SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Comparison of FEM with FVM applied to multiphysics cases - I

Introduction to OpenFOAM Finite Volume Toolbox

Raffaele Ponzini, CINECA

07/21

Univerza *v Ljubljani*

TECHNISCHE UNIVERSITÄT WIEN

CINECA consorzio interuniversitario

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Contents

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

Practical Introduction to OpenFOAM Finite Volume Toolbox based on my experience:

What is OpenFOAM

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- OpenFOAM was created by Henry Weller in 1989 under the name “FOAM” and was released open source as “OpenFOAM” by Henry Weller, Chris Greenshields and Mattijs Janssens in December 2004.
- OpenFOAM is today the standard de-facto, open source software for computational fluid dynamics (CFD).
- The OpenFOAM Foundation distributes OpenFOAM exclusively under the General Public License (GPL). The GPL gives users the freedom to modify and redistribute the software and a guarantee of continued free use, within the terms of the license.
- The current version is 8. In 2014, the development line of OpenFOAM, known as “OpenFOAM-dev” was released publicly on GitHub.

Classic CFD cycle

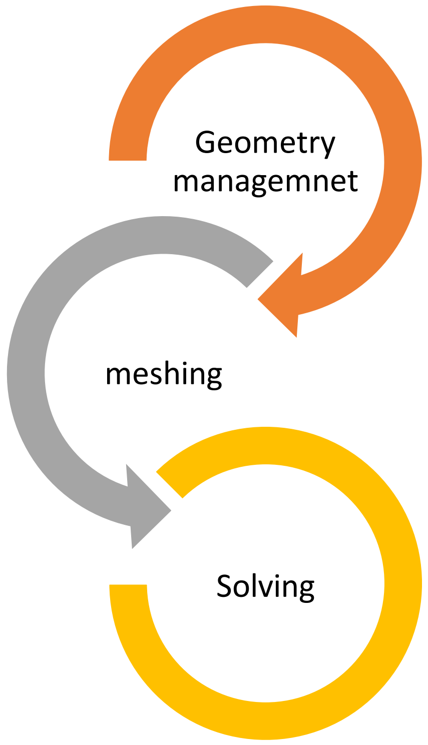
SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- Computational Fluid Dynamics (CFD) typical working cycle is made of a set of steps or elementary bricks that are equally relevant to allow engineers to perform their day by day activity:
  - Cad/geometry management
  - Meshing
  - Solving
  - Monitoring solution
  - Visualize/analyze results
- OpenFOAM (OF) is a complete toolbox in the sense that contains and embed all the necessary bricks to perform all the steps listed above

Classic CFD cycle

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- Usually, it is important to note that some activities are iterated in a cyclic way like that:
- This is a central point of common CFD workflow and the availability of robust and easy to use tools is essential
- OF is designed to allow the user to efficiently work and perform the necessary activities by means of a set of command lines functions
- Some functions works as stand-alone, other requires dictionary to be correctly executed



Dictionary in OpenFOAM

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- The concept of dictionary in OF is central
- A dictionary in OF is just a txt, human readable, file made of a set of entries (keys) that requires meaningful values in order to enable the correct execution of solvers and functions
- Each dictionary might contain sub-dictionaries each one with its own set of entries.
- If all the necessary entries are not given/defined the desired command or solver will not run and will exit with error
- Part of the complexity or unfriendliness of OF for beginners is related to that model; nevertheless, as we will see in the forthcoming there are *tutorials and templates* to support users setting up running cases with minimal effort

Geometry management

- Geometry input is the basis for every CFD modelling
- To handle geometries OF requires two main kind of triangulated files: Object files (.obj) or stereolithography (.stl) either as binary or ascii format.
- The quality of the triangulated surface is relevant for the quality of the final meshing procedure, but the essential condition is that the geometry is watertight. This condition is not strictly mandatory but it is more convenient to start with a watertight geometry to avoid meshing errors and/or wasting of time.
- OF have a set of command line to manage the geometry input but does not contains a geometry modeler

Geometry management

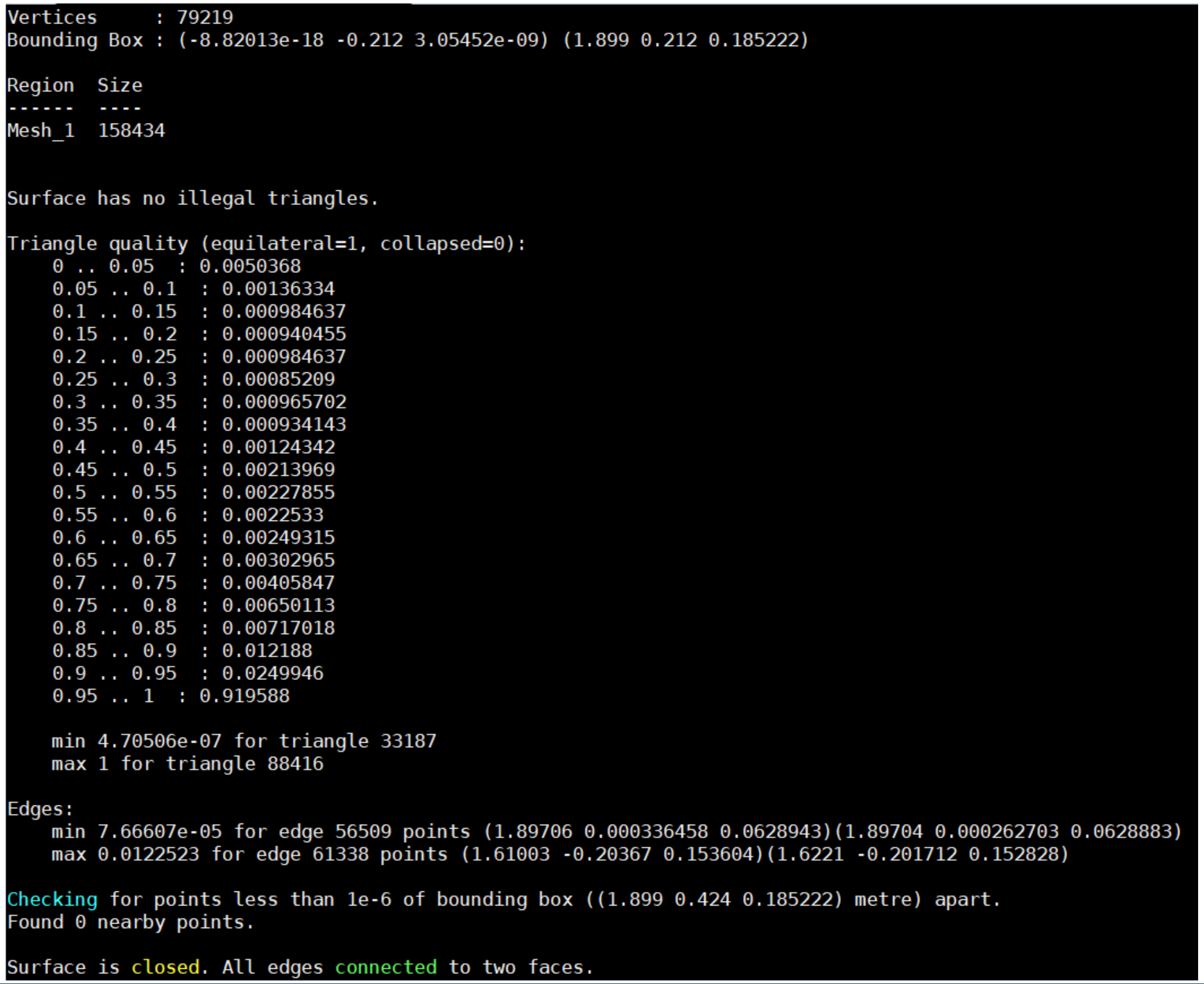
- There are several valuable open-source geometry modeler that can be used to design geometries from scratch or to modify/simplify incoming 3D CAD files: Salome, FreeCAD, Onshape, Blender among the others
- OF have instead a set of command line tools to:
  - Evaluate the quality of the input geometry file
  - Translate/Rotate/Scale the geometry dimensions
  - Orient normal directions

Geometry management

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- *surfaceCheck [OPTIONS] <surface file>*

- *Relevant info are:*
  - *Bounding box: you can check the main dimensions*
  - *Quality of triangulated surface*
  - *Opening/closeness of the geometry*



```
Vertices     : 79219
Bounding Box : (-8.82013e-18 -0.212 3.05452e-09) (1.899 0.212 0.185222)

Region  Size
------  ----
Mesh_1  158434


Surface has no illegal triangles.

Triangle quality (equilateral=1, collapsed=0):
    0 .. 0.05  : 0.0050368
    0.05 .. 0.1  : 0.00136334
    0.1 .. 0.15  : 0.000984637
    0.15 .. 0.2  : 0.000940455
    0.2 .. 0.25  : 0.000984637
    0.25 .. 0.3  : 0.00085209
    0.3 .. 0.35  : 0.000965702
    0.35 .. 0.4  : 0.000934143
    0.4 .. 0.45  : 0.00124342
    0.45 .. 0.5  : 0.00213969
    0.5 .. 0.55  : 0.00227855
    0.55 .. 0.6  : 0.0022533
    0.6 .. 0.65  : 0.00249315
    0.65 .. 0.7  : 0.00302965
    0.7 .. 0.75  : 0.00405847
    0.75 .. 0.8  : 0.00650113
    0.8 .. 0.85  : 0.00717018
    0.85 .. 0.9  : 0.012188
    0.9 .. 0.95  : 0.0249946
    0.95 .. 1  : 0.919588

    min 4.70506e-07 for triangle 33187
    max 1 for triangle 88416

Edges:
    min 7.66607e-05 for edge 56509 points (1.89706 0.000336458 0.0628943)(1.89704 0.000262703 0.0628883)
    max 0.0122523 for edge 61338 points (1.61003 -0.20367 0.153604)(1.6221 -0.201712 0.152828)

Checking for points less than 1e-6 of bounding box ((1.899 0.424 0.185222) metre) apart.
Found 0 nearby points.

Surface is closed. All edges connected to two faces.
```

Geometry management

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- *surfaceTransformPoints [OPTIONS] <surface file> <output surface file>*

options:

-rollPitchYaw <vector> transform in terms of '(roll pitch yaw)' in degrees

-rotate <(vectorA vectorB)> transform in terms of a rotation between <vectorA> and <vectorB> - eg, '((1 0 0) (0 0 1))'

-scale <vector>   scale by the specified amount - eg, '(0.001 0.001 0.001)' for a uniform [mm] to [m] scaling

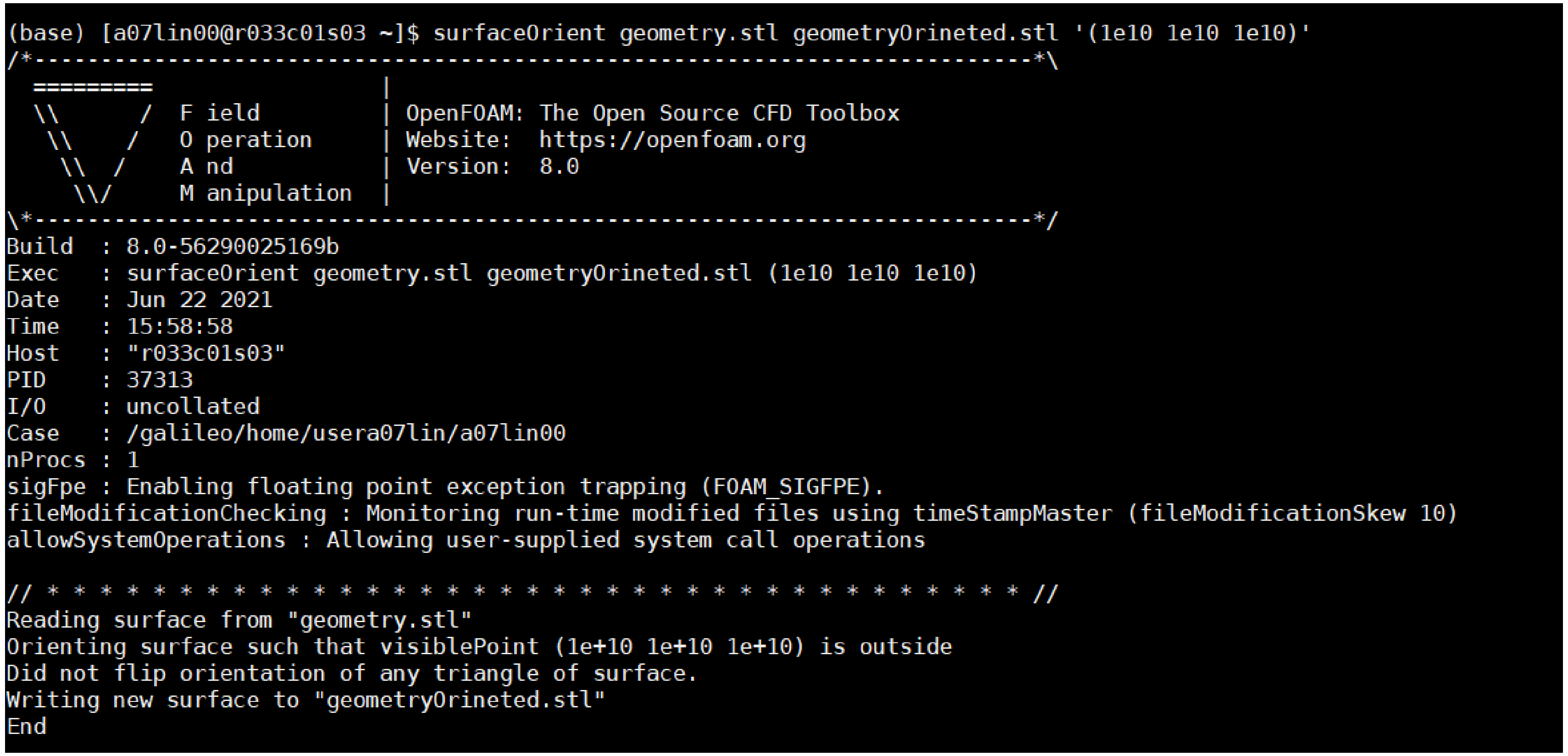
-translate <vector> translate by the specified <vector> - eg, '(1 0 0)'

-yawPitchRoll <vector> transform in terms of '(yaw pitch roll)' in degrees

Geometry management

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- *surfaceOrient [OPTIONS] <surface file> <output surface file> <visiblePoint>:*



```
(base) [a07lin00@r033c01s03 ~]$ surfaceOrient geometry.stl geometryOrineted.stl '(1e10 1e10 1e10)'
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  8.0
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
Build  : 8.0-56290025169b
Exec   : surfaceOrient geometry.stl geometryOrineted.stl (1e10 1e10 1e10)
Date   : Jun 22 2021
Time   : 15:58:58
Host   : "r033c01s03"
PID    : 37313
I/O    : uncollated
Case   : /galileo/home/usera07lin/a07lin00
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Reading surface from "geometry.stl"
Orienting surface such that visiblePoint (1e+10 1e+10 1e+10) is outside
Did not flip orientation of any triangle of surface.
Writing new surface to "geometryOrineted.stl"
End
```

Meshing

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- Once we are fine with geometry definition of the shape we want to study or use to define our problem space, we must produce a valid discretization.
- OF have two main tools to produce high quality meshes for both simple and complex (industrial level) geometries
  - *blockMesh:* fully structured hexahedral mesher
  - *snappyHexMesh:* unstructured hexa-dominant mesher
- Both tools are instructed using the typical OF dictionary based strategy
- Both tools produce high quality meshes

Meshing

- *blockMesh:* is the basis of the meshing tool in OF. It can be used in conjunction with *snappyHexMesh* or as standalone tool.
- Meshes created with blockMesh are usually 100% made of hexahedra even if other mesh cell shape are allowed.
- The usual/average usage of the *blockMesh* tool is to create the background starting mesh of the computational domain before move to *snappyHexMesh* to finalize the mesh.
- For a simple computational domain *blockMesh* is well suited but for industrial complex domain is not suited and requires the usage of *snappyHexMesh* as additional step.

Meshing

- *snappyHexMesh* is the standard OF meshing approach for complex industrial geometries
- As for the *blockMesh* it works by means of a dictionary where the user can instruct several input parameters to handle different parts of the meshing workflow as desired
- The main steps are:
  - **Castellated**: the discretization
  - **Refinement**: the background discretization is refined where needed (usually surfaces or gaps)
  - **Snappage**: starting from the refined castellated mesh a set of algorithms is applied to project the mesh faces into the geometry description as defined by the stl triangulated surfaces
  - **Layering**: once the mesh is defined allover the domain, the user can decide to add a prismatic boundary layer to selected surfaces (typical case of RANS solvers with wall function enabled)

Solvers

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- OF have a wide variety of solvers (about 60) included in the standard release
  - The solvers are divided by physics and then by solver for a specific problem with this logic:

```
├── basic
├── combustion
├── compressible
├── discreteMethods
├── DNS
├── electromagnetics
├── financial
├── heatTransfer
├── incompressible
├── lagrangian
├── multiphase
└── stressAnalysis
```

```
├── basic
│   ├── laplacianFoam
│   ├── potentialFoam
│   └── scalarTransportFoam
├── incompressible
│   ├── adjointShapeOptimizationFoam
│   ├── boundaryFoam
│   ├── icoFoam
│   ├── nonNewtonianIcoFoam
│   ├── pimpleFoam
│   ├── pisoFoam
│   ├── shallowWaterFoam
│   └── simpleFoam
```

Solvers

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

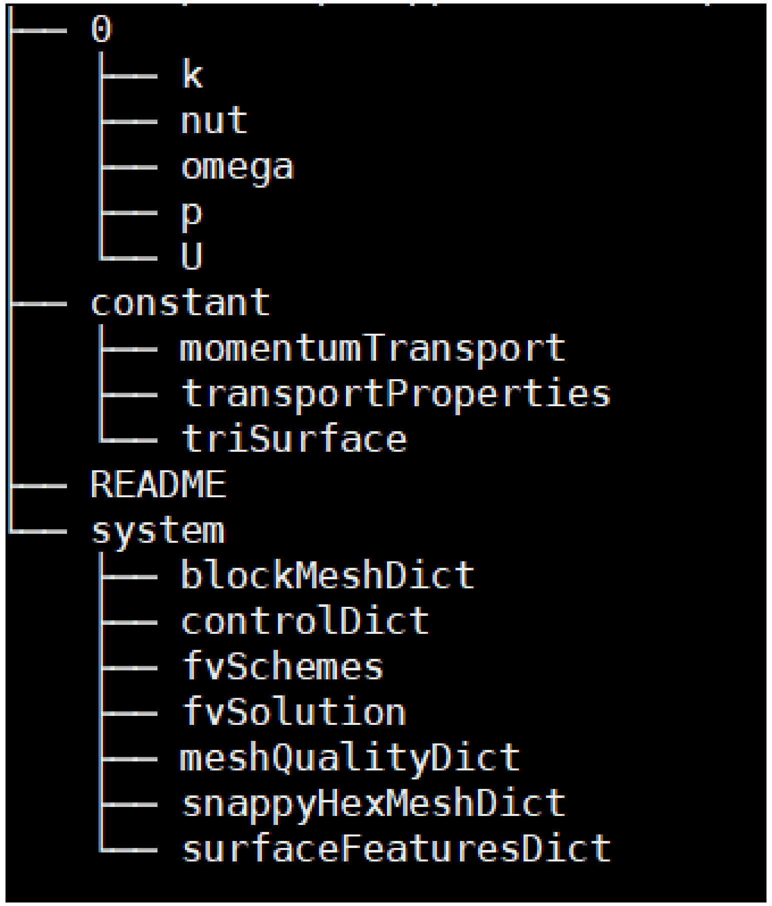
- OF comes also with a wide range of running tutorials
- A very convenient and practical way of doing is to:
  - Identify the physics of your problem
  - Look into the tutorials directories for such physics
  - Dive into the tutorial and try to understand it
  - Run it
  - Modify it your need
- There is also a more general approach based on given templates to solve different physical problems offered into the main distribution of OF under $FOAM_ETC/templates
- The templates can be used similarly to the tutorials to have a clean case base to be personalized
- In both cases (tutorials and templates) you will find a meaningful *README* file or an *AllRun* file that will show how to use the material

Solvers

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- The general case requires a set of mandatory directories o run in OF:
  - system/
  - constant/
  - 0/
- Depending on the solver type and thus on the physics that we want to solve there are a wide variety of required dictionary under these main directories

OpenFOAM-8.0/etc/templates/inflowOutflow/



```
├── 0
│   ├── k
│   ├── nut
│   ├── omega
│   ├── p
│   └── U
├── constant
│   ├── momentumTransport
│   ├── transportProperties
│   └── triSurface
├── README
└── system
    ├── blockMeshDict
    ├── controlDict
    ├── fvSchemes
    ├── fvSolution
    ├── meshQualityDict
    ├── snappyHexMeshDict
    └── surfaceFeaturesDict
```

Data sampling

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- Once the solver is set up an running there are several quantities that the user might want to monitor and or sampling
- OF have a convenient set of *functionObjects* to support a wide range of possible quantity sampling
- The full list is extremely large and you can access it by means of the *postProcess –list* command

Data sampling

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

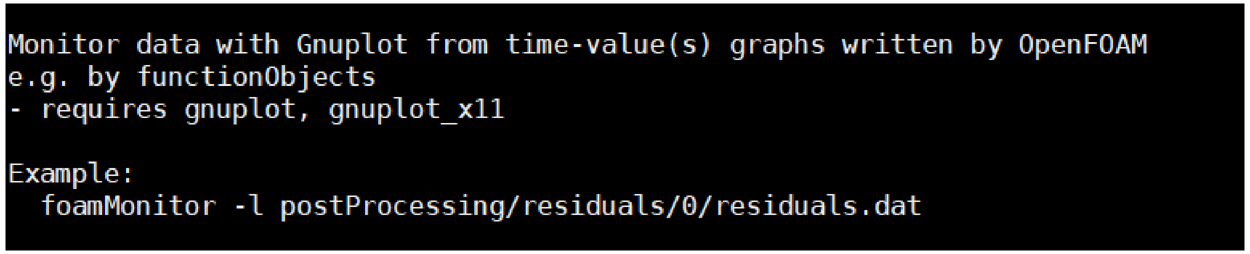
- The starting quantities that should be monitored are the so called *residuals*
- Residuals monitoring allows to get a primary understanding of the convergence of the numerical solution
- In few words residuals are a measure of the local imbalance of a conserved variable in each control volume. This value should tend to zero as the solving procedure evolves
- If residuals ‘diverges’ the numerical solution is not reliable
- If residuals ‘converges’ then form the numerical point of view we are fine and the physics must be checked

Data sampling

- Other quantities that should be monitored to have a correct understanding of the physical meaning of the solution are the values of computed quantities at boundaries
- Usually when for instance we are imposing velocity values at a given boundary we want to monitor the computed pressure value and vice-versa if we prescribe pressure.
- Other physically meaningful quantity for a given problem should be sampled, for instance forces acting on a given bluff body in the case of external aerodynamics problem

Plotting during calculations

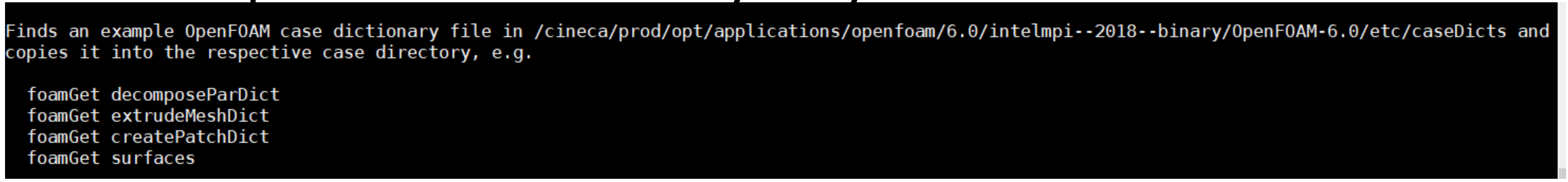
- Once data are sampled and monitored during calculation we ar able t get a plot of that quantities
- A quick way of doing is using the *foamMonitor* function:



```
Monitor data with Gnuplot from time-value(s) graphs written by OpenFOAM
e.g. by functionObjects
- requires gnuplot, gnuplot_x11

Example:
  foamMonitor -l postProcessing/residuals/0/residuals.dat
```

- In OF is equally simple sample planes, point probes, line probes and numbers of relevant quantities.
- In order to add monitored quantities in an easy way we can use the *foamGet* function



```
Finds an example OpenFOAM case dictionary file in /cineca/prod/opt/applications/openfoam/6.0/intelmpi--2018--binary/OpenFOAM-6.0/etc/caseDicts and
copies it into the respective case directory, e.g.

  foamGet decomposeParDict
  foamGet extrudeMeshDict
  foamGet createPatchDict
  foamGet surfaces
```

Output Visualization

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- The standard way to have a look at your solution is using the open-source viewer *paraFoam* that is and extension of the more well-known viewer Paraview developed by Kitware.
- Despite a set of convenient features available in *paraFoam* to support the OF user in visualizing the mesh you can obtain the same kind of visualization using the standard *Paraview* viewer.
- I will refer to *Paraview* in the forthcoming.

Output Visualization

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- In order to read and visualize an OF case output we can either read the *view.foam* file, i.e. reading the native OpenFOAM data format, or we can convert the output data into the standard VTK file format by means of the *foamToVTK* function.
- In my experience using the native file format is more convenient for meshing analysis while for flow field quantities there is no particular difference between the two file format

Resources and references

SCtrain | SUPERCOMPUTING KNOWLEDGE PARTNERSHIP

- CAD modeler open-source/free:
  - Freecad: https://www.freecadweb.org/
  - Salome: https://www.salome-platform.org/
  - Blender: https://www.blender.org/
  - Onshape: https://www.onshape.com/en/products/free
- Courses (that I have attempted in my life):
  - CFD-Direct (basic/applied): you get trained in a perfectly clear manner directly from the architects of OpenFOAM (https://cfd.direct/openfoam-training/)
  - Wolf-Dynamics: very practical and engineering problem solving oriented (http://www.wolfdynamics.com/our-services/training.html)
  - Red-Eng: similar to Wolf-Dynamics, available also for 1-to-1 dedicated training on specific topics (site: http://www.red-fluid.com/; email: info@red-fluid.com)

Resources and references

- OpenFOAM Official Material:
  - User Guide: https://cfd.direct/openfoam/user-guide/
  - Download: https://cfd.direct/openfoam/download/
- Courses (that I have attempted in my life):
  - CFD-Direct (basic/applied): you get trained in a perfectly clear manner directly from the architects of OpenFOAM (https://cfd.direct/openfoam-training/)
  - Wolf-Dynamics: very practical and engineering problem solving oriented (http://www.wolfdynamics.com/our-services/training.html)
  - Red-Eng: similar to Wolf-Dynamics, available also for 1-to-1 dedicated training on specific topics (site: http://www.red-fluid.com/; email: info@red-fluid.com)

Thank you for your attention!

http://sctrain.eu/

Univerza v Ljubljani

TECHNISCHE UNIVERSITÄT WIEN

CINECA consorzio interuniversitario

VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Co-funded by the Erasmus+ Programme of the European Union

This project has been funded with support from the European Commission.
This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.