

# Turbulence modeling hands-on tutorials

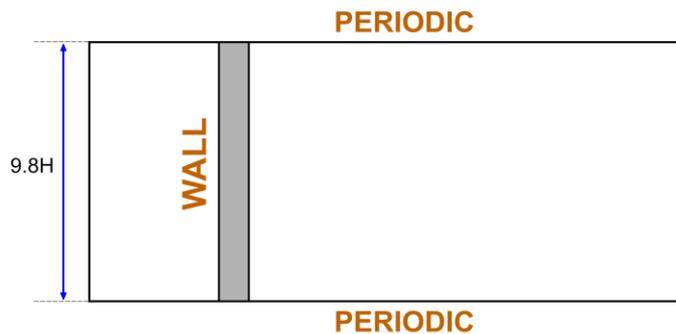
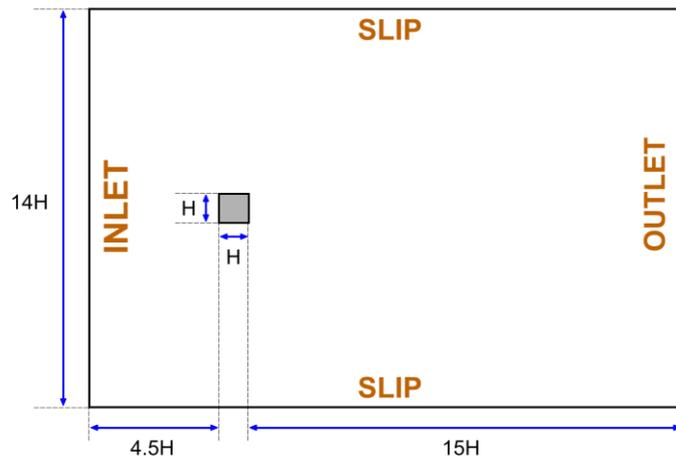
- Vortex shedding past square cylinder
- Let us run this case. Go to the directory:

```
$PTOFC/squarecil
```

- \$PTOFC is pointing to the directory where you extracted the training material.
- In the case directory, you will find the `README.FIRST` file. In this file, you will find the general instructions of how to run the case. In this file, you might also find some additional comments.
- You will also find a few additional files (or scripts) with the extension `.sh`, namely, `run_all.sh`, `run_mesh.sh`, `run_sampling.sh`, `run_solver.sh`, and so on. These files can be used to run the case automatically by typing in the terminal, for example, `sh run_solver`.
- We highly recommend you to open the `README.FIRST` file and type the commands in the terminal, in this way, you will get used with the command line interface and OpenFOAM® commands.
- If you are already comfortable with OpenFOAM®, use the automatic scripts to run the cases.

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder



### Physical and numerical side of the problem:

- The governing equations of the problem are the incompressible Navier-Stokes equations.
- To model the turbulence, we will use two approaches, LES and RANS.
- We are going to work in a 3D domain with periodic boundary conditions.
- This problem has plenty of experimental data for validation.

$$U_{in} = 0.535 \text{ m/s}$$

$$H = 0.04 \text{ m}$$

$$Re = 21400$$

Working fluid: Water

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

Turbulence model	Drag coefficient	Strouhal number	Computing time (s)
Laminar	2.81	0.179	93489
LES	2.36	0.124	77465
DES	2.08	0.124	70754
SAS	2.40	0.164	57690
URANS (WF)	2.31	0.130	67830
URANS (No WF)	2.29	0.135	64492
RANS	2.30	-	28246 (10000 iter)
Experimental values	2.05-2.25	0.132	-

### References:

Lyn, D.A. and Rodi, W., The flapping shear layer formed by flow separation from the forward corner of a square cylinder. *J. Fluid Mech.*, 267, 353, 1994.  
Lyn, D.A., Einav, S., Rodi, W. and Park, J.H., A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder. *Report. SFB 210 /E/100.*

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

- We will use this case to learn how to setup a turbulent case (RANS and LES).
- To run this case we will use the solvers *simpleFoam* (steady solver) and *pimpleFoam* (unsteady solver).
- To get fast outcomes, we will use a coarse mesh. But feel free to refine the mesh, especially close to the walls.
- Remember, as we are introducing new closure equations for the turbulence problem, we need to define initial and boundary conditions for the new variables.
- We will use a few **functionObjects** to compute some additional quantities, such as, Q criterion,  $y^+$ , minimum and maximum values, forces, time average, and online sampling.
- After finding the solution, we will visualize the results.
- We will also compare the numerical solution with the experimental results.
- At the end, we will do some plotting and advanced post-processing using gnuplot and Python.
- Have in mind that the unsteady case will generate a lot of data.
- You will find the instructions of how to run this case in the file *README.FIRST* located in the case directory.

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

- We select the turbulence model in the `turbulenceProperties` dictionary file.
- This dictionary file is located in the directory `constant`.
- To select the K-Omega SST turbulence model,

```
17 simulationType RAS; ← RANS type simulation
18
19 RAS ← RANS sub-dictionary
20 {
21     RASModel kOmegaSST; ← RANS model to use
22
23     turbulence on; ← Turn on/off turbulence. Runtime modifiable
24     printCoeffs on; ← Print coefficients at the beginning
25 }
```

- Remember, you need to assign boundary and initial conditions to the new variables (**k**, **omega**, and **nut**).

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

- We select the turbulence model in the `turbulenceProperties` dictionary file.
- This dictionary file is located in the directory `constant`.
- To select the LES (Smagorinsky) turbulence model,

```
17 simulationType LES; ← LES type simulation
18
19 LES ← LES sub-dictionary
20 {
21     LESModel Smagorinsky; ← LES model to use
22
23     turbulence on; ← Turn on/off turbulence. Runtime modifiable
24
25     printCoeffs on; ← Print coefficients at the beginning
26
27     delta cubeRootVol;
28     cubeRootVolCoeffs
29     {
30         deltaCoeff 1;
31     }
32 }
100 }
```

- Remember, you need to assign boundary and initial conditions to the new variables (**nut**).

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

- To select wall functions, follow this table,

Field	Wall functions – High RE	Resolved BL – Low RE
<b>nut</b>	<b>nutUSpaldingWallFunction</b>	<b>fixedValue 0</b> or a small number
<b>k</b>	<b>kqRWallFunction</b>	<b>fixedValue 0</b> or a small number
<b>omega</b>	<b>omegaWallFunction</b>	<b>omegaWallFunction</b>

- In this tutorial,
  - Use High RE for RANS.
  - Use High RE and Low RE for URANS.
  - Use High RE and Low RE for LES.

# Turbulence modeling hands-on tutorials

## Vortex shedding past square cylinder

- The initial value for the turbulent kinetic energy  $\kappa$  can be found as follows,

$$\kappa = \frac{3}{2}(UI)^2$$

- The initial value for the specific kinetic energy  $\omega$  can be found as follows,

$$\omega = \frac{\rho\kappa}{\mu} \frac{\mu_t}{\mu}^{-1}$$

- Use the following initial estimates,  $I = 5\%$  and  $\frac{\mu_t}{\mu} = 10$
- At this point, we are ready to run. But before running, remember to setup the right numerics in the dictionary files *fvSolution* and *fvSchemes*.
- Also, for the LES simulation try to keep the CFL number below 0.9
- Finally, do not forget to setup the **functionObjects** to compute the forces, do the sampling, and compute  $y^+$  on-the-fly.

# Turbulence hands-on tutorials

## Vortex shedding past square cylinder

- At this point, we are ready to run the simulation.
- To run the tutorial using `simpleFoam` (RANS simulation), type in the terminal:

```
1. $> foamCleanTutorials
2. $> blockMesh
3. $> checkMesh
4. $> rm -r 0
5. $> cp -r 0_org 0
6. $> decomposePar
7. $> mpirun -np 4 renumberMesh -overwrite -parallel
8. $> mpirun -np 4 simpleFoam -parallel | tee log
9. $> mpirun -np 4 simpleFoam -postProcess -func Q -parallel
10. $> mpirun -np 4 simpleFoam -postProcess -func yPlus -parallel
11. $> reconstructPar
12. $> paraFoam
```

# Turbulence hands-on tutorials

## Vortex shedding past square cylinder

- In step 1, we clean the case directory. This is not compulsory, but it is highly advisable.
- In steps 2-3 we generate the mesh and check its quality.
- In steps 4-5, we transfer the backup or original files to the directory 0. It is highly advisable to always keep backup files.
- In step 6 we decompose the mesh. At this point we are preparing to run in parallel.
- In step 7 we use `renumberMesh` to reduce the bandwidth of the coefficient matrices.
- In step 8 we run in parallel and save the standard output in the file `log`.
- In step 9 we compute the Q-criterion using the utility `postProcess`.
- In step 10 we compute the `yPlus` using the utility `postProcess`.
- In step 11 we reconstruct the solution.
- In step 12 we use `paraFoam` to visualize the solution.

# Turbulence hands-on tutorials

## Vortex shedding past square cylinder

- To run the tutorial using `pimpleFoam` (LES simulation), type in the terminal:

```
1. $> foamCleanTutorials
2. $> blockMesh
3. $> checkMesh
4. $> rm -r 0
5. $> cp -r 0_org 0
6. $> refineWallLayer '(square)' 0.7 -overwrite
7. $> refineWallLayer '(square)' 0.5 -overwrite
8. $> refineWallLayer '(square)' 0.3 -overwrite
9. $> mapFields ../RANS -consistent -noFunctionObjects
    -mapMethod cellPointInterpolate -sourceTime latestTime
```

# Turbulence hands-on tutorials

## Vortex shedding past square cylinder

- To run the tutorial using `pimpleFoam` (LES simulation), type in the terminal:

10. `$> decomposePar`

11. `$> mpirun -np 4 renumberMesh -overwrite -parallel`

12. `$> mpirun -np 4 pimpleFoam -parallel > log | tail -f log`

13. `$> mpirun -np 4 pimpleFoam -postProcess -func Q -parallel`

14. `$> mpirun -np 4 pimpleFoam -postProcess -func yPlus -parallel`

15. `$> reconstructPar`

16. `$> paraFoam`

# Turbulence hands-on tutorials

## Vortex shedding past square cylinder

- In step 1, we clean the case directory. This is not compulsory, but it is highly advisable.
- In steps 2 and 3 we generate the mesh and check its quality.
- In steps 4-5, we transfer the backup or original files to the directory 0. It is highly advisable to always keep backup files.
- In steps 6-8, we use the utility `refineWallLayer` to refine the mesh close to the walls (**square** in this case). This is helpful if you do not want to regenerate the mesh.
- In step 9, we interpolate the solution obtained using a RANS method on a coarse mesh, into the current finer mesh (the mesh was refined in steps 6-8). Remember, you need to have a solution in the directory `../RANS` (and preferably in serial).
- In step 10 we decompose the mesh. At this point we are preparing to run in parallel.
- In step 11 we use `renumberMesh` to reduce the bandwidth of the coefficient matrices.
- In step 12 we run in parallel and save the standard output in the file `log`.
- In step 13 we compute the Q-criterion using the utility `postProcess`.
- In step 14 we compute the `yPlus` using the utility `postProcess`.
- In step 15 we reconstruct the solution.
- In step 16 we use `paraFoam` to visualize the solution.