ME647 - Assignment 1

Instructions:

- Submit by **12/02/2025**, **1800 hrs**. Late submissions upto 1 day will be graded out of 60%, and later than 1 day will not be graded.
- If you make use of AI assistance for the coding, you need to *disclose it clearly* (there is no penalty). Remember that the solutions you get may not be accurate. Moreover, you **must be able to explain** and stand by your submitted solutions and codes.
- Submit your solutions as 1 zip file, called "FullName.zip" on HelloIITK. This should contain 1 PDF "FullName.pdf" with all the solutions and figures (make sure labels and fonts on figures are readable in the PDF). Also submit your Python/Matlab scripts - combine all the functions and codes to one file "FullName.py", which has different sections, and not multiple code files.
- Bonus 5 marks will be given if your answers are formatted in LaTeX (and are correct).
- While discussing with your classmates is encouraged, there will be **zero tolerance towards plagiarism**.

---

1 Fluid Flow Equations

The incompressible Navier-Stokes equations can be written for the $i-$th velocity component as

$$\frac{\partial u_i}{\partial t} + u_j \partial_j u_i = -\partial_i (p/\rho) + \nu \partial_j \partial_j u_i \tag{1}$$

along with the incompressibility constraint $\partial_i u_i = 0$. We will refer to Eq.1 as $(\text{NSE})_i$.

1. Derive the equation for the kinetic energy, by multiplying $(\text{NSE})_i$ with u_i, and defining $k = u_i u_i/2$. If we further integrate this equation over volumes V bounded by surface $\mathrm{d}\Omega$, far from physical walls and in the middle of a turbulent flow, which terms can be neglected and why? Give an interpretation for the remaining terms you find in the final equation. (4 + 1 Marks)
2. Analogously to the kinetic energy, we often encounter *enstrophy* which is given as $\omega_i \omega_i/2$. Take the curl of the Navier-Stokes equation by multiplying it with $\epsilon_{pqi}\partial_q(\text{NSE}_i)$ to obtain the transport equation of vorticity ω_p. Then multiply with ω_p and obtain an equation (somewhat) similar to the kinetic energy equation. Compare the terms with the kinetic energy equation. (5 + 5 Marks)

Show all the steps in the above derivations.

2 Turbulence Data Analysis

Attached you will find a two-dimensional cut from a direct numerical simulation of three-dimensional homogeneous, isotropic turbulence from the Johns Hopkins Turbulence Database. The details of your dataset are as follows, along with dimensionless parameters:

- Grid Size: $N_x \times N_y = 1024 \times 1024$ grid cells (which is a 2D plane, at $z = 0$, from a 1024^3 simulation). The lateral size $L_x = L_y = 2\pi$, and the length of each grid cell is $dx = L/N_x$. The kinematic viscosity $\nu = 0.000185$ and integral lengthscale $\mathcal{L} = 1.364$.
- Velocity fields in dataset: u, v, w or u_i with $i \in \{1, 2, 3\}$
- Boundaries: Periodic (i.e. $u_i(l, y) = u_i(l + 2\pi, y)$)

You will also find a simple Python code to load this data. Note that 1) Repeated indices, like $u_i u_i$, imply summation in Einstein notation, 2) Angle brackets $\langle \cdots \rangle$ denote spatial-averaging over all points. **Calculate and plot the following:**

1. Calculate the Reynolds Number and Kolmogorov Scales η, τ_η and u_η. For the large scales, use $\mathcal{L}$ as lengthscale and the root-mean-square velocity $\langle u^2(\mathbf{x})\rangle^{1/2}$ as a velocity scale U. What is the grid cell size with respect to the Kolmogorov length? (5 Marks)

2. Plot the two-dimensional normalized kinetic energy field $\widetilde{k} = k/\langle k \rangle$ where $k(\mathbf{x}) = u_i(\mathbf{x})u_i(\mathbf{x})/2$. Add a colorbar to the figure. (3 Marks)

3. Calculate and plot the vorticity. Since you have planar data, you can compute only one vorticity component with the in-plane velocity cross gradients. Hence calculate $\omega_z(\mathbf{x})$. Use a second-order central differencing scheme for approximating the gradients required. Plot the planar vorticity field normalized by its root-mean-square value $\omega_{\mathrm{rms}} = \langle \omega_z^2(\mathbf{x})\rangle^{1/2}$, with a colorbar. (5 Marks)

4. Plot the **PDFs**, **CDFs**, **Skewness** and **Kurtosis** of the quantities mentioned below, where each quantity is first normalized by its standard deviation and is centered at 0. Check that the pdf integrates to 1. Add a normal (Gaussian) distribution with mean 0 and standard deviation of 1 i.e. $\mathcal{N}(0, 1)$ as a guide-to-eye to compare with the various quantities in each figure. *Make sure to plot these on a logarithmic y-axis.* (14 Marks)

   (a) Each velocity component—u, v, w—in the same figure. Use different symbols to show them and add a legend.

   (b) The vorticity component ω_z.

   (c) The velocity gradients $\partial u/\partial x$ and $\partial v/\partial y$.

   (d) The enstrophy ω_z^2.

   (e) Lastly, plot the PDF of $u(x, y)$ for the range $x \in [0, 128], [129, 256], [257, 512]$ (using all of y), as 3 distributions in one figure. Here I am using x as grid index and not spatial location. What does this tell you about the data?

5. Let us consider the u velocity field. Use this to calculate and plot an approximate non-linear term $\mathrm{N_{lin}} = u_j \partial_j u \approx u\partial_x u + v\partial_y u$ and an approximate viscous term $\mathrm{Visc} = \nu\partial_j\partial_j u = \nu(\partial_x\partial_x + \partial_y\partial_y)u$. You can now plot the local Reynolds number field $\mathrm{Re} = \frac{|\mathrm{N_{lin}}|}{|\mathrm{Visc}|}$. How does this compare to the kinetic energy and vorticity fields? Summarize your observations. (8 Marks)

Hints and Suggestions

- You may use the following syntax to plot the 2D fields:

```
skip = 5
plt.pcolor(ke[::skip,::skip], vmin=0, vmax=10)
```

  (The above code will plot every 5th point of the dataset in the x and y directions, hence reducing memory usage. You can change vmin and vmax to alter the range where the data is clipped, to be able to show structures more clearly. For fields with negative values, use vmin=-val and vmax=+val, for some value of val, to have equal range on both sides.

  You can finally change to skip=1, to plot the full field, when you are satisfied with your plot.

- For PDFs, you should first create a histogram of the data, for which you will need to send in a 1D array to the histogram function, along with the number of bins. You can use, for example

```
hist, bins = np.histogram(field.ravel(), bins=100)
```

  to return a histogram of the field over 100 bins (the function actually returns the bin-edges, hence "bins" is 1 longer than the length of "hist"). You will have to convert this histogram to a PDF. The CDF is simply a cumulative integral that will run over the bin range you have created, which you can do with either a loop or with a python list

```
cdf = [ np.sum(pdf[:i]) for i in range(len(hist)) ]
```

- I highly recommend that you save the calculated PDFs and CDFs in small text files, and use a different script to plot the data. Typically plotting takes several iterations to optimize, this will save you time as you don't need to do the longer PDF calculation each time, and can simply read a small text file and plot it. You can use

```
np.savetxt(...)
np.loadtxt(...)
```

- You can save figures using

```
plt.savefig("Figure1.png", bbox_inches='tight', dpi=240)
```

- You can calculate Skewness and Kurtosis from the data field directly, and not as moments of the PDF.

- For a second order central differencing scheme, you may want to explore using the following function for faster calculations, rather than two nested loops over x and y.

```
np.roll( ... )
```

  Of course, do not forget to divide the 'differences' with the right length, to approximate $\partial u/\partial x \approx \Delta u/\Delta x$, where Δx will be measured in terms of the grid cell length.

- The Laplacian $(\partial_x\partial_x + \partial_y\partial_y)u$ can be calculated by applying a five point stencil.

All the best!