Gianluca D’Errico, Daniele Ettore, Politecnico di Milano and Dr. Federico Brusiani, University of Bologna

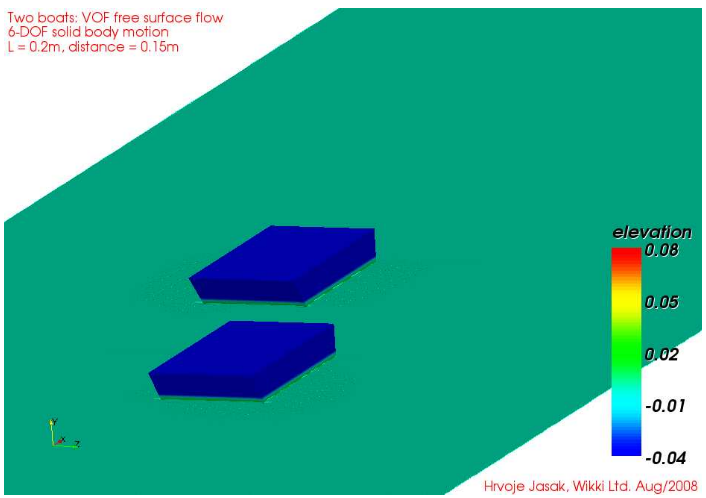



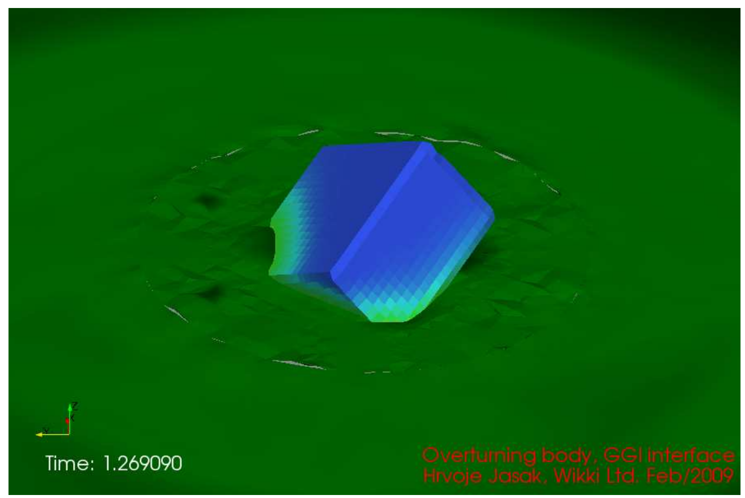


# Complex Mesh Motion

6-DOF Floating Body in Free Surface Flow

- **Flow solver**: turbulent VOF free surface, with moving mesh support
- **Mesh motion** depends on the forces on the hull: 6-DOF solver
- **6-DOF solver**: `ODE + ODESolver` energy-conserving numerics implemented using quaternions, with optional elastic/damped support
- Variable diffusivity Laplacian motion solver with 6-DOF boundary motion as the boundary condition for the mesh motion equation
- Topological changes to preserve mesh quality on capsize
- Coupled transient solution of flow equations and 6-DOF motion, force calculation and automatic mesh motion: custom solver is built from library components



# Stammtisch - What is it about?

- Stammtisch (German): regulars' table often setup in the pubs of rural communities
- here: meeting people working with OpenFOAM
- The plan: More casual and interactive than a user conference; Strengthen "local" community
- Initiators: Prof. Schütz & Prof. Janoske
- Stammtisch South: Mosbach $\rightarrow$ Stuttgart $\rightarrow$ München $\rightarrow$ Mosbach $\rightarrow$ Stuttgart $\rightarrow$ Mannheim $\rightarrow$ München $\rightarrow$ Ulm
- Stammtisch North: Wolfsburg $\rightarrow$ Braunschweig $\rightarrow$ Berlin $\rightarrow$ Bremen $\rightarrow$ Berlin $\rightarrow$ Dortmund
- Similar activities elsewhere – Will focus organisational issues

# Stammtisch - Typical Agenda

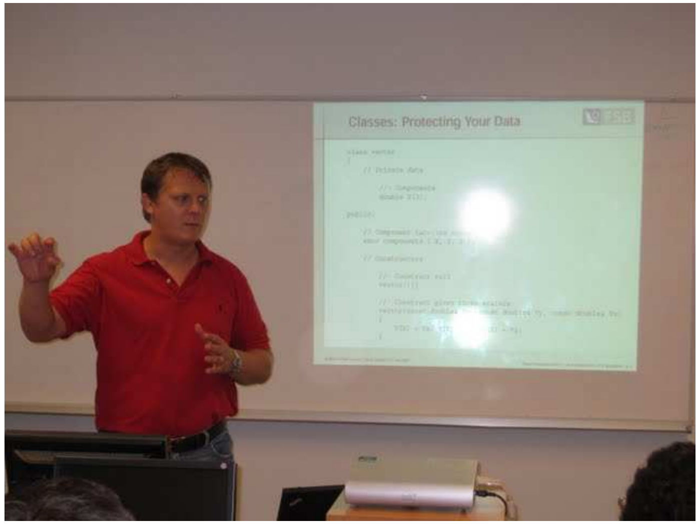
- Welcome (0:10)
- Technical Session,
  3-4 Presentations (1:30)
- Lunch Break (1:00)
- Birds of a Feather (BoF) grouping (0:15)
- BoF Session (3:00)
- BoF Summary (0:30)
- AOB & Next meeting (0:15)
- Dinner (and Beer) (?:??)

# Lessons learned

- Choose impartial venue (e.g. university)
- How to make sure that know-how is passed on from one venue to the other?
- Form a committee of former organisers, which offers advise & support to (new) organisers
- Use Doodle (or similar) to speed-up BoF grouping

# Summerschool - What is it about?

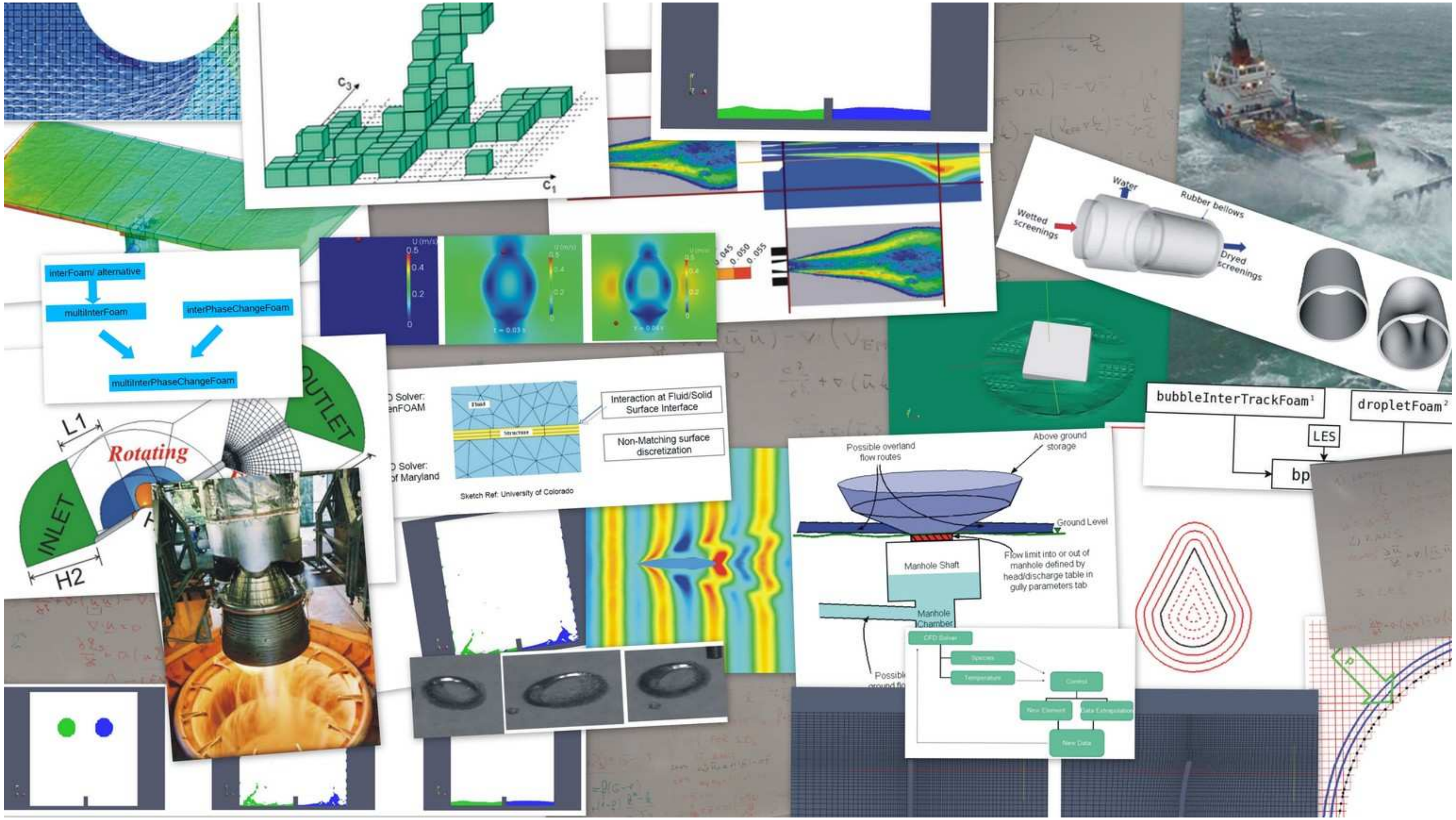
- providing tuition for a small and selected group
- students and researchers in academia and industry
- expand the physical modelling knowledge, numerics and programming skills of attendees
- Organisers: Prof. Hrvoje Jasak & Prof. Zdravko Terze



# Activities

- Each student works on his project
- direct supervision and one-to-one work
- Lectures on chosen topics:
  - All aspects of numerical modelling
  - Computational Continuum Mechanics and CFD
  - Object-Oriented Programming and C++
- Having fun together

# Last year's projects



# General Information

- When? $\rightarrow$ Once a year, the first two weeks in September
- Where? $\rightarrow$ At the University of Zagreb, Croatia
- How to apply?
  $\rightarrow$ Application with a description of your project, current problems and goals. **You missed the deadline for 2011!**
- Who can apply?
  $\rightarrow$ All students on MSc and PhD university courses
  $\rightarrow$ 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

Project Status Summary

- OpenFOAM is a free software, available to all at no charge: GNU Public License
- Object-oriented approach facilitates model implementation
- Equation mimicking opens new grounds in Computational Continuum Mechanics
- Extensive capabilities already implemented; open design for easy customisation
- Solvers are validated in detail and match the efficiency of commercial codes
- Open Source model dramatically changes the industrial CFD landscape: new business model
- Help to shape the community!
- **Sixth OpenFOAM Workshop**: Penn State University 13-16 June 2011 `http://www.openfoamworkshop.org`