AA216/CME345: MODEL REDUCTION

Proper Orthogonal Decomposition (POD)

Charbel Farhat
Stanford University
cfarhat@stanford.edu

Outline

$$
\begin{aligned}
\frac{d\mathbf{w}}{dt}(t) &= \mathbf{f}(\mathbf{w}(t), t) \\
\mathbf{y}(t) &= \mathbf{g}(\mathbf{w}(t), t) \\
\mathbf{w}(0) &= \mathbf{w}_0
\end{aligned}
$$

- $\mathbf{w} \in \mathbb{R}^N$: Vector of state variables
- $\mathbf{y} \in \mathbb{R}^q$: Vector of output variables (typically $q \ll N$)
- $\mathbf{f}(\cdot, \cdot) \in \mathbb{R}^N$: completes the specification of the high-dimensional system of equations

- Consider a fixed initial condition $\mathbf{w}_0 \in \mathbb{R}^N$
- Denote the associated state trajectory in the time-interval $[0, \mathcal{T}]$ by

$$\mathcal{T}_{\mathbf{w}} = \{\mathbf{w}(t)\}_{0 \leq t \leq \mathcal{T}}$$

- Consider a fixed initial condition $\mathbf{w}_0 \in \mathbb{R}^N$
- Denote the associated state trajectory in the time-interval $[0, \mathcal{T}]$ by

$$\mathcal{T}_{\mathbf{w}} = \{\mathbf{w}(t)\}_{0 \le t \le \mathcal{T}}$$

- The Proper Orthogonal Decomposition (POD) method seeks an orthogonal projector $\mathbf{\Pi}_{\mathbf{V},\mathbf{V}}$ of fixed rank k that minimizes the *integrated projection error*

$$\int_0^{\mathcal{T}} \|\mathbf{w}(t) - \mathbf{\Pi}_{\mathbf{V},\mathbf{V}}\mathbf{w}(t)\|_2^2 \, dt = \int_0^{\mathcal{T}} \|\mathcal{E}_{\mathbf{V}^\perp}(t)\|_2^2 dt = \|\mathcal{E}_{\mathbf{V}^\perp}\|^2 = J(\mathbf{\Pi}_{\mathbf{V},\mathbf{V}})$$

Theorem

Let $\widehat{\mathbf{K}} \in \mathbb{R}^{N \times N}$ be the real, symmetric, positive, semi-definite matrix defined as follows

$$\widehat{\mathbf{K}} = \int_0^T \mathbf{w}(t)\mathbf{w}(t)^T dt$$

Let $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_N \geq 0$ denote the ordered eigenvalues of $\widehat{\mathbf{K}}$ and $\widehat{\phi}_i \in \mathbb{R}^N, \; i = 1, \cdots, N$, denote their associated eigenvectors which are also referred to as the POD modes

$$\widehat{\mathbf{K}}\,\widehat{\phi}_i = \hat{\lambda}_i\,\widehat{\phi}_i, \;\; i = 1, \cdots, N$$

The subspace $\widehat{\mathcal{V}} = range(\widehat{\mathbf{V}})$ of dimension k that minimizes $J(\mathbf{\Pi}_{\mathbf{V},\mathbf{V}})$ is the invariant subspace of $\widehat{\mathbf{K}}$ associated with the eigenvalues $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_k$

- Solving the eigenvalue problem $\widehat{\mathbf{K}}\,\widehat{\phi}_i = \hat{\lambda}_i\,\widehat{\phi}_i$ is in general computationally intractable because: (1) The dimension N of the matrix $\widehat{\mathbf{K}}$ is usually large; and (2) this matrix is usually dense

- Solving the eigenvalue problem $\widehat{\mathbf{K}}\,\widehat{\phi}_i = \hat{\lambda}_i\,\widehat{\phi}_i$ is in general computationally intractable because: (1) The dimension N of the matrix $\widehat{\mathbf{K}}$ is usually large; and (2) this matrix is usually dense
- However, the state data is typically available under the form of discrete "snapshot" vectors

$$\{\mathbf{w}(t_i)\}_{i=1}^{N_{\text{snap}}}$$

- Solving the eigenvalue problem $\widehat{\mathbf{K}}\,\widehat{\phi}_i = \hat{\lambda}_i\,\widehat{\phi}_i$ is in general computationally intractable because: (1) The dimension N of the matrix $\widehat{\mathbf{K}}$ is usually large; and (2) this matrix is usually dense
- However, the state data is typically available under the form of discrete "snapshot" vectors

$$\{\mathbf{w}(t_i)\}_{i=1}^{N_{\text{snap}}}$$

- In this case, $\int_0^T \mathbf{w}(t)\mathbf{w}(t)^T\,dt$ can be approximated using a *quadrature rule* as follows

$$\mathbf{K} = \sum_{i=1}^{N_{\text{snap}}} \alpha_i\,\mathbf{w}(t_i)\mathbf{w}(t_i)^T$$

where $\alpha_i,\ i = 1, \cdots, N_{\text{snap}}$ are the quadrature weights

- Let $\mathbf{S} \in \mathbb{R}^{N \times N_{\text{snap}}}$ denote the snapshot matrix defined as follows

$$\mathbf{S} = \left[\sqrt{\alpha_1}\mathbf{w}(t_1) \quad \dots \quad \sqrt{\alpha_{N_{\text{snap}}}}\mathbf{w}(t_{N_{\text{snap}}})\right]$$

- It follows that

$$\boxed{\mathbf{K} = \mathbf{S}\mathbf{S}^T}$$

where $\mathbf{K}$ is still a large-scale $(N \times N)$ matrix

- Note that the *non-zero* eigenvalues of the matrix $\mathbf{K} = \mathbf{S}\mathbf{S}^T \in \mathbb{R}^{N \times N}$ are the same as those of the matrix $\mathbf{R} = \mathbf{S}^T\mathbf{S} \in \mathbb{R}^{N_{\text{snap}} \times N_{\text{snap}}}$
- Since usually $N_{\text{snap}} \ll N$, it is more economical to solve instead the symmetric eigenvalue problem

$$\mathbf{R}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i, \quad i = 1, \cdots, N_{\text{snap}}$$

- Note that the *non-zero* eigenvalues of the matrix $\mathbf{K} = \mathbf{S}\mathbf{S}^T \in \mathbb{R}^{N \times N}$ are the same as those of the matrix $\mathbf{R} = \mathbf{S}^T\mathbf{S} \in \mathbb{R}^{N_{\text{snap}} \times N_{\text{snap}}}$
- Since usually $N_{\text{snap}} \ll N$, it is more economical to solve instead the symmetric eigenvalue problem

$$\mathbf{R}\boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i, \quad i = 1, \cdots, N_{\text{snap}}$$

- However, if $\mathbf{S}$ is ill-conditioned, $\mathbf{R}$ is worse conditioned

$$\kappa_2(\mathbf{S}) = \sqrt{\kappa_2(\mathbf{S}^T\mathbf{S})} \Rightarrow \kappa_2(\mathbf{R}) = \kappa_2(\mathbf{S})^2$$

- If $\text{rank}(\mathbf{R}) = r$, then the first r POD modes ϕ_i are given by

$$\phi_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{S}\psi_i, \quad i = 1, \cdots, r$$

- If $\text{rank}(\mathbf{R}) = r$, then the first r POD modes ϕ_i are given by

$$\phi_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{S} \psi_i, \quad i = 1, \cdots, r$$

- Let $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_r \end{bmatrix}$ and $\mathbf{\Psi} = \begin{bmatrix} \psi_1 & \dots & \psi_r \end{bmatrix}$ with $\mathbf{\Psi}^T \mathbf{\Psi} = \mathbf{I}_r \Longrightarrow \mathbf{\Phi} = \mathbf{S}\mathbf{\Psi}\mathbf{\Lambda}^{-\frac{1}{2}}$ where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_r \end{bmatrix}$$

- If $\text{rank}(\mathbf{R}) = r$, then the first r POD modes ϕ_i are given by

$$\phi_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{S} \psi_i, \quad i = 1, \cdots, r$$

- Let $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_r \end{bmatrix}$ and $\mathbf{\Psi} = \begin{bmatrix} \psi_1 & \dots & \psi_r \end{bmatrix}$ with $\mathbf{\Psi}^T \mathbf{\Psi} = \mathbf{I}_r \Longrightarrow \mathbf{\Phi} = \mathbf{S} \mathbf{\Psi} \mathbf{\Lambda}^{-\frac{1}{2}}$ where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_r \end{bmatrix}$$

- $\mathbf{R} \psi_i = \lambda_i \psi_i, \quad i = 1, \cdots, N_{\text{snap}} \Rightarrow \mathbf{\Psi}^T \mathbf{R} \mathbf{\Psi} = \mathbf{\Psi}^T \mathbf{S}^T \mathbf{S} \mathbf{\Psi} = \mathbf{\Lambda}$
- Hence, $\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{\Psi}^T \underbrace{\mathbf{S}^T \mathbf{S}}_{\mathbf{R}^T} \underbrace{\mathbf{S}^T \mathbf{S}}_{\mathbf{R}} \mathbf{\Psi} \mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{\Lambda} \mathbf{\Psi}^T \mathbf{\Psi} \mathbf{\Lambda} \mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{\Lambda}$

- If $\text{rank}(\mathbf{R}) = r$, then the first r POD modes ϕ_i are given by

$$\phi_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{S}\psi_i, \quad i = 1, \cdots, r$$

- Let $\mathbf{\Phi} = \begin{bmatrix}\phi_1 & \dots & \phi_r\end{bmatrix}$ and $\mathbf{\Psi} = \begin{bmatrix}\psi_1 & \dots & \psi_r\end{bmatrix}$ with $\mathbf{\Psi}^T\mathbf{\Psi} = \mathbf{I}_r \Longrightarrow \mathbf{\Phi} = \mathbf{S}\mathbf{\Psi}\mathbf{\Lambda}^{-\frac{1}{2}}$ where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_r \end{bmatrix}$$

- $\mathbf{R}\psi_i = \lambda_i\psi_i, \quad i = 1, \cdots, N_{\text{snap}} \Rightarrow \mathbf{\Psi}^T\mathbf{R}\mathbf{\Psi} = \mathbf{\Psi}^T\mathbf{S}^T\mathbf{S}\mathbf{\Psi} = \mathbf{\Lambda}$
- Hence, $\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Psi}^T\underbrace{\mathbf{S}^T\mathbf{S}}_{\mathbf{R}^T}\underbrace{\mathbf{S}^T\mathbf{S}}_{\mathbf{R}}\mathbf{\Psi}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Lambda}\mathbf{\Psi}^T\mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{\Lambda}$
- Since the columns of $\mathbf{\Phi}$ are the eigenvectors of $\mathbf{K}$ ordered by decreasing eigenvalues, the optimal orthogonal basis of size $k \leq r$ is

$$\mathbf{V} = \begin{bmatrix}\mathbf{\Phi}_k & \mathbf{\Phi}_{r-k}\end{bmatrix}\begin{bmatrix}\mathbf{I}_k \\ \mathbf{0}\end{bmatrix} = \mathbf{\Phi}_k$$

- Parseval's theorem[1] (the Fourier transform is a unitary operator – that is, a surjective bounded operator on a Hilbert space preserving the inner product)

$$\lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \int_{-\frac{\mathcal{T}}{2}}^{\frac{\mathcal{T}}{2}} \|\mathbf{V}^T\mathbf{w}(t)\|_2^2 \, dt = \lim_{\mathcal{T},\,\Omega\to\infty} \frac{1}{2\pi\mathcal{T}} \int_{-\Omega}^{\Omega} \|\mathcal{F}\left[\mathbf{V}^T\mathbf{w}(t)\right]\|_2^2 \, d\omega$$

  where $\mathcal{F}[\mathbf{w}(t)] = \mathcal{W}(\omega)$ is the Fourier transform of $\mathbf{w}(t)$

- Consequence

$$\mathbf{V}^T\left(\lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \int_{-\frac{\mathcal{T}}{2}}^{\frac{\mathcal{T}}{2}} \mathbf{w}(t)\mathbf{w}(t)^T dt\right)\mathbf{V}$$

$$= \mathbf{V}^T\left(\lim_{\mathcal{T},\,\Omega\to\infty} \frac{1}{2\pi\mathcal{T}} \int_{-\Omega}^{\Omega} \mathcal{W}(\omega)\mathcal{W}(\omega)^* d\omega\right)\mathbf{V}$$

  (Proof: see Homework assignment #2)

[1] Rayleigh's energy theorem, Plancherel's theorem

- Let $\widetilde{\mathbf{K}}$ denote the analog to $\mathbf{K}$ in the frequency domain

$$\widetilde{\mathbf{K}} = \int_{-\Omega}^{\Omega} \mathcal{W}(\omega)\mathcal{W}(\omega)^* d\omega \approx \sum_{i=-N_{\text{snap}}^{\mathbb{C}}}^{N_{\text{snap}}^{\mathbb{C}}} \alpha_i \mathcal{W}(\omega_i)\mathcal{W}(\omega_i)^*$$

where $\omega_{-i} = -\omega_i$ is

- Let $\widetilde{\mathbf{K}}$ denote the analog to $\mathbf{K}$ in the frequency domain

$$\widetilde{\mathbf{K}} = \int_{-\Omega}^{\Omega} \mathcal{W}(\omega)\mathcal{W}(\omega)^* d\omega \approx \sum_{i=-N_{\text{snap}}^{\mathbb{C}}}^{N_{\text{snap}}^{\mathbb{C}}} \alpha_i \mathcal{W}(\omega_i)\mathcal{W}(\omega_i)^*$$

  where $\omega_{-i} = -\omega_i$ is

- The corresponding snapshot matrix is

$$\widetilde{\mathbf{S}} = \Big[\sqrt{\alpha_0}\mathcal{W}(\omega_0) \quad \sqrt{2\alpha_1}\text{Re}\left(\mathcal{W}(\omega_1)\right) \quad \ldots \quad \sqrt{2\alpha_{N_{snap}^{\mathbb{C}}}}\text{Re}\left(\mathcal{W}(\omega_{N_{snap}^{\mathbb{C}}})\right)$$
$$\sqrt{2\alpha_1}\text{Im}\left(\mathcal{W}(\omega_1)\right) \quad \ldots \quad \sqrt{2\alpha_{N_{snap}^{\mathbb{C}}}}\text{Im}\left(\mathcal{W}(\omega_{N_{snap}^{\mathbb{C}}})\right)\Big]$$

- Let $\widetilde{\mathbf{K}}$ denote the analog to $\mathbf{K}$ in the frequency domain

$$\widetilde{\mathbf{K}} = \int_{-\Omega}^{\Omega} \mathcal{W}(\omega)\mathcal{W}(\omega)^* d\omega \approx \sum_{i=-N_{\text{snap}}^{\mathbb{C}}}^{N_{\text{snap}}^{\mathbb{C}}} \alpha_i \mathcal{W}(\omega_i)\mathcal{W}(\omega_i)^*$$

where $\omega_{-i} = -\omega_i$ is

- The corresponding snapshot matrix is

$$\widetilde{\mathbf{S}} = \Big[\sqrt{\alpha_0}\mathcal{W}(\omega_0) \quad \sqrt{2\alpha_1}\text{Re}\left(\mathcal{W}(\omega_1)\right) \quad \ldots \quad \sqrt{2\alpha_{N_{snap}^{\mathbb{C}}}}\text{Re}\left(\mathcal{W}(\omega_{N_{snap}^{\mathbb{C}}})\right) \\ \sqrt{2\alpha_1}\text{Im}\left(\mathcal{W}(\omega_1)\right) \quad \ldots \quad \sqrt{2\alpha_{N_{snap}^{\mathbb{C}}}}\text{Im}\left(\mathcal{W}(\omega_{N_{snap}^{\mathbb{C}}})\right) \Big]$$

- It follows that

$$\widetilde{\mathbf{K}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T \qquad \widetilde{\mathbf{R}} = \widetilde{\mathbf{S}}^T\widetilde{\mathbf{S}} = \widetilde{\mathbf{\Psi}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{\Psi}}^T$$

$$\widetilde{\mathbf{\Phi}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{\Psi}}\widetilde{\mathbf{\Lambda}}^{-\frac{1}{2}} \qquad \widetilde{\mathbf{V}} = \begin{bmatrix} \widetilde{\mathbf{\Phi}}_k & \widetilde{\mathbf{\Phi}}_{N-r} \end{bmatrix} \begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix} = \widetilde{\mathbf{\Phi}}_k$$

$$\begin{aligned}\mathbf{f}(\mathbf{w}(t), t) &= \mathbf{A}\mathbf{w}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{g}(\mathbf{w}(t), t) &= \mathbf{C}\mathbf{w}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}$$

- Single input case: $p = 1 \Rightarrow \mathbf{B} \in \mathbb{R}^N$
- Time trajectory

$$\mathbf{w}(t) = e^{\mathbf{A}t}\mathbf{w}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau$$

- Snapshots in the time-domain for an impulse input $u(t) = \delta(t)$ and zero initial condition

$$\mathbf{w}(t_i) = e^{\mathbf{A}t_i}\mathbf{B}, \;\; t_i \geq 0$$

- In the frequency domain, the LTI system can be written as

$$j\omega_l\mathcal{W} = \mathbf{A}\mathcal{W} + \mathbf{B}, \;\; \omega_l \geq 0$$

and the associated **snapshots** are $\mathcal{W}(\omega_l) = (j\omega_l\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$

- How to sample the frequency domain?
  - approximate time trajectory for a zero initial condition

$$\mathbf{\Pi}_{\widetilde{\mathbf{V}},\widetilde{\mathbf{V}}}\mathbf{w}(t) = \widetilde{\mathbf{V}}\widetilde{\mathbf{V}}^T \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau$$

  - low-dimensional solution is accurate if the corresponding error is small — that is

$$\|\mathbf{w}(t) - \mathbf{\Pi}_{\widetilde{\mathbf{V}},\widetilde{\mathbf{V}}}\mathbf{w}(t)\| = \|(\mathbf{I} - \widetilde{\mathbf{V}}\widetilde{\mathbf{V}}^T)\int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau\|$$

    is small, which depends on the frequency content of $u(\tau)$
    $\Longrightarrow$ the sampled frequency band should contain the dominant frequencies of $u(\tau)$

- Given $\mathbf{A} \in \mathbb{R}^{N \times M}$, there exist two **orthogonal** matrices $\mathbf{U} \in \mathbb{R}^{N \times N}$ ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_N$) and $\mathbf{Z} \in \mathbb{R}^{M \times M}$ ($\mathbf{Z}^T\mathbf{Z} = \mathbf{I}_M$) such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T$$

where $\mathbf{\Sigma} \in \mathbb{R}^{N \times M}$ has diagonal entries

$$\mathbf{\Sigma}_{ii} = \sigma_i$$

satisfying

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(N,M)} \geq 0$$

and zero entries everywhere else

- $\{\sigma_i\}_{i=1}^{\min(N,M)}$ are the **singular values** of $\mathbf{A}$, and the columns of $\mathbf{U}$ and $\mathbf{Z}$ are the **left and right singular vectors** of $\mathbf{A}$, respectively

$$\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_N], \quad \mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_M]$$

- The SVD of a matrix provides many useful information about it (rank, range, null space, norm,...)
  - $\{\sigma_i^2\}_{i=1}^{\min(N,M)}$ are the eigenvalues of the symmetric positive, semi-definite matrices $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$
  - $\mathbf{A}\mathbf{z}_i = \sigma_i \mathbf{u}_i, \;\; i = 1, \cdots, \min(N, M)$
  - $\text{rank}(\mathbf{A}) = r$, where r is the index of the **smallest non-zero singular value**
  - if $\mathbf{U}_r = [\mathbf{u}_1 \cdots \mathbf{u}_r]$ and $\mathbf{Z}_r = [\mathbf{z}_1 \cdots \mathbf{z}_r]$ denote the singular vectors associated with the non-zero singular values and $\mathbf{U}_{N-r} = [\mathbf{u}_{r+1} \cdots \mathbf{u}_N]$ and $\mathbf{Z}_{M-r} = [\mathbf{z}_{r+1} \cdots \mathbf{z}_M]$, then
    - $\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{z}_1^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{z}_r^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{z}_i^T$
    - $\text{range}(\mathbf{A}) = \text{range}(\mathbf{U}_r) \qquad \text{range}(\mathbf{A}^T) = \text{range}(\mathbf{Z}_r)$
    - $\text{null}(\mathbf{A}) = \text{range}(\mathbf{Z}_{M-r}) \qquad \text{null}(\mathbf{A}^T) = \text{range}(\mathbf{U}_{N-r})$

- Given $\mathbf{A} \in \mathbb{R}^{N \times M}$ with $N \geq M$, **which matrix** $\mathbf{X} \in \mathbb{R}^{N \times M}$ **with** $\text{rank}(\mathbf{X}) = k < r = \text{rank}(\mathbf{A}) \leq M$ **minimizes** $\|\mathbf{A} - \mathbf{X}\|_2$**?**

Theorem (Schmidt-Eckart-Young-Mirsky)

$$\min_{\mathbf{X},\ rank(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1}(\mathbf{A}), \quad \textit{if}\ \sigma_k(\mathbf{A}) > \sigma_{k+1}(\mathbf{A})$$

- Hence, $\mathbf{X} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{z}_i^T$, where $\mathbf{A} = \mathbf{U}\Sigma\mathbf{Z}^T$, minimizes $\|\mathbf{A} - \mathbf{X}\|_2$

- Given $\mathbf{A} \in \mathbb{R}^{N \times M}$ with $N \geq M$, **which matrix** $\mathbf{X} \in \mathbb{R}^{N \times M}$ **with** $\text{rank}(\mathbf{X}) = k < r = \text{rank}(\mathbf{A}) \leq M$ **minimizes** $\|\mathbf{A} - \mathbf{X}\|_2$**?**

Theorem (Schmidt-Eckart-Young-Mirsky)

$$\min_{\mathbf{X},\ rank(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1}(\mathbf{A}), \quad \textit{if } \sigma_k(\mathbf{A}) > \sigma_{k+1}(\mathbf{A})$$

- Hence, $\mathbf{X} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{z}_i^T$, where $\mathbf{A} = \mathbf{U}\Sigma\mathbf{Z}^T$, minimizes $\|\mathbf{A} - \mathbf{X}\|_2$
- This minimizer is also the unique solution of the related problem (Eckart-Young theorem)

$$\min_{\mathbf{X},\ \text{rank}(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_F$$

- Given $\mathbf{A} \in \mathbb{R}^{N\times M}$ with $N \geq M$, **which matrix** $\mathbf{X} \in \mathbb{R}^{N\times M}$ **with** $\text{rank}(\mathbf{X}) = k < r = \text{rank}(\mathbf{A}) \leq M$ **minimizes** $\|\mathbf{A} - \mathbf{X}\|_2$**?**

Theorem (Schmidt-Eckart-Young-Mirsky)

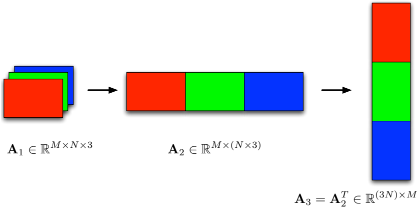
$$\min_{\mathbf{X},\ rank(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1}(\mathbf{A}), \quad if\ \sigma_k(\mathbf{A}) > \sigma_{k+1}(\mathbf{A})$$

- Hence, $\mathbf{X} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{z}_i^T$, where $\mathbf{A} = \mathbf{U}\Sigma\mathbf{Z}^T$, minimizes $\|\mathbf{A} - \mathbf{X}\|_2$
- This minimizer is also the unique solution of the related problem (Eckart-Young theorem)

$$\min_{\mathbf{X},\ \text{rank}(\mathbf{X})=k} \|\mathbf{A} - \mathbf{X}\|_F$$

- This result explains the concept of "low-rank" approximation and its connection with SVD

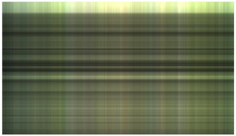
- Consider a color image in RGB representation made of $M \times N$ pixels, where $M < N$ (i.e., a landscape image)
  - this image can be represented by an $M \times N \times 3$ real matrix $\mathbf{A}_1$
  - $\mathbf{A}_1$ can be converted to a $3N \times M$ matrix $\mathbf{A}_3$ as follows



  - finally, $\mathbf{A}_3$ can be approximated using SVD as follows

$$\mathbf{A}_3 = \sigma_1 \mathbf{u}_1 \mathbf{z}_1^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{z}_r^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{z}_i^T$$

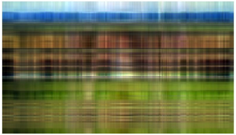
- Example: $\mathbf{A}_3 \in \mathbb{R}^{1497 \times 285}$



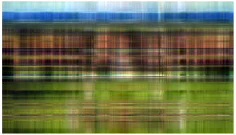
(a) rank 1

(b) rank 2

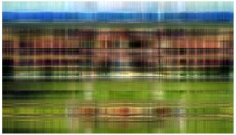
(c) rank 3



(d) rank 4



(e) rank 5



(f) rank 6


(g) rank 10


(h) rank 20


(i) rank 50


(j) rank 75


(k) rank 100


(l) rank 285

$\Longrightarrow$ SVD can be used for **data compression**

- The discretization of the POD by the method of snapshots requires computing the eigenspectrum of $\mathbf{K} = \mathbf{S}\mathbf{S}^T$

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Phi}^T \mathbf{S}\mathbf{S}^T \mathbf{\Phi} = \mathbf{\Lambda}$$

corresponding to its non-zero eigenvalues

- The discretization of the POD by the method of snapshots requires computing the eigenspectrum of $\mathbf{K} = \mathbf{S}\mathbf{S}^T$

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Phi}^T \mathbf{S}\mathbf{S}^T \mathbf{\Phi} = \mathbf{\Lambda}$$

  corresponding to its non-zero eigenvalues
- Link with the SVD of $\mathbf{S}$

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T = [\mathbf{U}_r \;\; \mathbf{U}_{N-r}] \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Z}^T$$

$$\Longrightarrow \mathbf{K} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T \quad \text{and} \quad \mathbf{U}^T\mathbf{K}\mathbf{U} = \mathbf{\Sigma}^2$$

$$\Longrightarrow \boxed{\mathbf{\Phi} = \mathbf{U}_r} \quad \text{and} \quad \mathbf{\Lambda}^{\frac{1}{2}} = \mathbf{\Sigma}_r \Leftrightarrow \mathbf{\Lambda} = \mathbf{\Sigma}_r^2$$

- The discretization of the POD by the method of snapshots requires computing the eigenspectrum of $\mathbf{K} = \mathbf{S}\mathbf{S}^T$

$$\boldsymbol{\Phi}^T \mathbf{K} \boldsymbol{\Phi} = \boldsymbol{\Phi}^T \mathbf{S}\mathbf{S}^T \boldsymbol{\Phi} = \boldsymbol{\Lambda}$$

  corresponding to its non-zero eigenvalues
- Link with the SVD of $\mathbf{S}$

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^T = [\mathbf{U}_r \quad \mathbf{U}_{N-r}] \left[\begin{array}{cc} \boldsymbol{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] \mathbf{Z}^T$$

$$\Longrightarrow \mathbf{K} = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T \quad \text{and} \quad \mathbf{U}^T\mathbf{K}\mathbf{U} = \boldsymbol{\Sigma}^2$$

$$\Longrightarrow \boxed{\boldsymbol{\Phi} = \mathbf{U}_r} \quad \text{and} \quad \boldsymbol{\Lambda}^{\frac{1}{2}} = \boldsymbol{\Sigma}_r \Leftrightarrow \boldsymbol{\Lambda} = \boldsymbol{\Sigma}_r^2$$

$$\Longrightarrow \boxed{\mathbf{U}_k \in \mathbb{R}^{N\times r} \text{ is to be identified with } \mathbf{X} \in \mathbb{R}^{N\times M}, N \geq M \geq r}$$

- The discretization of the POD by the method of snapshots requires computing the eigenspectrum of $\mathbf{K} = \mathbf{S}\mathbf{S}^T$

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Phi}^T \mathbf{S}\mathbf{S}^T \mathbf{\Phi} = \mathbf{\Lambda}$$

  corresponding to its non-zero eigenvalues
- Link with the SVD of $\mathbf{S}$

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T = [\mathbf{U}_r \quad \mathbf{U}_{N-r}] \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Z}^T$$

$$\Longrightarrow \mathbf{K} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T \quad \text{and} \quad \mathbf{U}^T\mathbf{K}\mathbf{U} = \mathbf{\Sigma}^2$$

$$\Longrightarrow \boxed{\mathbf{\Phi} = \mathbf{U}_r} \quad \text{and} \quad \mathbf{\Lambda}^{\frac{1}{2}} = \mathbf{\Sigma}_r \Leftrightarrow \mathbf{\Lambda} = \mathbf{\Sigma}_r^2$$

$$\Longrightarrow \boxed{\mathbf{U}_k \in \mathbb{R}^{N \times r} \text{ is to be identified with } \mathbf{X} \in \mathbb{R}^{N \times M}, N \geq M \geq r}$$

- Computing the SVD of $\mathbf{S}$ is usually preferred to computing the eigendecomposition of $\mathbf{R} = \mathbf{S}^T\mathbf{S}$ because, as noted earlier

$$\kappa_2(\mathbf{R}) = \kappa_2(\mathbf{S})^2$$

- How to choose the size k of the Reduced-Order Basis (ROB) $\mathbf{V}$ obtained using the POD method
  - start from the property of the Frobenius norm of $\mathbf{S}$

$$\|\mathbf{S}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})} \qquad \left(\text{recall } \|\mathbf{S}\|_F = \sqrt{\text{trace}(\mathbf{S}^T\mathbf{S})} = \sqrt{\text{trace}(\mathbf{S}\mathbf{S}^T)}\right)$$

- How to choose the size k of the Reduced-Order Basis (ROB) $\mathbf{V}$ obtained using the POD method
  - start from the property of the Frobenius norm of $\mathbf{S}$

$$\|\mathbf{S}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})} \qquad \left(\text{recall } \|\mathbf{S}\|_F = \sqrt{\text{trace}(\mathbf{S}^T\mathbf{S})} = \sqrt{\text{trace}(\mathbf{S}\mathbf{S}^T)}\right)$$

  - consider the error measured with the Frobenius norm induced by the truncation of the POD basis

$$\|(\mathbf{I}_N - \mathbf{V}\mathbf{V}^T)\mathbf{S}\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2(\mathbf{S})}$$

- How to choose the size k of the Reduced-Order Basis (ROB) $\mathbf{V}$ obtained using the POD method
  - start from the property of the Frobenius norm of $\mathbf{S}$

$$\|\mathbf{S}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})} \qquad \left(\text{recall } \|\mathbf{S}\|_F = \sqrt{\text{trace}(\mathbf{S}^T\mathbf{S})} = \sqrt{\text{trace}(\mathbf{S}\mathbf{S}^T)}\right)$$

  - consider the error measured with the Frobenius norm induced by the truncation of the POD basis

$$\|(\mathbf{I}_N - \mathbf{V}\mathbf{V}^T)\mathbf{S}\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2(\mathbf{S})}$$

  - the square of the relative error gives an indication of the magnitude of the "missing" information

$$\boxed{\mathcal{E}_{\text{POD}}(k) = \frac{\sum_{i=1}^{k} \sigma_i^2(\mathbf{S})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})} \Rightarrow 1 - \mathcal{E}_{\text{POD}}(k) = \frac{\sum_{i=k+1}^{r} \sigma_i^2(\mathbf{S})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})}}$$

- How to choose the size k of the ROB $\mathbf{V}$ obtained using the POD method (continue)

$$\mathcal{E}_{\text{POD}}(k) = \frac{\sum_{i=1}^{k} \sigma_i^2(\mathbf{S})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})}$$

  - $\mathcal{E}_{\text{POD}}(k)$ represents the relative energy of the snapshots captured by the k first POD basis vectors
  - k is usually chosen as the minimum integer for which

$$\boxed{1 - \mathcal{E}_{\text{POD}}(k) \leq \epsilon}$$

for a given tolerance $0 < \epsilon < 1$ (for instance $\epsilon = 0.1\%$)

- How to choose the size k of the ROB $\mathbf{V}$ obtained using the POD method (continue)

$$\mathcal{E}_{\text{POD}}(k) = \frac{\sum_{i=1}^{k} \sigma_i^2(\mathbf{S})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})}$$

  - $\mathcal{E}_{\text{POD}}(k)$ represents the relative energy of the snapshots captured by the k first POD basis vectors
  - k is usually chosen as the minimum integer for which

$$\boxed{1 - \mathcal{E}_{\text{POD}}(k) \leq \epsilon}$$

    for a given tolerance $0 < \epsilon < 1$ (for instance $\epsilon = 0.1\%$)
  - this criterion originates from turbulence applications

- Recall the model reduction error components

$$\begin{aligned} \mathcal{E}_{\text{PROM}}(t) &= \mathcal{E}_{\mathbf{V}^\perp}(t) + \mathcal{E}_{\mathbf{V}}(t) \\ &= \left(\mathbf{I}_N - \mathbf{\Pi}_{\mathbf{V},\mathbf{V}}\right)\mathbf{w}(t) + \mathbf{V}\left(\mathbf{V}^T\mathbf{w}(t) - \mathbf{q}(t)\right) \end{aligned}$$

  - denote $\mathcal{E}_{\text{PROM}}^{\text{snap}} = [\mathcal{E}_{\text{PROM}}(t_1) \quad \cdots \quad \mathcal{E}_{\text{PROM}}(t_{N_{\text{snap}}})]$

- Recall the model reduction error components

$$\begin{aligned}\mathcal{E}_{\text{PROM}}(t) &= \mathcal{E}_{\mathbf{V}^\perp}(t) + \mathcal{E}_{\mathbf{V}}(t) \\ &= \left(\mathbf{I}_N - \mathbf{\Pi}_{\mathbf{V},\mathbf{V}}\right)\mathbf{w}(t) + \mathbf{V}\left(\mathbf{V}^T\mathbf{w}(t) - \mathbf{q}(t)\right)\end{aligned}$$

  - denote $\mathcal{E}_{\text{PROM}}^{\text{snap}} = \left[\mathcal{E}_{\text{PROM}}(t_1) \quad \cdots \quad \mathcal{E}_{\text{PROM}}(t_{N_{\text{snap}}})\right]$
  - $\left\|\left[\mathcal{E}_{\mathbf{V}^\perp}(t_1) \quad \cdots \quad \mathcal{E}_{\mathbf{V}^\perp}(t_{N_{\text{snap}}})\right]\right\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2(\mathbf{S})}$

- Recall the model reduction error components

$$\begin{aligned}\mathcal{E}_{\text{PROM}}(t) &= \mathcal{E}_{\mathbf{V}^\perp}(t) + \mathcal{E}_{\mathbf{V}}(t) \\ &= \left(\mathbf{I}_N - \mathbf{\Pi}_{\mathbf{V},\mathbf{V}}\right)\mathbf{w}(t) + \mathbf{V}\left(\mathbf{V}^T\mathbf{w}(t) - \mathbf{q}(t)\right)\end{aligned}$$

  - denote $\mathcal{E}_{\text{PROM}}^{\text{snap}} = \left[\mathcal{E}_{\text{PROM}}(t_1) \quad \cdots \quad \mathcal{E}_{\text{PROM}}(t_{N_{\text{snap}}})\right]$
  - $\|[\mathcal{E}_{\mathbf{V}^\perp}(t_1) \quad \cdots \quad \mathcal{E}_{\mathbf{V}^\perp}(t_{N_{\text{snap}}})]\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2(\mathbf{S})}$
  - hence

$$1 - \mathcal{E}_{\text{POD}}(k) = \frac{\|[\mathcal{E}_{\mathbf{V}^\perp}(t_1) \quad \cdots \quad \mathcal{E}_{\mathbf{V}^\perp}(t_{N_{\text{snap}}})]\|_F^2}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})}$$

    and

$$1 - \mathcal{E}_{\text{POD}}(k) \leq \frac{\|\mathcal{E}_{\text{PROM}}^{\text{snap}}\|_F^2}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{S})}$$

  - note that the energy criterion is valid only for the sampled snapshots

- Consider the **parametrized steady-state** high-dimensional system of equations

$$\mathbf{f}(\mathbf{w}; \boldsymbol{\mu}) = \mathbf{0}, \ \boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^{d}, \ \boldsymbol{\mu} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_d]^T$$

- Consider the **parametrized steady-state** high-dimensional system of equations

$$\mathbf{f}(\mathbf{w}; \boldsymbol{\mu}) = \mathbf{0}, \ \ \boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d, \ \ \boldsymbol{\mu} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_d]^T$$

- Consider the goal of constructing a ROB and the associated projection-based PROM for computing the approximate solution

$$\mathbf{w}(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{q}(\boldsymbol{\mu}), \ \ \boldsymbol{\mu} \in \mathcal{D}$$

- Consider the **parametrized steady-state** high-dimensional system of equations

$$\mathbf{f}(\mathbf{w}; \boldsymbol{\mu}) = \mathbf{0}, \ \boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d, \ \boldsymbol{\mu} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_d]^T$$

- Consider the goal of constructing a ROB and the associated projection-based PROM for computing the approximate solution

$$\mathbf{w}(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{q}(\boldsymbol{\mu}), \ \boldsymbol{\mu} \in \mathcal{D}$$

- Question: How do we build a **global** ROB $\mathbf{V}$ that can capture the solution in the entire parameter domain $\mathcal{D}$?

- **Lagrange basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right)\right\} \Rightarrow N_{\text{snap}} = s$$

- **Lagrange basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right)\right\} \Rightarrow N_{\text{snap}} = s$$

- **Hermite basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \frac{\partial \mathbf{w}}{\partial \mu_1}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right), \frac{\partial \mathbf{w}}{\partial \mu_d}\left(\boldsymbol{\mu}^{(s)}\right)\right\}$$

$$\Rightarrow N_{\text{snap}} = s \times (d+1)$$

- **Lagrange basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right)\right\} \Rightarrow N_{\text{snap}} = s$$

- **Hermite basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \frac{\partial \mathbf{w}}{\partial \mu_1}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right), \frac{\partial \mathbf{w}}{\partial \mu_d}\left(\boldsymbol{\mu}^{(s)}\right)\right\}$$

$$\Rightarrow N_{\text{snap}} = s \times (d+1)$$

- **Taylor basis**

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \frac{\partial \mathbf{w}}{\partial \mu_1}\left(\boldsymbol{\mu}^{(1)}\right), \frac{\partial^2 \mathbf{w}}{\partial \mu_1^2}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \frac{\partial^q \mathbf{w}}{\partial \mu_1^q}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \frac{\partial \mathbf{w}}{\partial \mu_d}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \frac{\partial^q \mathbf{w}}{\partial \mu_d^q}\left(\boldsymbol{\mu}^{(1)}\right)\right\}$$

$$\Rightarrow N_{\text{snap}} = 1 + d + \frac{d(d+1)}{2} + \cdots + \frac{(d+q-1)!}{(d-1)!q!} = 1 + \sum_{i=1}^{q} \frac{(d+i-1)!}{(d-1)!i!}$$

- How one chooses the s parameter samples $\boldsymbol{\mu}^{(1)}, \cdots, \boldsymbol{\mu}^{(s)}$ where to compute the snapshots $\left\{\mathbf{w}\left(\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(\boldsymbol{\mu}^{(s)}\right)\right\}$?
  - the location of the samples in the parameter space will determine the accuracy of the resulting global PROM in the entire parameter domain $\mathcal{D} \subset \mathbb{R}^d$
- Possible approaches
  - uniform sampling for parameter spaces of moderate dimensions ($d \leq 5$) and moderately computationally intensive High-Dimensional Models (HDMs)
  - Latin Hypercube Sampling (LHS) for higher-dimensional parameter spaces and moderately computationally intensive HDMs
  - adaptive, goal-oriented, greedy sampling that exploits an error indicator to focus on the PROM accuracy, for higher-dimensional parameter spaces and computationally intensive HDMs

- Sampling methods grounded in statistics

- Sampling methods grounded in statistics
  - *Latin Hypercube Sampling (LHS).* First, the total number of sample points is set and then for each sample point, the row and column where the sample point is taken is remembered – ensures that the set of random numbers is representative of the real variability
    - latin square: A square grid containing sample positions where there is only one sample in each row and each column
    - a latin hypercube: Generalization of this concept to an arbitrary number of dimensions, where a single point is sampled in each axis-aligned hyperplane containing it

- Sampling methods grounded in statistics
  - *Latin Hypercube Sampling (LHS).* First, the total number of sample points is set and then for each sample point, the row and column where the sample point is taken is remembered – ensures that the set of random numbers is representative of the real variability
    - latin square: A square grid containing sample positions where there is only one sample in each row and each column
    - a latin hypercube: Generalization of this concept to an arbitrary number of dimensions, where a single point is sampled in each axis-aligned hyperplane containing it
  - *Orthogonal Sampling.* First, the sample space is divided into equally probable subspaces and then all sample points are simultaneously chosen, ensuring that the total set of sample points is a Latin Hypercube sample and that each subspace is sampled with the same density – ensures that the set of random numbers is a very good representative of the real variability

- Sampling methods grounded in statistics
  - *Latin Hypercube Sampling (LHS).* First, the total number of sample points is set and then for each sample point, the row and column where the sample point is taken is remembered – ensures that the set of random numbers is representative of the real variability
    - latin square: A square grid containing sample positions where there is only one sample in each row and each column
    - a latin hypercube: Generalization of this concept to an arbitrary number of dimensions, where a single point is sampled in each axis-aligned hyperplane containing it
  - *Orthogonal Sampling.* First, the sample space is divided into equally probable subspaces and then all sample points are simultaneously chosen, ensuring that the total set of sample points is a Latin Hypercube sample and that each subspace is sampled with the same density – ensures that the set of random numbers is a very good representative of the real variability
  - *Random Sampling.* New sample points are generated without taking into account the previously generated sample points – no specific guarantees

- Sampling methods grounded in statistics
  - *Latin Hypercube Sampling (LHS).* First, the total number of sample points is set and then for each sample point, the row and column where the sample point is taken is remembered – ensures that the set of random numbers is representative of the real variability
    - latin square: A square grid containing sample positions where there is only one sample in each row and each column
    - a latin hypercube: Generalization of this concept to an arbitrary number of dimensions, where a single point is sampled in each axis-aligned hyperplane containing it
  - *Orthogonal Sampling.* First, the sample space is divided into equally probable subspaces and then all sample points are simultaneously chosen, ensuring that the total set of sample points is a Latin Hypercube sample and that each subspace is sampled with the same density – ensures that the set of random numbers is a very good representative of the real variability
  - *Random Sampling.* New sample points are generated without taking into account the previously generated sample points – no specific guarantees
- None of these methods knows anything about the HDM or PROM to be constructed

- Ideally, one can build a PROM *progressively* and update it (increase its dimension) by considering additional samples $\boldsymbol{\mu}^{(i)}$ and corresponding solution snapshots at the locations of the parameter space where the *current* PROM is the most inaccurate – that is,

$$\boldsymbol{\mu}^{(i)} = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathbf{w}(\boldsymbol{\mu}) - \mathbf{V}\mathbf{q}(\boldsymbol{\mu})\|$$

  - $\mathbf{q}(\boldsymbol{\mu})$ can be efficiently computed
  - but the cost of obtaining $\mathbf{w}(\boldsymbol{\mu})$ can be high $\Rightarrow$ eventually an intractable approach

- Ideally, one can build a PROM *progressively* and update it (increase its dimension) by considering additional samples $\boldsymbol{\mu}^{(i)}$ and corresponding solution snapshots at the locations of the parameter space where the *current* PROM is the most inaccurate – that is,

$$\boldsymbol{\mu}^{(i)} = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathbf{w}(\boldsymbol{\mu}) - \mathbf{V}\mathbf{q}(\boldsymbol{\mu})\|$$

  - $\mathbf{q}(\boldsymbol{\mu})$ can be efficiently computed
  - but the cost of obtaining $\mathbf{w}(\boldsymbol{\mu})$ can be high $\Rightarrow$ eventually an intractable approach
- Idea: rely on an economical *a posteriori* error estimator/indicator
  - option 1: error bound

$$\|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| \leq \Delta(\boldsymbol{\mu})$$

  - option 2: error indicator based on the norm of the (affordable) residual

$$\|\mathbf{r}(\boldsymbol{\mu})\| = \|\mathbf{f}\left(\mathbf{V}\mathbf{q}(\boldsymbol{\mu}); \boldsymbol{\mu}\right)\|$$

- Ideally, one can build a PROM *progressively* and update it (increase its dimension) by considering additional samples $\boldsymbol{\mu}^{(i)}$ and corresponding solution snapshots at the locations of the parameter space where the *current* PROM is the most inaccurate – that is,

$$\boldsymbol{\mu}^{(i)} = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| = \underset{\boldsymbol{\mu} \in \mathcal{D}}{\text{argmax}} \|\mathbf{w}(\boldsymbol{\mu}) - \mathbf{V}\mathbf{q}(\boldsymbol{\mu})\|$$

  - $\mathbf{q}(\boldsymbol{\mu})$ can be efficiently computed
  - but the cost of obtaining $\mathbf{w}(\boldsymbol{\mu})$ can be high $\Rightarrow$ eventually an intractable approach
- Idea: rely on an economical *a posteriori* error estimator/indicator
  - option 1: error bound

$$\|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| \leq \Delta(\boldsymbol{\mu})$$

  - option 2: error indicator based on the norm of the (affordable) residual

$$\|\mathbf{r}(\boldsymbol{\mu})\| = \|\mathbf{f}\left(\mathbf{V}\mathbf{q}(\boldsymbol{\mu}); \boldsymbol{\mu}\right)\|$$

- For this purpose, $\mathcal{D}$ is typically replaced by a large discrete set of candidate parameters $\left\{\boldsymbol{\mu}^{\star^{(1)}}, \cdots, \boldsymbol{\mu}^{\star^{(c)}}\right\} \subset \mathcal{D}$

- Greedy procedure based on the norm of the residual as an error indicator

- Greedy procedure based on the norm of the residual as an error indicator
- Algorithm (given a termination criterion)
  1. randomly select a first sample $\boldsymbol{\mu}^{(1)}$
  2. solve the HDM-based problem

$$\mathbf{f}\left(\mathbf{w}(\boldsymbol{\mu}^{(1)}); \boldsymbol{\mu}^{(1)}\right) = \mathbf{0}$$

  3. build a corresponding ROB $\mathbf{V}$
  4. for $i = 2, \cdots$
  5. solve

$$\boldsymbol{\mu}^{(i)} = \underset{\boldsymbol{\mu} \in \left\{\boldsymbol{\mu}^{\star(1)}, \cdots, \boldsymbol{\mu}^{\star(c)}\right\}}{\operatorname{argmax}} \|\mathbf{r}(\boldsymbol{\mu})\|$$

  6. solve the HDM-based problem

$$\mathbf{f}\left(\mathbf{w}(\boldsymbol{\mu}^{(i)}); \boldsymbol{\mu}^{(i)}\right) = \mathbf{0}$$

  7. build a ROB $\mathbf{V}$ based on the snapshots (or in this case, samples) $\left\{\mathbf{w}(\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{w}(\boldsymbol{\mu}^{(i)})\right\}$

- Parameterized HDM

$$\frac{d\mathbf{w}}{dt}(t;\boldsymbol{\mu}) = \mathbf{f}\left(\mathbf{w}(t;\boldsymbol{\mu}), t; \boldsymbol{\mu}\right)$$

- Lagrange basis

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(t_1;\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(t_{N_t};\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(t_1;\boldsymbol{\mu}^{(s)}\right), \cdots, \mathbf{w}\left(t_{N_t};\boldsymbol{\mu}^{(s)}\right)\right\} \Rightarrow N_{\text{snap}} = s \times N_t$$

- *A posteriori* error estimator/indicator
  - option 1: error bound

$$\|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| = \left(\int_0^T \|\mathcal{E}_{\text{PROM}}(t;\boldsymbol{\mu})\|^2 \, dt\right)^{1/2} \leq \Delta(\boldsymbol{\mu})$$

  - option 2: error indicator based on the norm of the (affordable) residual

- Parameterized HDM

$$\frac{d\mathbf{w}}{dt}(t;\boldsymbol{\mu}) = \mathbf{f}\left(\mathbf{w}(t;\boldsymbol{\mu}), t;\boldsymbol{\mu}\right)$$

- Lagrange basis

$$\mathbf{V} \subset \text{span}\left\{\mathbf{w}\left(t_1;\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(t_{N_t};\boldsymbol{\mu}^{(1)}\right), \cdots, \mathbf{w}\left(t_1;\boldsymbol{\mu}^{(s)}\right), \cdots, \mathbf{w}\left(t_{N_t};\boldsymbol{\mu}^{(s)}\right)\right\} \Rightarrow N_{\text{snap}} = s \times N_t$$

- *A posteriori* error estimator/indicator
  - option 1: error bound

$$\|\mathcal{E}_{\text{PROM}}(\boldsymbol{\mu})\| = \left(\int_0^T \|\mathcal{E}_{\text{PROM}}(t;\boldsymbol{\mu})\|^2 \, dt\right)^{1/2} \leq \Delta(\boldsymbol{\mu})$$

  - option 2: error indicator based on the norm of the (affordable) residual

$$\|\mathbf{r}(\boldsymbol{\mu})\| = \left(\int_0^T \|\mathbf{r}(t;\boldsymbol{\mu})\|^2 \, dt\right)^{1/2} = \sqrt{\int_0^T \left\|\frac{d\mathbf{w}}{dt}(t;\boldsymbol{\mu}) - \mathbf{f}(\mathbf{Vq}(t;\boldsymbol{\mu}), t;\boldsymbol{\mu})\right\|^2 dt}$$

- Greedy procedure based on the residual norm as an error indicator

- Greedy procedure based on the residual norm as an error indicator
- Algorithm (given a termination criterion)
  1. randomly select a first sample $\boldsymbol{\mu}^{(1)}$
  2. solve the HDM-based problem
  $$\frac{d\mathbf{w}}{dt}(t;\boldsymbol{\mu}^{(1)}) = \mathbf{f}\left(\mathbf{w}(t;\boldsymbol{\mu}^{(1)}), t; \boldsymbol{\mu}^{(1)}\right)$$
  3. build a ROB $\mathbf{V}$ based on the snapshots
  $$\left\{\mathbf{w}(t_1;\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{w}(t_{N_t};\boldsymbol{\mu}^{(1)})\right\}$$
  4. for $i = 2, \cdots$
  5. solve
  $$\boldsymbol{\mu}^{(i)} = \underset{\boldsymbol{\mu}\in\left\{\boldsymbol{\mu}^{\star(1)},\cdots,\boldsymbol{\mu}^{\star(c)}\right\}}{\text{argmax}} \|\mathbf{r}(\boldsymbol{\mu})\|$$
  6. solve the HDM-based problem
  $$\frac{d\mathbf{w}}{dt}(t;\boldsymbol{\mu}^{(i)}) = \mathbf{f}\left(\mathbf{w}(t;\boldsymbol{\mu}^{(i)}), t; \boldsymbol{\mu}^{(i)}\right)$$
  7. build a ROB $\mathbf{V}$ based on the snapshots
  $$\left\{\mathbf{w}(t_1;\boldsymbol{\mu}^{(1)}), \cdots, \mathbf{w}(t_{N_t};\boldsymbol{\mu}^{(i)})\right\}$$

- Recall $1 - \mathcal{E}_{\text{POD}} \leq \epsilon; \qquad 0 < \epsilon < 1$

(m) $\epsilon < 10^{-1} \Rightarrow$ rank 2

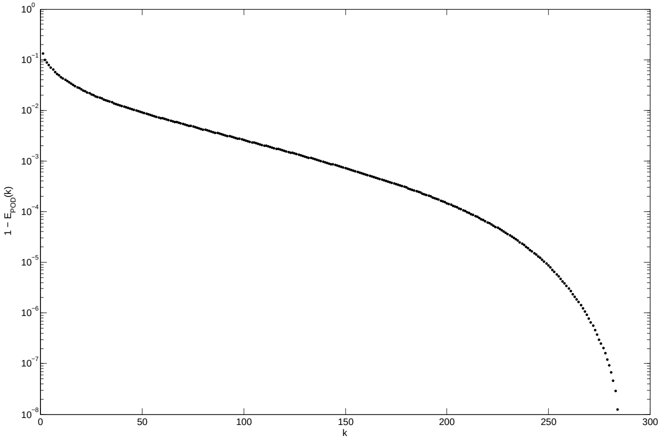
(n) $\epsilon < 10^{-2} \Rightarrow$ rank 47

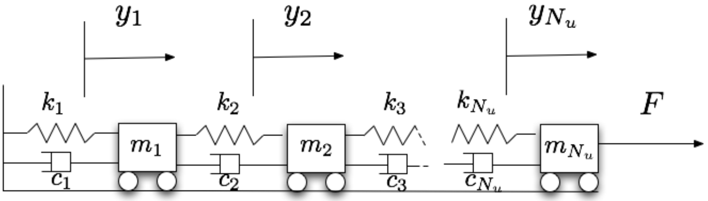
(o) $\epsilon < 10^{-3} \Rightarrow$ rank 138

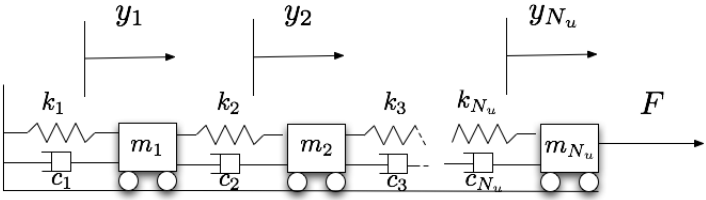
(p) $\epsilon < 10^{-4} \Rightarrow$ rank 210

(q) $\epsilon < 10^{-5} \Rightarrow$ rank 249

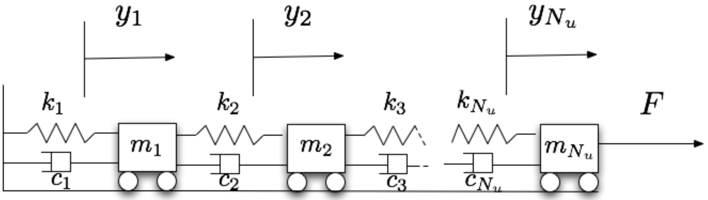
(r) $\epsilon < 10^{-6} \Rightarrow$ rank 269


1 − E_POD(k)
k
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
0
50
100
150
200
250
300


y_1
y_2
y_{N_u}
k_1
k_2
k_3
k_{N_u}
F
m_1
m_2
m_{N_u}
c_1
c_2
c_3
c_{N_u}

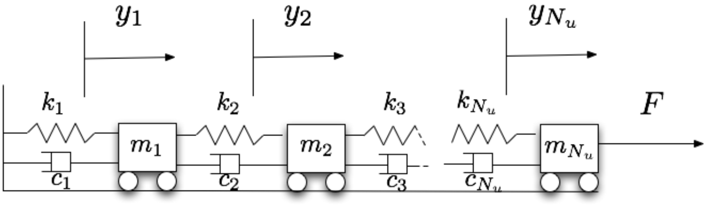


- LTI form



- LTI form
- $N_u = 48$ masses $\Rightarrow$ $N = 96$ degrees of freedom in state space form
- Transfer function of the HDM (frequency domain, $q = 1 \Rightarrow$ scalar)

$$\mathbf{H}(s;\boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu})\Big(s\mathbf{I}_N - \mathbf{A}(\boldsymbol{\mu})\Big)^{-1}\mathbf{B}(\boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu}), \quad s \in \mathbb{C}$$



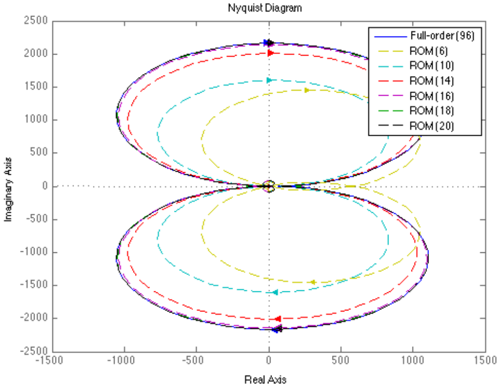
- LTI form
- $N_u = 48$ masses $\Rightarrow N = 96$ degrees of freedom in state space form
- Transfer function of the HDM (frequency domain, $q = 1 \Rightarrow$ scalar)

$$\mathbf{H}(s;\boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu})\Big(s\mathbf{I}_N - \mathbf{A}(\boldsymbol{\mu})\Big)^{-1}\mathbf{B}(\boldsymbol{\mu}) + \mathbf{D}(\boldsymbol{\mu}), \quad s \in \mathbb{C}$$

- Projection-based Model Order Reduction (PMOR) using POD in the frequency domain
- Transfer function of the PROM (frequency domain, $q = 1 \Rightarrow$ scalar)

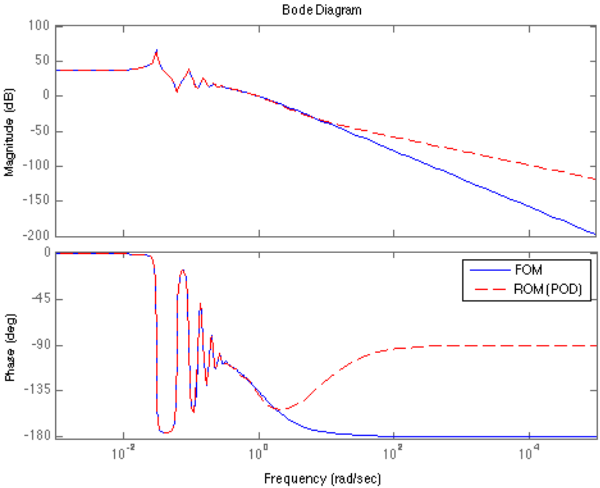
$$\mathbf{H}_r(s;\boldsymbol{\mu}) = \mathbf{C}_r(\boldsymbol{\mu})\Big(s\mathbf{I}_k - \mathbf{A}_r(\boldsymbol{\mu})\Big)^{-1}\mathbf{B}_r(\boldsymbol{\mu}) + \mathbf{D}_r(\boldsymbol{\mu}), \quad s \in \mathbb{C}$$

- Nyquist plots

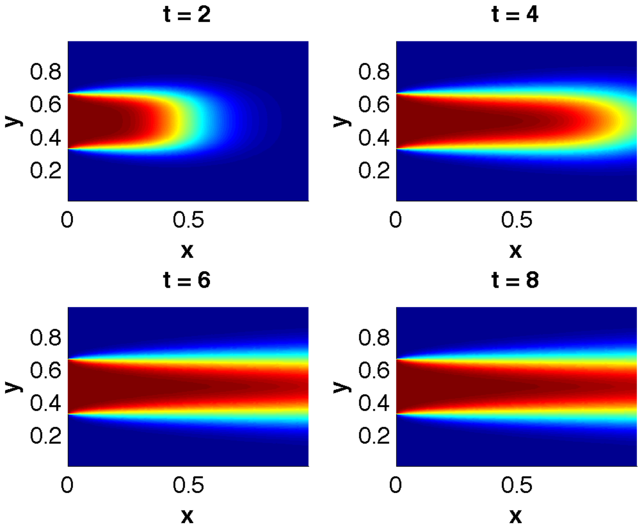


$\Rightarrow$ this leads to the choice of a PROM of size $k = 18$

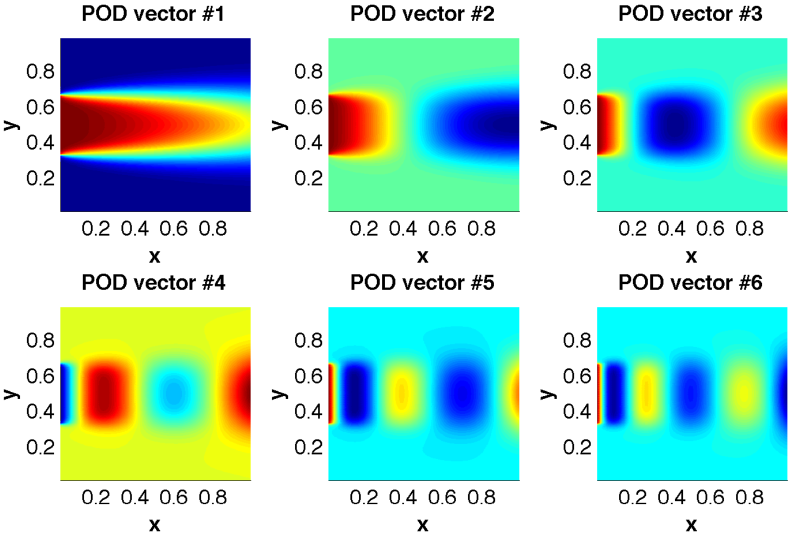
- Bode diagram for a PROM of size $k = 18$



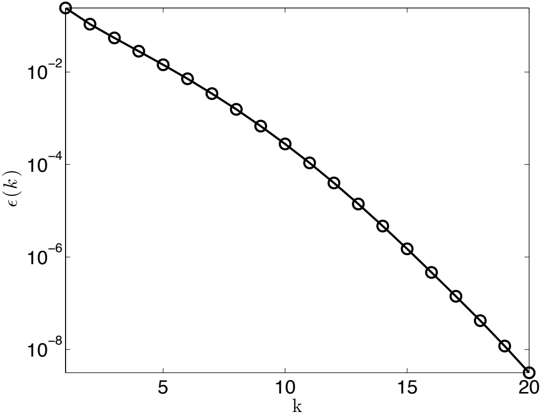
- HDM ($N = 5,402$)



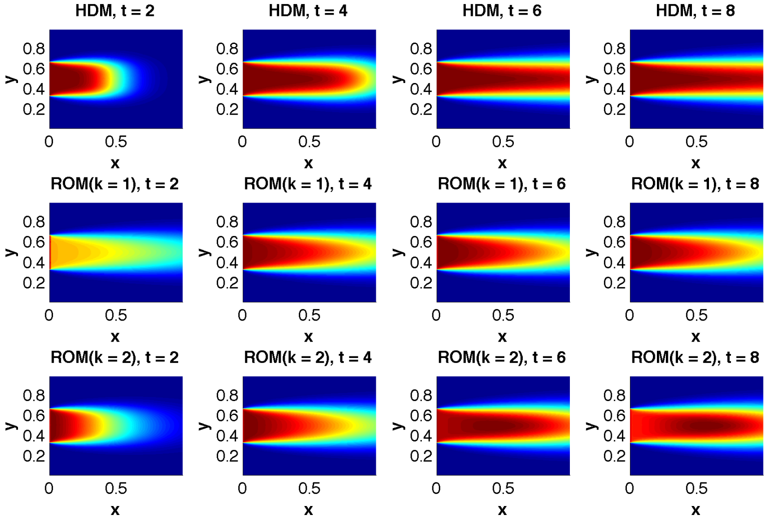
- POD modes


POD vector #1
POD vector #2
POD vector #3
POD vector #4
POD vector #5
POD vector #6
0.8
0.6
0.4
0.2
y
0.2 0.4 0.6 0.8
x

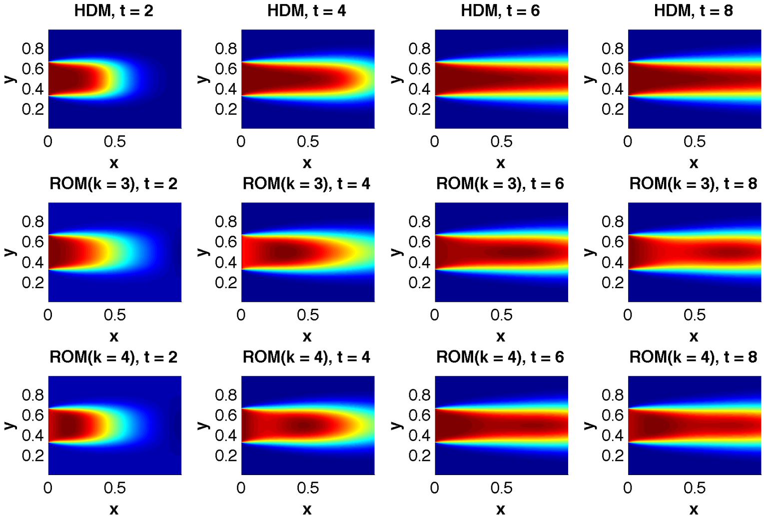
- Projection error (singular values decay)



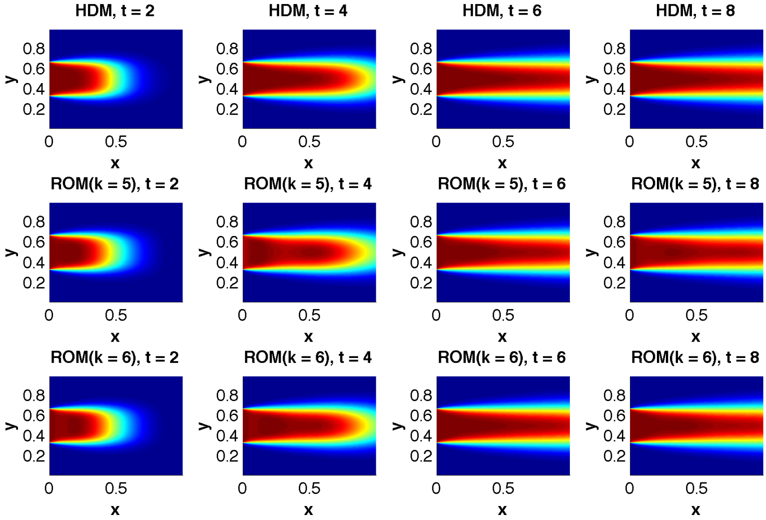
- POD-based PROM ($k = 1$ and $k = 2$)



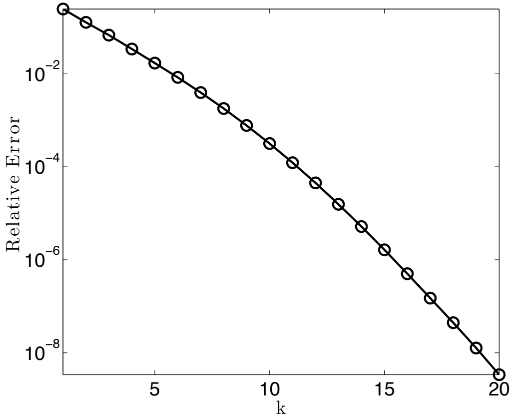
- POD-based PROM ($k = 3$ and $k = 4$)



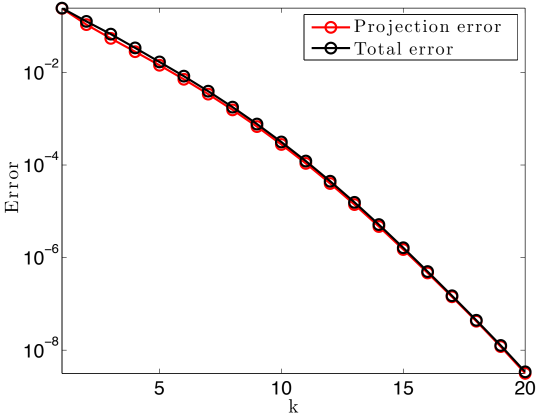
- POD-based PROM ($k = 5$ and $k = 6$)



- Model reduction error $\mathcal{E}_{\text{PROM}}(t)$



- Model reduction error $\mathcal{E}_{\text{PROM}}(t)$ and projection error $\mathcal{E}_{\mathbf{V}^{\perp}}(t)$



$\Rightarrow$ for this problem, $\mathcal{E}_{\mathbf{V}^{\perp}}(t)$ dominates $\mathcal{E}_{\mathbf{V}}(t)$