ME647 - Assignment 3 (75 Marks)

Please read the INSTRUCTIONS *carefully*:

- Submit by **20/04/2025, 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.**
- While discussing with your classmates is encouraged, there will be **zero tolerance towards plagiarism**.

Data

The details of your dataset, an extract from Johns Hopkins Turbulence Database, are as follows, along with dimensionless parameters:

- Grid Size: $N_x \times N_y \times 3 = 1024 \times 1024 \times 3$ grid cells (i.e. a stack of 3 crossections, at $z = 0, 1, 2$, from a 1024^3 simulation). The lateral size $L_x = L_y = 2\pi$, and the length of each grid cell is $dx = dy = dz = L/N_x$. The kinematic viscosity $\nu = 0.000185$ and integral lengthscale $\mathcal{L} = 1.364$.
- Data ordering is of the form u[x,y,z] — First index is the x-direction, second index is the y-direction, third in the z-direction.
- Boundaries: Periodic (i.e. $u_i(l, y) = u_i(l + 2\pi, y)$) along (x,y). The data-slice is not periodic in z.

1 Coherent Structures and Flow Topology

Read: Section 5.3.6.3 - "*Properties of the velocity gradient tensor*" in P.A. Davidson, and the papers shared.

The velocity gradient tensor A_{ij} at any point is given as $\partial_j u_i$ (it has 9 components, for each point in space). This tensor at each point can be diagonalized to its eigenvalues $\lambda_1, \lambda_2, \lambda_3$, ordered as $\lambda_1 > \lambda_2 > \lambda_3$. The three invariants of this tensor, P, Q and R are related to fundamental *flow-topologies*. Due to incompressibility, $P = \sum_i \lambda_i = 0$, and hence structures are described in the QR-plane, with vortex-stretching, vortex-compression, stable-node-saddle-saddle (axial strain) and unstable-node-saddle-saddle (biaxial strain), as shown below in Fig. 1.

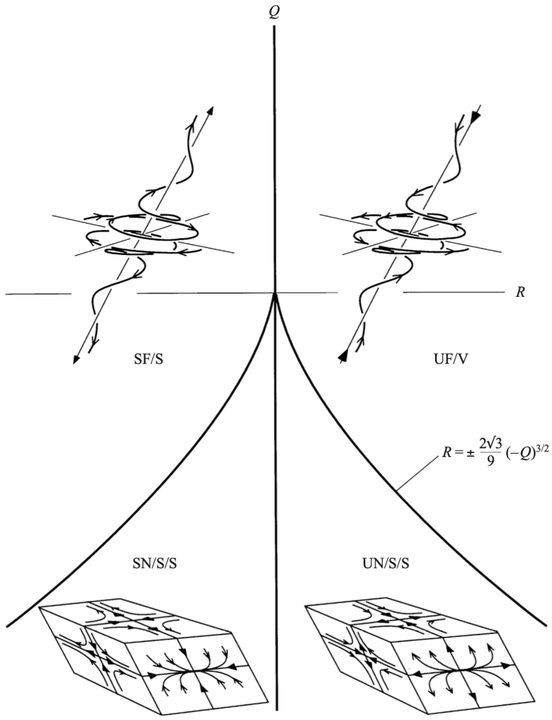

FIGURE 1. Non-degenerate local topologies for incompressible flows.

1. Write down the characteristic equation for A_{ij} and determine the three invariants P, Q, R in terms of the eigenvalues, and in terms of S_{ij} and R_{ij}. (5 Marks)

2. For each point in the 2D domain, calculate the velocity gradient tensor A_{ij}. Find the eigenvalues λ_i, and plot the PDF of λ_1, λ_2 and λ_3 (with the y-axis on a **log scale** and x-axis on **linear scale**, i.e. a semilog plot). (5 Marks)

   Note: The shape of the data you have now is $\mathbf{u}[N_x, N_y, 3]$, i.e. there are three planes in the z-direction. Therefore, you can work on the *middle plane* ($\mathbf{u}[:,:,1]$ in python or $\mathbf{u}[:,:,2]$ in Matlab), and calculate the z-direction gradients $\partial/\partial z$ using central differencing as well. The data is periodic in x, y, but not in z. Be careful if using in-built functions for gradient calculations.

3. Verify whether $\langle |P| \rangle \approx 0$ over the 2D domain, where $\langle \rangle$ is spatial averaging. (2 Marks)

4. Plot the fields of Q/Q_{rms} and R/R_{rms}. Make sure to pick the range of the colorbar such that the fields are clearly visible. Write your observations about the spatial organization of Q relative to the spatial organization of R. (8 Marks)

5. Make a scatter-plot/joint-distribution of $Q/\langle Q_w \rangle$ v/s $R/\langle Q_w \rangle^{3/2}$, where $\langle Q_w \rangle = \langle \omega^2 \rangle / 4$ in the $Q - R$ plane. Draw the descriminant line $27R^2/4 + Q = 0$. Check whether you recover the "***tear-drop***" profile of QR, which is *universal* in turbulence. Which two flow-topologies are more prominent? How does this link to the enstrophy equation you derived in an earlier assignment? (10 Marks)

   Suggestion: For the join-probability-distribution, you can mask out values below a small number (like 10^{-7} or so) with "np.NaN", to present the data more clearly. Since this is also a probability plot, the levels should be logarithmically-spaced for clarity. Pick axis range values appropriately for the profile to be clearly visible.

2 Lagrangian Aspects of Turbulence

A Lagrangian tracer particle simply follows the equation $\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t)$. Numerically this simply reduces to $\mathbf{x}_p^{t+\Delta t} = \mathbf{x}_p^t + \mathbf{u}(\mathbf{x}_p^t)\Delta t$, for the $p-$th tracer particle, using simple Euler integration (you can use better schemes if you know). You can work under the following assumptions and criteria:

- The velocity field of course evolves in time. However, here you can treat the velocity field to be *frozen*, and using only the u and v velocity components, evolve particles in-plane (any of the 3 z-planes are fine, but fix one). The Δt, therefore, gives you the evolution of a *pseudo-time* for tracers moving in a fixed velocity field.
- Randomly initialize N_p tracer particles in the 2D domain. Here note that the *physical* length is $0-2\pi$, whereas the index-based length is $0-N_x$. Hence, you have to interconvert correctly to evolve the particles over the *physical length* (since you are using the physical velocity) but probe for velocities at the current particle location (like $u[x_p, y_p], v[x_p, y_p]$) using index-based length.
- Naturally, x_p and y_p will *never* correspond to actual integer indices since your particle locations are always *off-grid*, hence you cannot query the data with $(u[x_p, y_p], v[x_p, y_p])$. You should, hence, obtain the velocity at particle-locations using ***bilinear interpolation***.
- By picking a small value of $\Delta t \approx \tau_\eta/10$ or so, you can smoothly evolve the particle trajectories in pseudotime t.
- When a Lagrangian trajectory exits the domain on any edge, you can use periodicity to compute its velocity. A simple way to do this, for the x direction for example, is $x_{pp} = (x_p + N_x)\%N_x$ in python, if (x_p, y_p) are index-based lengths. **Caution:** You should use this "periodic wrapping" of trajectories *only to get the velocity at particle-locations*, and *not to actually wrap the particle locations themselves*. Doing the latter ***will corrupt your calculations from MSD and diffusion below***.

1. Plot Lagrangian trajectories for $N_p = 20$ randomly initialized particles, for upto a total time of $\mathcal{T} = 1$, 5 and 10 (with $\Delta t \approx \tau_\eta/20$ or smaller, for which first calculate τ_η for this data). (10 Marks)
2. Now, when you increase N_p to 1000 (see a plot of it), some particles will appear "trapped". What could be the reason for this? (3 Marks)
3. Plot the mean-square-displacement as a function of (pseudo)time, $\langle \Delta x^2(t) \rangle = \langle |\mathbf{x}(t) - \mathbf{x}(0)|^2 \rangle$, on **loglog** scale, and check if the profile shows a ballistic ($\sim t^2$ at short times) to diffusive transition ($\sim t$ at long times). Use upto $N_p = 10^4$ particles for this calculation. (6 Marks)
4. Calculate the turbulent diffusivity coefficient D_T, by equating $\langle \Delta x^2(t) \rangle = D_T t$. How much larger is it than the diffusivity coefficient of a dye in water? (4 Marks)
5. **Richardson pair dispersion:** This measures the average separation of pairs of particles, initially separated over a distance ϵ, i.e. $\Delta r^2(t) = \langle |\mathbf{x}_1(t) - \mathbf{x}_2(t)|^2 \rangle_{N_p}$, where N_p now is the number of ***particle pairs***. For this, initialize **two-sets** of N_p particles each, say Set A and Set B. All particles of Set B are first identically initialized as Set A, and then their locations are perturbed, such that $\mathbf{x}_{B,i} = \mathbf{x}_{A,i} + \boldsymbol{\epsilon}_i$, and $|\boldsymbol{\epsilon}_i| = \epsilon$ is a two-dimensional perturbation of fixed amplitude but random direction.

(a) Plot pair trajectories for $N_p = 20$, $\epsilon = 0.5\Delta x$ (with $\Delta x = 2\pi/N_x$) and $\mathcal{T} = 10$. (6 Marks)

Note: Use different colours for Set 1 and Set 2, and plot trajectories using points, instead of connecting lines, as this gives a clearer picture. You can use:

```
plt.plot( xA, yA, '.', linestyle='None', color='k' )
plt.plot( xB, yB, '.', linestyle='None', color='r' )
```

(b) Plot the pair separation of trajectories $(\Delta r^2(t)/\epsilon^2)$ v/s t/τ_η, on a **loglog** scale, for $\epsilon/\Delta x = \{0.1, 0.5, 1.0, 5.0, 10\}$ on the same figure, for $\mathcal{T} = 10$ (if 10 is too short or too long, feel free to change it), and $N_p = 10^4$ (or less if your computer cannot take it!). Check if you observe a $\Delta r^2 \sim t^3$ scaling in the inertial range? Plot with a line fit to the loglog figure. (10 Marks)

(c) Plot the pair separation of trajectories $(\Delta r^2(t)/\epsilon^2)$ v/s t/τ_η on a **semilog** scale, where y is **log scale** and x is **linear** (time axis), for the different ϵ. Check if there is a region of exponential growth of the form $\Delta r^2 \sim \exp(\Lambda t)$, towards the early time separation (this should be a linear growth in the semilog plot). Plot a linear fit on the data, and report the value of the ***Lyapunov exponent*** Λ, which marks the region of *chaotic growth* of particle separation. (6 Marks)

All the best!