

Overview of Reduced-Order Modeling

Jeff Borggaard

Virginia Tech

1 September 2005

Outline

- Introduction
- KLE/POD
- Other Basis Selection Methods
- Computational Issues

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.

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”

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.

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

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.
- 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)

Weather Forecasting Methods

Reduced-Order Models (ROMs) have a long history:

- Principal Component Analysis
 - ▶ Hotelling, Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.*, 1933.
- Common Factor Analysis
 - ▶ Thurstone, *Multiple Factor Analysis*, 1947.
- Empirical Eigenfunctions
 - ▶ Lorenz, Empirical orthogonal functions and statistical weather prediction, *Statistical Forecasting Project*, 1956.

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]

Idea

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]:

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

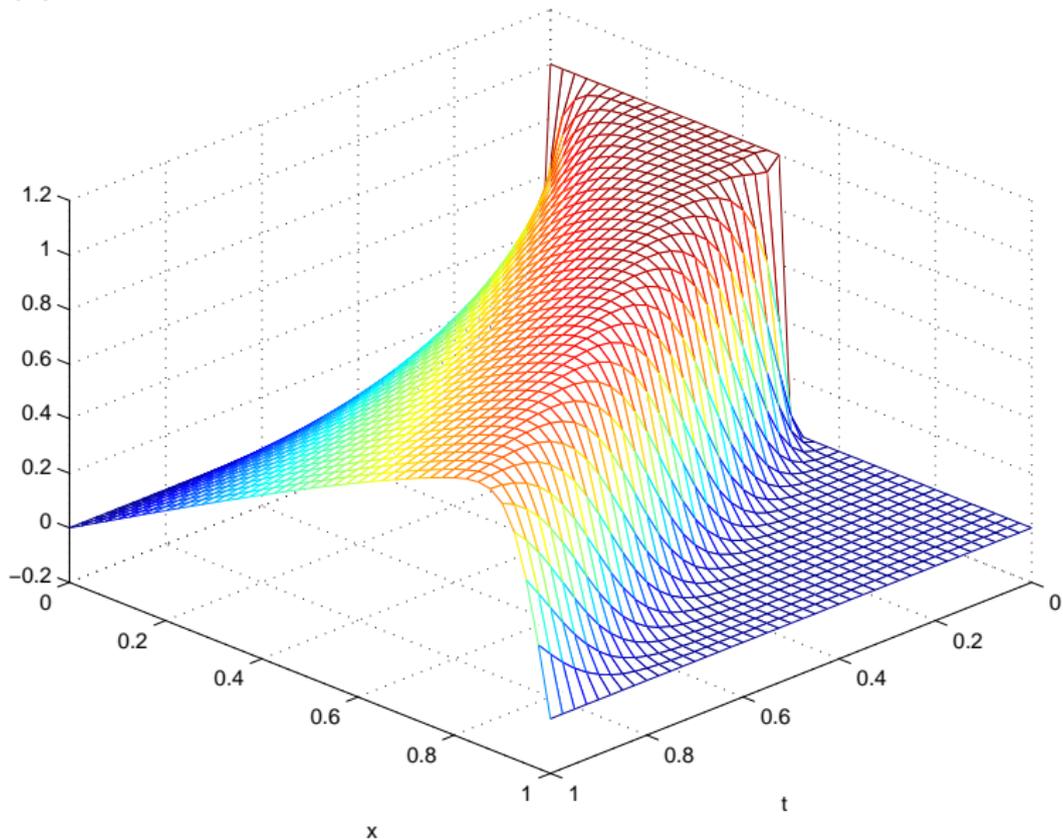
with boundary conditions $y(t, 0) = 0 = y(t, 1)$ and initial conditions

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

For this example, we take ε as 0.01.

Example: Burgers Equation

Simulation



Example: Burgers Equation

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

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

We denote the linear space spanned by this basis as

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

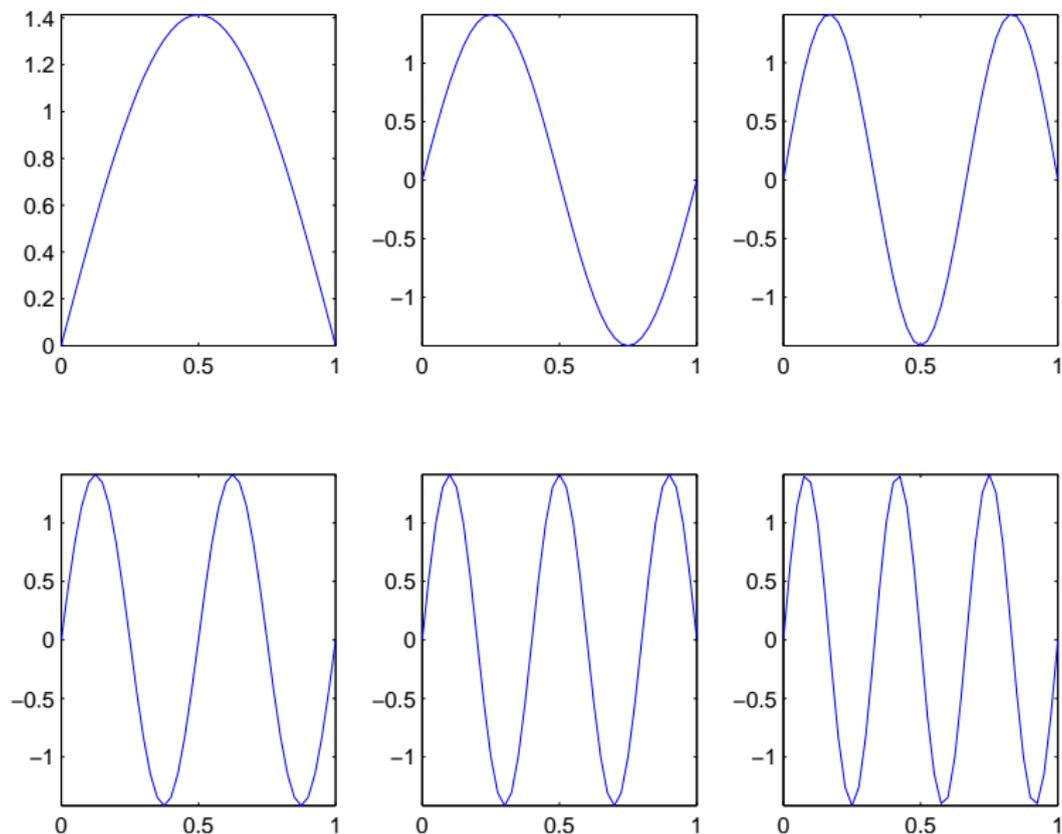
and represent a low order solution in this space:

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

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

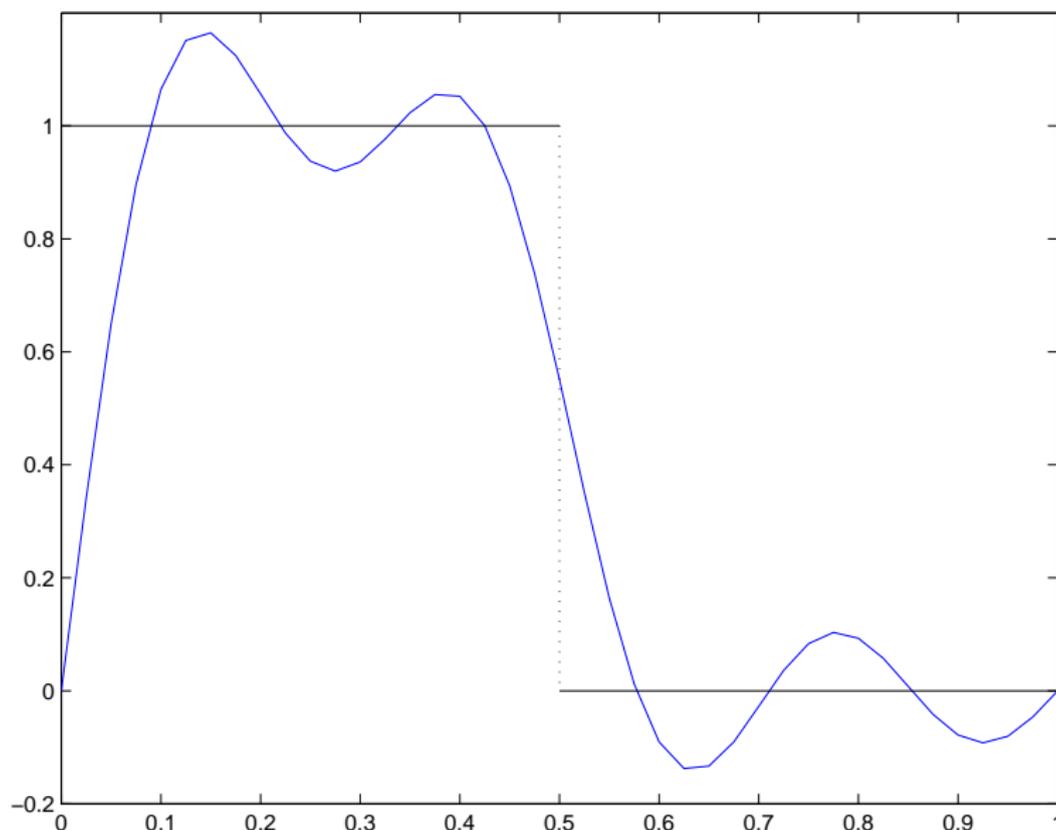
Example: Burgers Equation

Basis 1: Fourier Basis (based on eigenmodes for ∂_{xx})



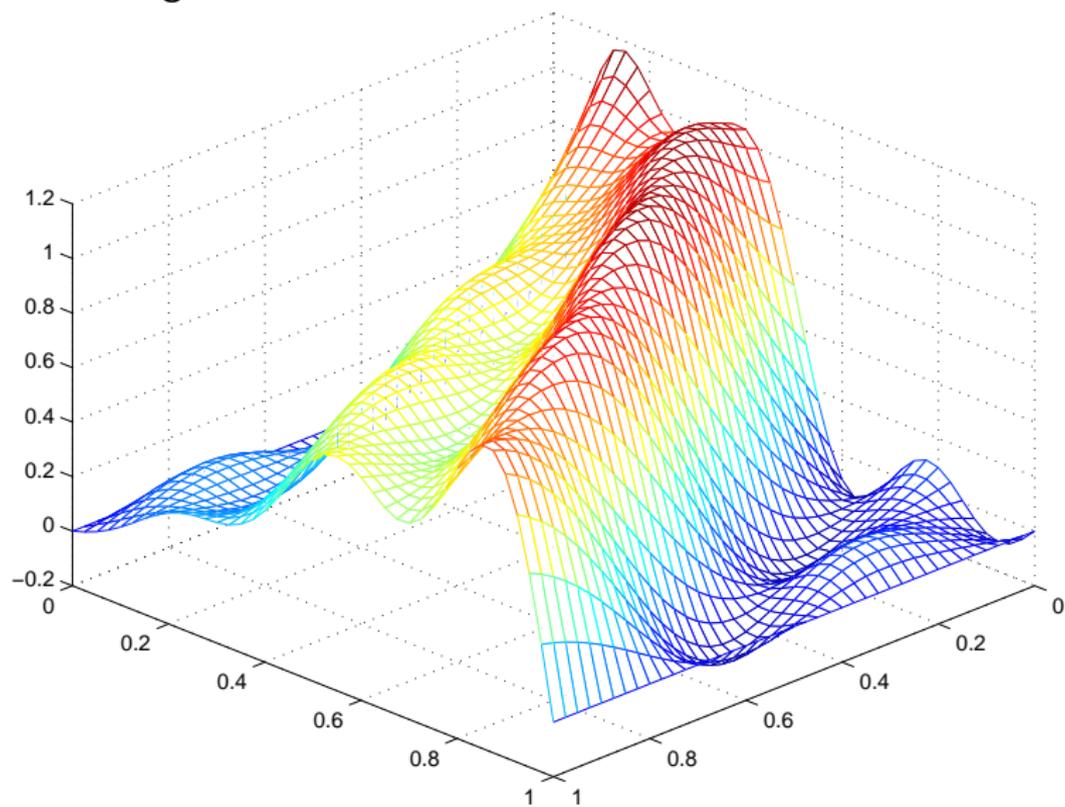
Example: Burgers Equation

Basis 1: Projected initial conditions, $r = 6$



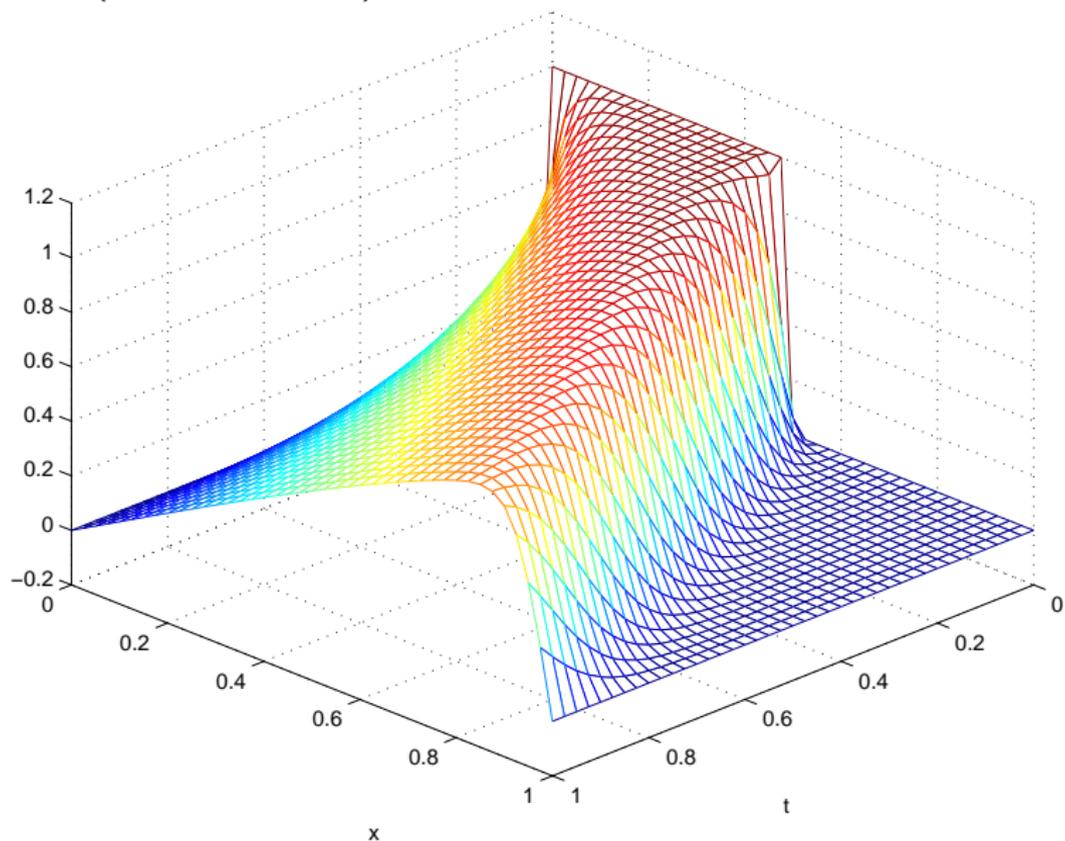
Example: Burgers Equation

Simulation using Fourier Basis



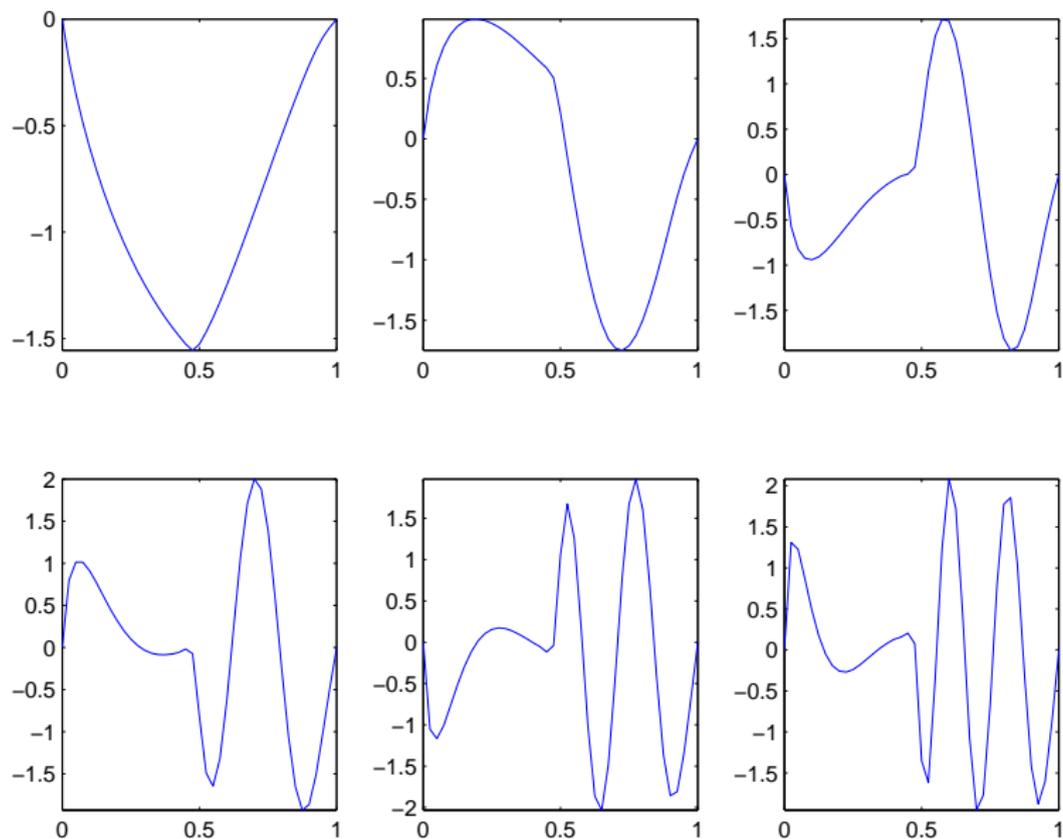
Example: Burgers Equation

Simulation (for comparison)



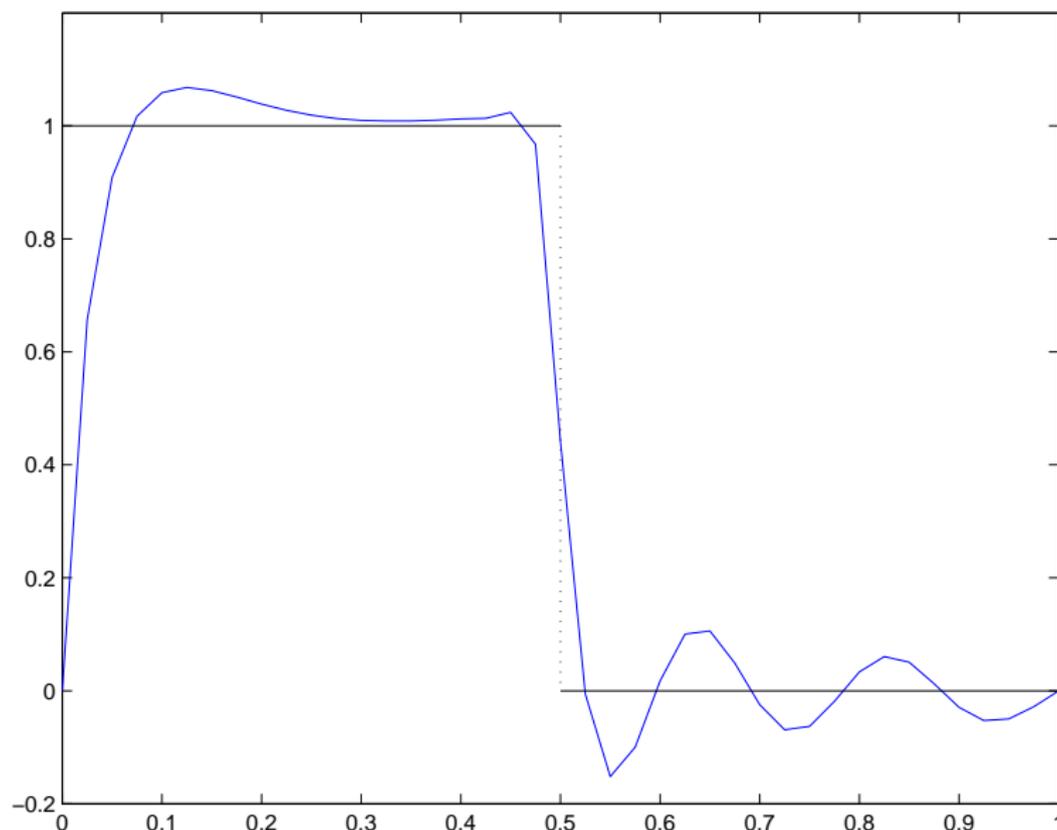
Example: Burgers Equation

Basis 2: Karhunen-Loève Expansion/Proper Orthogonal Decomposition



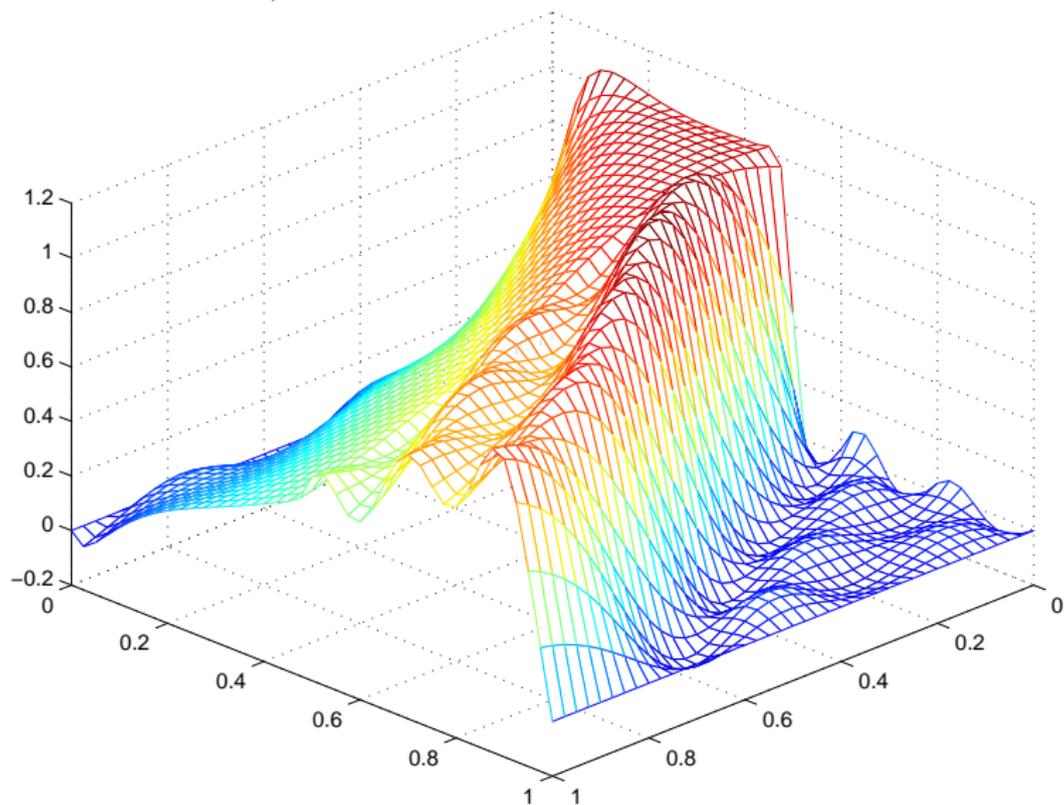
Example: Burgers Equation

Basis 2: Projected initial conditions, $r = 6$



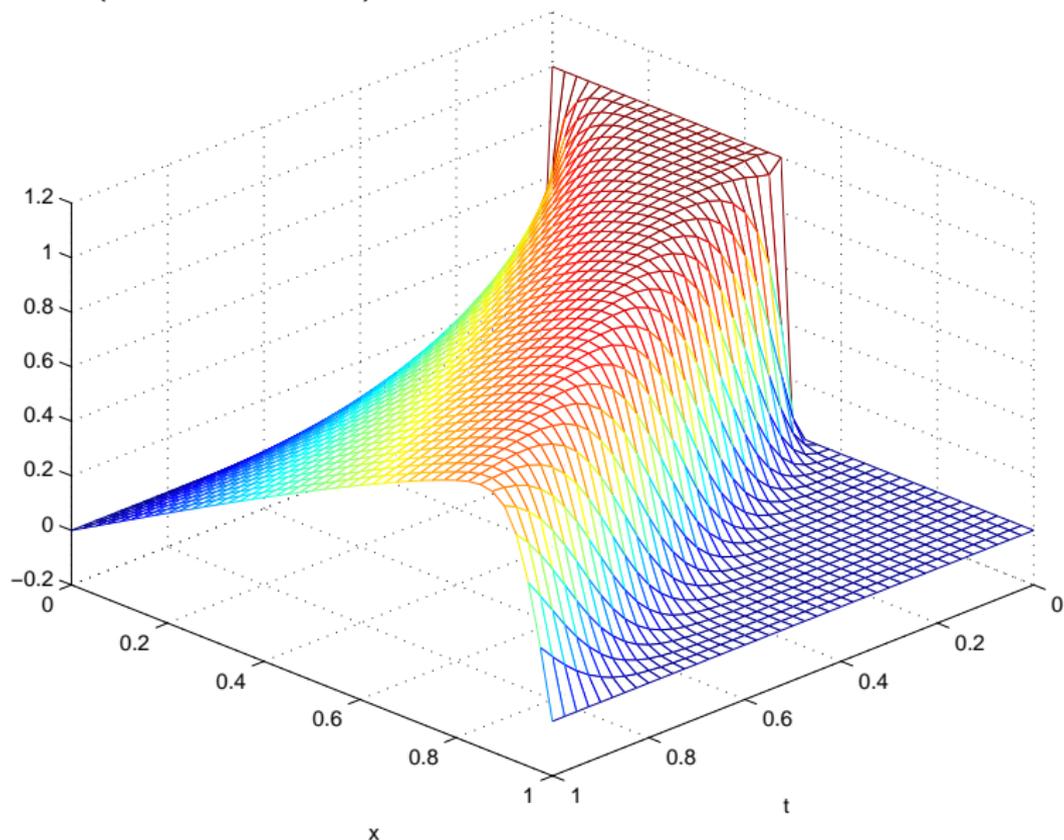
Example: Burgers Equation

Simulation using KLE/POD Basis



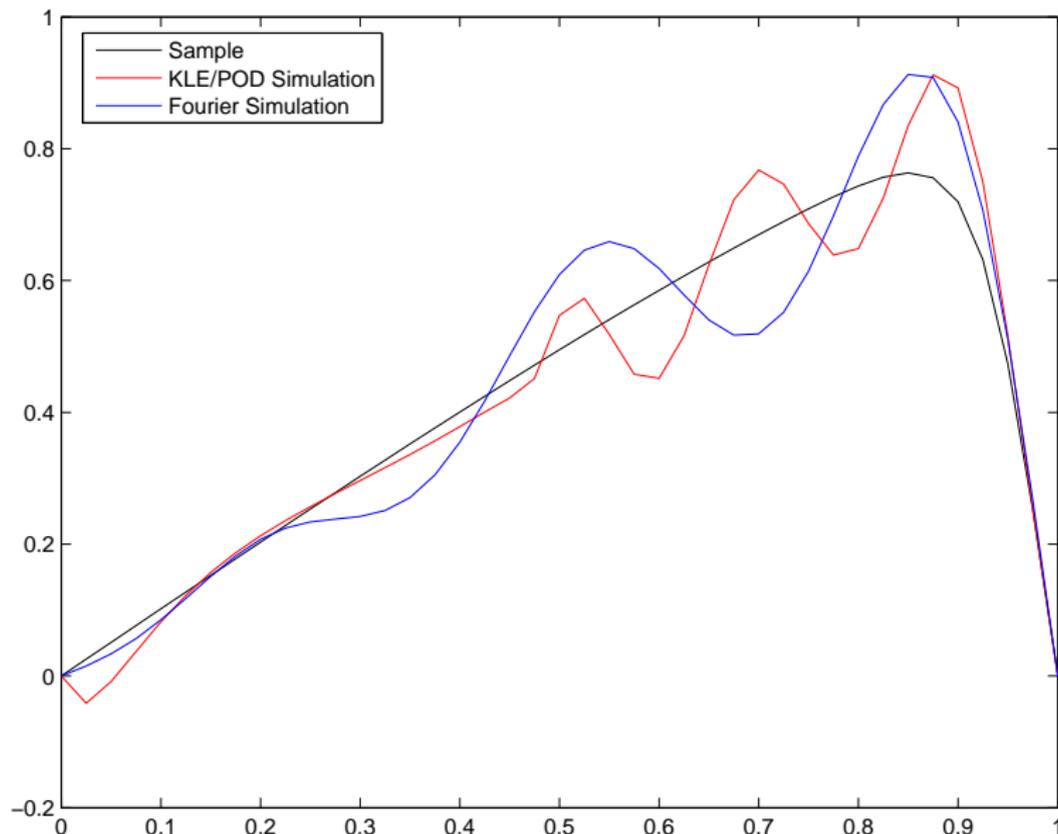
Example: Burgers Equation

Simulation (for comparison)



Example: Burgers Equation

Comparison of Both ROMs at $T = 1$



KLE/POD (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 $\mathbf{y}(t) \in \mathcal{H}$).

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

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

KLE/POD (2)

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, ϕ_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, ϕ_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.

KLE/POD (3)

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

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

is the *spatial auto-correlation function*.

KLE/POD (4)

When \mathcal{H} is finite dimensional, eg. when \mathbf{y} is the solution to a system of ordinary differential equations

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)) \quad \mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^n,$$

we have the analogous form for R^S :

$$\mathbf{R}^S = \frac{1}{T} \int_T \mathbf{y}(t) \mathbf{y}^T(t) dt$$

where the POD vectors are eigenvectors of the symmetric, non-negative definite matrix \mathbf{R}^S ,

$$\mathbf{R}^S \phi = \lambda \phi.$$

KLE/POD (5)

Since \mathbf{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, \mathbf{y} is real and the POD vectors are real.

$$\begin{aligned}\mathbf{R}^s \mathbf{P}^n &= \mathbf{P}^n \Lambda \\ \mathbf{R}^s &= \mathbf{P}^n \Lambda (\mathbf{P}^n)^T.\end{aligned}$$

KLE/POD (6)

In nearly every case, the integral in \mathbf{R}^s needs to be replaced by a quadrature.

Given a discrete set of time samples/snapshots

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

we approximate the matrix \mathbf{R}^s by

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

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

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

i.e. time snapshots form columns of \mathbf{Y} , then

$$\mathbf{R}^s = \frac{1}{m} \mathbf{Y} \mathbf{Y}^T.$$

KLE/POD (7)

SVD Interpretation

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

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

Note that if $\phi = u_i$, the i th column of \mathbf{U} , then

$$\mathbf{R}^s u_i = \underbrace{\frac{1}{m} \sigma_i^2}_{\lambda_i} u_i$$

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.

KLE/POD (8)

Approximating properties of the SVD

The optimal rank 1 approximation to \mathbf{Y} is

$$\tilde{\mathbf{Y}} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$$

in the sense that

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F \leq \|\mathbf{Y} - c\mathbf{u}\mathbf{v}^T\|_F$$

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

We also have the error bound

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F = \left\| \sum_{i=2}^{\min(m,n)} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \right\|_F = \left(\sum_{i=2}^{\min(m,n)} \sigma_i^2 \right)^{1/2}.$$

KLE/POD (9)

The analogous estimates hold for higher dimensional POD bases

$$\mathbf{U}^r = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_r] \in \mathbb{R}^{n \times r}, \quad \Sigma_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$$

$$\mathbf{V}^r = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_r] \in \mathbb{R}^{m \times r}$$

$$\tilde{\mathbf{Y}} = \mathbf{U}^r \Sigma^r (\mathbf{V}^r)^T = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

with approximation error

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

$$\phi_i = \mathbf{u}_i \quad \mathbf{a}_i = \sigma_i \mathbf{v}_i$$

KLE/POD (10)

Consider

$$\mathbf{R}^t = \mathbf{Y}^T \mathbf{Y}$$

then

$$\mathbf{R}^t \psi = \mathbf{Y} \mathbf{Y}^T \psi = \mathbf{V} \Sigma^T \Sigma \mathbf{V}^T \psi.$$

Note that if $\psi = \mathbf{v}_i$, then

$$\frac{1}{m} \mathbf{R}^t \mathbf{v}_i = \frac{1}{m} \underbrace{\sigma_i^2}_{\lambda_i} \mathbf{v}_i$$

we find \mathbf{v}_i ,

$$\mathbf{Y} \mathbf{v}_i = \mathbf{U} \Sigma \mathbf{V}^T \mathbf{v}_i = \mathbf{U} \Sigma \mathbf{e}_i = \sigma_i \mathbf{u}_i.$$

KLE/POD (11)

Thus, upon finding solutions to the eigenvalue problem

$$\frac{1}{m} \mathbf{R}^t \psi_i = \lambda_i \psi_i, \quad i = 1, 2, \dots, m$$

we can compute KLE/POD vectors as

$$\phi_i = \frac{1}{\sigma_i} \mathbf{Y} \psi_i, \quad i = 1, \dots, \min(m, n).$$

If $m < n$, then the remaining left singular vectors are not needed to represent \mathbf{Y} and would be *reduced*.

This is known as the *method of snapshots* [Sir87], computationally attractive if $m \ll n$.

KLE/POD (12)

An analogous method can be used to find \mathbf{v}_i given a \mathbf{u}_i (and σ_i).

$$\mathbf{Y}^T \mathbf{u}_i = \mathbf{V} \Sigma \mathbf{U}^T \mathbf{u}_i = \mathbf{V} \Sigma \mathbf{e}_i = \sigma_i \mathbf{v}_i.$$

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

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

$$y(t_j) \approx \sum_{i=1}^r a_i(t_j) \phi_i.$$

KLE/POD (13)

The following relations also hold:

$$\begin{aligned}\frac{1}{T} \int_{\mathcal{T}} y(t) a_i(t) dt &\approx \frac{1}{m} \sum_{j=1}^m y(t_j) a_i(t_j) \\ &= \mathbf{Y} \frac{1}{m} \sigma_i \psi_i = \frac{1}{\sigma_i} [\mathbf{Y} \lambda_i \psi_i] = \frac{1}{\sigma_i} \phi_i.\end{aligned}$$

and

$$\begin{aligned}\frac{1}{T} \int_{\mathcal{T}} a_i^2(t) dt &\approx \frac{1}{m} \sum_{j=1}^m a_i(t_j) a_i(t_j) \\ &= \frac{1}{m} \psi_i^T \psi_i \sigma_i^2 = \lambda_i,\end{aligned}$$

KLE/POD (14)

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

$$R^s(x, \bar{x}) \in L^2(\Omega \times \Omega).$$

KLE/POD (15)

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

$$\mathcal{R} : L^2(\Omega) \rightarrow L^2(\Omega)$$

defined as

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

We also know

- \mathcal{R} is self adjoint
- \mathcal{R} is bounded (by the norm of R^s)
- \mathcal{R} is non-negative definite
- \mathcal{R} is compact

KLE/POD (16)

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

$$\begin{aligned} y(t, x) &= \sum_{j=1}^{\infty} a_j(t) \phi_j(x) \\ &= \sum_{j=1}^{\infty} \sigma_j \psi_j(t) \phi_j(x) \end{aligned}$$

for almost all t .

KLE/POD (18)

An equivalent way to characterize ϕ_1 is as a solution to

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

with the optimal amplitude a_1 as

$$a_1(t) = \langle y(t, \cdot), \phi_1(\cdot) \rangle, \quad t \in \mathcal{T}.$$

KLE/POD (20)

Since

$$a_1(t) = \langle y(t, \cdot), \phi_1(\cdot) \rangle, \quad t \in \mathcal{T},$$

we have

$$\begin{aligned} \frac{1}{T} \int_{\mathcal{T}} a_1^2(t) dt &= \int_{\Omega} \phi_1^*(x) \int_{\Omega} R^s(x, \bar{x}) \phi_1(\bar{x}) d\bar{x} dx \\ &= \int_{\Omega} \lambda_1 \phi_1^*(x) \phi_1(x) dx \\ &= \lambda_1 \|\phi_1(\cdot)\|^2 \\ &= \lambda_1. \end{aligned}$$

KLE/POD (21)

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_ℓ 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. $\text{span}\{\tilde{\phi}_j\}$ (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,

$$\left\| y(t, \cdot) - \sum_{j=1}^n a_j(t) \phi_j(\cdot) \right\| \leq \left\| y(t, \cdot) - \sum_{j=1}^n \tilde{a}_j(t) \tilde{\phi}_j(\cdot) \right\|.$$

KLE/POD (23)

However, E_r is merely a heuristic for many reasons:

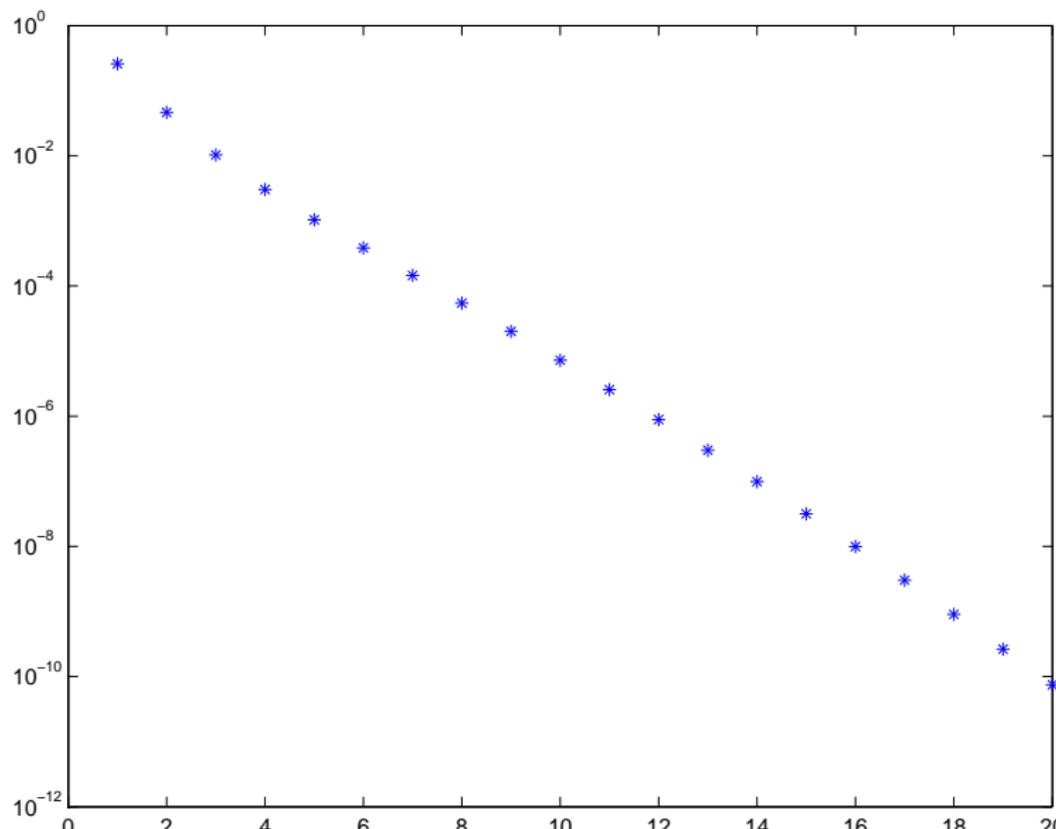
- Availability of $a_j(\cdot)$ coefficients
- Building models vs. Optimally representing data
 - ▶ vary initial conditions, boundary conditions, model parameters
 - ▶ (off design)
- Nonlinear models

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.

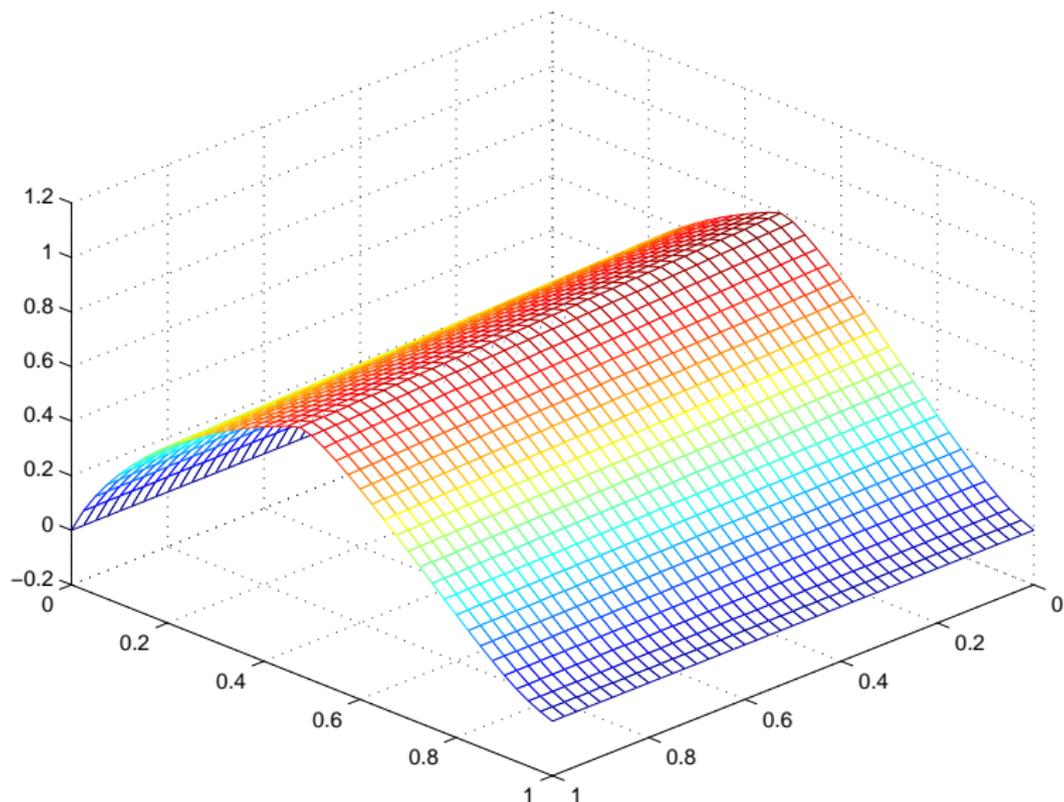
KLE/POD (24)

KLE/POD can be remarkably effective (Burgers' eq. example)



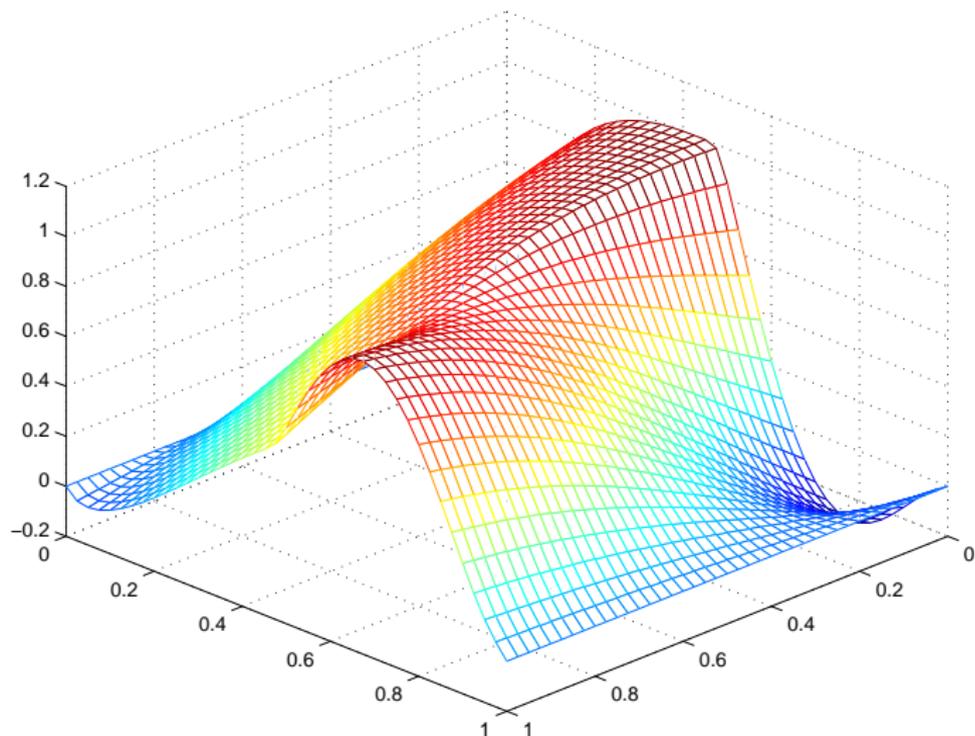
KLE/POD (25)

KLE/POD - Galerkin simulation with $r = 1$:



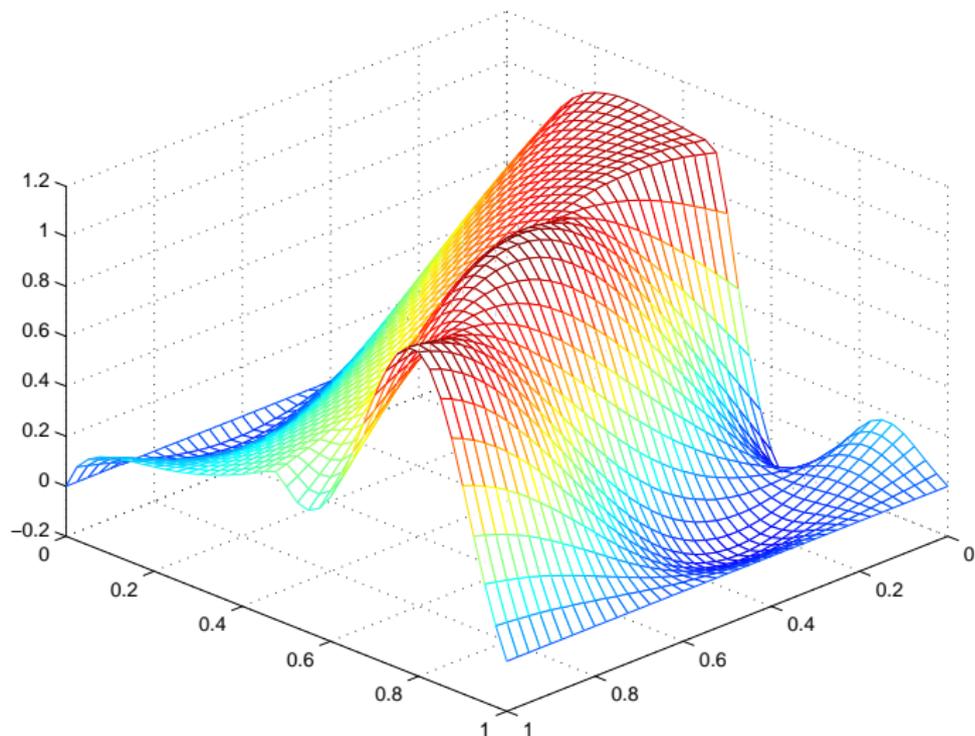
KLE/POD (26)

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



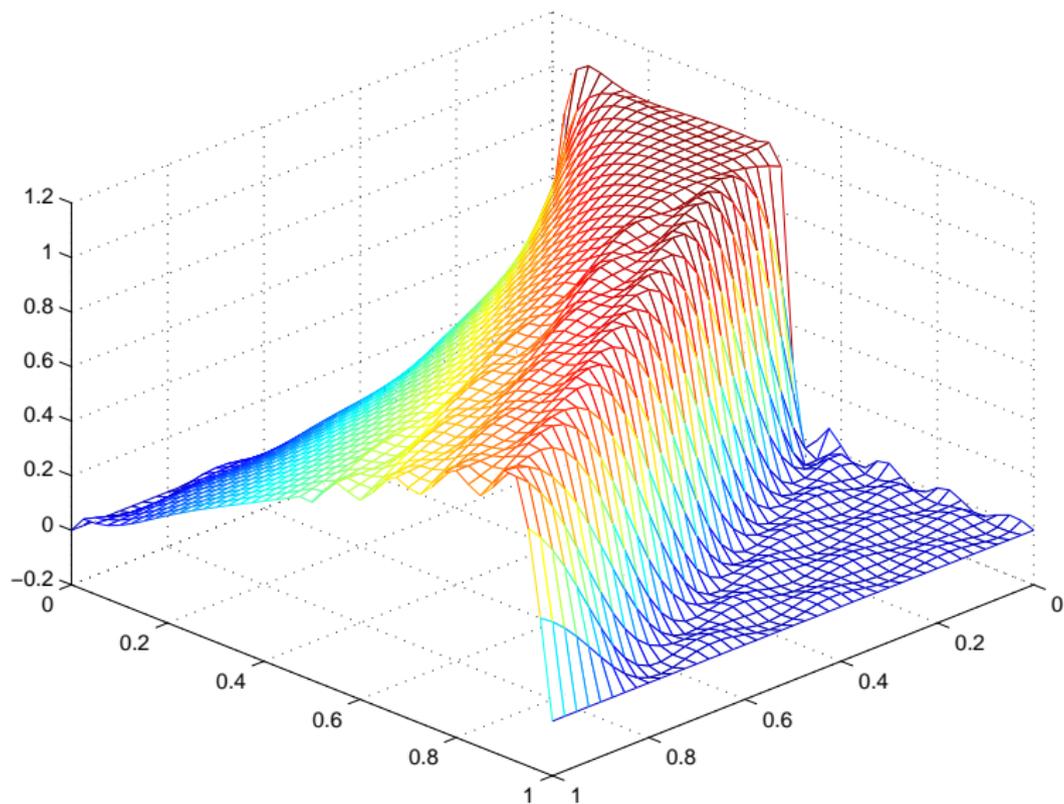
KLE/POD (27)

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



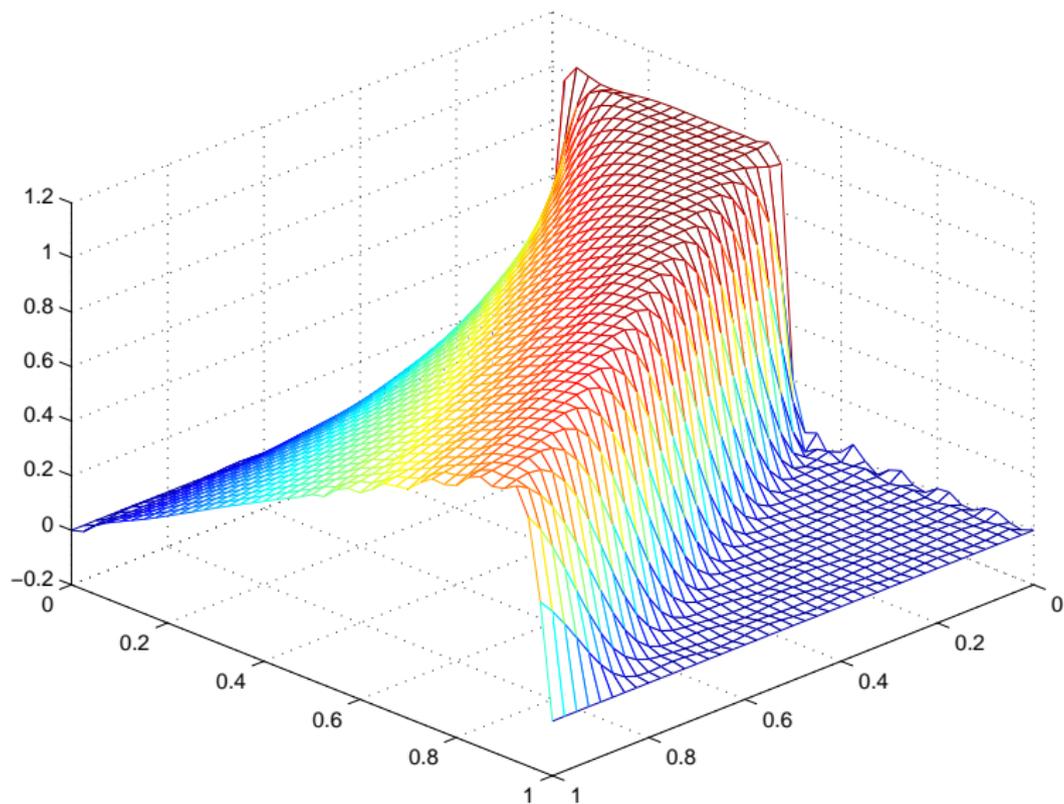
KLE/POD (28)

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



KLE/POD (29)

KLE/POD - Galerkin simulation with $r = 12$:



KLE/POD (30)

Linear properties satisfied by the sample are inherited by the basis.

Recall that

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

thus each ϕ_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} .

KLE/POD (31)

Consider, for example, the Navier Stokes equations

$$\begin{aligned}\mathbf{y}_t + \mathbf{y} \cdot \nabla \mathbf{y} &= -\nabla p + \frac{1}{Re} \Delta \mathbf{y} + \mathbf{f} \\ \nabla \cdot \mathbf{y} &= 0\end{aligned}$$

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.

KLE/POD for Discretized Problems (1)

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.

KLE/POD for Discretized Problems (2)

To approximate the POD modes, it makes sense to seek representations in this same basis, i.e.

$$\phi_i^N(x) = \sum_{j=1}^N (\phi_i)_j h_j(x), \quad i = 1, \dots, r.$$

In this case, substitution of y^N and ϕ^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 \mathbf{p} ,

$$\frac{1}{m} \mathbf{Y} \mathbf{Y}^T \mathbf{M} \mathbf{p} = \lambda \mathbf{p},$$

where \mathbf{M} is the so-called “mass matrix” with components

$$[M]_{ij} = \int_{\Omega} h_i(x) h_j(x) dx.$$

KLE/POD for Discretized Problems (3)

The matrix \mathbf{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 $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ be a Cholesky factorization and by premultiplication of \mathbf{L}^T we have

$$\frac{1}{m} \underbrace{\mathbf{L}^T \mathbf{Y} \mathbf{Y}^T \mathbf{L}}_{\mathbf{A}} \underbrace{\mathbf{L}^T \mathbf{p}}_{\mathbf{v}} = \lambda \underbrace{\mathbf{L}^T \mathbf{p}}_{\mathbf{v}}.$$

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

Since $\mathbf{v}\mathbf{v}^T = \mathbf{I} = \mathbf{p}\mathbf{L}\mathbf{L}^T\mathbf{p}$ the POD basis coefficients are \mathbf{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

$$\begin{aligned} TR^s(x, \bar{x}) &= \int_{-T/2}^{T/2} e^{-(x-t)^2} e^{-(\bar{x}-t)^2} dt \\ &= \sqrt{\frac{\pi}{2}} e^{-(x-\bar{x})^2/2} \left(\operatorname{erf}\left(\frac{T-x-\bar{x}}{\sqrt{2}}\right) + \operatorname{erf}\left(\frac{T+x+\bar{x}}{\sqrt{2}}\right) \right). \end{aligned}$$

Principle Interval Decomposition (2)

We have

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

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

$$\begin{aligned} R^s(x, \bar{x}) &= \sum_j c_j e^{2\pi i j(x-\bar{x})} \\ &= \sum_j c_j e^{2\pi i j x} e^{-2\pi i j \bar{x}} \end{aligned}$$

which indicates that

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

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

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

$\mathcal{T}_j = [t_{j-1}, t_j]$ are the Principle Intervals

$\tilde{\phi}_j$ are the Principle Modes

Principle Interval Decomposition (4)

We would like to choose the time intervals and modes so that the following estimate holds

$$\int_{\mathcal{T}} \|y(t, \cdot) - y_\ell(t, \cdot)\|^2 dt \leq \varepsilon \int_{\mathcal{T}} \|y(t, \cdot)\|^2 dt.$$

One means to enforce this estimate is to require

$$\int_{\mathcal{I}_j} \|y(t, \cdot) - y_\ell(t, \cdot)\|^2 dt \leq \varepsilon \int_{\mathcal{I}_j} \|y(t, \cdot)\|^2 dt.$$

Principle Interval Decomposition (5)

There are practical limits on the size of ε to keep the basis size small.

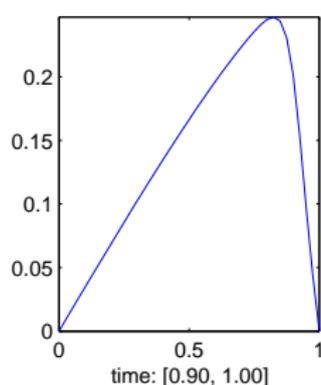
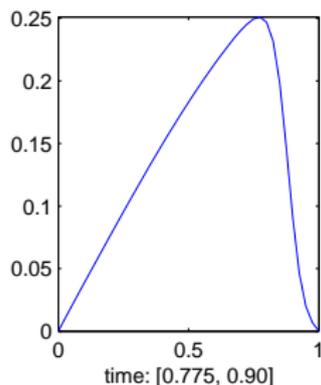
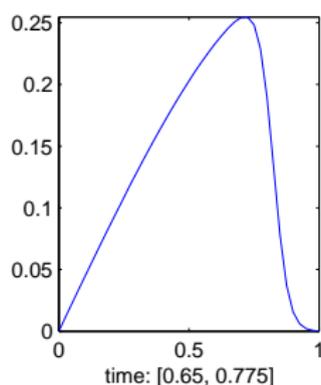
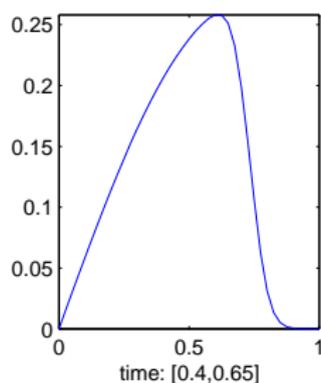
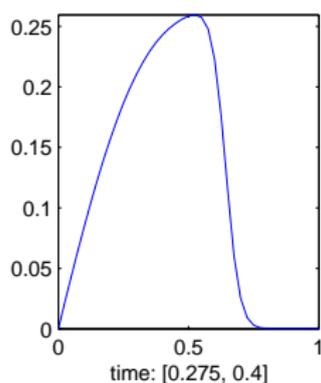
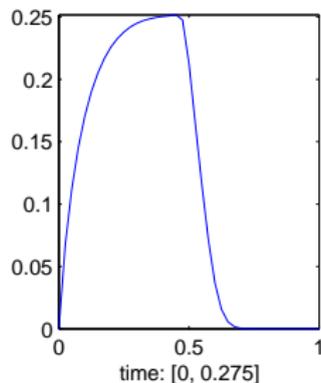
Once selected, the computation of principle intervals and modes follows a similar approach to POD. They satisfy the minimization problem

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

- 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

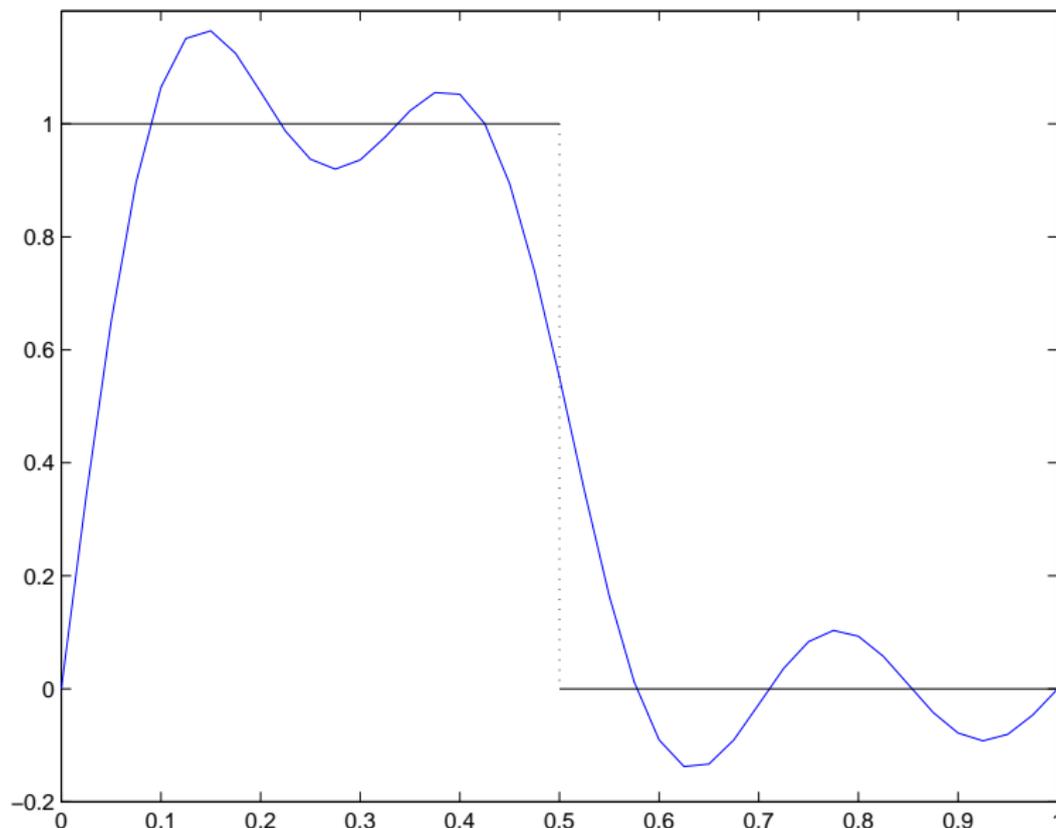
PID: Burgers Equation Example

Basis 3: Principle Interval Decomposition



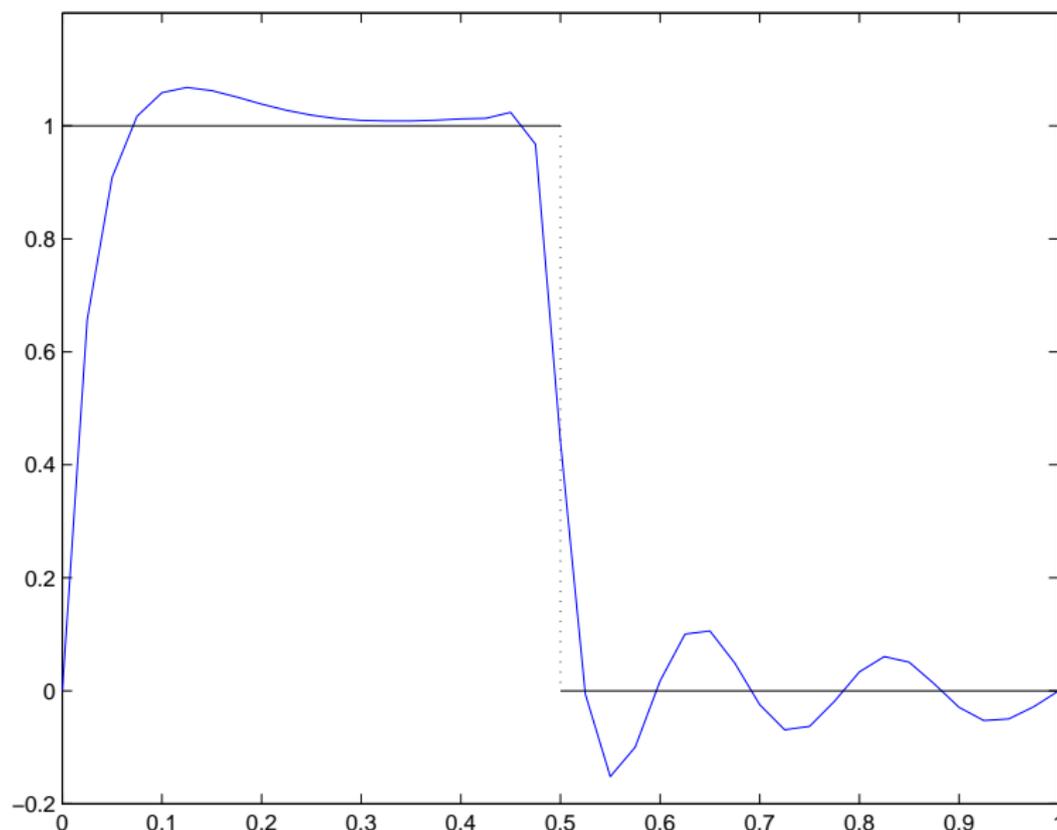
Example: Burgers Equation

Basis 1: Projected initial conditions, $r = 6$



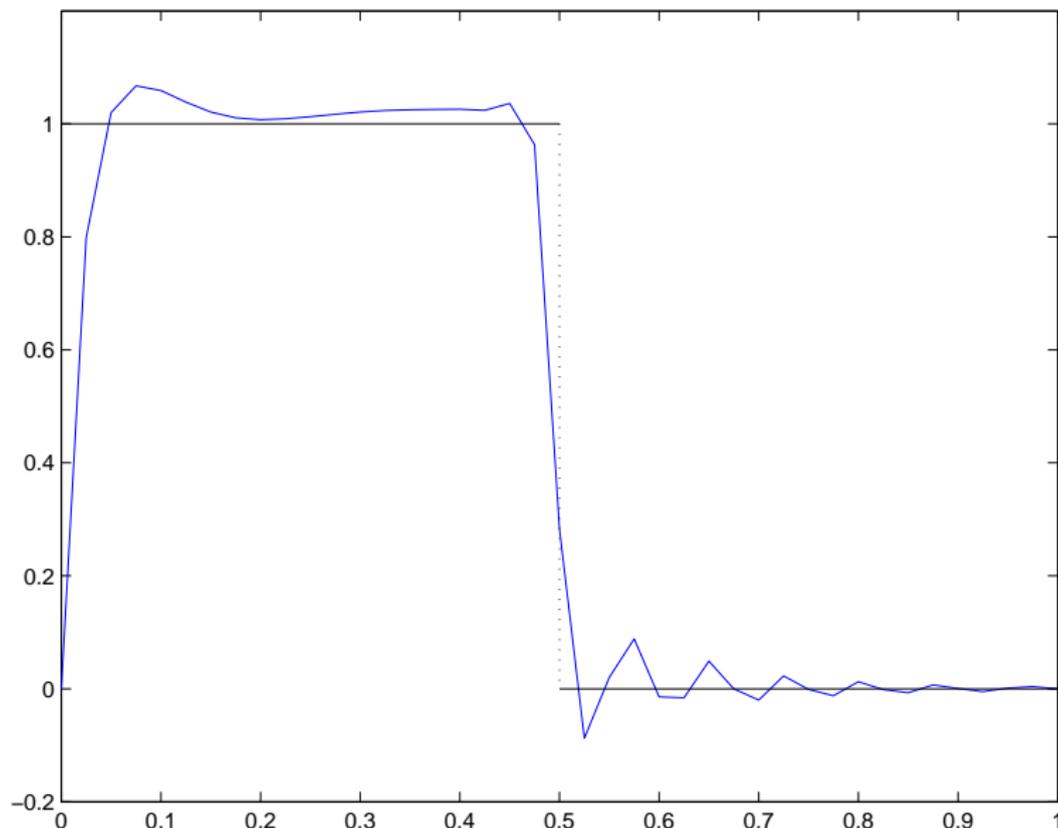
Example: Burgers Equation

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 $\varepsilon \rightarrow 0$, 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 Tesselations [BDGL03]

K-Means Clustering

Updating KLE/POD Bases (1)

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.

Updating KLE/POD Bases (2)

Let $\{\mathbf{y}(t_1), \mathbf{y}(t_2), \dots\}$ be given data and let

$$\mathbf{Y}_k = [\mathbf{y}(t_1) \ \mathbf{y}(t_2) \ \dots \ \mathbf{y}(t_k)] \in \mathbb{R}^{n \times k}$$

Algorithm: [GSA03]

Let $\mathbf{Y}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$ with $\mathbf{U}_k \in \mathbb{R}^{n \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$ and $\mathbf{V}_k \in \mathbb{R}^{k \times k}$.

$$\mathbf{Y}_k \mathbf{Y}_k^T = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T \mathbf{V}_k \Sigma_k \mathbf{U}_k^T = \underbrace{\mathbf{U}_k \Sigma_k}_{\hat{\mathbf{Y}}_k} \underbrace{\Sigma_k \mathbf{U}_k^T}_{\hat{\mathbf{Y}}_k^T} \equiv \hat{\mathbf{Y}}_k \hat{\mathbf{Y}}_k^T$$

Updating KLE/POD Bases (3)

Let

$$\mathbf{Y}_r = [\mathbf{y}(t_{k+1}) \cdots \mathbf{y}(t_{k+r})] \in \mathbb{R}^{n \times r}$$

be r additional snapshots.

Decompose \mathbf{Y}_r as

$$\mathbf{Y}_r = \mathbf{U}_k \Gamma + \mathbf{U}_k^\perp \Theta$$

where

$$\Gamma \in \mathbb{R}^{k \times r}, \quad \Theta \in \mathbb{R}^{r \times r}, \quad \text{and} \quad \mathbf{U}_k^\perp \in \mathbb{R}^{n \times r}$$

and

$$\mathbf{U}_k \mathbf{U}_k^\perp = \mathbf{0} \quad \text{and} \quad (\mathbf{U}_k^\perp)^T \mathbf{U}_k^\perp = \mathbf{I}_{r \times r}.$$

Updating KLE/POD Bases (4)

We can now write

$$\mathbf{Y}_{k+r} = \left[\hat{\mathbf{Y}}_k \mathbf{Y}_r \right] = \left[\mathbf{U}_k \mathbf{U}_k^\perp \right] \underbrace{\left[\begin{array}{cc} \Sigma_k & \Gamma \\ \mathbf{0} & \Theta \end{array} \right]}_{\hat{\mathbf{S}}}.$$

Let

$$\hat{\mathbf{S}} = \mathbf{T} \Sigma_{k+r} \hat{\mathbf{V}}^T.$$

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

$$\mathbf{Y}_{k+r} = \left[\mathbf{U}_k \mathbf{U}_k^\perp \right] \mathbf{T} \Sigma_{k+r} \hat{\mathbf{V}}^T \equiv \mathbf{U}_{k+r} \Sigma_{k+r} \hat{\mathbf{V}}^T.$$

We only need to keep $\hat{\mathbf{Y}}_{k+r} = \mathbf{U}_{k+r} \Sigma_{k+r}$.

Two-Step Process

Basis Selection:

- KLE/POD
- PID
- Sampling (CVT)

Model Construction:

- Galerkin
- Nonlinear-Galerkin
- Multiscale modeling:
 - ▶ LES, Patch Dynamics, Homogenization

KLE/POD Galerkin (1)

Given a low dimensional basis

$$\{\phi_1(\cdot), \phi_2(\cdot), \dots, \phi_r(\cdot)\}$$

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 $\mathcal{P}^r = \text{span}\{\phi_1, \dots, \phi_r\}$.

KLE/POD Galerkin (2)

To parallel our earlier discussion, we first consider the finite dimensional case.

Consider the ordinary differential equation

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^n$$

($r \ll n$).

Substitute the expression

$$\mathbf{y}_\ell(t) = \sum_{j=1}^r a_j(t) \phi_j$$

into the ODE and project the equations onto \mathcal{P}^r .

KLE/POD Galerkin (3)

This leads to

$$\sum_{j=1}^r \dot{a}_j(t) \phi_j = \mathbf{f} \left(\sum_{j=1}^r a_j(t) \phi_j \right) \quad \sum_{j=1}^r a_j(0) \phi_j = \mathbf{y}_0$$

and then

$$\dot{a}_i(t) = \left\langle \mathbf{f} \left(\sum_{j=1}^r a_j(t) \phi_j \right), \phi_i \right\rangle \quad a_i(0) = \langle \mathbf{y}_0, \phi_i \rangle, \quad i = 1, \dots, r,$$

since

$$\langle \phi_i, \phi_j \rangle = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases} .$$

KLE/POD Galerkin (4)

For the special case of linear, autonomous ODEs,

$$\mathbf{f}(\mathbf{y}) = \mathbf{A}\mathbf{y},$$

and the amplitude coefficient equations are

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

which simplifies to

$$\dot{\mathbf{a}}(t) = \underbrace{\Phi^T \mathbf{A} \Phi}_{r \times r} \mathbf{a}(t) \quad \mathbf{a}(0) = \Phi^T \mathbf{y}_0$$

where

$$\Phi = \underbrace{[\phi_1 | \phi_2 | \cdots | \phi_r]}_{n \times r}.$$

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

$$\langle y_t^N, v \rangle = -\langle y^N y_x^N, v \rangle - \varepsilon \langle y_x^N, v_x \rangle$$

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

$$\langle y^N(0, \cdot) - y_0(\cdot), v(\cdot) \rangle = 0 \quad \text{for all } v \in S^h.$$

KLE/POD Galerkin (6)

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

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

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

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

where

$$K_{ij} = \int_{\Omega} \phi'_j(x) \phi'_i(x) dx.$$

KLE/POD Galerkin (7)

For the nonlinear term $N(\mathbf{a})$, we define a sequence of matrices

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

then

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

Thus, the equation for the amplitude coefficients is

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

with

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

KLE/POD Galerkin (8)

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.

KLE/POD Galerkin (9)

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

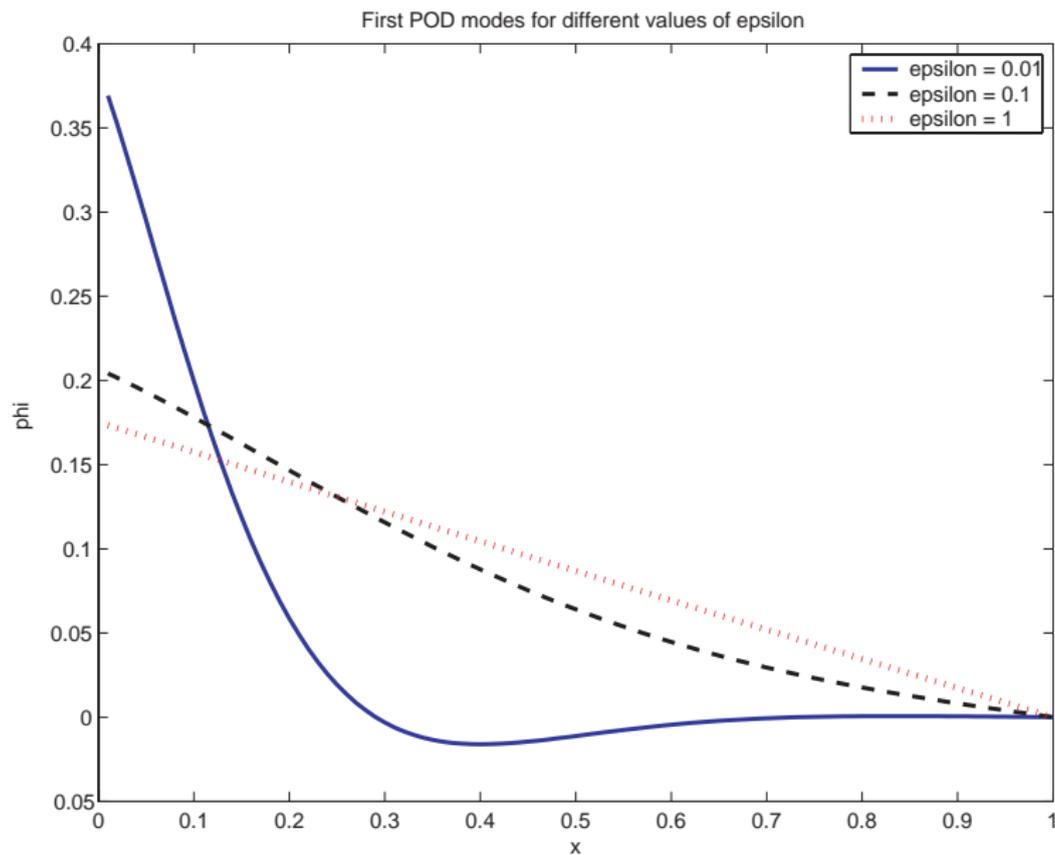
$$y(0, x) = 0, \quad y(t, 0) = \sin(t), \quad \text{and} \quad y(t, 1) = 0.$$

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



Snapshot Selection

Our approach:

We want to find ϕ to maximize the time averaged and ε 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

$$\begin{aligned} R^s(x, \bar{x}) &= \frac{1}{\varepsilon_2 - \varepsilon_1} \int_{\varepsilon_1}^{\varepsilon_2} \frac{1}{T} \int_{\mathcal{T}} y(t, x; \varepsilon) y(t, \bar{x}; \varepsilon) dt d\varepsilon \\ &\approx \frac{1}{T} \sum_{k=1}^g w_k \int_{\mathcal{T}} y(t, x; \varepsilon_k) y(t, \bar{x}; \varepsilon_k) dt. \end{aligned}$$

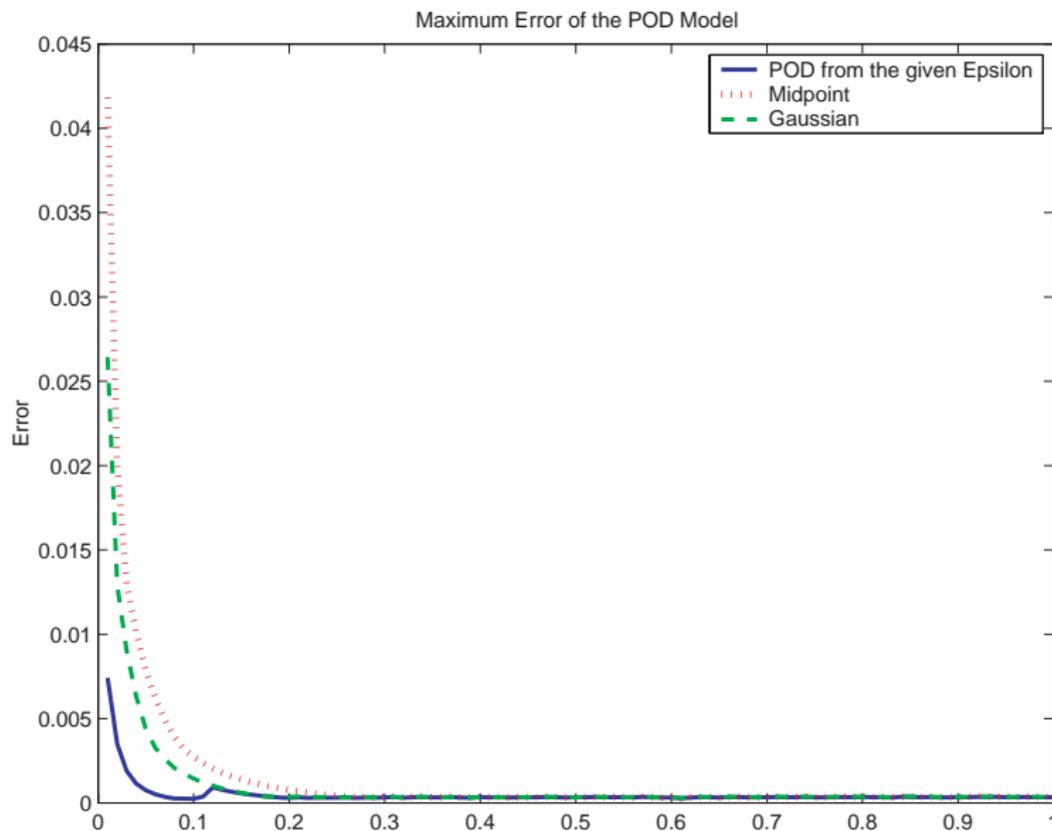
To compute coefficients of the POD basis vectors, define

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

and find eigenvectors of \mathbf{R}^s ...

Snapshot Selection

Comparison of Midpoint and Gauss quadrature rules



A Parallel KLE/POD Algorithm

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

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

(1) Initialization.

- (a) Calculate the dominant m right singular vectors of local $\mathbf{J}_k \mathbf{W}_k$: $\mathbf{v}_k^{(1)}$.
- (b) All-to-all send of local $\mathbf{v}_k^{(1)}$;
Receive $\{\mathbf{v}_i^{(1)}\}_{i \neq k}$ from other processors;
Assemble the starting block :
$$\mathbf{V}^{(1)} = \begin{bmatrix} \mathbf{v}_1^{(1)} & \mathbf{v}_2^{(1)} & \dots & \mathbf{v}_{n_p}^{(1)} \end{bmatrix}$$

A Parallel KLE/POD Algorithm

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

(2) Start Iteration.

For $j = 1, \dots, J_{max}$

(a) Calculate

$$\mathbf{W}_k^{(j)} = (\mathbf{J}_k \mathbf{W}_k \mathbf{V}^{(j)})^T \mathbf{J}_k \mathbf{W}_k \mathbf{V}^{(j)}.$$

(b) All-to-all send of local $\mathbf{W}_k^{(j)}$;

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

Sum blocks to get

$$\mathbf{W}^{(j)} = (\mathbf{W} \mathbf{V}^{(j)})^T \mathbf{W} \mathbf{V}^{(j)}.$$

(c) Calculate locally:

$$\mathbf{W}^{(j)} = \mathbf{Z}_1 \boldsymbol{\Sigma}_1^2 \mathbf{Z}_1^T;$$

$$\hat{\mathbf{U}}_k^{(1)} = \mathbf{W}_1 \mathbf{V}^{(1)} \mathbf{Z}_1 \boldsymbol{\Sigma}_1^{-1}.$$

A Parallel KLE/POD Algorithm

