OpenFOAM: A User View

Hrvoje Jasak

`h.jasak@wikki.co.uk`

Wikki Ltd, United Kingdom

Background

Objective

- Provide basic information about OpenFOAM software from User's viewpoint
  - What is it?
  - How do I run it?
  - What capabilities does it have?

Topics

- OpenFOAM: Executive Overview
- Running OpenFOAM
  - Structure of an OpenFOAM executable
  - OpenFOAM case: data organisation and management
  - Running the solver and viewing results
- Points of interest: why consider OpenFOAM

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 10,000-CPU CFD license!
- Software under active development, capabilities 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**

OpenFOAM: Executive Overview

Implementing Continuum Models by 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**

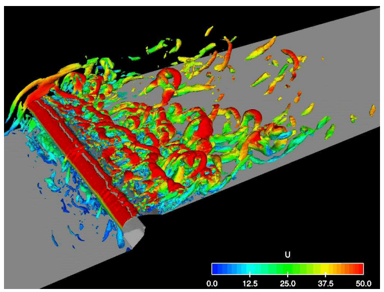
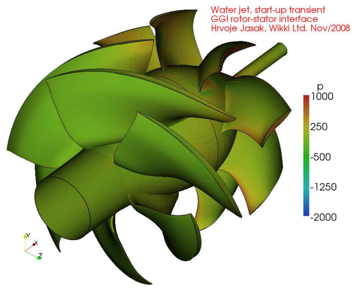
```
solve
(
   fvm::ddt(k)
 + fvm::div(phi, k)
 - fvm::laplacian(nu() + nut, k)
== nut*magSqr(symm(fvc::grad(U)))
 - fvm::Sp(epsilon/k, k)
);
```

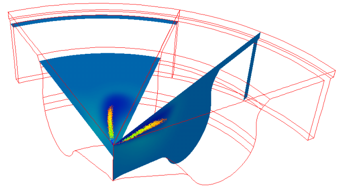
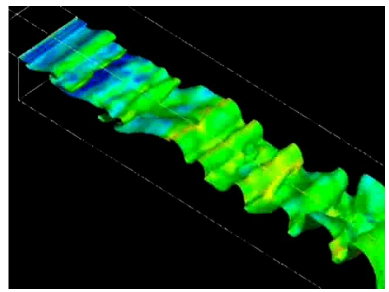
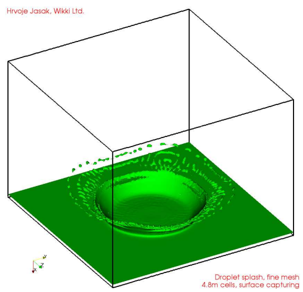
- Correspondence between the implementation and the original equation is clear

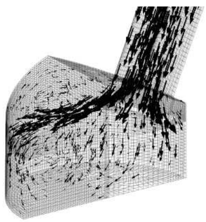
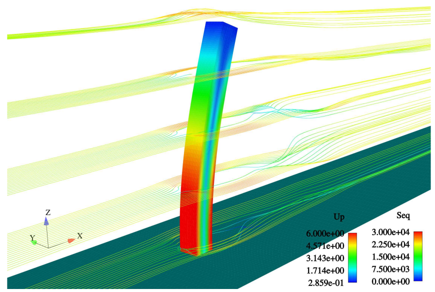
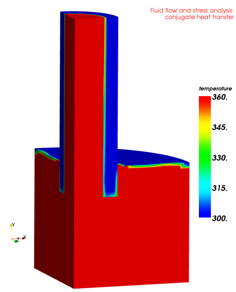
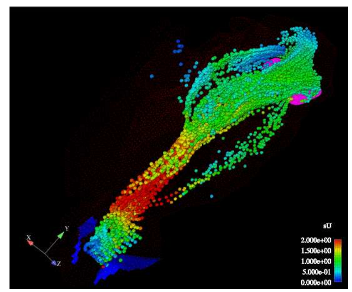
OpenFOAM: Executive Overview

Physical Modelling Capability Highlights

- Basic: Laplace, potential flow, passive scalar/vector/tensor transport
- Incompressible and compressible flow: segregated pressure-based algorithms
- Heat transfer: buoyancy-driven flows, conjugate heat transfer
- Multiphase: Euler-Euler, VOF free surface capturing and surface tracking
- RANS for turbulent flows: 2-equation, RSTM; full LES capability
- Pre-mixed and Diesel combustion, spray and in-cylinder flows
- Stress analysis, fluid-structure interaction, electromagnetics, MHD, etc.

Running OpenFOAM

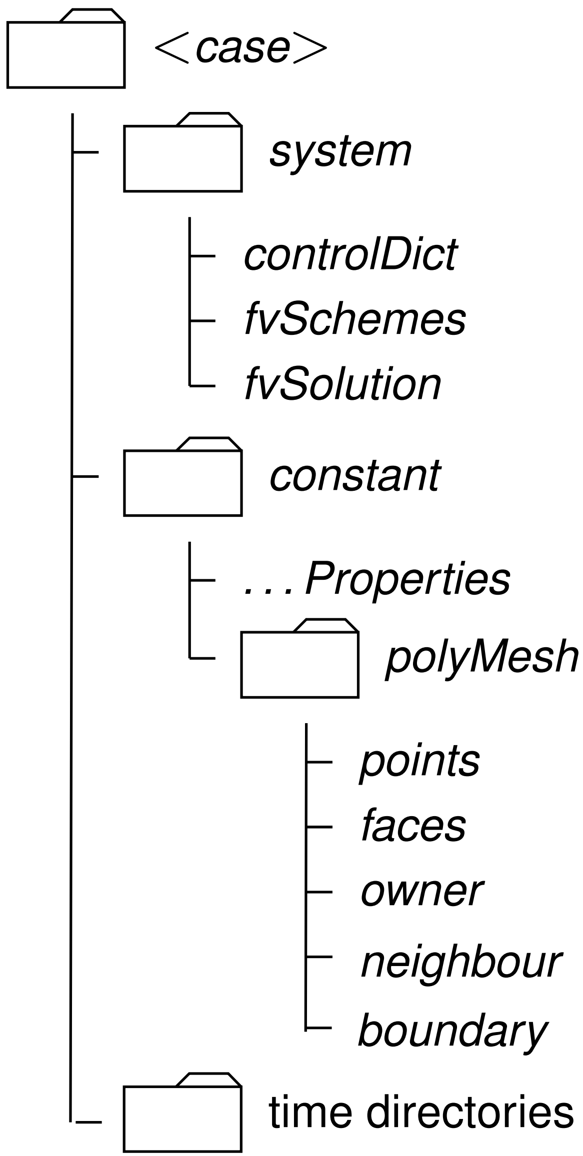
Structure of OpenFOAM

- OpenFOAM is assembled from components
  - Foundation libraries, containing discretisation, mesh handling etc. in re-usable form. Functionality shared across many application
  - Physical modelling libraries: thermo-physical models (liquids and gasses), viscosity models, turbulence models, chemical reactions interface
  - Utilities: mesh import and manipulation, parallel processing, post processor hook-up (reader module) and data manipulation
  - Customised, purpose-written and optimised executables for each physics segment. All are founded on common components
  - Linkage for user extensions and on-the-fly data analysis

OpenFOAM Executable

- Custom executable for specific physics segment: few 100s of lines of code
- Easy to read, understand, modify or add further capability
- Existing code is used as a basis for own development: simulation environment
- Low-level functions, eg. mesh handling, parallelisation, data I/O handled transparently: no need for special coding at top level

Running OpenFOAM



Data Organisation and Management

- Unlike “standard CFD practice”, in OpenFOAM `case` is a **directory**: each self-contained piece of heavy-weight data stored in its own file
- Light-weight data is presented in **dictionary form**: keyword-value pairs in free format. It can be changed and re-read during the run: solution steering
- Mesh data split into components for efficient management of moving mesh cases
- Time directories contain solution and derived fields (one per file)
- Support for compressed I/O: more efficient I/O and less disk space

Running OpenFOAM

Running OpenFOAM Applications

- Currently, each application is a stand-alone executable. Typically, multiple executables are used in a single run (pre, solution, data manipulation, graphics, post-processing). More interactivity on the way!
- On-the-fly data analysis can be built into top-level solver
- **Function object hook-up**: allowing user to add own post- or data-manipulation tools without changing top-level code. Executed once per time-step
- Extensions to functionality typically done using **run-time selection tables**: present on all user-defined choice points

Post-Processing

- Sampling, graphing and integral properties tools present as a part of foundation libraries. Example: sampling in space (line, plane) and time
- Consistency with discretisation: post-processing uses the same algorithms as the solver (eg. gradient calculation)
- Graphical post-processing hook-up: ParaView, Ensight, Fieldview
- “Post-processing” tools may be used in actual CFD, eg. sampling plane LES inlet

Point of Interest

Why Consider OpenFOAM?

- Open architecture
  - Access to complete source: no secret modelling tricks, no cutting corners
  - Both community-based and professional support available
  - Common platform for new R&D projects: shipping results of research into the hands of a customer with no delay
- Low-cost CFD
  - No license cost, portable to any computing platform (IBM Blue Gene)
  - Efficient on massively parallel computers, portable to new comms protocols
- Problem-independent numerics and discretisation
  - Tackle non-standard continuum mechanics problem: looking beyond the capabilities of commercial CFD
- Efficient environment for complex physics problems
  - Tackling difficult physics is made easier through equation mimicking
  - Utility-level tools readily available: parallelism, moving mesh
  - Track record in non-linear and strongly coupled problems
  - Excellent piece of C++ and software engineering! Decent piece of CFD