Structure of OpenFOAM

Open-Source CFD Course 2021 – Lecture 2

Miklós BALOGH and Josh DAVIDSON

2021

Table of Contents

Student projects

- List of possible projects put online this week
- Or propose your own project topic
- The sooner you choose your project the sooner you can get started on it....
- Reminder:
  - 4 page report due in week 13 (50% of final mark)

Review on theory

We will first review the theory of CFD...

Hydro-thermodynamical equation system

Conservation laws

- Momentum (Navier–Stokes equations):

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \left[\nabla^2 \boldsymbol{u} + \frac{1}{3}\nabla\left(\nabla \cdot \boldsymbol{u}\right)\right] + \boldsymbol{g}$$

- Mass (continuity):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0$$

- Energy:

$$\frac{\partial\left(\rho c_p T\right)}{\partial t} + \nabla \cdot (\rho c_p T \boldsymbol{u}) = \nabla \cdot (k \nabla T) + Q_\nu + Q_{ch.reaction}$$

Relationship between the material properties

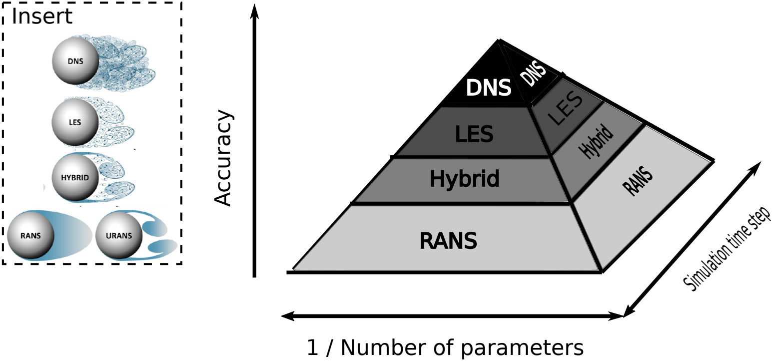
- Ideal gas law:

$$p = \rho R T$$

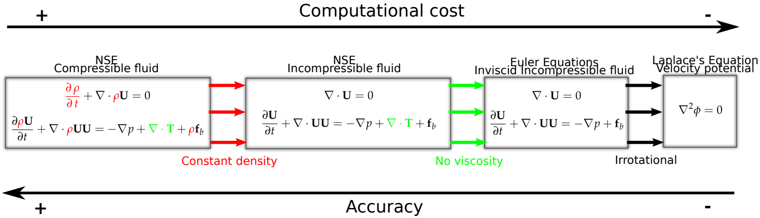
Numerical solution of the N–S equations

- While in general, the analytical solution of the N–S equation is not known, a numerical approximation is possible
- To numerically solve the N–S equations, the continuous partial differential equations are discretized into a system of linear algebraic equations, which can then be solved by computer. That is, the continuum is broken up into finite temporal and spatial portions to transform a calculus problem into an algebraic problem.
  - Spatial discretization (mesh: grid or cell network)
  - Temporal discretization (suitable time step, Δt)

Solution methods



Simplification of the Equations

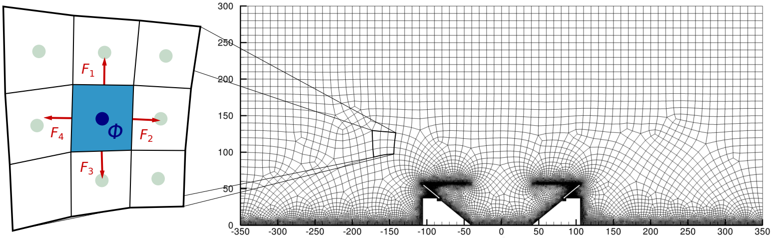


Numerical solution of the N–S equations

- Spatial discretization
  - Finite Volume Method (FVM)
  - Finite Element Method (FEM)
  - Finite Difference Method (FDM)
  - Spectral methods (e.g. for DNS on periodic domains)
  - Particle methods (e.g. SPH)
  - Lattice gas model, lattice-Boltzmann method
- Temporal discretization (unsteady problems)
  - Explicit and implicit schemes, stability criteria (e.g. CFL)
  - Local time-step, adaptive time-step control
- Pressure-velocity coupling
  - Pressure correction (sequential, e.g. SIMPLE, PISO)
  - Coupled: simultaneous solution of the equations

Finite Volume Method (FVM)

- Arbitrary cells (volumes)
- Conservation laws are applied on these in integral form



Finite Volume Method (FVM)

- General form of the conservation laws:

$$\frac{\partial}{\partial t}\int_V \rho\phi \mathrm{d}V + \oint_A F\mathrm{d}\vec{A} = \int_V S_V \mathrm{d}V + \oint_A S_A \mathrm{d}\vec{A}$$

- Where ϕ and F respectively
  - The conservative quantity per unit mass:

$$\phi = u_x/\rho,\ u_y/\rho,\ u_z/\rho,\ T/\rho,\ \ldots$$

  - The sum of convective and conductive fluxes:

$$F = F_{conv.} + F_{cond.} = \rho\phi\boldsymbol{u} - \rho\nabla\phi$$

Numerical analysis

Once a solution method for the NSE is selected, how do we perform the numerical analysis of a given problem/case in CFD...

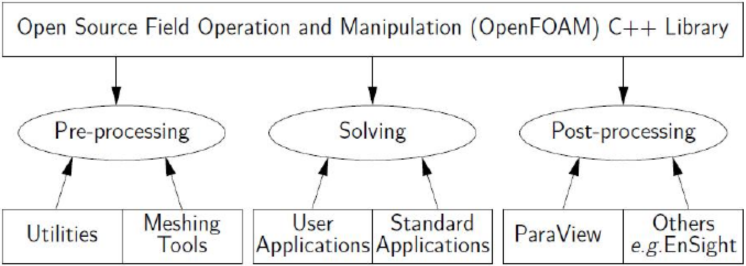
Steps of the numerical analysis

1. Construction of the geometry (computational domain)
2. Mesh generation
   - The basis of the spatial discretization
   - Decomposition of the domain to cells
3. Definition of the boundary conditions
   - Set value / set gradient / inlet / outlet / symmetry / etc
4. Definition of the initial conditions
   - Constant - predefined values
   - Patch - values given cell by cell (e.g. theoretical values)
   - Mapping - values from simulation (interpolation)
5. Simulation (numerical integration of the equations)
6. Post-processing

OpenFOAM structure

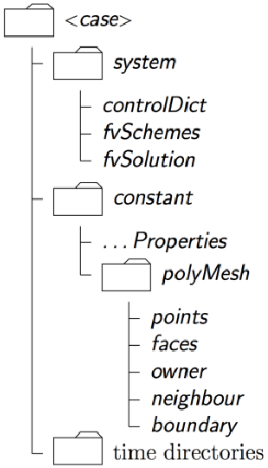
How do you do it in OpenFOAM???

OpenFOAM structure



[0] OpenFOAM User Guide (https://cfd.direct/openfoam/user-guide/)

OpenFOAM case folder directory



[0] OpenFOAM User Guide (https://cfd.direct/openfoam/user-guide/)

Steps 1 and 2

Construction of the geometry and the mesh generation is typically done hand-in-hand.

OpenFOAM structure

- OpenFOAM reads the geometry and mesh details from the files in the *constant/polyMesh* folder
- Can construct and modify these files using applications native to OpenFOAM:
  - *blockMesh*
  - *snappyHexMesh*
  - *refineMesh*
  - *transformPoints*
  - etc...
- Or with external programs (more on this next lecture)

Steps 1 and 2

OpenFOAM meshing applications

```
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$ ls
advanced  conversion  doc  generation  manipulation
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$ cd advanced/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/advanced$ ls
collapseEdges      modifyMesh  refineHexMesh    refineWallLayer  selectCells      splitCells
combinePatchFaces  PDRMesh     refinementLevel  removeFaces      snappyRefineMesh
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/advanced$ cd ..
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$ cd conversion/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/conversion$ ls
ansysToFoam  fireToFoam          foamToFireMesh  gmshToFoam       netgenNeutralToFoam  vtkUnstructuredToFoam
ccm          fluent3DMeshToFoam  foamToStarMesh  ideasUnvToFoam   plot3dToFoam         writeMeshObj
cfx4ToFoam   fluentMeshToFoam    foamToSurface   kivaToFoam       star4ToFoam
datToFoam    foamMeshToFluent    gambitToFoam    mshToFoam        tetgenToFoam
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/conversion$ cd ..
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$ cd generation/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/generation$ ls
Allwmake  blockMesh  extrude  extrude2DMesh  foamyMesh  snappyHexMesh
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/generation$ cd ..
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh$ cd manipulation/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/applications/utilities/mesh/manipulation$ ls
attachMesh     deformedGeom          mirrorMesh        orientFaceZone  setSet            stitchMesh
autoPatch      flattenMesh           moveDynamicMesh   polyDualMesh    setsToZones       subsetMesh
checkMesh      insideCells           moveEngineMesh    refineMesh      singleCellMesh    topoSet
createBaffles  mergeMeshes           moveMesh          renumberMesh    splitMesh         transformPoints
createPatch    mergeOrSplitBaffles   objToVTK          rotateMesh      splitMeshRegions  zipUpMesh
```

Step 3

Definition of the boundary conditions

OpenFOAM structure

- The boundaries are defined in Steps 1 and 2, and are written in *constant/polyMesh/boundary*
- The boundaries must be assigned a type
  - *empty*, *wall*, *patch*, *symmetryPlane*, *cyclic*, etc...
- The boundary conditions are then specificied for each field (e.g. p, U, etc), in the time directories
- More on the boundary conditions in the lecture in Week 7.

Step 3

OpenFOAM boundary types

```
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields/constraint$ ls
cyclic      cyclicAMI   empty       jumpCyclicAMI              processor        symmetry       wedge
cyclicACMI  cyclicSlip  jumpCyclic  nonuniformTransformCyclic  processorCyclic  symmetryPlane
```

OpenFOAM boundary conditions

```
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields$ ls
basic  constraint  derived  doc  fvPatchField
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields$ cd basic/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields/basic$ ls
basicSymmetry  coupled         extrapolatedCalculated  fixedValue  sliced     zeroGradient
calculated     directionMixed  fixedGradient           mixed       transform
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields/basic$ cd ..
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields$ cd derived/
josh@josh:~/OpenFOAM/OpenFOAM-v1806/src/finiteVolume/fields/fvPatchFields/derived$ ls
activeBaffleVelocity                 phaseHydrostaticPressure
activePressureForceBaffleVelocity    plenumPressure
advective                            pressureDirectedInletOutletVelocity
codedFixedValue                      pressureDirectedInletVelocity
codedMixed                           pressureInletOutletParSlipVelocity
cylindricalInletVelocity             pressureInletOutletVelocity
fan                                  pressureInletUniformVelocity
fanPressure                          pressureInletVelocity
fixedFluxExtrapolatedPressure        pressureNormalInletOutletVelocity
fixedFluxPressure                    pressurePIDControlInletVelocity
fixedInternalValueFvPatchField       prghPressure
fixedJump                            prghTotalHydrostaticPressure
fixedJumpAMI                         prghTotalPressure
```

Step 4

Definition of the initial conditions

OpenFOAM structure

- The initial values for all variables must be set
- The initial conditions are specificied for each field (e.g. p, U, etc), in the *0* time directory
- For complex intial conditions (e.g. non-uniform fields), a number of utilities are available
  - *setFields*, *mapFields*, etc..
  - See *OpenFOAMxxx/applicaitons/utilities/preProcessing*

Step 5

Simulation

OpenFOAM structure

- Simulating a case is performed by running a solver
  - See *OpenFOAMxxx/applicaitons/solvers*
- Important parameters to control the simulation are defined in *system/controlDict*
- Important parameters for the numerical schemes and solvers are defined in *system/fvSchemes* and *system/fvSolution*
- More on the Solvers and Settings in the lecture in Week 7

Step 6

Post Processing

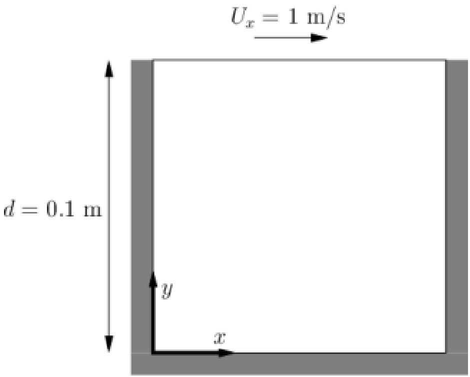
OpenFOAM structure

- Some information it written to the logs.
  - This can be extracted using the *foamLog* utility
- Information can be extracted from the variables saved in each time step directory
  - For visualisation, we have already looked at using ParaFoam in the labs
  - However, saving all the variables at every time step can over fill the memory, as well as slowing the simualtion down
- Functions cam be defined in *system/controlDict* to extract specific quantities of interest at specific sampling frequencies
- More on Post Processing in the lecture in Week 9

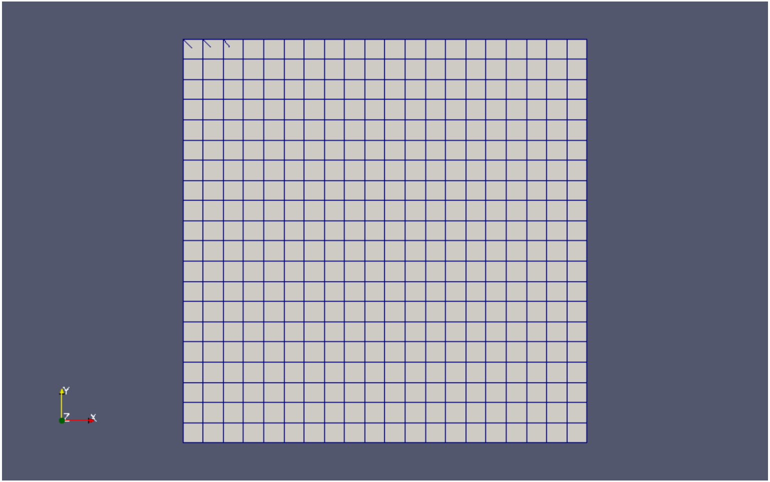
Example : Lid-driven cavity tutorial

Let's walk through the tutorial from the first lab sessions...

Lid-driven cavity – 1. Geometry



Lid-driven cavity – 2. Mesh



Lid-driven cavity – cont..

1 and 2 – Implementation

In this case, the geometry and mesh hare both created with the *blockMesh* utility, which we control using the *blockMeshDict* located in the *system* directory.

Lid-driven cavity – 3. Boundary conditions

Boundary types and conditions

- The boundaries are found in *constant/polyMesh* dirərctory.
- In this case we set them using the *blockMesh* utility.
- We set the boundary condition for each field vairable in the *0* directory.

Lid-driven cavity – 4. Initial conditions

Initial conditions

- In this case we set the initial condition for each field vairable in the *0* directory.

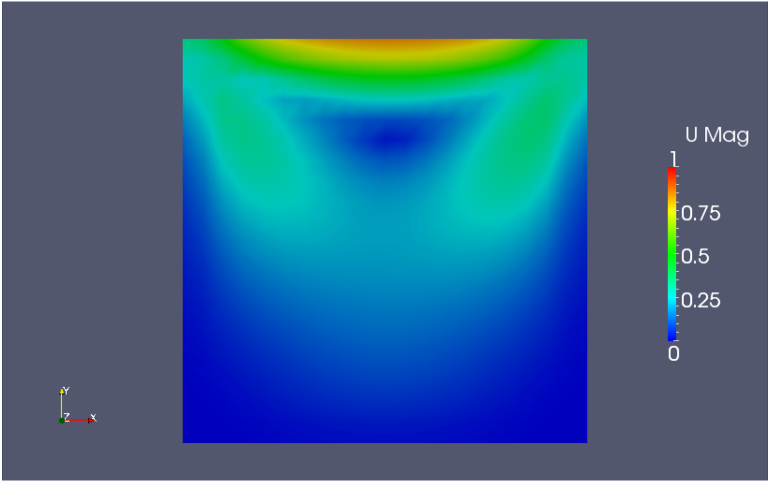
Lid-driven cavity – 5. Simulation

Simulation

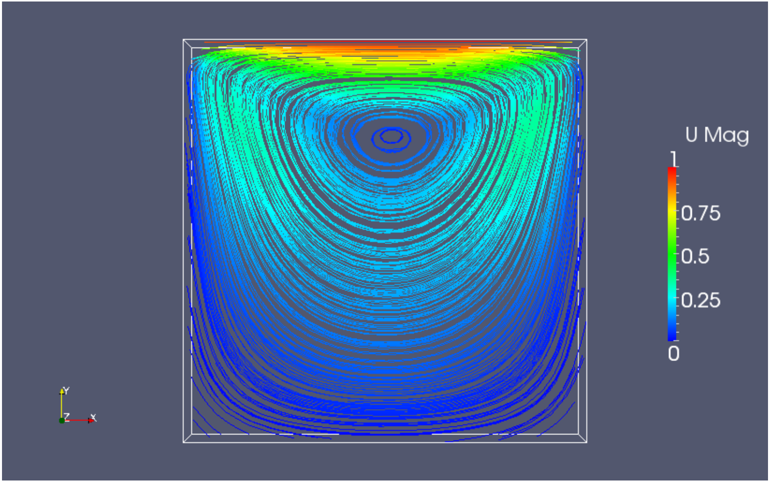
- We are using the *icoFoam* solver for this case
- The settings for the simulation can be found in the *system/controlDict* file
- The settings for the solver can be found in the *system/fvSchemes* and *system/fvSolutions* files

Lid-driven cavity – Postprocessing

Velocity



Streamlines



Questions?

Thanks for your attention!