Discrete Empirical Interpolation for Nonlinear Model Reduction

S. Chaturantabut and D.C. Sorensen

Support: AFOSR and NSF

CICADA / MIMS workshop on Numerics for Control and Simulation

Manchester, UK June 17, 2009

Projection Methods for MOR

Brief Intro to Gramian Based Model Reduction

Proper Orthogonal Decomposition (POD)

A Problem with POD

Nonlinear MOR: Discrete Empirical Interpolation (DEIM)

Variant of EIM : Barrault, Maday, Nguyen and Patera (2004)

Neural Modeling: Local Reduction $\Rightarrow$ Many Interactions

Gramian Based Model Reduction

Proper Orthogonal Decomposition (POD)

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t)), \quad \mathbf{z} = \mathbf{g}(\mathbf{y}(t), \mathbf{u}(t))$$

The Gramian

$$\mathcal{P} = \int_o^\infty \mathbf{y}(\tau)\mathbf{y}(\tau)^T d\tau$$

Eigenvectors of $\mathcal{P}$

$$\mathcal{P} = \mathbf{V}\mathbf{S}^2\mathbf{V}^T$$

Orthogonal Basis

$$\mathbf{y}(t) = \mathbf{V}\mathbf{S}\mathbf{w}(t)$$

PCA or POD Reduced Basis

Snapshot Approximation to $\mathcal{P}$

$$\mathcal{P} \approx \frac{1}{m} \sum_{j=1}^{m} \mathbf{y}(t_j)\mathbf{y}(t_j)^T = \mathbf{Y}\mathbf{Y}^T$$

Truncate SVD : $\mathbf{Y} = \mathbf{V}\mathbf{S}\mathbf{W}^T \approx \mathbf{V}_k\mathbf{S}_k\mathbf{W}_k^T$
Low Rank Approximation

$$\mathbf{y} \approx \mathbf{V}_k\hat{\mathbf{y}}_k(t)$$

Galerkin condition – Global Basis

$$\dot{\hat{\mathbf{y}}}_k = \mathbf{V}_k^T\mathbf{f}(\mathbf{V}_k\hat{\mathbf{y}}_k(t), \mathbf{u}(t))$$

Global Approximation Error ? ($\mathcal{H}_2$ bound for LTI)

$$\|\mathbf{y} - \mathbf{V}_k\hat{\mathbf{y}}_k\|_2 \approx \sigma_{k+1}$$

POD vs. FEM

- Both are Galerkin Projection
- POD - Global Basis fns. vs FEM - Local Basis fns.
- FEM - Complex Behavior via Mesh Refinement/ Higher Order
  High Dimension - Sparse Matrices
- POD - Complex Behavior contained in Global Basis
  Low Dimension - Dense Matrices
- Caveat: POD is ad hoc: Must sample rich set of inputs and parameter settings
- Qx: How to automate parameter/input sampling for POD

Problem Setting

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{F}(\mathbf{y}(t)), \tag{1}$$

- $\mathbf{y} : [0, T] \mapsto \mathbb{R}^n$,
- $\mathbf{A} \in \mathbb{R}^{n\times n}$,
- $\mathbf{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$, *nonlinear* function $\mathbf{F}(\mathbf{y}(t))_j = F(\mathbf{y}_j(t))$.
- E.g. FD discretized system of nonlinear PDEs $\Rightarrow$ Large n.

Reduced system (dim $k \ll n$):

- $\mathbf{y}(t) \to \mathbf{V}_k\tilde{\mathbf{y}}(t), \quad \mathbf{V}_k \in \mathbf{R}^{n\times k}$

$$\frac{d}{dt}\tilde{\mathbf{y}}(t) = \underbrace{\mathbf{V}_k^T\mathbf{A}\mathbf{V}_k}_{k\times k}\tilde{\mathbf{y}}(t) + \underbrace{\mathbf{V}_k^T}_{k\times n}\underbrace{\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t))}_{n\times 1}. \tag{2}$$

$\Rightarrow$ Computational complexity of nonlinear term still depends on n.

Nonlinear Approximation

- Define $\mathbf{f}(t) := \mathbf{F}(\mathbf{V}_k \tilde{\mathbf{y}}(t))$ and the nonlinear term

$$\boxed{\widehat{\mathbf{F}}(\mathbf{y}(t)) := \underbrace{\mathbf{V}_k^T}_{k \times n} \underbrace{\mathbf{f}(t)}_{n \times 1}.} \Rightarrow \textit{dependence} \text{ on } n$$

- **<u>Goal</u>**: Approx $\mathbf{f}(t)$ by projecting on $\mathbf{U} \in \mathbb{R}^{n \times m}$, $m \ll n$

$$\mathbf{f}(t) \approx \mathbf{U}\mathbf{c}(t)$$

- $\Rightarrow$ Approximation of nonlinear term:

$$\boxed{\widehat{\mathbf{F}}(\mathbf{y}(t)) \approx \underbrace{\mathbf{V}_k^T \mathbf{U}}_{\text{precomp:} k \times m} \underbrace{\mathbf{c}(t)}_{m \times 1}} \Rightarrow \textit{independence} \text{ of } n$$

- How to determine $\mathbf{c}(t)$?.....**Interpolation!**

Interpolation

Consider overdetermined linear system ($n >> m$):
Want:

$$\mathbf{f}(t) \approx \mathbf{U}\mathbf{c}(t)$$

Select m *rows* $\wp_1, \ldots, \wp_m$ of $\mathbf{U}$

Determine $\mathbf{c}(t)$ from m-by-m linear system:

$$\mathbf{f}_{\vec{\wp}}(t) = \mathbf{U}_{\vec{\wp}}\mathbf{c}(t)$$

$$\mathbf{P}^T\mathbf{f}(t) = (\mathbf{P}^T\mathbf{U})\mathbf{c}(t)$$

$$\mathbf{P} = [\mathbf{e}_{\wp_1}, \ldots, \mathbf{e}_{\wp_m}] \in \mathbb{R}^{n\times m}$$

$\mathbf{P}^T$ *"extracts rows"* $\wp_1, \ldots, \wp_m$.

Complexity Reduction

$$\begin{aligned} \mathbf{c}(t) &= \mathbf{U}_{\vec{\wp}}^{-1}\mathbf{f}_{\vec{\wp}}(t) \\ &= (\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(t) \end{aligned}$$

$$\begin{aligned} \mathbf{P}^T\mathbf{f}(t) &= \mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}(t)) \\ &= \mathbf{F}(\mathbf{P}^T\mathbf{V}_k\tilde{\mathbf{y}}(t)) \end{aligned}$$

Complexity: $m \cdot k \; + \; m$ **Evaluations** $F(\,\cdot\,)$

Interpolation

- Interpolation approximation of $\mathbf{f}(t)$

$$\boxed{\hat{\mathbf{f}}(t) = \mathbf{U}\mathbf{c}(t) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(t) = \mathbb{P}\,\mathbf{f}(t)}$$

- Interpolation $\Rightarrow$ Projection:

  $\mathbb{P} := \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T$ is an oblique projector onto $\mathrm{Range}\{\mathbf{U}\}$.

- The approximation is exact at entries $\wp_1, \ldots, \wp_m$:

$$\mathbf{P}^T\hat{\mathbf{f}}(t) = \mathbf{P}^T\mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(t) = \mathbf{P}^T\mathbf{f}(t)$$

- How to select the *interpolated indices* $\wp_1, \ldots, \wp_m$?

"Discrete Empirical Interpolation Method (DEIM)"

Discrete Empirical Interpolation Method (DEIM)

INPUT: $\mathbf{u}_1, \ldots, \mathbf{u}_m \in \mathbb{R}^n$ (linearly independent)

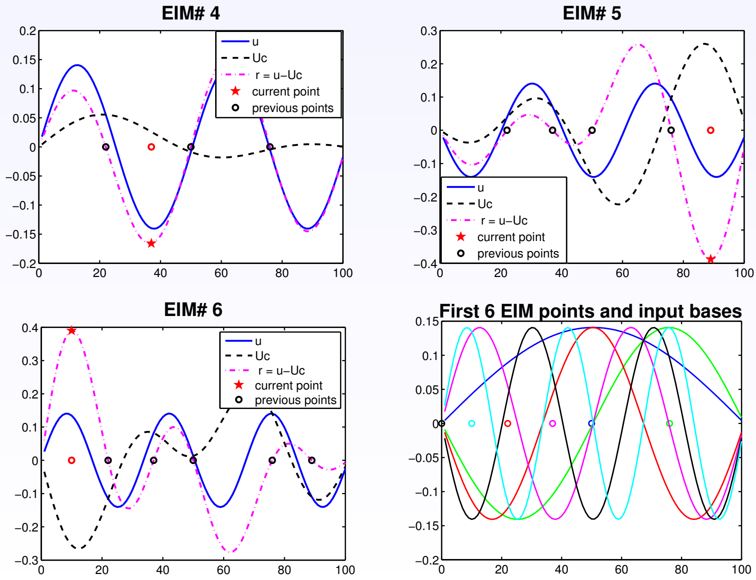
OUTPUT: $\vec{\wp} = [\wp_1, \ldots, \wp_m]$

- $[\rho, \ \wp_1] = max|\mathbf{u}_1|$

  $\mathbf{U} = [\mathbf{u}_1]$, $\vec{\wp} = [\wp_1]$,

- for $j = 2$ to m
  1. $\mathbf{u} \leftarrow \mathbf{u}_j$
  2. Solve $\mathbf{U}_{\vec{\wp}}\mathbf{c} = \mathbf{u}_{\vec{\wp}}$ for $\mathbf{c}$
  3. $\mathbf{r} = \mathbf{u} - \mathbf{U}\mathbf{c}$
  4. $[\rho, \ \wp_j] = max\{|\mathbf{r}|\}$
  5. $\mathbf{U} \leftarrow [\mathbf{U} \ \ \mathbf{u}]$, $\vec{\wp} \leftarrow \begin{bmatrix} \vec{\wp} \\ \wp_i \end{bmatrix}$

DEIM Point Selection



DEIM vs POD ERROR

$\hat{\mathbf{f}} = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}$

$$\boxed{\|\mathbf{f} - \hat{\mathbf{f}}\|_2 \leq \mathbf{C}^{EIM}\mathcal{E}_*.}$$

$\mathcal{E}_* = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}\|_2$ optimal 2-norm error (POD),

$\mathbf{C}^{EIM} = \|(\mathbf{P}^T U)^{-1}\|_2 \leq (1 + \sqrt{2n})^m$,

if $\{\mathbf{u}_i\}_{i=1}^m$ orthonormal.

Approximation of the Error Bound

For $\mu \in \mathcal{D}$, $\boxed{\|\mathbf{f}(\mu) - \hat{\mathbf{f}}(\mu)\|_2 \leq \mathbf{C}^{EIM} \mathcal{E}_*(\mu)}$

where

- $\mathcal{E}_*(\mu) = \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T)\mathbf{f}(\mu)\|_2$
- $\hat{\mathbf{f}}(\mu) = \mathbf{U}(\mathbf{P}^T\mathbf{U})^{-1}\mathbf{P}^T\mathbf{f}(\mu)$
- $\mathrm{Range}(\mathbf{U}) \approx \mathcal{M} = \mathrm{span}\{\mathbf{f}(\mu) | \mu \in \mathcal{D}\}$

$\Rightarrow$ **Must compute $\mathbf{f}(\mu)$ to obtain $\mathcal{E}_*(\mu)$ for the error bound.**

Snapshot matrix: $\mathbb{F} = [\mathbf{f}(\mu_1), \ldots, \mathbf{f}(\mu_{n_s})] \in \mathbb{R}^{n \times n_s}$,

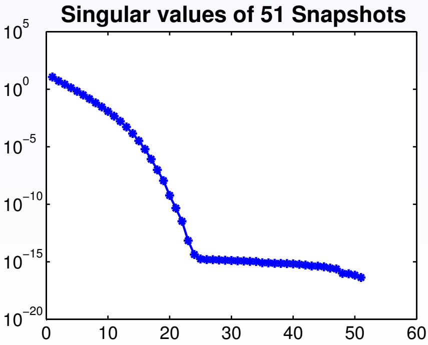
- Put $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{n \times m} \sim$ leading left sing. vects. of $\mathbb{F}$
  Then $\mathcal{E}_*(\mu) \lesssim \sigma_{m+1}$

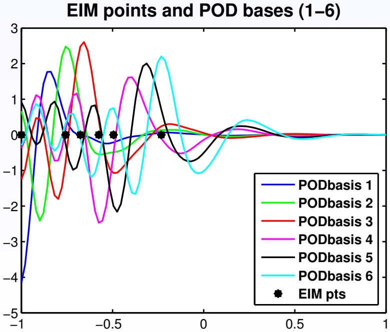
$$\boxed{\|\mathbf{f}(\mu) - \hat{\mathbf{f}}(\mu)\|_2 \lesssim \mathbf{C}^{EIM} \sigma_{m+1}.}$$

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

$$\mu \in [1,\pi]$$

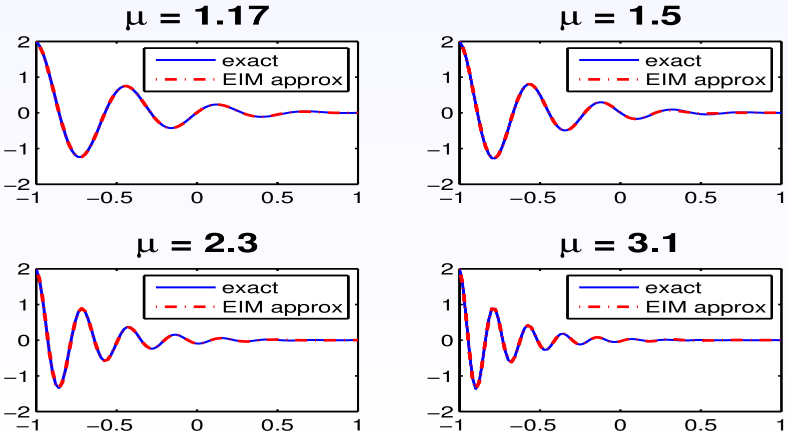
- $\mathbf{x} = [x_1,\ldots,x_n]^T$, x_i equidistant points in $[-1,1]$
- Using 51 snapshots uniformly selected from $[1,\pi]$ to construct POD basis



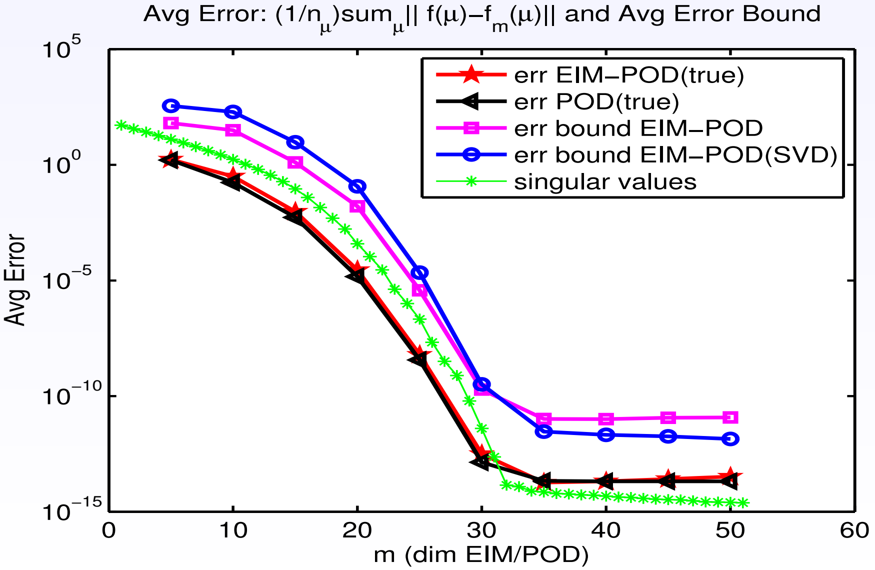


$DEIM : s(x;\mu) = (1-x)cos(3\pi\mu(x+1))e^{-(1+x)\mu}$

- **Full dim = 100, EIM dim = 10**
- Good approximation for arbitrary value of $\mu \in [1, \pi]$



Average Error Bound

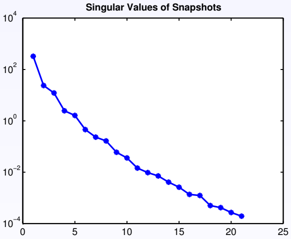


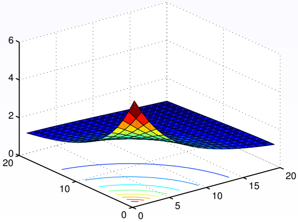
Discrete EIM for 2D

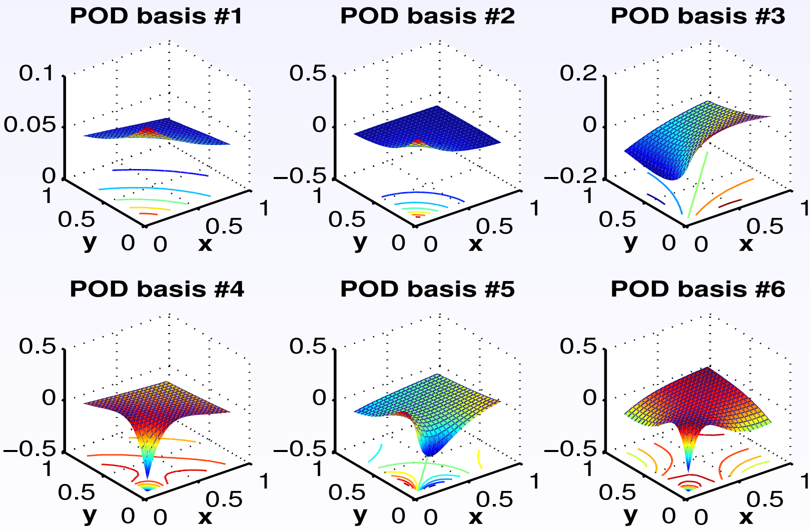
Consider a nonlinear parametrized function
$\mathbf{s} : \mathcal{D} \mapsto \mathbb{R}^{n_x \times n_y}$ defined by

$$\mathbf{s}(x_i, y_j; \mu) = \frac{1}{\sqrt{(x_i - \mu_1)^2 + (y_j - \mu_2)^2 + 0.1^2}},$$

- Spatial grid points $(x_i, y_j) \in \Omega = [0.1, 0.9]^2$ are equally spaced on Ω, for $i = 1, \ldots, n_x$ $j = 1, \ldots, n_y$.
- Parameter $\mu = (\mu_1, \mu_2) \in \mathcal{D} = [-1, -0.01]^2 \subset \mathbb{R}^2$.
- POD basis are constructed from 225 snapshots selected from equally-spaced grid points on parameter domain $\mathcal{D}$.







POD basis #1
POD basis #2
POD basis #3
POD basis #4
POD basis #5
POD basis #6
0.1
0.05
0
0.5
0
−0.5
0.2
0
−0.2
1
0.5
0
y
x

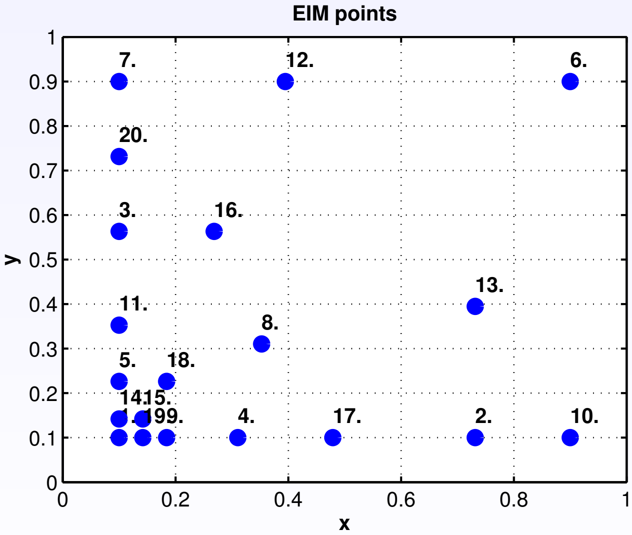


Figure: First 20 points selected by Discrete EIM from input POD basis

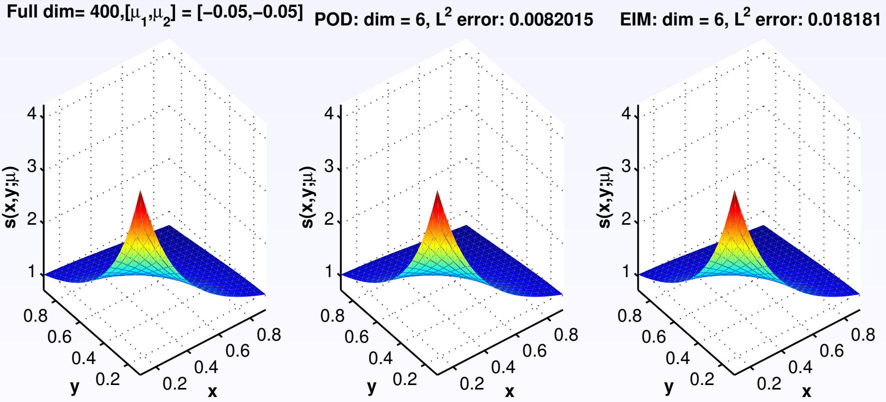
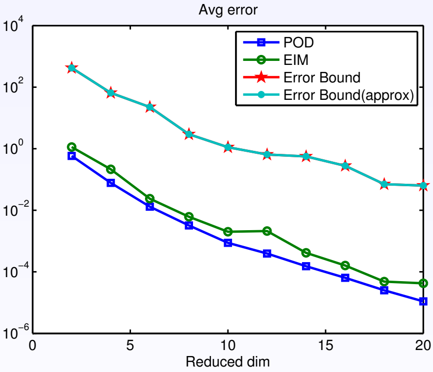
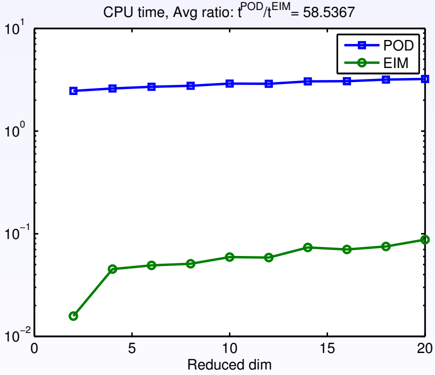


Figure: Compare the original nonlinear function (dim= 400) with POD and EIM approximations (dim=6) at parameter $\mu = (-0.05, -0.05)$.





- Avg err $= \frac{1}{n_\mu} \sum_{i=1}^{n_\mu} \|\mathbf{s}(\cdot;\mu^i) - \hat{\mathbf{s}}(\cdot;\mu^i)\|_2, \; n_\mu = 625.$
- $\mu^i = (\mu_1^i, \mu_2^i) \in \mathcal{D}.$
- Original dimension $= 400$.

DEIM Steps

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{F}(\mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0$$

1) Run trajectory and collect snapshots $\mathbb{Y} = [\mathbf{y}^1, \mathbf{y}^2, \ldots \mathbf{y}^m]$ perhaps with several different inputs and parameter values.

2) Truncate SVD of snapshots $\mathbb{Y}$ to get a POD basis for Trajectory

3) Collect nonlinear snapshots $\mathbb{F} = [\mathbf{F}(\mathbf{y}^1), \mathbf{F}(\mathbf{y}^2), \ldots, \mathbf{F}(\mathbf{y}^m)]$

4) Truncate SVD of nonlinear snapshots $\mathbb{F}$ to get another POD basis for the nonlinear term

5) Select DEIM interpolation points and approximate nonlinear term via collocation in the non-linear POD basis

6) Construct the nonlinear ROM from the reduced linear and nonlinear terms

General: *Complexity to Evaluate Component* $\mathbf{F}_{\wp_i}(\,\cdot\,)$

Note $\mathbf{y}_j \approx \mathbf{V}_k(j,:)\tilde{\mathbf{y}} \;\Rightarrow\; \mathbf{F}(\mathbf{y}) \approx \mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}})$

Hence $\mathbf{P}^T\mathbf{F}(\mathbf{y})$ is Approximately

$$\mathbf{P}^T\mathbf{F}(\mathbf{V}_k\tilde{\mathbf{y}}) = [F_{\wp_1}(\mathbf{V}_k\tilde{\mathbf{y}}), F_{\wp_2}(\mathbf{V}_k\tilde{\mathbf{y}}), \ldots, F_{\wp_m}(\mathbf{V}_k\tilde{\mathbf{y}})]^T \in \mathbb{R}^m.$$

Component

$$F_{\wp_i}(\mathbf{y}) = F_{\wp_i}([\mathbf{y}_{j_1}, \mathbf{y}_{j_2}, \ldots, \mathbf{y}_{j_{n_i}}]^T)$$

Depends Only on n_i Variables $\mathbf{y}_{j_1}, \mathbf{y}_{j_2}, \ldots, \mathbf{y}_{j_{n_i}}$

$$\textit{Dependency List}: \;\; \mathbf{j}_{\wp_i} = [j_1, j_2, \ldots, j_{n_i}]^T$$

Complexity to Evaluate $F_{\wp_i}(\mathbf{V}_k(\mathbf{j}_{\wp_i},:)\tilde{\mathbf{y}})$

is $n_i \times k$ Flops $\;+\;$ complexity of $F_{\wp_i}(\,\cdot\,)$

Sparse Evaluation via CSR Data Structure

Compressed Sparse Row:

- $irstart(i) \;=\;$ start of $\wp_i$-th row with $irstart(m+1) = n_{\vec{\wp}} + 1$.
- $jrow(irstart(i) : irstart(i+1)-1) \;=\; \mathbf{j}_{\wp_i} = [j_1, j_2, \ldots, j_{n_i}]^T$ (Dep. List component $\wp_i$)

$$
jrow \quad = \quad [\underbrace{\overset{\overset{irstart(1)}{\downarrow}}{\boxed{j_1^{\wp_1}}}, \ldots, j_{n_1}^{\wp_1}}_{\mathbf{j}_{\wp_1}}, \;\; \underbrace{\overset{\overset{irstart(2)}{\downarrow}}{\boxed{j_1^{\wp_2}}}, \ldots, j_{n_2}^{\wp_2}}_{\mathbf{j}_{\wp_2}}, \;\; \ldots, \;\; \underbrace{\overset{\overset{irstart(m)}{\downarrow}}{\boxed{j_1^{\wp_m}}}, \ldots, j_{n_m}^{\wp_m}}_{\mathbf{j}_{\wp_m}}]^T
$$

F- Evaluation Scheme:

for $i = 1 : m$
  $\mathbf{j}_{\wp_i} = jrow(irstart(i) : irstart(i+1)-1)$
  $\tilde{F}_{\wp_i}(\tilde{\mathbf{y}}) = F_{\wp_i}(\mathbf{V}_k(\mathbf{j}_{\wp_i}, :)\tilde{\mathbf{y}})$
end

The FitzHugh-Nagumo(F-N) Equations

Let $x \in [0, L], t \geq 0,$

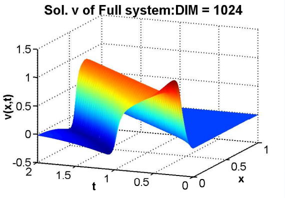
$$\begin{aligned} \varepsilon v_t(x,t) &= \varepsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) \\ w_t(x,t) &= bv(x,t) - \gamma w(x,t), \end{aligned}$$

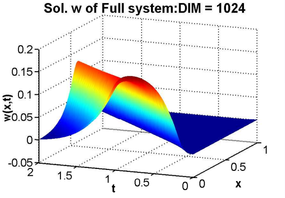
where $f(v) = v(v-0.1)(1-v)$ with initial conditions and boundary conditions:

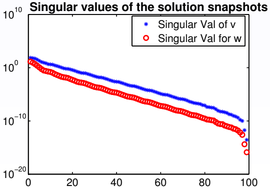
$$\begin{aligned} v(x,0) = \quad & 0, & w(x,0) = 0,\ x \in [0,L] \\ v_x(0,t) = \quad & -i_0(t), & v_x(L,t) = 0,\ t \geq 0 \end{aligned}$$

where $L = 1,\ \varepsilon = 0.015,\ b = 0.5,\ \gamma = 2,$
$i_0(t) = 50000t^3 exp(-15t),\ t \in [0,2].$

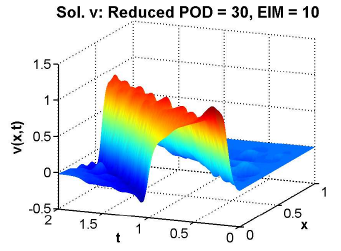
- Neural modeling: v = voltage variable, w= recovery variable.
- Simplified version of the Hodgkin-Huxley.

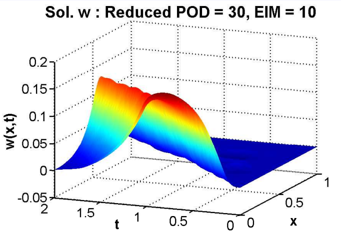






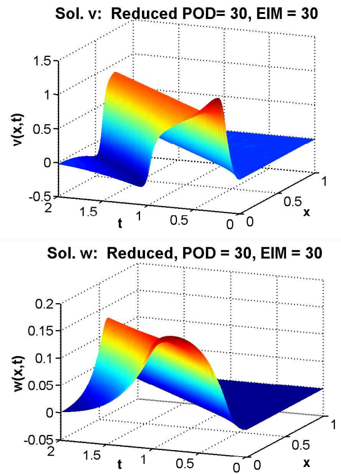
F-N: POD=30, EIM=10





click figure for movie

F-N: POD=30, EIM=30



click figure for movie

F-N: Average Relative Error

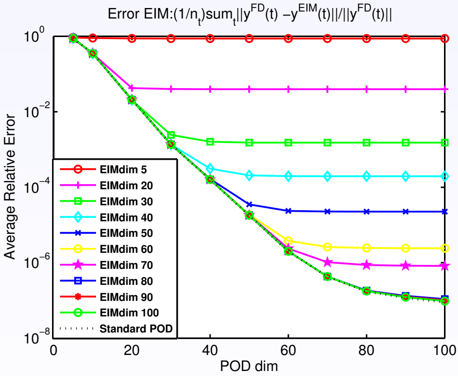


Figure: Error: EIM vs FULL

$\mathcal{E}_2 := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|y^{FD}(t_j)-y^{EIM}(t_j)\|_2}{\|y^{FD}(t_j)\|_2}.$

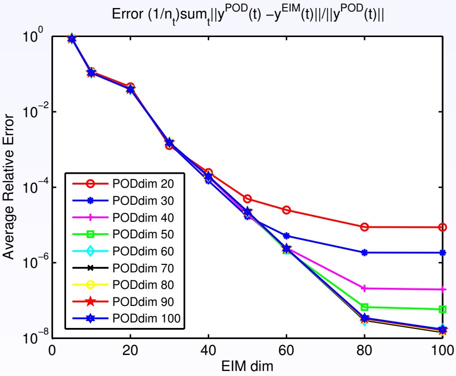


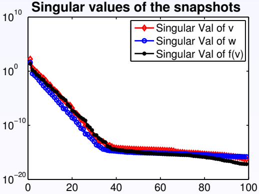
Figure: Error: EIM vs POD

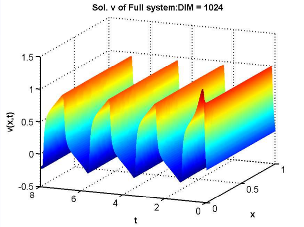
$\mathcal{E}_3 := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|y^{POD}(t_j)-y^{EIM}(t_j)\|_2}{\|y^{POD}(t_j)\|_2}.$

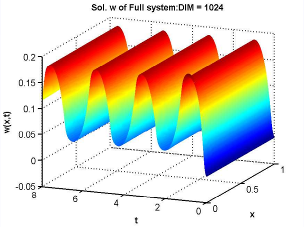
FitzHugh-Nagumo(FN) Limit Cycle

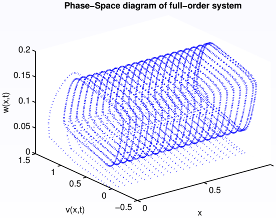
$$\begin{aligned} \varepsilon v_t(x,t) &= \varepsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) + {\color{red}c} \\ w_t(x,t) &= bv(x,t) - \gamma w(x,t) + {\color{red}c}, \quad {\color{red}c} = 0.05 \end{aligned}$$

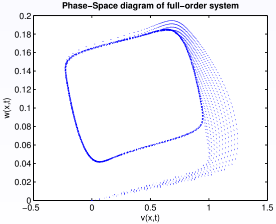
- Modeling cardiac electrical activity: $v =$ voltage , $w=$ recovery
- Periodic solutions with limit cycles.









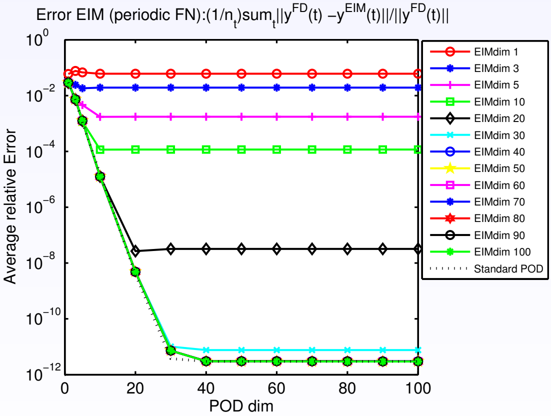


F-N Lim Cycle:

- Full dim = **1024** Reduced dim: POD=**5**/EIM=**5**

click figure for movie

F-N Lim Cycle: Avg Relative Error DEIM



Reduced Order Neural Modeling

Steve Cox Tony Kellems

LINEAR MODELS

Balanced Truncation Optimal $\mathcal{H}_2$

Nan Xiao and Derrick Roos Ryan Nong

Complex Model (Dim 160 K) $\rightarrow$ 20 variable ROM

NONLINEAR MODELS

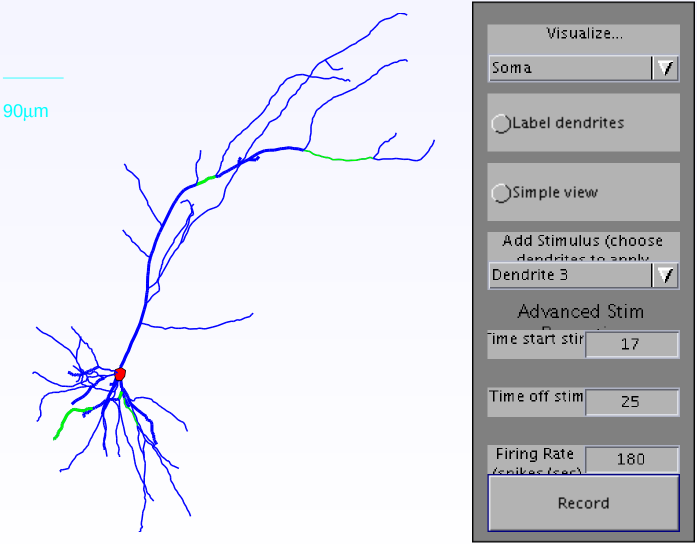
Empirical Interpolation (EIM) - Patera et al.

Saifon Chaturantabut T. Kellems

Nonlinear H-H Neuron Model (Dim 1198) $\rightarrow$ 30 variable ROM

Complex nonlinear behavior well approximated

Neuron Cell

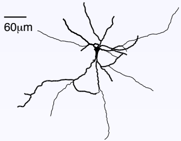


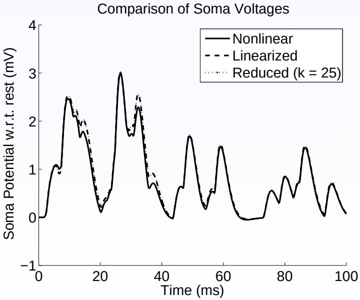
90µm
Visualize...
Soma
Label dendrites
Simple view
Add Stimulus (choose
Dendrite 3
Advanced Stim
ime start stir
17
Time off stim
25
Firing Rate
180
Record

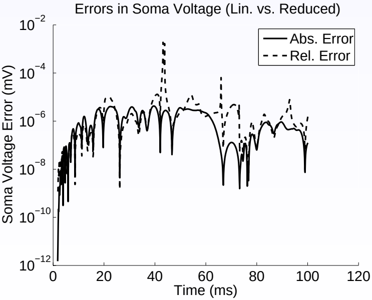
Linear ROM Results on Realistic Neuron

AR-1-20-04-A (Rosenkranz lab)

Full system size 6726 Reduced system size 25

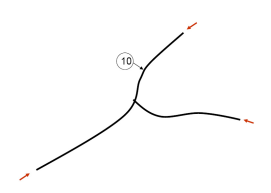


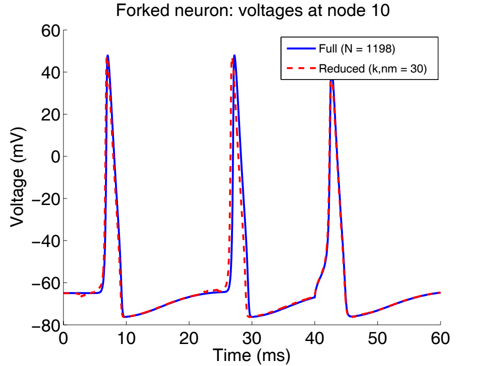




EIM Reduction of Hodgkin-Huxley Fiber

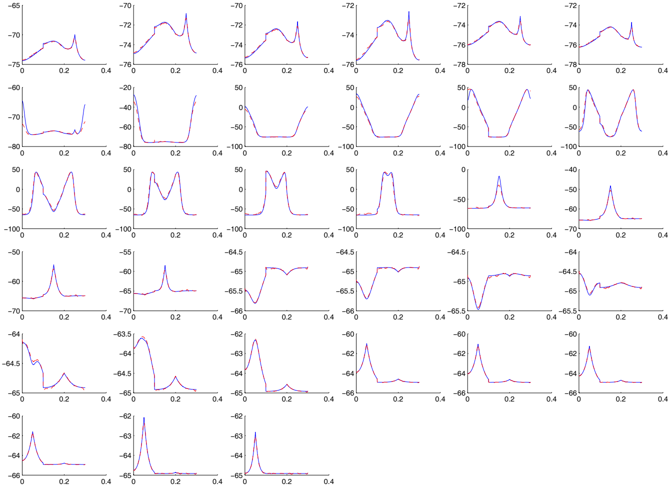
Three Inputs





EIM Reduction of HH : 3 inputs

Voltage Profile at Various Times (Full vs ROM)



Viscous Fingering: Horizontal Flow Through Porous Medium

Homogeneous 2D porous medium, Constant permeability K.
Incompressible Solvent injected at left: Uniform velocity U along x.
Evolution governed by conservation of mass, momentum, energy:

$$
\begin{aligned}
\nabla \cdot \mathbf{u} &= 0 \\
\nabla P &= -\frac{\mu}{K}\mathbf{u} \\
\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c &= D\nabla^2 c + f(c), \\
\rho c_p \left[\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T\right] &= \kappa \nabla^2 c + (-\triangle H) f(c),
\end{aligned}
$$

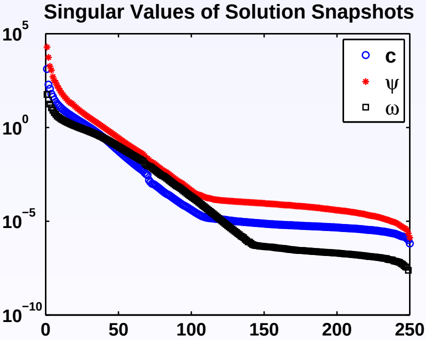
$\mathbf{u} \in \mathbb{R}^2$ is velocity with x and y components;
$P =$ pressure; $c =$ concentration; $T =$ temperature;
$f = -c(k_a + k_r c)(c - c_0)$: reaction rate;
$\mu =$ viscosity, (depends on c and T); K, D constants.

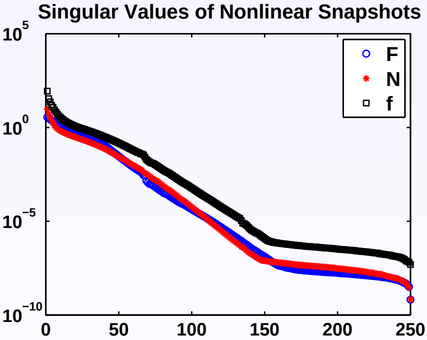
Miscible Flow Visualization

- Full dim = **15,000** Reduced dim: DEIM=**40**

click figure for movie

Error and Comparison of Computing Time





Dimension	Avg Rel Error of c	CPU time (sec)
Full 15000 (FD)	-	2.138×10^3
POD40	4.066×10^{-4}	2.442×10^2
POD40/DEIM40	2.045×10^{-3}	1.275

Error and Comparison of Computing Time

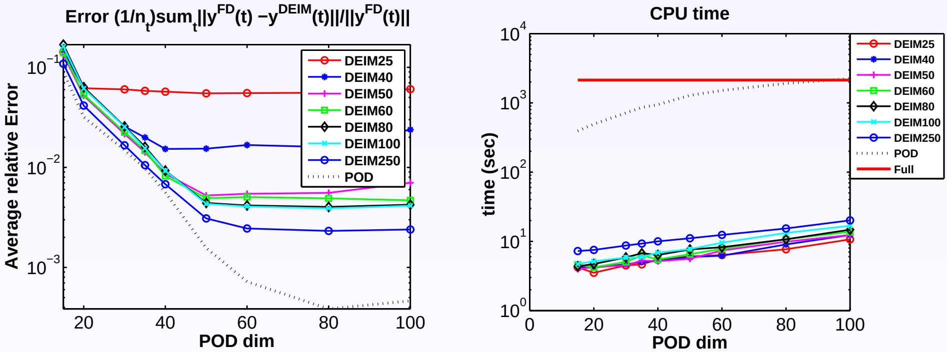


Figure: Average relative errors of the POD-DEIM reduced system compared with (left) and the cpu time of the full system, POD reduced system, and POD-DEIM reduced system(right).

Summary

- **CAAM TR09-05** DEIM, S. Chaturantabut and D.C. Sorensen
- **TR09-12** Neural Modeling, A. R. Kellems, S. Chaturantabut,
  D. C. Sorensen, and S. J. Cox

  http://www.caam.rice.edu/ sorensen/

Empirical Gramian Based Model Reduction: (POD)

Efficient Approximation of Nonlinear Term (DEIM)
Complexity of nonlinear term proportional to no. reduced vars.

DEIM Point Selection Algorithm

Demonstrated Accuracy and Efficiency
Nonlinear MOR via DEIM

Neural Modeling - Single Cell ROM $\Rightarrow$ Many Interactions
Example of important class of problems