Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods

Saifon Chaturantabut

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

November 20, 2008
Introduction

- Model Problem

Methods

- Overview of Techniques
- Proper Orthogonal Decomposition (POD)
- Nonlinear Approximation
  - Empirical Interpolation Method (EIM)

Numerical Results

- Burgers’ Equation
- FitzHugh-Nagumo (FN) System

Conclusions

Future Work
Outline

1. Introduction
   - Model Problem

2. Methods
   - Overview of Techniques
   - Proper Orthogonal Decomposition (POD)
   - Nonlinear Approximation
     - Empirical Interpolation Method (EIM)

3. Numerical Results
   - Burgers’ Equation
   - FitzHugh-Nagumo (FN) System

4. Conclusions

5. Future Work
Model Problem: Unsteady 1D Burgers’ Equation

For \( x \in [0, 1] \), \( t \geq 0 \),

\[
\frac{\partial}{\partial t} y(x, t) - \nu \frac{\partial^2}{\partial x^2} y(x, t) + \frac{\partial}{\partial x} (s(y(x, t))) = 0, \quad s(y(x, t)) = \frac{y(x, t)^2}{2}
\]

\[
y(0, t) = y(1, t) = 0, \quad \forall t \geq 0
\]

\[
y(x, 0) = y_0(x), \quad \forall x \in [0, 1]
\]

- Commonly used model problem to test algorithm for Navier-Stokes type equation
  - Simple nonlinearity that is sufficient to test algorithm
- Used in gas dynamics modeling, turbulence modeling and control theory
- Numerical solution obtained from discretization is often used
**Discretized system**
Use FE basis to construct system of order $N$:

$$M_h \frac{d}{dt} y(t) + \nu K_h y(t) - N_h(y(t)) = 0 \Rightarrow y_h(x, t)$$

**Problem:** High Accuracy $\iff$ High Complexity ($N$ LARGE)  
$\Rightarrow$ Expensive in Computation  
$\Rightarrow$ Impossible to do simulation in reasonable time

**Goal:**  
Construct a *reduced* system of order $k$,

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

such that
- $k \ll N$
- $\tilde{y}_h(x, t) \approx y_h(x, t)$
- the *reduced* system can be solved *efficiently*
Outline

1. Introduction
   - Model Problem

2. Methods
   - Overview of Techniques
   - Proper Orthogonal Decomposition (POD)
   - Nonlinear Approximation
     - Empirical Interpolation Method (EIM)

3. Numerical Results
   - Burgers’ Equation
   - FitzHugh-Nagumo (FN) System

4. Conclusions

5. Future Work
Overview

Nonlinear PDE

\[ \downarrow \]

FULL Discretized System (ODE)
\[ \text{dim} = N \]

\[ \downarrow \]

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

\[ \downarrow \]

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

\[ \downarrow \]

POD-Galerkin

\[ \leftarrow \leftarrow \] FE-Galerkin

\[ \downarrow \]

Solve ODE

\[ \leftarrow \leftarrow \] POD

\[ \leftarrow \leftarrow \] POD-Galerkin

REDUCED Discretized System (ODE)

Linear Term: \( \text{dim} = k \ll N \)

Non-Linear Term: \( \text{dim} \sim N \)

Nonlinear Approx

\[ \leftarrow \leftarrow \] EIM

\[ \leftarrow \leftarrow \] Precomputing

(i) Memory Saving
(ii) Accuracy

REDUCED Discretized System (ODE)

Linear Term: \( \text{dim} = k \ll N \)

Non-Linear Term: \( \text{dim} = m \ll N \)

(iii) Efficiency
(Time Saving)
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 a Hilbert space (i.e., Complete Inner Product Space).
- Given $y_1, \ldots, y_n \in X$. Define $y_i \equiv \text{snapshot } i$, $\forall i$. Let $\mathcal{Y} \equiv \text{span}\{y_1, y_2, \ldots, y_n\} \subset X$.
- 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:

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

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

- Solve by SVD
Direct POD

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

$$M_h \frac{d}{dt} y(t) + \nu K_h y(t) - N_h(y(t)) = 0$$

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

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

where the nonlinear term:

$$\tilde{N}(\tilde{y}(t)) = U^T N_h(U\tilde{y}(t))$$

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

Recall the problem from **Direct POD**:
\[
\tilde{\mathbf{N}}(\tilde{y}(t)) = U^T \mathbf{N}_h(U\tilde{y}(t)) \quad \text{with dimensions } k \times N
\]

\[
\mathbf{N}_h(U\tilde{y}(t)) \quad \text{with dimensions } N \times 1
\]

**WANT**: \[
\tilde{\mathbf{N}}(\tilde{y}(t)) \leftarrow C \hat{\mathbf{N}}(\tilde{y}(t)) \quad \text{with dimensions } k \times m \quad \text{and } m \times 1
\]

\[
\hat{\mathbf{N}}(\tilde{y}(t)) \quad \text{with dimensions } k \times N
\]

\[
k, m \ll N \quad \text{Independent of } N
\]

2 Approaches:
- **Precomputing Technique**: Simple nonlinear
- **Empirical Interpolation Method (EIM)**: General nonlinear
Desired Form of Nonlinear Approximation

From Burgers’ equation, \( s(y(x, t)) = \frac{y(x, t)^2}{2} \) generates nonlinear terms \( \mathbf{N}_h, \tilde{\mathbf{N}} \).

\[
\Phi(x) = \Psi(x) \mathbf{U}_k, \\
\text{where} \begin{cases}
\text{FE basis:} & \Psi(x) \equiv [\psi_1(x), \ldots, \psi_N(x)], \\
\text{Reduced basis:} & \Phi(x) \equiv [\phi_1(x), \ldots, \phi_k(x)], \\
\text{Projection:} & \mathbf{U}_k \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{N \times k}.
\end{cases}
\]

Note:

\[
\tilde{\mathbf{N}}(\tilde{y}(t)) = \mathbf{U}_k^T \mathbf{N}_h(\mathbf{U}_k \tilde{y}(t)) \iff \begin{cases}
\mathbf{N}_h(y(t)) = \int_{\Omega} \psi'(x)^T s(\psi(x)y(t)) \, dx \\
\tilde{\mathbf{N}}(\tilde{y}(t)) = \mathbf{U}_k^T \int_{\Omega} \psi'(x)^T s(\psi(x)\mathbf{U}_k \tilde{y}(t)) \, dx \end{cases}
\]

Want: \( \hat{s}(x, t) \approx s(\Psi(x)\mathbf{U}_k \tilde{y}(t)) \) in the form that separates \( x \) and \( t \).
Interpolation Method

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

$$\hat{s}(x; \mu) = \sum_{j=1}^{m} q_j(x) \beta_j(\mu),$$

where $\text{span}\{q_j\}_{j=1}^{m} \simeq \mathcal{M}^s \equiv \{s(\cdot; \mu) : \mu \in \mathcal{D}\}$ and $\beta_j(\mu)$ is specified from the interpolation points $\{z_j\}_{j=1}^{m}$, in $\Omega$:

$$s(z_i; \mu) = \sum_{j=1}^{m} q_j(z_i) \beta_j(\mu),$$

for $i = 1, \ldots, m$. 
Interpolation Method (cont')

Coefficient Approximation: Matrix Form

- Given
  Basis: \( Q(x) \equiv [q_1(x), \ldots, q_m(x)] \)
  Interpolation Points: \( z \equiv [z_1, \ldots, z_m]^T \in \Omega^m \)

- Let
  Coefficient: \( \beta(\mu) \equiv [\beta_1(\mu), \ldots, \beta_m(\mu)]^T \in \mathbb{R}^m \)
  Function at \( z \): \( s(z; \mu) \equiv [s(z_1; \mu), \ldots, s(z_m; \mu)]^T \in \mathbb{R}^m \)

\[
Q(z) = \begin{bmatrix}
q_1(z_1) & q_2(z_1) & \cdots & q_m(z_1) \\
q_1(z_2) & q_2(z_2) & \cdots & q_m(z_2) \\
\vdots & \vdots & \ddots & \vdots \\
q_1(z_m) & q_2(z_m) & \cdots & q_m(z_m)
\end{bmatrix} \in \mathbb{R}^{m \times m},
\]

\[
\begin{cases}
\hat{s}(x; \mu) = Q(x)\beta(\mu) \\
\hat{s}(z; \mu) = s(z; \mu)
\end{cases} \implies \beta(\mu) = (Q(z))^{-1}s(z; \mu)
\]

\[
\hat{s}(x; \mu) = Q(x)(Q(z))^{-1}s(z; \mu)
\]
How to apply interpolation approx to nonlinear term in the PDE?

- **Interpolation Approximation:**

\[
s(\psi(x)U_k\tilde{y}(t)) \simeq \hat{s}(x, t) = Q(x)(Q(z))^{-1}s(z, t),
\]

- Approximation for the Nonlinear term:

\[
\tilde{N}(\tilde{y}(t)) = \left[ U_k^T \int_\Omega \psi'(x)^T s(\psi(x)U_k\tilde{y}(t)) dx \right]_{N \times 1} \simeq \left[ U_k^T \left( \int_\Omega \psi'(x)^T Q(x) dx \right) (Q(z))^{-1} \right]_{N \times 1} s(z, t) = \tilde{N}(\tilde{y}(t))
\]

- Finally, reduced system of order \(k\) and \(m\) (independent of \(N\)):

\[
\tilde{M} \frac{d}{dt} \tilde{y}(t) + \nu \tilde{K}\tilde{y}(t) - C\tilde{N}(\tilde{y}(t)) = 0,
\]
Empirical Interpolation Method (EIM) [Patera; 2004]

EIM Algorithm: Given a set of basis functions \{\xi_i\}_{i=1}^m, the set of \( m \) EIM interpolation points \( z = [z_1, \ldots, z_m]^T \) is constructed as follows.

EIM steps

1. Set \( z_1 = \text{arg ess sup}_{x \in \Omega} |\xi_1(x)| \).
   \[
   q_1(x) = \frac{\xi_1(x)}{\xi_1(z_1)}; \quad Q^1(z_1) = q_1(z_1) = 1.
   \]

2. For \( L = 2, \ldots, m \),
   1. Solve \( \rho^{L-1} \) from :
      \[
      Q^{L-1}(z^{L-1})\rho^{L-1} = \xi_L(z^{L-1}),
      \]
   2. Define
      \[
      r_L(x) = \xi_L(x) - Q^{L-1}(x)\rho^{L-1},
      \]
   3. Set
      \[
      z_L = \text{arg ess sup}_{x \in \Omega} |r_L(x)|
      \]
   4. Set
      \[
      q_L(x) = \frac{r_L(x)}{r_L(z_L)}
      \]
**Empirical Interpolation Method (EIM)**[Patera; 2004]

**EIM steps: Given basis \( \{ \xi_i \}_{i=1}^m \)**

1. **Set** \( z_1 = \arg \text{ess sup}_{x \in \Omega} |\xi_1(x)|. \)
   \[ q_1(x) = \frac{\xi_1(x)}{\xi_1(z_1)}; \]
   \[ Q^1(z^1) = q_1(z_1) = 1. \]

2. **For** \( L = 2, \ldots, m, \)
   
   1. **Solve** \( \rho^{L-1} \) from:
   \[ Q^{L-1}(z^{L-1})\rho^{L-1} = \xi_L(z^{L-1}), \]

3. **Define**
   \[ r_L(x) = \xi_L(x) - Q^{L-1}(x)\rho^{L-1}, \]

4. **Set**
   \[ z_L = \arg \text{ess sup}_{x \in \Omega} |r_L(x)| \]
   \[ q_L(x) = \frac{r_L(x)}{r_L(z_L)} \]

**Note:** \( \text{span}\{q_1, \ldots, q_L\} = \text{span}\{\xi_1, \ldots, \xi_L\} \)

**Ex:** Input \( \{\xi_L\}_{L=1}^m \) are eigenvectors of discrete Laplacian (Dr. Sorensen)

---

**EIM # 1**

- **Original basis**
- **EIM basis (normalized)**
- **EIM basis (before normalized)**
- **EIM point (current)**

**EIM # 3**

- **Original basis**
- **EIM basis (normalized)**
- **EIM basis (before normalized)**
- **EIM point (current)**
- **EIM points (previous)**
Empirical Interpolation Method (EIM)[Patera; 2004]

EIM steps: Given basis \{\xi_i\}_{i=1}^m

1. Set \[ z_1 = \arg \text{ess sup}_{x \in \Omega} |\xi_1(x)|. \]
   \[ q_1(x) = \frac{\xi_1(x)}{\xi_1(z_1)}; \]
   \[ Q^1(z_1) = q_1(z_1) = 1. \]

2. For \( L = 2, \ldots, m, \)
   
   - Solve \( \rho_{L-1} \) from:
     \[ Q^{L-1}(z^{L-1}) \rho_{L-1} = \xi_L(z^{L-1}), \]

3. Define
   \[ r_L(x) = \xi_L(x) - Q^{L-1}(x) \rho_{L-1}, \]

4. Set
   \[ z_L = \arg \text{ess sup}_{x \in \Omega} |r_L(x)|. \]
   \[ q_L(x) = \frac{r_L(x)}{r_L(z_L)} \]

Note: \( \text{span}\{q_1, \ldots, q_L\} = \text{span}\{\xi_1, \ldots, \xi_L\} \)
Ex: Input \( \{\xi_L\}_{L=1}^m \) are eigenvectors of discrete Laplacian (Dr. Sorensen)
Ex: EIM for a nonlinear function

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

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

- Using 51 snapshots uniform selected from \([1, \pi]\) to construct POD basis
- Good approximation for arbitrary value of \( \mu \in [1, \pi] \)
Ex: EIM for a nonlinear function (cont'):
\[ s(x; \mu) = (1 - x) \cos(3\pi \mu (x + 1)) e^{-(1+x)\mu} \]
Outline

1. Introduction
   - Model Problem

2. Methods
   - Overview of Techniques
   - Proper Orthogonal Decomposition (POD)
   - Nonlinear Approximation
     - Empirical Interpolation Method (EIM)

3. Numerical Results
   - Burgers’ Equation
   - FitzHugh-Nagumo (FN) System

4. Conclusions

5. Future Work

Saifon Chaturantabut
Dimension Reduction for Unsteady Nonlinear PDEs via EIM
Plots of Numerical Solutions from 3 Approaches

- Direct POD
- Precomputed POD
- EIM-POD

Singular values of the Snapshots

Figure: SVD

Figure: Direct POD
Introduction

Methods

Results

Conclusions

Future Work

extras

Burgers’ Eqn

FitzHugh-Nagumo system

Figure: Precomputed POD

Figure: EIM-POD
**ACCURACY vs. COMPLEXITY**

Relative 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)

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)
Application I: The FitzHugh-Nagumo (FN) System

Let \( x \in [0, L], \ t \geq 0, \)

\[
\varepsilon v_t(x, t) = \varepsilon^2 v_{xx}(x, t) + f(v(x, t)) - w(x, t) \\
w_t(x, t) = bv(x, t) - \gamma w(x, t),
\]

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

\[
v(x, 0) = 0, \quad w(x, 0) = 0, \quad x \in [0, L] \\
v_x(0, t) = -i_0(t), \quad v_x(L, t) = 0, \quad t \geq 0
\]

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

- Neuron modeling: \( v = \) voltage variable, \( w = \) recovery variable.
- Simplified version of the Hodgkin-Huxley.
Application I: EIM-POD Reduced System (POD=30, EIM=10)

Sol. \( v(x,t) \) of reduced System (POD–EIM): dim(POD) = 30, dim(EIM) = 10

Sol. \( w(x,t) \) of reduced System (POD–EIM): dim(POD) = 30, dim(EIM) = 10
Application I: EIM-POD Reduced System (POD=30, EIM=30)

Sol. $v$ of reduced System (POD–EIM): \( \text{dim}(\text{POD}) = 30, \text{dim}(\text{EIM}) = 30 \)

Sol. $w$ of reduced System (POD–EIM): \( \text{dim}(\text{POD}) = 30, \text{dim}(\text{EIM}) = 30 \)
**Application I: Average Relative Error**

**Figure:** EIM-POD approach:

\[
E_2 := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|y^{FD}(t_j) - y^{EIM}(t_j)\|_2}{\|y^{FD}(t_j)\|_2}
\]

**Figure:** EIM-POD approach vs (corresponding) POD approach (same dimension for POD but without EIM)

\[
E_3 := \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{\|y^{POD}(t_j) - y^{EIM}(t_j)\|_2}{\|y^{POD}(t_j)\|_2}
\]
Application II: The FitzHugh-Nagumo (FN) System

Let \( x \in [0, L], \ t \geq 0, \)

\[
\varepsilon v_t(x, t) = \varepsilon^2 v_{xx}(x, t) + f(v(x, t)) - w(x, t) + c \]

\[
w_t(x, t) = b v(x, t) - \gamma w(x, t) + c,
\]

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

\[
v(x, 0) = 0, \quad w(x, 0) = 0, \quad x \in [0, L] \\
v_x(0, t) = -i_0(t), \quad v_x(L, t) = 0, \quad t \geq 0
\]

where \( L = 1, \varepsilon = 0.015, b = 0.5, \gamma = 2, \)

\( i_0(t) = 50000t^3 \exp(-15t), c = 0.05, t \in [0, 2]. \)

- Modeling in cardiac electrical activity: \( v = \) voltage variable, \( w = \) recovery variable.
- Periodic solutions with limit cycles.

**Figure:** The singular values of the 100 snapshots of \( v, w, \) and \( f(v) \)
Application II: EIM-POD Reduced System (POD=5, EIM=1)
Application II: EIM-POD Reduced System (POD=5, EIM=5)
Application II: Avg Relative Error for EIM-POD approach

Error EIM (periodic FN): \( \frac{1}{n_x} \sum_t ||y^{FD}(t) - y^{EIM}(t)||/||y^{FD}(t)|| \)

![Graph showing average relative error for EIM-POD approach](image-url)
Outline

1 Introduction
   - Model Problem

2 Methods
   - Overview of Techniques
   - Proper Orthogonal Decomposition (POD)
   - Nonlinear Approximation
     - Empirical Interpolation Method (EIM)

3 Numerical Results
   - Burgers’ Equation
   - FitzHugh-Nagumo (FN) System

4 Conclusions

5 Future Work
Conclusions

- Unique Contribution: Clear description of the EIM
  ⇒ Successful Implementation of EIM with POD
- EIM is comparable to widely accepted methods, such as precomputing technique
- the reduced system from EIM-POD approach preserves the asymptotic behavior of the original full order system
- The error from the EIM-POD approach depends both from EIM and POD.
  ⇒ flat tail in error plots for each dim of EIM
  ⇒ bounded below by the POD approach
- The results suggest that EIM with POD basis is a promising model reduction technique for general parabolic of type convection-diffusion nonlinear PDEs.
Outline

1 Introduction
   - Model Problem

2 Methods
   - Overview of Techniques
   - Proper Orthogonal Decomposition (POD)
   - Nonlinear Approximation
     - Empirical Interpolation Method (EIM)

3 Numerical Results
   - Burgers’ Equation
   - FitzHugh-Nagumo (FN) System

4 Conclusions

5 Future Work

Saifon Chaturantabut
Dimension Reduction for Unsteady Nonlinear PDEs via EIM
Future Work

- Extending to higher dimensions (2D and 3D).
- Applying and performing the comparative study for the Best Point Interpolation Method (BPIM).
- Improving/Extending the existing error bound estimation for EIM.
- Developing software based on the EIM-POD procedure for constructing reduced order systems.
- Constructing a reduced order system for a nonlinear model by employing the information from its corresponding linear model.
Extending to higher dimensions (2D and 3D)

- Many realistic models are defined on physical domains in 2D/3D.

- Framework of the EIM algorithm is applicable to parametrized PDEs defined on any open and bounded physical domain $\Omega \subset \mathbb{R}^d$ for $d = 1, 2, 3$.

- $\Rightarrow$ Analogous, but more complicated, procedures can be used.
Applying and performing the comparative study for the Best Point Interpolation Method (BPIM)

- BPIM: optimization process (error minimization).
- EIM: greedy selection process.

⇒ BPIM gives higher *accuracy* but more *expensive* than EIM.
- BPIM does not improve much accuracy in practice

**GOAL:** Compare the *gain in accuracy* and the *loss of computational efficiency* from using BPIM instead of EIM.
Improving/Extending the existing error bound estimation for EIM

- EIM for FE [Patara, 2004]
  - *A priori* error bound (based on *Lebesgue constant*): loose bound.
  - *A posteriori* error bound: (i) rigorous, (ii) sharp in practice, but (iii) applicable under restricted assumption.

- EIM for FD [Sorensen, 2008]
  - *A posteriori* error bound: (i) rigorous (ii) efficient (iii) generalized to any problem that EIM is applicable.
  - Defined inductively on the iterative steps for selecting the EIM interpolation points.
  - The derivation gives an insight into EIM selection process.

- Extension of *a posteriori* error in FD case to FE case is possible but not straightforward due to the continuity of the spatial domain of the FE systems.
Let $\{u_i\}_{i=1}^{m}$ be orthonormal basis, and

- $\bar{M} = \bar{P}^T \bar{U},$ where $\bar{P}$, $\bar{U}$ are from previous iterations in EIM,
- $U = [\bar{U} \quad u]$, $p = e_{j_{\text{max}}}$, $u$ are from current iteration of EIM,
- $\alpha^T = p^T \bar{U} \bar{M}^{-1},$
- $C_{\text{EIM}} = (\max\{\|\bar{M}^{-1}\|_1, \rho^{-1}(1 + \|\bar{M}^{-1}\bar{P}^T u\|_1)\}) (1 + \|\alpha\|_1),$
- $\|w\|_2 = \|(I - UU^T)f\|_2$ is the residual of the best 2-norm approximation to $f$ from the space $\text{Range}(U)^a$.

Then

$$\|f - \hat{f}\|_2 \leq C_{\text{EIM}} \|w\|_2.$$
Software based on the EIM-POD procedure: Recall...

Nonlinear PDE

\[ \downarrow \]

FULL Discretized System (ODE)

\[ \text{dim} = N \]

\[ \downarrow \]

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

\[ \downarrow \]

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

\[ \downarrow \]

REduced Discretized System (ODE)

Linear Term: \( \text{dim} = k \ll N \)

Non-Linear Term: \( \text{dim} \sim N \)

\[ \leftarrow \rightarrow \leftarrow \rightarrow \]

(i) Memory Saving

(ii) Accuracy

Nonlinear Approx - EIM - Precomputing

\[ \leftarrow \rightarrow \leftarrow \rightarrow \]

(iii) Efficiency

(Time Saving)
Challenging issues in developing this software

1. How to develop this software so that users can use with minimum knowledge inside procedure

2. Generalization of the procedure to a broad class of parametrized nonlinear systems both from FE and FD discretizations, e.g.
   - Selecting snapshots,
   - Constructing a projection basis and interpolation points,
   - Computing coefficient function approximations.

3. Incorporating automatic order selection (selection of the reduced dimension) based on error analysis.

4. Integrating with the PDE solvers (e.g. MATLAB) and FEM (finite element method) softwares (e.g. Matlab PDE Toolbox, freeFEM®, CosmosWorks®)
Constructing a reduced order system for a nonlinear model by employing the information from its corresponding linear model

Motivations:

- Some physical system can be modeled by
  - Linear system as a simplified model;
  - Nonlinear system as a realistic model.
- Too expensive to generate snapshots from original full-order system.
- Linear system is generally easier to solve and may have analytic solution.
- E.g. Neuron modeling and Polymer dynamics modeling
Modeling for polymer dynamics: Bead-Spring Model

Define

$$\hat{Q}(t) = \left[ \hat{Q}_1^x(\hat{t}), \hat{Q}_1^y(\hat{t}), \hat{Q}_1^z(\hat{t}), \ldots, \hat{Q}_N^x(\hat{t}), \hat{Q}_N^y(\hat{t}), \hat{Q}_N^z(\hat{t}) \right]^T \in \mathbb{R}^{3N}.$$ 

$$d\hat{Q}(\hat{t}) = d\hat{t} \left[ K\hat{Q}(\hat{t}) - \alpha A\hat{F}(\hat{Q}(\hat{t})) \right] + d\hat{W}(\hat{t}),$$

where

$$\hat{F}(\hat{Q}) = \begin{bmatrix} Z(\hat{Q}_1)\hat{Q}_1 \\ Z(\hat{Q}_2)\hat{Q}_2 \\ \vdots \\ Z(\hat{Q}_N)\hat{Q}_N \end{bmatrix}, \quad Z(\hat{Q}_i) = \begin{cases} 1 & :\text{Linear} \\ \frac{1}{1 - |\hat{Q}_i|^2/b} & :\text{Nonlinear.} \end{cases}$$

$$|\hat{Q}_i|^2 = (\hat{Q}_i^x)^2 + (\hat{Q}_i^y)^2 + (\hat{Q}_i^z)^2.$$

- \(N_b\): number of beads.
- \(N = N_b - 1\): number of spring vectors \(\hat{Q}_i\).
Reduced order system

- **Linear Model:**

  1. Linear Model: \( \dot{\tilde{Q}}(t) = d\hat{t} \left( K - \alpha A \right) \tilde{Q}(t) + d\tilde{W}(t) \Rightarrow \text{Schur } B = U G U^T. \)

  2. Analytic solution of the linear model: *Rouse Modes*

  Projection matrix of dimension \(3k\): \( U_k = \begin{bmatrix} \tilde{u}_1 & \ldots & \tilde{u}_{3k} \end{bmatrix} \in \mathbb{R}^{3N \times 3k}. \)

- **Nonlinear Model**

  1. **LINEAR PART:** use projection basis from linear model

  \[
  d\tilde{Q}(t) = d\hat{t} \begin{bmatrix} \tilde{Q}(t) \end{bmatrix}_K - \alpha \begin{bmatrix} \tilde{Q}(t) \end{bmatrix}_F(U_k \tilde{Q}(t)) + \begin{bmatrix} \tilde{W}(t) \end{bmatrix}_K d\tilde{W}(t)
  \]

  2. **NONLINEAR PART:** use EIM for each coordinate \(x, y, \) and \(z\) with input basis constructed from snapshots of linear model

  \[
  \alpha U_k^T A \tilde{F}(U_k \tilde{Q}(t)) \approx C F^{EIM}(\tilde{Q}(t)).
  \]
Numerical Results (Nb = 10): Wi=10

**Figure: Avg viscosity**

**Figure: Avg springs’ length**
Future work for polymer modeling

- Extend to system with higher Weissenberg number (a major numerical difficulty in the computation of polymeric fluid dynamics)
- Extend to larger systems: more number of beads
- Extend to FE discretized system
QUESTION ?
DETAILS FOR POLYMER
Reduced order system

1. **LINEAR PART**: use projection basis from
   - **Linear Model**:
     - From *linear model*: \( B := K - \alpha A \in \mathbb{R}^{3N \times 3N} \)
     - Schur decomposition of \( B \): \( B = UGU^T. \)
   - **Analytic solution of the linear model** \( \rightsquigarrow \) *Rouse Modes*:
     For each coordinate \( x, y, \) and \( z \), the \( j^{th} \) Rouse mode of the \( \ell^{th} \) spring is \( \tilde{u}_{\ell j} = \sin(\ell \pi \frac{t}{N_b}) \), for \( \ell = 1, \ldots, N \), and \( j = 0, \ldots, N \).

   Projection matrix of dimension \( 3k \): \( U_k = \begin{bmatrix} u_{i_1} & \ldots & u_{i_3k} \end{bmatrix} \in \mathbb{R}^{3N \times 3k} \).

   \[
   d\tilde{Q}(t) = dt \begin{bmatrix} U_k^T K U_k \tilde{Q}(t) - \alpha U_k^T \hat{A}F(U_k \tilde{Q}(t)) \end{bmatrix} + U_k^T d\hat{W}(t), \quad (1)
   \]

2. **NONLINEAR PART**: use EIM for each coordinate \( x, y, \) and \( z \) with input basis \( \{ \hat{F}^x(\tilde{u}_j) \}_{j=1}^m, \{ \hat{F}^y(\tilde{u}_j) \}_{j=1}^m, \{ \hat{F}^z(\tilde{u}_j) \}_{j=1}^m \).

   \[
   \alpha U_k^T \hat{A}F(U_k \tilde{Q}(\hat{\tau})) \approx C F_{\text{EIM}}(\tilde{Q}(\hat{\tau})).
   \]
Details
http://www.cheme.cornell.edu/cheme/people/profile/projects.cfm?netid=
Long polymers are generally perceived as flexible chains and as such a interesting system to study is the ideal chain. The ideal chain is made up of segments of fixed length with random orientation of the end points. This system was studied early on from a statistical mechanics framework and maps to real systems by adjusting the segment length according to the stiffness of a polymer (i.e. any flexible polymer has a number of polymer repeat units after which there is no correlation between orientations of the starting and ending repeat unit). The segment length is known and the Kuhn step length.
The statistical mechanics treatment of the ideal chain system gives rise to an expression for a potential of an ideal chain based on its end to end distance. This gives rise to a restoring force to stretching based solely on the Entropic contribution to free energy and thus the potential is referred to as the entropy spring potential. The exact form of this expression is the inverse Langevin function and has widely used approximations such as the FENE potential.
\[ K(\hat{t}) = \begin{pmatrix} \kappa(\hat{t}) & \kappa(\hat{t}) & \cdots & \kappa(\hat{t}) \\ \kappa(\hat{t}) & \kappa(\hat{t}) & \cdots & \kappa(\hat{t}) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\hat{t}) & \kappa(\hat{t}) & \cdots & \kappa(\hat{t}) \end{pmatrix} \in \mathbb{R}^{3(N_b-1) \times 3(N_b-1)}, \]

Recall that \( \kappa = \lambda \kappa \). Then, we have \( \kappa(\hat{t}) = \lambda \dot{\gamma}(\hat{t}) \) for shear flow and \( \kappa(\hat{t}) = \lambda \dot{\epsilon}(\hat{t}) \) for uniaxial elongational flow.

\[ \Lambda = \frac{\lambda}{\lambda_H} = \frac{b}{b+3} \left( \frac{N_b^2 - 1}{3} \right) \]

\[ \mathbb{F}(\mathbb{Q}) = \begin{bmatrix} Z(\mathbb{Q}_1)\mathbb{Q}_1 \\ Z(\mathbb{Q}_2)\mathbb{Q}_2 \\ \vdots \\ Z(\mathbb{Q}_{N_b-1})\mathbb{Q}_{N_b-1} \end{bmatrix} \in \mathbb{R}^{3(N_b-1)} \] where, for \( i = 1, \ldots, N_b - 1 \),

\[ Z(\mathbb{Q}_i) = \begin{cases} 1 & : \text{Linear force} \\ \frac{1}{1 - \frac{1}{b^2} \mathbb{Q}_i^2 / b} & : \text{Nonlinear force (FENE)} \end{cases} \]

as defined in (??) with \( \mathbb{Q}_i = \begin{bmatrix} \dot{\mathbb{Q}}_i^x \\ \dot{\mathbb{Q}}_i^y \\ \dot{\mathbb{Q}}_i^z \end{bmatrix} \) and \( \mathbb{Q}_i^2 = |\mathbb{Q}_i|^2 = (\dot{\mathbb{Q}}_i^x)^2 + (\dot{\mathbb{Q}}_i^y)^2 + (\dot{\mathbb{Q}}_i^z)^2 \).
Recall that the nonlinear function $F$ is defined as:

$$F(Q) = \begin{bmatrix}
\frac{1}{1-|Q_1|^2/b} & \frac{\hat{Q}_1^X}{\hat{Q}_1} & \frac{\hat{Q}_1^Y}{\hat{Q}_1} & \frac{\hat{Q}_1^Z}{\hat{Q}_1} \\
\frac{1}{1-|Q_2|^2/b} & \frac{\hat{Q}_2^X}{\hat{Q}_2} & \frac{\hat{Q}_2^Y}{\hat{Q}_2} & \frac{\hat{Q}_2^Z}{\hat{Q}_2} \\
\vdots & \vdots & \vdots & \vdots \\
\frac{1}{1-|Q_{Nb-1}|^2/b} & \frac{\hat{Q}_{Nb-1}^X}{\hat{Q}_{Nb-1}} & \frac{\hat{Q}_{Nb-1}^Y}{\hat{Q}_{Nb-1}} & \frac{\hat{Q}_{Nb-1}^Z}{\hat{Q}_{Nb-1}}
\end{bmatrix} \in \mathbb{R}^{3(Nb-1)}$$

where, for $i = 1, \ldots, Nb - 1$, $|Q_i|^2 = (\hat{Q}_i^X)^2 + (\hat{Q}_i^Y)^2 + (\hat{Q}_i^Z)^2$.
Define

$$F^X(Q) = \begin{bmatrix}
\frac{1}{1-|Q_1|^2/b} \hat{Q}_1^X \\
\frac{1}{1-|Q_2|^2/b} \hat{Q}_2^X \\
\vdots \\
\frac{1}{1-|Q_{Nb-1}|^2/b} \hat{Q}_{Nb-1}^X
\end{bmatrix}, \quad F^Y(Q) = \begin{bmatrix}
\frac{1}{1-|Q_1|^2/b} \hat{Q}_1^Y \\
\frac{1}{1-|Q_2|^2/b} \hat{Q}_2^Y \\
\vdots \\
\frac{1}{1-|Q_{Nb-1}|^2/b} \hat{Q}_{Nb-1}^Y
\end{bmatrix}, \quad F^Z(Q) = \begin{bmatrix}
\frac{1}{1-|Q_1|^2/b} \hat{Q}_1^Z \\
\frac{1}{1-|Q_2|^2/b} \hat{Q}_2^Z \\
\vdots \\
\frac{1}{1-|Q_{Nb-1}|^2/b} \hat{Q}_{Nb-1}^Z
\end{bmatrix} \in \mathbb{R}^{3(Nb-1)}$$

Note that in the reduced system (1) $Q = U_k Q$. Each of the nonlinear functions $F^Z(Q), F^Z(Q)$, and $F^Z(Q)$, can be approximated by using the EIM.
Details  We will use the vectors of the nonlinear function evaluated at the Rouse modes, which span the solution space of the linear Rouse model. The number of Rouse modes is the number of beads in the chain. The \( j^{th} \) Rouse mode of the \( \ell^{th} \) spring is \( s_{\ell j} = \sin(\ell \delta_j) \), for \( \ell = 1, 2, \ldots, N \), and \( j = 0, 1, \ldots, N \). Define

\[
\delta_j := \frac{j \pi}{N_b}, \quad s_j := \begin{bmatrix} \sin(\delta_j) \\ \sin(2\delta_j) \\ \sin(3\delta_j) \\ \vdots \\ \sin(N\delta_j) \end{bmatrix} \in \mathbb{R}^N, \quad \text{and} \quad S := \begin{bmatrix} s_1 & \cdots & s_N \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad (4)
\]

for \( j = 1, \ldots, N \). Note that the mode \( j = 0 \) is just a free translation and therefore it is not used here. Note also that \( s_j \) is an eigenvector of the discrete Laplacian: \( L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ \vdots & \vdots & \vdots \\ -1 & 2 & -1 \end{bmatrix} \in \mathbb{R}^{(N_b-1) \times (N_b-1)} \), i.e., \( Ls_j = \lambda_j s_j \), where \( \lambda_j = 2 - 2 \cos(\delta_j) \) for \( j = 1, \ldots, N_b - 1 \).

The snapshots of nonlinear functions from the solution space of the linear model can be obtained as follows. For a fixed \( j \in \{1, \ldots, N\} \), substitute

\[
\begin{align*}
\hat{Q}_x^\ell &= \hat{Q}_y^\ell = \hat{Q}_z^\ell = \sin(\ell \delta_j) = s_{\ell j} \\
\hat{Q}^\ell &= \begin{bmatrix} \hat{Q}_x^\ell \\ \hat{Q}_y^\ell \\ \hat{Q}_z^\ell \end{bmatrix} = s_{\ell j} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \Rightarrow |\hat{Q}^\ell|^2 = 3s_{\ell j}^2, \quad \text{and} \quad \hat{Q}^X = \hat{Q}^Y = \hat{Q}^Z = s_j.
\end{align*}
\]

Define

\[
F_j := \begin{bmatrix} 1 \\ 1-3s_{1j}^2/b_s \\ 1-3s_{2j}^2/b_s \\ \vdots \\ 1-3s_{Nj}^2/b_s \end{bmatrix} \in \mathbb{R}^N, \quad \text{and} \quad F := \begin{bmatrix} F_1 & \cdots & F_N \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad (5)
\]

\[\text{1 Strobl, G. The Physics of Polymer, Springer 1997, p. 265}\]
Computing POD basis
POD basis can be found by using necessary and sufficient optimality conditions. Equivalently, POD basis can also be computed as follows.

- Define linear symmetric operator

\[ \mathcal{L} = [\langle y_i, y_j \rangle_X] \in \mathbb{R}^{n \times n} \]

- POD basis \( \{\phi_i\}_{i=1}^k \):

\[ \phi_i = \frac{1}{\sigma_i} \sum_{j=1}^n (v_i)_j y_j \]

for \( i = 1, \ldots, k \leq r \) where \( r = \text{rank}(\mathcal{L}) \) and

\[ \mathcal{L} v_i = \sigma_i^2 v_i, \]

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

- NOTE: \( v_i \in \mathbb{R}^n \), but \( \phi_i, y_j \in X \)
Posteriori Error bound of EIM approx $\hat{s}_m$

Suppose $\xi_{m+1}$ is not in $\text{span}\{\xi_1, \ldots, \xi_m\}$, and $s(\cdot; \mu) \in W_{m+1}^s := \text{span}\{\xi_1, \ldots, \xi_{m+1}\} \subset M^s := \{s(\cdot; \mu) | \mu \in \mathcal{D}\}$. Define

$$\begin{align*}
\hat{\epsilon}(\mu) &:= |s(z_{m+1}; \mu) - \hat{s}_m(z_{m+1}; \mu)|, \\
\mathcal{E}(x; \mu) &:= \hat{\epsilon}(\mu) q_{m+1}(x),
\end{align*}$$

where $z_{m+1}$ and $q_{m+1}(x)$ are the $(m+1)^{th}$ EIM point and basis. Then

1. $s(x; \mu) - \hat{s}_m(x; \mu) = \pm \mathcal{E}(x; \mu)$,

2. $\|s(\cdot; \mu) - \hat{s}_m(\cdot; \mu)\| \leq \hat{\epsilon}(\mu)$.

Note: in general, the assumption that $s(\cdot; \mu) \in W_{m+1}^s$ does not hold and the error bound above may not be exact (or applicable).
**EIM on indices:**

Given basis \( \{ \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m \} \subset \mathbb{R}^n \), and (nonlinear) function \( \mathbf{f} \in \mathbb{R}^n \),

1. Set \( \mathbf{u} \leftarrow \mathbf{u}_1 \)
2. Compute
   \[
   [\rho \quad j_{\text{max}}] = \max |\mathbf{u}|
   \]
3. Set \( \mathbf{q} = \mathbf{u}/\rho \), \( \rho = j_{\text{max}} \)
4. Set \( \mathbf{U} = [\mathbf{u}], \mathbf{Q} = [\mathbf{q}] \),
   \[
   \mathbf{P} = [\mathbf{e}_p], \mathbf{R} = [\rho]
   \]
5. For \( i = 2 \) to \( m \)
   - Set \( \mathbf{u} \leftarrow \mathbf{u}_i \)
   - Solve \( \mathbf{r} \) from \( \mathbf{P}^T \mathbf{Q} \mathbf{r} = \mathbf{P}^T \mathbf{u} \)
     \[ \Rightarrow \mathbf{r} = (\mathbf{P}^T \mathbf{Q})^{-1} \mathbf{P}^T \mathbf{u} \]
   - Form \( \mathbf{z} = \mathbf{u} - \mathbf{Q} \mathbf{r} \)
   - Compute
     \[
     [\rho \quad j_{\text{max}}] = \max \left\{ |\mathbf{z}| \right\}
     \]
   - Set \( \mathbf{q} = \mathbf{z}/\rho \), \( \rho = j_{\text{max}} \)
   - Set \( \mathbf{U} = [\mathbf{U} \ \mathbf{u}], \mathbf{Q} = [\mathbf{Q} \ \mathbf{q}] \),
   - \( \mathbf{P} = [\mathbf{P} \ \mathbf{e}_p], \mathbf{R} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ 0 & \rho \end{bmatrix} \)

The final approximation:

\[
\hat{\mathbf{f}} = \mathbf{Q}(\mathbf{P}^T \mathbf{Q})^{-1} \mathbf{P}^T \mathbf{f}
\]

**ERROR BOUND** \( \| \mathbf{f} - \hat{\mathbf{f}} \| \) [Sorensen, 2008]

Define \( \mathbf{M} = \mathbf{P}^T \mathbf{U} \) and \( \tilde{\mathbf{M}} = \tilde{\mathbf{P}}^T \tilde{\mathbf{U}} \),

- \( \tilde{\mathbf{P}}, \tilde{\mathbf{U}} \) are from previous iterations in EIM,
- \( \mathbf{P} = [\tilde{\mathbf{P}} \ \mathbf{p}]; \mathbf{U} = [\tilde{\mathbf{U}} \ \mathbf{u}], \mathbf{p} = \mathbf{e}_{j_{\text{max}}} \), \( \mathbf{u} \) are from current iteration of EIM.

Then

\[
\| \mathbf{f} - \hat{\mathbf{f}} \|_2 \leq C^{\text{EIM}} \| \mathbf{w} \|_2,
\]

where

\[
C^{\text{EIM}} = \left( \max \left\{ \| \tilde{\mathbf{M}}^{-1} \|_1, \rho^{-1}(1 + \| \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{P}}^T \mathbf{U} \|_1) \right\} \right) (1 + \| \alpha \|_1),
\]

with \( \alpha^T = \mathbf{p}^T \tilde{\mathbf{U}} \tilde{\mathbf{M}}^{-1} \), and \( \| \mathbf{w} \| = \| (I - \mathbf{U} \mathbf{U}^T) \mathbf{f} \| \) is the residual of the best 2-norm approximation to \( \mathbf{f} \) from the space Range(\( \mathbf{U} \))\

Notably, the residual norm can be arbitrary in \( \mathbb{R}^n \).

\[
a^*_i = \mathbf{u} \mathbf{u}^T \mathbf{f} \text{ minimizes } \| \mathbf{f} - \hat{\mathbf{f}} \|_2.
\]