ME647 - Assignment 2 (50 Marks)

Please read the below INSTRUCTIONS *carefully*:

- Submit by **23/03/2025, Sunday**, **1800 hrs**. Late submissions upto 1 day will be graded out of 60%, and later than 1 day will not be graded.
- **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.**
- If you use Jupyter notebooks, please convert the code-only parts to a ".py" and share, MATLAB users can share ".m".
- Assignments not submitted as one, well annotated PDF, **will not be graded.**
- 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**.

Turbulence Scaling Laws

You can work with the same data as the previous assignment: 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. $u_i(l, y) = u_i(l + 2\pi, y)$)

Calculate, plot, and comment on the following measurements:

1. Verify Parseval's theorem using a line-cut of the data (say u along x at $y = 0$) and its Fourier transform. Explain the result. (2 Marks)
2. The energy spectra $E(k) = (|\widehat{u}|^2 + |\widehat{v}|^2)/2$, where $\widehat{u}$ is the Fourier Transform of the u velocity field and k is the wavenumber. Calculate the energy spectra along lines in the $x-$direction, and average over the y direction. Plot the averaged energy spectrum as a function of k (calculate the k values based upon the length of the data and resolution, appropriately) on a *log-log scale*. Label and show the scaling exponent $k^{-5/3}$ in the inertial range with a line-fit. (5 Marks)

3. In the above example, the spectra is calculated in 1D, and hence k is simply a wavenumber. Now compute the two-dimensional energy spectrum using the full 2D velocity fields (u, v) to obtain $E(\mathbf{k})$ where $\mathbf{k} = k_x\hat{i} + k_y\hat{j}$. This is a spectral field in wavevector space. Perform a shell-averaging over wavenumber shells $k - 1/2 \leq k < k + 1/2$ for $k \in [k_{\min}, k_{\max}]$, to obtain the spherically averaged energy spectrum $E(k) = \sum_{k-1/2 \leq k < k+1/2} E(\mathbf{k})$ over the scalar wavenumber $k = \sqrt{k_x^2 + k_y^2}$. Again label and show the scaling exponent in the inertial range with a line-fit, and compare the spectra to the one in the previous problem. (10 Marks)

4. Plot the longitudinal and transverse velocity correlation functions from the velocity fields. (8 Marks)

5. Longitudinal structure functions are defined as

$$S_p(r) = \langle \Delta u^p(r) \rangle = \left\langle \left| \left([\mathbf{u}(\mathbf{x} + \mathbf{r}) - \mathbf{u}(\mathbf{x})] \cdot \frac{\mathbf{r}}{|r|} \right) \right|^p \right\rangle \propto r^{\zeta_p} \qquad (1)$$

where the angle brackets denote averaging over all orientations of $\mathbf{r}$ (for a fixed $|\mathbf{r}|$) and over many centres $\mathbf{x}$. For ease of coding, you can take $\mathbf{r}$ along the $x-$direction alone, or for more accuracy, take arbitrary orientations of $\mathbf{r}$ at each point $\mathbf{x}$. Keep in mind, and use the fact, that the data is periodic in space. Perform the averaging over many points!

   (a) Plot $S_p(r)$ v/s r for $p \in \{1, 2, 3, 4, 5, 6, 7\}$ and $r \in [0, L/2]$ on a log-log scale. Plot these in the same figure, with clean legends and labels. (10 Marks)

   (b) Verify Kolmogorov's $4/5th$ law for $S_3(r)$. (3 Marks)

   (c) It is typically difficult to find the scaling (inertial) range to obtain the structure function scaling exponents ζ_p (it needs a *lot* of statistics). You can, however, plot $S_p(r)$ v/s $S_3(r)$ instead, on a log-log scale, which improves the scaling range significantly, i.e. you find the scaling of $S_p(r)$ versus $S_3(r)$, and not with r itself. This technique, used widely in turbulence study, is known as *ESS: Extended Self-Similarity* (note that $\zeta_3 = 1$, by default in ESS). Plot the ESS profiles of all the $S_p(r)$, on the same plot, using clear legends and labels. (7 Marks)

   (d) To the ESS plots above, add line fits in the scaling-range to obtain the scaling exponents ζ_p and errorbars for the different $p-$values. Then plot the scaling exponents ζ_p v/s p, with errorbars, against Kolmogorov's predction of $\zeta_p = p/3$. What is the source of this anomalous scaling of exponents, and why does the deviation prominently increase at large values of p? (5 Marks)

Hints and Suggestions

- To calculate the Fourier transform, look into the following functions:

```
np.fft.fftn, np.fft.fftshift, np.fft.fftfreq
```

  Using np.fft.fftshift over the n-dimensional Fourier transform rearranges the data such that the zeroth-mode falls in the center of the domain, and hence Fourier modes with the same wavenumber $|\mathbf{k}|$ are on a thin spherical-shell of radius k, in the range $k - 1/2 \leq k < k + 1/2$. You can then calculate the relative distance of each point in spectral space to this center, which will give the wavenumber corresponding to each point.

- Practically, $k - 1/2 \leq k < k + 1/2$ is equivalent to simply binning all the values in the range $k \in [k, k + 1]$ to the wavenumber k.

- Use Linear Regression linear fits to the log-log data, in the scaling range only. For this you may use

```
np.polyfit
```

which gives both the linear fit, as well as the fitting error.

All the best!