

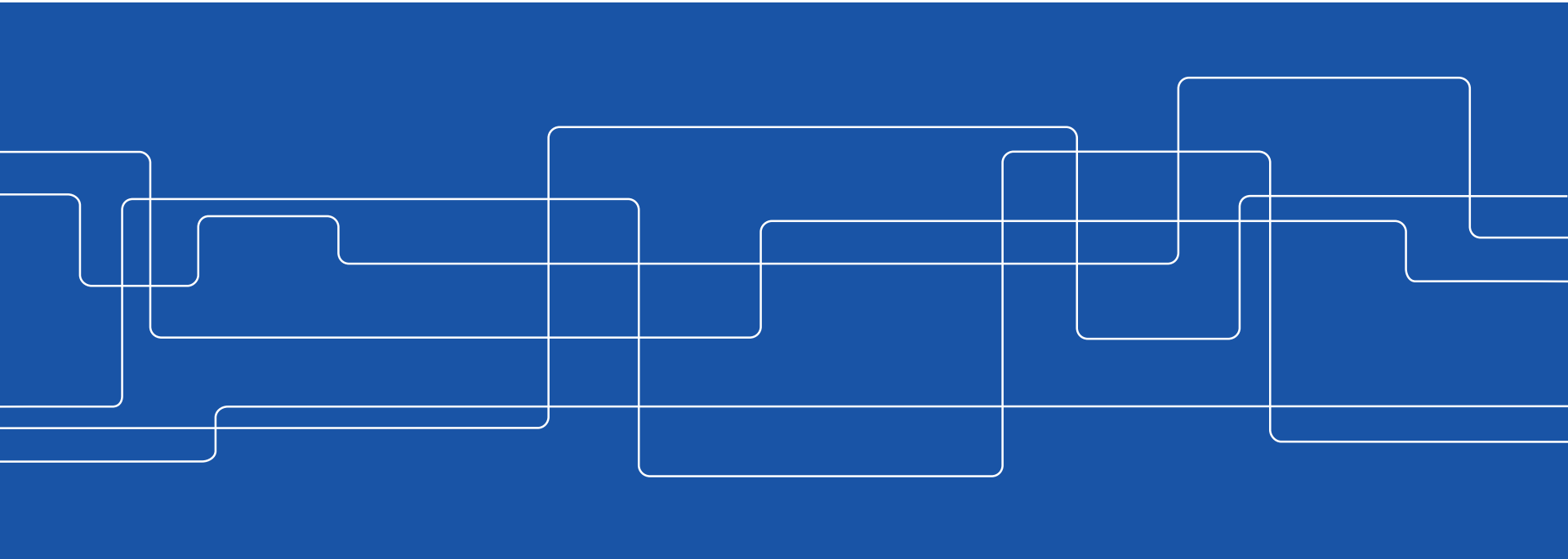


SimEx Group, 29<sup>th</sup> Oct 2020

KTH ROYAL  
INSTITUTE  
OF TECHNOLOGY

# OpenFOAM tutorial Part I

Marco Atzori, Narges Tabatabaei





Plan for today:

Part I: Marco

Part II: Narges



## Plan for today:

### Part I: Marco

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver! (maybe)

### Part II: Narges



# Plan for today:

## Part I: Marco

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver! (maybe)

## Part II: Narges

1. Additional notes on boundary conditions and control dictionary
2. Using finite-volume options
3. Modify a turbulent model



# General information

## First reference:

- “A tensorial approach to computational continuum mechanics using object-oriented techniques”, Weller *et al.* (1998), *Comput. Physics*, **12**, 620

## General methodology:

- OpenFOAM uses the finite-volume formulation, and a combination of different algorithms to solve the Navier-Stokes eq. (*and any other equation you want*).

## Different distributions and many external package:

- OpenFOAM Foundation: <https://openfoam.org/>
- ESI: <https://openfoam.com/>
- Foam Extend: <https://sourceforge.net/projects/foam-extend/>

What we will use today: <https://github.com/OpenFOAM>



## Assuming you have compiled the code\*:

- The last (and optional) step of the instructions was adding to your *.bashrc*:

```
alias of7x='source ~/OpenFOAM/OpenFOAM-7/etc/bashrc '
```

- This defines variables such as:

```
$WM_PROJECT_DIR=~/OpenFOAM/OpenFOAM-7
```

```
$FOAM_TUTORIALS=~/OpenFOAM/OpenFOAM-7/tutorials
```

```
$FOAM_RUN=~/OpenFOAM/marco-7/run
```

- and allow the usage of executables:

```
blockMesh
```

```
icoFoam
```

```
pisoFoam
```

```
simpleFoam
```

```
etc...
```

(\*) Building instructions here:  
<https://github.com/AtzoriMarco/tutorialOF>



## Assuming you have compiled the code\*:

- The last (and optional) step of the instructions was adding to your `.bashrc`:

```
alias of7x='source ~/OpenFOAM/OpenFOAM-7/etc/bashrc '
```

- This defines variables such as:

```
$WM_PROJECT_DIR=~/OpenFOAM/OpenFOAM-7
```

```
$FOAM_TUTORIALS=~/OpenFOAM/OpenFOAM-7/tutorials
```

```
$FOAM_RUN=~/OpenFOAM/marco-7/run
```

**Note: certain variables  
depend on the user name!**

- and allow the usage of executables:

```
blockMesh
```

```
icoFoam
```

```
 pisoFoam
```

```
simpleFoam
```

```
etc...
```

(\*) Building instructions here:  
<https://github.com/AtzoriMarco/tutorialOF>



# Structure of the code

```
~/OpenFOAM/OpenFOAM-7$
```

```
.  
├── Allwmake  
├── applications  
├── bin  
├── COPYING  
├── doc  
├── etc  
├── platforms  
├── README.org  
├── src  
├── test  
├── tutorials  
└── wmake
```





# Structure of the code

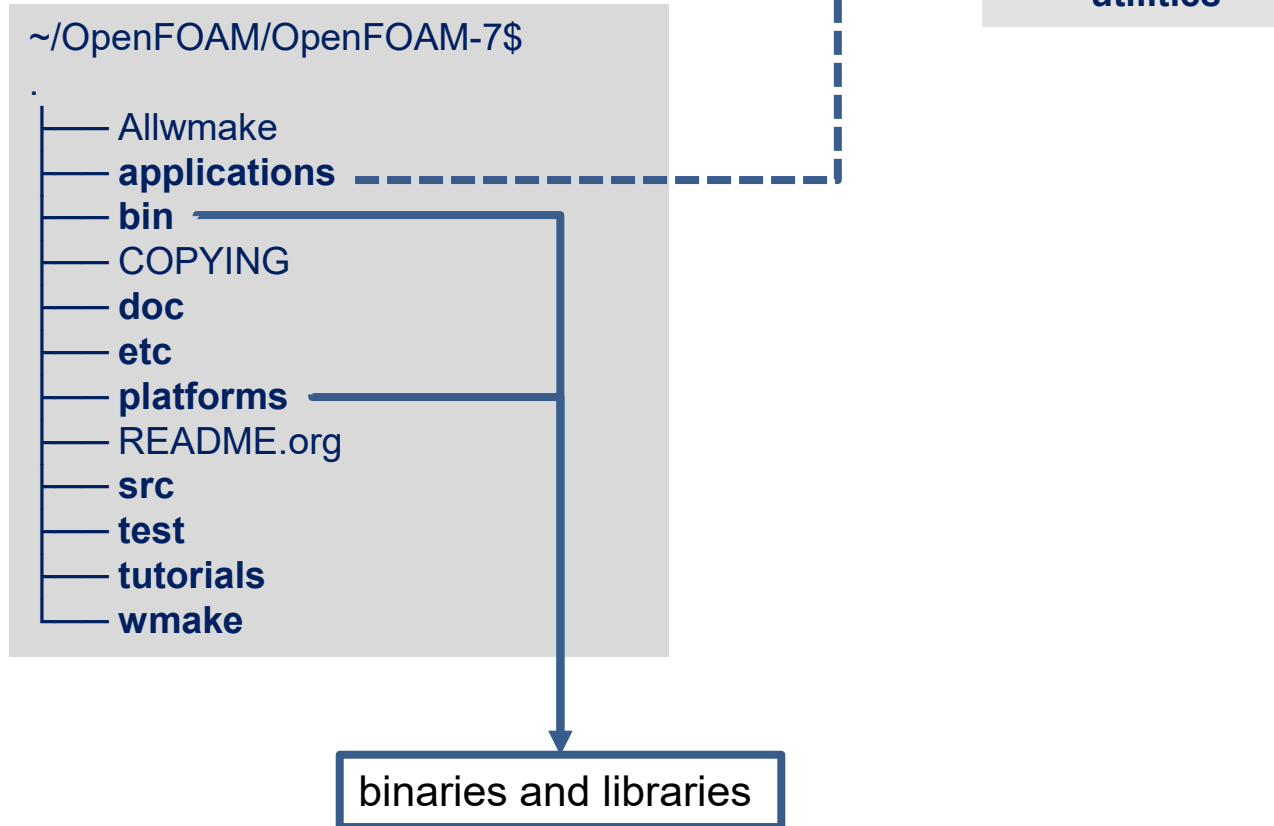
~/OpenFOAM/OpenFOAM-7\$

- Allwmake
- **applications**
- **bin**
- COPYING
- **doc**
- **etc**
- **platforms**
- README.org
- **src**
- **test**
- **tutorials**
- **wmake**

- Allwmake
- **solvers**
- **test**
- **utilities**

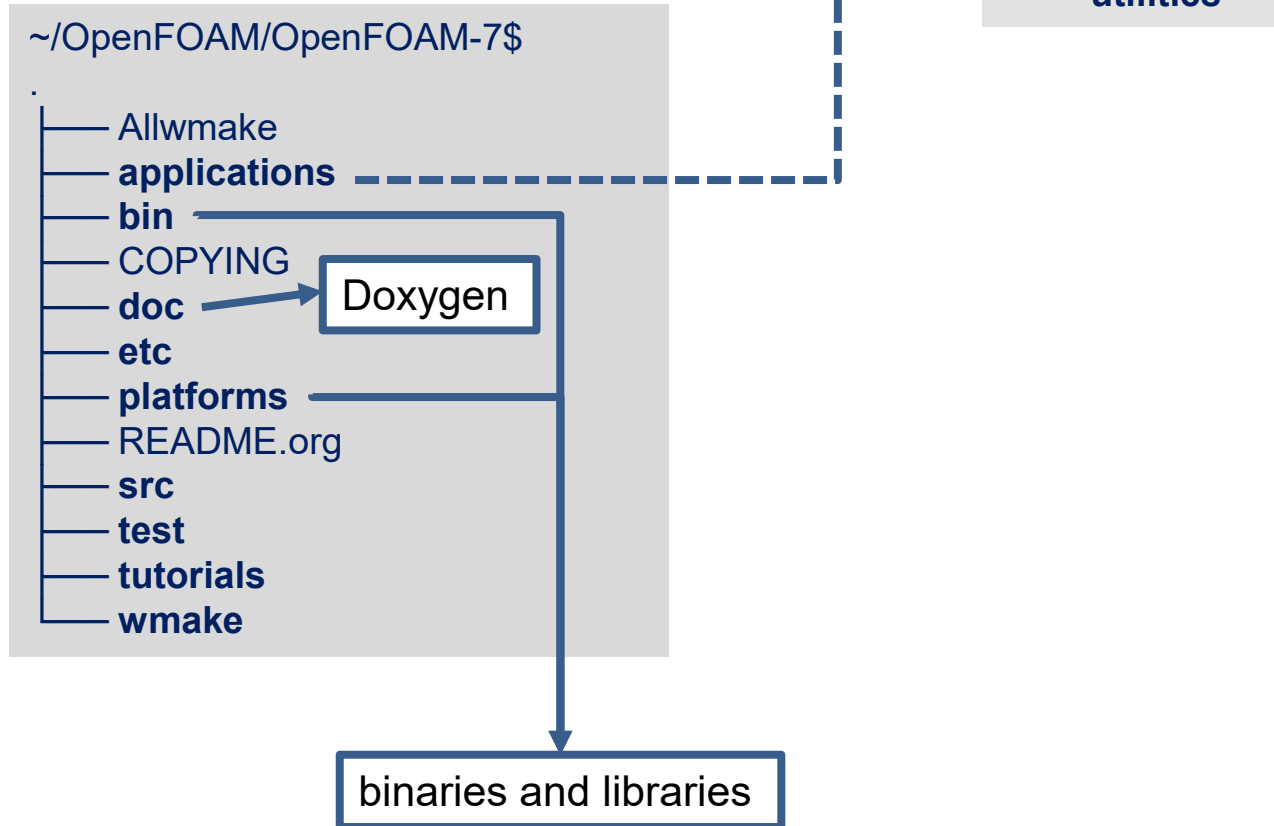


# Structure of the code



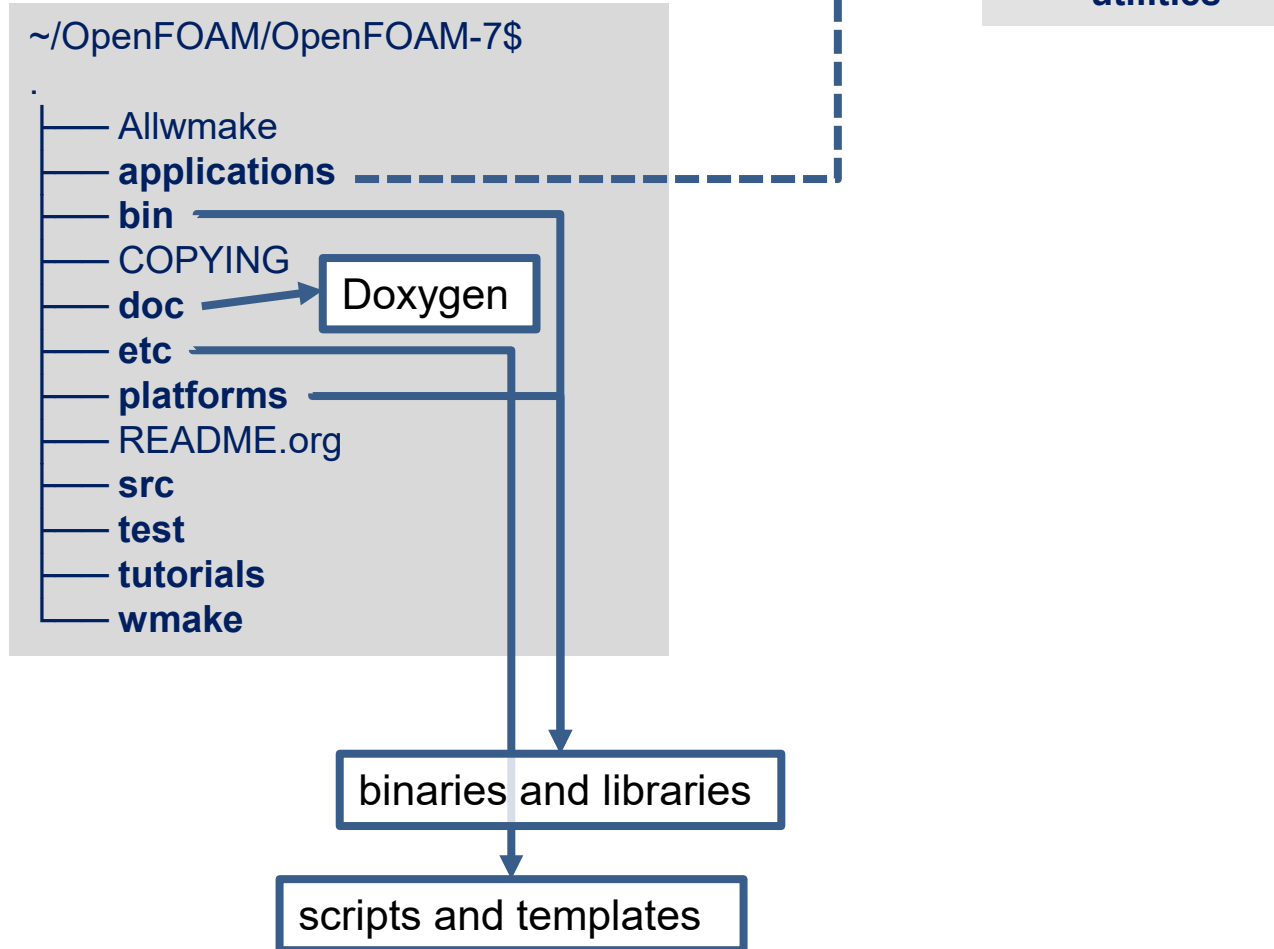


# Structure of the code





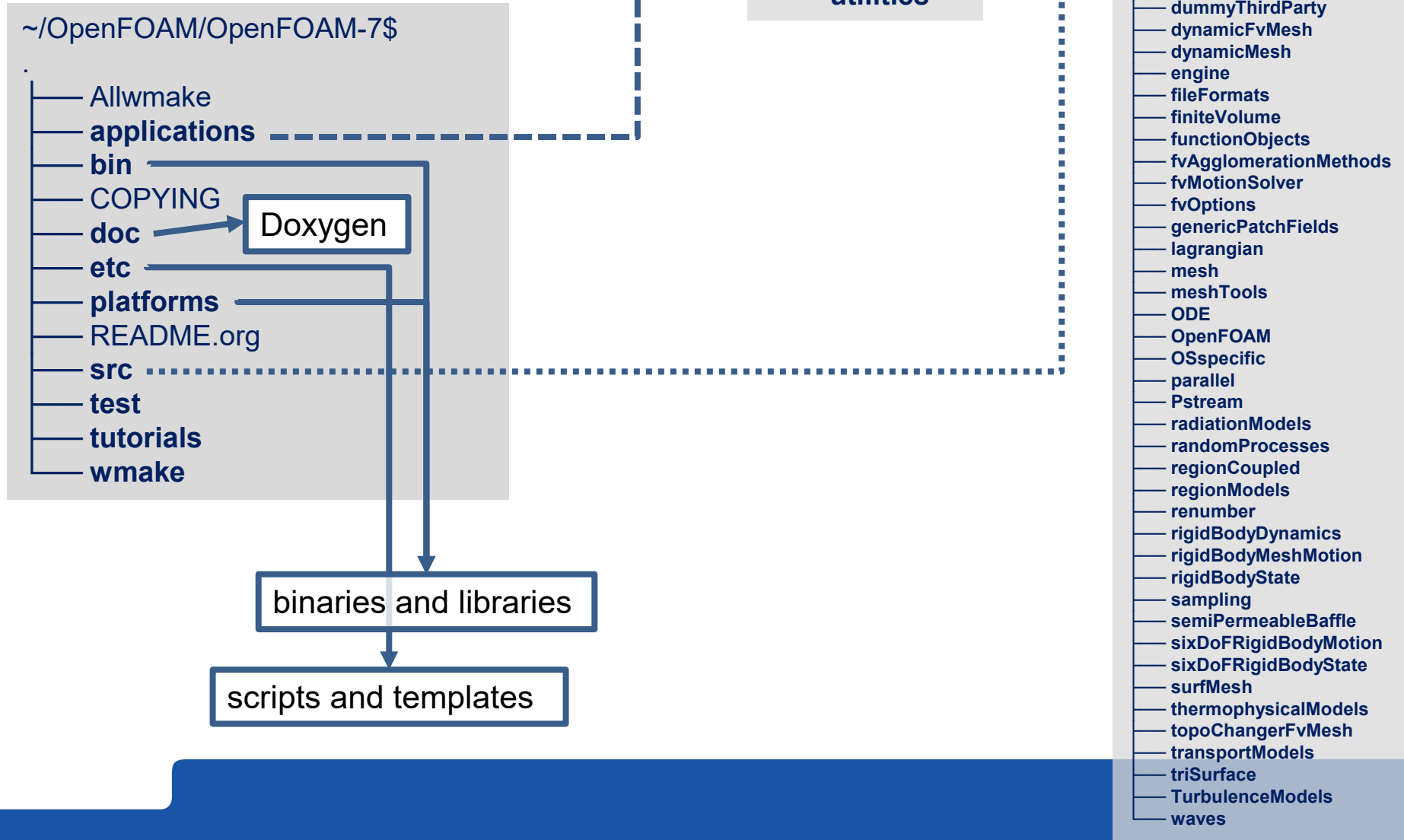
# Structure of the code





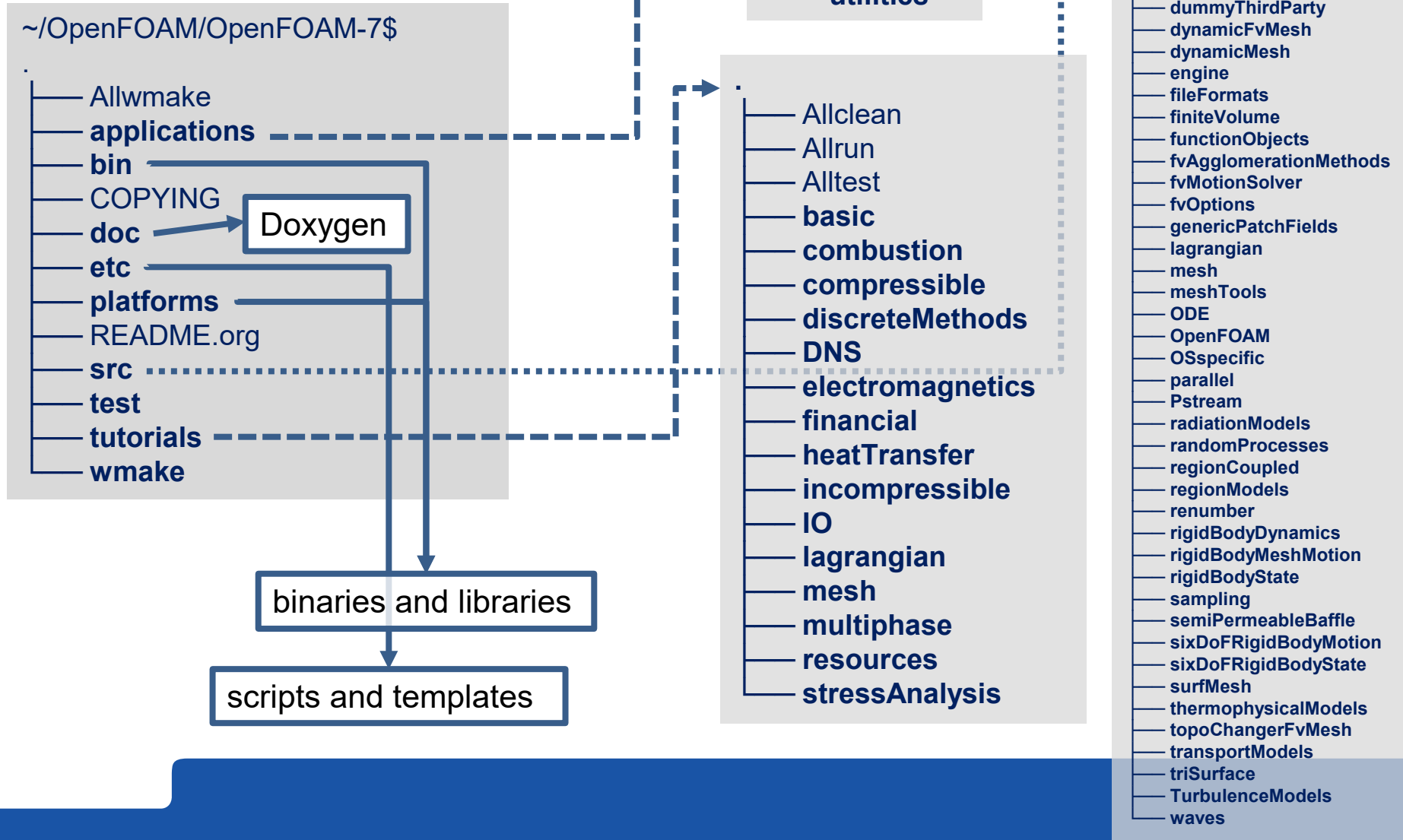
# Structure of the code

## PART 1: Introduction





# Structure of the code



## PART 1: Introduction



## Tutorial *cavity*

- Located in:

```
~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity
```



## Tutorial *cavity*

- Located in:

~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity

Time-dependent N.S. eq.,  
without turbulent models

Incompressible flows

Case name





## Tutorial *cavity*

- Located in:

~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity

Time-dependent N.S. eq.,  
without turbulent models

Incompressible flows

Case name

- It contains:

```
.
├── 0
│   ├── p
│   └── U
├── constant
│   └── transportProperties
├── system
│   ├── blockMeshDict
│   ├── controlDict
│   ├── fvSchemes
│   └── fvSolution
```



# Tutorial *cavity*

- Located in:

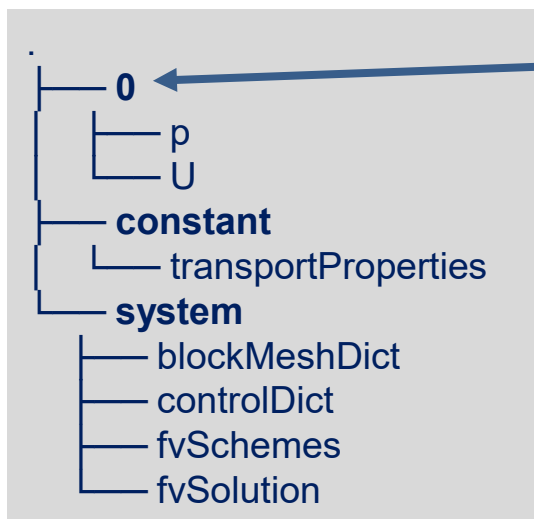
~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity

Time-dependent N.S. eq.,  
without turbulent models

Incompressible flows

Case name

- It contains:



Time folder(s): Eulerian  
(and Lagrangian)  
variables, boundary  
and “initial” conditions

$U_{\text{wall}} = 0$   
etc...



# Tutorial *cavity*

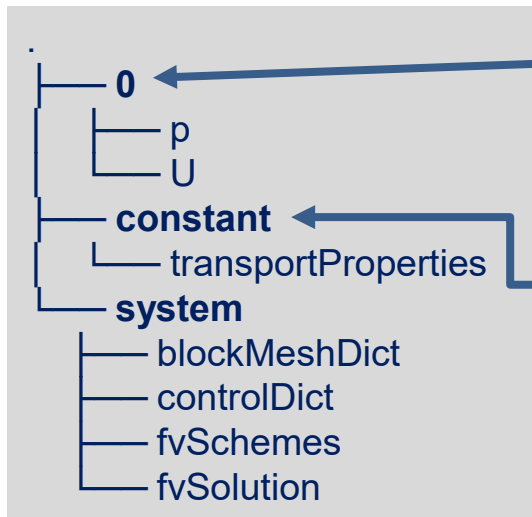
- Located in:

~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity

Incompressible flows

Case name

- It contains:



Time folder(s): Eulerian  
(and Lagrangian)  
variables, boundary  
and “initial” conditions

$U_{\text{wall}} = 0$   
etc...

Physical properties (and  
modelling) (and mesh)

$\nu = 1/Re$



# Tutorial *cavity*

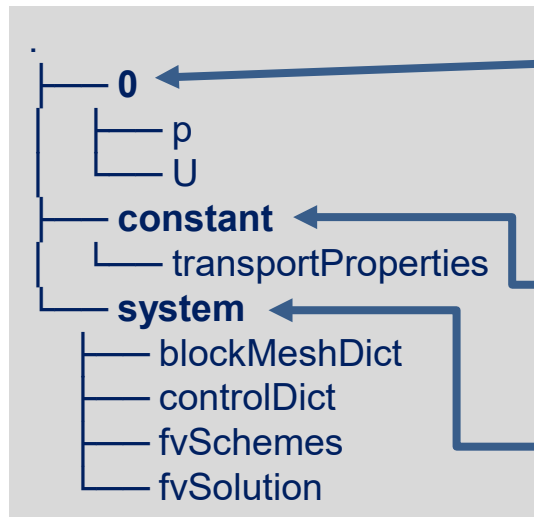
- Located in:

~/OpenFOAM/OpenFOAM-7/tutorials/incompressible/icoFoam/cavity/cavity

Incompressible flows

Case name

- It contains:



Time folder(s): Eulerian  
(and Lagrangian)  
variables, boundary  
and "initial" conditions

$U_{\text{wall}} = 0$   
etc...

Physical properties (and  
modelling) (and mesh)

$\nu = 1/Re$

Options and additional  
dictionaries.

$\Delta t = \dots$   
discretization...  
tolerances...



## Run the case

- Create the mesh:

```
blockMesh
```

- Run the solver:

```
icoFoam
```

- Show the result:

```
paraFoam -builtin
```



## Run the case

- Create the mesh:

`blockMesh`

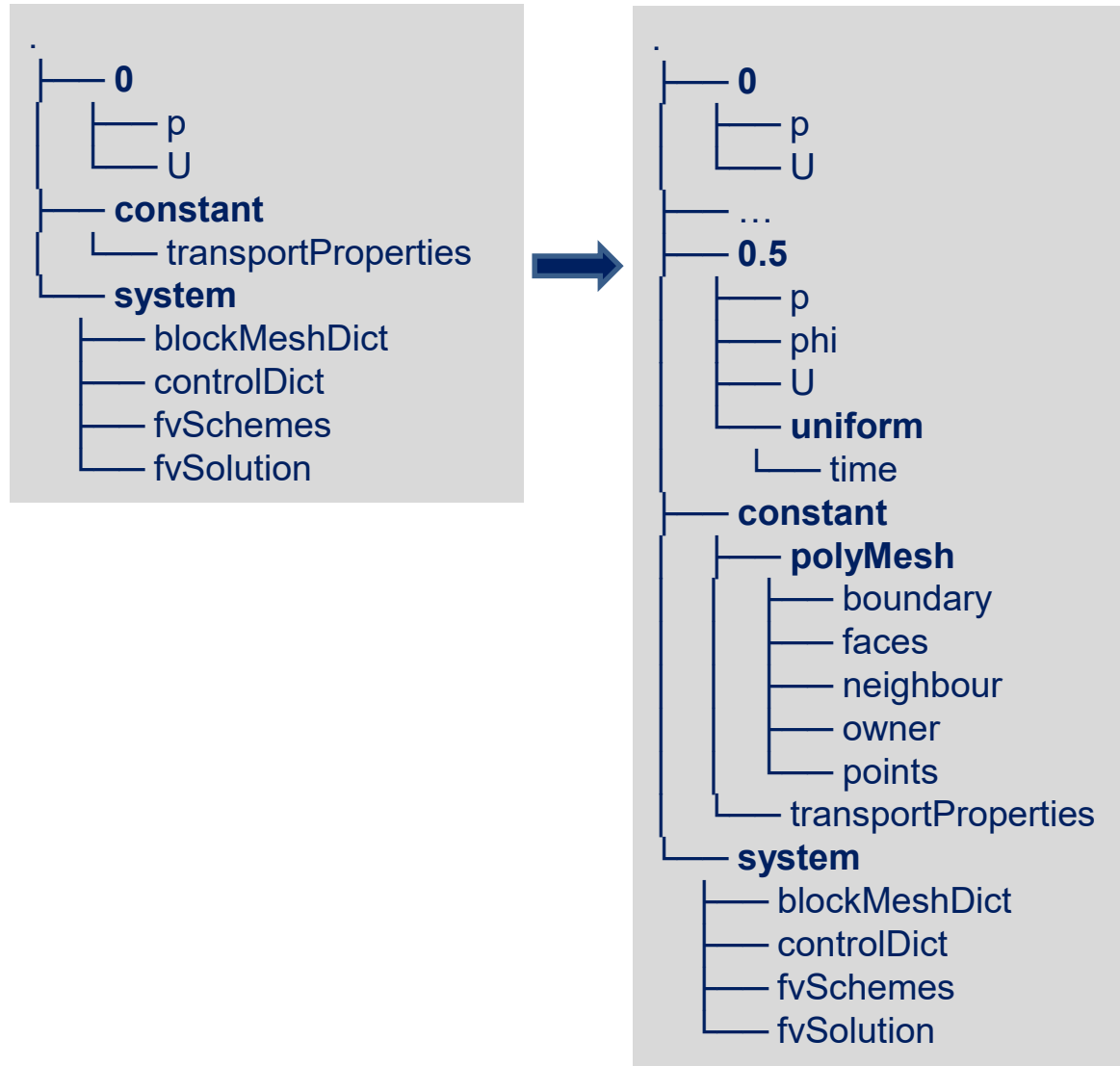
- Run the solver:

`icoFoam`

- Show the result:

`paraFoam -builtin`

## PART 2: Your first case





## Run the case

- Create the mesh:

`blockMesh`

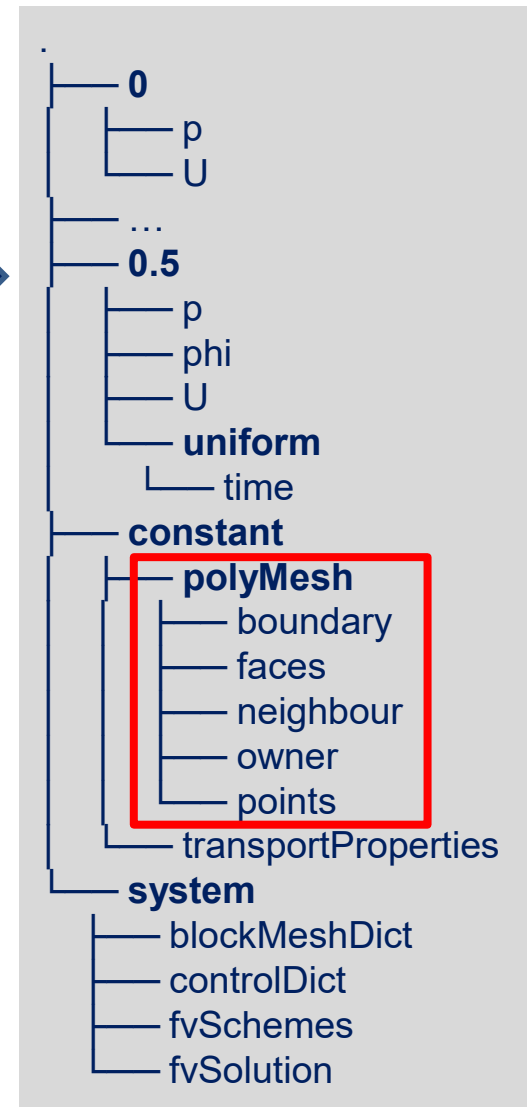
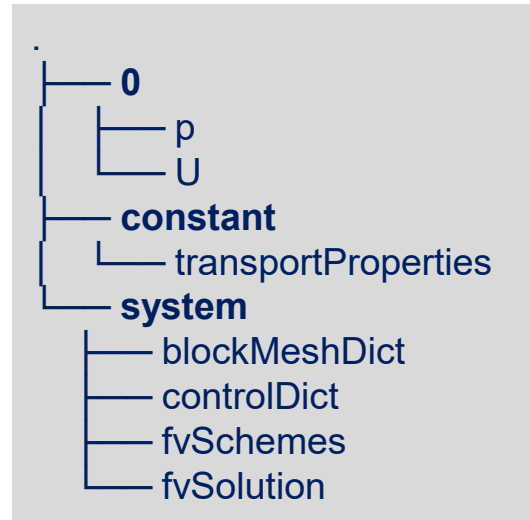


- Run the solver:

`icoFoam`

- Show the result:

`paraFoam -builtin`





## Run the case

- Create the mesh:

blockMesh



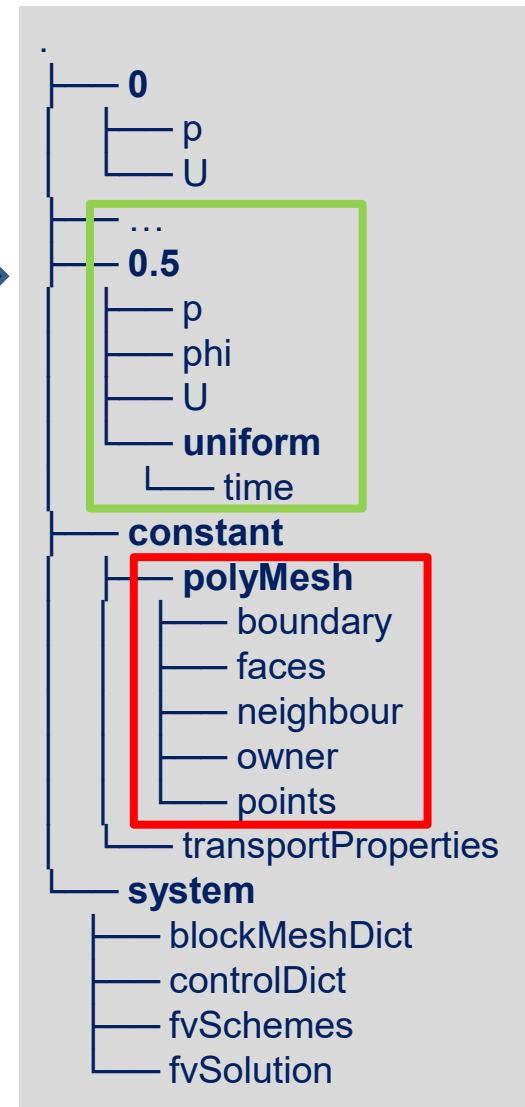
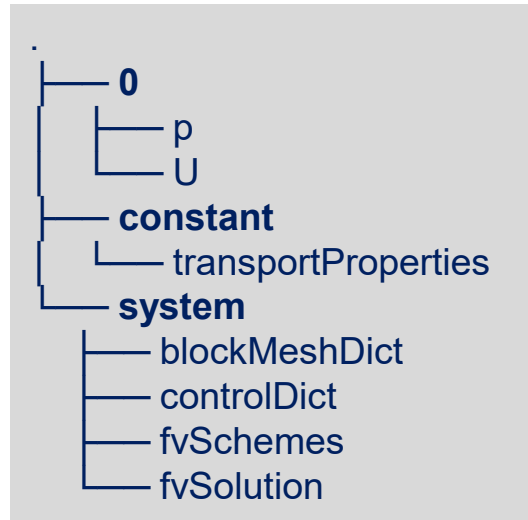
- Run the solver:

icoFoam



- Show the result:

paraFoam -builtin







# Run the case

- Create the mesh:

blockMesh



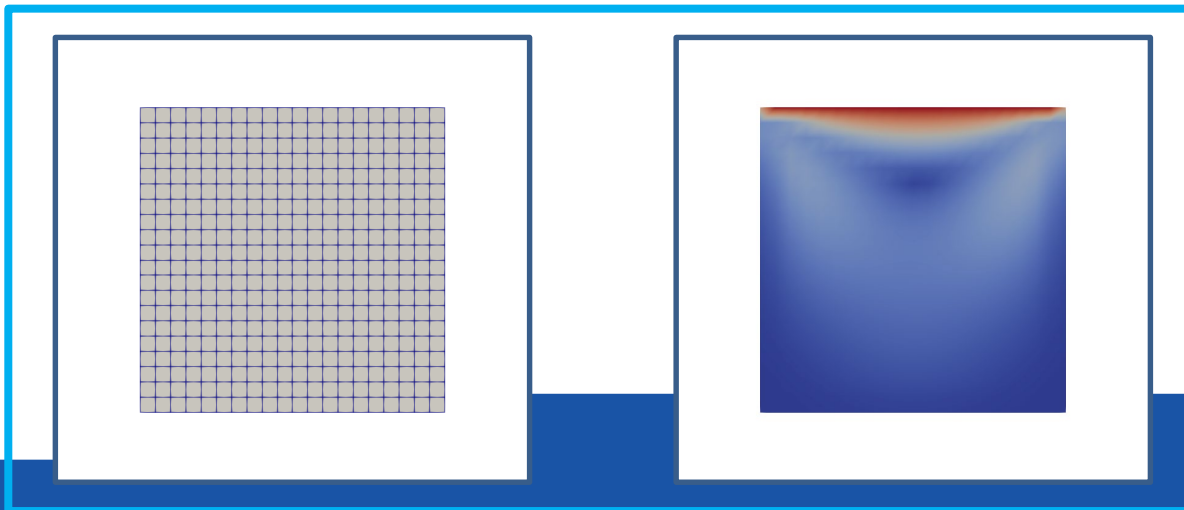
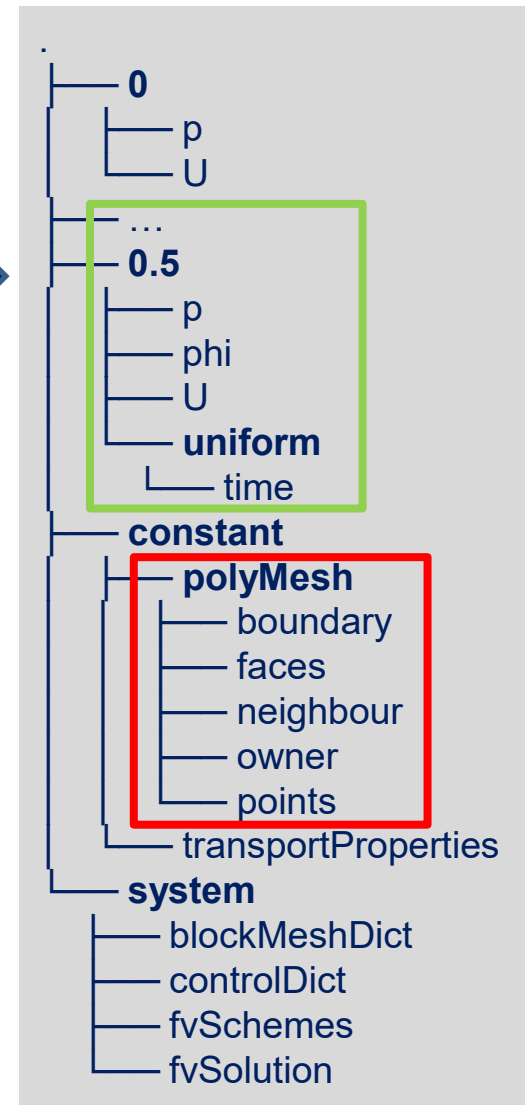
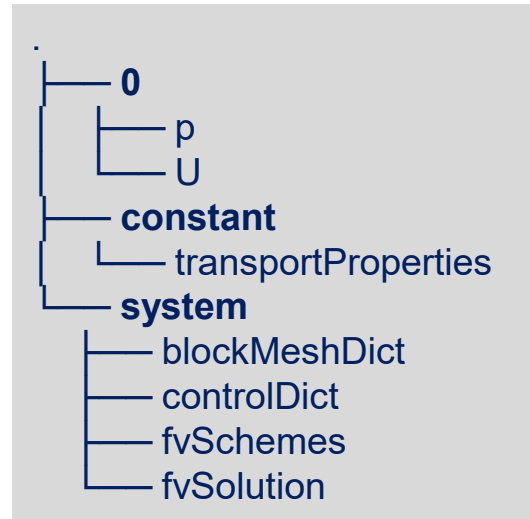
- Run the solver:

icoFoam



- Show the result:

paraFoam -builtin



Figures: (left) grid and (right) velocity magnitude.



## A (slightly) more realistic test case

- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`

```
.  
├── doc  
├── pisoKinematicParcelFoam  
├── pitzDaily2  
└── README.md
```



## A (slightly) more realistic test case

- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`

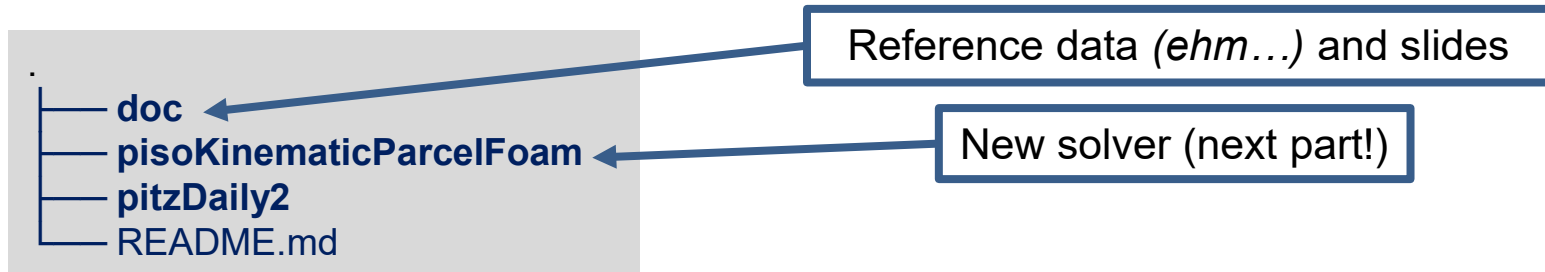
```
.  
├── doc  
├── pisoKinematicParcelFoam  
├── pitzDaily2  
└── README.md
```

Reference data (*ehm...*) and slides



## A (slightly) more realistic test case

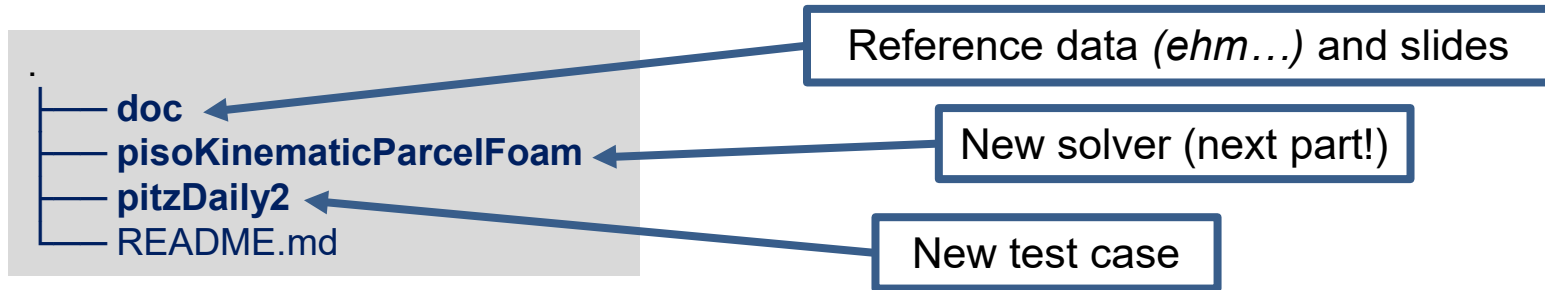
- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`





## A (slightly) more realistic test case

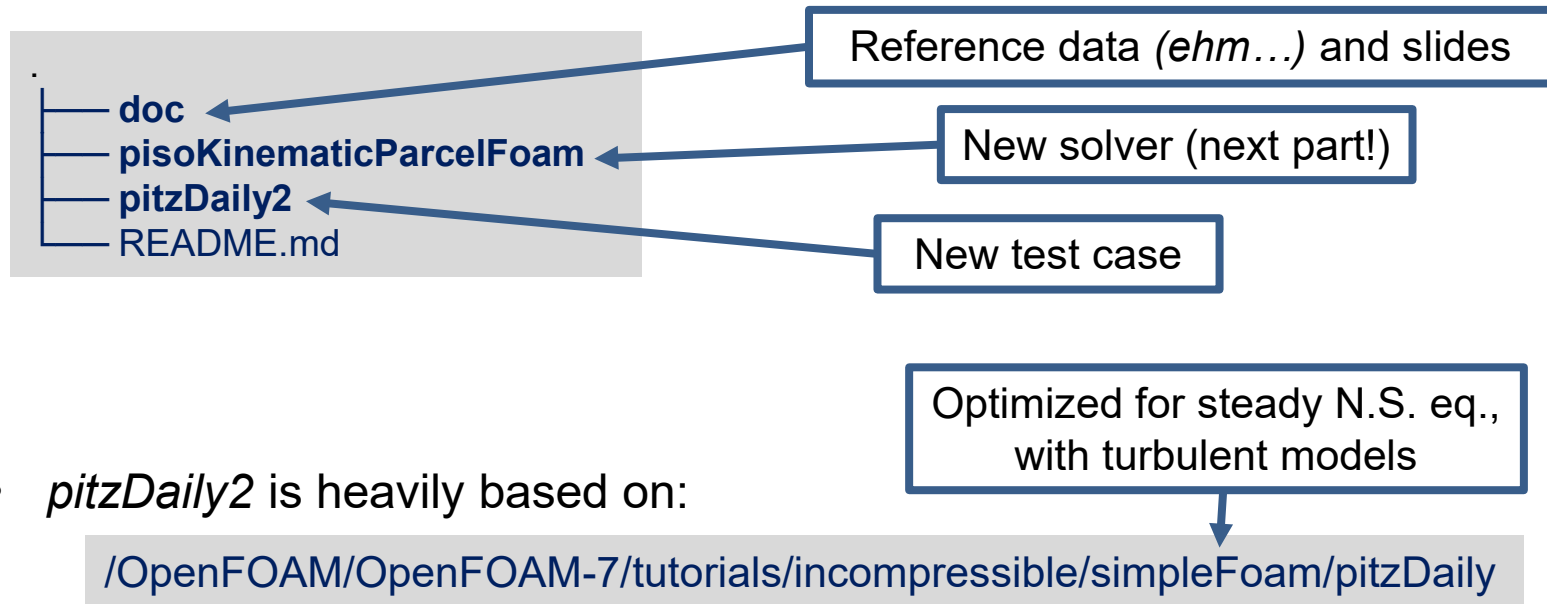
- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`





## A (slightly) more realistic test case

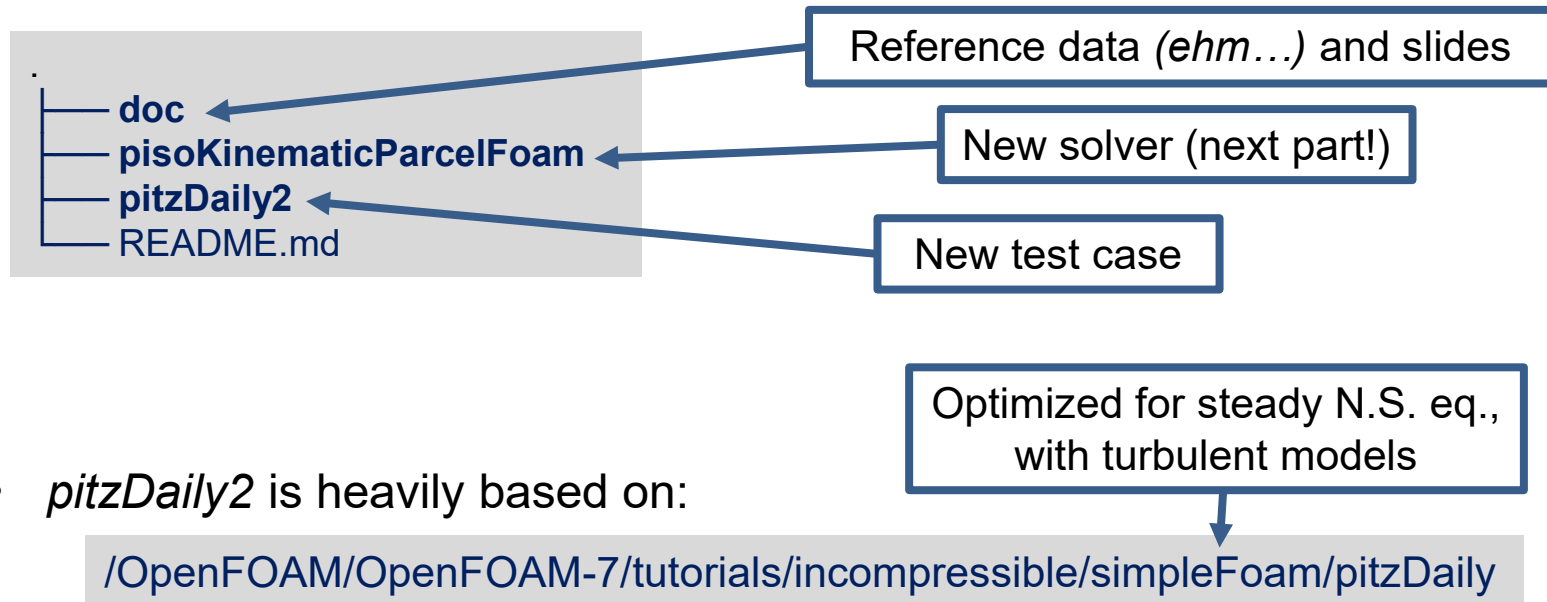
- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`





## A (slightly) more realistic test case

- Clone the repository: `git clone https://github.com/AtzoriMarco/tutorialOF.git`



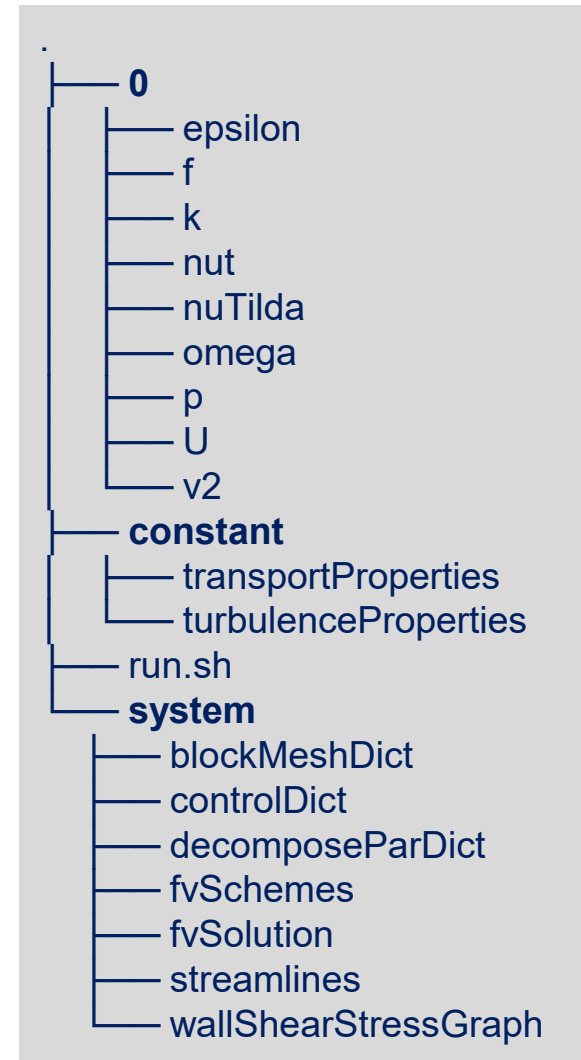
- pitzDaily2* is heavily based on:

`/OpenFOAM/OpenFOAM-7/tutorials/incompressible/simpleFoam/pitzDaily`

- Differences between *pitzDaily2* and *pitzDaily*:
  1. Parallel running;
  2. Lower residual tolerances;
  3. Compute and sample the wall-shear stress.



## Structure of the case







# Structure of the case

```
#!/bin/sh

. $WM_PROJECT_DIR/bin/tools/RunFunctions
application=$(getApplication)

runApplication foamCleanTutorials

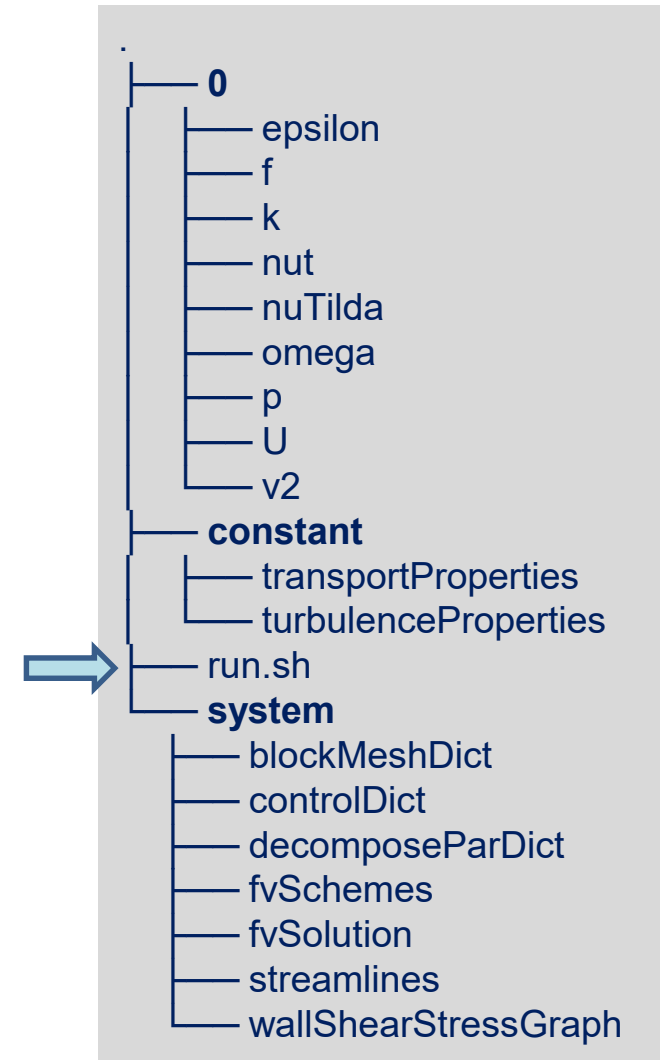
runApplication blockMesh

runApplication decomposePar

runParallel $application

runApplication reconstructPar

simpleFoam -postProcess -latestTime > log.postProcess
```





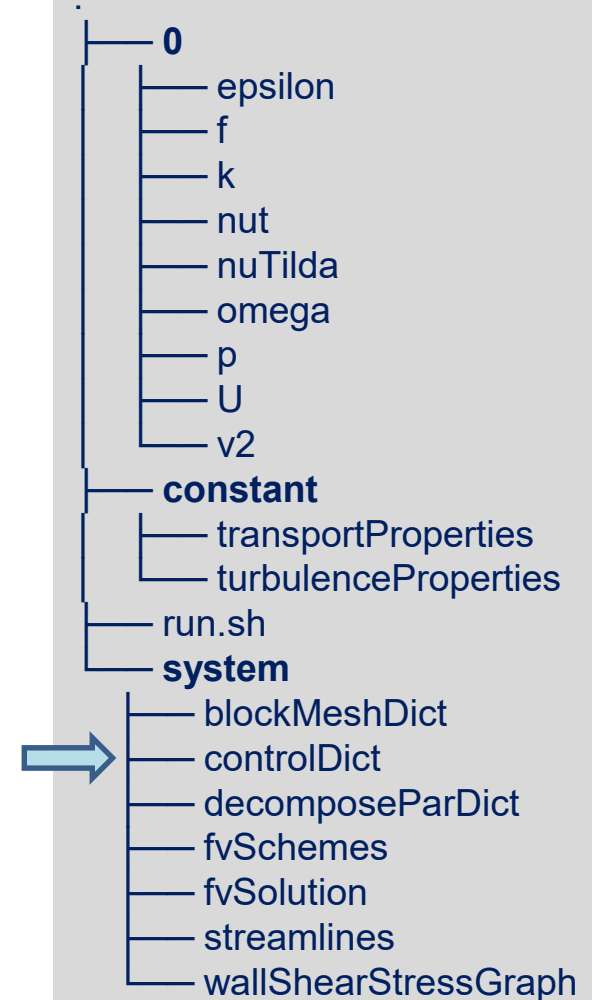
# Structure of the case

```

application      simpleFoam;
startFrom        startTime;
startTime        0;
stopAt           endTime;
endTime          2000;
deltaT           1;
writeControl     timeStep;
writeInterval    100;
purgeWrite       0;
writeFormat      ascii;
writePrecision   6;
writeCompression off;
timeFormat       general;
timePrecision    6;
runTimeModifiable true;

functions
{
    #includeFunc wallShearStress
    #includeFunc wallShearStressGraph
    #includeFunc streamlines
}

```





# Structure of the case

```

application      simpleFoam;
startFrom        startTime;
startTime        0;
stopAt           endTime;
endTime          2000;
deltaT           1;
writeControl      timeStep;
writeInterval     100;
purgeWrite        0;
writeFormat       ascii;
writePrecision    6;
writeCompression  off;
timeFormat        general;
timePrecision     6;
runTimeModifiable true;
  
```

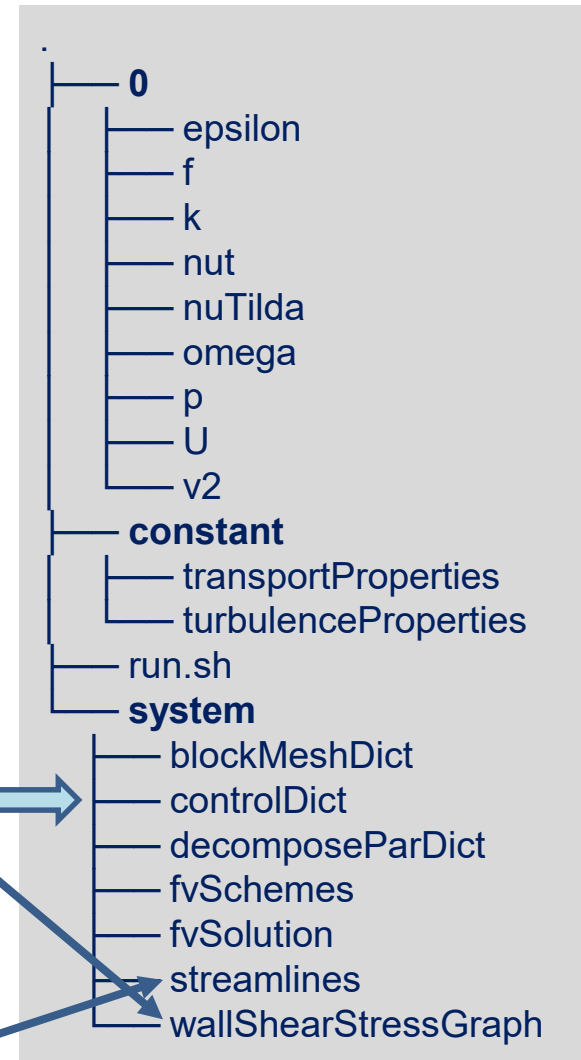
Built-in function  
*wallShearStress*

```

functions
{
    #includeFunc wallShearStress
    #includeFunc wallShearStressGraph
    #includeFunc streamlines
}
  
```

User-defined  
function, based  
on *sample*

User-defined  
function, based  
on *streamlines*





## Structure of the case

```

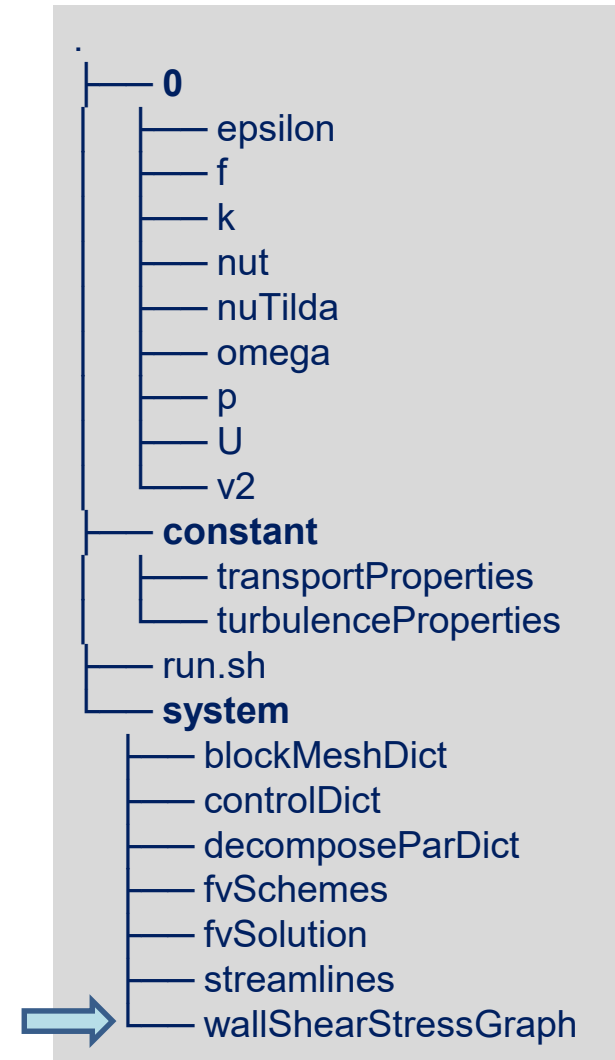
start (0 -0.0254 0);
end (0.206 -0.0254 0);
fields (wallShearStress);

// Sampling and I/O settings
#includeEtc "caseDicts/postProcessing/graphs/sampleDict.cfg"

// Override settings here, e.g.
setConfig
{
    axis x;
}

// Must be last entry
#includeEtc "caseDicts/postProcessing/graphs/graph.cfg"

```





## Structure of the case

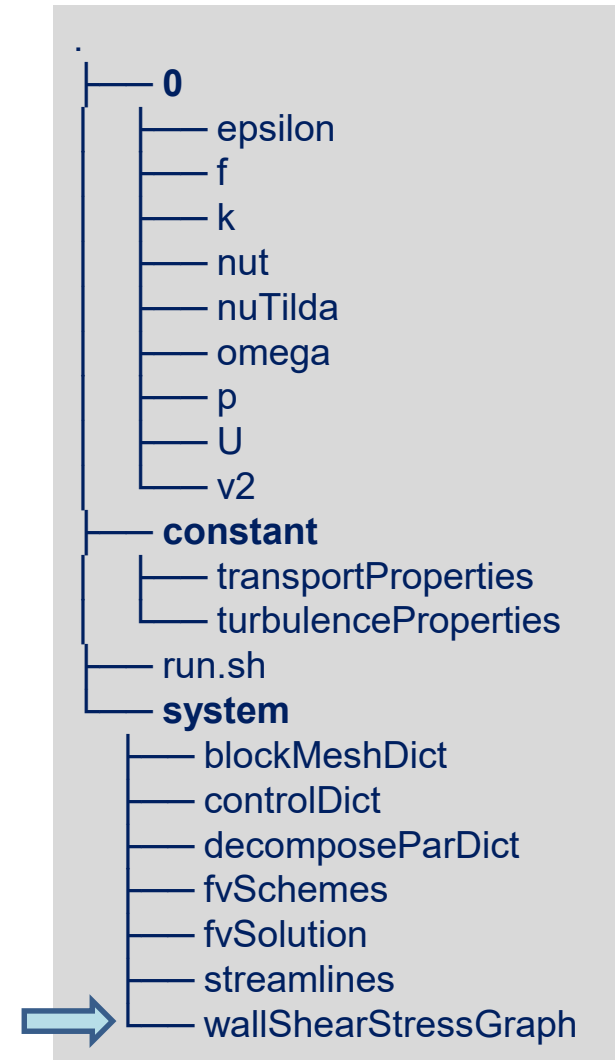
```
start (0 -0.0254 0);
end (0.206 -0.0254 0);
fields (wallShearStress);
```

```
// Sampling and I/O settings
#includeEtc "caseDicts/postProcessing/graphs/sampleDict.cfg"
```

```
// Override settings here, e.g.
setConfig
{
    axis x;
}
```

```
// Must be last entry
#includeEtc "caseDicts/postProcessing/graphs/graph.cfg"
```

```
interpolationScheme cellPoint;
setFormat raw;
setConfig
{
    type lineUniform; // lineCell, lineCellFace
    axis distance; // x, y, z, xyz
    nPoints 100;
}
```





## Structure of the case

```
start (0 -0.0254 0);
end (0.206 -0.0254 0);
fields (wallShearStress);
```

```
// Sampling and I/O settings
#includeEtc "caseDicts/postProcessing/graphs/sampleDict.cfg"
```

```
// Override settings here, e.g.
setConfig
{
    axis x;
}
```

```
// Must be last entry
#includeEtc "caseDicts/postProcessing/graphs/graph.cfg"
```

```
interpolationScheme cellPoint;
setFormat raw;
setConfig
{
    type lineUniform; // lineCell, lineCellFace
    axis distance; // x, y, z, xyz
    nPoints 100;
}
```

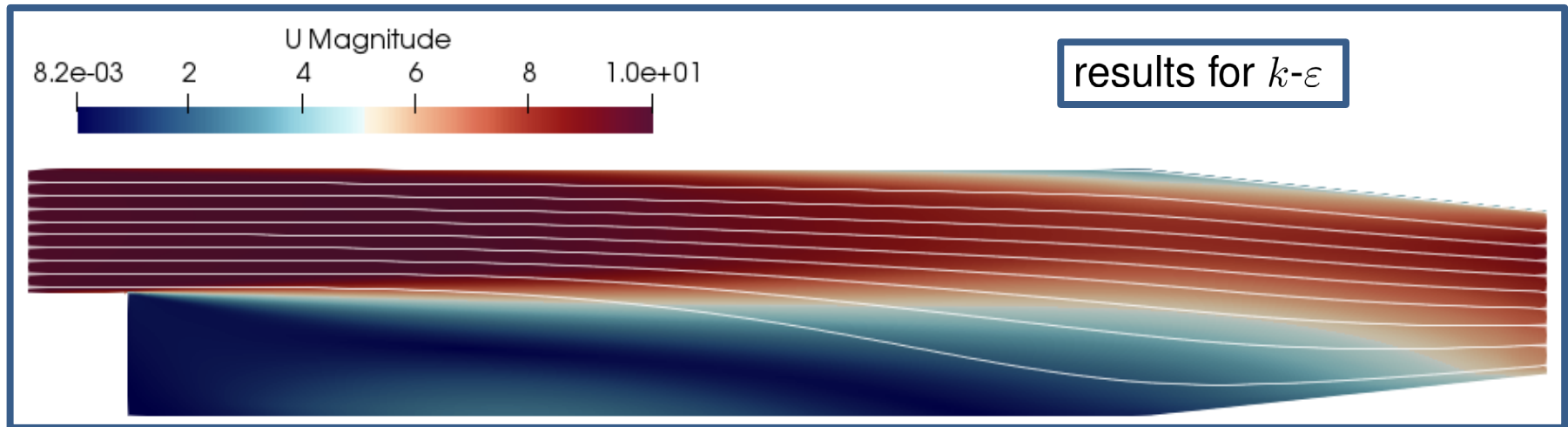
```

.
├── 0
│   └── caseDicts
│       ├── postProcessing
│       │   ├── graphs
│       │   │   ├── sampleDict.cfg
│       │   │   └── graph.cfg
│       └── system
│           ├── blockMeshDict
│           ├── controlDict
│           ├── decomposeParDict
│           ├── fvSchemes
│           ├── fvSolution
│           ├── streamlines
│           └── wallShearStressGraph

```

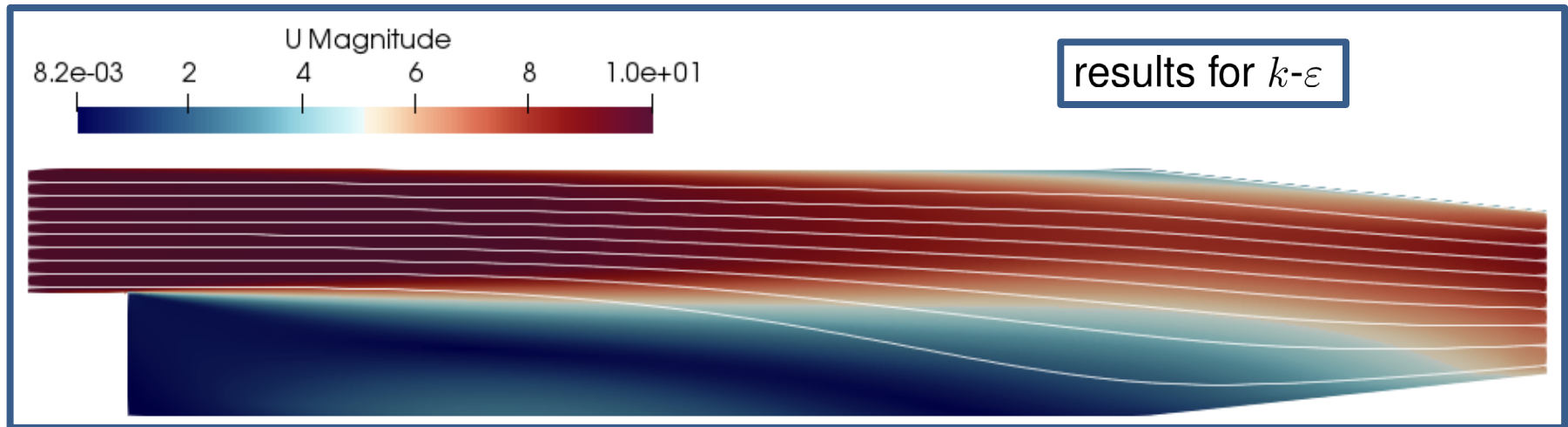


# Result





# Result



- Try to measure the length of the separation bubble?





## Standard solvers

- The source code for all solvers is located in:

```
~/OpenFOAM/OpenFOAM-7/applications/solvers
```

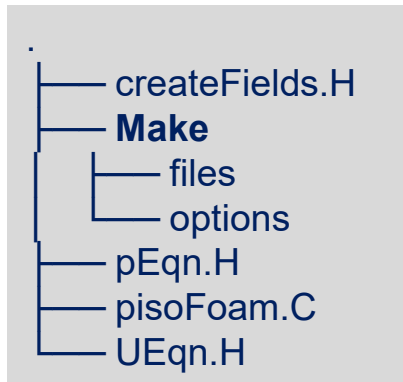


## Standard solvers

- The source code for all solvers is located in:

```
~/OpenFOAM/OpenFOAM-7/applications/solvers
```

- Most solvers have a similar structure (e.g. in *pisoFoam*):



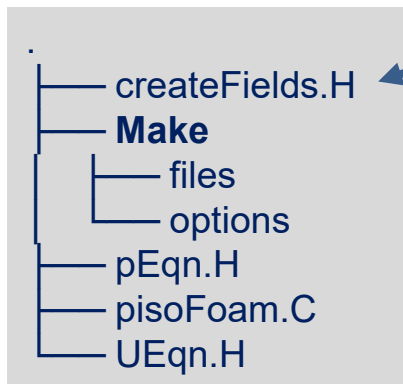


## Standard solvers

- The source code for all solvers is located in:

`~/OpenFOAM/OpenFOAM-7/applications/solvers`

- Most solvers have a similar structure (e.g. in *pisoFoam*):



Variable initialization

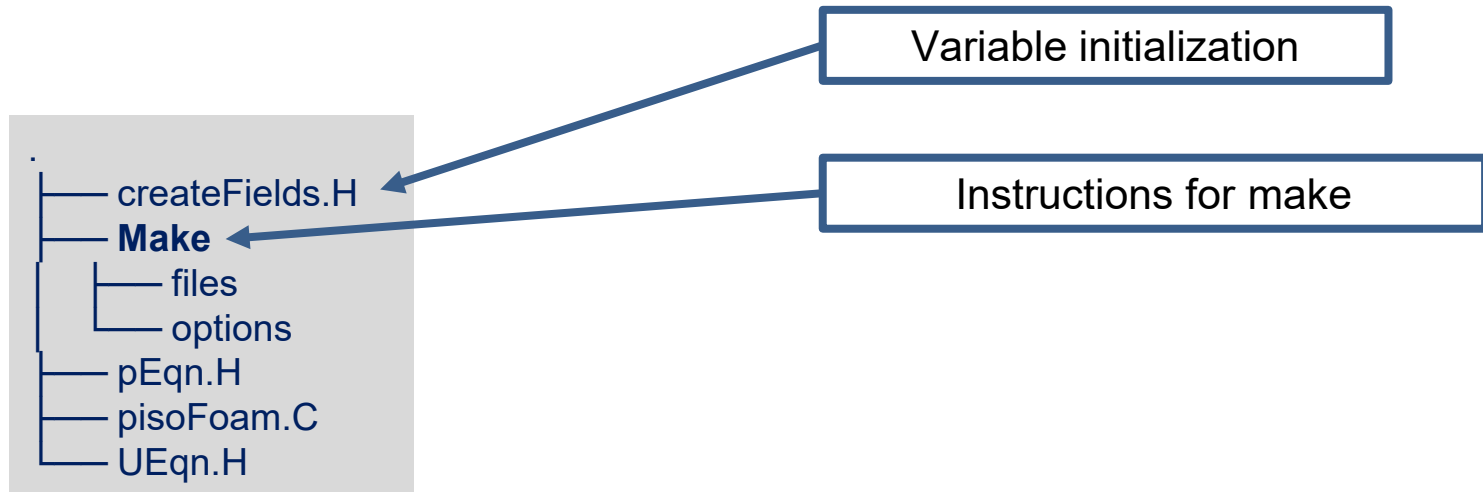


## Standard solvers

- The source code for all solvers is located in:

`~/OpenFOAM/OpenFOAM-7/applications/solvers`

- Most solvers have a similar structure (e.g. in *pisoFoam*):



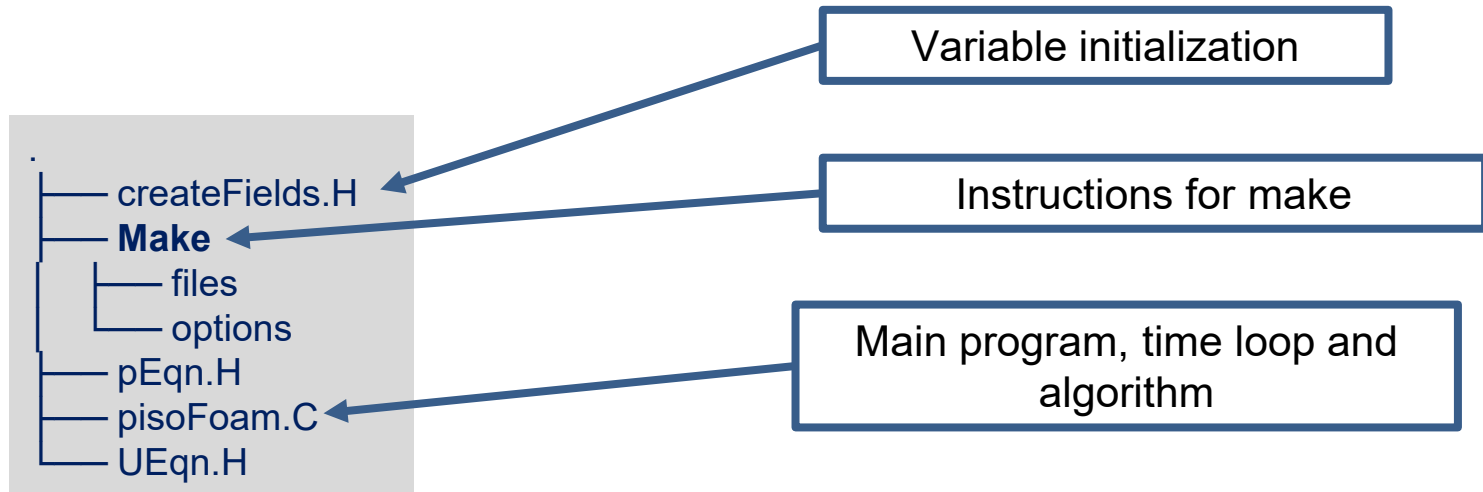


## Standard solvers

- The source code for all solvers is located in:

`~/OpenFOAM/OpenFOAM-7/applications/solvers`

- Most solvers have a similar structure (e.g. in *pisoFoam*):



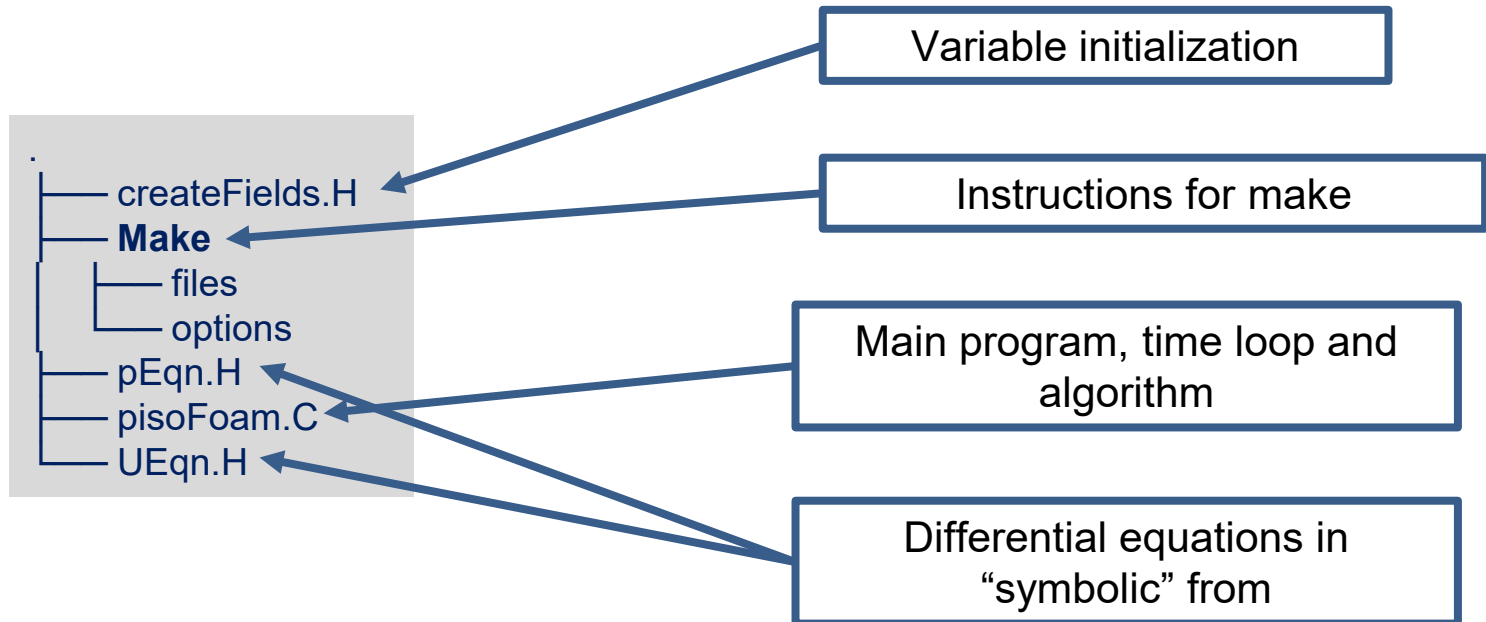


## Standard solvers

- The source code for all solvers is located in:

`~/OpenFOAM/OpenFOAM-7/applications/solvers`

- Most solvers have a similar structure (e.g. in *pisoFoam*):





# How to create a new solver


- ➡ 1. (Check if it does not exist already...)



## How to create a new solver

1. (Check if it does not exist already...)

➡ 2. It is a good idea to start from the existing ones!



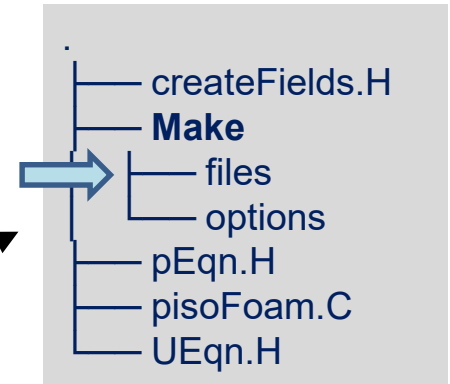
```
.
├── createFields.H
├── Make
│   ├── files
│   └── options
├── pEqn.H
├── pisoFoam.C
└── UEqn.H
```





## How to create a new solver

1. (Check if it does not exist already...)
2. It is a good idea to start from the existing ones!

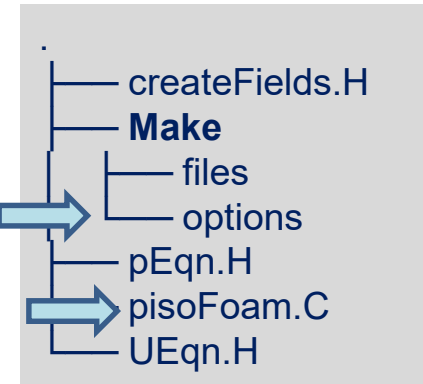


- ➡ 3. Modify *Make/files* for name and path.



## How to create a new solver

1. (Check if it does not exist already...)
2. It is a good idea to start from the existing ones!
3. Modify *Make/files* for name and path.
- ➡ 4. Add libraries with relevant tools.

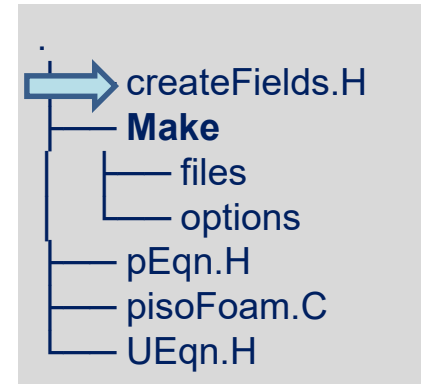




## How to create a new solver

1. (Check if it does not exist already...)
2. It is a good idea to start from the existing ones!
3. Modify *Make/files* for name and path.
4. Add libraries with relevant tools.

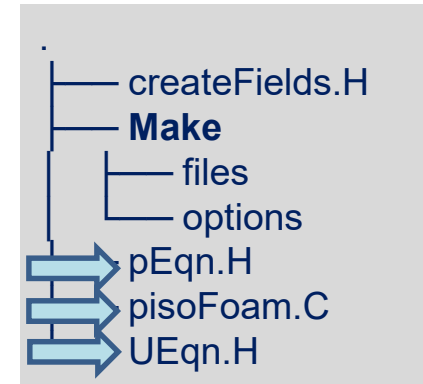
➡ 5. Initialize variables.





## How to create a new solver

1. (Check if it does not exist already...)
2. It is a good idea to start from the existing ones!
3. Modify *Make/files* for name and path.
4. Add libraries with relevant tools.
5. Initialize variables.



- ➡ 6. Write / modify equations / add operations.



## Example: *pisokinematicParcelFoam*

- Add one-way coupled colliding particles to incompressible flow.



## Example: *isoKinematicParcelFoam*

- Add one-way coupled colliding particles to incompressible flow.

`~/OpenFOAM/OpenFOAM-7/applications/solvers/lagrangian/icoUncoupledKinematicParcelFoam`

+

`~/OpenFOAM/OpenFOAM-7/applications/solvers/incompressible/pisoFoam`



## Example: *pisoKinematicParcelFoam*

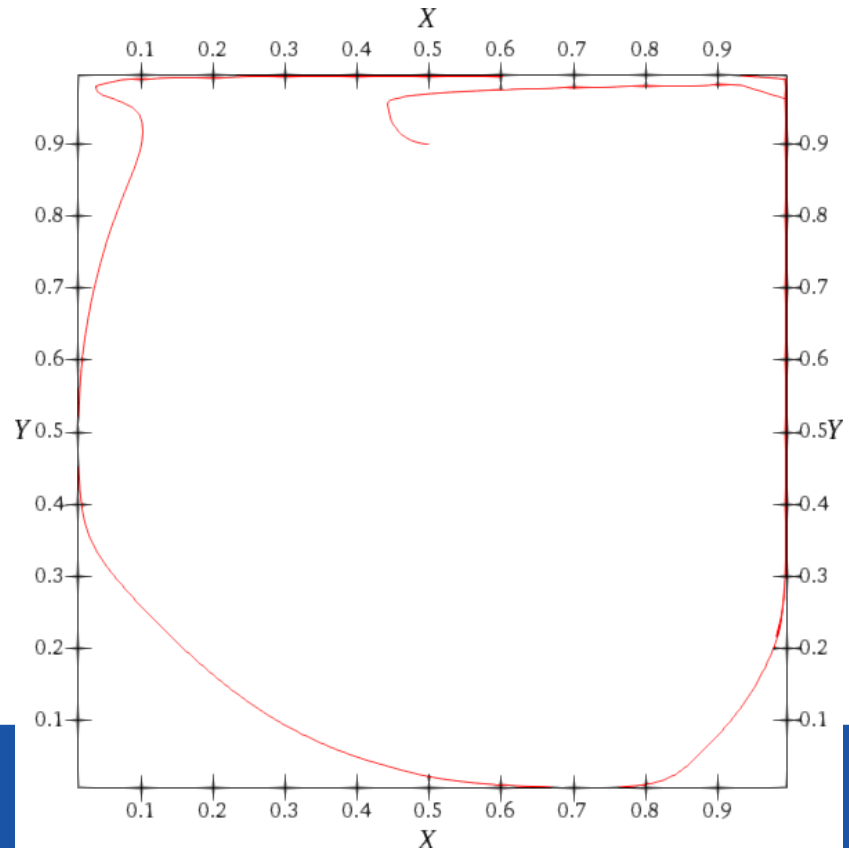
- Add one-way coupled colliding particles to incompressible flow.

`~/OpenFOAM/OpenFOAM-7/applications/solvers/lagrangian/icoUncoupledKinematicParcelFoam`

+

`~/OpenFOAM/OpenFOAM-7/applications/solvers/incompressible/pisoFoam`

■  $Re = 1,000, \rho_p/\rho_f = 1$





## Example: *pisoKinematicParcelFoam*

- Add one-way coupled colliding particles to incompressible flow.

`~/OpenFOAM/OpenFOAM-7/applications/solvers/lagrangian/icoUncoupledKinematicParcelFoam`

+

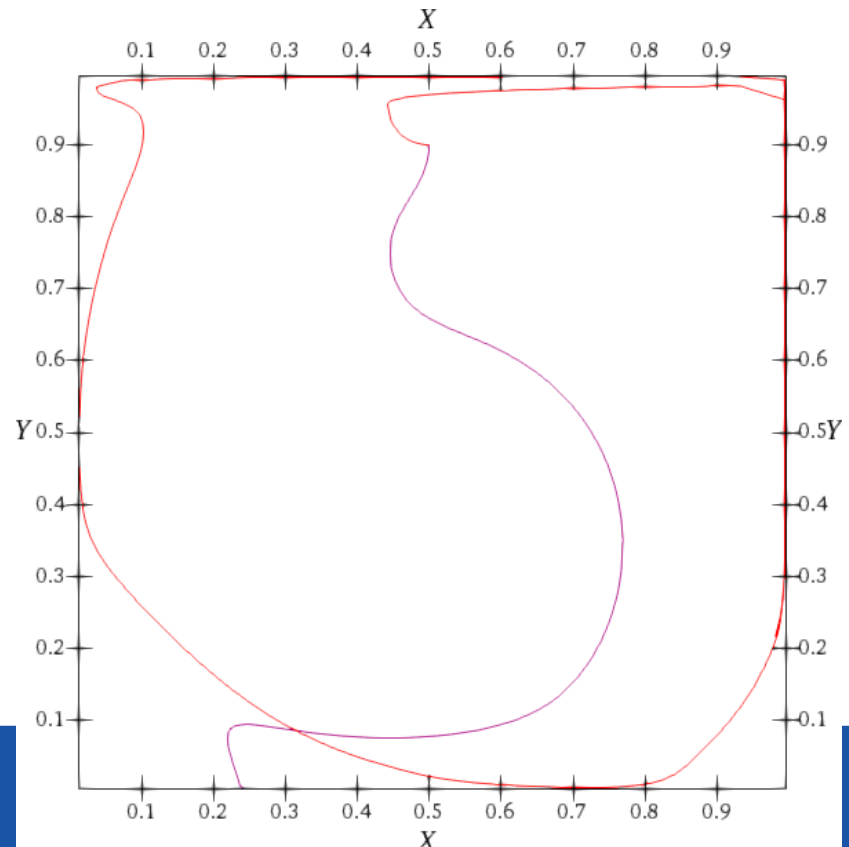
`~/OpenFOAM/OpenFOAM-7/applications/solvers/incompressible/pisoFoam`



$Re = 1,000, \rho_p/\rho_f = 1$



$Re = 1,000, \rho_p/\rho_f = 10$







## Example: *pisoKinematicParcelFoam*

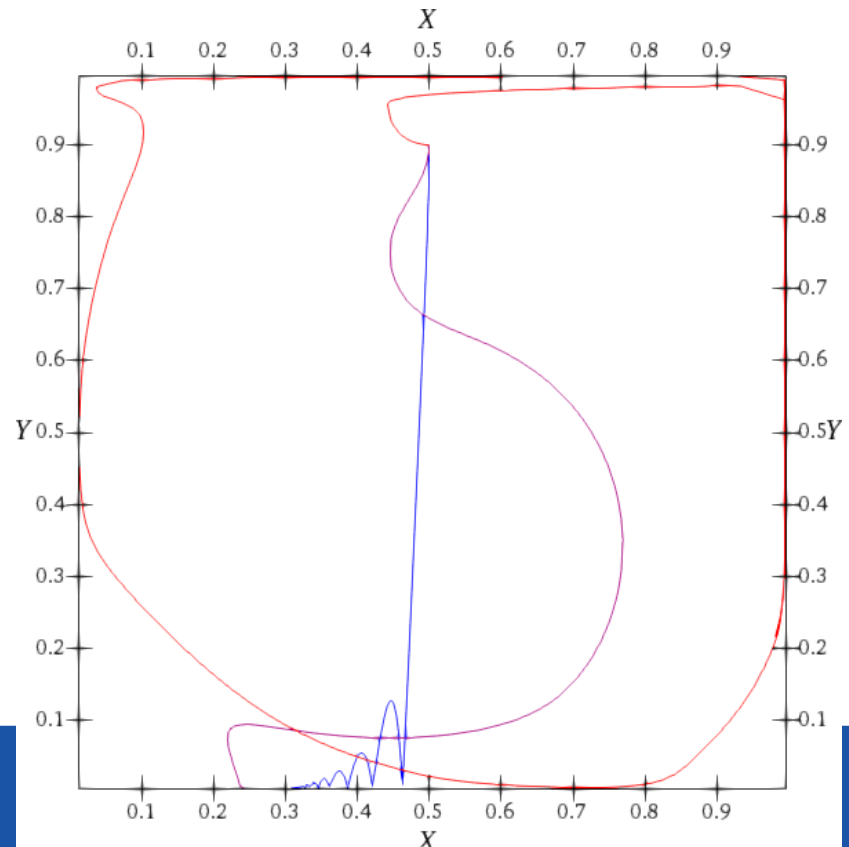
- Add one-way coupled colliding particles to incompressible flow.

`~/OpenFOAM/OpenFOAM-7/applications/solvers/lagrangian/icoUncoupledKinematicParcelFoam`

+

`~/OpenFOAM/OpenFOAM-7/applications/solvers/incompressible/pisoFoam`

- $Re = 1,000, \rho_p/\rho_f = 1$
- $Re = 1,000, \rho_p/\rho_f = 10$
- $Re = 1,000, \rho_p/\rho_f = 100$





## Example: *pisoKinematicParcelFoam*

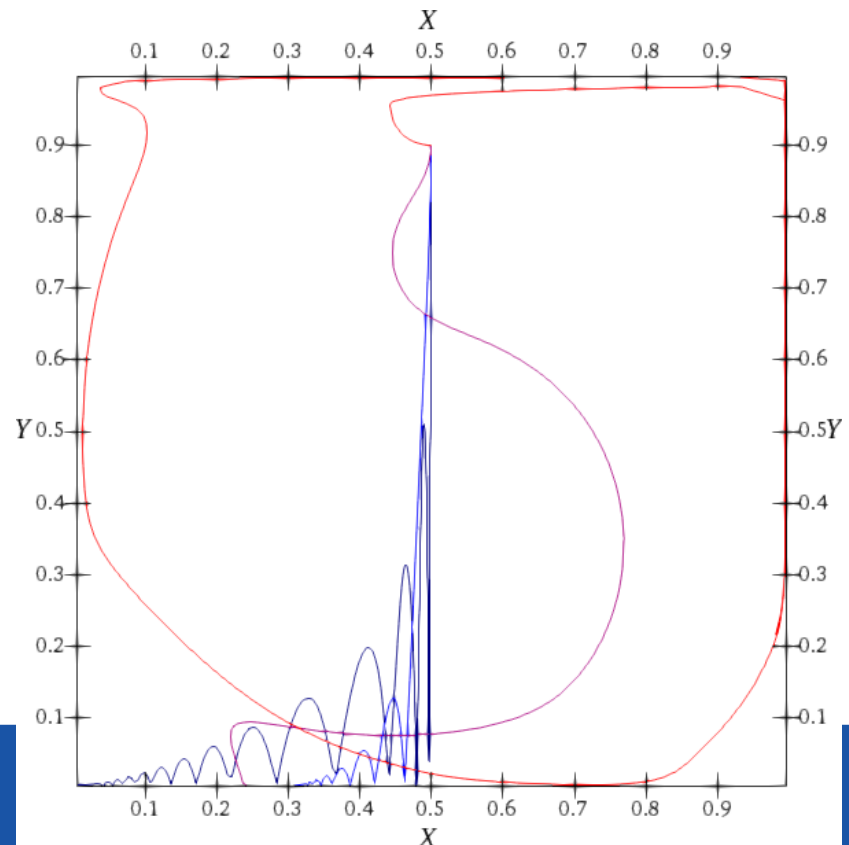
- Add one-way coupled colliding particles to incompressible flow.

`~/OpenFOAM/OpenFOAM-7/applications/solvers/lagrangian/icoUncoupledKinematicParcelFoam`

+

`~/OpenFOAM/OpenFOAM-7/applications/solvers/incompressible/pisoFoam`

- $Re = 1,000, \rho_p/\rho_f = 1$
- $Re = 1,000, \rho_p/\rho_f = 10$
- $Re = 1,000, \rho_p/\rho_f = 100$
- $Re = 1,000, \rho_p/\rho_f = 1,000$





# Summary

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver!



# Summary

Just check that everything is properly built.

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver!



# Summary

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver!

**Just check that everything is properly built.**

- **Try different turbulence models!**
- **Use the wall-shear stress to study the separation bubble!**



# Summary

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver!

**Just check that everything is properly built.**

- **Try different turbulence models!**
- **Use the wall-shear stress to study the separation bubble!**

**Study drag/gravity!**



# Summary

1. Introduction
2. Your first case
3. Your second case (with post-processing!)
4. Let's create a new solver!

**Just check that everything is properly built.**

- **Try different turbulence models!**
- **Use the wall-shear stress to study the separation bubble!**

**Study drag/gravity!**

**Thanks for your kind attention!**