

CFD on HPC – OpenFOAM

Literature and OF Tools



Aleksander GRM

Literature



There are two main web pages with links to literature:

- ▶ openfoam.org (on PC workstations)
- ▶ cfd.direct (on HPC systems)

User guides:

- ▶ User Guide #1
- ▶ User Guide #2
- ▶ Programming Guide – Learn c++ code!

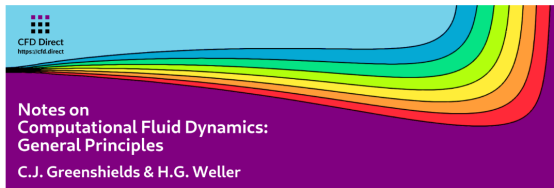
My GitHub repository:

- ▶ [OpenFOAM_School@github](https://github.com/OpenFOAM_School)

Link to GoogleDrive location for myOpenFOAM



[link to the book](#)



About the Book

Notes on Computational Fluid Dynamics (CFD) was written for people who use CFD in their work, research or study, providing essential *knowledge* to perform CFD analysis with confidence. It offers a modern perspective on CFD with the finite volume method, as implemented in OpenFOAM and other popular general-purpose CFD software. Fluid dynamics, turbulence modelling and boundary conditions are presented alongside the numerical methods and algorithms in a series of short, digestible topics, or *notes*, that contain complete, concise and relevant information. The book benefits from the experience of the authors: Henry Weller, core developer of OpenFOAM since writing its first lines in 1989; and, Chris Greenshields, who has delivered over 650 days of CFD training with OpenFOAM.

Contents

- Preface
- Symbols
- 1 Introduction
- 2 Fluid Dynamics
- 3 Numerical Method
- 4 Boundary Conditions
- 5 Algorithms and Solvers
- 6 Introduction to Turbulence
- 7 Reynolds-Averaged Turbulence Modelling
- 8 Sample Problems
- Index

ISBN 978-1-3999-2078-0, 291 pages.

[link to the book](#)

GMSH Tools



Set GMSH environment

```
1 1. load module gmsh:
2   $> ml av gmsh (check target version)
3   $> ml gmsh/4.11.1-foss-2022a
4
5 2. Run gmsh:
6   $> gmsh
```



To use Python environment we need only to load it

```
1 1. load module python:  
2   $> ml av python (check target version)  
3   $> ml python-version  
4  
5 2. Create new env:  
6   $> python3 -m venv local  
7  
8 3. Activate new env:  
9   $> source local/bin/activate
```



To be able to run advanced GMSH examples we need to set up Python environment

```
1 1. load module python:
2   $> ml av python (check target version)
3   $> ml python-version
4
5 2. Create new env:
6   $> python3 -m venv local
7
8 3. Activate new env:
9   $> source local/bin/activate
10
11 4. Install new packages (active env local):
12  $(local)> pip install numpy scipy sympy matplotlib gms
```


OpenFOAM Tools



To use OpenFOAM environment we need to load

```
1 List available modules:
2   $> module avail openfoam
3   $> ml av openfoam (equivalent with upper command)
4
5 For OpenFOAM to be running on HPC@ULFS we need to load this modules:
6   $> ml OpenFOAM
7   $> source $FOAM_BASH (set new OpenFOAM environment variables)
```



Additional modules for OpenFOAM environment

```
1 Load additional modules to support gnuplot in OpenFOAM:
2   $> ml OpenFOAM
3   $> source $FOAM_BASH
4
5   $> ml av gnuplot (for foamMonitor application)
6   $> ml spider gnuplot-version (see needed additional modules to load)
7
8   and load your Python envn with
9   $> source work/Python/local/bin/activate (or path where your local
    Python is)
```



Add the following part at the end in `system/controlDict`

```
1 functions
2 {
3     #includeFunc residuals
4 }
```

Create residual dictionary file `system/residuals` and include

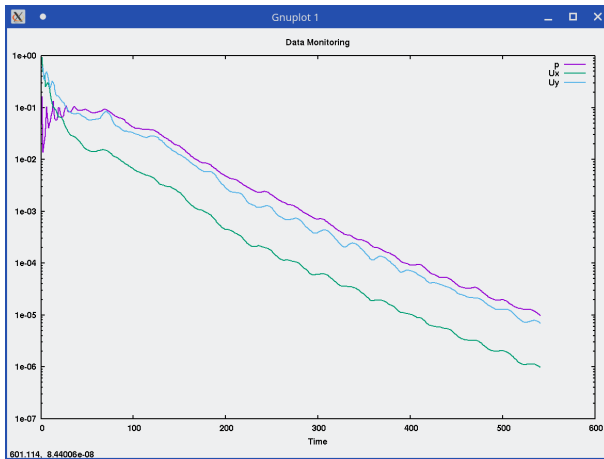
```
1 #includeEtc "caseDicts/postProcessing/numerical/residuals.cfg"
2
3 fields (p U);
```

Run monitor command

```
1 foamMonitor -l -r 1 postProcessing/residuals/0/residuals.dat
```



```
1 foamMonitor -l -r 1 postProcessing/residuals/0/residuals.dat
```



Parallel run OpenFOAM @ HPC



Parallel run is executed via **srun** command. All commands are packed in shell script and run with command

```
$> sbatch parallel_run.sh
```

To check your queue use command

```
$> squeue or only for user jobs $> squeue --me
```

Links to Slurm help:

- ▶ [Slurm help](#)
- ▶ [Slurm help @ FS-HPC](#)

Check HPC constellation

```
$> sinfo
```



Bash script containing all commands to run OpenFOAM at HPC

```
1 #!/bin/bash
2 #SBATCH --export=ALL,LD_PRELOAD=
3 #SBATCH --partition=haswell
4 #SBATCH --mem=0
5 #SBATCH --ntasks 32
6 #SBATCH --ntasks-per-node=16
7
8 module purge
9 module load OpenFOAM
10 source $FOAM_BASH
11
12 # Source tutorial run functions
13 source $FOAM_ETC/./bin/tools/RunFunctions
```




```
14 # Path to running case
15 caseName="cavityFine"
16 cd $caseName
17
18 # decompose the case (number of decompositions is equal to --ntasks)
19 runApplication decomposePar
20
21 # run parallel
22 echo "Start $(getApplication) in parallel. Log is written in case/log.$(
    getApplication)!"
23 srun --mpi=pmix $(getApplication) -parallel
24
25 # Check the running process with: tail -f case/log.$(getApplication)
```



Thank you for attention!



EuroHPC
Joint Undertaking



REPUBLIC OF SLOVENIA
MINISTRY OF HIGHER EDUCATION,
SCIENCE AND INNOVATION

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 1011301903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia.