Problem Setup: The objective of this homework is to perform low-rank approximation on the data generated by simulating an incompressible flow over a cylinder. We assume the flow is incompressible and therefore it is governed by incompressible Navier-Stokes equations given by:

$$\frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2}\right), \tag{1}$$

$$\frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2}\right), \tag{2}$$

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = 0, \tag{3}$$

where $\mathbf{u}(x,y,t) = u_x(x,y,t)\mathbf{i} + u_y(x,y,t)\mathbf{j}$ is the velocity vector field, $p(x,y,t)$ is the pressure field and $Re = \rho V_\infty D/\mu$ is the Reynolds number, where ρ is the density, V_∞ is free-stream velocity, and D is the cylinder diameter. Equations (1) and (2) represent the conservation of x-momentum and y-momentum, respectively, while Equation (3) is the conservation of mass. The equations presented above are in nondimensional form, and all quantities appearing in Equations (1) through (3) are nondimensional. You do not need to use any of these equations for this homework. They are provided solely to introduce the notation.

Data: For the purpose of this assignment, the above equations were solved using the spectral element method. The snapshots of the numerical simulation of the Navier-Stokes equations for flow over a cylinder at $Re = 100$ can be found in `CYLINDER.mat`, which includes:

- `nx`: the number of mesh points in the x direction.
- `ny`: the number of mesh points in the y direction.
- `X`: the matrix of x coordinates of the mesh points. The size of the matrix is `ny` by `nx`.
- `Y`: the matrix of y coordinates of the mesh points. The size of the matrix is `ny` by `nx`.
- `dx` and `dy`: the Δx and Δy of the mesh points, respectively.
- `Ux`: the matrix of snapshots of u_x velocity. The size of the matrix is `nx.ny` by `nt`, where `nx.ny` is the total number of mesh points and `nt` is the number of time snapshots.

The snapshots of matrix `U` and `V` are the numerical solutions of Navier-Stokes equations $u_x(x,y,t_k)$ and $u_y(x,y,t_k)$, with $k = 1,\ldots,$ `nt` and $t_k = (k-1)\Delta t$, where $\Delta t = 0.2$.

Code: Two auxiliary M-files can also be downloaded from CANVAS: (1) `drive.m`: This M-file calls and plots $u_x(x,y,t)$ in a time loop; (2) `DEIM.m`: This function returns the Discrete Empirical Interpolation Method (DEIM) indices based on the QR algorithm.

In this assignment, you will only use the `Ux` matrix as the data source.

Problem 1 (Low-Rank Approximation with SVD)

1. Compute the POD modes and plot the first five POD modes. Use `svd(Ux,'econ')` for faster computation. Let `Ûx` denote the rank-r approximation of `Ux`.

2. In this part, we need to compute the data compression achieved by using SVD. To this end, store the rank-r matrix `Ûx` in the factorized form: `Ûx` $= U\Sigma V^T$, where you only need to store U, Σ and V matrices. Store these three matrices in a `mat` file. Also, store the full-rank matrix `Ux` in a `mat` file. Perform this operation for rank $r = [5, 10, 15]$. Plot the size of these files versus r. Plot the compression ratio for all these low-rank approximations.

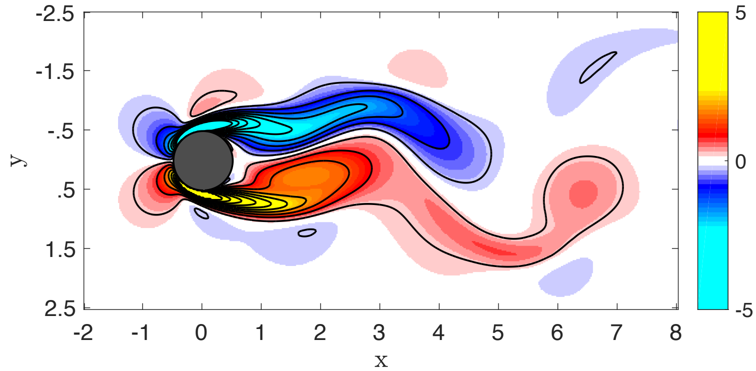

Figure 1: The contours of the $\dfrac{\partial u}{\partial y}$ for flow over cylinder at $Re = 100$.

3. Let the relative error be defined as:

$$e(r) = \frac{\|\mathtt{Ux} - \hat{\mathtt{U}}\mathtt{x}\|\|_F}{\|\mathtt{Ux}\|\|_F}, \tag{4}$$

   where $\| \cdot \|_F$ is the Frobenius norm.

Plot $e(r)$ versus r for $r = [5, 10, 15]$. Use smilogyy. Also, plot $e(r)$ versus the storage size.

Problem 2 (Low-Rank Approximation with Optimal CUR)

1. Write a code to compute the rank-r approximation of `Ux` using optimal CUR, where the exact left and right singular vectors of `Ux` are used to determine the DEIM indices. Visualize the sensor location (row indices) for $r = 5$ on a contour plot of `Ux` at $t = 0$.

2. Plot the low-rank approximation error of optimal CUR, i.e., $e(r)$, versus r for $r = [5, 10, 15]$. On the same figure, show the SVD errors calculated in Problem 1, Part 3. Explain your observations. Use smilogyy.

3. Plot the singular values of the rank-r matrix obtained from optimal CUR and those obtained from SVD. Plot these results in three figures - one for each $r = [5, 10, 15]$. Use smilogyy.

Problem 3 (Low-Rank Approximation with Iterative CUR)

1. Write a code to compute a rank-r approximation of `Ux` using iterative CUR, where the singular vectors of `Ux` are not known. Perform 10 iterations.

2. Plot the low-rank approximation error of iterative CUR, i.e., $e(r)$, versus r for $r = [5, 10, 15]$. On the same figure, show the SVD errors calculated in Problem 1, Part 3 as well as optimal CUR errors calculated in Problem 2, Part 2. Explain your observations. Use smilogyy.

3. Plot the singular values of the rank-r matrix obtained from iterative CUR versus iteration counts (from 1 to 10). On the same plot, show the singular values obtained by SVD. The SVD singular values are not a function of iteration counts. Therefore you can show them as horizontal lines. Plot these results in three figures - one for each $r = [5, 10, 15]$. Use smilogyy.