Overview of Reduced-Order Modeling

Jeff Borggaard

Virginia Tech

1 September 2005
Outline

- Introduction
- KLE/POD
- Other Basis Selection Methods
- Computational Issues
Reduced-order models are useful when a complex system needs to be simulated in real-time, when multiple simulations are required, or when low dimensional models are needed for control.
Reduced-order models are useful when a complex system needs to be simulated in real-time, when multiple simulations are required, or when low dimensional models are needed for control.

They can also illuminate fundamental mechanisms in models

- nearby flows (visualize influence of parameters)
- discover how energy is transferred through different “modes”
**Concept**

- Reduced-order models are useful when a complex system needs to be simulated in real-time, when multiple simulations are required, or when low dimensional models are needed for control.
- They can also illuminate fundamental mechanisms in models
  - nearby flows (visualize influence of parameters)
  - discover how energy is transferred through different “modes”

- Given a time-dependent, large (or infinite dimensional) system with low dimensional dynamics, find a low dimensional model that approximates the system well.
Reduced-order models are useful when a complex system needs to be simulated in real-time, when multiple simulations are required, or when low dimensional models are needed for control.

They can also illuminate fundamental mechanisms in models

- nearby flows (visualize influence of parameters)
- discover how energy is transferred through different “modes”

Given a time-dependent, large (or infinite dimensional) system with low dimensional dynamics, find a low dimensional model that approximates the system well.

Generate a small number of *global* basis functions using simulations that are “typical.”
Reduced-order models are useful when a complex system needs to be simulated in real-time, when multiple simulations are required, or when low dimensional models are needed for control.

They can also illuminate fundamental mechanisms in models
- nearby flows (visualize influence of parameters)
- discover how energy is transferred through different “modes”

Given a time-dependent, large (or infinite dimensional) system with low dimensional dynamics, find a low dimensional model that approximates the system well.

Generate a small number of global basis functions using simulations that are “typical.”

Build a reduced-order basis using information obtained from given samples (simulations)
Reduced-Order Models (ROMs) have a long history:

- **Principal Component Analysis**

- **Common Factor Analysis**

- **Empirical Eigenfunctions**
Other Origins

- **Statistics:** Karhunen [Karhunen, 1946], Loève [Loève, 1955]
- **Control:** Principle Component Analysis, Balanced Truncation [Moore, 1981]
- **Fluids:** Proper Orthogonal Decomposition [Lumley 1967, Sirovich, 1987]
We will restrict our attention to models that are

- nonlinear
- time dependent
- parameter dependent
  - coefficients
  - initial conditions
  - boundary conditions
Two-Step Process

Basis Selection:
- Karhunen-Loève Expansion (KLE)/Proper Orthogonal Decomposition (POD)
- Principle Interval Decomposition (PID)
- Sampling, e.g. CVT

Model Construction:
- Galerkin
- Multiscale modeling ideas:
  - Nonlinear-Galerkin
  - LES, Patch Dynamics, Homogenization
- Updating
Example: Burgers Equation

To get the basic idea, we develop a reduced-order model for Burgers equation. This example is from Kunisch and Volkwein [KV99]:

\[ y_t(t, x) + \frac{1}{2} (y^2(t, x))_x = \varepsilon y_{xx}(t, x) \]

with boundary conditions \( y(t, 0) = 0 = y(t, 1) \) and initial conditions

\[ y(0, x) = \begin{cases} 
1 & x \leq 0.5 \\
0 & \text{otherwise}
\end{cases}. \]

For this example, we take \( \varepsilon \) as 0.01.
Example: Burgers Equation

Simulation
Construct an “optimal” low dimensional basis on which to represent the solution $y$

$$\{\phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_r(\cdot)\}.$$ 

We denote the linear space spanned by this basis as

$$\mathcal{P}^r = \text{span} \{\phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_r(\cdot)\},$$

and represent a low order solution in this space:

$$y_\ell(t, x) = \sum_{j=1}^{r} a_j(t) \phi_j(x) \in \mathcal{P}^r.$$ 

Build a model for the amplitude coefficients $\{a_j(\cdot)\}$, eg. Galerkin projection of governing equations.
Example: Burgers Equation

Basis 1: Fourier Basis (based on eigenmodes for $\partial_{xx}$)
Example: Burgers Equation

Basis 1: Projected initial conditions, $r = 6$
Example: Burgers Equation

Simulation using Fourier Basis
Example: Burgers Equation

Simulation (for comparison)
Example: Burgers Equation

Basis 2: Karhunen-Loève Expansion/Proper Orthogonal Decomposition
Example: Burgers Equation

Basis 2: Projected initial conditions, $r = 6$
Example: Burgers Equation

Simulation using KLE/POD Basis
Example: Burgers Equation

Simulation (for comparison)
Example: Burgers Equation

Comparison of Both ROMs at $T = 1$
Beginning with a “sample” $y(t, x)$ from either
- Analytical Methods
- Computational Simulation(s)
- Experimental Measurements
- etc.

with $y(t, \cdot) \in \mathcal{H}$ for each $t \in \mathcal{T}$ (or $y(t) \in \mathcal{H}$).

The first POD mode (basis function) maximizes the Rayleigh quotient,

$$\max_{\phi} \left\{ \frac{1}{T} \int_{\mathcal{T}} \frac{\left| \langle y(t, \cdot), \phi(\cdot) \rangle \right|^2}{\langle \phi(\cdot), \phi(\cdot) \rangle} \, dt \right\}.$$
One Interpretation:

\[
\max_{\phi} \left\{ \frac{1}{T} \int_{\mathcal{T}} \frac{|\langle y(t, \cdot), \phi(\cdot) \rangle|^2}{\langle \phi(\cdot), \phi(\cdot) \rangle} \, dt \right\}.
\]

The first POD mode, $\phi_1$, can be interpreted as the direction which maximizes the time averaged projection of $y$.

Subsequent modes maximize this quotient over the orthogonal complement to the span of the current basis elements.

The second POD mode, $\phi_2$, would satisfy

\[
\max_{\phi \perp \phi_1} \left\{ \frac{1}{T} \int_{\mathcal{T}} \frac{|\langle y(t, \cdot), \phi(\cdot) \rangle|^2}{\langle \phi(\cdot), \phi(\cdot) \rangle} \, dt \right\},
\]

The remaining modes are defined similarly.
The POD mode must satisfy the necessary condition

$$\langle R^s(x, \cdot), \phi(\cdot) \rangle = \lambda \phi(x)$$

Considering $\mathcal{H} = L^2(\Omega)$, we solve

$$\int_{\Omega} R^s(x, \bar{x})\phi(\bar{x})d\bar{x} = \lambda \phi(x),$$

where

$$R^s(x, \bar{x}) = \frac{1}{T} \int_{T} y(t, x)y^*(t, \bar{x})dt$$

is the spatial auto-correlation function.
When \( \mathcal{H} \) is finite dimensional, e.g. when \( y \) is the solution to a system of ordinary differential equations

\[
\dot{y}(t) = f(y(t)) \quad y(0) = y_0 \in \mathbb{R}^n,
\]

we have the analogous form for \( R^s \):

\[
R^s = \frac{1}{T} \int_T y(t)y^T(t)dt
\]

where the POD vectors are eigenvectors of the symmetric, non-negative definite matrix \( R^s \),

\[
R^s \phi = \lambda \phi.
\]
Since $R^s$ is a symmetric, non-negative definite matrix, we know

- the eigenvalues are real
- there is a full set of orthonormal eigenvectors
- these span $\mathbb{R}^n$

For the ODE, $y$ is real and the POD vectors are real.

\[
R^s P^n = P^n \Lambda \\
R^s = P^n \Lambda (P^n)^T.
\]
In nearly every case, the integral in $R^s$ needs to be replaced by a quadrature.

Given a discrete set of time samples/snapshots 

$$\{t_1, t_2, \ldots, t_m\},$$

we approximate the matrix $R^s$ by

$$R^s = \frac{1}{T} \int_T y(t)y^T(t)dt \approx \frac{1}{T} \sum_{j=1}^{m} \Delta t_j y(t_j)y^T(t_j).$$

If the time samples are equi-spaced, $\Delta t_j = T/m$, and $Y$ is the matrix with components

$$[Y]_{ij} = y_i(t_j),$$

i.e. time snapshots form columns of $Y$, then

$$R^s = \frac{1}{m} YY^T.$$
KLE/POD (7)

SVD Interpretation

Let \( \mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \) be a singular value decomposition for \( \mathbf{Y} \), then

\[
\mathbf{R}_s \phi = \frac{1}{m} \mathbf{Y} \mathbf{Y}^T \phi = \frac{1}{m} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T \phi = \mathbf{U} \left( \frac{1}{m} \mathbf{\Sigma}^2 \right) \mathbf{U}^T \phi.
\]

Note that if \( \phi = \mathbf{u}_i \), the \( i \)th column of \( \mathbf{U} \), then

\[
\mathbf{R}_s \mathbf{u}_i = \frac{1}{m} \sigma_i^2 \frac{\mathbf{u}_i}{\sqrt{\lambda_i}}
\]

so there is a clear connection between the eigenvalue problem for \( \mathbf{R}_s \) and the SVD of \( \mathbf{Y} \):

- POD vectors are left singular vectors
- Eigenvalues of \( \mathbf{R}_s \) are the “time average” of the square of the singular values.
Approximating properties of the SVD

The optimal rank 1 approximation to $Y$ is

$$\tilde{Y} = \sigma_1 u_1 v_1^T$$

in the sense that

$$\|Y - \tilde{Y}\|_F \leq \|Y - cuv^T\|_F$$

for any constant $c$, $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^m$.

We also have the error bound

$$\|Y - \tilde{Y}\|_F = \left\| \sum_{i=2}^{\min(m,n)} \sigma_i u_i v_i^T \right\|_F = \left( \sum_{i=2}^{\min(m,n)} \sigma_i^2 \right)^{1/2}.$$
The analogous estimates hold for higher dimensional POD bases

\[ U^r = [u_1 \ u_2 \ \cdots \ u_r] \in \mathbb{R}^{n \times r}, \quad \Sigma_r = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r \times r} \]

\[ V^r = [v_1 \ v_2 \ \cdots \ v_r] \in \mathbb{R}^{m \times r} \]

\[ \tilde{Y} = U^r \Sigma_r (V^r)^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \]

with approximation error

\[ \|Y - \tilde{Y}\|_F^2 = \sum_{i=r+1}^{\min(m,n)} \sigma_i^2 \]

\[ \phi_i = u_i \quad a_i = \sigma_i v_i \]
Consider

\[ R^t = Y^T Y \]

then

\[ R^t \psi = YY^T \psi = V \Sigma^T \Sigma V^T \psi. \]

Note that if \( \psi = v_i \), then

\[ \frac{1}{m} R^t v_i = \frac{1}{m \sigma_i^2} v_i \]

we find \( v_i \),

\[ Y v_i = U \Sigma V^T v_i = U \Sigma e_i = \sigma_i u_i. \]
Thus, upon finding solutions to the eigenvalue problem

\[ \frac{1}{m} R^t \psi_i = \lambda_i \psi_i, \quad i = 1, 2, \ldots, m \]

we can compute KLE/POD vectors as

\[ \phi_i = \frac{1}{\sigma_i} Y \psi_i, \quad i = 1, \ldots, \min(m, n). \]

If \( m < n \), then the remaining left singular vectors are not needed to represent \( Y \) and would be reduced.

This is known as the method of snapshots [Sir87], computationally attractive if \( m \ll n \).
An analogous method can be used to find $v_i$ given a $u_i$ (and $\sigma_i$).

$$Y^T u_i = V\Sigma U^T u_i = V\Sigma e_i = \sigma_i v_i.$$ 

We now “interpret” our POD eigenvectors $\phi$ (of $\frac{1}{m}YY^T$) and $\psi$ (of $\frac{1}{m}Y^TY$) in terms of the original solution.

$$\langle y(t_j), \phi_i \rangle = a_i(t_j) = [\sigma_i \psi_i]_j$$

$$y(t_j) \approx \sum_{i=1}^{r} a_i(t_j) \phi_i.$$
The following relations also hold:

\[
\frac{1}{T} \int_T y(t)a_i(t)dt \approx \frac{1}{m} \sum_{j=1}^{m} y(t_j)a_i(t_j) = Y \frac{1}{m} \sigma_i \psi_i = \frac{1}{\sigma_i} \left[ Y \lambda_i \psi_i \right] = \frac{1}{\sigma_i} \phi_i.
\]

and

\[
\frac{1}{T} \int_T a_i^2(t)dt \approx \frac{1}{m} \sum_{j=1}^{m} a_i(t_j)a_i(t_j) = \frac{1}{m} \psi_i^T \psi_i \sigma_i^2 = \lambda_i,
\]
The infinite dimensional case has similar behavior.

In most cases [HLB96],

\[ R^s(x, \bar{x}) = \frac{1}{T} \int_T y(t, x)y^*(t, \bar{x})dt \]

satisfies

- \( R^s(x, x) \) is finite for each \( x \)
- \( \int_\Omega R^s(x, x)dx < \infty \).

This allows us to conclude

\[ R^s(x, \bar{x}) \in L^2(\Omega \times \Omega). \]
The result of $R^s(x, \bar{x}) \in L^2(\Omega \times \Omega)$ is that $R^s$ is the kernel of the operator

$$R : L^2(\Omega) \to L^2(\Omega)$$

defined as

$$R\phi(\cdot) \equiv \int_{\Omega} R^s(\cdot, \bar{x})\phi(\bar{x})d\bar{x}.$$ 

We also know

- $R$ is self adjoint
- $R$ is bounded (by the norm of $R^s$)
- $R$ is non-negative definite
- $R$ is compact
Thus, we know

- The eigenvalues are real and non-negative
- The only accumulation point is at 0
- There exists an orthonormal basis of eigenvectors for the range of $\mathcal{R}$

Thus, the countable set of eigenvalues are real and can be ordered

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_j \geq \cdots$$

and $\lambda_j \geq 0$ for all $j$. 
KLE/POD (17)

We also have the following expansions

\[ R^s(x, \bar{x}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j^*(\bar{x}) \]

and

\[ y(t, x) = \sum_{j=1}^{\infty} a_j(t) \phi_j(x) \]

\[ = \sum_{j=1}^{\infty} \sigma_j \psi_j(t) \phi_j(x) \]

for almost all \( t \).
An equivalent way to characterize $\phi_1$ is as a solution to

$$\min_{a,\phi} \left\{ \frac{1}{T} \int_T \|y(t, \cdot) - a(t)\phi(\cdot)\|^2 dt \mid \|\phi(\cdot)\| = 1 \right\}$$

with the optimal amplitude $a_1$ as

$$a_1(t) = \langle y(t, \cdot), \phi_1(\cdot) \rangle, \quad t \in T.$$
Since
\[ a_1(t) = \langle y(t, \cdot), \phi_1(\cdot) \rangle, \quad t \in \mathcal{T}, \]
we have
\[
\frac{1}{T} \int_{\mathcal{T}} a_1^2(t) dt = \int_{\Omega} \phi_1^*(x) \int_{\Omega} R_s(x, \bar{x}) \phi_1(\bar{x}) d\bar{x} \; dx
\]
\[
= \int_{\Omega} \lambda_1 \phi_1^*(x) \phi_1(x) dx
\]
\[
= \lambda_1 \| \phi_1(\cdot) \|^2
\]
\[
= \lambda_1.
\]
The natural strategy is to build low order models by truncating the sequence

\[ y_\ell(t, x) \equiv \sum_{j=1}^{r} a_j(t) \phi_j(x) \approx \sum_{j=1}^{\infty} a_j(t) \phi_j(x) = y(t, x). \]

There are natural heuristics for choosing \( r \) based on the sequence \( \{\lambda_j\} \). The relative error in \( y_\ell \) is

\[ \frac{\|y - y_\ell\|}{\|y\|} = \frac{\sum_{j=r+1}^{\infty} \lambda_j}{\sum_{j=1}^{\infty} \lambda_j} = 1 - \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{\infty} \lambda_j} \equiv E_r. \]
KLE/POD (22)

Note: The KLE/POD leads to the optimal low order basis on which to approximate $y$ in the sense that for any other linear space, eg. span$\{\tilde{\phi}_j\}$ (orthonormal), with

$$y(t, x) = \sum_{j=1}^{\infty} \tilde{a}_j(t) \tilde{\phi}_j(x)$$

we have the estimate

$$\sum_{j=1}^{n} \int_{T} a_j^2(t) dt \geq \sum_{j=1}^{n} \int_{T} \tilde{a}_j^2(t) dt$$

in other words,

$$\left\| y(t, \cdot) - \sum_{j=1}^{n} a_j(t) \phi_j(\cdot) \right\| \leq \left\| y(t, \cdot) - \sum_{j=1}^{n} \tilde{a}_j(t) \tilde{\phi}_j(\cdot) \right\|.$$
However, $E_r$ is merely a heuristic for many reasons:

- Availability of $a_j(\cdot)$ coefficients
- Building models vs. Optimally representing data
  - vary initial conditions, boundary conditions, model parameters
  - (off design)
- Nonlinear models

In other words, the choice of POD basis functions is motivated by best approximation of the sample $y$.

This approximation can be very low order in practice.
KLE/POD (24)

KLE/POD can be remarkably effective (Burgers’ eq. example)
KLE/POD - Galerkin simulation with $r = 1$: 

![Graph showing a 3D plot with axes labeled 0 to 1 on the x, y, and z axes. The plot represents a surface that varies from (0,0,0) to (1,1,1).]
KLE/POD (26)

KLE/POD - Galerkin simulation with $r = 2$:
KLE/POD (27)

KLE/POD - Galerkin simulation with $r = 3$: 

![Graph showing a 3D simulation with $r = 3$.]
KLE/POD (28)

KLE/POD - Galerkin simulation with $r = 9$: 

![Graph](image-url)
KLE/POD (29)

KLE/POD - Galerkin simulation with $r = 12$:
Linear properties satisfied by the sample are inherited by the basis.

Recall that

$$\phi_i = \frac{1}{\sigma_i} Y \psi_i$$

thus each $\phi_i$ is a linear combination of the time snapshots.

Let $\mathcal{P}$ be a closed linear subspace of $L^2$. Then if all snapshots are in $\mathcal{P}$, the KLE/POD basis functions are also in $\mathcal{P}$. 
Consider, for example, the Navier Stokes equations

\[
\begin{align*}
 \mathbf{y}_t + \mathbf{y} \cdot \nabla \mathbf{y} &= -\nabla p + \frac{1}{Re} \Delta \mathbf{y} + \mathbf{f} \\
 \nabla \cdot \mathbf{y} &= 0
\end{align*}
\]

subject to \( \mathbf{y}(0, \cdot) = \mathbf{y}_0(\cdot) \) and \( \mathbf{y} = \mathbf{0} \) on the boundary.

Thus, if each snapshot satisfies \( \nabla \cdot \mathbf{y}(t, \cdot) = 0 \), then each POD basis element will. Likewise for the homogeneous boundary conditions.
In many problems of interest, $y(t, x)$ will be the solution to a partial differential equation. Finite dimensional approximations may have the form

$$y^N(t, x) = \sum_{k=1}^{n} y_k(t) h_k(x)$$

where $\{h_k(\cdot)\}_{k=1}^{n}$ is a given set of basis functions:

- finite elements
- special functions, polynomials, etc.
To approximate the POD modes, it makes sense to seek representations in this same basis, i.e.

\[ \phi^N_i(x) = \sum_{j=1}^{N} (\phi_i)_j h_j(x), \quad i = 1, \ldots, r. \]

In this case, substitution of \( y^N \) and \( \phi^N \) into

\[ \int_{\Omega} R^s(x, \bar{x}) \phi(\bar{x}) d\bar{x} = \lambda \phi(x) \]

leads to the matrix eigenvalue problem for POD coefficients \( p \),

\[ \frac{1}{m} YY^T Mp = \lambda p, \]

where \( M \) is the so-called “mass matrix” with components

\[ [M]_{ij} = \int_{\Omega} h_i(x) h_j(x) dx. \]
The matrix $M$ is symmetric and positive definite. Thus, we can use a change of variables to create a symmetric eigenvalue problem (see eg. [SF73], [KV99]).

Let $M = LL^T$ be a Cholesky factorization and by premultiplication of $L^T$ we have

$$\frac{1}{m} L^T YY^T L L^T p = \lambda L^T p.$$ 

Hence, we have a complete set of eigenvectors, with real, non-negative eigenvalues, etc.

Since $v v^T = I = p L L^T p$ the POD basis coefficients are $M$-orthogonal.
Principle Interval Decomposition (1)

One limitation of KLE/POD is the fact that modes must be good over the entire time interval. For example, consider the solution to the wave equation

\[ y(t, x) = y_0(x - ct) \]

and \( y_0(x) = e^{-x^2} \).

Then the spatial correlation function

\[
TR^s(x, \bar{x}) = \int_{-T/2}^{T/2} e^{-(x-t)^2} e^{-(\bar{x}-t)^2} dt \\
= \sqrt{\frac{\pi}{2}} e^{-(x-\bar{x})^2/2} \left( \text{erf} \left( \frac{T - x - \bar{x}}{\sqrt{2}} \right) + \text{erf} \left( \frac{T + x + \bar{x}}{\sqrt{2}} \right) \right).
\]
Principle Interval Decomposition (2)

We have

$$\lim_{T \to \infty} TR^s(x, \bar{x}) = \sqrt{\frac{\pi}{2}} e^{-(x-\bar{x})^2} = f(x - \bar{x})$$

When $R^s(x, \bar{x}) = f(x - \bar{x})$, we say $R^s$ is homogeneous.

With this property, we can expand $R^s$ as

$$R^s(x, \bar{x}) = \sum_j c_j e^{2\pi ij(x-\bar{x})}$$

which indicates that

$$\phi_j(x) = e^{2\pi ijx}$$

are the eigenfunctions of $R(x, \bar{x})$... the Fourier modes.
Principle Interval Decomposition (3)

Instead of looking for basis vectors that are good representations to our signal (solution) over the entire time, we decompose the time into principle intervals $[IJz00]$.

Thus, we seek solutions of the form

$$y_{\ell}(t, x) \approx \sum_{j} \tilde{a}_j(t) \tilde{\phi}_j(x)$$

where

$$\tilde{a}_j(t) = \begin{cases} a_j(t) & t \in T_j \\ 0 & \text{otherwise} \end{cases}$$

$T_j = [t_{j-1}, t_j]$ are the Principle Intervals

$\tilde{\phi}_j$ are the Principle Modes
We would like to choose the time intervals and modes so that the following estimate holds

\[ \int_T \| y(t, \cdot) - y_\ell(t, \cdot) \|^2 \, dt \leq \varepsilon \int_T \| y(t, \cdot) \|^2 \, dt. \]

One means to enforce this estimate is to require

\[ \int_{T_j} \| y(t, \cdot) - y_\ell(t, \cdot) \|^2 \, dt \leq \varepsilon \int_{T_j} \| y(t, \cdot) \|^2 \, dt. \]
Principle Interval Decomposition (5)

There are practical limits on the size of $\varepsilon$ to keep the basis size small.

Once selected, the computation of principle intervals and modes follows a similar approach to POD. They satisfy the minimization problem

$$\min_{a, \phi} \left\{ \frac{1}{t_{j+1} - t_j} \int_{t_j}^{t_{j+1}} \| y(t, x) - a(t) \phi(x) \|^2 dt \mid \| \phi(\cdot) \| = 1 \right\}.$$ 

- PID modes are computed sequentially in time
- The length of the time interval is determined simultaneously with each mode (as large as possible that still satisfies the estimate)
- The “snapshot” approach is often attractive
- POD updating strategy later
Basis 3: Principle Interval Decomposition

Jeff Borggaard (Virginia Tech)
Example: Burgers Equation

Basis 1: Projected initial conditions, $r = 6$
Example: Burgers Equation

Basis 2: Projected initial conditions, \( r = 6 \)
Example: Burgers Equation

Basis 3: Projected initial conditions, $r = 6$
Principle Interval Decomposition (6)

Note that the PID basis looks very similar to actual solution snapshots.

As $\varepsilon \to 0$, the PID basis becomes the time snapshots used to construct $R^s$.

PID can detect significant modes that are only important over a small period of time.

This suggests the more straight-forward approach of using sampling techniques.
Sampling (1)

Centroidal Voronoi Tessellations [BDGL03]

K-Means Clustering
Given a $k$ dimensional basis on which to build a reduced-order model, there are many reasons to update this basis:

- More data from a forecast
- More refined simulations are available
- Computation of a PID basis
- TR-POD
- etc.

We can either compute additional basis elements orthogonal to the given set, or recompute a new (perhaps larger) basis using this new information.
Let \( \{y(t_1), y(t_2), \cdots \} \) be given data and let

\[
Y_k = [y(t_1) \ y(t_2) \ \cdots \ y(t_k)] \in \mathbb{R}^{n \times k}
\]

Algorithm: [GSA03]

Let \( Y_k = U_k \Sigma_k V_k^T \) with \( U_k \in \mathbb{R}^{n \times k} \), \( \Sigma_k \in \mathbb{R}^{k \times k} \) and \( V_k \in \mathbb{R}^{k \times k} \).

\[
Y_k Y_k^T = U_k \Sigma_k V_k^T V_k \Sigma_k U_k^T = U_k \Sigma_k \left( \Sigma_k U_k \right)^T \equiv \hat{Y}_k \hat{Y}_k^T
\]
Updating KLE/POD Bases (3)

Let
\[ Y_r = [y(t_{k+1}) \cdots y(t_{k+r})] \in \mathbb{R}^{n\times r} \]
be \( r \) additional snapshots.

Decompose \( Y_r \) as
\[ Y_r = U_k \Gamma + U_k^\perp \Theta \]
where
\[ \Gamma \in \mathbb{R}^{k\times r}, \quad \Theta \in \mathbb{R}^{r\times r}, \quad \text{and} \quad U_k^\perp \in \mathbb{R}^{n\times r} \]
and
\[ U_k U_k^\perp = 0 \quad \text{and} \quad \left(U_k^\perp\right)^T U_k^\perp = I_{r\times r}. \]
We can now write

\[ Y_{k+r} = \hat{Y}_k Y_r = \left[ U_k U_k^\perp \right] \begin{bmatrix} \Sigma_k & \Gamma \\ 0 & \Theta \end{bmatrix} \hat{S} \].

Let

\[ \hat{S} = T \Sigma_{k+r} \hat{V}^T. \]

This SVD is inexpensive since \( \hat{S} \) is \((k + r) \times (k + r)\). With this information, we can compute

\[ Y_{k+r} = \left[ U_k U_k^\perp \right] T \Sigma_{k+r} \hat{V}^T \equiv U_{k+r} \Sigma_{k+r} \hat{V}^T. \]

We only need to keep \( \hat{Y}_{k+r} = U_{k+r} \Sigma_{k+r} \).
Two-Step Process

Basis Selection:
- KLE/POD
- PID
- Sampling (CVT)

Model Construction:
- Galerkin
- Nonlinear-Galerkin
- Multiscale modeling:
  - LES, Patch Dynamics, Homogenization
Given a low dimensional basis

\[ \{ \phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_r(\cdot) \} \]

from one of the above strategies, the completion of our reduced-order model

\[ y_\ell(t, x) = \sum_{j=1}^{r} a_j(t) \phi_j(x) \]

requires construction of the amplitude coefficients \( \{ a_j(\cdot) \}_{j=1}^{r} \).

These can be obtained by Galerkin projection of the differential equation model onto \( \mathcal{P}^r = \text{span}\{\phi_1, \ldots, \phi_r\} \).
To parallel our earlier discussion, we first consider the finite dimensional case.

Consider the ordinary differential equation

\[ \dot{y}(t) = f(y(t)), \quad y(0) = y_0 \in \mathbb{R}^n \]

\((r \ll n)\).

Substitute the expression

\[ y_\ell(t) = \sum_{j=1}^{r} a_j(t) \phi_j \]

into the ODE and project the equations onto \(P^r\).
This leads to

$$\sum_{j=1}^{r} \dot{a}_j(t) \phi_j = f \left( \sum_{j=1}^{r} a_j(t) \phi_j \right) \quad \sum_{j=1}^{r} a_j(0) \phi_j = y_0$$

and then

$$\dot{a}_i(t) = \left\langle f \left( \sum_{j=1}^{r} a_j(t) \phi_j \right), \phi_i \right\rangle$$

$$a_i(0) = \left\langle y_0, \phi_i \right\rangle, \quad i = 1, \ldots, r,$$

since

$$\left\langle \phi_i, \phi_j \right\rangle = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}.$$
For the special case of linear, autonomous ODEs,

\[ f(y) = Ay, \]

and the amplitude coefficient equations are

\[
\dot{a_i}(t) = \left\langle A \left( \sum_{j=1}^{r} a_j(t) \phi_j \right), \phi_i \right\rangle
\]

which simplifies to

\[
\dot{a}(t) = \Phi^T A \Phi a(t) \quad a(0) = \Phi^T y_0
\]

where

\[
\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_r \end{bmatrix}_{n \times r}
\]
KLE/POD Galerkin (5)

For PDEs, we follow the standard finite element approach.

As an example, consider Burgers equation

\[
y_t(t, x) + \frac{1}{2} (y^2(t, x))_x = \varepsilon y_{xx}(t, x).
\]

The Galerkin finite element problem is:

Find \( y^N \in S^h \) such that

\[
\langle y^N_t, v \rangle = -\langle y^N y^N_x, v \rangle - \varepsilon \langle y^N_x, v_x \rangle
\]

for all \( v \in S^h \). The solution satisfies the projected initial conditions

\[
\langle y^N(0, \cdot) - y_0(\cdot), v(\cdot) \rangle = 0 \quad \text{for all} \quad v \in S^h.
\]
KLE/POD Galerkin (6)

Using $y_\ell(t, \cdot) \in P^r \subset S^h$ instead of $y^N(t, \cdot) \in S^h$, the KLE/POD Galerkin equations for the amplitude coefficients satisfy

$$\left\langle \sum_{j=1}^{r} \dot{a}_j(t) \phi_j(\cdot), \phi_i(\cdot) \right\rangle = - \left\langle \sum_{j=1}^{r} a_j(t) \phi_j(\cdot) \sum_{k=1}^{r} a_k(t) \phi'_k(\cdot), \phi_i(\cdot) \right\rangle$$

$$= - \varepsilon \left\langle \sum_{j=1}^{r} a_j(t) \phi'_j(\cdot), \phi'_i(\cdot) \right\rangle,$$

where $\phi_i \in P^r$ for all $i = 1, \ldots, r$. Using the orthogonality property of $\{\phi_j\}$, and the linearity of the inner product, we have

$$\dot{a}_i(t) = -N(a) - \varepsilon \sum_{j=1}^{r} K_{ij} a_j(t) \quad i = 1, \ldots, r$$

where

$$K_{ij} = \int_{\Omega} \phi'_j(x) \phi'_i(x) dx.$$
For the nonlinear term $N(a)$, we define a sequence of matrices

$$\left[ T^{(i)} \right]_{jk} = \int_{\Omega} \phi_j(x) \phi'_k(x) \phi_i(x) dx$$

then

$$N(a) = \left\langle \sum_{j=1}^{r} a_j(t) \phi_j(\cdot) \sum_{k=1}^{r} a_k(t) \phi'_k(\cdot), \phi_i(\cdot) \right\rangle = a^T T^{(i)} a$$

Thus, the equation for the amplitude coefficients is

$$\dot{a}_i(t) = -a^T T^{(i)} a - \varepsilon [K a]_i, \quad i = 1, \ldots, r$$

with

$$a_i(0) = \int_{\Omega} y_0(x) \phi_i(x) dx.$$
Time varying boundary conditions.

**Approaches:**

- Homogenize the equations, build a POD basis for the homogenized equation, then assemble the full model. Let $\mu(t,x)$ be a function that satisfies the boundary conditions and utilize the centering trajectory approach discussed below.

- Build POD bases with nonzero boundary conditions. Project the boundary conditions onto the POD basis. These are constraints.
Stabilized POD

Recent numerical experiments suggest that if a stabilized Galerkin finite element method is required for the simulation due to strong advection (for example, Burgers equation with $\varepsilon \ll 1$) then this same stabilized Galerkin procedure should be used to integrate the coefficient equations (Sachs et al.).
KLE/POD Limitations/Extensions

If the data is not in the snapshots/samples, it is not well represented in the KLE/POD basis.

There is an “art” to choosing appropriate samples. This can be guided by mathematics.
Snapshot Selection

Problem (w/ D. Sutton): solve

\[ y_t(t, x) = \varepsilon y_{xx}(t, x) \]

subject to

\[ y(0, x) = 0, \quad y(t, 0) = \sin(t), \quad \text{and} \quad y(t, 1) = 0. \]

and denote the solution by \( y(t, x; \varepsilon) \).

Want to develop a reduced-order model that is appropriate for a wide range of

\[ \varepsilon \in (\varepsilon_1, \varepsilon_2). \]
Snapshot Selection

First POD modes for different values of epsilon

epsilon = 0.01
epsilon = 0.1
epsilon = 1

Jeff Borggaard (Virginia Tech)
Our approach:

We want to find $\phi$ to maximize the time averaged and $\varepsilon$ averaged projection

$$\max_{\phi} \left\{ \frac{1}{\Delta \varepsilon} \int_{\varepsilon_1}^{\varepsilon_2} \frac{1}{T} \int_{\mathcal{T}} \frac{|\langle y(t, \cdot; \varepsilon), \phi(\cdot) \rangle|^2}{\langle \phi(\cdot), \phi(\cdot) \rangle} \, dt \, d\varepsilon \right\}.$$
Snapshot Selection

Our previous discussion holds upon defining

\[ R^s(x, \bar{x}) = \frac{1}{\varepsilon_2 - \varepsilon_1} \int_{\varepsilon_1}^{\varepsilon_2} \frac{1}{T} \int_T y(t, x; \varepsilon)y(t, \bar{x}; \varepsilon) dt \, d\varepsilon \]

\[ \approx \frac{1}{T} \sum_{k=1}^{g} w_k \int_T y(t, x; \varepsilon_k)y(t, \bar{x}; \varepsilon_k) dt. \]

To compute coefficients of the POD basis vectors, define

\[ R^s = \frac{1}{m} \left[ w_1 Y(\varepsilon_1) Y^T(\varepsilon_1) + \cdots + w_g Y(\varepsilon_g) Y^T(\varepsilon_g) \right] M \]

and find eigenvectors of \( R^s \)...
Comparison of Midpoint and Gauss quadrature rules

Maximum Error of the POD Model

- POD from the given Epsilon
- Midpoint
- Gaussian
A Parallel KLE/POD Algorithm

The following algorithm is known as the filtered subspace iteration (Gugercin):

Each processor, $P_k$, $k = 1, 2, \ldots, n_p$ performs concurrently:

1. **Initialization.**
   
   (a) Calculate the dominant $m$ right singular vectors of local $J_k W_k$: $V_k^{(1)}$.
   
   (b) All-to-all send of local $V_k^{(1)}$; Receive $\{V_i^{(1)}\}_{i \neq k}$ from other processors; Assemble the starting block: $V^{(1)} = \begin{bmatrix} V_1^{(1)} & V_2^{(1)} & \ldots & V_{n_p}^{(1)} \end{bmatrix}$
A Parallel KLE/POD Algorithm

The following algorithm is known as the *filtered subspace iteration*:

(2) **Start Iteration.**

For $j = 1, \ldots, J_{\text{max}}$

(a) Calculate

\[ W_k^{(j)} = (J_k W_k V^{(j)})^T J_k W_k V^{(j)}. \]

(b) All-to-all send of local $W_k^{(j)}$;

Receive $\{W_i^{(j)}\}_{i \neq k}$ from other processors;

Sum blocks to get

\[ W^{(j)} = (WV^{(j)})^T WV^{(j)}. \]

(c) Calculate locally:

\[ W^{(j)} = Z_1 \Sigma_1^2 Z_1^T; \]

\[ \hat{U}_k^{(1)} = W_1 V^{(1)} Z_1 \Sigma_1^{-1}. \]
A Parallel KLE/POD Algorithm

\begin{align*}
\mathbf{u}_1 & \quad \mathbf{u}_2 & \quad \mathbf{u}_3 \\
\mathbf{u}_4 & \quad \mathbf{u}_5 & \quad \mathbf{u}_6
\end{align*}