CFD on HPC – OpenFOAM

01 – Introduction

Aleksander GRM – 2025

Outline

Short review

Content

- Small introduction into PDEs
- Solution methods
- Mesh description

The mathematical modelling of real systems is in most cases narrowed down to the mathematical model described by **Partial Differential Equations** (**PDE**).

Example: PDE describing waves motion on a free surface

$$\boxed{\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)} = c^2 \Delta u,$$

where the wave height is

$$u = u(t, \boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \quad \Omega \subseteq \mathbb{R}^2,$$

in the direction z, where $z \perp \Omega$.

In general we solve two types of problems

- **Initial Value problem (IVP)**:
  time t in independent variable of the problem, we solve **time-dependant** problem, so we need

  inital condition: $u_0 = u(t = t_0, \boldsymbol{x})$, where $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^n$
- **Boundary Value Problem (BVP)**:
  time t is not part of the problem, we solve **time-independent** problem, so we need

  boundary condition: $u_0 = u(\boldsymbol{x})$, where $\boldsymbol{x} \in \Gamma \subseteq \mathbb{R}^{n-1}$ $(\Gamma = \partial\Omega)$,

  where Γ is **boundary** of the computational domain. Many times you may see $\partial\Omega$.

IVP condition is obtained with the solution of BVP (start BVP with intuitive initialization)!

Solution methods

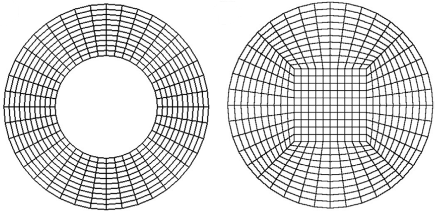
PDE can be solved in different ways

- **analytical methods**: mostly solving linear problems, or problems involving small parameter ϵ. Methods used are: separation of variables, series expansion, Perturbation methods, Laplace transform, Complex analysis methods, ...
- **numerical methods**: solve problems that is not possible to solve with analytical methods. In general we distinguish:
  - finite difference method (FDM) - solving **strong** form
  - finite volume method (FVM) - solving **weak** form
  - finite element method (FEM) - solving **weak** form
- **special numerical approaches**: use of FDM, FVM and FEM in different combinations
  - Immersed Boundary Method - IBM
  - Smothed Particle Hydrodynamics - SPH (mesh less method)

Numerical method - Mesh

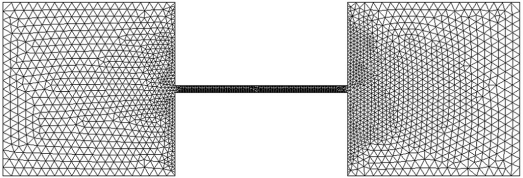
All numerical methods, except SPH, need the mesh. Mesh divides computational domain onto cells/elements

Structured grid



better convergence

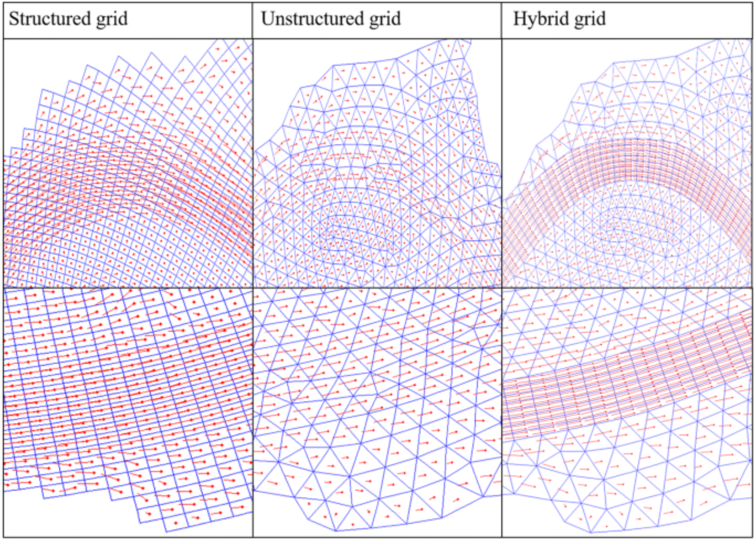
Unstructured grid



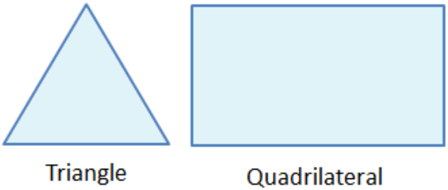
worse convergence

Very often we have a combination of both types! (next pages)

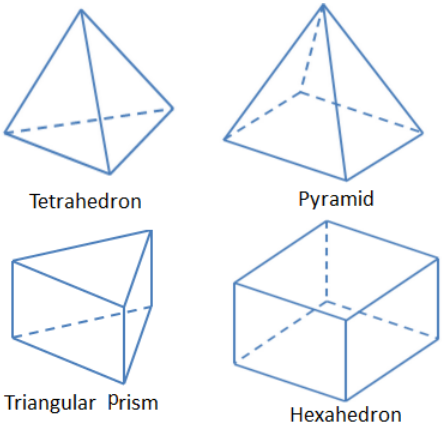
Numerical method - Mesh



2D mesh

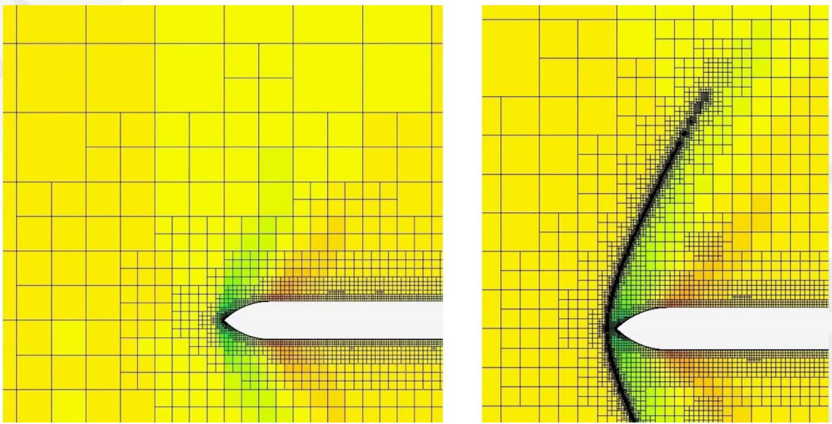


3D mesh



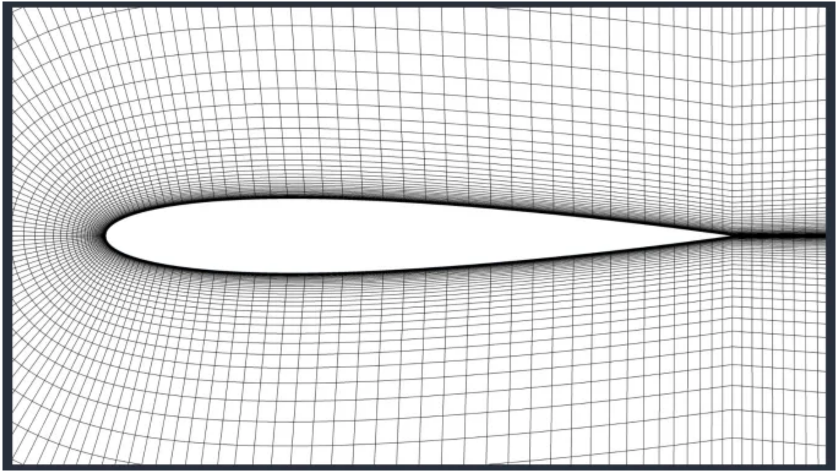
Mesh refinement

Some numerical solutions are only possible with **mesh refinement**!



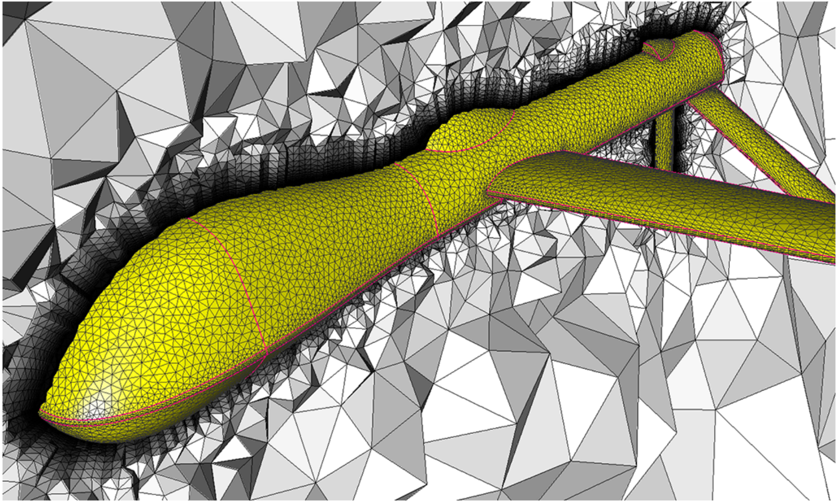
formation of shock waves - space entry simulation

Boundary Layer Mesh – 2D

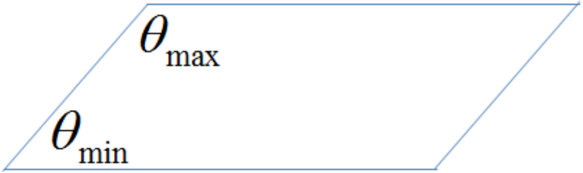


external flow – foil geometry

Boundary Layer Mesh – 3D



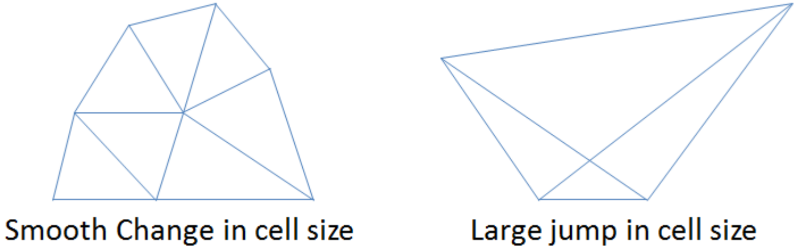
external flow – raptor geometry



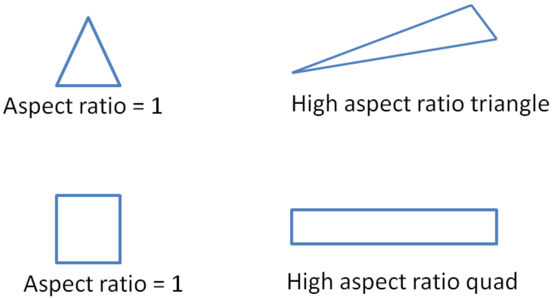
Skewness can be determined in many different ways, but always determines ratio of inner cell sides/angles!

$$\text{skewness} = \frac{\text{optimal cell size} - \text{cell size}}{\text{optimal cell size}}$$

It measures the deviation from optimal geometry (equidistant triangle)!

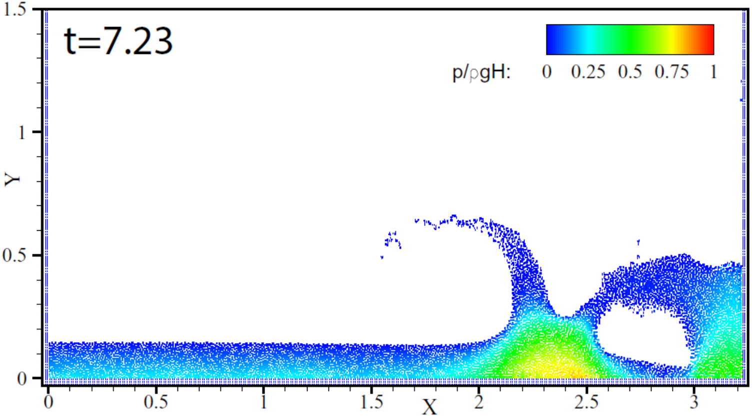


Smoothness measures the speed of cell size transition.

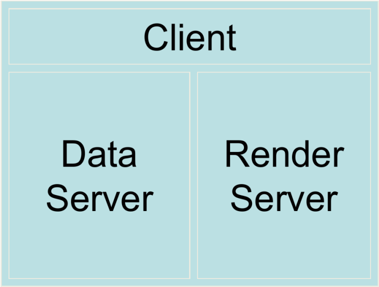


Aspect ratio measures the ratio of longest to the shortest side in a cell. Ideally it should be equal to 1 to ensure best results.

In mesh-less methods, we use particles that fill the space. They're mainly used where the shape of the surface changes over time, e.g. waves, continuous casting, solidification, etc.

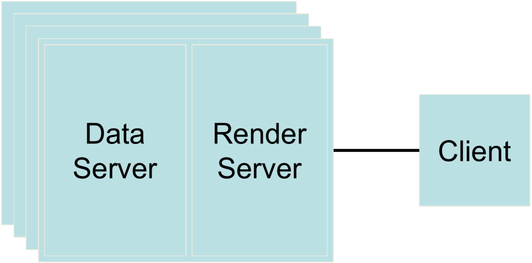


ParaVIEW



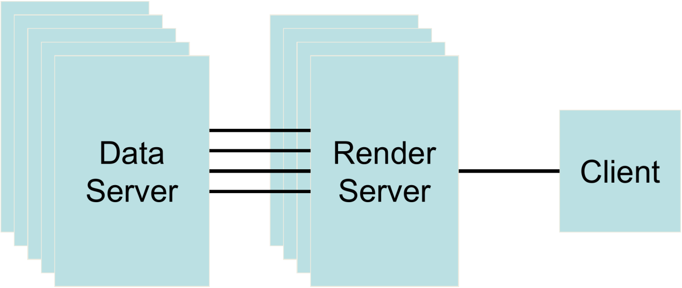
- In standalone mode, the client, data server, and render server are all combined into a single serial application.
- When you run the ParaView application, you are automatically connected to a built-in server so that you are ready to use the full features of ParaView.
- **This is how we are going to work.**

ParaVIEW architecture – Client-Server execution



- In client-server mode, you execute the **pvserver** program on a parallel machine and connect to it with the ParaView client application.
- The **pvserver** program has both the data server and render server embedded in it, so both data processing and rendering take place there.
- The client and server are connected via a socket.

Client-Render Server-Data Server execution



- In this mode, all three applications are running in separate programs.
- The client is connected to the render server via a single socket connection.
- The render server and data server are connected by many socket connections, one for each process in the render server.

http://www.ParaView.org/in-situ/

- **What is that?**

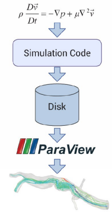
  Catalyst is a library for in situ (also known as co-processing or co-visualization) integration into simulations and other applications. In other words, the post-processing is done as the simulation runs. No need to read the simulation data to do post-processing.

- **Where can I get it?**

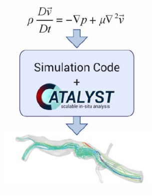
  You already have it. It is distributed with ParaView binaries.

Advantages of in situ analysis and visualization

- The post-processing is done as the simulation is running, we do not need to wait for the simulation to end.
- No more expensive IO.
- We have the full computational power of the supercomputer available to do this processing.
- Writing the results from the analysis and visualization will be much smaller and faster than the original full simulation data.



Traditional post-processing approach



In situ analysis and visualization,
co-processing or co-visualization

- paraFoam is the main post-processor distributed with OpenFOAM®.
- paraFoam is a wrapper of a third-party open source product named ParaView (www.ParaView.org).
- **In theory paraFoam and ParaView are the same.**
- The main post-processing tool provided with OpenFOAM® is a reader module to run with ParaView. The module is compiled into two libraries, `PV4FoamReader` and `vtkPV4Foam`.
- To visualize data obtained from OpenFOAM, you do not need to use paraFoam, you can use an stand-alone version of ParaView.
- In the user guide, it is written that it is highly recommended to use paraFoam.
- However, we have found that paraFoam can be slower than ParaView.
- **Therefore, we prefer to use ParaView.**

ParaVIEW vs ParaFOAM

- Basically, when we use paraFoam we load the libraries `PV4FoamReader` and `vtkPV4Foam`.
- We also create the empty file `case.OpenFOAM` used by paraFoam to load the case.
- If you do not want to use paraFoam, just create the empty file `case.OpenFOAM` and open it with ParaView.

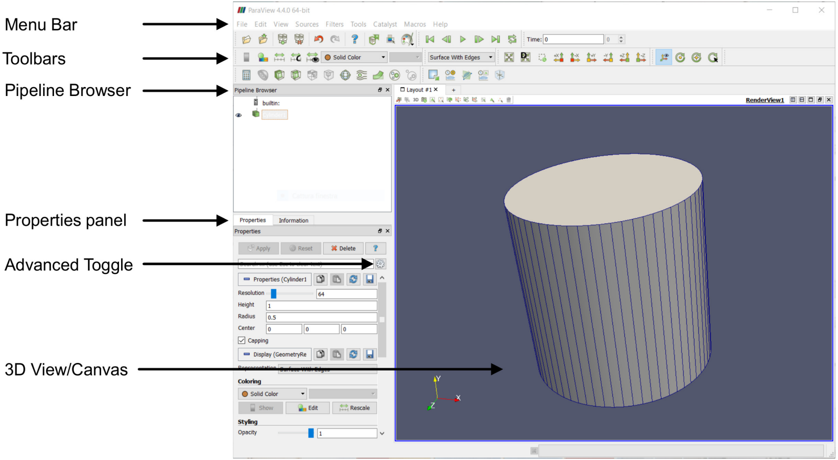
  In the terminal type

  - `$> touch case.OpenFOAM`
  - `$> paraview case.OpenFOAM`

  Or in one single step

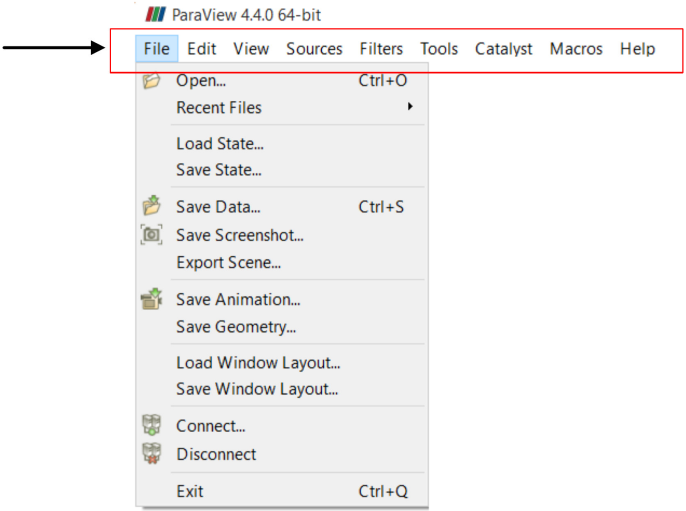
  - `$> touch case.OpenFOAM | paraview case.OpenFOAM`

Default User Interface

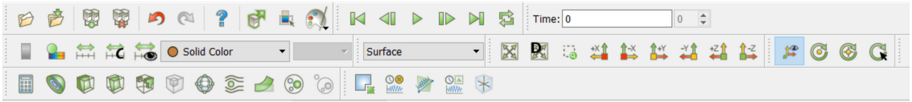


GUI layout is highly configurable, it is easy to change the look of the window.

Menu bar


ParaView 4.4.0 64-bit
File Edit View Sources Filters Tools Catalyst Macros Help
Open... Ctrl+O
Recent Files
Load State...
Save State...
Save Data... Ctrl+S
Save Screenshot...
Export Scene...
Save Animation...
Save Geometry...
Load Window Layout...
Save Window Layout...
Connect...
Disconnect
Exit Ctrl+Q

Toolbars

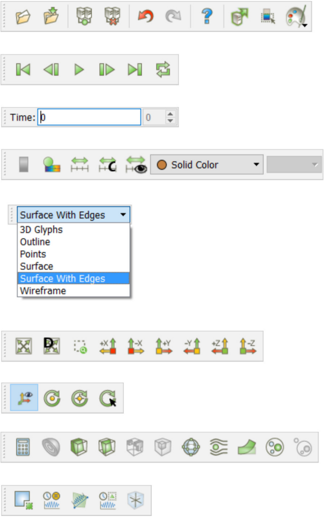


- The toolbars provide quick access to the most commonly used features within ParaView.
- By right clicking on the bar or selecting

  `View` $\rightarrow$ `Toolbars`

  you can show/hide what features to see on the toolbar.

Toolbars

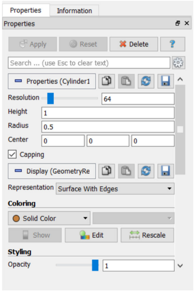


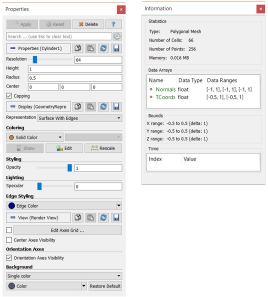
- Main Controls
- VCR Controls
- Current Time Controls
- Active Variable Controls
- Representation Toolbar
- Camera Controls
- Center Axes Controls
- Common Filters
- Data Analysis Toolbar

Pipeline Browser

- ParaView manages the reading and filtering of data with a pipeline.
- The pipeline browser allows you to view the pipeline structure and select pipeline objects.
- It provides a convenient list of pipeline objects with an indentation style that shows the pipeline structure.
- By clicking on the eyeball icon you can show/hide objects.

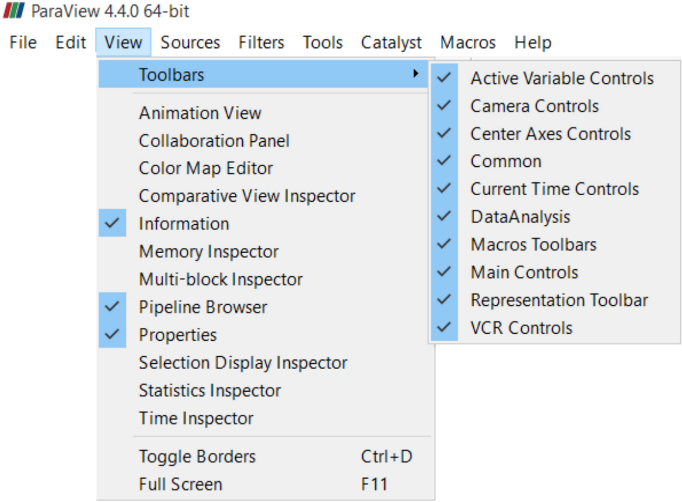
Pipeline Browser



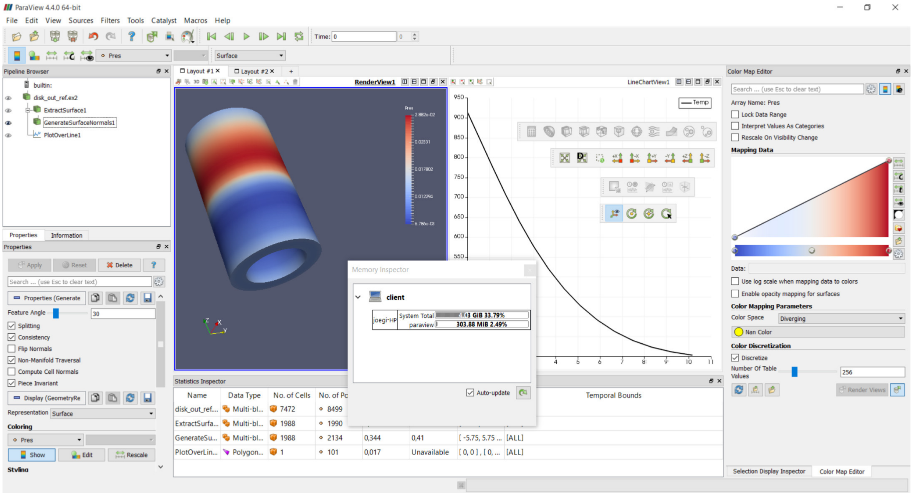


- The properties panel allows you to view and change the parameters of the current pipeline object.
- On the properties panel there is an advanced properties toggle that shows and hides advanced controls.
- The properties are by default coupled with an information tab that shows basic summary of the data produced by the pipeline object.

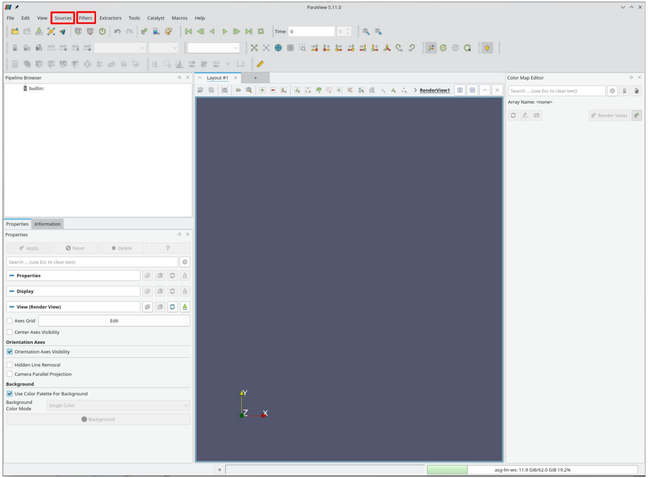
User Interface Components



User Interface Customization



Show the use of basic **Sources** and basic **Filters**



It will be shown in between the simple case of cylinder in cavity!

Introduction to OpenFOAM

- OpenFOAM folder structure
- Literature and Basic details
- Introductory example

Top level OF structure

Directory structure of OpenFOAM system as donwloaded from a **GIT** repository

- `OF system folder`
  - `applications --` source code for solvers
  - `bin --` bash scripts
  - `doc --` documentation
  - `etc --` compile & runtime controls
  - `platforms --` platform specific compiled binaries
  - `src --` source codes ot the system
  - `tutorials --` pre-configured cases
  - `wmake --` compile script system

Files containing initial condition for all dependant variables

```
0
├── U -- velocity
├── p -- pressure
├── k -- turbulent
├── epsilon -- turbulent
└── T -- scalar transport
```

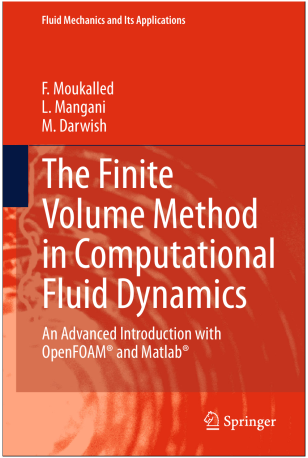
Folder containing files with constant data and mesh

```
constant
├── turbulenceProperties -- turbulent model properties
├── physicalProperties -- viscosity model & flow type
└── polyMesh -- computational mesh
    ├── boundary
    ├── points
    ├── faces
    ├── owner
    └── neighbour
```

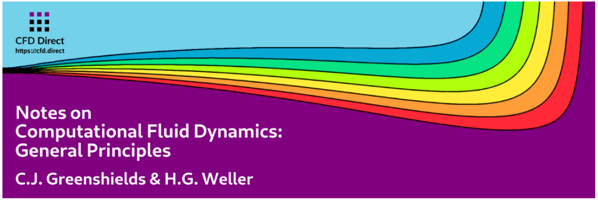
Folder containing system files

```
system
├── controlDict -- simulation controls
├── fvSchemes -- discretization schemes
├── fvSolution -- solution procedures
├── decomposeParDict -- domain decomposition & parallelization
├── residuals -- simulation residuals for post-process
└── ...
```

Good starting books



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

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

link to the book

Dimensions in OF are set in a list

No.	Property	SI unit	USCS unit
1	Mass	kilogram (kg)	pound-mass (lbm)
2	Length	metre (m)	foot (ft)
3	Time	second (s)	second (s)
4	Temperature	Kelvin (K)	degree Rankine (∘R)
5	Quantity	mole (mol)	mole (mol)
6	Current	ampere (A)	ampere (A)
7	Luminous intensity	candela (cd)	candela (cd)

Example: kinematic viscosity ν [m^2/s]

- value is set in a file `constants/transportProperties`

```
nu      [0 2 -1 0 0 0 0]      0.01;
```

OF diversity

- many turbulent flows (k-ε, k-ω, k-ω-SST,...)
- many boundary conditions
- multi phase flows models
- internal combustion models
- DNS
- stress analysis + FSI
- and many more ...
- look into `www.openfoam.org` - User Guide

Example of a simple case in OF – Cavity flow

1. create mesh: `blockMesh`
2. check mesh quality: `checkMesh`
3. run solver: `foamRun`
4. check residuals:
   `foamMonitor -l postPorocessing/residuals/0/residuals.dat`
5. preview results: `paraFoam -builtin`

Show and try in **`HPC@FS`** system!

Sponsorship

EURO SLING

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 101101903. 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.