Disclaimer

“This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.”

Introductory OpenFOAM® Course

From 8^{th} to 12^{th} July, 2013

University of Genoa, DICCA

Dipartimento di **I**ngegneria **C**ivile, **C**himica e **A**mbientale

UNIVERSITÀ DEGLI STUDI
DI GENOVA

Your Lecturer

Joel GUERRERO

joel.guerrero@unige.it

UNIVERSITÀ DEGLI STUDI
DI GENOVA

guerrero@wolfdynamics.com

wolf dynamics

Acknowledgements

These slides and the tutorials presented are based upon personal experience, OpenFOAM® source code, OpenFOAM® user guide, OpenFOAM® programmer's guide, and presentations from previous OpenFOAM® training sessions and OpenFOAM® workshops.

We gratefully acknowledge the following OpenFOAM® users for their consent to use their material:

- Hrvoje Jasak. Wikki Ltd.
- Hakan Nilsson. Department of Applied Mechanics, Chalmers University of Technology.
- Eric Paterson. Applied Research Laboratory Professor of Mechanical Engineering, Pennsylvania State University.

Today's lecture

1. Extending OpenFOAM® capabilities
    - PyFoam
    - swak4Foam
    - setDiscreteFields
2. OpenFOAM-1.6-ext
3. More applications
4. FiPy

Today’s lecture

1. Extending OpenFOAM® capabilities
   - PyFoam
   - swak4Foam
   - setDiscreteFields
2. OpenFOAM-1.6-ext
3. More applications
4. FiPy

Extending OpenFOAM® capabilities

The following utilities might come in handy:

PyFoam

A python library to control OpenFOAM® runs and manipulate OpenFOAM® data. Comes with a number of utilities that should make your life easier.

swak4Foam

A library that combines the functionality of groovyBC (a library to implement user defined boundary conditions without programming) and funkySetFields (an utility that sets the initial value of a scalar or a vector field without programming).

setDiscreteFields

This utility can set profiles of any vector/scalar fields at specified boundary patches by using discrete profile data.

Extending OpenFOAM® capabilities

For source code download and installation instructions go to:

PyFoam

http://openfoamwiki.net/index.php/Contrib_PyFoam

http://openfoamwiki.net/index.php/Contrib_PyFoam

swak4foam

http://openfoamwiki.net/index.php/Contrib/swak4Foam

http://openfoamwiki.net/index.php/Contrib/swak4Foam

setDiscreteFields

http://openfoamwiki.net/index.php/Contrib_setDiscreteFields

http://openfoamwiki.net/index.php/Contrib_setDiscreteFields

I highly advice you to install these utilities (and those that you might consider handy) in your user directory **$WM_PROJECT_USER_DIR.**

Today's lecture

1. ~~Extending OpenFOAM® capabilities~~
    - ~~PyFoam~~
    - ~~swak4Foam~~
    - ~~setDiscreteFields~~
2. **OpenFOAM-1.6-ext**
3. More applications
4. FiPy

OpenFOAM-1.6-ext. Extend Project

OpenFOAM-1.6-ext

http://www.extend-project.de/

http://www.extend-project.de/

The Extend-Project is a Community-driven Release of OpenFOAM®

The goal of the Extend-Project is to open the OpenFOAM® CFD toolbox to community contributed extensions in the spirit of the Open Source development model.

OpenFOAM-1.6-ext. Extend Project

- The official OpenFOAM® version is exclusively maintained by OpenCFD Ltd, and does not incorporate user or community contributions.
- The OpenFOAM-extend project is maintained by Wikki Ltd (http://www.wikki.co.uk/) and is totally driven by the community.
- OpenFOAM-1.6-ext is compatible with the OpenFOAM-1.6.x and 1.7.1 versions of the code and incorporate most developments and changes from the above versions and ensure top-level compatibility.
- In some cases, the differences between the official release and the extended release are caused by bug fixes and algorithms improvements, considered more important than inter-operability.

OpenFOAM-1.6-ext. Extend Project

- The OpenFOAM-extend project is open to community contributions, none of which (or in rarely cases) get adapted into the official OpenFOAM® version. Therefore the extended version has a vast library of capabilities unmatched by the official OpenFOAM® project, including:

  - Improvements in accuracy and stability on tetrahedral and tet-dominant meshes.
  - New discretization schemes.
  - Implicit coupled multi-domain solver.
  - Block-implicit multi-equation matrix support.
  - Proper Orthogonal Decomposition (POD) data analysis tools.
  - Dynamic remeshing classes, based on tetrahedral edge swapping.
  - Radial Basis Function interpolation and mesh motion classes.
  - Turbomachinery features: GGI interface, cyclic GGI, partial overlap GGI.

OpenFOAM-1.6-ext. Extend Project

- The OpenFOAM-extend project is open to community contributions, none of which (or in rarely cases) get adapted into the official OpenFOAM® version. Therefore the extended version has a vast library of capabilities unmatched by the official OpenFOAM® project, including:

  - Overlapping grids support.
  - Basic implementation of OpenMP wrapping for multi-core support.
  - Support to GPU computing.
  - Finite area method and liquid film support .
  - Immersed boundary method (IBM).
  - IC engine-related developments
  - Solid stress analysis, contact stress, plasticity, crack propagation, fluid-structure interaction.

… among others.

Today's lecture

1. ~~Extending OpenFOAM® capabilities~~
    - ~~PyFoam~~
    - ~~swak4Foam~~
    - ~~setDiscreteFields~~
2. ~~OpenFOAM-1.6-ext~~
3. **More applications**
4. FiPy

More applications

Optimization in OpenFOAM® using DAKOTA

Recently, I started to work with optimization in **OpenFOAM®**, for this I am using the DAKOTA toolkit.

You can download DAKOTA toolkit at the following link:

http://dakota.sandia.gov/

More applications

DAKOTA in a nutshell

Design and **A**nalysis tool**K**it for **O**ptimization and **T**erascale **A**pplications

DAKOTA is a general-purpose software toolkit for performing systems analysis and design on high performance computers. DAKOTA provides algorithms for design optimization, uncertainty quantification, parameter estimation, design of experiments, and sensitivity analysis, as well as a range of parallel computing and simulation interfacing services.

DAKOTA is developed and supported by U.S. Sandia National Labs

Sandia National Laboratories

More applications

DAKOTA in a nutshell

Dakota capabilities

Parameter Studies – Sensitivity Analysis

Design of Experiments (DOE) - Design and Analysis of Computer Experiments (DACE)

Uncertainty Quantification (UQ)

Optimization - Gradient-based, derivative-free local, and global methods

Calibration or Nonlinear Least Squares Capabilities

More applications

Mesh manipulation in OpenFOAM® using Mesquite

Recently, I started to work with mesh manipulation and mesh quality improvement in **OpenFOAM®**, for this I am using the **Mesquite** library.

You can download Mesquite libary from the following link:

http://www.cs.sandia.gov/optimization/knupp/Mesquite.html

If this link does not work you can try

http://www.cs.sandia.gov/web1400/1400_download.html

You will need to register and choose the application you want to download

More applications

Mesh manipulation in OpenFOAM® using Mesquite

Why am I interested in mesh manipulation and mesh quality improvement in OpenFOAM®?

As you may know, OpenFOAM® does not have utilities to improve the mesh quality.   So, in order to avoid going back to the geometry generation and meshing stage to improve or fix the mesh, I am trying to use Mesquite quality metrics and mesh quality improvement algorithms to automatically fix/improve the mesh.

More applications

Mesquite in a nutshell

Mesquite = **Mes**h **Qu**ality **I**mprovement **T**oolkit

- Mesquite is designed to provide a stand-alone, portable, comprehensive suite of mesh quality improvement algorithms.
- Mesquite software design is based on a mathematical framework that improves mesh quality by solving an optimization problem to guide the movement of mesh vertices.
- Mesquite is not intended to be a mesh generation tool. It can serve as a post-processor to a mesh generation procedure, a mesh pre-processor to a non-adaptive simulation code, or as an algorithm for in-core adaptive mesh quality improvement.
- Mesquite is intended to be linked to either a meshing code or to a simulation code.
- Mesquite is developed and supported by U.S. Sandia National Labs

Sandia National Laboratories

Today's lecture

1. ~~Extending OpenFOAM® capabilities~~
    - ~~PyFoam~~
    - ~~swak4Foam~~
    - ~~setDiscreteFields~~
2. ~~OpenFOAM-1.6-ext~~
3. ~~More applications~~
4. **FiPy**

FiPy

FiPy: A Finite Volume PDE Solver Using Python
http://www.ctcms.nist.gov/fipy/
http://www.ctcms.nist.gov/fipy/

This one is not directly related to OpenFOAM®, but it might be useful if you want to learn python and how to program the FVM without the hustle and bustle of OpenFOAM®.

FiPy is an object oriented, partial differential equation (PDE) solver, written in Python, based on a standard finite volume (FV) approach. The framework has been developed in the Metallurgy Division and Center for Theoretical and Computational Materials Science (CTCMS), in the Material Measurement Laboratory (MML) at the National Institute of Standards and Technology (NIST).

Thank you for your attention