NSF CyberTraining Winter Workshop

Introduction to Proper Orthogonal Decomposition

concepts, formulation and applications

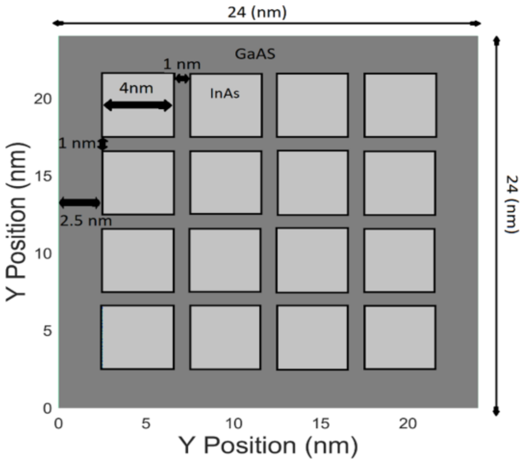
Part III: POD - Examples & Method of Snapshots

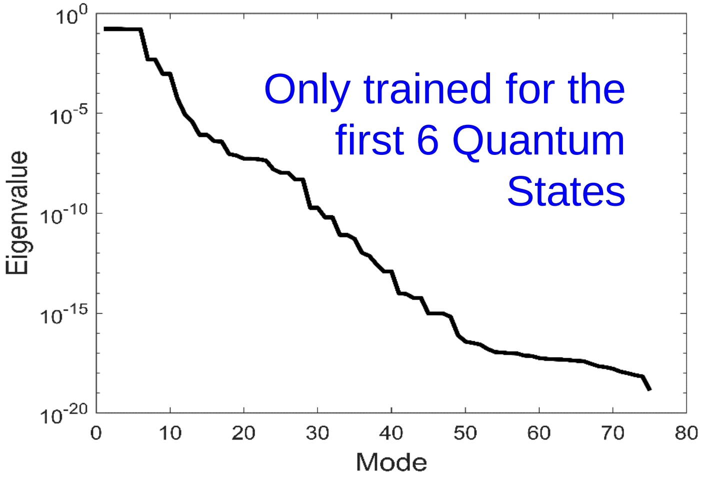
Ming-Cheng Cheng

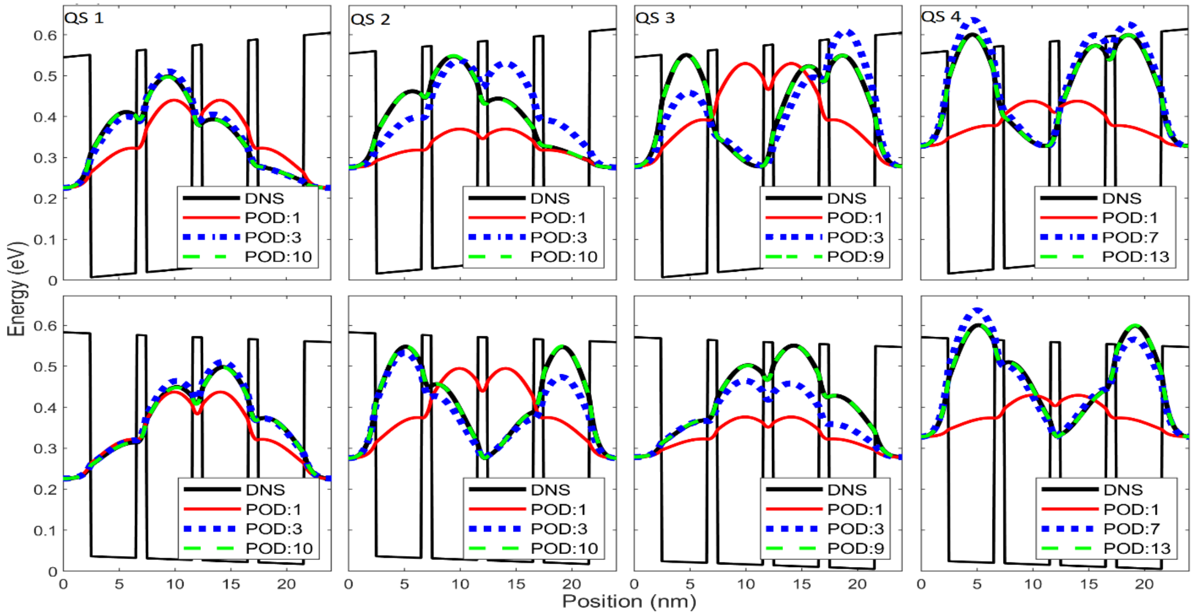
Department of Electrical & Computer Engineering
Clarkson University, Potsdam, NY 13699

Supported by NSF OAC-2118079

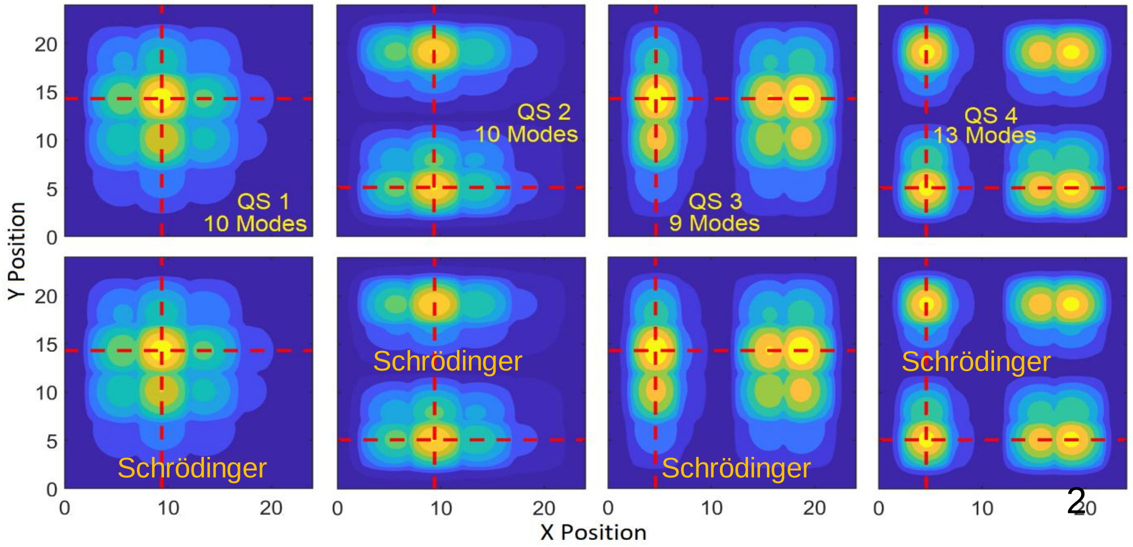
2D Quantum Dot Structures







One-mode POD model predicts the average of the collected data

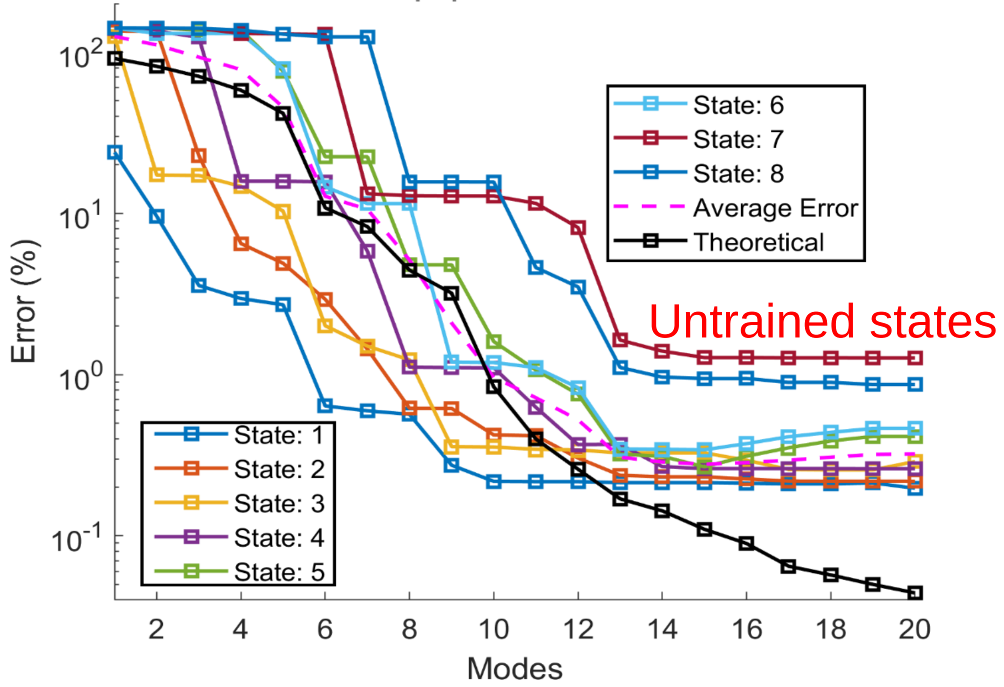


- **Training:** Apply electric fields in *x* and *y* directions **separately** to collect N_S sets of WF data
- **Simulation:** Demonstrated at electric field in a arbitrary direction

2D Quantum Dot Structures

Quantum State	POD Energy (eV)	DNS Energy (eV)	Difference (%)
1	0.226511	0.22602	0.21716
2	0.276652	0.275789	0.312542
3	0.27938	0.278522	0.30753
4	0.32952	0.32831	0.36803
5	0.357149	0.355488	0.466291
6	0.359669	0.358011	0.461969
7	0.409963	0.407977	0.48561
8	0.410194	0.40821	0.4847

$$Err_M = \sqrt{\int_\Omega \left[\psi_{P,M}(\vec{r}) - \psi(\vec{r})\right]^2 d\Omega} \bigg/ \sqrt{\int_\Omega \psi(\vec{r})^2 d\Omega}$$



Theoretical LS error

$$Err_{LS} = \sqrt{\sum_{i=M+1}^{N_S} \lambda_i \bigg/ \sum_{i=1}^{N_S} \lambda_i}$$

Average LS errors over all QSs

$$Err_{M,ls,av} = \sqrt{\sum_{i=1}^{N_{QS}} \int_\Omega \left[\psi_{M,P,n} - \psi_n\right]^2 d\Omega \bigg/ N_{QS}}$$

Approach		6 Modes	8 Modes	10 Modes	12 Modes	14 Modes
POD time (s)	Eigenvalue solver	0.0000543	0.0000586	0.0000609	0.0000729	0.0000748
	Post Processing	0.0102472	0.0139769	0.0166791	0.0188029	0.0233854
DNS Eigenvalue Solver time (s)		2.1544969				

For a 2D quantum-dot structure, if 10 modes are used, a computational speedup near 130 times can be achieved.

States 7 & 8 were NOT trained

The computation was performed in Matlab on an *i7* laptop PC.

Solution of Eigenvalues and Eigenfunctions of 2-point Correlation Data

$$\int_{\Omega'} \langle Q(\vec{r},t) \otimes Q(\vec{r}\,',t)\rangle \eta(\vec{r}\,') d\Omega' = \lambda \eta(\vec{r}), \qquad (1)$$

$$\langle Q(\vec{r},t) \otimes Q(\vec{r}\,',t)\rangle = \frac{1}{N_s}\sum_{j=1}^{N_s} Q(\vec{r},t_j)Q(\vec{r}\,',t_j). \qquad (2)$$

The N_r x N_r eigenvalue problem for a large-scale multi-dimensional structure may be enormously large and numerically prohibitive.

The Method of Snapshots: transform the eigenvalue problem from a N_r x N_r space domain to an N_s x N_s sampling domain. $N_r >> N_S$

- For a 3D problem, N_r may be on the order of several 100,000’s.
- For a dynamic problem, N_s is on the order of several 100’s – several 1,000’s; for steady problems, 10 -1,000

- Start with the discrete eigenvalue problem. Insert Eq. (2) into (1)

$$\frac{1}{N_s}\sum_{j=1}^{N_s} Q(\vec{r},t_j) \int_{\Omega'} Q(\vec{r}\,',t_j)\eta(\vec{r}\,') d\Omega' = \lambda \eta(\vec{r}),$$

Solution of Eigenvalues and Eigenfunctions of 2-point Correlation Data

$$\frac{1}{N_t}\sum_{j=1}^{N_s} Q(\vec{r},t_j)\int_{\Omega'} Q(\vec{r}',t_j)\eta(\vec{r}')d\Omega' = \lambda\eta(\vec{r}),$$

- Define the projection of the *j*th sampled data set onto the POD space as

$$u(t_j) = \int_{\Omega'} Q(\vec{r}',t_j)\eta(\vec{r}')d\Omega',$$

The eigenvalue problem becomes

$$\frac{1}{N_s}\sum_{j=1}^{N_s} Q(\vec{r},t_j)\,u(t_j) = \lambda\eta(\vec{r})$$

- Multiply both sides of by $Q(\vec{r},t_i)$ and perform an integral on each side,

$$\frac{1}{N_s}\sum_{j=1}^{N_s}\left[\int_{\Omega} Q(\vec{r},t_i)\,Q(\vec{r},t_j)d\Omega\right]u(t_j) = \lambda\int_{\Omega} Q(\vec{r},t_i)\eta(\vec{r})d\Omega$$

Define $A_{ij} = \frac{1}{N_s}\int_{\Omega} Q(\vec{r},t_i)\,Q(\vec{r},t_j)d\Omega$ (2 point correlation in time) ➔ $\mathbf{A}\vec{u} = \lambda\vec{u}$

Solution of Eigenvalues and Eigenfunctions of 2-point Correlation Data

- **Eigenvalue Problem in the sampling domain**

$$\mathbf{A}\vec{u} = \lambda\vec{u}; \quad or \quad \begin{bmatrix} A_{11} & \cdots & A_{1j} & \cdots & A_{1N_S} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ A_{i1} & \cdots & A_{ij} & \cdots & A_{iN_S} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ A_{N_t 1} & \cdots & A_{N_t j} & \cdots & A_{N_t N_S} \end{bmatrix} \begin{bmatrix} u(t_1) \\ \vdots \\ u(t_j) \\ \vdots \\ u(t_{N_S}) \end{bmatrix} = \lambda \begin{bmatrix} u(t_1) \\ \vdots \\ u(t_j) \\ \vdots \\ u(t_{N_S}) \end{bmatrix}$$

Once the eigenvectors, $[\vec{u}_1, \vec{u}_2, \vec{u}_3, \ldots. \vec{u}_{Ns}]$ are determined, each POD mode is obtained from a linear combination of numerical observations,

$$\eta_i(\vec{r}) = \frac{1}{\lambda N_s} \sum_{j=1}^{N_s} Q(\vec{r}, t_j) u_i(t_j)$$

The first N_s POD eigenvalues are identical to those derived from the N_s snapshots.

Concept Demonstration

Generation of Eigenvalues and POD modes

Use a dynamic function of $f(x,t)$ to train the POD modes. The function is described by 4 distinct characteristics,

$$f(x, t_i) = \sum_{j=1}^{4} f_j(x, t_i)\,,$$

- Space: $x = [x_1, x_2, \ldots x_k, \ldots, x_{Nx-1}, x_{Nx}]^T = [0, 2, 4, \ldots, 198]^T$ with $N_x = 100$ and grid size $\Delta x = 2$
- Time: $\Delta t = 0.2$ with the number of samples $N_s = 20$
- $f(x,t)$ includes a **Gaussian** function, **2 sinusoidal** functions with different frequencies in space and one **polynomial**. All these functions evolve in time described by different dynamic evolution rates either increasing or decreasing in time.

```
Nx = 100;
Ns = 20;
dx = 2;
xx = linspace(0,Nx*2-dx,Nx)';
tt = 0.2*linspace(0,Ns-1,Ns)';
```

- $f_1(x,t) = fgaussian = 500e^{\left(-\left(\frac{x-100}{30}\right)^2\right)}e^{-t}$
- $f_2(x,t) = fcos = 200\left[\cos\left(\frac{2\pi x}{x_{max}-20\Delta x}\right)+1\right]\left[1-e^{\left(-\frac{t^2}{2}\right)}\right]$
- $f_3(x,t) = fsin = 200\left[\sin\left(\frac{6\pi x}{x_{max}-20\Delta x}\right)+1\right]e^{\left(-\frac{t^2}{5}\right)}$
- $f_4(x,t) = fpoly = (0.3x^2 - 100x + 200)\,[1-e^{-t}]$

```
% 4 distinct functions
fgau = 500*exp(-((xx(1:Nx)-100)/30).^2); % decreasing in time
fcos= 200*(cos(2*pi/(xx(Nx)-20*dx)*xx(:))+1); % increasing in time, stating from 0
fsin= 200*(sin(6*pi/(xx(Nx)-20*dx)*xx(:))+1);% decreasing in time
fpoly = 0.3*xx(:).^2 - 100*xx(:) + 200; % increasing in time, stating from 0

%% Total function with dynamic evolution
ff_t = zeros(Nx,Ns);
for i=1:Ns
    ff_t(:,i) = fgau(:).*exp(-tt(i)) + fcos(:).*(1-exp(-tt(i)^2/2)) + ...
    fsin(:).*exp(-tt(i)^2/5) + fpoly(:).*(1-exp(-tt(i)));
end
```

Construct the 2-point correlation (N_x x N_x) matrix for $f(x,t)$

- Average of the 2-point (autocorrelation) correlation matrix for $f(x,t)$: $\mathbf{F} = \frac{1}{N_s}\sum_{j=1}^{N_s} f\left(x, t_j\right) \otimes f\left(x', t_j\right)$ (average over Ns samples in time)

$$f\left(x, t_j\right) \otimes f\left(x', t_j\right) = \begin{bmatrix} f\left(x_1, t_j\right) \\ f\left(x_2, t_j\right) \\ . \\ . \\ . \\ f\left(x_{Nx}, t_j\right) \end{bmatrix} \cdot \begin{bmatrix} f\left(x_1, t_j\right) \\ f\left(x_2, t_j\right) \\ . \\ . \\ . \\ f\left(x_{Nx}, t_j\right) \end{bmatrix}^T$$

```
% Average of the 2-point spatial correlation matrix over time
FF = zeros(Nx,Nx);
for i = 1:Ns
   FF = FF + ff_t(:,i)*ff_t(:,i)';
end
FF = FF/Ns;
```

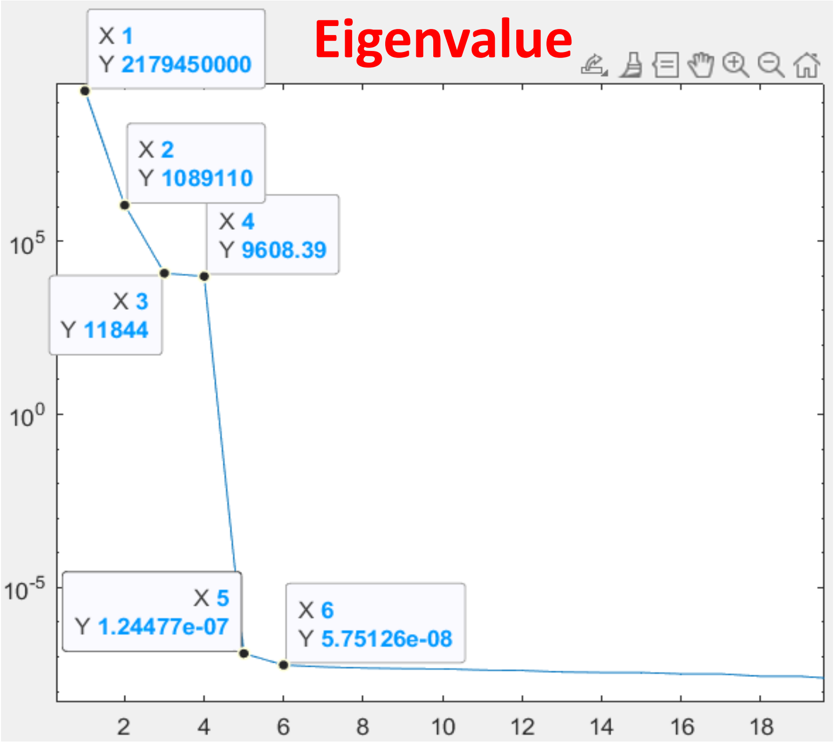
Solve the eigenvalues λ_i and the eigenfunctions (POD modes) η_i from f

Clarkson UNIVERSITY defy convention

The integral is not needed because of an equal spatial division is used

Arrange the eigenvalues in descending order and the POD modes with the same order

```
[V,DD] = eig(FF);
EigenV = flip(diag(DD));
V1 = flip(V,2);
semilogy(EigenV)
```



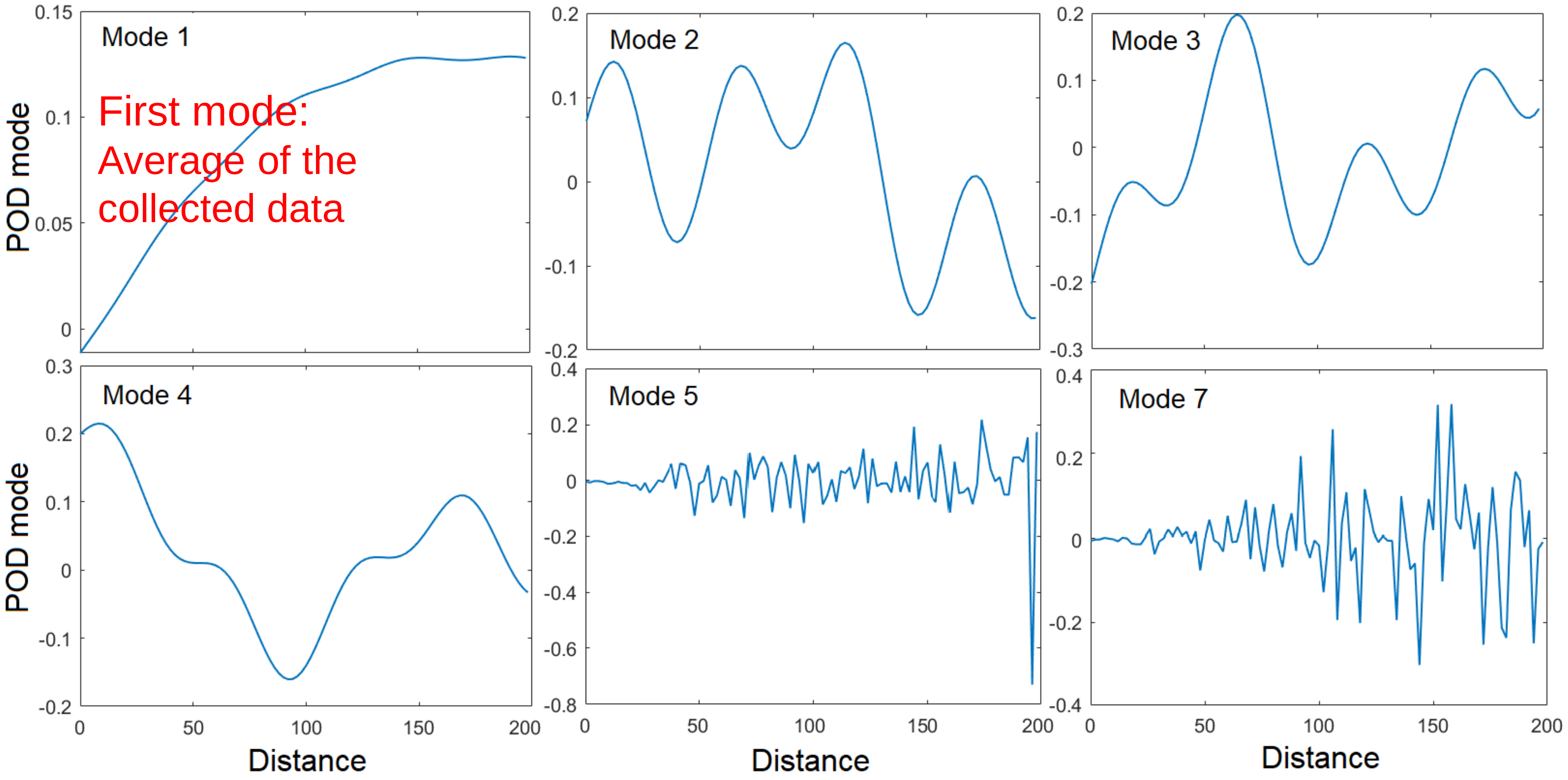
Eigenvalue Number

- The eigenvalue drops from the first mode to the 4th mode by 5 orders of magnitude and **becomes nearly zero beyond 4th mode.** Why?
- Note that $\frac{\lambda_1}{\lambda_5} \approx 10^{16}$ and λ becomes nearly flat because of the computer precision
- The decomposition effectively captures 4 distinct characteristics from the data with just 4 modes.
- That is, 4 modes are good enough to represent this dynamic problem.

NSF

POD Modes

Normalize the POD mode; i.e., make $\sum_{k=1}^{N_x} \left|\eta_j(x_k)\right|^2 = 1$ for $j = 1$ to N_x



Orthonormality

Verify the orthonormality

i.e., $\sum_{k=1}^{N_x} \eta_i(x_k)\eta_j(x_k) = \delta_{ij}$

```
% Verify the orthonormality
modeij_sum = zeros(10,10);
for i = 1:10
   for j = 1:10
      modeij_sum(i,j) = sum(V1(:,i).*V1(:,j));
   end
end
```

modeij_sum

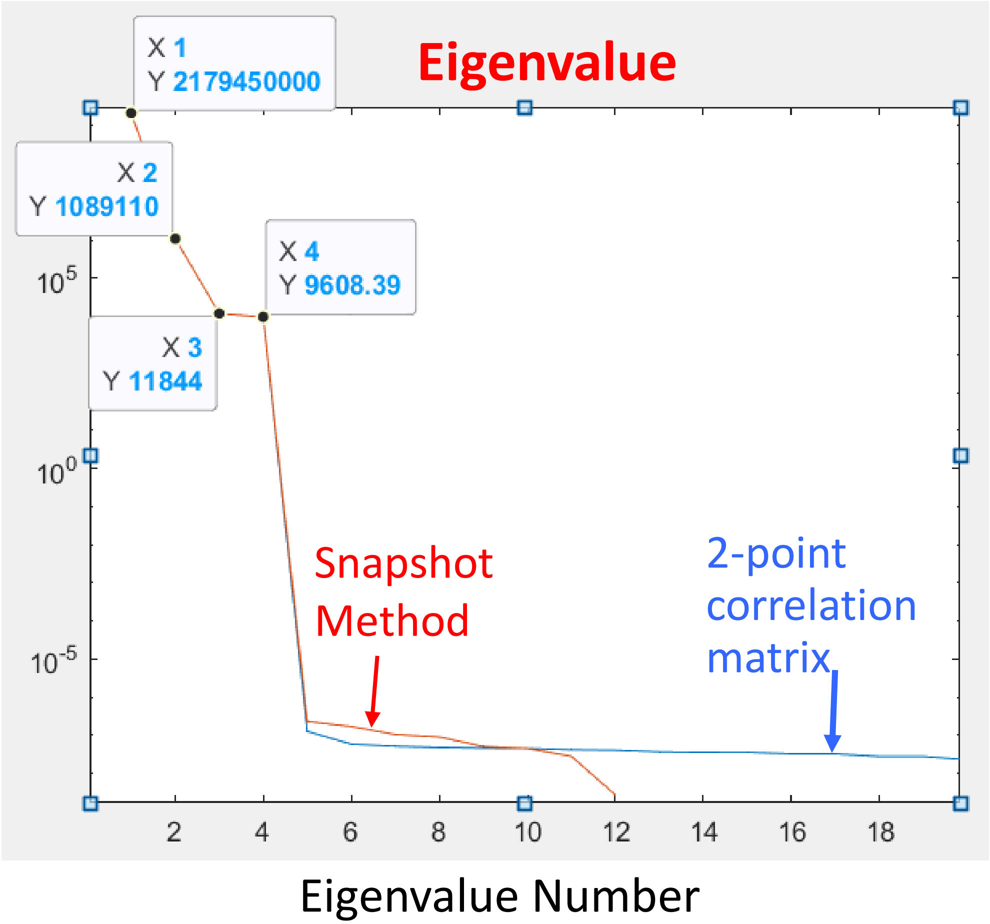
10x10 double

	1	2	3	4	5	6	7	8	9	10
1	1.0000	-6.9389e-17	-1.8215e-17	-1.0408e-16	1.3184e-16	1.2490e-16	1.0192e-17	9.5410e-18	-2.7756e-17	9.2157e-18
2	-6.9389e-17	1.0000	-2.4286e-17	-1.3791e-16	1.1796e-16	-1.9429e-16	-1.1948e-16	-5.2042e-18	1.2750e-16	-2.4937e-17
3	-1.8215e-17	-2.4286e-17	1.0000	-1.3791e-16	6.2450e-17	-1.3878e-17	1.7022e-17	3.1659e-17	5.6379e-18	-1.8513e-17
4	-1.0408e-16	-1.3791e-16	-1.3791e-16	1.0000	9.5410e-18	0	1.0029e-17	1.1926e-17	8.3917e-17	1.1506e-17
5	1.3184e-16	1.1796e-16	6.2450e-17	9.5410e-18	1.0000	2.2204e-16	8.2616e-17	8.7604e-17	-2.8623e-17	7.3726e-18
6	1.2490e-16	-1.9429e-16	-1.3878e-17	0	2.2204e-16	1	-2.9490e-17	-1.1102e-16	-4.8572e-17	-2.1142e-17
7	1.0192e-17	-1.1948e-16	1.7022e-17	1.0029e-17	8.2616e-17	-2.9490e-17	1.0000	-1.5938e-17	-1.8377e-17	-4.8654e-18
8	9.5410e-18	-5.2042e-18	3.1659e-17	1.1926e-17	8.7604e-17	-1.1102e-16	-1.5938e-17	1.0000	-3.0011e-16	1.0503e-17
9	-2.7756e-17	1.2750e-16	5.6379e-18	8.3917e-17	-2.8623e-17	-4.8572e-17	-1.8377e-17	-3.0011e-16	1.0000	-6.1925e-17
10	9.2157e-18	-2.4937e-17	-1.8513e-17	1.1506e-17	7.3726e-18	-2.1142e-17	-4.8654e-18	1.0503e-17	-6.1925e-17	1.0000

Method of Snapshots: $A_{ij} = \frac{1}{N_s}\int_\Omega Q(\vec{r},t_i)\,Q(\vec{r},t_j)\,d\Omega$

$$F_{snap} = \frac{1}{N_s}\sum_{i=1}^{Nx}\begin{bmatrix} f(x_i,t_1) \\ f(x_i,t_2) \\ \vdots \\ f(x_i,t_{N_s}) \end{bmatrix} \cdot \begin{bmatrix} f(x_i,t_1) \\ f(x_i,t_2) \\ \vdots \\ f(x_i,t_{N_s}) \end{bmatrix}^T$$

```
% Average of the snapshot matrix over space
F_snap = zeros(Ns,Ns);
for i = 1:Nx
   F_snap = F_snap + ff_t(i,:)'*ff_t(i,:);
end
F_snap = F_snap/Ns;
[Vsnap,Dsnap] = eig(F_snap);
EigenVsnap = flip(diag(Dsnap));
V1snap = flip(Vsnap,2);
semilogy(EigenVsnap);
```



The eigenvalues generated from the method of snapshots are identical to the first N_S eigenvalues calculated from the autocorrelation (2-point correlation) matrix.

Next. obtain POD Modes from

- 2-point correlation (autocorrelation) matrix
- Method of Snapshots

Verify that they are indeed identical for the first 4 modes.

Plot modes from these 2 approaches on top of each other for the first 6 modes

POD modes derived from the 2-point correlation matrix