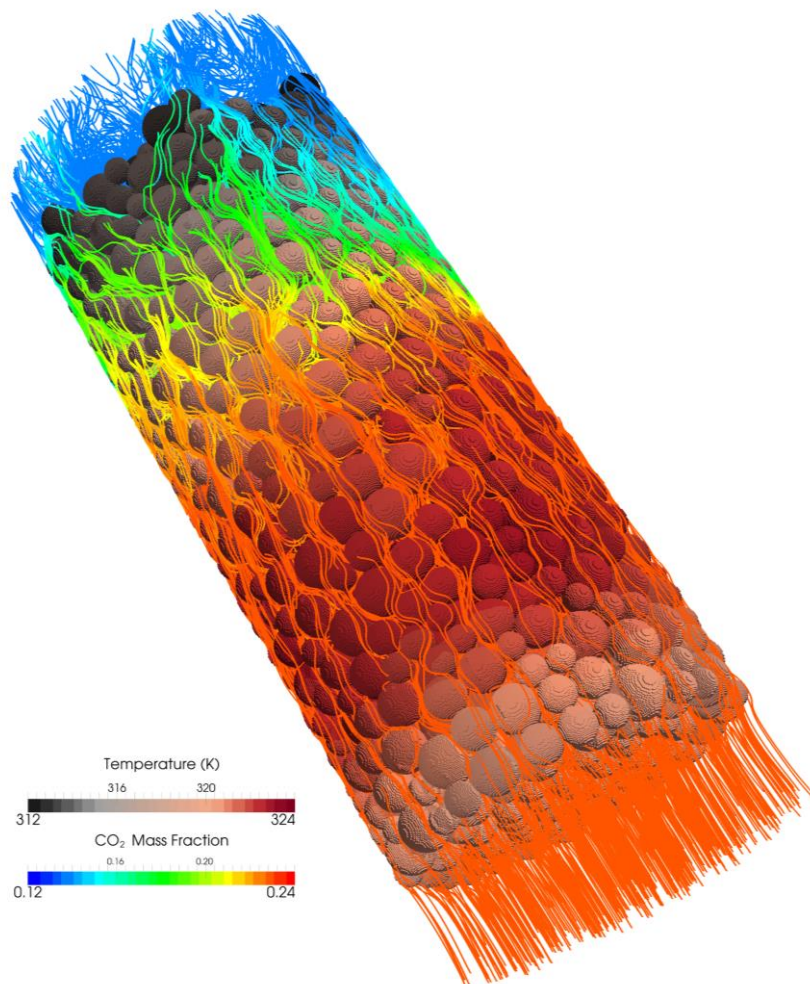


OpenFOAM[®] Basic Training

Tutorial Seven



3rd edition, Feb. 2015



This offering is not approved or endorsed by ESI[®] Group, ESI-OpenCFD[®] or the OpenFOAM[®] Foundation, the producer of the OpenFOAM[®] software and owner of the OpenFOAM[®] trademark.

Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>


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).


 Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Attribution–NonCommercial–ShareAlike 3.0 Unported (CC BY–NC–SA 3.0)

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

With the understanding that:

Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;

The author's moral rights;

Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

pisoFoam – pitzDaily (turbulence, transient)

Simulation

Use the pisoFoam solver, run a backward facing step case for 0.2 s with different turbulence models:

- Smagorinsky (LES)
- oneEqEddy (LES)
- kEpsilon (RAS)

Objectives

- Understanding turbulence models
- Understanding the difference between transient and steady state simulation
- Finding appropriate turbulent model

Post processing

Display the results of U and the turbulent viscosity in two separate contour plots at three different time steps. Compare with steady state simulation (example 6).

Step by step simulation

Copy tutorial

Copy the tutorial from the following directory to your working directory:

```
~/OpenFOAM/OpenFOAM-
2.3.0/tutorials/incompressible/pisoFoam/les
/pitzDaily
```

0 directory

Set the turbulence model initial and boundary values.

Note: For different turbulent models, different constant files should be modified (check example 6).

constant directory

As mentioned in example 6, in turbulenceProperties the turbulent model type has to be set.

```
// * * * * * //
simulationType LESModel;
// ***** //
```

For setting a turbulence model, if RAS models are being used, in the constant directory there is RASProperties file and we should modify it, but if LES models are used LESProperties file should be found and modified.

```
// * * * * * //
LESModel      oneEqEddy;
delta         cubeRootVol;
printCoeffs   on;
cubeRootVolCoeffs
{
    deltaCoeff      1;
}
PrandtlCoeffs
{
    delta           cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff      1;
    }
    smoothCoeffs
    {
        delta           cubeRootVol;
        cubeRootVolCoeffs
        {
            deltaCoeff      1;
        }
        maxDeltaRatio  1.1;
    }
    Cdelta         0.158;
```

```

}

vanDriestCoeffs
{
    delta            cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff    1;
    }

    smoothCoeffs
    {
        delta            cubeRootVol;
        cubeRootVolCoeffs
        {
            deltaCoeff    1;
        }

        maxDeltaRatio    1.1;
    }

    Aplus            26;
    Cdelta            0.158;
}

smoothCoeffs
{
    delta            cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff    1;
    }

    maxDeltaRatio    1.1;
}

// ***** //

```

Running simulation

```
>blockMesh
```

```
> pisoFoam
```

Exporting simulation

The simulation results are as follows:

For the kEpsilon model after 0.1 s the results are similar to the steady state simulation. Therefore, it can be assumed it has reached the steady state. Other models do not have a steady situation and are fluctuating all the time, so they require averaging for obtaining steady state results.

kEpsilon and other RAS models use averaging to obtain the turbulence values, but LES does not include any averaging by default. Therefore, LES simulations should use a higher grid resolution (smaller cells) and smaller time steps (for reasonable Co number). Contour plots or other LES results should be presented time averaged over reasonable number of time steps (not done in this example).

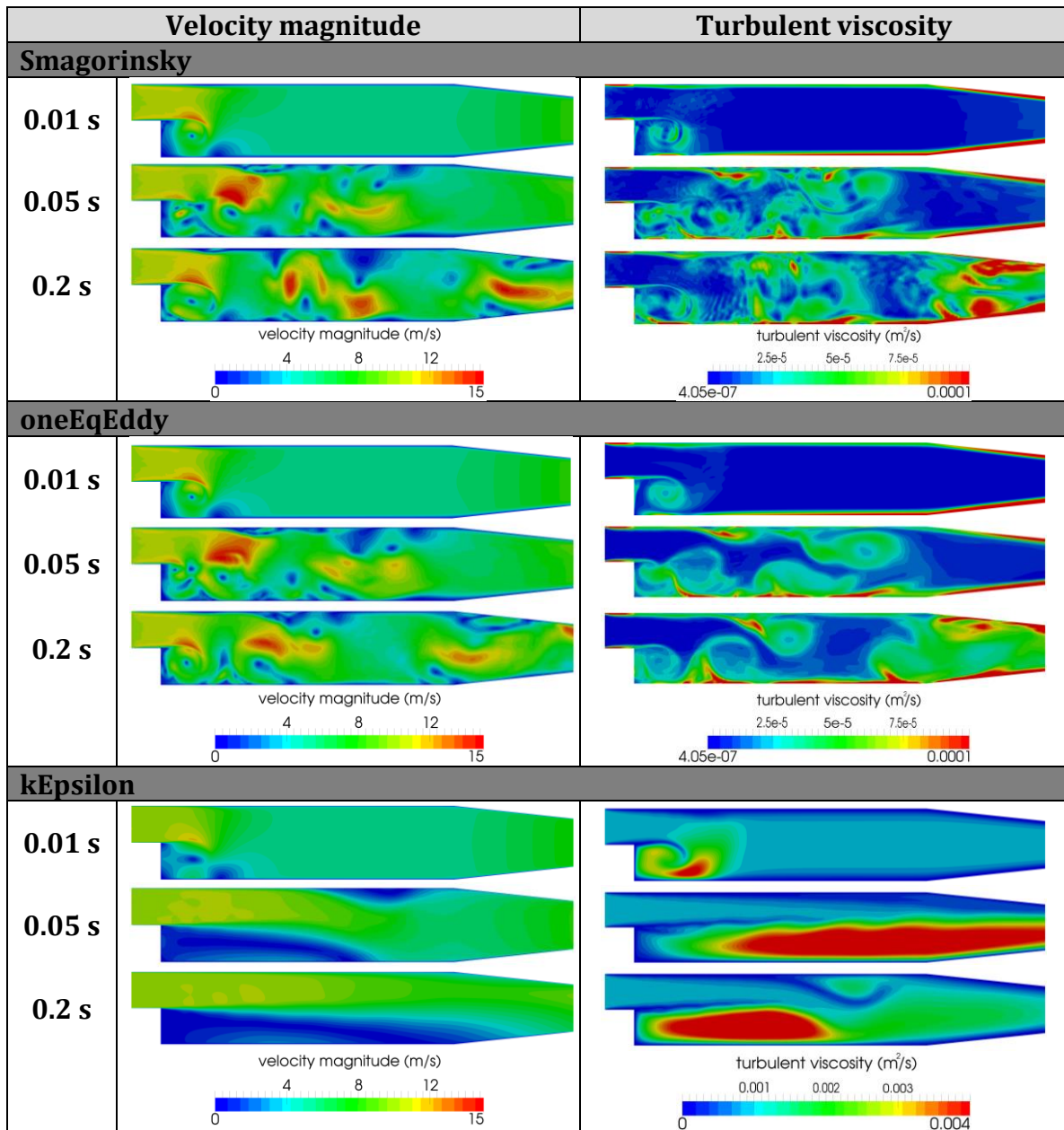


Figure 7.1 Comparison of different turbulent models for transient simulation.