

Basics of FVM using OpenFOAM with Applications

Marcus Ang

Department of Mechanical Engineering

Wichita State University

9/28/2023 -10/6/2023

Outline

1. Why OpenFOAM?
2. Introduction to HPC and BeoShock
3. Introduction to CFD workflow (T1 Setting up environment)
4. Meshing in a Nutshell (T2 Motorbike)
5. Introduction to Discretization (T3 Scalar Transport)
6. Additional Resources
7. Homework

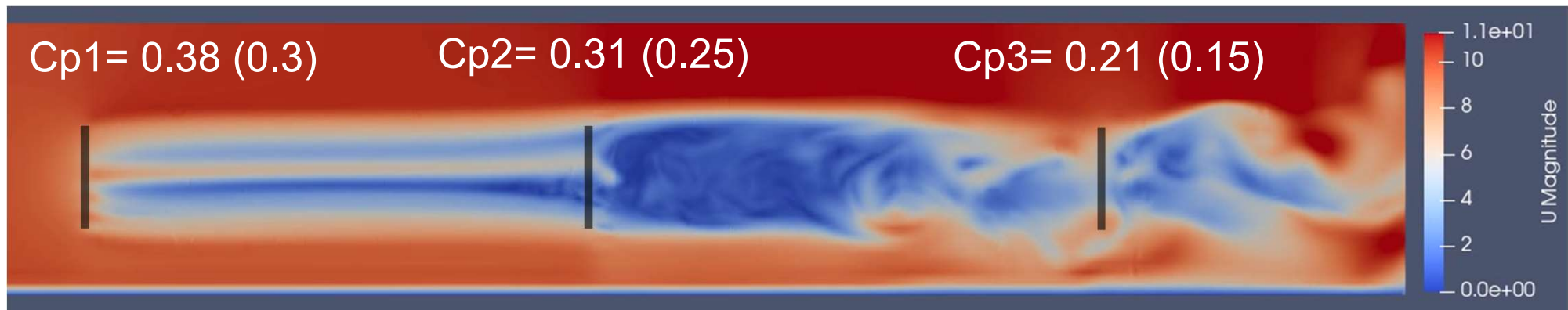
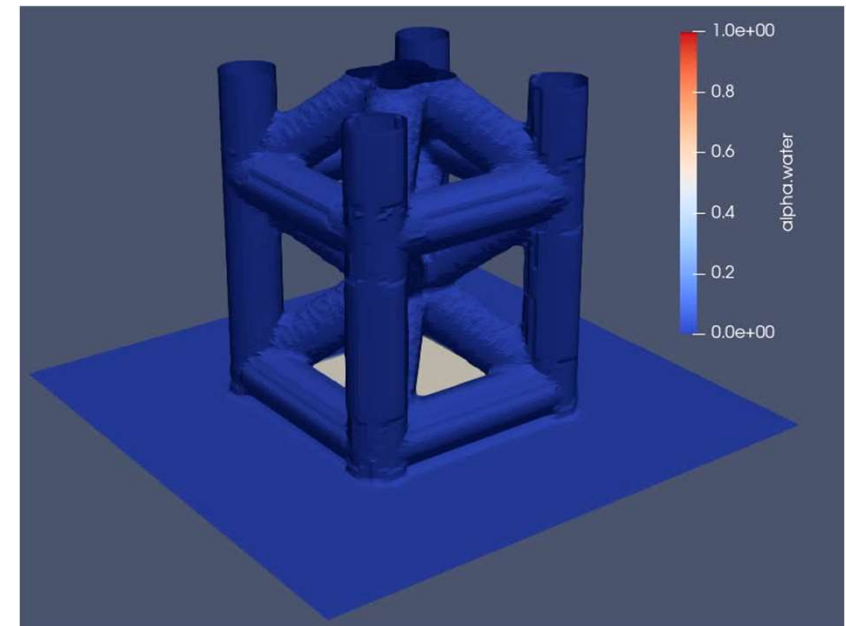
Acknowledgements

Materials are adapted from

- OpenFOAM User Guide
- Wolf Dynamic (Joel Guerrero)
- József Nagy
- Hrvoje Jasak

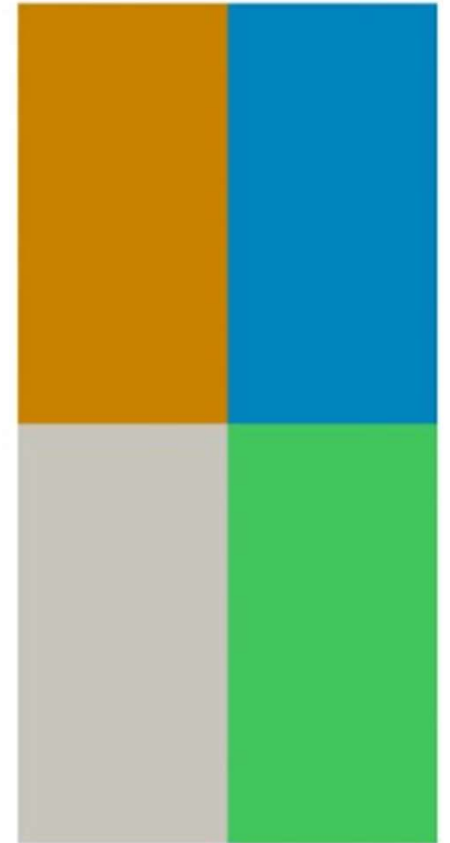
Why Open▽FOAM

- Top 3 most used CFD software.
- Top 5 most used software on HPC.
- Trusted by industry and academia.
- Extremely high parallelization.
- Open source
- Many applications and many more community repositories



Introduction to HPC and BeoShock

- High-performance computing (HPC) refers to the use of supercomputers or computing clusters
- Multiple CPUs work in parallel on different part of the same problem
- Large memory to handle massive datasets
- The HPC environment has a job scheduler and software are loaded in modules.



Introduction to HPC and BeoShock

1. GPU

- gpu201901
- gpu201902

Processors	2x Intel Xeon Gold 6240 (18 cores each at 2.60 GHz)
RAM	384 GB
Hard Drive	446 GB SSD
Networking	Broadcom BCM57412 10Gb Ethernet Mellanox ConnectX-5 100Gb InfiniBand
GPU	2x nVidia Tesla V100

2. High Memory

- highmem201901
- highmem201902

Processors	2x Intel Xeon Gold 6240 (18 cores each at 2.60 GHz)
RAM	1.5 TB
Hard Drive	223 GB SSD
Networking	Broadcom BCM57412 10Gb Ethernet Mellanox ConnectX-5 100Gb InfiniBand

3. Compute

- compute201901 through compute201916

Processors	2x Intel Xeon Gold 6240 (18 cores each at 2.60 GHz)
RAM	192 GB
Hard Drive	223 GB SSD
Networking	Intel XL710 10Gb Ethernet Mellanox ConnectX-5 100Gb InfiniBand

Introduction to CFD workflow

- Pre-processing
 - CAD model
 - Meshing (blockMesh, snappyHexMesh)
- Solver
 - Boundary conditions
 - Initial conditions and fluid properties
 - Turbulence model
 - Discretization schemes
 - Solver control
- Post Processing
 - Visualizing results (ParaView)

T1: Setting up environment

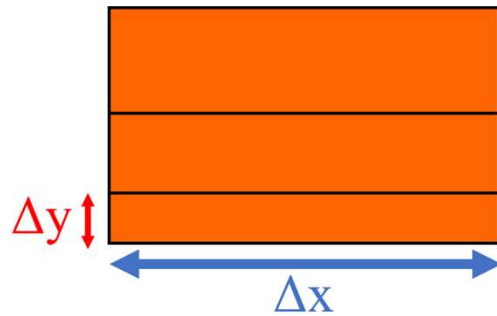
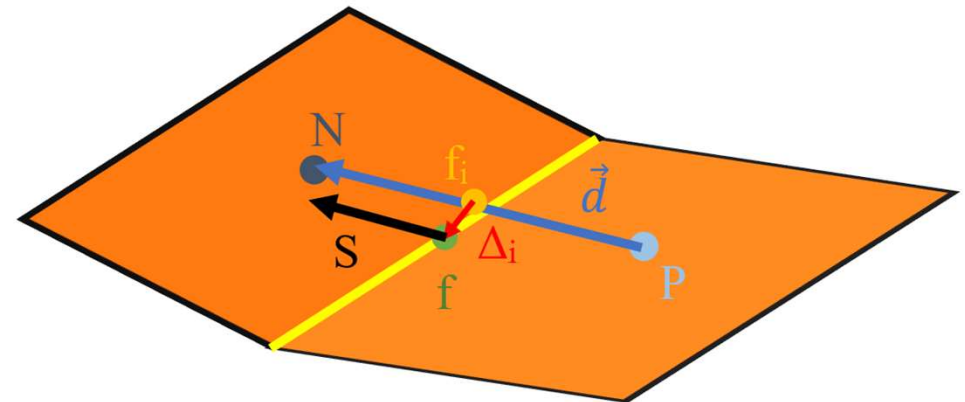
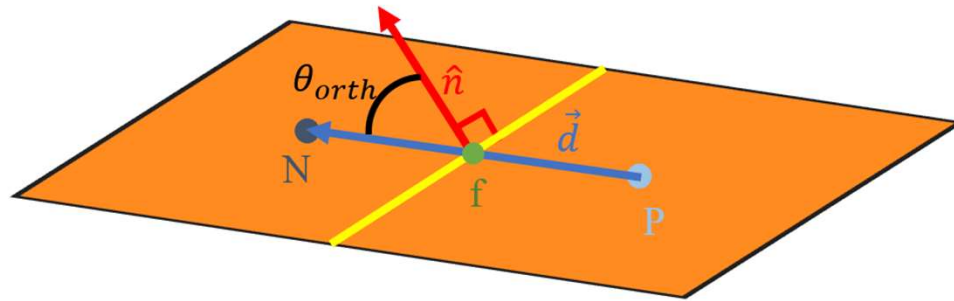
By the end of the tutorial, you should be able to

- Use headnodes and OnDemand
- Load OpenFOAM module
- Copy tutorial case to your \$HOME
- Navigate the file directory
- Understand basic Linux syntax and batch submission

Meshing in a Nutshell

- Meshing is the most important part of CFD
- Depending on the problem complexity we may need a very fine mesh → Computational Cost!
- GIGO! Garbage In Garbage Out!
- Unfortunately, there is no theory for mesh generation and mesh quality assessment
- General rule of thumb.. Does it look good?
- blockMesh and snappyHexMesh is considered in this workshop, however, external meshers may also be used

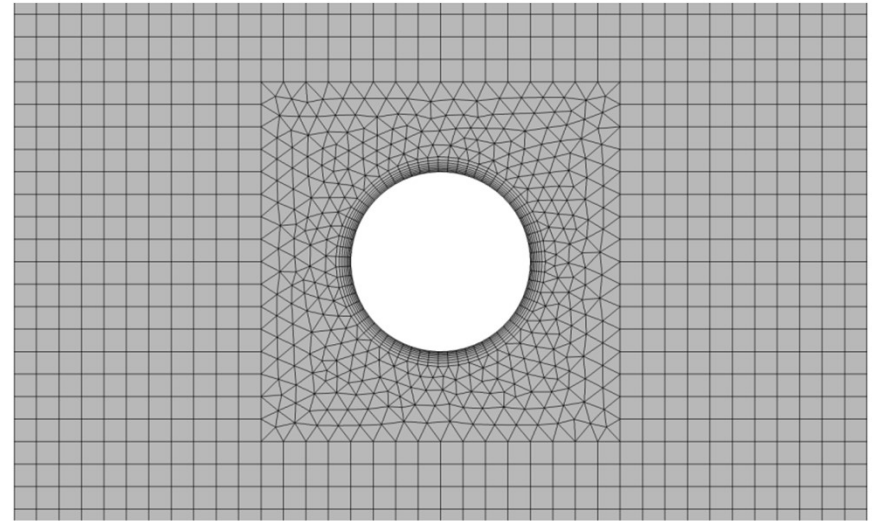
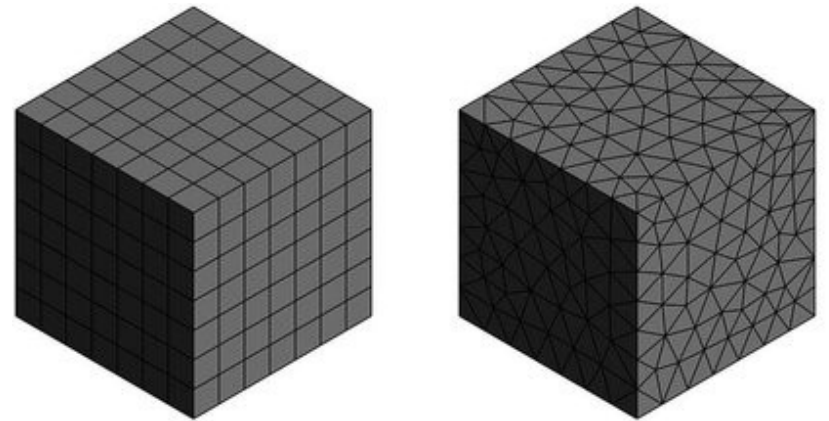
Mesh Quality Metrics



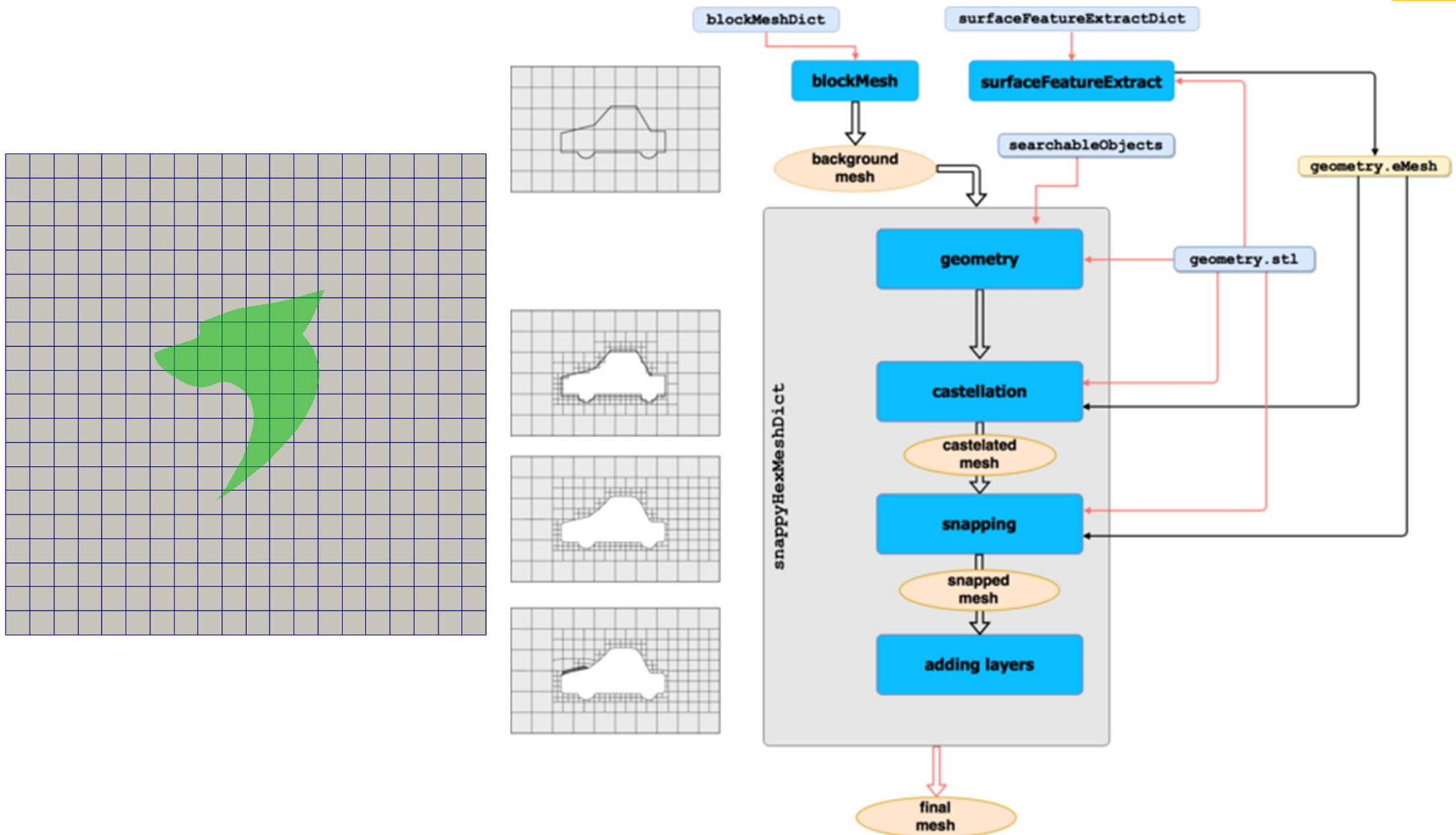
snappyHexMesh (sHM)

sHM is annoying... but

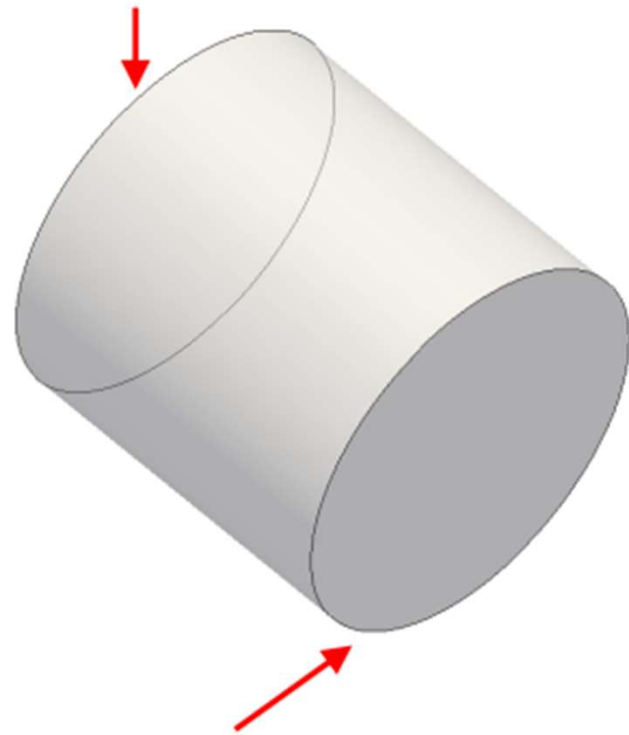
- Meshing is automated
- Parallel processing
- Guaranteed mesh quality
- Hex-dominated meshes
- Open-source
- Programmable
- “Easy” to refine mesh



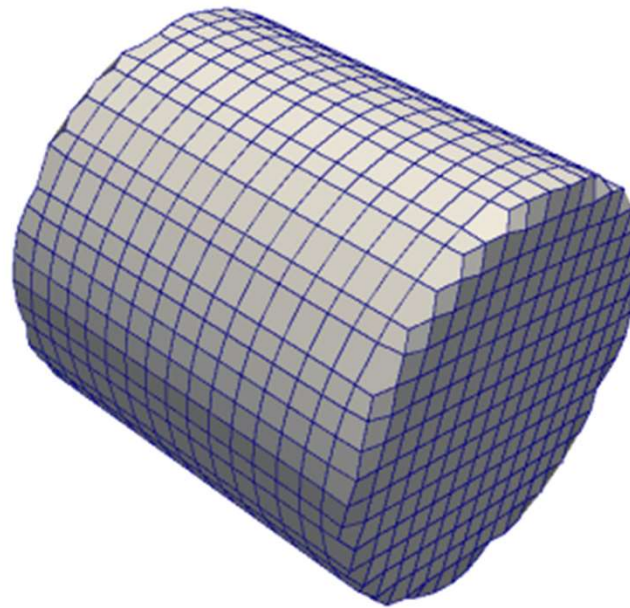
snappyHexMesh (sHM)



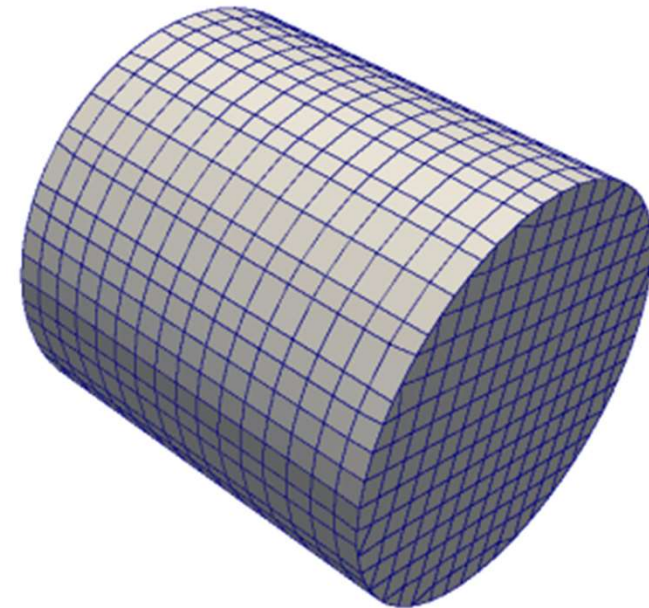
Sharp angle/edges



Sharp angle/edges



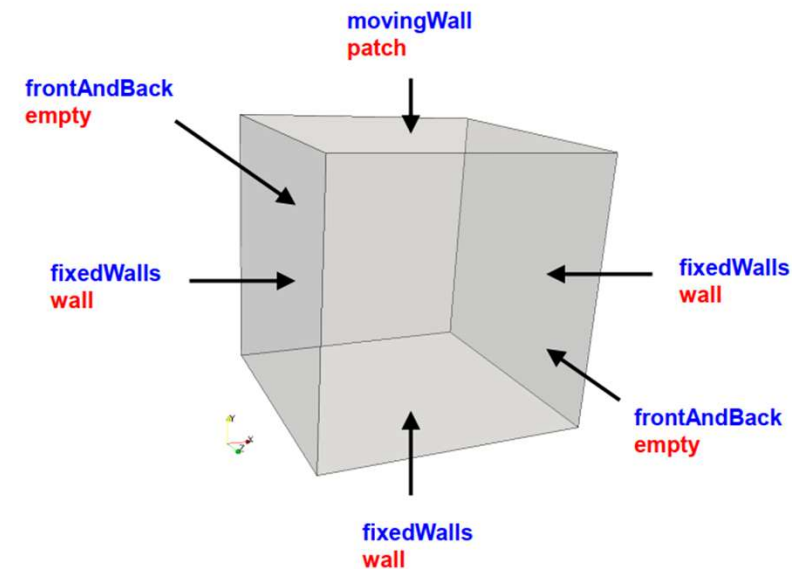
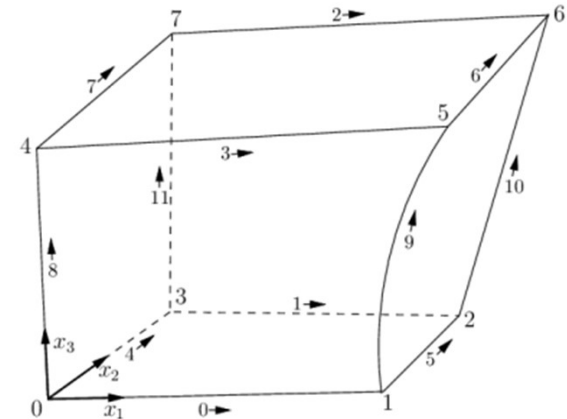
No edge refinement



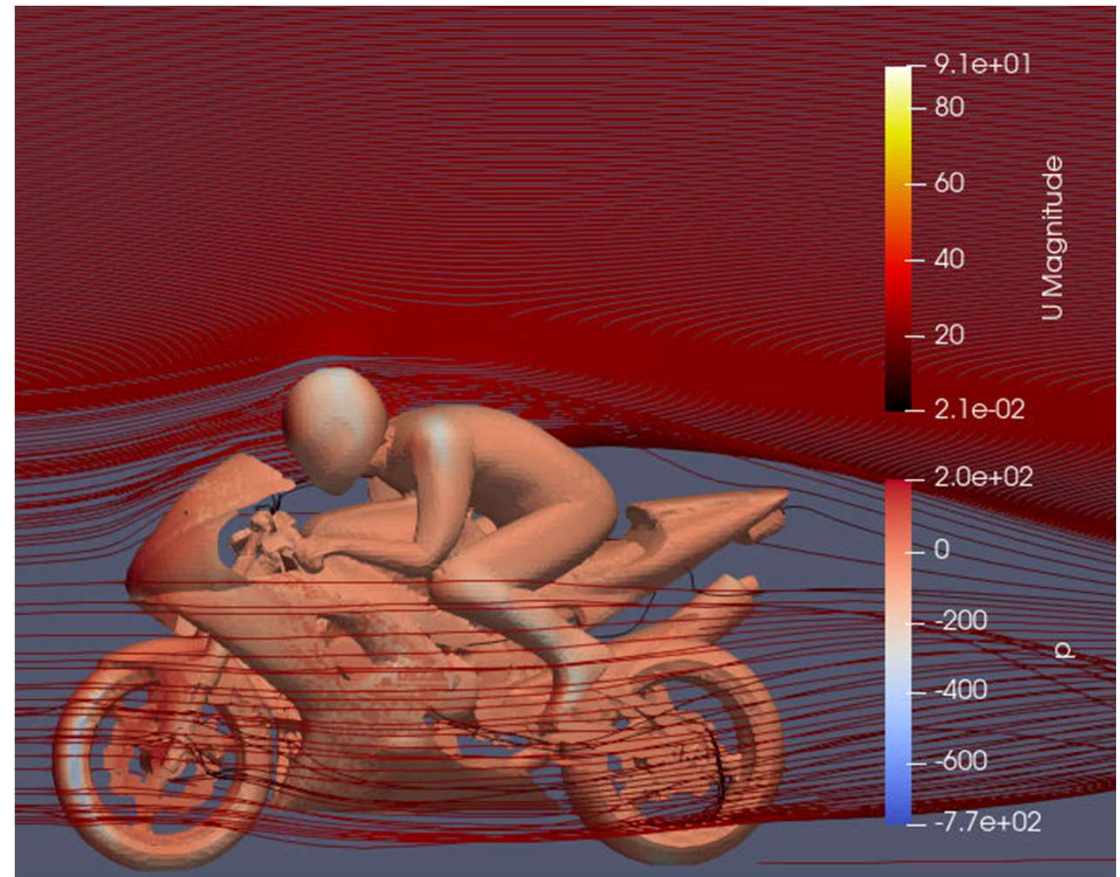
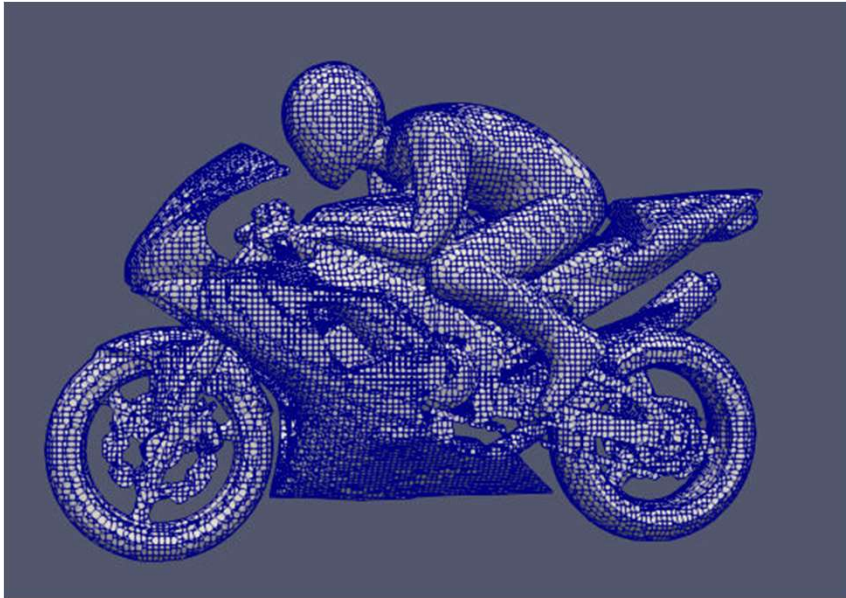
Edge refinement

blockMesh

- The most simple mesh is a cube, however this mesh is the starting point for sHM
- We specify vertices, connectivity and the number of cells in each direction
- Here we also specify the boundary patches



T2: Motorbike



T2: Motorbike

By the end of the tutorial, you should be able to

- Understand the structure of an OpenFOAM case file
- Understand meshing procedures in OpenFOAM
- Understand mesh decomposition
- Understand job submission script for SLURM
- Run case in parallel and reconstruct results
- Postprocess results in ParaView with tools including extractBlock, slice, streamTracer

Introduction to Discretization

- We can simplify the Navier-Stokes equations for a general passive scalar.

The Advection-Diffusion Equation (in 1D):

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = \Gamma \frac{\partial^2 \phi}{\partial x^2}$$

- First term is temporal term
- Second term is convective term
- Third term is the diffusion term
- Generally, we want at least second-order accuracy for all terms

Introduction to Discretization

- More generally, in most CFD manual it is written as

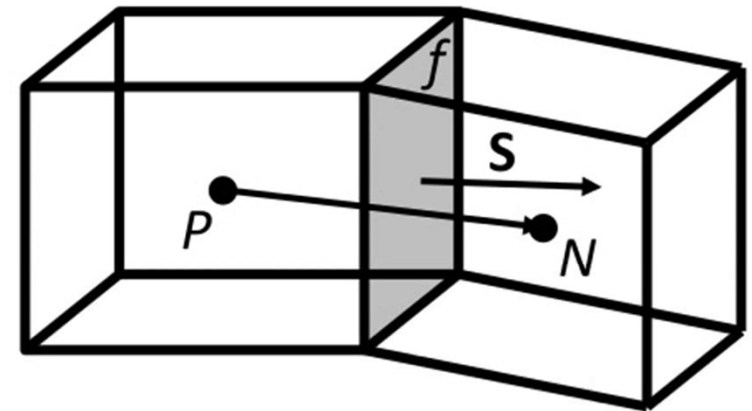
$$\frac{\partial \rho \Phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \Phi) = \nabla \cdot (D \nabla \Phi) + S_{\Phi}(\Phi)$$

- We apply integration to a finite volume

$$\begin{aligned} \int_t^{t+\Delta t} \int_V \frac{\partial \rho \Phi}{\partial t} dV dt + \int_t^{t+\Delta t} \int_V \nabla \cdot (\rho \mathbf{u} \Phi) dV dt \\ = \int_t^{t+\Delta t} \int_V \nabla \cdot (D \nabla \Phi) dV dt \end{aligned}$$

- Recall: Gauss Divergence Theorem

$$\int_V \nabla \cdot \Phi dV = \int_S d\mathbf{S} \cdot \Phi$$



Temporal Discretization

- Backward time scheme
- Crank-Nicolson time scheme
- Euler implicit time scheme

$$\frac{\partial}{\partial t}(\phi) = \frac{\phi - \phi^o}{\Delta t}$$

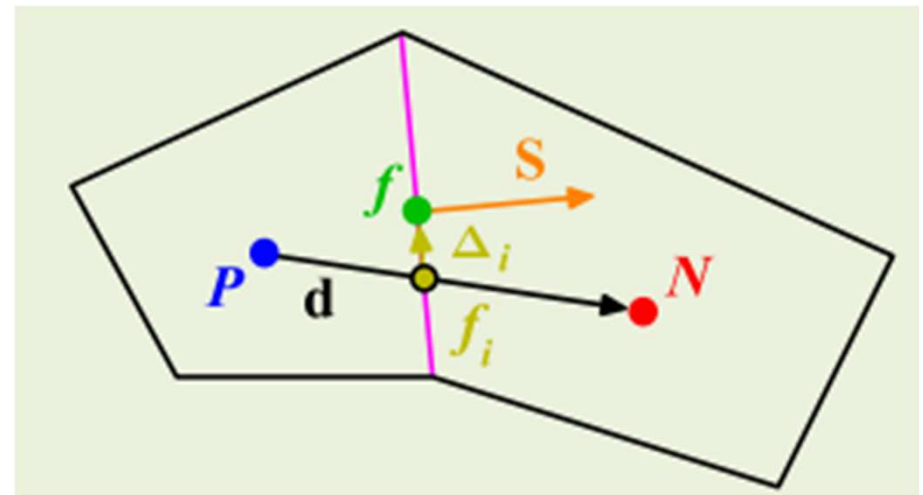
- Local Euler implicit/explicit time scheme
- Steady state time scheme

Convective Discretization

- Using gauss theorem, it can be discretized to

$$\int_V \nabla \cdot (\rho \mathbf{u} \Phi) dV = \int_S d\mathbf{S} \cdot (\rho \mathbf{u} \Phi) \approx \sum_f \mathbf{S} \cdot (\rho \mathbf{u})_f \Phi_f$$

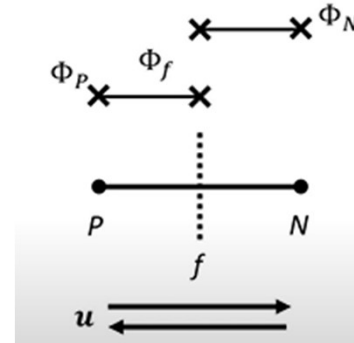
- Recall that we only know the value at the cell center, however the formulation now requires values at face center! Some interpolation scheme is needed...



Convective Discretization

- Gauss Upwind (1st order)

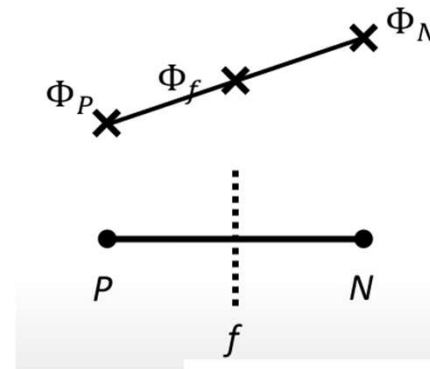
$$\Phi_f = \begin{cases} \Phi_P & \text{for } (\mathbf{u} \cdot \mathbf{n}) > 0 \\ \Phi_N & \text{for } (\mathbf{u} \cdot \mathbf{n}) < 0 \end{cases}$$



- Gauss Linear (2nd order)

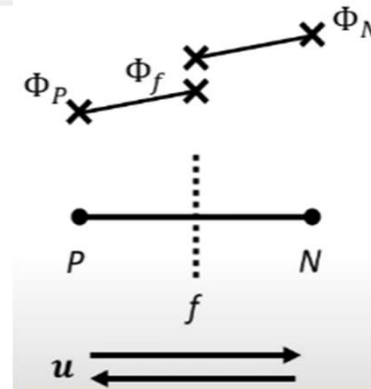
$$\Phi_f = f_x \Phi_P + (1 - f_x) \Phi_N$$

$$f_x = \frac{|x_f - x_N|}{|x_f - x_N| + [x_f - x_P]}$$



- Gauss Linear upwind (2nd order)

$$\Phi_f = \Phi_{P,N} + \nabla \Phi \cdot \mathbf{r}$$



Diffusion Discretization

- Similarly, we can discretize the diffusion term

$$\int_V \nabla \cdot (D \nabla \Phi) dV = \int_S d\mathbf{S} \cdot (D \nabla \Phi) \approx \sum_f D_f (\mathbf{S} \cdot \nabla \Phi)_f$$

- However, notice that Laplacian operator is just the divergence of the gradient.
- Therefore, typically in OpenFOAM, we use the syntax
Gauss <divergence interpolation scheme> <gradient scheme>

Diffusion Discretization

Gradient Discretization

- Gauss Linear (2nd Order)
- Least Square (2nd Order)
- The gradient term is extremely important in the FVM formulation as high gradients affects stability and conservation. Undershoot or overshoot due to oscillations must be bounded and is solved by introducing a limiter.
- Note that in the diffusion discretization, we actually use surfaceNormalGradient however, we typically use the same scheme.

$$(\nabla \phi)_P = \frac{1}{V_P} \sum_f (\mathbf{S}_f \phi_f)$$

Brief Note on Stability

- Thus far, we focused on spatial derivatives. The stability of the scheme is typically obtained via a Von Neumann Stability analysis. The fineness of the grid is typically based on how much detailed physics the simulation needs to capture.
- However, for temporal derivative, there is a very strict criteria known as the Courant-Friedrichs-Lewy (CFL) condition.

$$C = \frac{u \Delta t}{\Delta x}$$

- Courant number must be less than 1 to be stable, however, we want this to be as low as possible.

T3: Scalar Transport

By the end of the tutorial, you should be able to

- Understand basic discretization method
- Understand errors associated with first and second order accurate discretization
- Manipulate dataset in ParaView
- Calculate the courant number



Additional aspects not covered

- Mesh quality metrics
- Non-orthogonal corrections
- Boundary conditions
- Solution algorithm → fvSchemes
- Grid convergence
- Domain decomposition methods

Additional Resources

- OpenFOAM tutorials
- [OpenFOAM Wiki](#)
- [Wolf Dynamics - Customer-driven flow solutions](#)
- [CFD Online \(cfd-online.com\)](#)
- [Numerical Solutions | OpenFOAM Knowledge Base \(holzmann-cfd.com\)](#)

Homework

- Lid-Driven Cavity Flow
 - Follow instructions from [2.1 Lid-driven cavity flow \(openfoam.com\)](https://openfoam.com/docs/guides/2.1-lid-driven-cavity-flow/) up to 2.1.7 Increasing the Reynolds number.
 - Report on the significance of increasing mesh resolution and effects of higher Reynolds number.
- Heat Conduction in a Pipe
 - Go through the provided case directory and check the user guide on the new boundary conditions and solvers.
 - Run the simulation and report the temperature profiles.
- 2D Backward Facing Step Validation
 - Recreate the geometry from E. Erturk (2008) and validate the solution at $Re = 100$. *Only use the provided files when stuck!
 - Report the location of reattachment.