Study of Fluid Dynamics using Dynamic Mode Decomposition

A MAJOR PROJECT REPORT

19MAT117

Submitted by

Team-7

CH.EN.U4AIE20078	VENKATAKRISHNAN. R
CH.EN.U4AIE20058	SARTHAK YADAV
CH.EN.U4AIE20055	SAKILAM ABHIMA
CH.EN.U4AIE20060	SHAIK HUZAIFA FAZIL
CH.EN.U4AIE20063	B SIVA JYOTHI NATHA REDDY

In partial fulfilment for the award of the degree

of

BACHELORS OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

AMRITA SCHOOL OF ENGINEERING, CHENNAI

AMRITA VISHWA VIDYAPEETHAM

JULY,2021.

ACKNOWLEDGEMENT

We would like to offer our sincere pranams at the lotus feet of Universal guru, MATA AMRITANANDAMAYI DEVI who blessed us with her grace to make this a successful major project.

We express our deep sense of gratitude to Dr. Prasanna Kumar R, Chairperson, for his constant help, suggestions, and inspiring guidance. We are grateful to our guide Shri. Tharasi Dilleswar Rao sir, Department of Computer Science and Engineering, ASE, Chennai for his invaluable support and guidance during the major project work.

We would also like to extend our gratitude to our director Shri. Manikandan, Principal Dr. Shankar who has always encouraged us. We are also thankful to all our classmates who have always been a source of strength, for always being there and extending their valuable bits of help to the successful completion of this work.

Introduction :

The method is based on taking snapshots of data x_k from a dynamical system at a set of times t_k, where k = 1,2,3,...,m. DMD is algorithmically a regression of data onto locally linear dynamics $x_{k+1} = Ax_k$, where A is chosen to minimise $\left|\left|x_k - Ax_k\right|\right|_2$ over the k = 1,2,3,...,m-1 snapshots. The benefits of this method are its ease of implementation and the fact that it makes essentially no assumptions about the underlying system. A singular value decomposition (SVD) of the snapshot matrix created from the data x_k is used to calculate the algorithm's cost.

DMD has a wide range of applications and interpretations. The DMD algorithm, in particular, can be conceived of as enabling three primary objectives.

I) **Diagnostics :** The DMD method was first developed as a diagnostic tool for characterising complex fluid flows. The technique, in particular, extracts crucial low-rank spatiotemporal elements from a wide range of high-dimensional systems, allowing for physically interpretable results in terms of spatial structures and their related temporal responses. The diagnostic nature of DMD allows for the data-driven discovery of fundamental, low-rank structures in complex systems, similar to how POD analysis in fluid flows, plasma physics, atmospheric modelling, and other fields allows for the discovery of fundamental, low-rank structures in complex systems.

II) **State estimation and future state prediction :** Using the prevailing spatiotemporal structures in the data to develop dynamical models of the underlying processes seen is a more advanced and challenging application of the DMD algorithm. This is a significantly more complex process, especially as DMD can only build the best-fit (least-square) linear dynamical system to the nonlinear dynamical system that generates the data. In contrast to the diagnostic goal, the purpose is to predict the condition of the system in the absence of measurements. The fact that the underlying dynamics can exhibit multiscale dynamics in both time and space confounds DMD's regressive nature. Regardless, DMD is effective for constructing an usable linear dynamical model thanks to a number of essential tactics such as intelligent data sampling and updating the regression. This generative model technique can then be used to predict the future state of dynamical systems, and it has been applied successfully in a variety of applications.

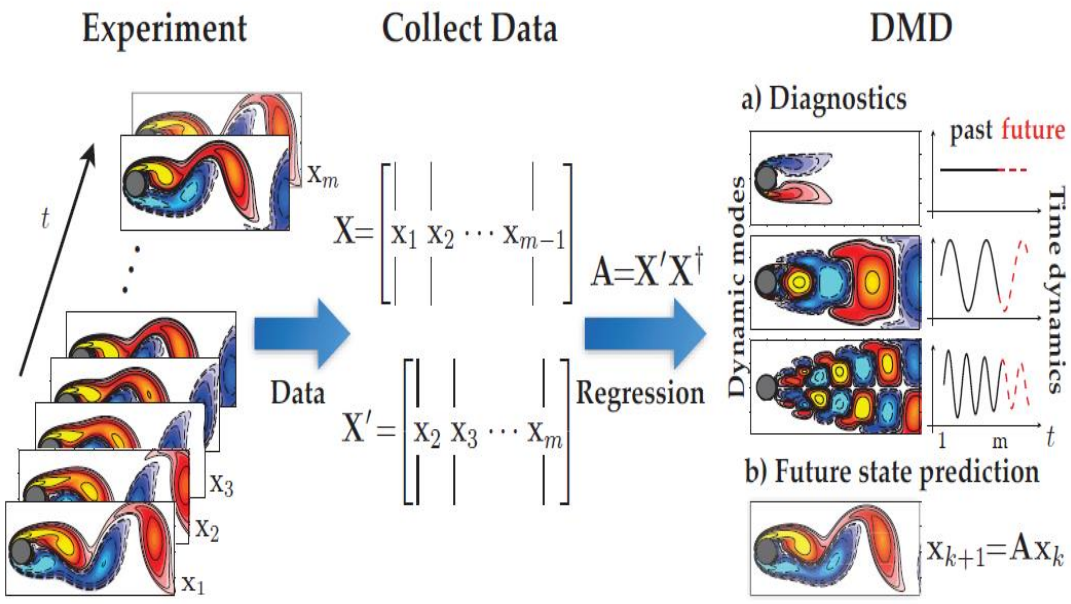
III) **Control :** The ultimate, and most difficult, purpose of the DMD algorithm is to enable feasible and robust control schemes directly from data sampling. Given that we're using a linear dynamical model to forecast the future of a nonlinear dynamical system, it's logical to assume that the two models will only agree for a limited period of time in the future. The hope is that this accurate prediction window will last long enough to allow a control decision to influence the system's future state. In this scenario, the DMD method allows for a completely data-driven approach to control theory, resulting in a compelling mathematical framework for controlling complicated dynamical systems with unknown or difficult-to-model governing equations.

Defining DMD :

The DMD approach decomposes data into a series of dynamic modes based on snapshots or observations of a system across time. The Arnoldi method, one of the workhorses of fast computational solvers, is closely related to the mathematics underpinning the extraction of dynamic information from time-resolved snapshots. There are two variables in the data collection process:

n = number of spatial points saved per time snapshot,

m = number of snapshots taken.



Suppose we have a dynamical system and two sets of data,

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix},$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}'_1 & \mathbf{x}'_2 & \cdots & \mathbf{x}'_{m-1} \\ | & | & & | \end{bmatrix},$$

so that $x'_k = F(x_k)$, where F is the map in for time Δt. The leading eigendecomposition of the best-fit linear operator A related the data $X' \approx AX$:

$$A = X'X^{\dagger}$$

The eigenvectors of A are the DMD modes, also known as dynamic modes, and each DMD mode corresponds to a specific eigenvalue of A.

The DMD Algorithm :

In practise, when the state dimension n is big, analysing the matrix A directly can be difficult. Instead, DMD considers a rank-reduced representation in terms of a POD-projected matrix $\widetilde{A}$ to avoid A's eigendecomposition. The DMD algorithm works like this:

1) Take Singular value Decomposition (SVD) of X

$$X \approx U\Sigma V^{*}$$

Where * denotes the conjugate transpose, $U \in C^{n\times r}, \Sigma \in C^{r\times r}, \text{and } V \in C^{m\times r}$. Here r is the rank of the reduced SVD approximation to X. The left singular vectors U are POD modes. The columns of U are orthonormal, so U*U = I; similarly, V*V = I.
At this point in the method, the SVD reduction is used to accomplish a low-rank truncation of the data. If there is low-dimensional structure in the data, the singular values of Σ will drop abruptly to zero, with probably only a few dominating modes remaining.

2) The matrix A from may be obtained by using the pseudoinverse of X obtained via the SVD:

$$A = X'V\Sigma^{-1}U^{*}$$

In practise, computing $\widetilde{A}$, the r x r projection of the complete matrix A onto POD modes, is more efficient computationally:

$$\widetilde{A} = U^{*}AU = U^{*}X'V\Sigma^{-1}$$

On POD coordinates, the matrix $\widetilde{A}$, defines a low-dimensional linear description of the dynamical system:

$$\tilde{x}_{k+1} = \widetilde{A}\tilde{x}_{k}$$

It is possible to reconstruct the high-dimensional state $x_k = U\tilde{x}_k$.

3) Compute the eigendecomposition of $\widetilde{A}$:

$$\widetilde{A}W = W\Lambda$$

where columns of W are eigenvectors and Λ is a diagonal matrix containing the corresponding eigenvalues λk .

4) Finally, using W and Λ, we can rebuild A's eigendecomposition. The eigenvalues of A are determined by Λ, while the eigenvectors of A (DMD modes) are determined by columns of φ.

The DMD framework has a number of advantages, one of which is the ease with which it may be expressed in terms of well-known linear algebra techniques. DMD may now be extended to include multiresolution, actuation, and control, time-delay coordinates and probabilistic formulations, noise mitigation, and sparse measurements using compressed sensing. Another significant benefit of DMD is that it is totally based on measurement data. As a result, it may be used in a variety of fields, such as fluid dynamics, video processing, epidemiology, neuroscience, and finance.

DMD in fluid dynamics

1) Snapshot based method :

It is typically helpful to employ snapshot-based methods when studying a time series of data comprising either velocity or vorticity fields at a grid of geographical places. To begin, two-dimensional or three-dimensional vector field data is flattened into a single tall column vector at time t_k:

$$\mathbf{x}(\mathbf{r}, t_k) = \begin{bmatrix} x(r_{1,1}, t_k) & x(r_{1,2}, t_k) & \cdots & x(r_{1,p}, t_k) \\ x(r_{2,1}, t_k) & x(r_{2,2}, t_k) & \cdots & x(r_{2,p}, t_k) \\ \vdots & \vdots & \ddots & \vdots \\ x(r_{q,1}, t_k) & x(r_{q,2}, t_k) & \cdots & x(r_{q,p}, t_k) \end{bmatrix}$$

$$\Longrightarrow \quad \mathbf{x}_k = \begin{bmatrix} x(r_{1,1}, t_k) \\ x(r_{1,2}, t_k) \\ \vdots \\ x(r_{2,1}, t_k) \\ \vdots \\ x(r_{q,p}, t_k) \end{bmatrix}$$

x specifies the flow variable of interest, r is the spatial coordinate, and the index k denotes the kth time step in this two-dimensional vector field example. A snapshot of data is represented by the vector$x_k \in R^n$. This not only removes the spatial relationship between neighbours, but it also permits vector inner products to be used to compare photos taken at various periods. The size n of a snapshot is determined by the vector field's original size as well as the number

of flow variables of interest, which can easily number in the hundreds of thousands, if not millions or billions. These snapshots can be combined to form the data matrix X:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}$$

For snapshot-based approaches, this data matrix is a frequent starting point. The state dimension n in fluid systems is often significantly bigger than the number of snapshots m, resulting in a tall and narrow matrix X. In contrast to the statistics literature, where a collection of time data is normally arranged as row vectors in a short and fat matrix, spatial measurements are collected as column vectors.

2) Proper Orthogonal Decomposition (POD) :

POD is a common method for reducing dimensionality in fluids, resulting in a hierarchical decomposition of flow data into a set of spatially associated modes. POD is frequently expressed as the SVD of the data matrix X:

$$X = U\sum V'$$

Because the state dimension n is substantially bigger than the number of snapshots m, the rank of X can only be m, implying that there can only be m x m nonzero blocks of singular values Σ.

The SVD method of decomposing a data matrix is both strong and numerically stable. For high-dimensional data related with fluid velocity or vorticity fields, the method of snapshots is commonly used to compute the leading terms in the SVD. In particular, a m x m column wise correlation matrix X'X consisting of inner products of the columns of X can be constructed. This matrix's eigen decomposition is related to the SVD:

$$X'X = V\sum U'U\sum V' = V(\Sigma)^2V'$$

$$X'XV = V(\Sigma)^2$$

It is possible to create the leading m columns of U (i.e., POD modes) by computing V and E from the spectral decomposition of X'X:

$$U = XV\Sigma^{-1}$$

Before computing POD modes, it is common practise to subtract the mean of X (i.e., the mean velocity or vorticity field), in which case POD is similar to PCA from statistics. It's worth noting that Sirovich's method of snapshots was published the same year as Sirovich and Kirby's breakthrough on eigenfaces. Face recognition has subsequently become a core machine-learning task, thanks to this paper's strategy for extracting dominating patterns in human faces from data.

The matrices U and V are used by POD to disentangle the space and temporal correlations. The geographical correlations in the data are only present in matrix U, but all temporal information is found in matrix V. Many approaches, such as Galerkin projection, ignore the information from Σ and V that remains. In fact, if only U is required, POD can be computed using non-time-resolved data X, with the V matrix being mostly irrelevant. DMD uses the temporal correlations in V to rearrange the leading column of U for time-resolved data, resulting in dominant patterns in the POD subspace that are coherent in both space and time.

3) Galerkin projection of Navier – Stokes equation :

Galerkin projection can be used to obtain a ROM of a dynamical system given an orthogonal basis from POD. The state vector x is approximated by a finite sum of POD modes in this manner, and this expansion is simply replaced into the dynamical system. It is possible to create an induced dynamical system on the POD mode coefficients using the orthogonality of the POD coordinate system. When compared to the state-space size n, the resulting dynamical system often has a substantially smaller dimension r.

Consider the incompressible Navier–Stokes equations, in which q(ξ, t) denotes a continuous vector field of the fluid's velocity components in space and time:

$$\frac{\partial \mathrm{q}}{\partial \mathrm{t}} + (\mathrm{q}.\nabla)\mathrm{q} = -\mathrm{p} + \frac{1}{\mathrm{Re}}\nabla^2\mathrm{q}$$

$$\nabla.\,\mathrm{q} = 0$$

In this nondimensionalized formulation, the only parameter is the Reynolds number Re = LU/ν, where L is a length scale, U is a velocity scale, and ν is the kinematic fluid viscosity.

Written in coordinates, the Navier–Stokes equations become

$$\frac{\partial \mathrm{q_j}}{\partial \mathrm{t}} + \sum \mathrm{q_i}\frac{\partial \mathrm{q_j}}{\partial \xi_\mathrm{i}} = -\frac{\partial \mathrm{p}}{\partial \xi_\mathrm{j}} + \frac{1}{\mathrm{Re}}\sum_\mathrm{i}\frac{\partial^2 \mathrm{q_j}}{\partial \xi_\mathrm{i}^2}$$

$$\sum_{j} q_i \frac{\partial q_j}{\partial \xi_i} = 0$$

This can be expressed as a dynamical system in terms of the continuous variable q's high-dimensional discretization x:

$$\dot{x} = Lx + Q(x, x)$$

where L represents a linear operator and Q represents a bilinear operator. The convection term $(q \cdot \nabla)q$ relates to quadratic nonlinearity.

The state x near an equilibrium $\bar{x}$ may be approximated by expanding the velocity field as a sum of $\bar{x}$ and the POD modes $\{\psi_i\}_{i=1}^{r}$ (ψ are the columns of U)

$$x \approx \bar{x} + \sum_{i=1}^{r} a_i \psi_i$$

Substituting the POD expansion we get :

$$\begin{aligned}
\dot{a}_k \psi_k &= \mathbf{L}(\bar{\mathbf{x}} + a_i \psi_i) + \mathbf{Q}\left(\bar{\mathbf{x}} + a_i \psi_i, \bar{\mathbf{x}} + a_j \psi_j\right) \\
&= \mathbf{L}\bar{\mathbf{x}} + a_i \mathbf{L}\psi_i + \mathbf{Q}(\bar{\mathbf{x}}, \bar{\mathbf{x}}) + a_j \mathbf{Q}(\bar{\mathbf{x}}, \psi_j) + a_i \mathbf{Q}(\psi_i, \bar{\mathbf{x}}) + a_i a_j \mathbf{Q}(\psi_i, \psi_j) \\
&= \underbrace{\mathbf{L}\bar{\mathbf{x}} + \mathbf{Q}(\bar{\mathbf{x}}, \bar{\mathbf{x}})}_{=0} + a_i \underbrace{[\mathbf{L}\psi_i + \mathbf{Q}(\bar{\mathbf{x}}, \psi_i) + \mathbf{Q}(\psi_i, \bar{\mathbf{x}})]}_{=\tilde{\mathbf{L}}\psi_i} + a_i a_j \mathbf{Q}(\psi_i, \psi_j),
\end{aligned}$$

where summation over the indices i and j is assumed.

Finally, because the modes ψi are orthonormal, we take the inner product of both sides with ψk :

$$\begin{aligned}
\dot{a}_k &= a_i \langle \tilde{\mathbf{L}}\psi_i, \psi_k \rangle + a_i a_j \langle \mathbf{Q}(\psi_i, \psi_j), \psi_k \rangle \\
&= \hat{\mathbf{L}}_{ik} a_i + \hat{\mathbf{Q}}_{ijk} a_i a_j,
\end{aligned}$$

where $\hat{L}_{i\,j}$ and $\hat{Q}_{i\,j\,k}$ are new linear and bilinear operators on mode coefficients. The pressure term in disappears in this POD-Galerkin procedure because each of the POD modes satisfies incompressibility, and so the inner product of the pressure gradient with ψ_k vanishes.

A ROM for a nonlinear dynamical system is generated by Galerkin projection onto POD modes. In a mean-field approximation, these systems can be adapted to incorporate both an equilibrium and a mean flow. Despite its advantages, the traditional POD-Galerkin technique has a number of drawbacks:

- Because the operators $\hat{L}$ and $\hat{Q}$ are dense, even for a small dimension r, simulating these Galerkin systems may be computationally expensive. While a fluid state may be high dimensional, the corresponding equations are often sparse, allowing for quick numerical systems like spectral methods.
- POD-Galerkin systems are frequently unstable, and studying them is tough.

Application of DMD in fluids:

DMD has become a frequently utilised approach in fluid dynamics since its introduction to the fluid dynamics community and subsequent connection to nonlinear dynamical systems. Many later investigations in fluid dynamics have been framed by DMD's original presentation as a generalisation of global stability analysis.

DMD to extract structure from fluids data:

DMD has been used to investigate mixing, acoustics, and combustion in a range of flow geometries (jets, cavity flow, wakes, channel flow, boundary layers, and so on).
DMD has been utilised to investigate a variety of cavity flow issues. DMD has been utilised in particular to investigate self-sustained oscillations caused by unstable boundary layer separation at the leading edge. DMD has also been calculated using time-resolved PIV measurements of an incompressible cavity.
DMD has also been used to investigate wake flows. A finite-thickness flat plate with an elliptic leading edge and active blowing/suction was used to examine flow separation in an early experiment. DMD was utilised in particular to find a mode associated with the frequency lock-on of a high-frequency separated shear layer instability linked to the separation bubble and low-frequency wake modes. A gurney flap's wake has also been explored.
DMD has also been used to study boundary layer flows, revealing rough walls, compressible turbulent boundary layer interaction with a fluttering glass, and transitional boundary layer near-wall structures. The flow of turbulent channel flows has also been studied.
DMD has also been used to study more unusual flows, such as a simulated high-speed train. Shock turbulence boundary layer interaction (STBLI) has also been studied, with DMD being used to detect a pulsating separation bubble accompanied by shockwave motion. Self-excited oscillations in detonation waves have also been studied using DMD. Identifying hairpin vortices, deconstructing flow past a surface-mounted cube, modelling shallow-water

equations, researching nanofluids through a square cylinder, and determining the growth rate of instabilities in annular liquid sheets are some of the other challenges.

Example: Re = 100 flow around cylinder wake

We show DMD on a dataset that represents a time series of fluid vorticity fields for the wake behind a circular cylinder at Reynolds number Re = 100 in this example. $\mathrm{Re} = \frac{\mathrm{DU}_\infty}{\mathrm{v}}$, where D is the cylinder diameter, U is the free-stream velocity, and v is the kinematic fluid viscosity, measures the ratio of inertial to viscous forces. Because Re = 100 is greater than the critical Reynolds number $\mathrm{Re}_{\mathrm{crit}} \approx 47$, the flow encounters a supercritical Hopf bifurcation, resulting in laminar vortex shedding. The three-dimensional flow is represented by this limit cycle, which is stable.

An immersed boundary projection method (IBPM) fluid solver2 based on Taira and Colonius' rapid multidomain approach is used to simulate the two-dimensional Navier–Stokes equations. The computational domain is divided into four grids, the finest of which covers a domain of 9x4 and the largest of which covers a domain of 72x32, with lengths nondimensionalized by cylinder diameter; each grid includes 450x200 points, resulting in 50 points per cylinder diameter. We utilise a Δt = 0.02 time step, which is small enough to satisfy the CFL requirement.

1) Snapshots of Data:

The figure below shows a snapshot of the cylinder wake data. On the left, vorticity contours are illustrated, and on the right, a time history of vorticity probe sensors s1 and s2. We gather $m = 150$ images at regular intervals in time, $10\Delta t$, sampling five episodes of vortex shedding after simulations converge to steady-state vortex shedding.

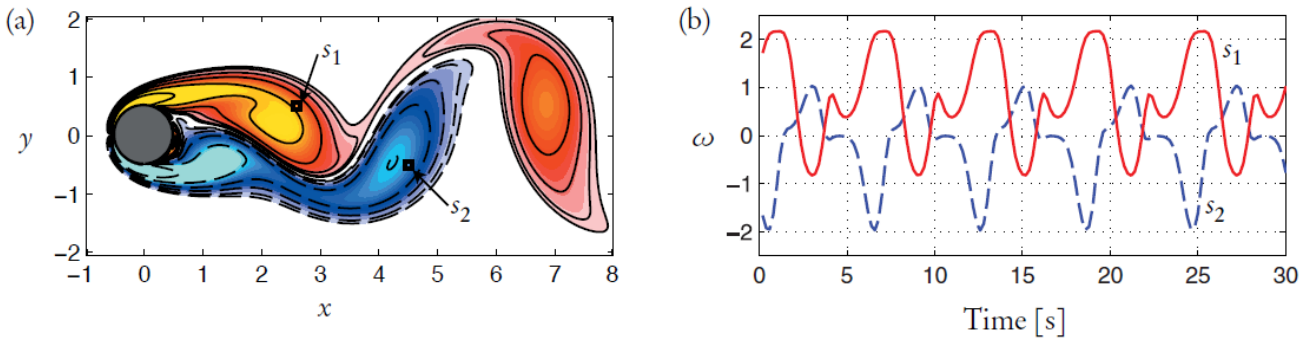


Fig. Example of vorticity field (a) and two sensors (b) for the wake behind a cylinder at Re = 100.

The duration of vortex shedding is about 300Δt, with Δt = 0.02, corresponding to a Strouhal number of about $\mathrm{St} = \frac{\mathrm{fA}}{\mathrm{U}_\infty} = 0.16$, which matches experimental results.

Each vorticity field snapshot from the best computational domain is reshaped into a big vector x_k, which is made up of columns from the matrices $\mathrm{X}_1^{\mathrm{m}-1}$ and $\mathrm{X}_2^{\mathrm{m}}$. To obtain symmetric solutions, it is necessary to sample a large number of periods. It's also possible to add negative vorticity fields to the snapshot matrix, which enforces symmetry.

2) POD modes for wake cylinder:

The POD modes in this example occur in energetic pairs, depicting the singular values. Despite the fact that the SVD is a separation of variables that results in a spatiotemporal decomposition, it can approximate the relevant structures for the travelling wave solution to the cylinder wake. POD can be applied to data from either a fluid velocity field or a vorticity field; in this case, we're using vorticity fields.

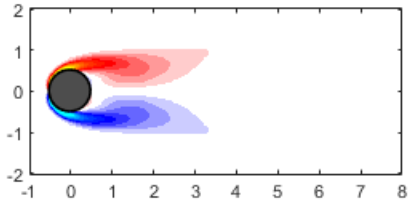
3) DMD of the wake cylinder:

DMD uses the same fundamental snapshot information as POD to apply to the cylinder wake, making it a data-driven method that works equally well with data from simulations or experiments.

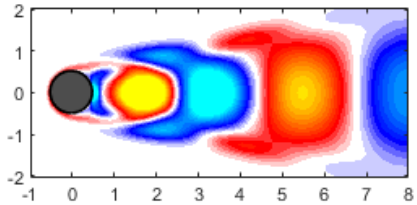
Unlike POD, DMD not only gives modes, but also a set of associated eigenvalues that determine a low-dimensional dynamical system for how the mode amplitudes evolve over time. Furthermore, unlike POD, the DMD computation does not deduct the mean, hence the first mode (represented as mode 1 in the middle panel) corresponds to a non-changing backdrop mode (i.e., it has zero eigenvalue). Because the POD gives a harmonic decomposition, the modal amplitudes are essentially sinusoidal in time at harmonic frequencies of the dominating vortex shedding, the DMD modes appear comparable to the POD modes in this example.

Results :

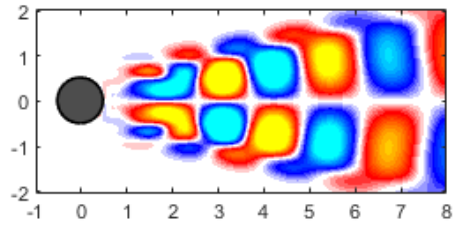
a) **Initial Stage**



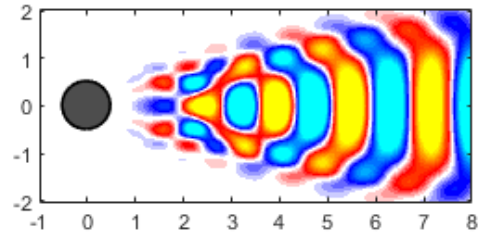
b)



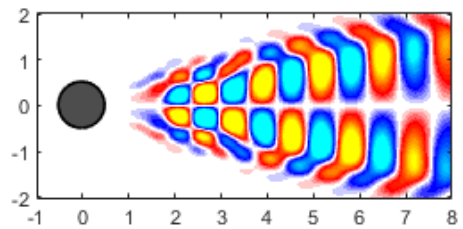
c)



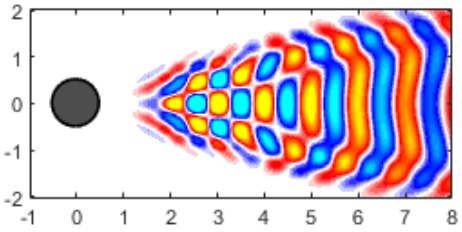
d)



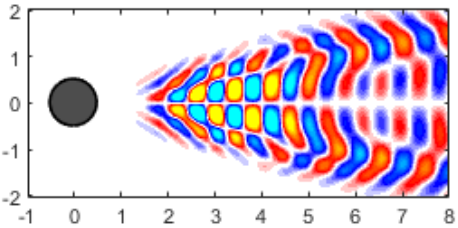
e)



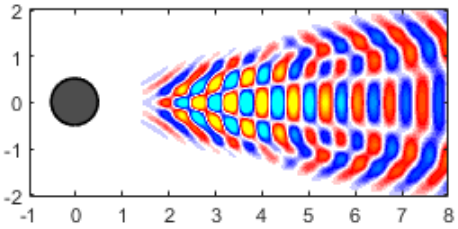
f)



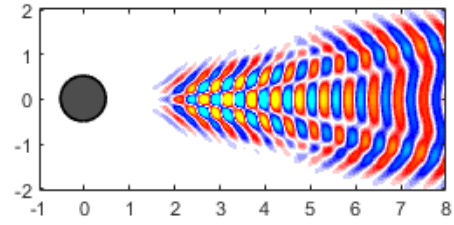
g)



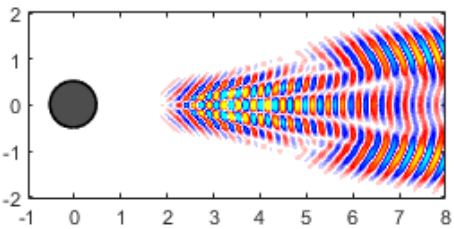
h)



i)



j) **Final Stage *:***



Appendix :

```
load('CYLINDER_ALL.mat')
X=VORTALL(:,1:end-1);
Y=VORTALL(:,2:end);
[U S V]=svd(X,'econ')
semilogy(diag(S),'linewidth',4)
xlim([0 150])
ylim([0 10000])

r=100;

U=U(:,1:r);

S=S(1:r,1:r);

V=V(:,1:r);
A=U'*Y*V*inv(S)
[w eigs]=eig(A)
P=Y*V*inv(S)*w

for i=1:20
plotCylinder(reshape(real (P(:,i)),nx,ny),nx,ny);
plotCylinder(reshape(imag (P(:,i)),nx,ny),nx,ny);
end

function f2 = plotCylinder(VORT,ny,nx)
f2 = figure
vortmin = -5;  % only plot what is in -5 to 5 range
vortmax = 5;
VORT(VORT>vortmax) = vortmax;  % cutoff at vortmax
VORT(VORT<vortmin) = vortmin;  % cutoff at vortmin
imagesc(VORT); % plot vorticity field
load CCcool.mat
colormap(CC);  % use custom colormap
set(gca,'XTick',[1 50 100 150 200 250 300 350 400 449],'XTickLabel',{'-
1','0','1','2','3','4','5','6','7','8'})
set(gca,'YTick',[1 50 100 150 199],'YTickLabel',{'2','1','0','-1','-2'});
set(gcf,'Position',[100 100 300 130])
axis equal
hold on
contour(VORT,[-5.5:.5:-.5 -.25 -.125],':k','LineWidth',1.2)
contour(VORT,[.125 .25 .5:.5:5.5],'-k','LineWidth',1.2)
theta = (1:100)/100'*2*pi;
x = 49+25*sin(theta);
y = 99+25*cos(theta);
fill(x,y,[.3 .3 .3])  % place cylinder
plot(x,y,'k','LineWidth',1.2) % cylinder boundary
set(gcf,'PaperPositionMode','auto')
end
```