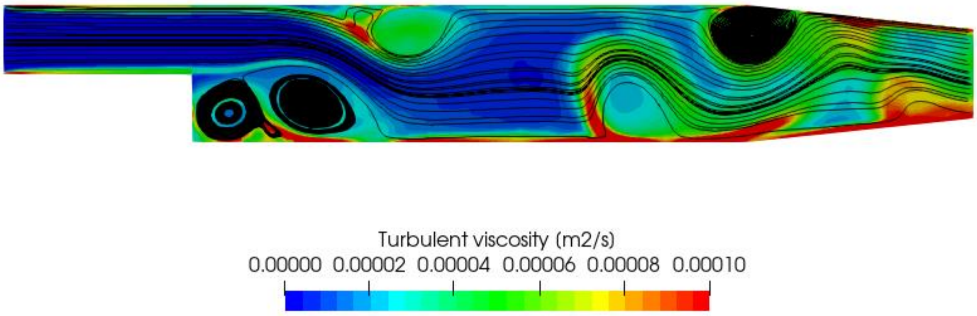
Tutorial Seven

Turbulence - Transient



Bahram Haddadi

7th edition, March 2025

Contributors:

- Bahram Haddadi
- Christian Jordan
- Michael Harasek
- Clemens Gößnitzer
- Sylvia Zibuschka
- Yitong Chen
- Vikram Natarajan
- Jozsef Nagy

TU WIEN ICEBE IMAGINEERING NATURE

Technische Universität Wien
Institute of Chemical, Environmental
& Bioscience Engineering

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.

Available from: www.fluiddynamics.at

Background

1. Large eddy simulation (LES)

In Large Eddy Simulation (LES), turbulence is modeled by distinguishing between large-scale eddies and small-scale eddies within a fluid flow. The fundamental idea behind LES is that large eddies are dependent on the geometry and flow conditions, whereas small eddies exhibit more universal behavior. This assumption allows for a computationally efficient approach to turbulence modeling by resolving only the large eddies while modeling the small-scale eddies using Sub-Grid Scale (SGS) models.

Compared to Reynolds-Averaged Navier-Stokes (RANS) models, which completely model turbulence effects, LES provides a higher-fidelity simulation since large eddies are explicitly resolved rather than approximated. However, LES requires higher computational resources than RANS but significantly less than Direct Numerical Simulation (DNS), making it an effective trade-off between accuracy and computational feasibility.

Mathematically, it is like separating the velocity field into a resolved and sub-grid part using a filter function. The resolved part of the field represents the large eddies, while the sub grid part of the velocity represents the small eddies whose effect on the resolved field is included through the sub grid-scale model. Formally, one may think of filtering as the convolution of a function with a filtering kernel G:

$$\bar{u}_i(\vec{x}) = \int G(\vec{x} - \vec{\xi})u(\vec{\xi})d\vec{\xi}$$

resulting in

$$u_i = \bar{u}_i + u'_i$$

Where $\bar{u}_i$ is the resolvable scale part and u'_i is the subgrid-scale part. However, most practical (and commercial) implementations of LES use the grid itself as the filter and perform no explicit filtering. The filtered equations are developed from the incompressible Navier-Stokes equations of motion:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\upsilon \frac{\partial u_i}{\partial x_j}\right)$$

Substituting in the decomposition $u_i = \bar{u}_i + u'_i$ and $\mathrm{p} = \bar{\mathrm{p}} + \mathrm{p}'$ and then filtering the resulting equation gives the equations of motion for the resolved field:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\upsilon \frac{\partial \bar{u}_i}{\partial x_j}\right) + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j}$$

We have assumed that the filtering operation and the differentiation operation commute, which is not generally the case. It is thought that the errors associated with this assumption are usually small, though filters that commute with differentiation have been developed. The extra term $\partial\tau_{ij}/\partial x_j$ arises from the non-linear advection terms, because:

$$u_j \frac{\partial u_i}{\partial x_j} \neq \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j}$$

and hence

$$\tau_{ij} = \bar{u}_i\bar{u}_j - \overline{u_i u_j}$$

Similar equations can be derived for the sub grid-scale field. Sub grid-scale turbulence models usually employ the Boussinesq hypothesis, and seek to calculate (the deviatoric part of) the SGS stress using:

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_t\bar{S}_{ij}$$

where $\bar{S}_{ij}$ is the rate-of-strain tensor for the resolved scale defined by

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right)$$

and μ_t is the subgrid-scale turbulent viscosity. Substituting into the filtered Navier-Stokes equations, we then have:

$$\frac{\partial\bar{u}_i}{\partial t} + \bar{u}_j\frac{\partial\bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial\bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left([\upsilon + \upsilon_t]\frac{\partial\bar{u}_i}{\partial x_j}\right)$$

where we have used the incompressibility constraint to simplify the equation and the pressure is now modified to include the trace term $\tau_{kk}\delta_{ij}/3$.

2. k-Eqn model

The k-equation (kEqn) Large Eddy Simulation (LES) turbulence model is designed to capture sub-grid scale (SGS) turbulence effects by solving a transport equation for the SGS turbulence kinetic energy (k). This approach enhances the accuracy of simulations involving complex turbulent structures.

The governing equation for the SGS turbulence kinetic energy k in the kEqn LES model is:

$$\frac{\partial\rho\mathrm{k}}{\partial t} + \nabla\cdot(\rho\, u\, k) = \nabla\cdot(\rho\, D_\mathrm{k}\,\nabla k) + \rho G - \frac{2}{3}\rho k\,(\nabla\cdot u) - \frac{C_\mathrm{e}\,\rho\, k^{1.5}}{\Delta} + \mathrm{S_k}$$

Where G is the production term of turbulence kinetic energy, C_e is the model coefficient (default value: 1.048), Δ is the filter width and S_k is the source term.

This equation accounts for the transport, production, and dissipation of SGS turbulence kinetic energy, providing a comprehensive representation of turbulent flow dynamics.

3. Smagorinsky-Lilly model

One of the most widely used SGS models is the Smagorinsky-Lilly model, which provides a simple way to estimate the sub-grid scale eddy viscosity:

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2(\mathrm{C}_s\Delta)^2|\bar{S}|\mathrm{S}_{ij}$$

In the Smagorinsky-Lilly model, the eddy viscosity is modeled by

$$\mu_{sgs} = \rho(C_s\Delta)^2|\bar{S}|$$

Where the filter width is usually taken to be

$$\Delta = (Volume)^{1/3}$$

and

$$\bar{S} = \sqrt{2S_{ij}S_{ij}}$$

The effective viscosity is calculated from

$$\mu_{eff} = \mu_{mol} + \mu_{sgs}$$

The Smagorinsky constant usually has the value: $C_s = 0.1 - 0.2$

Physical Interpretation

- The Smagorinsky model assumes that the energy cascade in turbulence is local, meaning small eddies interact mostly with nearby structures.
- The filter width Δ\Delta determines the size of the smallest resolved structures.
- The Smagorinsky constant C_s is a tunable parameter that affects model accuracy.
    - Higher C_s leads to stronger damping of small eddies.
    - A lower C_s may lead to unresolved turbulence effects.

Smagorinsky model is simple and computationally efficient, while providing reasonable approximations for turbulent energy dissipation, it works well for high-Reynolds-number flows. On the other hand, the model does not account for near-wall effects accurately, leading to overdamping of turbulence in boundary layers and the constant C_s needs tuning for different flow conditions.

incompressibleFluid – pitzDaily

Tutorial outline

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

- Smagorinsky (LES)
- kEqn (LES)
- kEpsilon (RAS)

Objectives

- Understanding turbulence models
- Transient vs steady state simulation
- Finding appropriate turbulence model

Data 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 (Tutorial Six).

1. Pre-processing

1.1. Copying tutorial

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

`$FOAM_TUTORIALS/incompressibleFluid/pitzDailyLESDevelopedInlet`

Replace 0 directory with 0 directory from the following tutorial:

`$FOAM_TUTORIALS/incompressibleFluid/pitzDaily`

1.2. 0 directory

Set the proper turbulence model initial and boundary conditions and values.

Note: *For different turbulent models, different files should be modified (check Tutorial Six).*

For kEpsilon model, the epsilon file need to be added and on the walls, for all three properties: k, epsilon and nut, the wall-functions should be applied (based on the y$^+$ value) and proper initial values to be set. For more information: https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence.html

1.3. constant directory

As mentioned in Tutorial Six, in *momentumTransport* the turbulent model type has to be set. The `simulationType` can be changed to `LES` or `RAS`. Depending on which type is selected, the corresponding sub-dictionary needs to be specified. Below is the *momentumTransport* file for the kEqn model, which is an LES model.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * *//
simulationType  LES;

LES
{
LESModel        kEqn;
turbulence      on;
printCoeffs     on;
delta           cubeRootVol;

dynamicKEqnCoeffs
{
    filter      simple;
}

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;
}
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * *//
```

Note: For Smagorinsky, you can find the sample dictionary, including the relevant settings in the following:

```
$FOAM_TUTORIALS/multiphaseEuler/LBend/constant/momentumTrasport.gas
```

2. Running simulation

```
>blockMesh
```

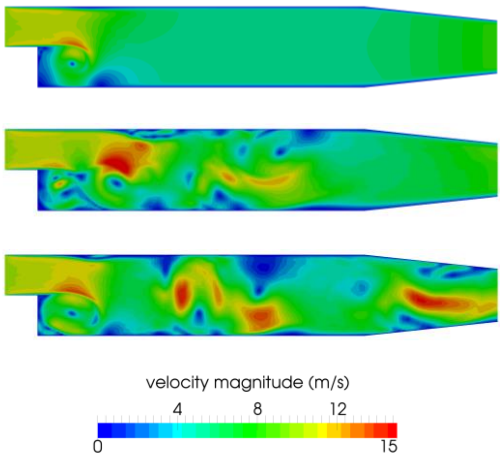
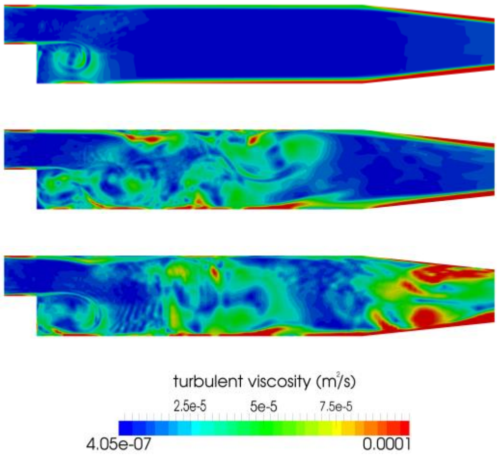
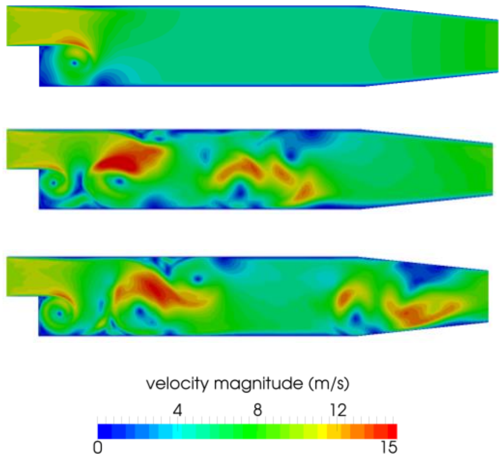
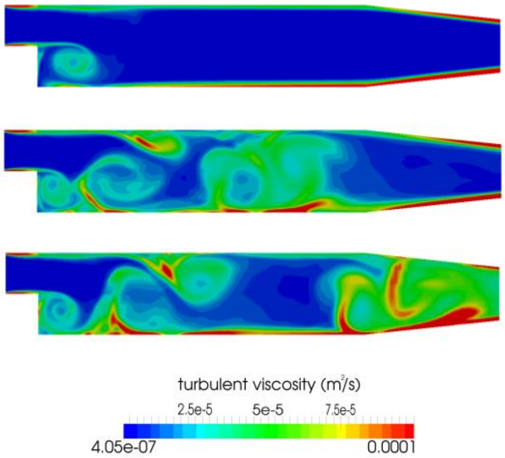
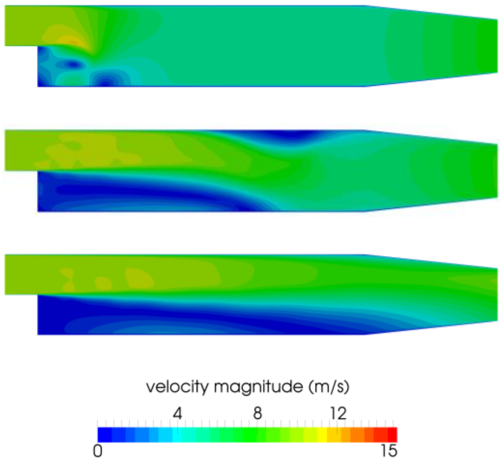
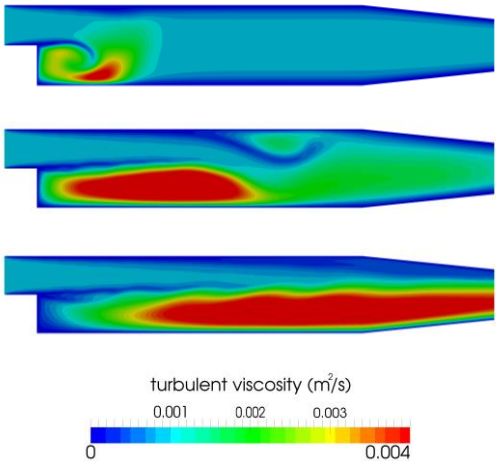
```
>foamRun -solver incompressibleFluid
```

3. Post-processing

The simulation results are as follows:

For the kEpsilon model after 0.2 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 tutorial).

	Velocity magnitude	Turbulent viscosity
Smagorinsky		
0.01 s **0.05 s** **0.2 s**		
kEqn		
0.01 s **0.05 s** **0.2 s**		
kEpsilon		
0.01 s **0.05 s** **0.2 s**		

Comparison of different turbulent models for transient simulation.