Proper Orthogonal Decomposition (POD)

Saifon Chaturantabut

Advisor: Dr. Sorensen
CAAM699
Department of Computational and Applied Mathematics
Rice University

September 5, 2008

Outline

1. Introduction/Motivation
   - Low Rank Approximation and POD
   - Model Reduction for ODEs and PDEs
2. Proper Orthogonal Decomposition(POD)
   - Introduction to POD
   - POD in Euclidean Space
   - POD in General Hilbert Space
3. Numerical Results
   - Solutions from Full and Reduced Systems of PDE: 1D Burgers'Equation
4. Problem: POD for PDEs with nonlinearities
   - Nonlinear Approximation
   - Error and Computational Time

Problem in finite dimensional space

- Given $y_1, \ldots, y_n \in \mathbb{R}^m$. Let $\mathscr{Y} = span\{y_1, y_2, \ldots, y_n\} \subset \mathbb{R}^m$; $r = rank(\mathscr{Y})$
- $y_1, \ldots, y_n \in \mathbb{R}^m$ possible (almost) linearly dependent $\Rightarrow$ NOT a good basis for $\mathscr{Y}$
- Goal: Find orthonormal basis vectors $\{\phi_1, \ldots, \phi_k\}$ that *best* approximates $\mathscr{Y}$, for given $k < r$
- Solution: Use Low Rank Approximation
- Form a matrix of known data:

$$Y = \begin{bmatrix} | & & | \\ \mathbf{y}_1 & \ldots & \mathbf{y}_n \\ | & & | \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Low Rank Approximation

Singular Value Decomposition (SVD)

Let $Y \in \mathbb{R}^{m \times n}$, $r = rank(Y)$, and $k < r$.

Problem: Low Rank Approximation

$$\min_{\hat{Y}} \{ \| Y - \hat{Y} \|_F^2 : rank(\hat{Y}) = k \}$$

Solution

$\hat{Y}^* = U_k \Sigma_k V_k^T$ with min error $\| Y - \hat{Y}^* \|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$

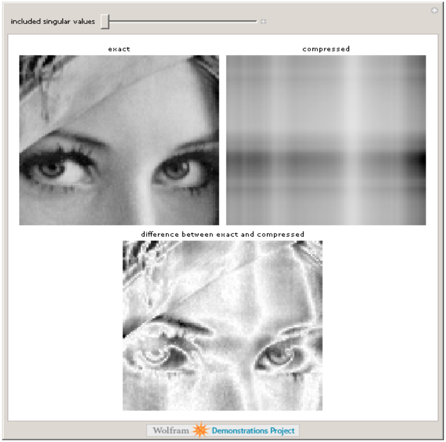
where $Y = U \Sigma V^T$ is SVD of Y; $\Sigma = diag(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r}$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$

Optimal orthonormal basis of rank k =: POD basis

$\hat{Y}^* = \sum_{i=1}^{k} \sigma_i u_i v_i^T \Rightarrow \boxed{\{\phi\}_{i=1}^{k} = \{u_i\}_{i=1}^{k}}$

EX: Image Compression via SVD

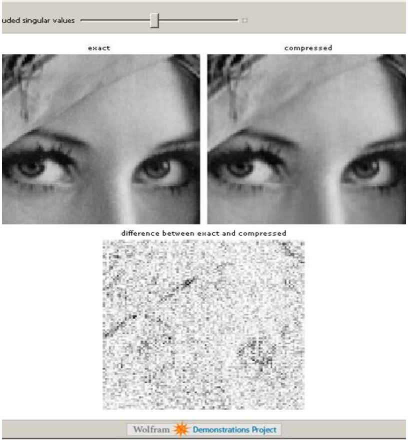
1 % Low Rank Approximation



source: http://demonstrations.wolfram.com/demonstrations.wolfram.com

EX: Image Compression via SVD

50 % Low Rank Approximation



source: http://demonstrations.wolfram.com/demonstrations.wolfram.com

Model Reduction for ODEs

- Full-order system (dim = N)

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{K}\mathbf{y}(t) + \mathbf{g}(t) + \mathbf{N}(\mathbf{y}(t)) \Rightarrow \boxed{y(t)}$$

- Reduced-order system (dim = $k < N$)

  Let $\mathbf{y} = \mathbf{U}_k\tilde{\mathbf{y}}$, for $\mathbf{U}_k \in \mathbb{R}^{N\times k}$, with orthonormal columns: $\mathbf{U}_k^T\mathbf{U}_k = \mathbf{I} \in \mathbb{R}^{k\times k}$.

$$\mathbf{U}_k \frac{d}{dt}\tilde{\mathbf{y}}(t) = \mathbf{K}\mathbf{U}_k\tilde{\mathbf{y}}(t) + \mathbf{g}(t) + \mathbf{N}(\mathbf{U}_k\mathbf{y}(t)) \Rightarrow \frac{d}{dt}\tilde{\mathbf{y}}(t) = \underbrace{\mathbf{U}_k^T\mathbf{K}\mathbf{U}_k}\,\tilde{\mathbf{y}}(t) + \underbrace{\mathbf{U}_k^T\mathbf{g}(t)} + \mathbf{U}_k^T\mathbf{N}(\mathbf{U}_k\mathbf{y}(t))$$

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \tilde{\mathbf{K}}\tilde{\mathbf{y}}(t) + \tilde{\mathbf{g}}(t) + \mathbf{U}_k^T\mathbf{N}(\mathbf{U}_k\tilde{\mathbf{y}}(t)) \Rightarrow \boxed{\tilde{\mathbf{y}}(t)}$$

How to construct $\mathbf{U_k}$? .....Use POD!

Model Reduction for PDEs

Ex. Unsteady 1D Burgers'Equation

$$\frac{\partial}{\partial t}y(x,t) - \nu\frac{\partial^2}{\partial x^2}y(x,t) + \frac{\partial}{\partial x}\left(\frac{y(x,t)^2}{2}\right) = 0 \qquad x \in [0,1], t \geq 0$$

$$y(0,t) = y(1,t) = 0,\ t \geq 0,\ \ y(x,0) = y_0(x),\ x \in [0,1],$$

Discretized system (Galerkin)

- Full-order system (dim = N): FE basis $\{\varphi_i\}_{i=1}^N$

$$\mathbf{M}_h\frac{d}{dt}\mathbf{y}(t) + \nu\mathbf{K}_h\mathbf{y}(t) - \mathbf{N}_h(\mathbf{y}(t)) = 0 \Rightarrow \boxed{y_h(x,t) = \sum_{i=1}^N \varphi_i(x)\mathbf{y}_i(t)}$$

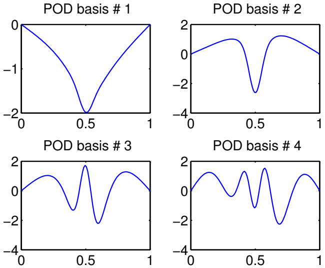
- Reduced-order system (dim = $k < N$): orthonormal basis $\{\phi_i\}_{i=1}^k$

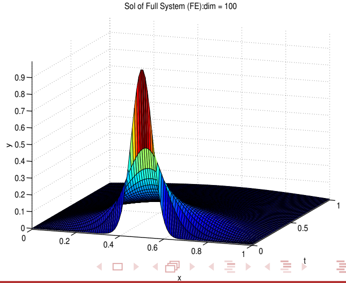
$$\tilde{\mathbf{M}}\frac{d}{dt}\tilde{\mathbf{y}}(t) + \nu\tilde{\mathbf{K}}\tilde{\mathbf{y}}(t) - \tilde{\mathbf{N}}(\tilde{\mathbf{y}}(t)) = 0 \Rightarrow \boxed{\tilde{y}_h(x,t) = \sum_{i=1}^k \phi_i(x)\tilde{\mathbf{y}}_i(t)}$$

How to construct $\{\phi_i\}_{i=1}^k$? .....Use POD!

Proper Orthogonal Decomposition(POD)

- POD is a method for finding a low-dimensional approximate representation of:
  - large-scale dynamical systems, e.g. signal analysis, turbulent fluid flow
  - large data set, e.g. image processing
- $\equiv$ SVD in Euclidean space.
- Extracts basis functions containing characteristics from the system of interest
- Generally gives a *good* approximation with *substantially lower* dimension





Definition of POD

- Let X be Hilbert Space(I.e. Complete Inner Product Space) with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$
- Given $y_1, \dots, y_n \in X$. Define $y_i \equiv$ *snapshot* i, $\forall i$. Let $\mathscr{Y} \equiv span\{y_1, y_2, \dots, y_n\} \subset X$.
- Given $k \leq n$, POD generates a set of orthonormal basis of dimension k, which minimizes the error from approximating the *snapshots*:

POD basis $\equiv$ Optimal solution of:

$$\min_{\{\phi\}_{i=1}^{k}} \sum_{j=1}^{n} \|y_j - \hat{y}_j\|^2, \text{ s.t. } \langle \phi_i, \phi_j \rangle = \delta_{ij}$$

where $\hat{y}_j(x) = \sum_{i=1}^{k} \langle y_j, \phi_i \rangle \phi_i(x)$, an approximation of y_j using $\{\phi\}_{i=1}^{k}$.

- Solve by SVD

Derivation for POD in Euclidean Space: E.g. $X = \mathbb{R}^m$

$$\min_{\{\phi\}_{i=1}^k} \quad \sum_{j=1}^n \|y_j - \sum_{i=1}^k (y_j^T \phi_i)\phi_i\|_2^2 \equiv E(\phi_1, ..., \phi_k)$$

$$s.t. \quad \phi_i^T \phi_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad i, j = 1, ..., k$$

Lagrange Function:

$$L(\phi_1, \ldots, \phi_k, \lambda_{11}, \ldots, \lambda_{ij}, \ldots, \lambda_{kk}) = E(\phi_1, ..., \phi_k) + \sum_{i,j=1}^{k} \lambda_{ij}(\phi_i^T \phi_j - \delta_{ij})$$

KKT Necessary Conditions:

- $\frac{\partial}{\partial \phi_i} L = 0 \Leftrightarrow \left\{ \begin{array}{l} \sum_{j=1}^n y_j(y_j^T \phi_i) = \lambda_{ii}\phi_i \\ \lambda_{ij} = 0, \text{ if} i \neq j \end{array} \right\}.$
- $\phi_i^T \phi_j = \delta_{ij}$

Optimal Condition: Symmetric m-by-m eigenvalue problem

$$YY^T\phi_i = \lambda_i\phi_i$$

where $\lambda_i = \lambda_{ii}$, $Y = \begin{bmatrix} | & & | \\ y_1 & \dots & y_n \\ | & & | \end{bmatrix}$ for $i = 1, \dots, k$

Error for POD basis:

$$\sum_{j=1}^{n} \|y_i - \sum_{i=1}^{k}(y_j^T\phi_i)\phi_i\|_2^2 = \sum_{i=k+1}^{r} \lambda_i$$

- $\lambda_i\phi_i = YY^T\phi_i = \sum_{j=1}^{n} y_j(y_j^T\phi_i) \Rightarrow \lambda_i = \sum_{j=1}^{n}(y_j^T\phi_i)^2$
- Since $\phi_i^T\phi_j = \delta_{ij}$ and $span(Y) = span\{\phi_1, \dots, \phi_r\}$ (from SVD), then $y_j = \sum_{i=1}^{r}(y_j^T\phi_i)\phi_i$, for$j = 1, \dots, n$, and

$$\sum_{j=1}^{n} \|y_j - \sum_{i=1}^{k}(y_j^T\phi_i)\phi_i\|_2^2 = \sum_{j=1}^{n}\sum_{i=k+1}^{r}(y_j^T\phi_i)^2 = \sum_{i=k+1}^{r}\lambda_i$$

SOLUTION for POD basis in $\mathbb{R}^m$

- Recall Optimality Condition and the POD Error:

$$\boxed{\begin{array}{c} YY^T\phi_i = \lambda_i\phi_i, i = 1, \dots, k \\ \sum_{j=1}^{n} \|y_j - \sum_{i=1}^{k}(y_j^T\phi_i)\phi_i\|_2^2 = \sum_{i=k+1}^{r} \lambda_i \end{array}}$$

- Optimal solution:

$$\boxed{\begin{array}{c} \text{POD basis: } \{\phi_i^*\}_{i=1}^k = \{u_i\}_{i=1}^k \\ \text{Lagrange multiplier: } \lambda_i^* = \sigma_i^2 \end{array}}, i = 1, \dots, k,$$

  can be obtained by the SVD of $Y \in \mathbb{R}^{m\times n}$ or EVD of $YY^T \in \mathbb{R}^{m\times m}$, i.e.,

$$Y = U\Sigma V^T \Rightarrow YY^T u_i = \sigma_i^2 u_i, i = 1, \dots, r,$$

  where $\Sigma = diag(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r\times r}$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$; $U = [u_1, \dots, u_r] \in \mathbb{R}^{m\times r}$ and $V = [v_1, \dots, v_r] \in \mathbb{R}^{n\times r}$ have orthonormal columns.

- *This is equivalent to SVD solution for Low Rank Approximation*

SOLUTION for POD basis in a Hilbert Space X

Two approaches:

1. - Define linear symmetric operator $\mathscr{F}(w) = \sum_{j=1}^{n} \langle w, y_j \rangle y_j$, $w \in X$. Find eigenfunction $u_i \in X$:

$$\boxed{\begin{array}{c} \text{EVD: } \mathscr{F}(u_i) = \sigma_i^2 u_i \\ \sum_{j=1}^{n} \langle u_i, y_j \rangle y_j = \sigma_i^2 u_i \end{array}} \sim \boxed{(\text{cf. } YY^T u_i = \sigma_i^2 u_i)}$$

   - $\boxed{\text{Sol: } \phi_i^* = u_i, \lambda_i^* = \sigma_i^2, i = 1, \ldots, k}$

2. - Define linear symmetric operator $\mathscr{L} = [\langle y_i, y_j \rangle] \in \mathbb{R}^{n \times n}$. Find eigenvector $v_i \in \mathbb{R}^n$:

$$\boxed{\begin{array}{c} \text{EVD: } \mathscr{L} v_i = \sigma_i^2 v_i \\ [\langle y_i, y_j \rangle] v_i = \sigma_i^2 v_i \end{array}} \sim \boxed{(\text{cf. } Y^T Y v_i = \sigma_i^2 v_i)}$$

   - $\boxed{\text{Sol: } \phi_i^* = \frac{1}{\sigma_i} \sum_{j=1}^{n} (v_i)_j y_j, \lambda_i^* = \sigma_i^2} \sim \boxed{U_k = Y V_k \Sigma_k^{-1}}$
   - NOTE: $v_i \in \mathbb{R}^n$,but $u_i, y_j \in X$

Remark: Practical way for computing the POD basis for discretized PDEs

- Let $\{\varphi_i\}_{i=1}^N \subset X \equiv H^1(\Omega)$ be finite element(FE) basis.
- FE snapshots(solutions): $\{y_j\}_{i=j}^n \in X$ at time $\{t_j\}_{j=1}^n$.

$$y_j = y(x, t_j) = \sum_{i=1}^{N} \hat{Y}_{ij}(t_j)\varphi_i(x).$$

- $$\mathscr{L} = [\langle y_i, y_j\rangle] = [\sum_{k,\ell=1}^{m} \hat{Y}_{ik}\hat{Y}_{j\ell}\langle\varphi_i, \varphi_j\rangle] = \hat{Y}^T M \hat{Y} \in \mathbb{R}^{n\times n},$$

  where $M = [\langle\varphi_i, \varphi_j\rangle] \in \mathbb{R}^{N\times N}$.
- POD basis:

$$\phi_i^* = \frac{1}{\sigma_i}\sum_{j=1}^{n}(v_i)_j y_j,$$

  where

$$\mathscr{L} v_i = \sigma_i^2 v_i,$$

  with $v_1, v_2, \ldots, v_k$ corresponding to $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$.

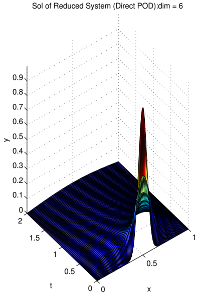
Numerical Results for PDEs

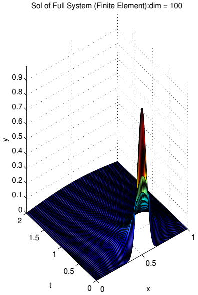
Recall the 1D Burgers'Equation $x \in [0,1], t \geq 0$:

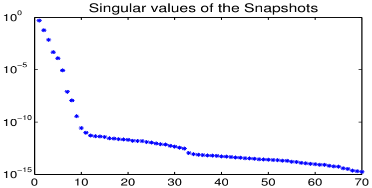
$$\frac{\partial}{\partial t} y(x,t) - \nu \frac{\partial^2}{\partial x^2} y(x,t) + \frac{\partial}{\partial x}\left(\frac{y(x,t)^2}{2}\right) = 0,$$

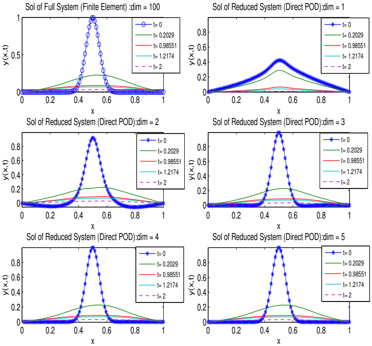
$$y(0,t) = y(1,t) = 0, \ t \geq 0,$$

$$y(x,0) = y_0(x), \ x \in [0,1].$$









Problem: POD for PDEs with nonlinearities

If we apply the POD basis directly to construct a discretized system, the original system of order N:

$$\mathbf{M}_h \frac{d}{dt}\mathbf{y}(t) + \nu \mathbf{K}_h \mathbf{y}(t) - \mathbf{N}_h(\mathbf{y}(t)) = 0$$

become a system of order $k \ll N$:

$$\tilde{\mathbf{M}} \frac{d}{dt}\tilde{\mathbf{y}}(t) + \nu \tilde{\mathbf{K}} \tilde{\mathbf{y}}(t) - \tilde{\mathbf{N}}(\tilde{\mathbf{y}}(t)) = 0,$$

where the **nonlinear term :**

$$\boxed{\tilde{\mathbf{N}}(\tilde{\mathbf{y}}(t)) = \underbrace{U_k^T}_{k \times N} \underbrace{\mathbf{N}_h(U_k \tilde{\mathbf{y}}(t))}_{N \times 1}}$$

$\Rightarrow$ Computational Complexity still depends on N!!

Nonlinear Approximation

Recall the problem from bf Direct POD :

$$\tilde{\mathbf{N}}(\tilde{\mathbf{y}}(t)) = \underbrace{U^T}_{k\times N} \underbrace{\mathbf{N}_h(U\tilde{\mathbf{y}}(t))}_{N\times 1}$$

WANT:

$$\boxed{\tilde{\mathbf{N}}(\tilde{\mathbf{y}}(t)) \leftarrow \underbrace{C}_{k\times n_m} \underbrace{\hat{\mathbf{N}}(\tilde{\mathbf{y}}(t))}_{n_m\times 1}} \dashrightarrow \boxed{k, n_m \ll N} \dashrightarrow \text{Independent of } N$$

2 Approaches:

- **Precomputing Technique:** Simple nonlinear
- **Empirical Interpolation Method (EIM):** General nonlinear

ACCURACY vs. COMPLEXITY

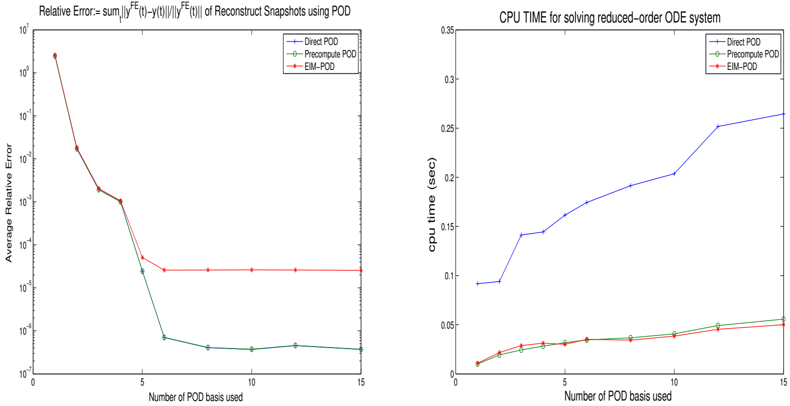


Figure: LEFT: Error $E_{avg} = \frac{1}{n_t}\sum_{i=1}^{n_t} \frac{\|\tilde{y}_h(\cdot,t_i)-y_h(\cdot,t_i)\|_X}{\|y_h(\cdot,t_i)\|_X}$ .RIGHT: CPU time (sec)

QUESTION ?

Empirical Interpolation Method (EIM) [Patera; 2004]

- Approximates nonlinear parametrized functions
- Let $s(x;\mu)$ be a parametrized function with spatial variable $x \in \Omega$ and parameter $\mu \in \mathscr{D}$.
- Function approximation from EIM:

$$\hat{s}(x;\mu) = \sum_{m=1}^{n_m} q_m(x)\beta_m(\mu),$$

where $span\{q_m\}_{m=1}^{n_m} \simeq \mathscr{M}^s \equiv \{s(\cdot;\mu) : \mu \in \mathscr{D}\}$ and $\beta_m(\mu)$ is specified from the interpolation points $\{z_m\}_{m=1}^{n_m}$, in Ω:

$$s(z_i;\mu) = \sum_{m=1}^{n_m} q_m(z_i)\beta_m(\mu),$$

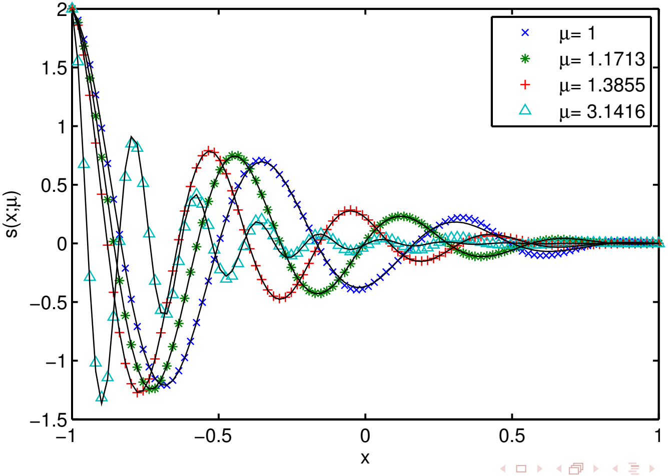
for $i = 1, \ldots, n_m$.

EIM: Numerical Example

$$s(x;\mu) = (1-x)cos(3\pi\mu(x+1))e^{-(1+x)\mu},$$

where $x \in [-1, 1]$ and $\mu \in [1, \pi]$.

Plot of Approximate Functions (dim = 10) with Exact Functions (in black solid line)



Plots of Numerical Solutions from 3 Approaches

- Direct POD
- Precomputed POD
- EIM-POD

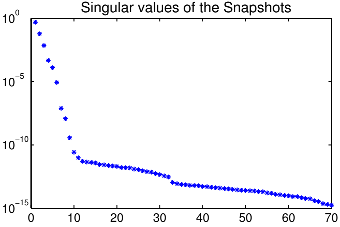


Figure: SVD

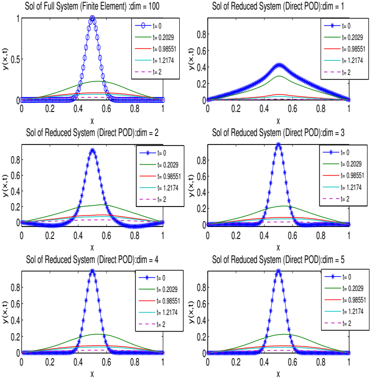


Figure: Direct POD

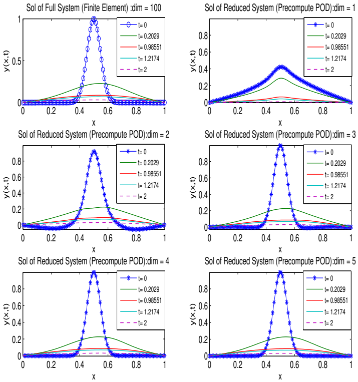


Figure: Precomputed POD

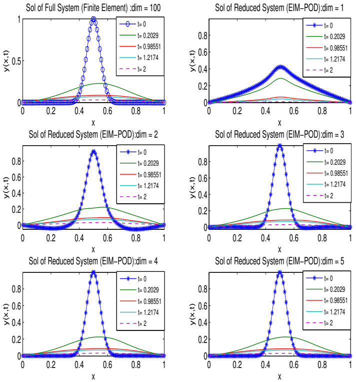


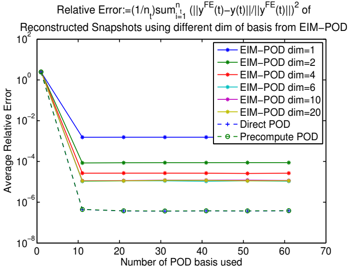
Figure: EIM-POD

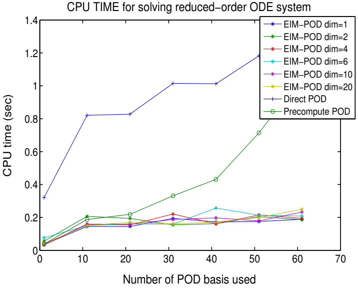
Conclusions

- Unique Contribution: Clear description of the EIM $\Rightarrow$ Successful Implementation of EIM with POD
- EIM is comparable to widely accepted methods, such as precomputing technique
- The results suggest that EIM with POD basis is a promising model reduction technique for more general nonlinear PDEs.

Future Work

- Extend to higher dimensions
- Extend to PDEs with more general nonlinearities
- Apply to practical problem, e.g. optimal control problem





Overview

Nonlinear PDE

$\Downarrow$ $\leftarrow - -$ FE-Galerkin

FULL Discretized System (ODE)
dim $= N$

$\Downarrow$ $\leftarrow - -$ Solve ODE

SNAPSHOTS: $\{y(x, t_\ell)\}_{\ell=1}^{n_s}$

$\Downarrow$ $\leftarrow - -$ $\boxed{\text{POD}}$

POD basis: $\{\phi_i\}_{i=1}^{k}$

$\Downarrow$ $\leftarrow - -$ POD-Galerkin

REDUCED Discretized System (ODE)
Linear Term: dim = $k \ll N$
Non-Linear Term: dim $\sim N$

$\longrightarrow$ $\longrightarrow$ (i) Memory Saving (ii) Accuracy

$\Downarrow$ $\leftarrow - -$ Nonlinear Approx -EIM -Precomputing

REDUCED Discretized System (ODE)
Linear Term: dim = $k \ll N$
Non-Linear Term: dim = $n_m \ll N$

$\longrightarrow$ $\longrightarrow$ (iii) Efficiency (Time Saving)