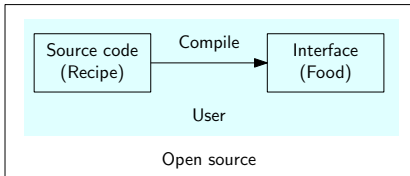# Introduction to `OpenFOAM`

Vachan Potluri

April 2023

What

- ▶ CFD software (but without GUI)
- ▶ **Open** source **F**ield **O**peration **A**nd **M**anipulation
- ▶ Open source $\implies$ source code is given to user

What

- ▶ CFD software (but without GUI)
- ▶ **Open** source **F**ield **O**peration **A**nd **M**anipulation
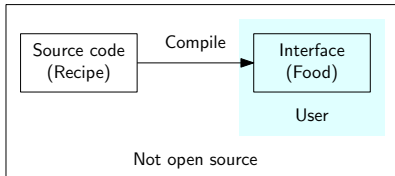- ▶ Open source $\implies$ source code is given to user

What

- ▶ CFD software (but without GUI)
- ▶ **Open** source **F**ield **O**peration **A**nd **M**anipulation
- ▶ Open source $\implies$ source code is given to user

No GUI $\implies$ hard to learn

Why

- ▶ Free
- ▶ Fast
- ▶ User customisable: solve any equation you desire

What

- CFD software (but without GUI)
- **Open** source **F**ield **O**peration **A**nd **M**anipulation
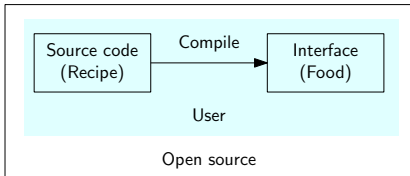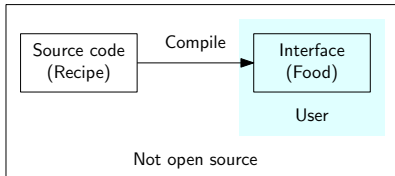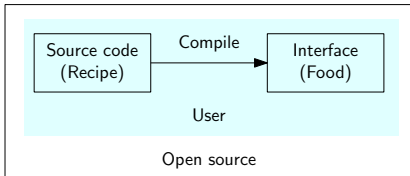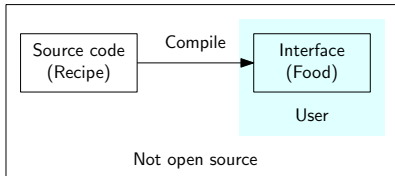- Open source $\implies$ source code is given to user

No GUI $\implies$ hard to learn

Why
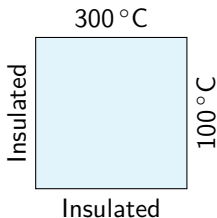
- Free
- Fast
- User customisable: solve any equation you desire

Let's jump right away into doing some simulations

Case 1
Heat conduction in a square plate

$$\frac{\partial T}{\partial t} = D_T \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

Other settings

- $D_T = 1\,\mathrm{m^2/s}$
- $L = 2\,\mathrm{m}$
- End time $10\,\mathrm{s}$

Case 1
Heat conduction in a square plate

$$\frac{\partial T}{\partial t} = D_T \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

Other settings
- $D_T = 1\,\mathrm{m}^2/\mathrm{s}$
- $L = 2\,\mathrm{m}$
- End time $10\,\mathrm{s}$

- laplacianFoam is the "solver" to be used for heat conduction equation

Case 1
Heat conduction in a square plate
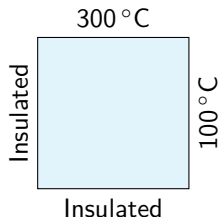
$$\frac{\partial T}{\partial t} = D_T \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$
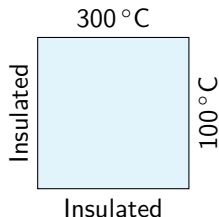

300 °C
Insulated
100 °C
Insulated

Other settings

- $D_T = 1\,\mathrm{m^2/s}$
- $L = 2\,\mathrm{m}$
- End time $10\,\mathrm{s}$

- `laplacianFoam` is the "solver" to be used for heat conduction equation
- Visualise using `paraFoam -builtin`
  - Mesh representation
  - Data array selection
  - Navigating times
  - Changing color map

OpenFOAM's simulation setup

Case structure for `laplacianFoam`

```
case_name/
├─ 0/
│  └─ T
├─ constant/
│  ├─ physicalProperties
│  └─ polyMesh/*
└─ system/
   ├─ controlDict
   ├─ fvSchemes
   └─ fvSolution
```

► Every simulation is setup using certain "setting" files

► These files are grouped into 3 folders: `0/`, `constant/` and `system/`

► All the setting files are text files

OpenFOAM's simulation setup

Case structure for laplacianFoam

```
case_name/
├─ 0/
│  └─ T
├─ constant/
│  ├─ physicalProperties
│  └─ polyMesh/*
└─ system/
   ├─ controlDict
   ├─ fvSchemes
   └─ fvSolution
```

- ▶ Every simulation is setup using certain "setting" files

- ▶ These files are grouped into 3 folders: 0/, constant/ and system/

- ▶ All the setting files are text files

- ▶ Only mandatory files are shown here, there can be additional files also

- ▶ When a simulation is done, OpenFOAM generates corresponding time files

- ▶ We will learn about these files by doing some variations of the square plate simulation

Case 1, variation 1
Change the thermal diffusivity

- In `constant/physicalProperties`, you can change D_T

Case 1, variation 1
Change the thermal diffusivity

- In `constant/physicalProperties`, you can change D_T

- Things to note
  - FoamFile "header"
  - Units of D_T
  - Units in OpenFOAM are specified using powers of 7 fundamental units: [kg m s K mol A cd]
    - [1 1 -2 0 0 0 0] is $kg\,m/s^2 \implies$ force
    - [0 0 1 0 0 1 0] is $As \implies$ charge
    - [1 2 -2 -1 -1 0 0] is $J/mol\,K \implies$ universal gas constant

Case 1, variation 1
Change the thermal diffusivity

▶ In `constant/physicalProperties`, you can change D_T

▶ Things to note
  • `FoamFile` "header"
  • Units of D_T
  • Units in `OpenFOAM` are specified using powers of 7 fundamental units:
    [kg m s K mol A cd]
      ◦ [1 1 -2 0 0 0 0] is kg m/s^2 $\implies$ force
      ◦ [0 0 1 0 0 1 0] is A s $\implies$ charge
      ◦ [1 2 -2 -1 -1 0 0] is J/mol K $\implies$ universal gas constant

▶ Try
  • Reduce D_T to $0.01\,\text{m}^2/\text{s}$ and see the solution evolution
  • Can you guess whether the solution will evolve faster or slower?

▶ Tips
  • `foamListTimes -rm` deletes all time folders other than 0/
  • Reload files in `ParaView` by right-clicking

Case 1, variation 2
Change the boundary conditions

- ▶ In 0/T you can set initial condition (IC) and boundary conditions (BCs)

- ▶ Things to note
  - internalField $\implies$ IC
  - Names of different boundaries
  - type of different boundaries $\implies$ type of BC

Case 1, variation 2
Change the boundary conditions

- ▶ In `0/T` you can set initial condition (IC) and boundary conditions (BCs)

- ▶ Things to note
  - `internalField` $\implies$ IC
  - Names of different boundaries
  - `type` of different boundaries $\implies$ type of BC

- ▶ Special BC type empty
  - By default `OpenFOAM` does 3d simulations
    - Check this in `ParaView`
  - empty BC in a direction tells `OpenFOAM` to not consider that direction

Case 1, variation 2
Change the boundary conditions

- ▶ In `0/T` you can set initial condition (IC) and boundary conditions (BCs)

- ▶ Things to note
  - `internalField` $\implies$ IC
  - Names of different boundaries
  - `type` of different boundaries $\implies$ type of BC

- ▶ Special BC type empty
  - By default `OpenFOAM` does 3d simulations
    - Check this in `ParaView`
  - empty BC in a direction tells `OpenFOAM` to not consider that direction

- ▶ Try
  - Change bottom BC to $-100\,^{\circ}$C
  - Change IC
  - Do a 1d simulation by setting top and bottom boundaries as `zeroGradient`
    - Verify the solution using "Plot Over Line" in `ParaView` or `gradTx` value

Case 1, variation 3
Change time settings

- ▶ `system/controlDict` contains all the main controls of the simulation

- ▶ Things to note
  - `startFrom`, `stopAt`
  - `startTime`, `endTime`
  - `deltaT`
  - `writeControl`, `writeInterval`

Case 1, variation 3
Change time settings

- ▶ system/controlDict contains all the main controls of the simulation

- ▶ Things to note
  - startFrom, stopAt
  - startTime, endTime
  - deltaT
  - writeControl, writeInterval

- ▶ For heat conduction equation
  - "Stable" time step value (Δt_s) satisfies $D_T \dfrac{\Delta t_s}{\Delta x^2} < 1$
  - To determine end time (t_e) for steady state:

$$t_e \gg \frac{L^2}{D_T} \implies \text{Steady state reached}$$

Case 1, variation 3
Change time settings

- ▶ `system/controlDict` contains all the main controls of the simulation

- ▶ Things to note
  - startFrom, stopAt
  - startTime, endTime
  - deltaT
  - writeControl, writeInterval

- ▶ For heat conduction equation
  - "Stable" time step value (Δt_s) satisfies $D_T \dfrac{\Delta t_s}{\Delta x^2} < 1$
  - To determine end time (t_e) for steady state:

$$t_e \gg \frac{L^2}{D_T} \implies \text{Steady state reached}$$

- ▶ Try
  - Increase D_T to $10\,\text{m}^2/\text{s}$
  - For correct simulation, time step also has to be changed

Case 1, variation 4
Change the mesh

- ▶ Multiple ways to create mesh in `OpenFOAM`
  - Create in a different software (e.g. ANSYS) and import to `OpenFOAM`
  - Use `OpenFOAM`'s tools to create mesh

Case 1, variation 4
Change the mesh

- ▶ Multiple ways to create mesh in `OpenFOAM`
    - Create in a different software (e.g. ANSYS) and import to `OpenFOAM`
    - Use `OpenFOAM`'s tools to create mesh

- ▶ `blockMesh` is one such tool in `OpenFOAM` to create meshes

Case 1, variation 4
Change the mesh

- ▶ Multiple ways to create mesh in `OpenFOAM`
    - Create in a different software (e.g. ANSYS) and import to `OpenFOAM`
    - Use `OpenFOAM`'s tools to create mesh
- ▶ `blockMesh` is one such tool in `OpenFOAM` to create meshes
- ▶ `blockMesh` reads an additional file `system/blockMeshDict` to create mesh

Case 1, variation 4
Change the mesh

- ▶ Multiple ways to create mesh in `OpenFOAM`
  - Create in a different software (e.g. ANSYS) and import to `OpenFOAM`
  - Use `OpenFOAM`'s tools to create mesh
- ▶ `blockMesh` is one such tool in `OpenFOAM` to create meshes
- ▶ `blockMesh` reads an additional file `system/blockMeshDict` to create mesh
- ▶ `system/blockMeshDict` has 3 components
  1. Specify points or vertices
  2. Create "blocks" using these vertices
  3. Define boundaries using the vertices

- ▶ `OpenFOAM` defines a local coordinate system (LCS) for every block

- ▶ Blocks are created using a list of points (p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7) ordered in a specific way
  1. Point p_0 is the origin of LCS
  2. Line p_0-p_1 is along x_1 direction
  3. Line p_1-p_2 is along x_2 direction
  4. Points p_0-p_3 define plane $x_3 = 0$
  5. Points p_4-p_7 are obtained by translating points p_0-p_3 in x_3 direction

  Suppose a block is defined using (p_0 p_1 p_2 p_3 p_4 p_5 p_6 p_7).
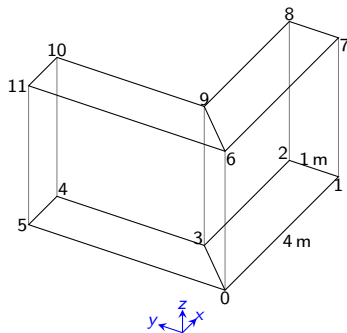 Which one of these is correct?
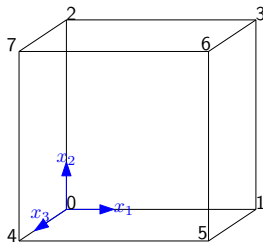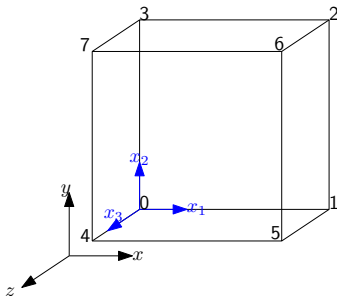
- ▶ Boundaries are defined using a list of points such that right hand thumb rule points outward the block

- ▶ The boundary names in `system/blockMeshDict` and `0/T` must match

- ▶ Boundaries are defined using a list of points such that right hand thumb rule points outward the block

- ▶ The boundary names in `system/blockMeshDict` and `0/T` must match

- ▶ Try
  - Increasing mesh resolution
  - Using mesh grading
  - Changing geometry

- ▶ Tips
  - Run `blockMesh` command to update mesh
  - Run `checkMesh` command to check if the mesh has no issues
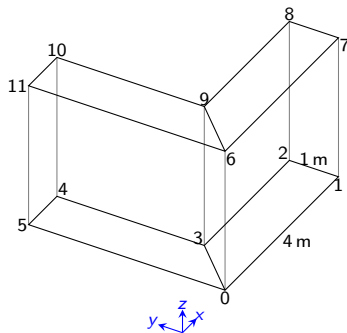 - You can view different mesh regions (e.g. boundaries) individually in `ParaView`

Heat conduction in an L-clamp

Case 2
Heat conduction in an L-clamp

- ▶ This geometry can be constructed using two blocks
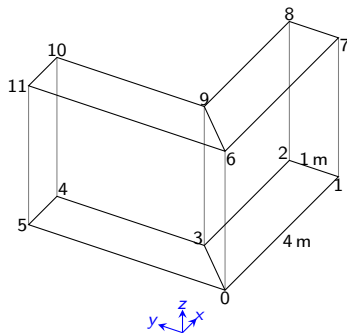   - ❶ Block 1:
      $(p_0 \; p_1 \; p_2 \; p_3 \; p_6 \; p_7 \; p_8 \; p_9)$
    - ❷ Block 2:
      $(p_0 \; p_3 \; p_4 \; p_5 \; p_6 \; p_9 \; p_{10} \; p_{11})$

Case 2
Heat conduction in an L-clamp

▶ BCs
   • One end (p_4 p_5 p_{11} p_{10}) of clamp at $100\,^\circ$C
   • Other end (p_1 p_2 p_8 p_7) at $0\,^\circ$C
   • All other boundaries insulated
   • `empty` BC for top and bottom planes

▶ Tip: you can see the mesh in `ParaView` without running the simulation

▶ This geometry can be constructed using two blocks
   ❶ Block 1:
     (p_0 p_1 p_2 p_3 p_6 p_7 p_8 p_9)
   ❷ Block 2:
     (p_0 p_3 p_4 p_5 p_6 p_9 p_{10} p_{11})

Case 2
Heat conduction in an L-clamp

- This geometry can be constructed using two blocks
  1. Block 1:
     $(p_0\ p_1\ p_2\ p_3\ p_6\ p_7\ p_8\ p_9)$
  2. Block 2:
     $(p_0\ p_3\ p_4\ p_5\ p_6\ p_9\ p_{10}\ p_{11})$
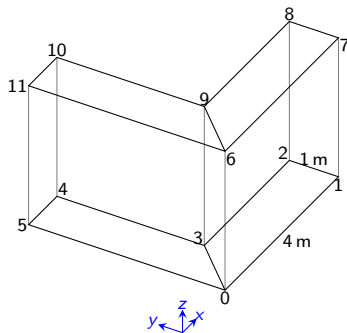
- BCs
  - One end ($p_4\ p_5\ p_{11}\ p_{10}$) of clamp at $100\,^{\circ}$C
  - Other end ($p_1\ p_2\ p_8\ p_7$) at $0\,^{\circ}$C
  - All other boundaries insulated
  - `empty` BC for top and bottom planes
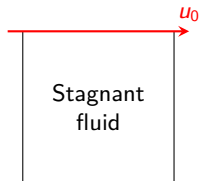- Tip: you can see the mesh in `ParaView` without running the simulation
- Use $D_T = 1\,\mathrm{m^2/s}$; set end time and time step accordingly

$$D_T \frac{\Delta t_s}{\Delta x^2} \leq 1$$

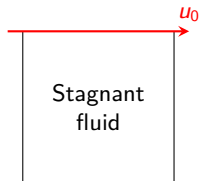$$t_e \gg \frac{L^2}{D_T}$$

Case 3

Lid driven cavity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial(p/\rho)}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial(p/\rho)}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

Case 3
Lid driven cavity

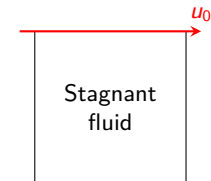$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial(p/\rho)}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial(p/\rho)}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

▶ `icoFoam` is one solver for "transient laminar" incompressible flow

Case 3
Lid driven cavity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial(p/\rho)}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$
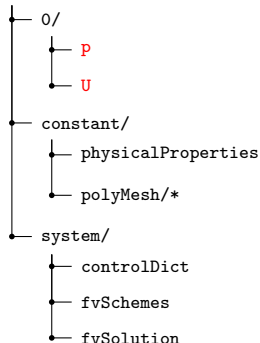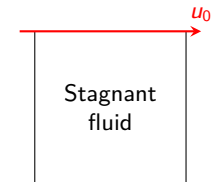
$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial(p/\rho)}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

▶ `icoFoam` is one solver for "transient laminar" incompressible flow

Case 3
Lid driven cavity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial(p/\rho)}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial(p/\rho)}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$
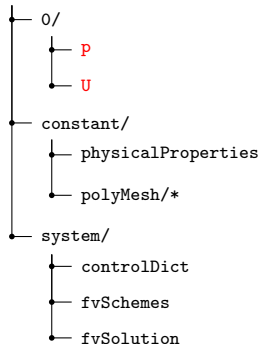
► `icoFoam` is one solver for "transient laminar" incompressible flow
  • `p` in `icoFoam` is p/ρ

Case 3
Lid driven cavity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial(p/\rho)}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$
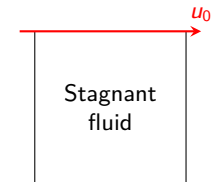
$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial(p/\rho)}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

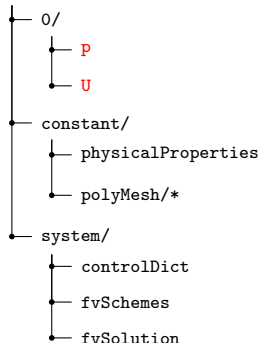▶ `icoFoam` is one solver for "transient laminar" incompressible flow
  • p in `icoFoam` is p/ρ

▶ We will setup this case from scratch using `OpenFOAM`'s "tutorial cases"

▶ And learn some techniques in `ParaView`
  • Extracting 2d slice of a 3d domain
  • Plot over line
  • Visualising streamlines

Case 4

Incompressible flat plate boundary layer

Incompressible flow simulation of air
over a flat plate

$\rho_\infty = 0.097\,19\,\text{kg/m}^3, \ u_\infty = 149.3\,\text{m/s}$

$p_o = 63.71\,\text{kPa}, \ \nu = 1.493 \times 10^{-4}\,\text{m}^2/\text{s}$

Case 4

Incompressible flat plate boundary layer

Incompressible flow simulation of air over a flat plate

$\rho_\infty = 0.097\,19\,\text{kg/m}^3$, $u_\infty = 149.3\,\text{m/s}$

$p_o = 63.71\,\text{kPa}$, $\nu = 1.493 \times 10^{-4}\,\text{m}^2/\text{s}$

- ▶ Estimate BL thickness

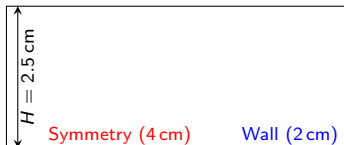$$\frac{\delta}{x} = \frac{5}{\sqrt{\text{Re}_x}} \approx 0.7\,\text{mm}$$

Case 4

Incompressible flat plate boundary layer
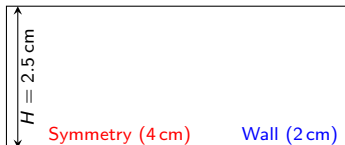
Incompressible flow simulation of air over a flat plate

$\rho_\infty = 0.097\,19\,\text{kg/m}^3$, $u_\infty = 149.3\,\text{m/s}$

$p_o = 63.71\,\text{kPa}$, $\nu = 1.493 \times 10^{-4}\,\text{m}^2/\text{s}$

- Estimate BL thickness

$$\frac{\delta}{x} = \frac{5}{\sqrt{\text{Re}_x}} \approx 0.7\,\text{mm}$$

- Use $\Delta y_w \approx 0.1\,\text{mm}$ and grade the mesh away from wall

$$\frac{H}{\Delta y_w} = \frac{r^n - 1}{r - 1}$$

Case 4
Incompressible flat plate boundary layer

Incompressible flow simulation of air over a flat plate

$\rho_\infty = 0.097\,19\,\mathrm{kg/m^3}$, $u_\infty = 149.3\,\mathrm{m/s}$

$p_o = 63.71\,\mathrm{kPa}$, $\nu = 1.493 \times 10^{-4}\,\mathrm{m^2/s}$

- Estimate BL thickness
$$\frac{\delta}{x} = \frac{5}{\sqrt{\mathrm{Re}_x}} \approx 0.7\,\mathrm{mm}$$

- Use $\Delta y_w \approx 0.1\,\mathrm{mm}$ and grade the mesh away from wall
$$\frac{H}{\Delta y_w} = \frac{r^n - 1}{r - 1}$$

- Also grade in x direction to have nearly square cells at leading edge

Case 4
Incompressible flat plate boundary layer

Incompressible flow simulation of air over a flat plate

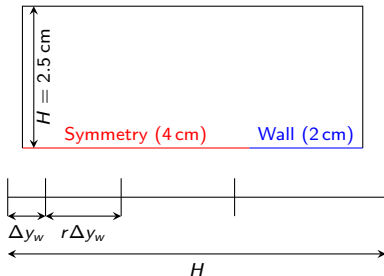$\rho_\infty = 0.097\,19\,\text{kg/m}^3, \ u_\infty = 149.3\,\text{m/s}$

$p_o = 63.71\,\text{kPa}, \ \nu = 1.493 \times 10^{-4}\,\text{m}^2/\text{s}$

▶ Estimate BL thickness

$$\frac{\delta}{x} = \frac{5}{\sqrt{\text{Re}_x}} \approx 0.7\,\text{mm}$$

▶ Use $\Delta y_w \approx 0.1\,\text{mm}$ and grade the mesh away from wall

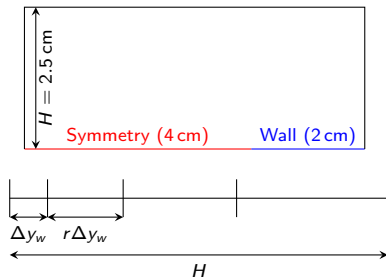$$\frac{H}{\Delta y_w} = \frac{r^n - 1}{r - 1}$$

▶ Also grade in x direction to have nearly square cells at leading edge

▶ Time step and end time:

$$\frac{u\Delta t_s}{\Delta x} < 1, \ t_e \gg \frac{L}{u_\infty}$$

- `OpenFOAM` has many "post-processing" tools
- Post-processing is what you do after a simulation

- ▶ `OpenFOAM` has many "post-processing" tools
- ▶ Post-processing is what you do after a simulation
- ▶ Suppose we wish to compare these results with Blasius solution

$$c_f := \frac{\tau_w}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{0.664}{\sqrt{\mathrm{Re}_x}}$$

$$\delta_2 := \int_0^\infty \frac{u}{u_\infty}\left(1 - \frac{u}{u_\infty}\right) = 0.665\sqrt{\frac{\nu x}{u_\infty}}$$

- ▶ `OpenFOAM` has many "post-processing" tools
- ▶ Post-processing is what you do after a simulation
- ▶ Suppose we wish to compare these results with Blasius solution

$$c_f := \frac{\tau_w}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{0.664}{\sqrt{\mathrm{Re}_x}}$$

$$\delta_2 := \int_0^\infty \frac{u}{u_\infty}\left(1 - \frac{u}{u_\infty}\right) = 0.665\sqrt{\frac{\nu x}{u_\infty}}$$

- ▶ We need two things
  1. Compute shear stress from velocity data
  2. Obtain the simulation data on wall and at a given x location

- ▶ `OpenFOAM` has many "post-processing" tools
- ▶ Post-processing is what you do after a simulation
- ▶ Suppose we wish to compare these results with Blasius solution

$$c_f := \frac{\tau_w}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{0.664}{\sqrt{\text{Re}_x}}$$

$$\delta_2 := \int_0^\infty \frac{u}{u_\infty}\left(1 - \frac{u}{u_\infty}\right) = 0.665\sqrt{\frac{\nu x}{u_\infty}}$$

- ▶ We need two things
  1. Compute shear stress from velocity data
  2. Obtain the simulation data on wall and at a given x location
- ▶ Use `postProcess -func "grad(U)"` for the first

- ▶ `OpenFOAM` has many "post-processing" tools
- ▶ Post-processing is what you do after a simulation
- ▶ Suppose we wish to compare these results with Blasius solution

$$c_f := \frac{\tau_w}{\frac{1}{2}\rho_\infty u_\infty^2} = \frac{0.664}{\sqrt{\mathrm{Re}_x}}$$

$$\delta_2 := \int_0^\infty \frac{u}{u_\infty}\left(1 - \frac{u}{u_\infty}\right) = 0.665\sqrt{\frac{\nu x}{u_\infty}}$$

- ▶ We need two things
  - ❶ Compute shear stress from velocity data
  - ❷ Obtain the simulation data on wall and at a given x location
- ▶ Use `postProcess -func "grad(U)"` for the first
- ▶ For the second, use `postProcess -func -<filename>` with the corresponding file `system/`
  - • Alternatively, `ParaView` can be used to export "Plot Over Line" data

fvSchemes and fvSolution

```
case_name/
├── 0/
│   └── T
├── constant/
│   ├── physicalProperties
│   └── polyMesh/*
└── system/
    ├── controlDict
    ├── fvSchemes
    └── fvSolution
```

fvSchemes and fvSolution

```
case_name/
├── 0/
│   └── T
├── constant/
│   ├── physicalProperties
│   └── polyMesh/*
└── system/
    ├── controlDict
    ├── fvSchemes
    └── fvSolution
```

$$\frac{\partial T}{\partial t} = D_T \nabla^2 T$$

$$\downarrow$$

$$\texttt{AT} = \texttt{b}$$

$$\downarrow$$

Solution

fvSchemes and fvSolution

```
case_name/
├── 0/
│   └── T
├── constant/
│   ├── physicalProperties
│   └── polyMesh/*
└── system/
    ├── controlDict
    ├── fvSchemes
    └── fvSolution
```

$$\frac{\partial T}{\partial t} = D_T \nabla^2 T$$

$\Big\downarrow$ Discretisation

$$AT = b$$

$\Big\downarrow$ Linear algebra solution

Solution

fvSchemes and fvSolution

```
case_name/
   ├── 0/
   │     └── T
   ├── constant/
   │     ├── physicalProperties
   │     └── polyMesh/*
   └── system/
         ├── controlDict
         ├── fvSchemes
         └── fvSolution
```

$$\frac{\partial T}{\partial t} = D_T \nabla^2 T$$

Discretisation

$$\mathtt{AT = b}$$

Linear algebra
solution

Solution

▶ fvSchemes: discretisation procedure

▶ fvSolution: linear algebra solution
procedure

fvSchemes and fvSolution

```
case_name/
├── 0/
│   └── T
├── constant/
│   ├── physicalProperties
│   └── polyMesh/*
└── system/
    ├── controlDict
    ├── fvSchemes
    └── fvSolution
```

$$\frac{\partial T}{\partial t} = D_T \nabla^2 T$$
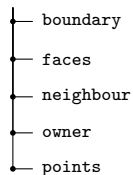
Discretisation

$$AT = b$$

Linear algebra solution

Solution

▶ fvSchemes: discretisation procedure

▶ fvSolution: linear algebra solution procedure

▶ Not a good idea to change these

▶ Simply use the settings from tutorial cases

polyMesh folder

```
polyMesh/
├── boundary
├── faces
├── neighbour
├── owner
└── points
```
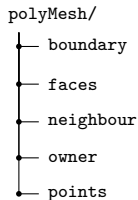
▶ OpenFOAM-specific mesh format
  - Each cell can have any
    number of faces
  - Each face can have any
    number of edges
  - Highly flexible!

polyMesh folder

▶ points: contains a list of all points in the mesh

```
polyMesh/
├── boundary
├── faces
├── neighbour
├── owner
└── points
```

▶ OpenFOAM-specific mesh format
  • Each cell can have any number of faces
  • Each face can have any number of edges
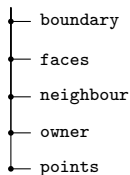 • Highly flexible!
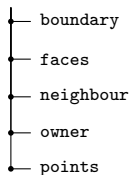
polyMesh folder

polyMesh/

├── boundary

├── faces

├── neighbour

├── owner

└── points

▶ `OpenFOAM`-specific mesh format
  • Each cell can have any number of faces
  • Each face can have any number of edges
  • Highly flexible!

▶ `points`: contains a list of all points in the mesh

▶ `faces`: list of all faces in the mesh
  • Every face is generated using a list of points
  • Ordering is such that normal points away from owner cell
  • For a face common to two cells, `OpenFOAM` assigns cell with smallest id as "owner"

polyMesh folder

polyMesh/
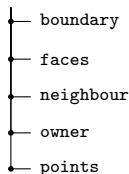├── boundary
├── faces
├── neighbour
├── owner
└── points

▶ OpenFOAM-specific mesh format
  • Each cell can have any number of faces
  • Each face can have any number of edges
  • Highly flexible!

▶ points: contains a list of all points in the mesh

▶ faces: list of all faces in the mesh
  • Every face is generated using a list of points
  • Ordering is such that normal points away from owner cell
  • For a face common to two cells, OpenFOAM assigns cell with smallest id as "owner"

▶ owner: list of cell ids that "own" the faces

▶ neighbour: list of cell ids that are neighbour to the faces

polyMesh folder

polyMesh/
├── boundary
├── faces
├── neighbour
├── owner
└── points

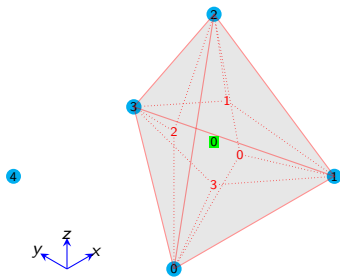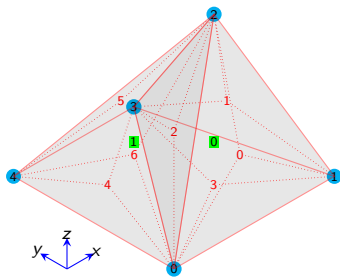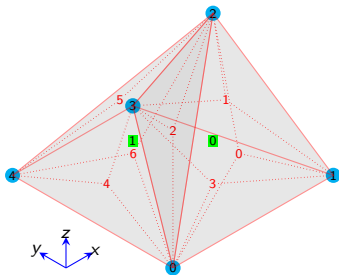- ▶ `OpenFOAM`-specific mesh format
  - Each cell can have any number of faces
  - Each face can have any number of edges
  - Highly flexible!

- ▶ `points`: contains a list of all points in the mesh
- ▶ `faces`: list of all faces in the mesh
  - Every face is generated using a list of points
  - Ordering is such that normal points away from owner cell
  - For a face common to two cells, `OpenFOAM` assigns cell with smallest id as "owner"
- ▶ `owner`: list of cell ids that "own" the faces
- ▶ `neighbour`: list of cell ids that are neighbour to the faces
- ▶ `boundary`: boundary faces information

points	faces	owner	neighbour
(0,0,0) *// 0*	(0,1,2) *// 0*	0	-1
(1,0,0)	(1,3,2)	0	-1
(0,-0.25,1.5)	(2,3,0)	0	1
(0.25,0.5,0.5)	(0,3,1)	0	-1
(0,1,0) *// 4*	(0,4,3)	1	-1
	(2,3,4)	1	-1
	(0,2,4) *// 6*	1	-1
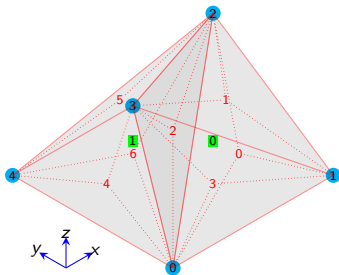
points	faces	owner	neighbour
(0,0,0) *// 0*	(0,1,2) *// 0*	0	-1
(1,0,0)	(1,3,2)	0	-1
(0,-0.25,1.5)	(2,3,0)	0	1
(0.25,0.5,0.5)	(0,3,1)	0	-1
(0,1,0) *// 4*	(0,4,3)	1	-1
	(2,3,4)	1	-1
	(0,2,4) *// 6*	1	-1

▶ boundary file needs to be edited when using empty BC

Resources

- https://openfoam.org/download/windows/: installation on Windows
  - A sure-shot procedure
  - Follow every line of this guide with utmost attention

Resources

- https://openfoam.org/download/windows/: installation on Windows
  - A sure-shot procedure
  - Follow every line of this guide with utmost attention
- `OpenFOAM` has many versions. Two major ones are
  - https://openfoam.org/ (I was using this)
  - https://www.openfoam.com/

Resources

- ▶ https://openfoam.org/download/windows/: installation on Windows
  - A sure-shot procedure
  - Follow every line of this guide with utmost attention
- ▶ `OpenFOAM` has many versions. Two major ones are
  - https://openfoam.org/ (I was using this)
  - https://www.openfoam.com/
- ▶ Both these websites have links to many useful resources
  - Basic linux guide
  - User guide. Also available as pdf in your installation at `$FOAM_INST_DIR/OpenFOAM-10/doc/Guides/`

Resources

- ▶ https://openfoam.org/download/windows/: installation on Windows
  - A sure-shot procedure
  - Follow every line of this guide with utmost attention
- ▶ `OpenFOAM` has many versions. Two major ones are
  - https://openfoam.org/ (I was using this)
  - https://www.openfoam.com/
- ▶ Both these websites have links to many useful resources
  - Basic linux guide
  - User guide. Also available as pdf in your installation at `$FOAM_INST_DIR/OpenFOAM-10/doc/Guides/`
- ▶ User guide is the most resourceful place to start for a beginner
- ▶ Some youtube channels are very helpful
  - Wolf Dynamics `OpenFOAM-9` tutorials (linear, very detailed)
  - CFD basics, CFD intermediate and many other playlists by József Nagy (be careful with versions while following these)

Resources

- ▶ https://openfoam.org/download/windows/: installation on Windows
  - A sure-shot procedure
  - Follow every line of this guide with utmost attention
- ▶ `OpenFOAM` has many versions. Two major ones are
  - https://openfoam.org/ (I was using this)
  - https://www.openfoam.com/
- ▶ Both these websites have links to many useful resources
  - Basic linux guide
  - User guide. Also available as pdf in your installation at `$FOAM_INST_DIR/OpenFOAM-10/doc/Guides/`
- ▶ User guide is the most resourceful place to start for a beginner
- ▶ Some youtube channels are very helpful
  - Wolf Dynamics `OpenFOAM-9` tutorials (linear, very detailed)
  - CFD basics, CFD intermediate and many other playlists by József Nagy (be careful with versions while following these)
- ▶ When you face an error, search for its solution in these Q&A websites
  - https://askubuntu.com/: Ubuntu related issues
  - https://www.cfd-online.com/Forums/openfoam/: an unofficial `OpenFOAM` user forum (some posts are decades old here!)