Introduction to CFD – FVM – Part 01

EuroCC workshop

Aleksander GRM

June, 2023

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 $z \perp \Omega$.

Boundary conditions

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

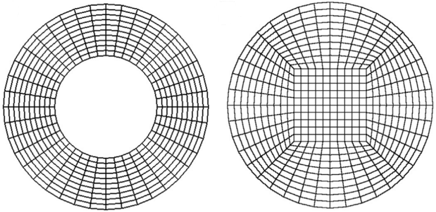
- **analytical methods**: mostly solving linear problems, or problems involving small parameter ϵ. Methods used are: separation of variables, series expansion, Perturbation methods, Laplace transform, Complex analysis methods, ...

- **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)

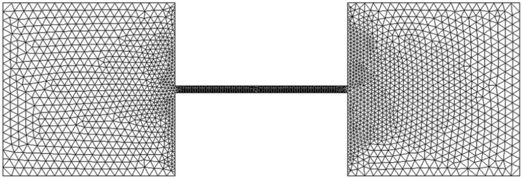
All numerical methods, except SPH, need the mesh. Mesh divides computational domain onto cells/elements

Structured grid



better convergence

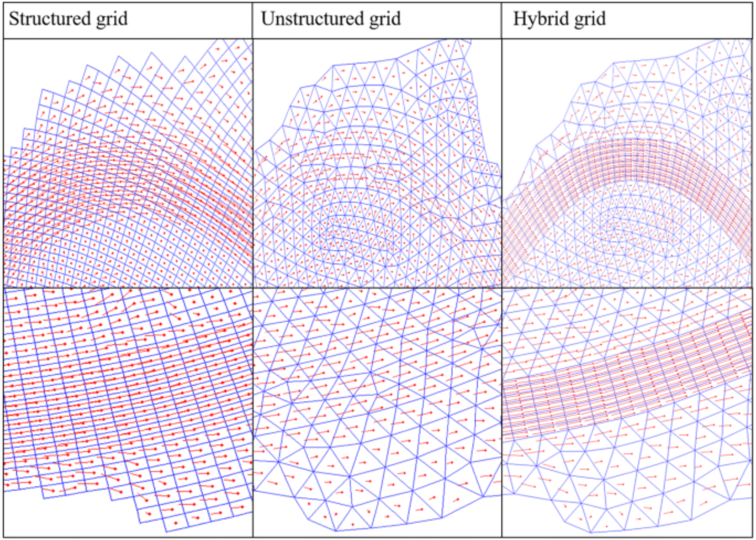
Unstructured grid



worse convergence

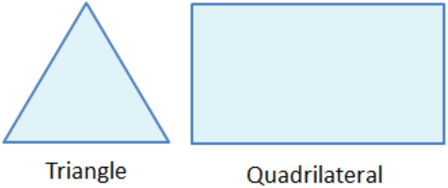
Very often we have a combination of both types! (next pages)

Numerical method - Mesh

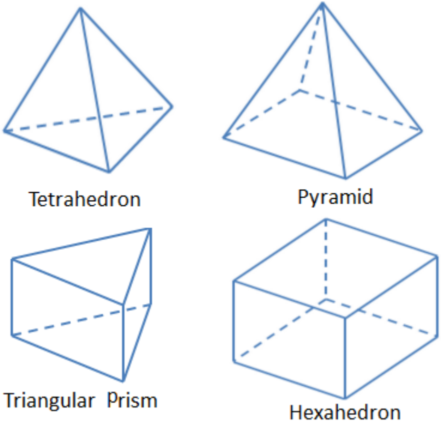


Mesh Cells/elements

2D mesh

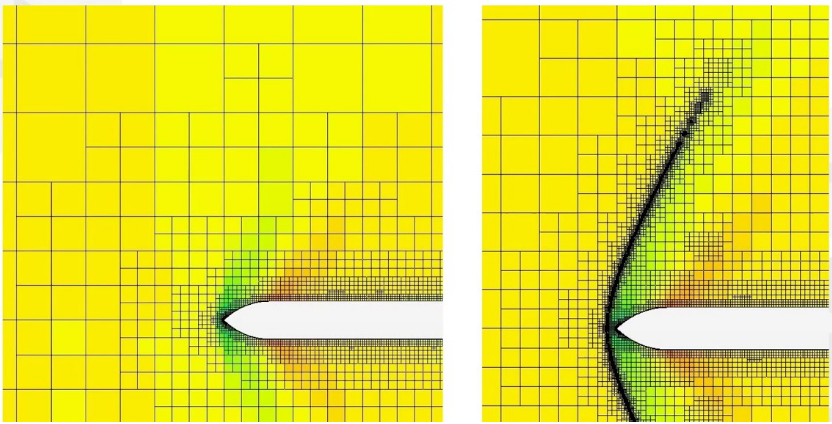


3D mesh



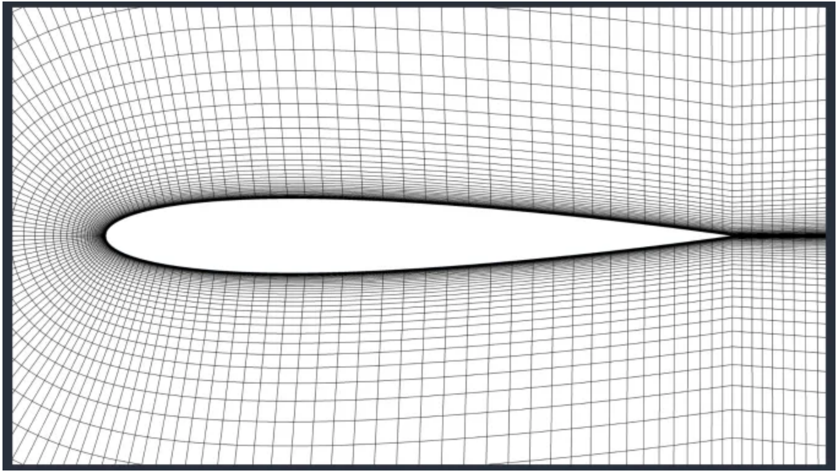
Mesh refinement

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



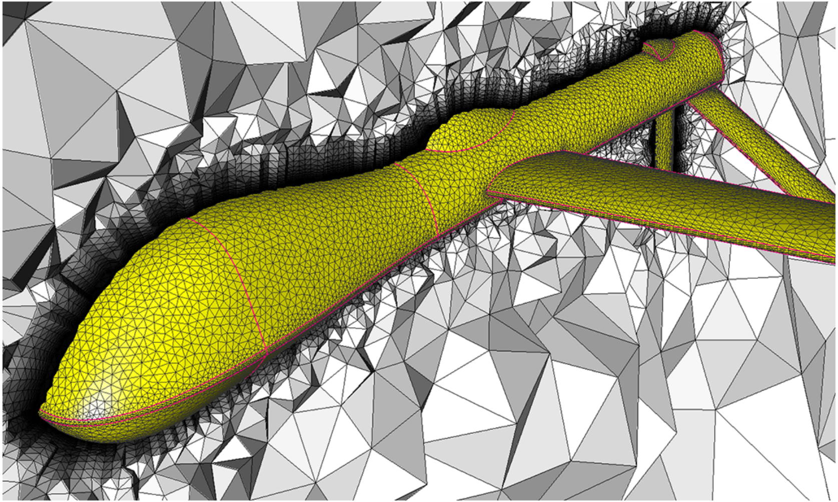
formation of shock waves - space entry simulation

Boundary Layer Mesh – 2D



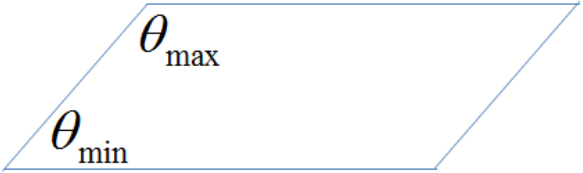
external flow – foil geometry

Boundary Layer Mesh – 3D



external flow – raptor geometry

Mesh quality – Skewness

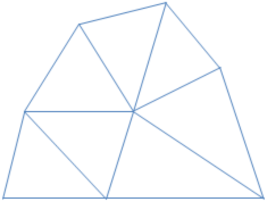


Skewness can be determined in many different ways, but always determines ratio of inner cell sides/angles!

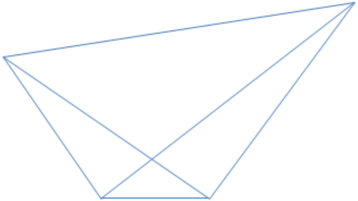
$$\text{skewness} = \frac{\text{optimal cell size} - \text{cell size}}{\text{optimal cell size}}$$

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

Mesh quality – Smoothness



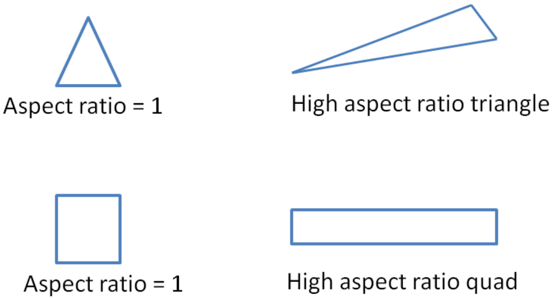
Smooth Change in cell size



Large jump in cell size

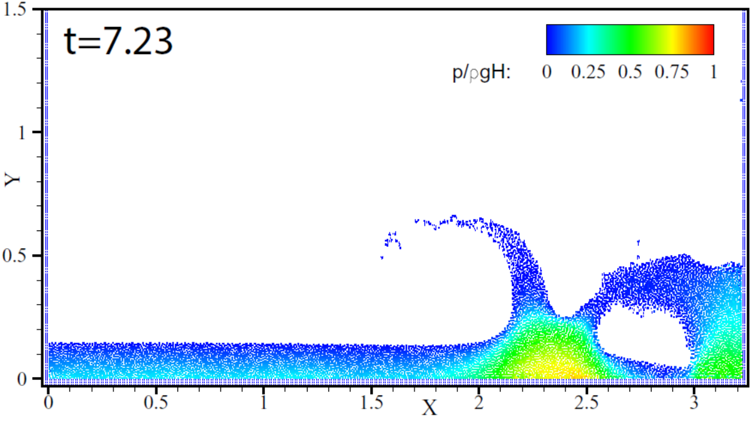
Smoothness measures the speed of cell size transition.

Mesh quality – Aspect ratio



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.



Finite volume method

There are two types of reference frame describing the flow field

Lagrange frame of reference (moving frame with flow – **material volume** - MV)

The Lagrangian specification of the flow field is a way of looking at fluid motion where the observer follows an individual fluid parcel as it moves through space and time.

This can be visualized as sitting in a boat and drifting down a river.

Euler frame of reference (fixed frame – **control volume** - CV)

The Eulerian specification of the flow field is a way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes.

This can be visualized by sitting on the bank of a river and watching the water pass the fixed location.

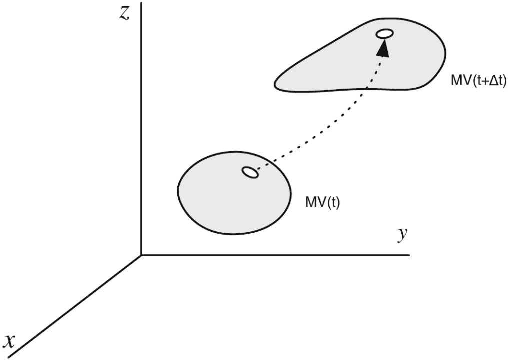
Motion in material volume (MV) is described by the mapping $\boldsymbol{x}(t, \boldsymbol{x}_0)$, that maps particle $\boldsymbol{x}_0$ from initial position in time t_0 to future position in time t

$$\boxed{\boldsymbol{x}(t, \boldsymbol{x}_0) : (t_0, \boldsymbol{x}_0) \rightarrow (t, \boldsymbol{x}_0)}$$

describing the particle $\boldsymbol{x}_0$ path in time t (**path line**).

Flow velocity can be founds as

$$\boxed{\boldsymbol{v}(t, \boldsymbol{x}(t, \boldsymbol{x}_0)) = \frac{\partial}{\partial t}\boldsymbol{x}(t, \boldsymbol{x}_0)}$$



Relation between the motion in Lagrange and Euler reference frame.

The derivative (rate of change) of a field variable $\phi(t, \mathbf{x}(t))$

with respect to fixed position

$$\frac{\partial}{\partial t}\phi(t, \mathbf{x}(t))$$

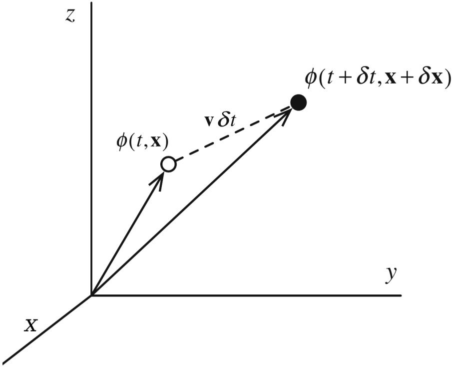
is called Euler derivative.

---

the derivative following a moving fluid parcel

$$\frac{D}{Dt}\phi(t, \mathbf{x}(t)) = \left(\frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla)\right)\phi$$

is called Lagrange derivative.



$$\text{Term I} = \frac{\mathrm{d}}{\mathrm{d}t}\left[\int_{\Omega(t)} (\rho\,\phi)\,\mathrm{d}\,V\right]$$
$$= \int_{\Omega}\left[\frac{\partial}{\partial t}(\rho\,\phi) + \nabla\cdot(\rho\,\boldsymbol{v}\,\phi)\right]\mathrm{d}\,V = \int_{\Omega}\left[\frac{D}{Dt}(\rho\,\phi) + \rho\,\phi\,\nabla\cdot\boldsymbol{v}\right]\mathrm{d}\,V$$

where $\Omega(t)$ is **material volume** and Ω is **control volume** and $\boldsymbol{v}$ is fluid velocity.

Change of the ϕ over time Δt within the material volume		Surface flux of the ϕ over time Δt across the control volume		Source/Sink of ϕ over time Δt within the control volume
Term I	=	**Term II**	+	**Term III**

The most general PDE (scalar or vector) encountered in fluid flow is

$$\boxed{\frac{\partial \phi}{\partial t} + \nabla \cdot \boldsymbol{F}(\phi) = Q(\phi)}$$

- **fluxes**: $\boldsymbol{F}(\phi)$
- **sources**: $Q(\phi)$

showing **conservation law**. It will be used to demonstrate the FVM discretization process!

Fluxes are normally two

- **advection**: $\boldsymbol{F}_{\mathrm{A}}(\phi) = \boldsymbol{v}\,\phi$
- **diffusion**: $\boldsymbol{F}_{\mathrm{D}}(\phi) = -D(\phi)\,\nabla\phi$

General conservation principle

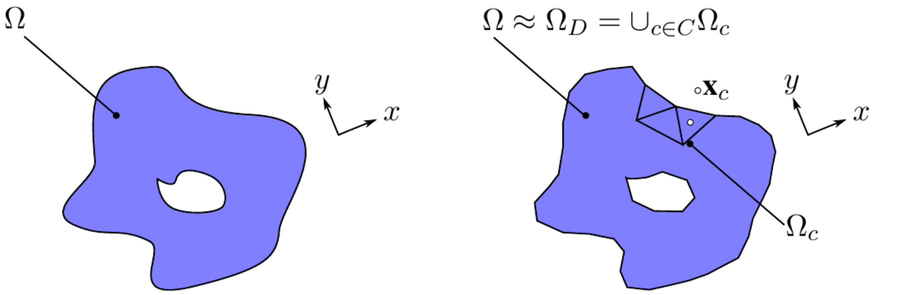
Change of the ϕ over time Δt within the material volume		Surface flux of the ϕ over time Δt across the control volume		Source/Sink of ϕ over time Δt within the control volume
Term I	$=$	**Term II**	$+$	**Term III**

$$\int_\Omega \left[\frac{\partial \phi}{\partial t}\right] \mathrm{d}\,V = \int_\Omega \left[\nabla \cdot \boldsymbol{F}(\phi)\right] \mathrm{d}\,V + \int_\Omega Q(\phi)\,\mathrm{d}\,V$$

The conservation equation for ADS PDE in fluid flow can be expressed as

$$\int_\Omega \left[\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi\, \mathbf{v})\right] \mathrm{d}\,V = \int_\Omega \nabla \cdot (D(\phi)\, \nabla \phi)\,\mathrm{d}\,V + \int_\Omega Q(\phi)\,\mathrm{d}\,V$$

Definition of basic cell Ω_c and ϕ Taylor expansion



$\phi = \phi(\mathbf{x}, t)$ $\qquad$ $\phi(\mathbf{x}, t) \doteq \phi(x_c) + \nabla \phi(\mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c)$

(a) Continuous flow domain. $\qquad$ (b) Discretized flow domain.

$$\phi(\boldsymbol{x}) = \phi(\boldsymbol{x}_c) + \nabla \phi(\boldsymbol{x}_c)\,(\boldsymbol{x} - \boldsymbol{x}_c) + \nabla\nabla\phi(\boldsymbol{x}_c) : (\boldsymbol{x} - \boldsymbol{x}_c) \otimes (\boldsymbol{x} - \boldsymbol{x}_c) + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{x}_c\|^3)$$

Volume average in basic cell Ω_c

Expressing volume average over Ω_c introduces averaged variable ϕ_c over Ω_c at the point $\boldsymbol{x}_c$, which are constant over cell Ω_c

$$\begin{aligned}\phi_c =& \phi(\boldsymbol{x}_c) \\ &+ \frac{1}{|\Omega_c|}\nabla\phi(\boldsymbol{x}_c)\int_{\Omega_c}(\boldsymbol{x}-\boldsymbol{x}_c)\,\mathrm{d}\,V \\ &+ \frac{1}{|\Omega_c|}\nabla\nabla\phi(\boldsymbol{x}_c) : \int_{\Omega_c}(\boldsymbol{x}-\boldsymbol{x}_c)\otimes(\boldsymbol{x}-\boldsymbol{x}_c)\,\mathrm{d}\,V + \ldots\end{aligned}$$

by the definition of $\boldsymbol{x}_c$ to be the **cell centre** it follows

$$\int_{\Omega_c}(\boldsymbol{x}-\boldsymbol{x}_c)\,\mathrm{d}\,V = 0.$$

The **average value** of ϕ over the finite volume Ω_c is exactly equal to the value of ϕ at the centroid $\boldsymbol{x}_c$ of Ω_c for a **linear** ϕ (method is at least 2^{nd} order).

Finite volume method order

$$\boxed{\phi_c \approx \frac{1}{|\Omega_c|}\int_{\Omega_c} \phi_{\text{linear}}(\boldsymbol{x}_c)\,\mathrm{d}V + \mathcal{O}(\|\boldsymbol{x}-\boldsymbol{x}_c\|^2)}$$

Cell–centered FVM is at least 2nd order method

The domain discretization of the unstructured FVM that assigns cell- average values ϕ_c of ϕ at centroids $\boldsymbol{x}_c$ of the cells Ω_c is ***second-order accurate***.

The average value of ϕ over the finite volume Ω_c is exactly equal to the value of ϕ at the centroid $\boldsymbol{x}_c$ of Ω_c for a linear ϕ, because for a linear ϕ the higher-order derivatives are zero. In other words, the cell-average (cell-centered) value at the centroid of finite volumes recovers values of linear fields exactly. A method that exactly recovers values of linear functions is at least second-order accurate.

$$\int_{\Omega_c} \frac{\partial \phi}{\partial t} \,\mathrm{d}V = \left(\frac{\partial \phi}{\partial t}\right)_c |\Omega_c| + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{x}_c\|^2)$$

we use **finite difference** to approximate temporal term

$$\text{backward Euler:} \quad \left(\frac{\partial \phi}{\partial t}\right)_c^{n+1} = \frac{\phi_c^{n+1} - \phi_c^n}{\delta t} + \mathcal{O}(\delta t)$$

$$\text{BDS2:} \quad \left(\frac{\partial \phi}{\partial t}\right)_c^{n+1} = \frac{3\phi_c^{n+1} - 4\phi_c^n + \phi_c^{n-1}}{2\delta t} + \mathcal{O}(\delta t^2)$$

where $n+1$ is new time step, n is current time step and $n-1$ is previous time step.

Temporal term is discretized with special FD scheme (e.g. BDS2)!

FVM discretization of equations – Advection term

Convert volume integral to surface integral, using divergence theorem

$$\int_{\Omega_c} \nabla \cdot (\phi\, \boldsymbol{v})\, \mathrm{d}V = \int_{\partial\Omega_c} \phi\, \boldsymbol{v} \cdot \boldsymbol{n}\, \mathrm{d}S.$$

In the case if many surfaces enclose the volume Ω_c and forms surface enclosure $\partial\Omega_c$ we can write

$$\partial\Omega_c = \bigcup_{f \in F_s} S_f,$$

where F_c is the index set of the faces S_f of the cell Ω_c. The integral can be transformed into a sum over all cell surfaces S_f

$$\int_{\partial\Omega_c} \phi\, \boldsymbol{v} \cdot \boldsymbol{n}\, \mathrm{d}S = \sum_{f \in F_c} \int_{S_f} \phi\, \boldsymbol{v} \cdot \boldsymbol{n}\, \mathrm{d}S.$$

$$\int_{\partial\Omega_c} \phi\, \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S = \sum_{f\in F_c} \int_{S_f} \phi\, \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S.$$

Averaging ϕ over surface S_f in face centre $\boldsymbol{x}_f$ we obtain

$$\phi_f = \frac{1}{|S_f|}\int_{S_f} \phi\,\mathrm{d}S + \mathcal{O}(\|\boldsymbol{x}-\boldsymbol{x}_c\|^2),$$

where $\phi_f = \phi(\boldsymbol{x}_f)$. The advection term discretization reduces to

$$\boxed{\int_{\partial\Omega_c} \phi\, \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S = \sum_{f\in F_c} \phi_f\, \boldsymbol{v}_f\cdot\boldsymbol{S}_f + \mathcal{O}(\|\boldsymbol{x}-\boldsymbol{x}_c\|^2)}$$

The rest of terms is discretized in a similar way!
(Look in the free book **The OpenFOAM – Technology Primer**)

Introduction to OpenFOAM

Content

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

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
```

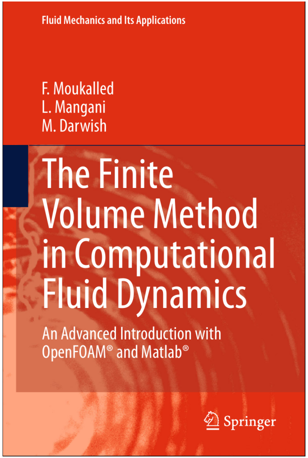
Folder containing files with constant data and mesh

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

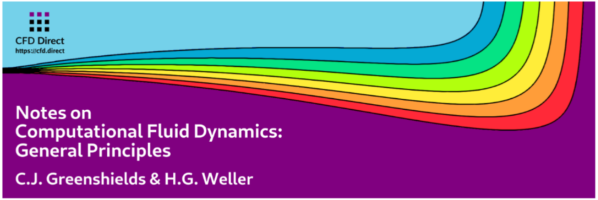
Folder containing system files

```
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 $[\mathrm{m}^2/\mathrm{s}]$

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

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

OF diversity

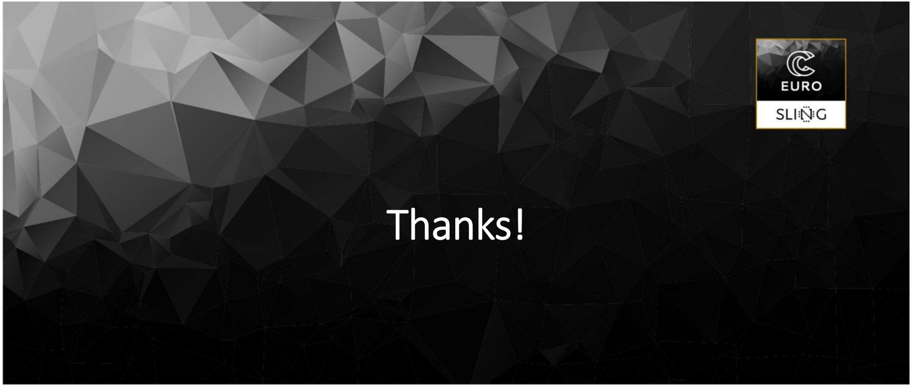
- many turbulent flows (k-ε, k-ω, k-ω-SST,...)
- many boundary conditions
- multi phase flows models
- internal combustion models
- DNS
- stress analysis + FSI
- and many more ...
- look into `www.openfoam.org` - User Guide

Example of a simple case in OF – Cavity flow

1. create mesh: `blockMesh` (internal OF command)
2. check mesh quality: `checkMesh` (internal OF command)
3. run solver: `icoFoam` (internal OF command)
4. preview results: `paraFoam` (internal OF command)

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

Sponsorship



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, United Kingdom, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Switzerland, Turkey, Republic of North Macedonia, Iceland, Montenegro

EuroHPC Joint Undertaking