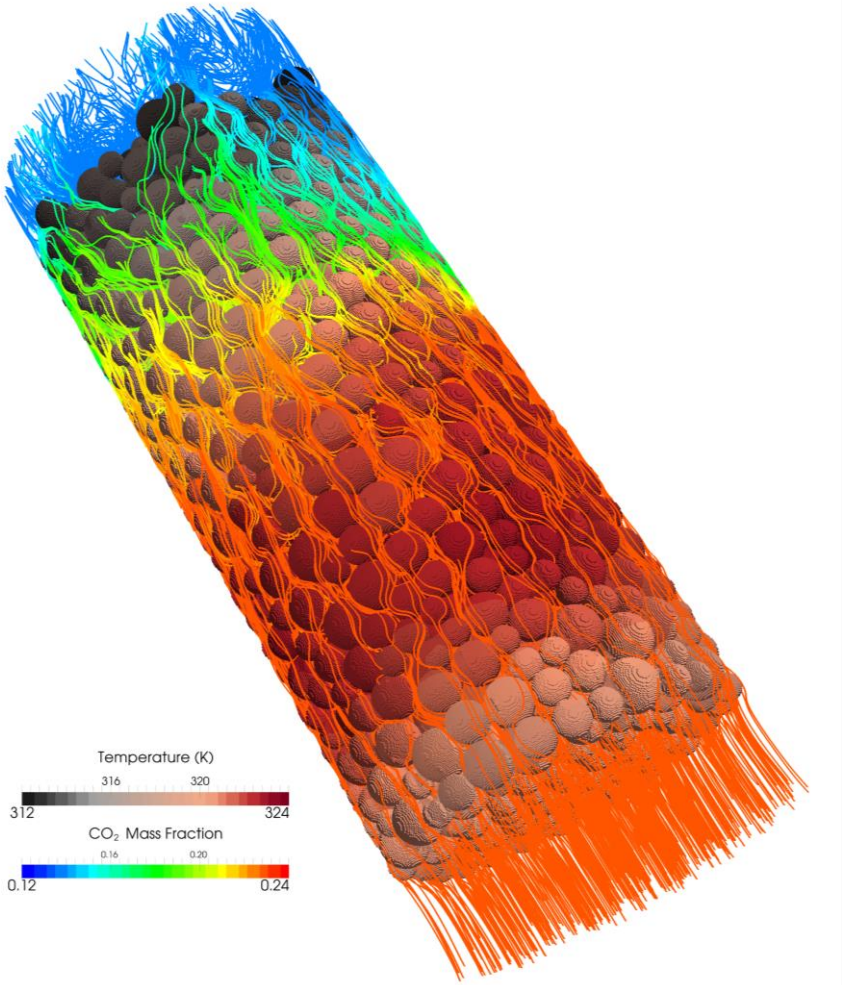
OpenFOAM® Basic Training

Tutorial Six



3rd edition, Feb. 2015

www.thvt.at

Editors and Contributors:

- Bahram Haddadi (TU Wien)
- Christian Jordan (TU Wien)
- Jozsef Nagy (JKU Linz)
- Clemens Gößnitzer (TU Wien)
- Vikram Natarajan (TU Wien)
- Sylvia Zibuschka (TU Wien)
- Michael Harasek (TU Wien)

Cover picture from:

- Bahram Haddadi, The image presented on the cover page has been prepared using the Vienna Scientific Cluster (VSC).

simpleFoam – pitzDaily (turbulence, steady state)

Simulation

Use simpleFoam solver, run a steady state simulation with following turbulence models:

- kEpsilon (RAS)
- kOmega (RAS)
- LRR (RAS)

Objectives

- Understanding turbulence modeling
- Understanding steady state simulation

Post processing

Show the results of U and the turbulent viscosity in two separate contour plots.

Step by step simulation

Copy tutorial

```
~/OpenFOAM/OpenFOAM-2.3.0/tutorials/incompressible/simpleFoam

/pitzDaily
```

0 directory

When a turbulent model is chosen, the value of its constants and its boundary values should be set in the appropriate files, for example in kEpsilon model the k and epsilon files should be edited. e.g. epsilon:

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 14.855;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 14.855;
    }
    outlet
    {
        type            zeroGradient;
    }
    upperWall
    {
        type            epsilonWallFunction;
        value           uniform 14.855;
    }
    lowerWall
    {
        type            epsilonWallFunction;
        value           uniform 14.855;
    }
    frontAndBack
    {
        type            empty;
    }
}
// ************************************************************************* //
```

Note: Here is a list of files which should be available at 0 directory and need to be modified for each turbulence model:

- *laminar: no file*
- *kEpsilon (RAS): k and epsilon*
- *kOmega (RAS): k and omega*
- *LRR (RAS): k, epsilon and R*
- *Smagorinsky (LES): nuSgs*
- *oneEqEddy (LES): k and nuSgs*

- *SpalartAllmaras (LES): nuSgs and nuTilda*

Some files are available, e.g. epsilon, k and nuTilda, some files should be created by the user, e.g. R, omega. Templates for these files can be also found in the examples of older versions of OpenFOAM®, e.g. 1.7.1.

Note: Missing R file can be created by OpenFOAM®. In the constant directory in RASProperties file set the `RASModel` *to* `kEpsilon`*. The turbulenceProperties file is also needed. Copy it from another tutorial and set* `simulationType` *to* `RASModel` *in the file. Run the command "R" from terminal, it will create the R file in the 0 directory.*

constant directory

The type of simulation turbulence model is set in turbulenceProperties file, e.g. it is a RASModel or LESModel (this file is not available in this tutorial, but can be copied from other tutorials). For choosing a specific turbulence model the RASProperties file should be checked (e.g. here kEpsilon).

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

RASModel        kEpsilon;

turbulence      on;

printCoeffs     on;

// ************************************************************************* //
```

Note: For the laminar model both turbulenceProperties and RASProperties should be set to `laminar`*. In the RASProperties set* `turbulence` *and also* `printCoeffs` *to* `off`*.*

system directory

Note: Since it is a steady state simulation controlDict `endTime` *show the number of iterations instead of time and deltaT should be* `1`*, because it is the amount of increase in the iteration number.*

Running simulation

```
>blockMesh

>simpleFoam
```

Note: When the solution converges, "`SIMPLE solution converged in … iterations`*" message will be displayed in the Shell window. If nothing happens and you see no message after a while (it is not the case in here, it converges after a short time), then you should check the residuals which are displayed in the Shell window manually (you should* `check initial` *residual values, it shows the difference between this iteration and the last one), if all of the* `Initial residual` *(see below) values are close to amounts you have set in the fvSolution then you can stop simulation (ctrl+c).*

```
Time = 817

smoothSolver:  Solving for Ux, Initial residual = 0.00013826, Final residual =
9.87886e-06, No Iterations 2
smoothSolver:  Solving for Uy, Initial residual = 0.000994709, Final residual =
7.317e-05, No Iterations 2
GAMG:  Solving for p, Initial residual = 0.00192871, Final residual =
0.000174838, No Iterations 7
time step continuity errors : sum local = 0.000840075, global = 6.13868e-05,
cumulative = -0.193739
smoothSolver:  Solving for epsilon, Initial residual = 0.000175322, Final
residual = 1.138e-05, No Iterations 2
smoothSolver:  Solving for k, Initial residual = 0.000404928, Final residual =
2.99083e-05, No Iterations 2
ExecutionTime = 20.11 s  ClockTime = 20 s

SIMPLE solution converged in 817 iterations
```

Exporting simulation

The simulation results are as follows:

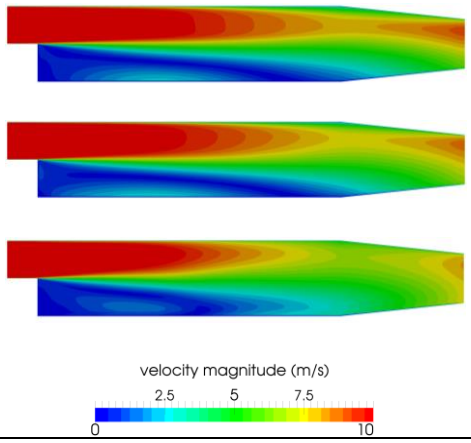
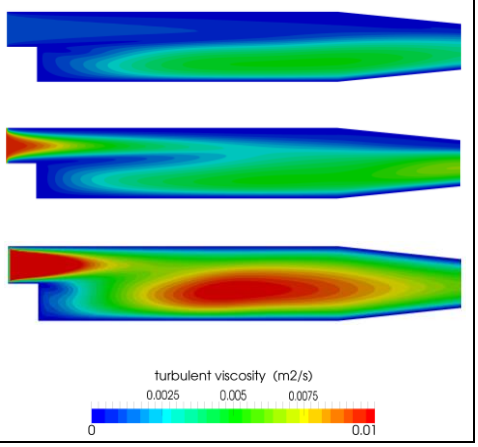
RAS model	Velocity magnitude	Turbulent viscosity
kEpsilon		
kOmega		
LRR		

Figure 6.1 Comparison of different turbulent models at steady state