M Part 1 of Assignment 2: Large Eddy Simulation

You can do the assignment on your own or in a group of two. You will receive data from a DNS of fully developed flow in a channel. It is recommended (but the not required) that you use LaTeX(an example of how to write in LaTeXis available on the course www page). It is available on Linux. On Windows you can use, for example, Lyx (`www.lyx.org`) or MikTex (`www.miktex.org`) which are both free to download.

The equations that have been solved are given in Eq. L.1. The Re number based on the friction velocity and the half channel width is $Re_\tau = u_\tau h/\nu = 500$ ($h = \rho = u_\tau = 1$ so that $\nu = 1/Re_\tau$).

A $96 \times 96 \times 96$ mesh has been used. The streamwise, wall-normal and spanwise directions are denoted by x (x_1), y (x_2) and z (x_3) respectively. The cell size in x and z directions are $\Delta x = 0.0654$ and $\Delta z = 0.0164$. Periodic boundary conditions were applied in the x and z direction (homogeneous directions). The size of the domain is (L, h, Z_{max}) in (x, y, z), see Fig.L.1.

M.1 Time history

At the www-page

`http://www.tfd.chalmers.se/~lada/comp_turb_model/`

you find a file `u_v_time_4nodes.dat` with the time history of u and v. The file has eight columns of u and v at four nodes: $y/\delta = 0.0039$, $y/\delta = 0.0176$, $y/\delta = 0.107$ and $y/\delta = 0.47$. With $u_\tau = 1$ and $\nu = 1/Re_\tau = 1/500$ this correspond to $y^+ = 1.95$, $y^+ = 8.8$, $y^+ = 53.5$ and $y^+ = 235$. The sampling time step is 0.0033 (every second time step).

Use Matlab or Octave. Octave is a Matlab clone which can be downloaded for free. Download the Matlab/Octave program `pl_time.m` which loads and plots the time history of u. Run the program `pl_time.m`. Recall that the velocities have been scaled with the friction velocity u_τ, and thus what you see is really u/u_τ. The time history of u at $y/\delta = 0.0176$ and $y/\delta = 0.107$ are shown. Study the time history of the blue line ($y/\delta = 0.0176$) in greater detail. Make a zoom between, for example, $t = 10$ and $t = 11$ and $u_{min} = 3$ and $u_{min} = 21$. This is conveniently done with the command

```
axis([10 11 3 21])
```

In order to see the value at each sampling time step, change the plot command to

```
plot(t,u2,'b-',t,u2,'bo')
```

Use this technique to zoom and to look at the details of the time history. Alternatively, you can use the zoom buttons above the figure.

Plot u for all four nodes. How does the time variation of u vary for different positions? Why? Plot also v at the four different positions. What is the differences between u and v?

M.2 Time averaging

Compute the average of the u velocity at node 2. Add the following code (before the plotting section)

```
umean=mean(u2)
```

Here the number of samples is $n = 5000$ (the entire `u2` array). Find out how many samples must be used to get a correct mean value. Start by trying with 100 samples as

```
umean_100=mean(u2(1:100))
```

Do the same exercise for the other three nodes.

M.3 Auto correlation

Auto correlation is defined in Section 10.2 Compute the autocorrelation for v_1' using the Matlab command

```
imax=500;
two_uu_1_mat=autocorr(u1,imax);
```

where we set the maximum separation in time to $imax = 500$ (i.e. we carry out the integration in Eq. 10.11 not to infinity, but to $imax \cdot \Delta t$). Note that the `autocorr` command returns the normalized autocorrelation, i.e. B_{11}^{norm}, see Eq. 10.10. Plot the autocorrelation as

```
plot(t(1:imax),two_uu_1_mat(1:imax),'linew',2)
xlabel('t')
ylabel('u')
handle=gca
set(handle,'fontsi',[20])
```

Compute the integral time scale B_{11} as

```
dt=t(1);
int_T_1=trapz(two_uu_1_mat)*dt;
```

Plot the normalized autocorrelation and compute the integral time scales also for the other three points.

In Section M.1 you time averaged the velocities to get the mean value. You investigated how few samples you could use. In reality it is not only the number of samples that is relevant, but also that they are *independent*. To find out if two samples are independent, it is convenient to use the integral time scale, T_{int}. If the samples are separated by T_{int} seconds they are independent. Hence, re-do the averaging you did in Section M.2 but use samples every T_{int} second.

Let us use Taylor's frozen turbulence hypothesis to compute the integral length scale. This hypothesis assumes that – if the turbulence level is not too strong – the velocity fluctuation at point x and time t is the same as that at time $(t - \tau)$ at point $(x-\xi)$ where $\tau = (x-\xi)/\langle u \rangle$ (it takes time τ for the particle to travel from point $(x-\xi)$ to point x with a velocity $\langle u \rangle$). The hypothesis assumes that the turbulence is frozen between point $(x - \xi)$ and x. When we want to find the velocity fluctuation at point $(x-\xi)$ at time $(t-\tau)$ we can instead take it at point x at time t. The Taylor hypothesis makes it possible to compute the integral lengthscale from the integral timescale as

$$L_{int} = \langle u \rangle \int_0^\infty B_{11}^{norm}(\hat{t}) d\hat{t} = \langle u \rangle T_{int} \tag{M.1}$$

Compute the integral lengthscale.

M.4 Probability density/histogram

Histogram (also called probability density function, PDF) can give additional useful information, see Section 7. With a probability density, f_v, of the v velocity, the mean velocity is computed as

$$\langle v \rangle = \int_{-\infty}^{\infty} v f_v(v) dv \tag{M.2}$$

Normalize the probability density function so that

$$\int_{-\infty}^{\infty} f_v(v) dv = 1 \tag{M.3}$$

Here we integrate over v. The mean velocity can of course also be computed by integrating over time, as we do when we define a time average. Compute the PDF as

```
u3_fluct=u3-mean(u3);
[pdf3 u3_pdf]= hist(u3_fluct,20)
```

Here we have divided u_3 into 20 bins which span the variation of u_3, i.e. $[\min(u_3), \max(u_3)]$. The variable $pdf3$ is a vector of length 20 whose elements gives the number of samples in each bin. Plot the histogram as

```
norm3=sum(pdf3)*(u3_pdf(2)-u3_pdf(1));
plot(u3_pdf,pdf3/norm3,'linew',2)
xlabel('u3')
ylabel('PDF')
handle=gca
set(handle,'fontsi',[20])
```

where $norm3$ is the integral in Eq. 7.3. You find that the PDF is rather symmetric. Compute and plot the PDFs of the points close to the wall and you will find that they are more skewed. Skewness, S, is a variable that quantify the skewness and it is defined as

$$S_{v'} = \frac{1}{v_{rms}^3} \int_{-\infty}^{\infty} v'^3 f_{v'}(v') dv'$$

Compute it as

```
urms3=std(u3_fluct);
S=mean(u3_fluct.^3)/urms3^3;
```

where $urms3 = \langle v_3'^2 \rangle^{1/2}$. Verify that the magnitude of S is large for the walls close to the wall.

M.5 Frequency spectrum

One way to verify that the LES you have performed resolves the turbulence properly, is to look at the spectra of the resolved turbulence. One can analyze the time history of a variable at a point. You do a FFT of that signal to get the Fourier coefficients a_i and then plot a_i^2. Then you get the frequency spectrum, i.e. how much energy resides in each frequency. The other way to do it is to look at the instantaneous velocity along a grid line, and do a FFT of that signal. Then you get the energy spectrum as a function

of the wave number, i.e. the inverse of the wave length. In this case you must average over many instants to get a reasonably smooth spectrum.

The Matlab file `pl_spectrum.m` does a FFT of the time history of fluctuating velocity, u', ($u = \langle u \rangle + u'$) and plots a_i^2. The same data file as in the previous exercise is used (`u_v_time_4nodes.dat`). It uses the Matlab function `pwelch`. Run the program by typing

```
pl_spectrum
```

As we mentioned in the Lecture Notes, in a well-resolved LES we want to have the cut-off in the inertial subrange where the kinetic energy decays as the wave number (or frequency) up to the power of $-5/3$. Thus we want the resolved turbulence to have a behaviour like this for high frequencies. As you can see from the plot, this is the case. Actually, you find that for even higher frequencies the kinetic energy decays even faster. The high frequencies exist because we over-resolve in time compared to the resolution in space. This is usually the case when the maximum CFL number (Courant-Friedrichs-Lewy condition) is set to one, since CFL in many points is much smaller than one (CFL=1 in one cell means that a fluid partcle is transported across that cell during one time step). Investigate this by using only, say, every 4^{th} of the samples, i.e.

```
% 4*dt
m=4;
i=1:m:n;
dt=m*t(1);
[px_L,f_L]=pwelch(u4(i),nmax,[],[],1/dt);
plot(f_L,px_L,'r--','linew',2)
```

Has the region in which the energy spectrum decays fast vanished? If not, use even coarser sampling (set `m=8`). You may note that the low region of the frequency spectrum is also modified. The coarser sampling that is used, the lower frequencies appear in the spectrum. The reason is that we are using `nmax=256` samples, and the coarser the sampling, the higher the lowest frequencies.

Look at the spectra for the three other points (make changes in `pl_spectrum.m`).

It should be mentioned that spectra may not be a reliable measure of resolution [102, 103]. Two-point correlations are usually better.