



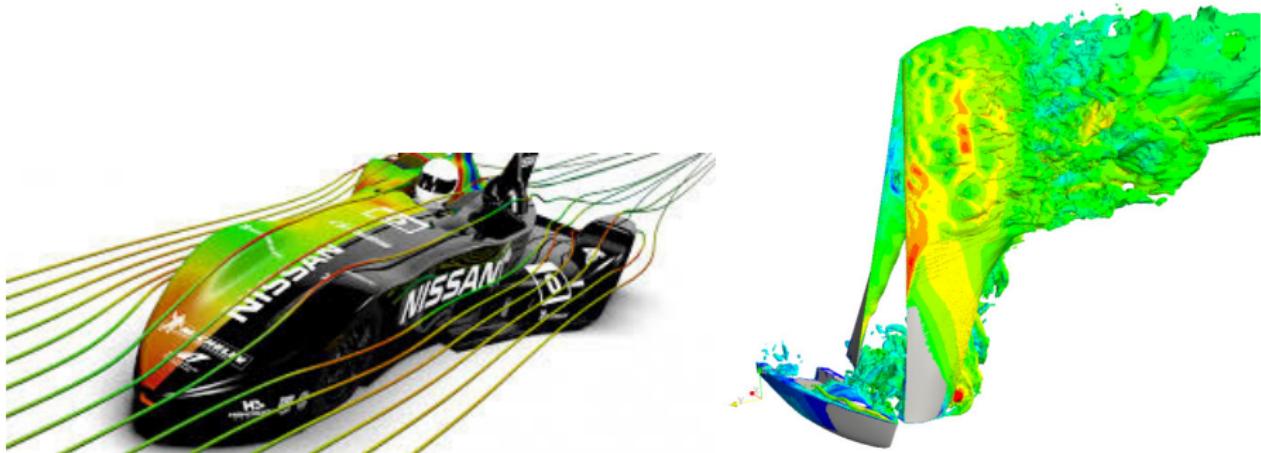
051176 - Computational Techniques for Thermochemical Propulsion
Master of Science in Aeronautical Engineering

A (brief) introduction to CFD

Prof. **Federico Piscaglia**
Dept. of Aerospace Science and Technology (DAER)
POLITECNICO DI MILANO, Italy
federico.piscaglia@polimi.it

What is CFD?

H. K. Versteeg, W. Malalasekera: "Computational Fluid Dynamics (CFD) is the analysis of systems involving fluid flow, heat transfer and associated phenomena such as chemical reactions by means of computer-based simulations"



source: www.totalsimulation.co.uk via Google

In other words, the discretized form of the Navier-Stokes equations in CFD is solved with the help of a computer...

CFD History

Aerospace industry is the first and most relevant in the use of CFD

- EARLY '60s: beginning of CFD
- '70s: first success came to prominence
- '80s: first CFD-service industry started up
- '90's: CFD industry expanded significantly
- 2000-PRESENT: CFD is quite well established in most of industrial workflows
- Current challenge: make complex CFD easier to apply on daily-basis for not expert users.

Find more information about the CFD history here:

http://aero-comlab.stanford.edu/Papers/NASA_Presentation_20121030.pdf

PROS:

- + allows the study of complex flow phenomena
- + cheaper than experiments
- + non-intrusive insight of the physics;
- + repeatable
- + faster (than experiments) to conduct parametric studies

CONS:

- results from CFD rely on assumption (models)
- CFD is NOT a replacement for experiments, but a compliment;
- CFD is still not reliable enough to completely replace experiments

...helps to know:

- the governing physics of phenomena
- numerical analysis to prevent code crashing
- how to work on **UNIX/Linux-based systems**
- **python/bash scripting** and (some) programming
- CAD software

```
+ fvn::SuSp((2.0/3.0)*alpha*rho*divU, k_)
+ fvn::Sp(alpha*rho*epsilon_/k_, k_)
+ kSource()
};
```

- PRE-PROCESSING

- The geometry and physical bounds of the problem can be defined using computer aided design (CAD). The fluid volume (or fluid domain) is extracted.
- The fluid domain is discretized into several control volumes (the mesh).
- The physical modeling (= form of the governing equations to solve) and the fluid properties are defined, together with the boundary conditions at all bounding surfaces of the fluid domain and the initial conditions.

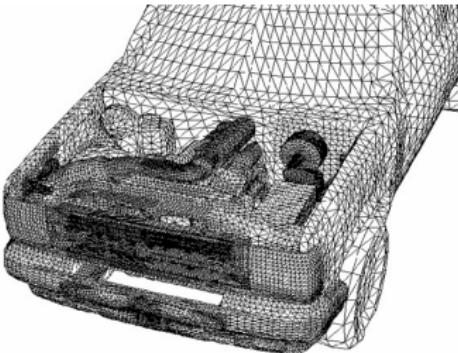
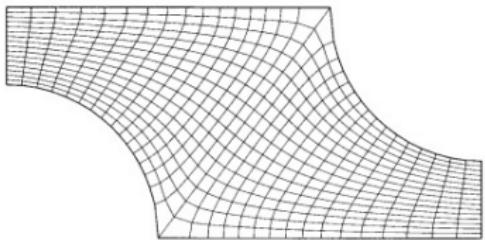
- **CALCULATION:** simulation is started and governing equations are solved iteratively.

- **POST-PROCESSING** is used for the analysis and visualization of the resulting solution.

How does a CFD code work?



CFD codes are structured around the numerical algorithms that can tackle fluid flow problems. All the codes contain three main elements:



- 1) A **pre-processor**: it consists of the input of a flow problem to a CFD program by means of an operator-friendly interface and the subsequent transformation of this input into a form suitable for use by the solver. The user activities at the pre-processing stage involve:
 - Definition of the region of interest (computational domain) and its subdivision into a number of smaller, non-overlapping sub-domains: a mesh of cells
 - Selection of the physical and chemical phenomena that need to be modelled.
 - Definition of fluid properties
 - Specification of appropriate boundary conditions at cells which coincide with or touch the boundary domain.

How does a CFD code work?



CFD codes are structured around the numerical algorithms that can tackle fluid flow problems. All the codes contain three main elements:

2) A **numerical solver**: different techniques (finite element, finite difference, spectral elements) exist for numerical solution of the governing equations. They all perform the following operations:

- Approximation of the unknown flow variables by means of simple functions
- Discretization by substitution of the approximation into governing flow equations and subsequent mathematical manipulations
- Solution of the algebraic equations.

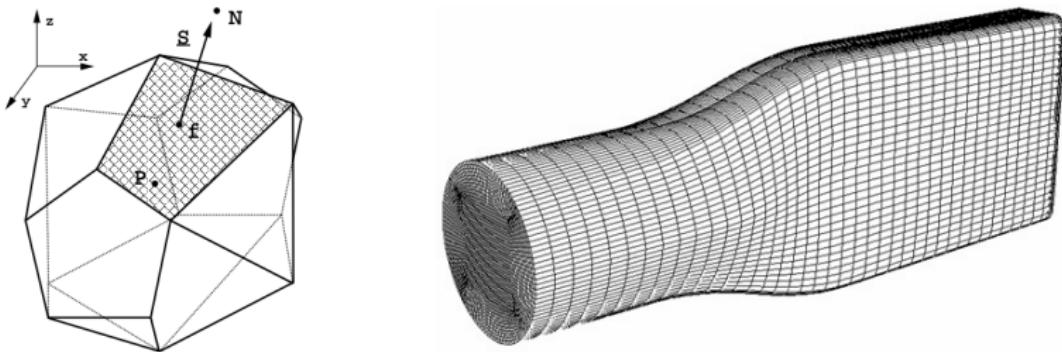
Most of the widely used commercial CFD codes use the **finite volume method** for spatial discretization, operating as follows:

- formal integration of the governing equations of fluid flow over all the control volumes (cells) of the solution domain (mesh)
- Discretization involves the substitution of a variety of finite-difference-type approximations for the terms in the integrated equation representing flow processes such as convection, diffusion and sources. This converts the integral equations into a system of algebraic equations.
- Solution of the algebraic equations by an iterative method.

How does a CFD code work?



The Finite Volume (FV) method



The FV method uses the integral form of the conservation equations as its starting point:

- the solution domain is divided into a finite number of continuous control volumes, where conservation equations are applied;
- variable values are calculated at the centroid of each CV
- FV method can accommodate any type of grid;
- the method is conservative by construction, as long as surface integrals (representing convective and diffusive fluxes) are the same for the CVs sharing the boundary.

DISADVANTAGES: high-order methods are difficult to implement in a FV framework.

How does a CFD code work?



GOVERNING EQUATIONS

Constitutive relations, together with the governing equations for a continuum, create a closed system of partial differential equations for Newtonian fluids:

Mass conservation:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{u}) = 0$$

Momentum conservation:

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \frac{2}{3} \mu \nabla \vec{u} \vec{I} + \nabla \cdot [\mu (\nabla \vec{u} + (\nabla \vec{u})^T)] + \rho \vec{g} + S$$

Energy equation:

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \vec{u}) = -\nabla p \vec{u} + \frac{2}{3} \mu (\nabla \cdot \vec{u}) \vec{u} + \nabla \cdot [\mu (\nabla \vec{u} + (\nabla \vec{u})^T) \vec{u}] + \nabla \cdot (\lambda T) + \rho \dot{Q} + \rho \vec{g} \vec{u}$$

How does a CFD code work?



Constitutive relations for Newtonian fluids

A system of 5 differential equations describes the fluid motion:

- mass conservation equation;
- three equations for momentum conservation (along x, y and z components)
- energy equation.

To achieve the closure of the system, the so-called **constitutive relations** are needed; their formulation depends on the properties of the continuous medium. For Newtonian fluids, the following set of constitutive relations is used:

- The internal energy equation (or enthalpy), defining the internal energy as function of the pressure p and temperature T :

$$u = u(p, T) \text{ or } h = h(p, t)$$

- The equation of state: $\rho = \rho(p, T)$
- The Fourier's law of heat conduction: $q = -k\nabla T$ or $q = -\alpha\nabla h$
- Generalized form of the Newton's law of viscosity:

$$\vec{\tau} = \mu \left[\nabla \vec{U} + (\nabla \vec{U}^T) \right] + \left(\frac{2}{3} \mu \nabla \cdot \vec{u} \right) \vec{I}$$

How does a CFD code work?



Conservation of scalar quantities: the integral form of the equation describing conservation of a scalar quantity Φ (**TRANSPORT EQUATION**) can be expressed in its general form as:

$$\underbrace{\frac{\partial}{\partial t} \int_{\Omega} \rho \phi d\Omega}_{\text{temporal derivative}} + \underbrace{\int_{\Omega} \rho \phi (\vec{u} - \vec{u}_b) \cdot \vec{n} dS}_{\text{convection term}} = \underbrace{\int_{\Omega} \Gamma \operatorname{grad} \Phi \cdot \vec{n} dS}_{\text{diffusion term}} + \underbrace{\sum f_{\Phi}}_{\text{source/sink terms}}$$

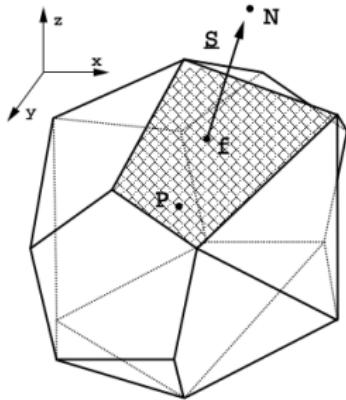
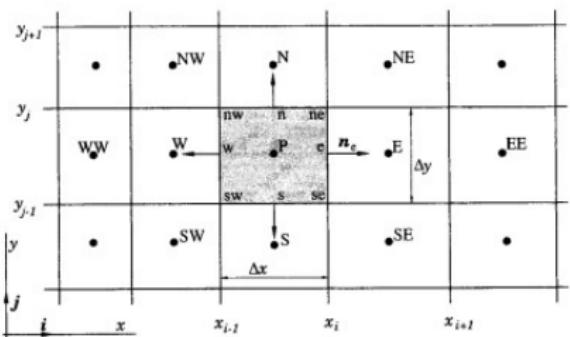
where f_{Φ} represents the transport of Φ by mechanisms other than convection and any sources or sinks of the scalar.

Example in OpenFOAM:

```
tmp<fvVectorMatrix> tUEqn
(
    fvm::ddt(rho, U) + fvm::div(phi, U)
 ==
 - turbulence->divDevRhoReff(U)
 fvOptions(rho, U)
);

solve(UEqn == -fvc::grad(p));
```

How does a CFD code work?



Approximation of surface integrals:

$$\int_S f dS = \sum_k \int_{S_k} f dS = \sum_k f_k S_k$$

- to calculate the surface integral exactly, one would need to know the integrand f everywhere on the surface S_p .
- this information is not available, as only **node center** values are calculated, so an **APPROXIMATION MUST BE INTRODUCED**. The best way to do it is:
 - the integral is approximated in terms of the variable values at the face center
 - cell face values are approximated in terms of the CV-centered values

How does a CFD code work?



In the numerical solver, the result of the discretization process is a system of algebraic equations, which are linear or non linear according to the nature of the partial differential equation(s) from which they are derived.

The diagram shows a 5x5 grid of nodes. A central node has four neighbors: top (T), bottom (B), left (L), and right (R). The coefficients for these nodes are labeled: A_W for the left neighbor, A_S for the bottom neighbor, A_N for the top neighbor, and A_E for the right neighbor. The grid also includes boundary nodes with their own local stencils. To the right of the grid, a matrix equation is shown:

$$\begin{bmatrix} \text{Stencil} \\ \vdots \\ \text{Stencil} \end{bmatrix} * \begin{bmatrix} q_W \\ q_S \\ q_P \\ q_N \\ q_B \\ q_E \end{bmatrix} = \begin{bmatrix} Q_W \\ Q_S \\ Q_P \\ Q_N \\ Q_B \\ Q_E \end{bmatrix}$$

- linear system must be solved implicitly → inverting the coefficient matrix is too expensive!
- preconditioning is needed
- linear solvers (multigrid, Conjugate Gradient for symmetric matrices; bi-conjugate gradients for non-symmetric matrices) lead to different performances
- coefficient matrix must be diagonal dominant: boundary conditions influences off-diagonal terms!

How does a CFD code work?



3) A **post-processor**: it is used to visualize and extract relevant simulation data.

- Graphical pre-processing:
 - Domain geometry and grid display
 - Vector plots
 - Line and shaded contour plots
 - Particle tracking
 - View manipulation
- Data analysis
 - sampling
 - probing
 - computation of derived quantities (local or averaged)

Solving a flow problem in CFD



In solving fluid flow problems it is necessary to be aware that the underlying physics is complex and the results generated by a CFD code as at best as good as the physics embedded in it and at worst as good as its operator.

- It is necessary to identify and formulate the flow problem in terms of physical and chemical phenomena that need to be considered.
 - Is it 2D or 3D?
 - Is density constant or not?
 - Is the flow turbulent or laminar?
- Then, the governing equations of fluid flow are solved. They must represent mathematical statements of the conservation laws of physics:
 - The **mass** of a fluid is conserved
 - The **rate of change of momentum** equals the sum of the forces on a fluid particle
 - The **rate of change of energy** is equal to the sum of the rate of heat addition and the work done on a fluid particle (first law of thermodynamics).

The behavior of the fluid will be described in terms of macroscopic properties, such as velocity, pressure, density, temperature and their space and time derivatives.

Tool for the class: OpenFOAM®

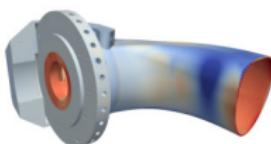
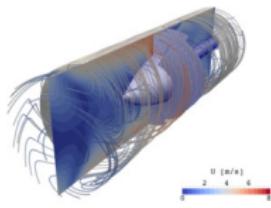
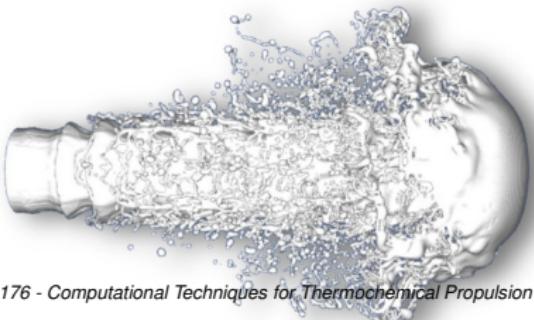


- Ready-to-use CFD code

- applications and state-of-the-art models for CFD already available in the code
- polyhedral mesh support
- include mesh generator, conversion from different mesh formats

- Open-source, object-oriented C++ at no license costs:

- code customization
- maximum code re-use
- minimum code maintainance
- research in a collaborative environment



OpenFOAM's history

- Early '90s: the research code **FOAM** was created by **Henry Weller** and **Hrvoje Jasak** at the IMPERIAL COLLEGE OF LONDON, under the supervision of Prof. David Gosman. Objective of the work: create an object oriented platform where it was possible to efficiently implement and test numerical and physical models.
- 2000-2004: FOAM becomes a (partially) open-source commercial code, distributed by Nabla Ltd. The code has limited success.
- 2005: FOAM becomes OpenFOAM® and it is released under the GPL License and it is distributed by OpenCFD. Number of user and developers rapidly increases (> 20k).
- 2013: the OpenFOAM® trademark is acquired by ESI-OpenCFD®

Today OpenFOAM® is one of the most famous CFD codes and it is widely used both for research and for industrial applications.

What can I do by OpenFOAM?



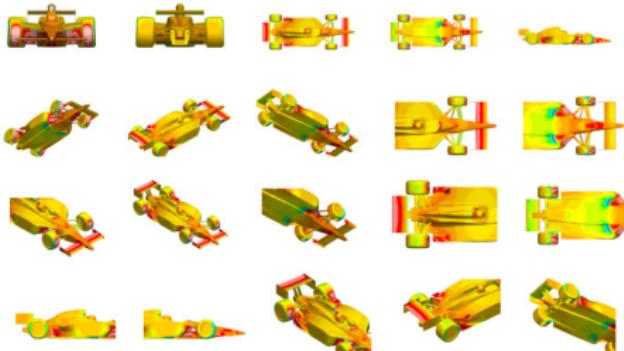
Physical Modelling Capability Highlights:

- **Basic:** Laplace, potential flow, passive scalar/vector/tensor transport
- **Incompressible and compressible flows**
- **Heat transfer:** buoyancy-driven flows, conjugate heat transfer
- **Multiphase flows:**
 - Euler-Euler
 - Volume of Fluid (VoF)
 - free surface capturing and surface tracking
 - Lagrangian (liquid and solids)
- RANS and LES **turbulence modeling**
- **Combustion** modeling
- **Wall-film modeling**
- **Multi-physics:** fluid-structure interaction

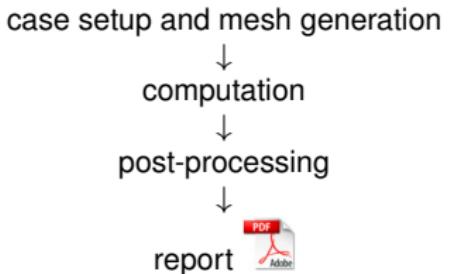
What can I do by OpenFOAM?



EXTERNAL AERODYNAMICS



FULLY-AUTOMATIC WORKFLOW:

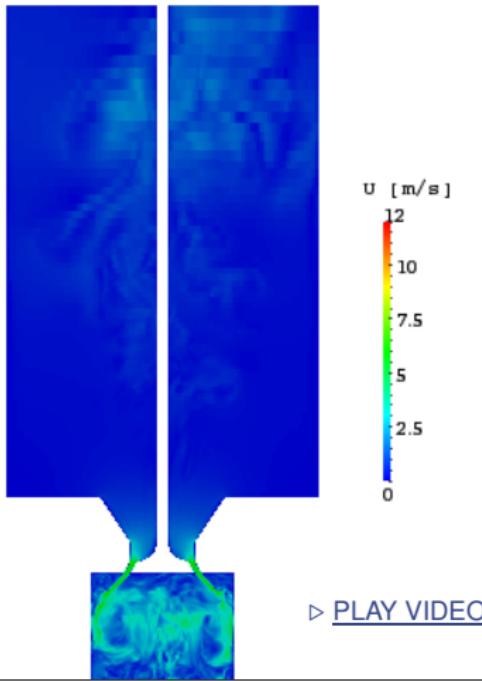


```
//-----  
// Using GLX wrapper... yes  
//-----  
plotResiduals: true;  
averageCoeffs: false;  
saveCuttingPlanes: yes;  
saveWallFields: no;  
writeReport: yes;  
genCuttingPlanes: yes;  
calcFunctionObjects: on;  
calcWallFields: yes;  
writeReport: yes;  
[I] startXCoord =  
[I] endXCoord =  
[I] carBodyFields = cp wallShearStress yPlus  
[I] cuttingPlanesFields = helicity cp cpTot U  
[I] ...  
//-----
```

What can I do by OpenFOAM?



Simulation of TURBULENCE



Features:

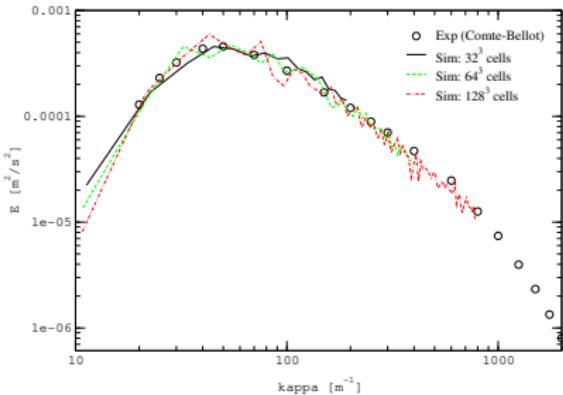
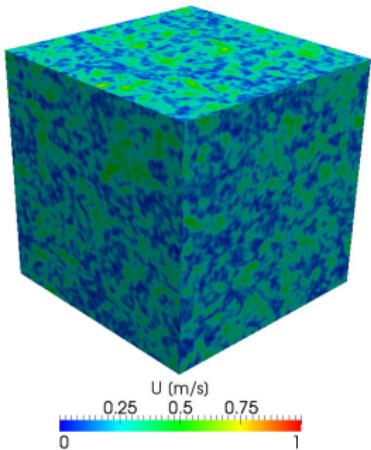
- algorithms for POD
- dynamic mesh
- fully hexahedral mesh
- compressible WALE
- topological changes

A. Montorfano, F. Piscaglia, M. Schmitt et al. "Comparison of Direct and Large Eddy Simulations of the Turbulent Flow in a Valve/Piston Assembly", *Flow Turbulence Combust* (2015) 95:461–480. DOI 10.1007/s10494-015-9620-6

What can I do by OpenFOAM?

Simulation of TURBULENCE

Evaluation of synthetic turbulence generation methods to study how far any unphysical spectrum generated by the methods penetrates into a flow field before it relaxes on the spectrum.



Picture kindly provided by *Dirk Dietzel* (advisor: Prof. C. Hasse), Freiberg University of Technology.

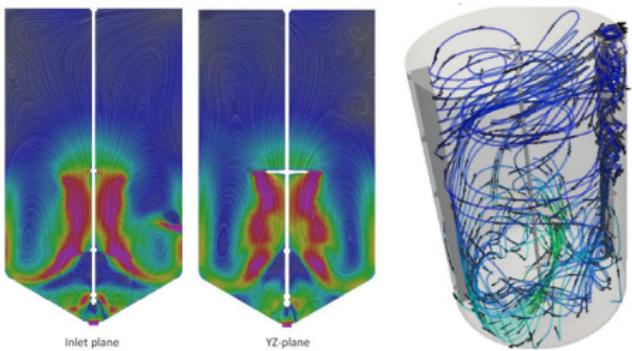
D. Dietzel, D. Messig, C. Hasse, **F. Piscaglia, A. Montorfano**, G. Olinek, O. Stein, A. Kronenburg. *Evaluation of scale resolving turbulence generation methods for LES of turbulent flows*. Computer and Fluids, 2014

What can I do by OpenFOAM?

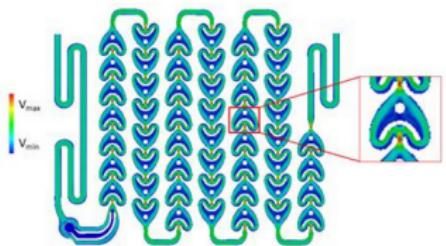


CHEMISTRY

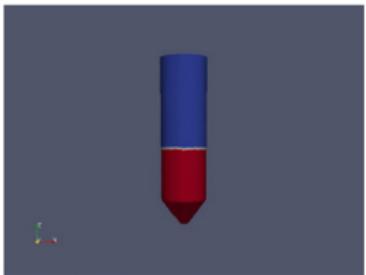
Flow simulation in a mixer-vessel



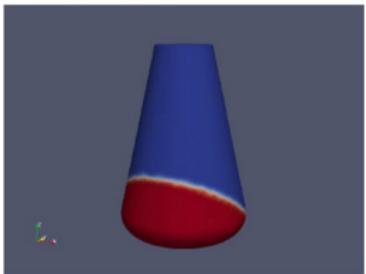
Gas-Liquid
and Liquid-Liquid Flow
in An
Advanced-Flow
Reactor
(AFR)



Bio-reactor modeling



20mL, 150rpm, 50mm



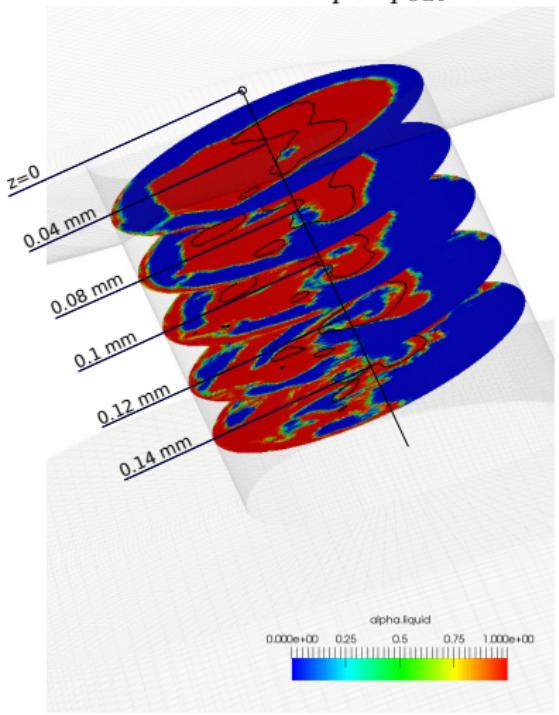
100mL, 120rpm, 50mm

What can I do by OpenFOAM?

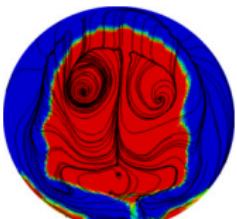


MULTIPHASE FLOWS - VOF

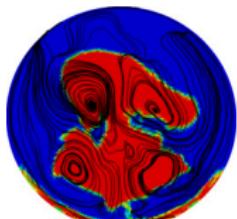
Contour line at $p = p_{sat}$



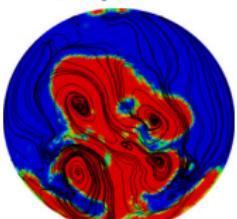
$z = 0.04 \text{ mm}$



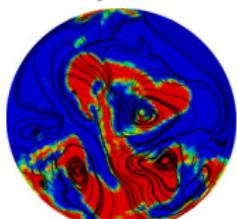
$z = 0.08 \text{ mm}$



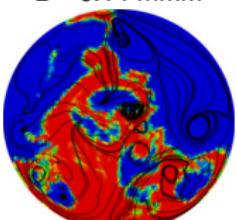
$z = 0.1 \text{ mm}$



$z = 0.12 \text{ mm}$



$z = 0.14 \text{ mm}$

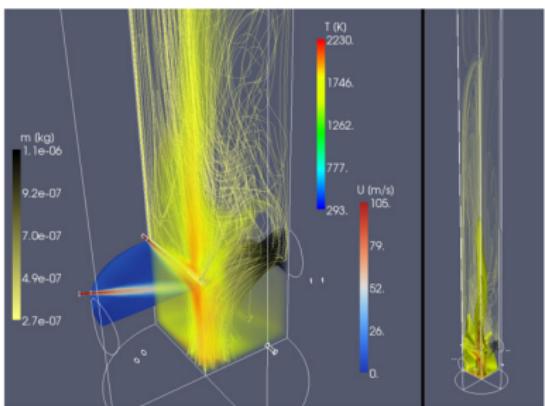
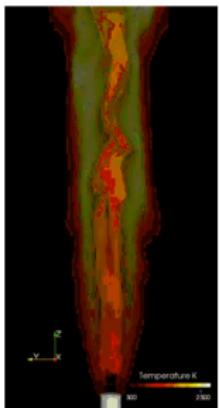


What can I do by OpenFOAM?

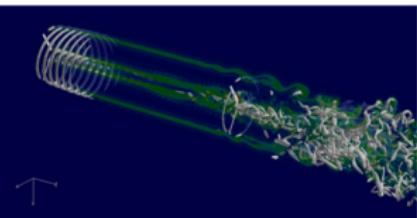
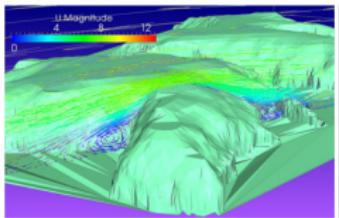
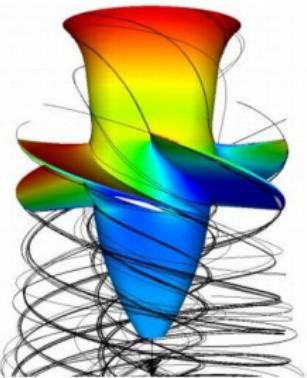


ENERGY SYSTEMS

Coal combustion and gasification

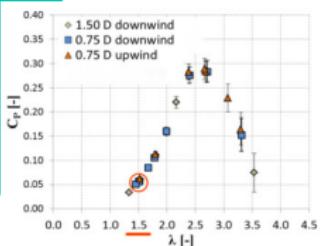
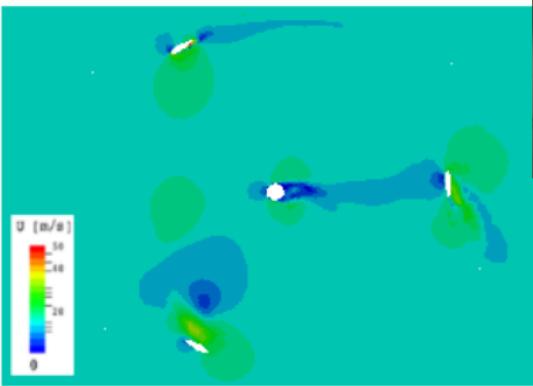


Turbomachinery



Wind energy analysis
(wind field and flow
in turbines)

What can I do by OpenFOAM?



Vertical Axis Wind Turbine (experiments conducted by Turbomachinery Group, PoliMi)

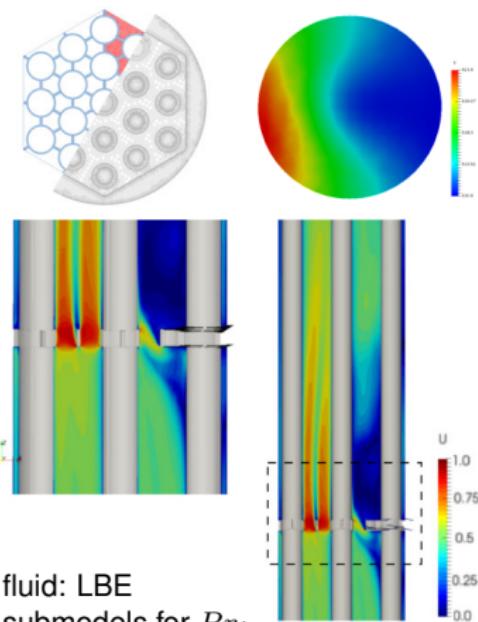
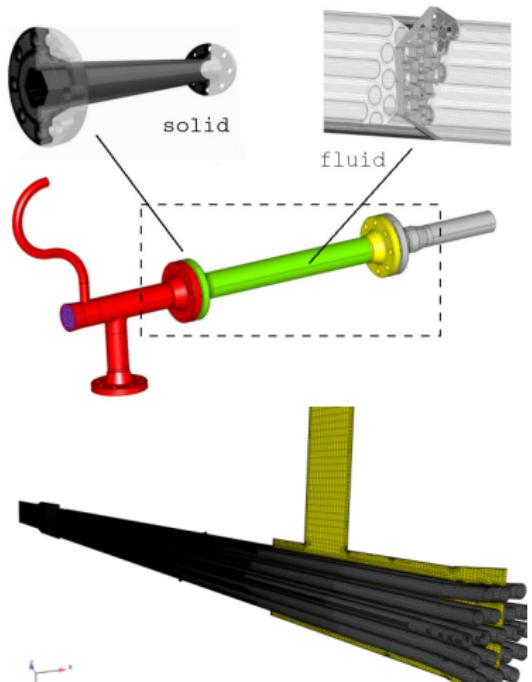
Example of application: H-type Darrieus turbine

- Full-scale simulation with dynamic mesh (AMI)
- Comparison with experiments (global coeffs)
- Transient simulation of turbine start up

A. Montorfano, F. Piscaglia et al. "Application of a Dynamic Model with length scale-dependent RANS/LES hybrid functioning to a Wind Turbine Simulation". **Third Symposium on OpenFOAM® in Wind Energy, 2016**

What can I do by OpenFOAM?

CONJUGATE HEAT TRANSFER

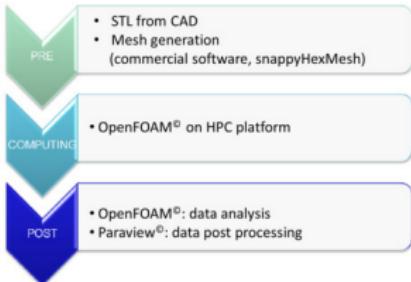
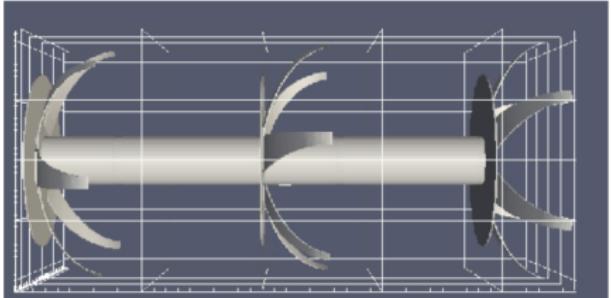


- fluid: LBE
- submodels for Pr_t

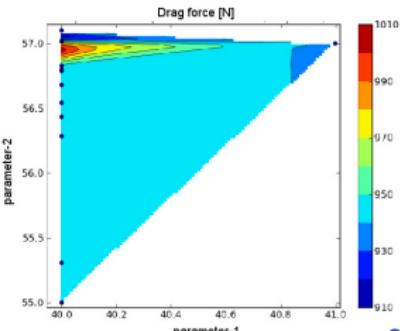
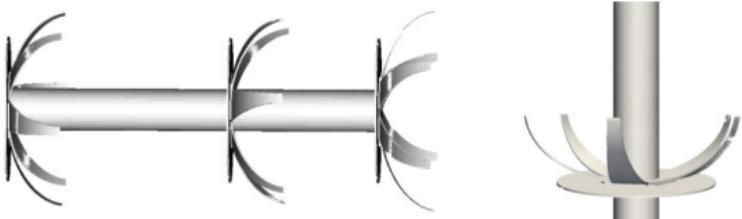
"CFD Analyses of the Internal Blockage in the NACIE-up fuel pin bundle simulator". The 17th Int. Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-17), Xi'an, China, 2017.

What can I do by OpenFOAM?

SHAPE OPTIMIZATION (OpenFOAM+Dakota)



- Optimization loop to find the geometric configuration maximizing the drag of the device, used to monitor pipe systems with high pressure Methane
- **Parametric optimization is performed by coupling OpenFOAM® with DAKOTA**, an open-source Multilevel Parallel Object-Oriented Framework for design optimization



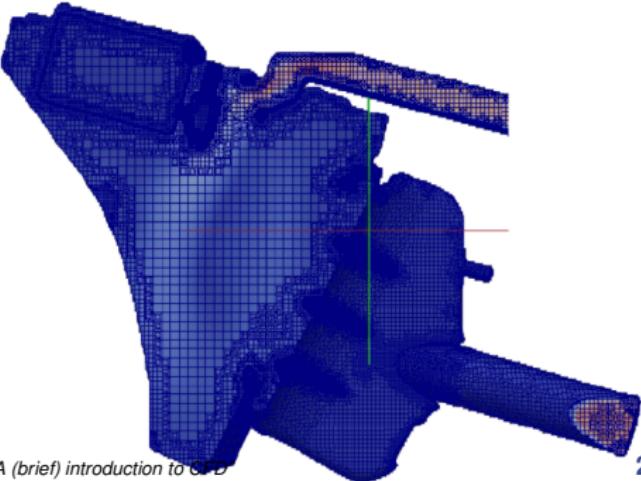
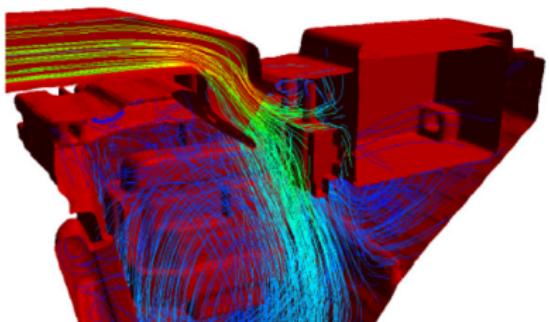
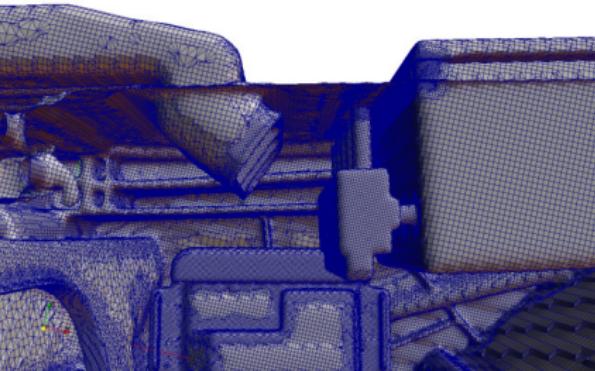
What can I do by OpenFOAM?



SHAPE OPTIMIZATION (OpenFOAM+Dakota)

snappyHexMesh

- The FVM method allows polyhedral support: fewer cells per volume, minimal distortion,near-wall layers
- Avoiding user-interaction: **reliable automatic meshing**

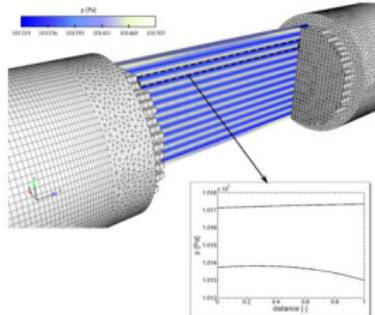
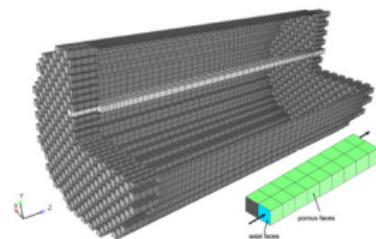
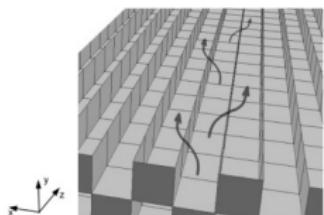


What can I do by OpenFOAM?

AFTERTREATMENT MODELING

dpfFoam solver for compressible flows through porous media:

- Explicit staggered and parallel porous solver with friction model
- New internal face condition (`pressureJump`) to model:
 - steady-state propagation of a sudden finite change in flow properties
 - thin membranes with known velocity/pressure-drop characteristics, by the implementation of the Darcy law
- Implementation of the transport equations for soot, filtration and deposition model
- automatic mesh generation and case setup
- **Validation** against experiments^{1,2}



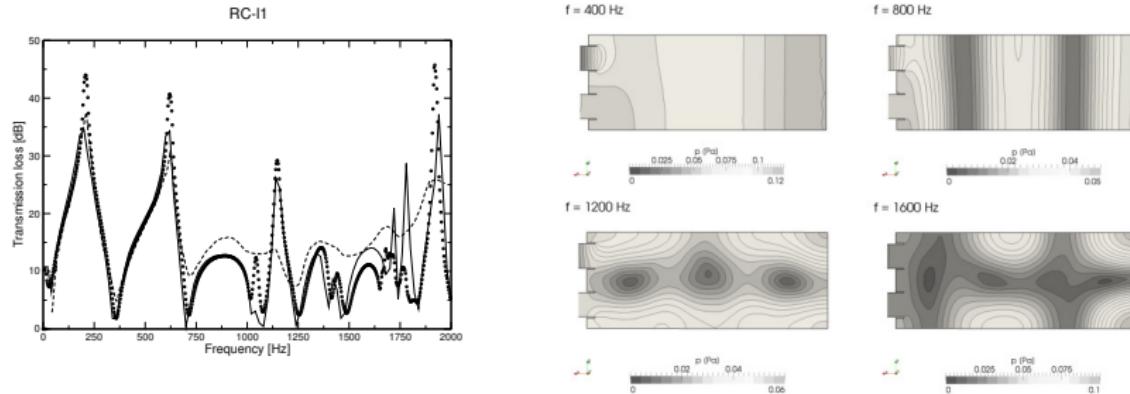
¹ F. Piscaglia, A. Montorfano et al. **SAE Technical Paper 2009-01-1965**

² F. Piscaglia, A. Montorfano et al. **SAE Technical Paper 2009-24-0137**
051176 - Computational Techniques for Thermochemical Propulsion - "A (brief) introduction to CFD"

What can I do by OpenFOAM?



NON-LINEAR ACOUSTICS



- **Implementation of a true non-reflecting outlet based on the NSCBC theory:** variables are computed on the boundaries by solving the conservation equations as in the inner domain
- absence of reflection is enforced by correcting the amplitude of the ingoing characteristic (wave reflected by the boundary)

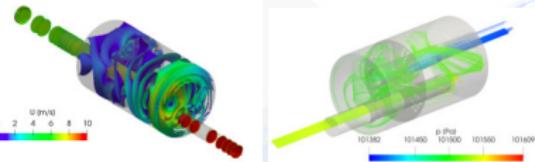
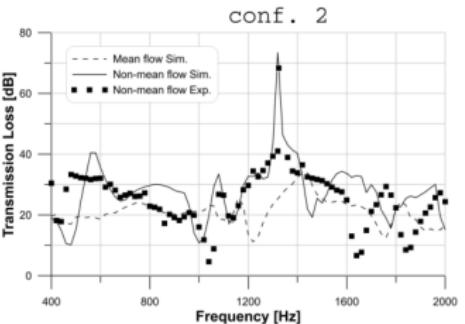
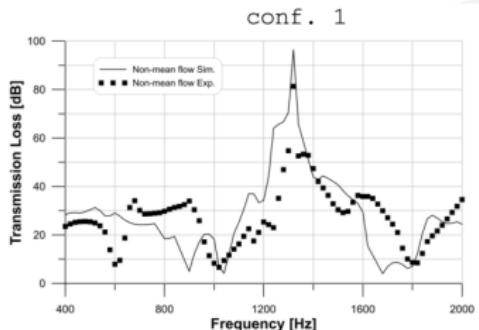
F. Piscaglia, A. Montorfano et al. "Development of a non-reflecting boundary condition for multi-dimensional non-linear duct acoustic computation", **Journal of Sound and Vibration**, Volume 332, Issue 4, Pages 922-935, ISSN 0022-460X, 10.1016/j.jsv.2012.09.030.

What can I do by OpenFOAM?



NON-LINEAR ACOUSTICS/HEAT TRANSFER

Low-temperature energy-recovery from the muffler. Influence of the heat transfer on the acoustic performance of the muffler of a passenger car ([LOGAN](#)).

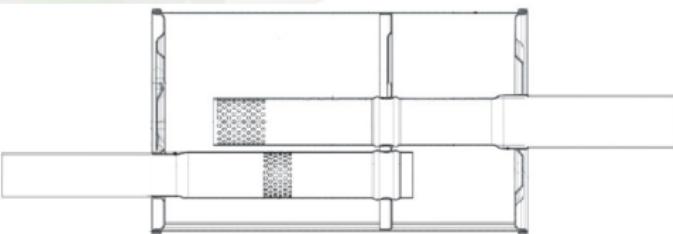


Collaborative work with:

Prof. Andres Agudelo
MSc. Student Manuel Echeverri
From Universidad de Antioquia (Colombia)
Research Group: GIMEL



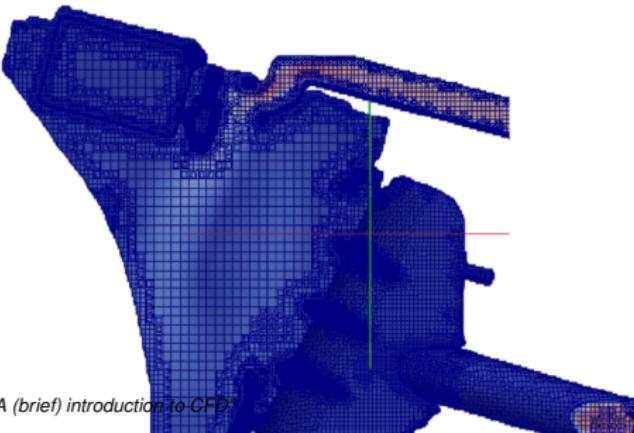
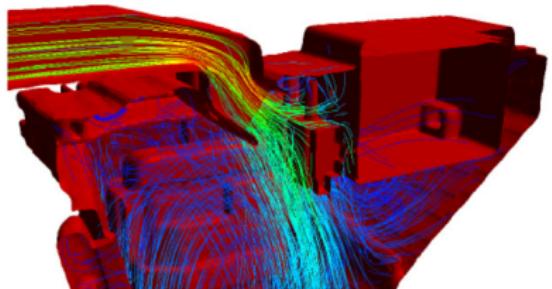
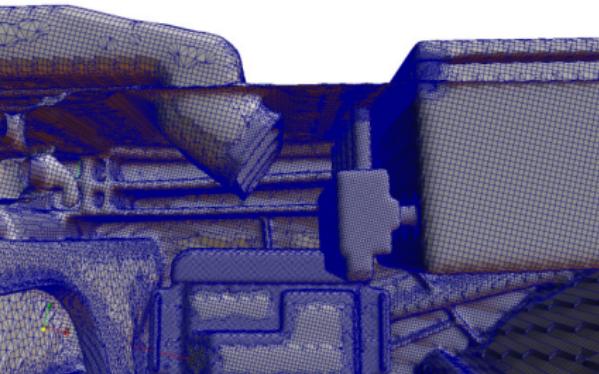
UNIVERSIDAD
DE ANTIOQUIA



What can I do by OpenFOAM?

AUTOMATIC MESHING

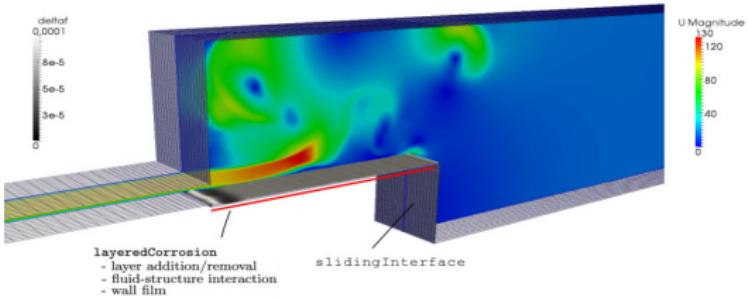
- The FVM method allows polyhedral support: fewer cells per volume, minimal distortion, near-wall layers, richer connectivity
- Polyhedral cell support, combined with automatic mesh motion and topological changes gives a state-of-the-art mesh handling module
- Avoiding user-interaction:
reliable automatic meshing



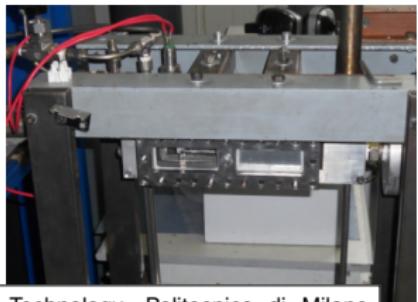
What can I do by OpenFOAM?

LIQUEFYING HYBRID PROPELLANTS

chromoFoam (*Combustion of Hybrid ROcket MOtors*): a PIMPLE (merged PISO+SIMPLE) solver with dynamic mesh and layering for solid propellant and turbulent diffusion flames with reacting Lagrangian parcels, surface film and pyrolysis modeling.



- **Dynamic mesh modeling** of the propellant corrosion driven by surface regression rate
- **LES turbulence modeling** of the fluid flow
- **Surface-film modeling** (propellant gasification, droplet entrainment)
- **Heat-transfer modeling** at the propellant's interfaces (fluid-liquid-liquid-solid)



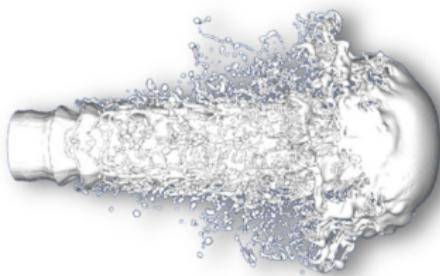
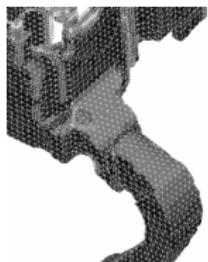
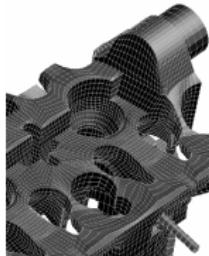
SPLab (Space Propulsion Laboratory), Dept. of Aerospace Science and Technology, Politecnico di Milano
(Prof. L. Galfetti)

... and a lot more!

Tool for the class: OpenFOAM®



OpenFOAM® is an **open source, freely available CFD Toolbox**, licensed under the GNU General Public Licence, written in highly efficient **C++ object-oriented programming**. It uses the **finite volume approach** to solve systems of partial differential equations ascribed on **any 3D unstructured mesh of polyhedral cells**. **Domain decomposition parallelism is integrated at a low level** so that the implementation of parallel solvers can be performed without the need for any “parallel-specific” coding.



Objective: open source implementation of existing knowledge and an object-oriented platform for easy and collaborative future development

1. Completely open software platform using **object-oriented design**
 2. Extensive modelling capabilities in library form: **component re-use**
 3. Collaborative and project-driven model development
- This furthers the **research and collaboration** by removing proprietary software issues: **source code and algorithmic details available to all**

OpenFOAM® is an open source platform for CFD and complex physics simulations, based on fundamental ideas of object orientation, layered software design and equation mimicking

MAIN FEATURES:

1. Implementing complex physical models through equation mimicking
2. OpenFOAM®: Object-oriented software for Computational Continuum Mechanics
3. Layered software development and open source collaboration platform

Basic Components

- Scalars, vectors and tensors with algebra
- Computational mesh; mesh motion, adaptive refinement, topological changes
- Fields (scalar, vector, tensor) and boundary conditions: Dirichlet, Neumann etc.
- Sparse matrix support with linear solver technology

Discretisation Classes

- Implemented as interpolation, differentiation and discretisation operators
- All discretisation methods use identical basic components, e.g. common mesh and matrix support. Better testing and more compact software implementation

Physical Modelling Libraries and Top-Level Solvers

- Libraries encapsulate interchangeable models answering to a common interfaces
- Models implement the interface functions, isolated with run-time selection
- Custom-written and optimised top-level solvers for class of physics

Utilities

- Common functionality is needed for most simulation methods

Top-Level Solver Implementation



Models can be implemented accordingly to the object-oriented structure of the code, so that **partial differential equations** are expressed in their **natural language**:

```
label Eqn = solve
(
    fvm::div(phi,U)
 - fvm::Sp(fvc::div(phi), U)
 + turbulence->divRhoR(U)
 ==
 gradP
 + source
)
```

Object orientation

- **Correspondence** between the **implementation** and the **original equation** is clear
- Each model answers the interface of its class, but its implementation is separated and independent of other models
- Model implementation and inter-equation coupling can be examined independently from the related numerical issues
- New components do not disturb existing code

Application Development in OpenFOAM®

- Custom-written top-level solvers are written for each class of physics
- Writing top-level code is very similar to manipulating the equations
- Ultimate user-coding capabilities: components can be re-used to handle most problems in computational continuum mechanics

Layered Development

- Design encourages code **re-use: developing shared tools**
- Classes and functional components developed and tested in isolation
 - Vectors, tensors and field algebra
 - Mesh handling, refinement, mesh motion, topological changes
 - Discretisation, boundary conditions
 - Matrices and linear solver technology
 - Physics by segment in library form
- Library level mesh, pre-, post- and data handling utilities
- Model-to-model interaction handled through common interfaces
- **New components do not disturb existing code: fewer bugs**

Thank you for your attention!

contact: federico.piscaglia@polimi.it