SAND2012-7087P

Polynomial Chaos based uncertainty propagation

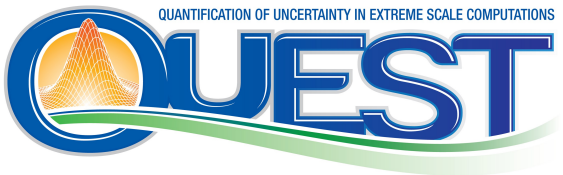
Intrusive and Non-Intrusive Methods

Bert Debusschere[†], Habib Najm, Khachik Sargsyan, Cosmin Safta

[†] **bjdebus@sandia.gov**

Sandia National Laboratories,
Livermore, CA

USC UQ Summer School



Acknowledgement

Omar Knio — Duke University, Raleigh, NC
Roger Ghanem — University of Southern California, Los Angeles, CA
Olivier Le Maître — LIMSI-CNRS, Orsay, France
and many others ...

This work was supported by:

- US Department of Energy (DOE), Office of Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC) and Applied Mathematics Research (AMR) programs.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The Goals of this lecture

- Provide broad introduction to Polynomial Chaos-based forward UQ
  - Propagate uncertainties from inputs to model prediction
  - Only parametric uncertainties covered here, no model uncertainty
- What you should take away from this lecture
  - An understanding of Polynomial Chaos expansions to represent random variables
  - An understanding of intrusive and non-intrusive uncertainty propagation methods
  - An appreciation for some of the challenges and associated mitigation approaches in forward UQ
- While this overview is broad, it is definitely not exhaustive

Outline

Polynomial Chaos Expansions (PCEs)

- Representation of random variables with PCEs
- Convergence
- Galerkin projection
- Basis types

One-Dimensional Hermite Polynomials

$$\begin{aligned}\psi_0(x) &= 1 \\ \psi_k(x) &= (-1)^k e^{x^2/2} \frac{\mathrm{d}^k}{\mathrm{d}x^k} e^{-x^2/2}, \qquad k = 1, 2, \ldots\end{aligned}$$

$$\psi_1(x) = x, \quad \psi_2(x) = x^2 - 1, \quad \psi_3(x) = x^3 - 3x, \ \ldots$$

The Hermite polynomials form an orthogonal basis over $[-\infty, \infty]$ with respect to the inner product

$$\langle \psi_i \psi_j \rangle \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi_i(x) \psi_j(x) w(x) dx = \delta_{ij} \left\langle \psi_i^2 \right\rangle$$

where $w(x)$ is the weight function

$$w(x) = e^{-x^2/2}$$

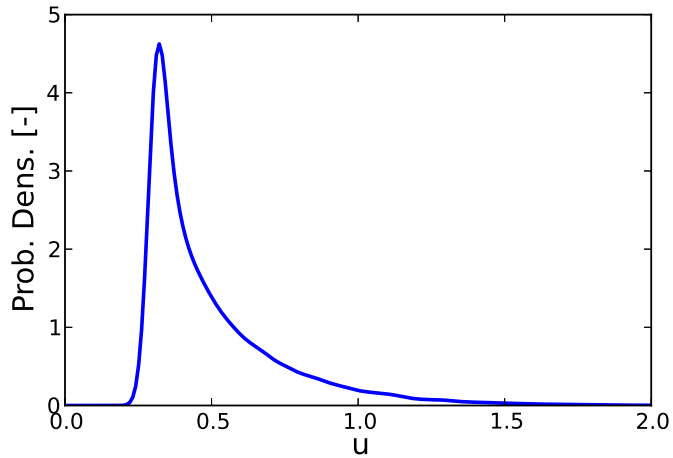
Note that $\frac{e^{-x^2/2}}{\sqrt{2\pi}}$ is the density of a standard normal random variable

One Dimensional Polynomial Chaos Expansion

Consider:

$$u = \sum_{k=0}^{P} u_k \psi_k(\xi)$$

- u: Random Variable (RV) represented with 1D PCE
- u_k: PC coefficients (deterministic)
- ψ_k: 1D Hermite polynomial of order k
- ξ: Gaussian RV



$$u = 0.5 + 0.2\psi_1(\xi) + 0.1\psi_2(\xi)$$

A random quantity is represented with an expansion consisting of functions of random variables multiplied with deterministic coefficients

- Set of deterministic PC coefficients fully describes RV
- Separates randomness from deterministic dimensions

Multidimensional Hermite Polynomials

The multidimensional Hermite polynomial $\Psi_i(\xi_1, \ldots, \xi_n)$ is a tensor product of the 1D Hermite polynomials, with a suitable multi-index $\boldsymbol{\alpha^i} = (\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i)$,

$$\Psi_i(\xi_1, \ldots, \xi_n) \quad = \quad \prod_{k=1}^{n} \psi_{\alpha_k^i}(\xi_k)$$

For example, 2D Hermite polynomials:

i	p	Ψ_i
0	0	1
1	1	ξ_1
2	1	ξ_2
3	2	$\xi_1^2 - 1$
4	2	$\xi_1 \xi_2$
5	2	$\xi_2^2 - 1$
...	...	...

Multidimensional Inner Products — Orthogonality

$$\langle \Psi_i \Psi_j \rangle \equiv \int \dots \int \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) g(\xi_1) g(\xi_2) \cdots g(\xi_n) \mathrm{d}\xi_1 \mathrm{d}\xi_2 \cdots \mathrm{d}\xi_n$$

$$= \prod_{k=1}^{n} \left\langle \psi_{\alpha_k^i}(\xi_k) \psi_{\alpha_k^j}(\xi_k) \right\rangle = \delta_{ij} \left\langle \Psi_i^2 \right\rangle$$

where,

$$g(\xi) = \frac{e^{-\xi^2/2}}{\sqrt{2\pi}}$$

such that,

$$u = \sum_{k=0}^{P} u_k \Psi_k \quad \Rightarrow \quad \langle \Psi_i u \rangle = \sum_{k=0}^{P} u_k \langle \Psi_i \Psi_k \rangle = u_i \left\langle \Psi_i^2 \right\rangle$$

$$\Rightarrow \quad u_i = \frac{\langle u \Psi_i \rangle}{\left\langle \Psi_i^2 \right\rangle}$$

Multidimensional Polynomial Chaos Expansion

Consider:

$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi_1, \ldots, \xi_n)$$

- u: Random Variable (RV) represented with multi-D PCE
- u_k: PC coefficients (deterministic)
- Ψ_k: Multi-D Hermite polynomials up to order p
- ξ_i: Gaussian RV
- n: Dimensionality of stochastic space
- $P+1$: Number of PC terms: $P+1 = \frac{(n+p)!}{n!p!}$

The number of stochastic dimensions represents the number of independent inputs, degrees of freedom that affect the random variable u

- E.g. one stochastic dimension per uncertain model parameter
- Contributions from each uncertain input can be identified
- Compact representation of a random variable and its dependencies

More Formally: Probability Spaces and Random Variables

- Let $(\Theta, \mathfrak{S}, \mathbb{P})$ be a probability space
  - Θ is an event space
  - $\mathfrak{S}$ is a σ-algebra on Θ
  - $\mathbb{P}$ is a probability measure on $(\Theta, \mathfrak{S})$
- Random variables are functions $X : \Theta \to \mathbb{R}$ with a measure corresponding to their image:
  - if $X^{-1}(A) \in \mathfrak{S}$, then define $\mu(A) = \mathbb{P}(X^{-1}(A))$.
  - $p(x) = d\mu/dx$: the density of the random variable X (with respect to Lebesgue measure on $\mathbb{R}$).
  - Expectation: $\langle f \rangle = \int f \, d\mu = \int f \, p(x) \, dx$
- Let $\xi : \Theta \to \mathbb{R}^N$ such that for $i = 1, \ldots, N$ each $\xi_i : \Theta \to \mathbb{R}$, be a set of random variables
- $\mathfrak{S}(\xi)$: σ-algebra generated by the set ξ of random variables
- $L^2(\Theta, \mathfrak{S}(\xi), \mathbb{P})$: Hilbert space of real-valued random variables defined on $(\Theta, \mathfrak{S}(\xi), \mathbb{P})$ with finite second moments

[Ernst et al. 2011]

Convergence Theorem

A random variable $u(\theta)$ in $L^2(\Theta, \mathfrak{S}(\xi), \mathbb{P})$ can be described by a Polynomial Chaos (PC) expansion in terms of:
the infinite-dimensional i.i.d. Gaussian basis $\boldsymbol{\xi} = \{\xi_i(\theta)\}_{i=1}^{\infty}$;

$$\begin{aligned} u(\theta) &= a_0\Gamma_0 + \sum_{i_1=1}^{\infty} a_{i_1}\Gamma_1(\xi_{i_1}(\theta)) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty} a_{i_1 i_2}\Gamma_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta)) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}\sum_{i_3=1}^{\infty} a_{i_1 i_2 i_3}\Gamma_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta)) + \ldots \end{aligned}$$

where Γ_p is the Polynomial Chaos of order p, $\Gamma_0 = 1$, and

$$\Gamma_p(\xi_{i_1}, \ldots, \xi_{i_p}) = (-1)^p e^{\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}} \frac{\partial^p}{\partial\xi_{i_1}\ldots\partial\xi_{i_p}} e^{-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}}$$

Notes on the PC construction

- The Polynomial Chaoses are by construction orthogonal with respect to the Gaussian probability measure
- They are thus identical with the corresponding multidimensional Hermite Polynomials
- The first four PCs are given by

$$
\begin{aligned}
\Gamma_0 &= 1 \\
\Gamma_1(\xi_i) &= \xi_i \\
\Gamma_2(\xi_{i_1}, \xi_{i_2}) &= \xi_{i_1}\xi_{i_2} - \delta_{i_1 i_2} \\
\Gamma_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) &= \xi_{i_1}\xi_{i_2}\xi_{i_3} - \xi_{i_1}\delta_{i_2 i_3} - \xi_{i_2}\delta_{i_1 i_3} - \xi_{i_3}\delta_{i_1 i_2} \\
&\dots
\end{aligned}
$$

[R.G. Ghanem and P.D. Spanos, 1991]

A more compact notation ... and finite dimensionality and order

- An L_2 random variable $u(\boldsymbol{x}, t, \theta)$ can be described by a PC expansion in terms of:
  - Hermite polynomials Ψ_k, $k = 1, \ldots, \infty$;
  - the associated infinite-dimensional Gaussian basis $\{\xi_i(\theta)\}_{i=1}^{\infty}$;
  - spectral mode strengths $u_k(\boldsymbol{x}, t)$, $k = 1, \ldots, \infty$.
- Truncated to finite dimension n and order p, the PC expansion for u is written as

$$u(\boldsymbol{x}, t, \theta) \simeq \sum_{k=0}^{P} u_k(\boldsymbol{x}, t)\Psi_k(\boldsymbol{\xi}(\theta))$$

where $\boldsymbol{\xi}(\theta) = \{\xi_1(\theta), \cdots, \xi_n(\theta)\}$, and $P + 1 = \frac{(n+p)!}{n!p!}$

Generalized Polynomial Chaos

PC Type	Domain	Density $w(\xi)$	Polynomial	Free parameters
Gauss-Hermite	$(-\infty, +\infty)$	$\frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}}$	Hermite	none
Legendre-Uniform	$[-1, 1]$	$\frac{1}{2}$	Legendre	none
Gamma-Laguerre	$[0, +\infty)$	$\frac{x^\alpha e^{-\xi}}{\Gamma(\alpha+1)}$	Laguerre	$\alpha > -1$
Beta-Jacobi	$[-1, 1]$	$\frac{(1+\xi)^\alpha (1-\xi)^\beta}{2^{\alpha+\beta+1} B(\alpha+1,\beta+1)}$	Jacobi	$\alpha > -1, \beta > -1$

Inner product: $\langle \psi_i \psi_j \rangle \equiv \int_a^b \psi_i(\xi)\psi_j(\xi)w(\xi)d\xi$

- Wiener-Askey scheme provides a hierarchy of possible continuous PC bases, see Xiu and Karniadakis, SISC, 2002.
  - Legendre-Uniform PC is a special case of Beta-Jacobi PC
  - Beta-Jacobi PC allows tailored accuracy of the PC representation across the domain
- Input parameter domain often dictates the most convenient choice of PC
- Polynomials functions can also be tailored to be orthogonal w.r.t. chosen, arbitrary density

Postprocessing / Analysis

- Moments
- Plotting PDFs of RVs represented with PCEs
- When is a PCE accurate enough?

Moments of RVs described with PCEs

$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$

- Expectation: $\langle u \rangle = u_0$
- Variance σ^2

$$\begin{aligned}
\sigma^2 &= \left\langle (u - \langle u \rangle)^2 \right\rangle \\
&= \left\langle (\sum_{k=1}^{P} u_k \Psi_k(\xi))^2 \right\rangle \\
&= \left\langle \sum_{k=1}^{P} \sum_{j=1}^{P} u_j u_k \Psi_j(\xi) \Psi_k(\xi) \right\rangle \\
&= \sum_{k=1}^{P} \sum_{j=1}^{P} u_j u_k \left\langle \Psi_j(\xi) \Psi_k(\xi) \right\rangle \\
&= \sum_{k=1}^{P} u_k^2 \left\langle \Psi_k(\xi)^2 \right\rangle
\end{aligned}$$

Plotting PDFs of RVs corresponding to PCEs

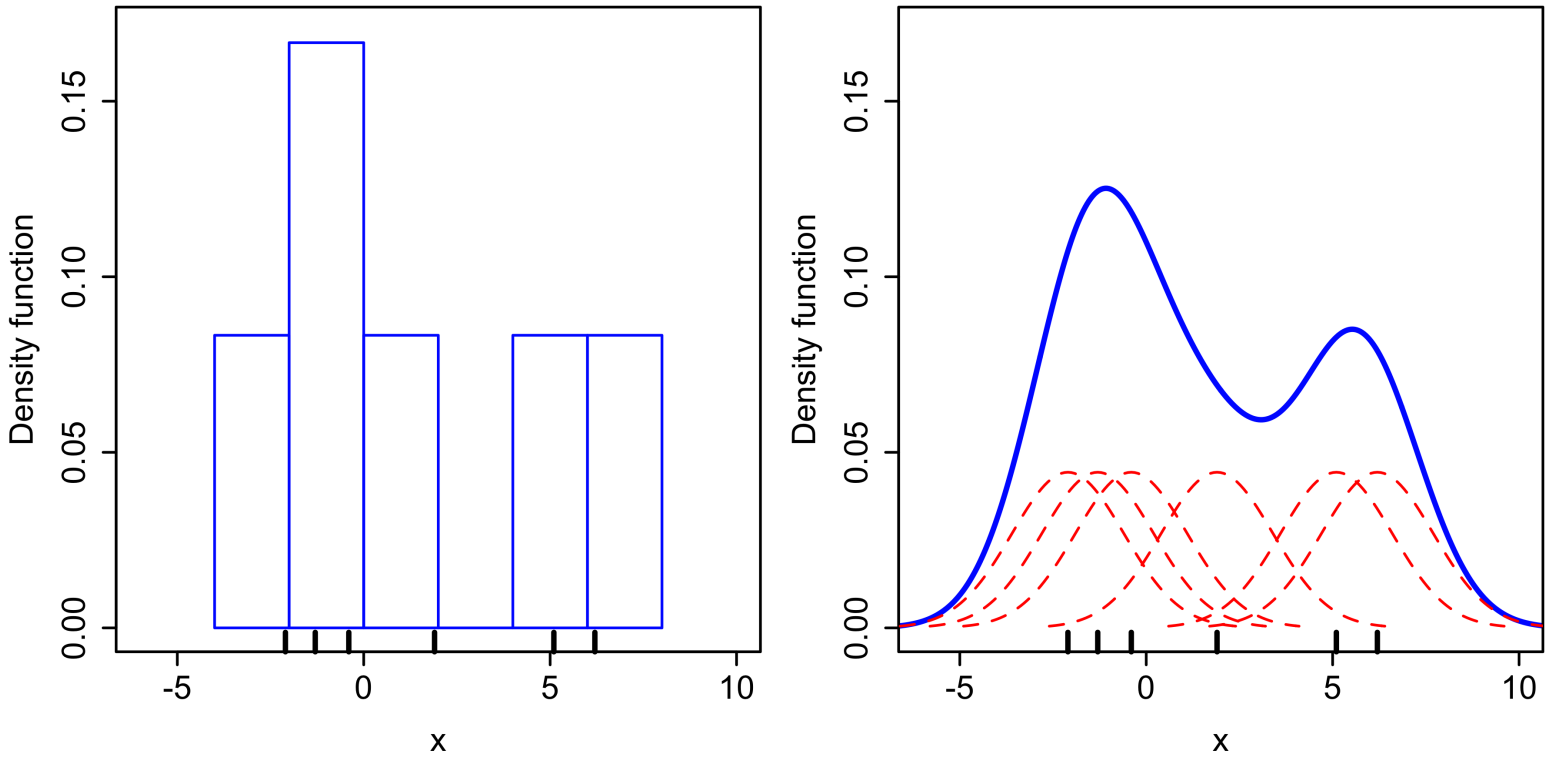
$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$

- Analytical formula formula for PDF(u) exists
  - Involves polynomial root finding, and is hard to generalize to multi-D
- PCE is cheap to sample
  - Brute-force sampling and bin samples into histogram
  - Use Kernel Density Estimation (KDE) to get smoother PDF with fewer samples u_i

$$\mathrm{PDF}(u) = \frac{1}{N_s h} \sum_{i=1}^{N_s} K\left(\frac{u - u_i}{h}\right)$$

K is the kernel, h is the bandwidth

Comparison of histograms and KDE



- Bandwidth h needs to be chosen carefully to avoid oversmoothing

How do I know my PCE is converged?

- Approximation error in PCE is topic of a lot of research
- Rules of thumb:
  - Higher order PC coefficients should decay
  - Increase order until results no longer change
  - Not always fail-proof ...

Propagation of Uncertain Inputs Represented with PCEs

- Galerkin projection approaches project uncertain quantity onto space covered by PC basis functions
  - Relying on orthogonality of basis functions

$$u_k = \frac{\langle u\Psi_k \rangle}{\langle \Psi_k^2 \rangle}, \quad k = 0, \ldots, P$$

  - Residual orthogonal to space of basis functions
  - Two different approaches
    - Intrusive: project governing equations
    - Non-intrusive: project sampled model outpus
- Collocation approaches
  - Match PCE to random variable at chosen sample points

Intrusive Spectral Stochastic UQ Formulation: ODE Example

- Sample ODE with parameter λ:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \lambda u$$

- Let λ be uncertain; introduce $\xi \sim \mathcal{N}(0,1)$.
- Express λ and u using PCEs in ξ:

$$\lambda = \sum_{k=0}^{P} \lambda_k \Psi_k(\xi), \qquad u(t) = \sum_{k=0}^{P} u_k(t) \Psi_k(\xi)$$

- Substitute in ODE and apply a Galerkin projection on $\Psi_i(\xi)$,

Galerkin Projection on $\Psi_i(\xi)$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{k=0}^{P} u_k(t)\Psi_k(\xi)\right) = \left(\sum_{p=0}^{P}\lambda_p\Psi_p(\xi)\right)\left(\sum_{q=0}^{P} u_q(t)\Psi_q(\xi)\right)$$

$$\sum_{k=0}^{P}\frac{\mathrm{d}u_k(t)}{\mathrm{d}t}\Psi_k(\xi) = \sum_{p=0}^{P}\sum_{q=0}^{P}\lambda_p u_q(t)\Psi_p(\xi)\Psi_q(\xi)$$

$$\left\langle\sum_{k=0}^{P}\frac{\mathrm{d}u_k(t)}{\mathrm{d}t}\Psi_k(\xi)\Psi_i(\xi)\right\rangle = \left\langle\sum_{p=0}^{P}\sum_{q=0}^{P}\lambda_p u_q(t)\Psi_p(\xi)\Psi_q(\xi)\Psi_i(\xi)\right\rangle$$

$$\sum_{k=0}^{P}\frac{\mathrm{d}u_k(t)}{\mathrm{d}t}\left\langle\Psi_k(\xi)\Psi_i(\xi)\right\rangle = \sum_{p=0}^{P}\sum_{q=0}^{P}\lambda_p u_q(t)\left\langle\Psi_p(\xi)\Psi_q(\xi)\Psi_i(\xi)\right\rangle$$

$$\frac{\mathrm{d}u_i}{\mathrm{d}t}\left\langle\Psi_i^2\right\rangle = \sum_{p=0}^{P}\sum_{q=0}^{P}\lambda_p u_q\left\langle\Psi_p\Psi_q\Psi_i\right\rangle$$

Resulting Spectral ODE system

- $(P+1)$-dimensional ODE system

$$\frac{du_i}{dt} = \sum_{p=0}^{P}\sum_{q=0}^{P} \lambda_p u_q C_{pqi}, \quad i = 0, \ldots, P$$

  where $C_{pqi} = \langle \Psi_p \Psi_q \Psi_i \rangle / \langle \Psi_i^2 \rangle$
- The tensor C_{pqi} can be evaluated once and stored for any given PC order and dimension
- This tensor is sparse, i.e. many elements are zero

1D 4^{th}-Order C_{ijk} Example : Hermite polynomials

$\langle \Psi_i \Psi_j \Psi_k \rangle$	value
$\langle \Psi_0 \Psi_0 \Psi_0 \rangle$	1
$\langle \Psi_0 \Psi_1 \Psi_1 \rangle$	1
$\langle \Psi_0 \Psi_2 \Psi_2 \rangle$	2
$\langle \Psi_0 \Psi_3 \Psi_3 \rangle$	6
$\langle \Psi_0 \Psi_4 \Psi_4 \rangle$	24
$\langle \Psi_1 \Psi_1 \Psi_2 \rangle$	2
$\langle \Psi_1 \Psi_2 \Psi_3 \rangle$	6
$\langle \Psi_1 \Psi_3 \Psi_4 \rangle$	24
$\langle \Psi_2 \Psi_2 \Psi_2 \rangle$	8
$\langle \Psi_2 \Psi_2 \Psi_4 \rangle$	24
$\langle \Psi_2 \Psi_3 \Psi_3 \rangle$	36
$\langle \Psi_2 \Psi_4 \Psi_4 \rangle$	192
$\langle \Psi_3 \Psi_3 \Psi_4 \rangle$	216
$\langle \Psi_4 \Psi_4 \Psi_4 \rangle$	1728

k	$\langle \Psi_k^2 \rangle$
0	1
1	1
2	2
3	6
4	24

- $C_{ijk} = \langle \Psi_i \Psi_j \Psi_k \rangle \, / \, \langle \Psi_k^2 \rangle$
- and,

$$\begin{aligned} \langle \Psi_i \Psi_j \Psi_k \rangle &= \langle \Psi_i \Psi_k \Psi_j \rangle = \\ \langle \Psi_j \Psi_i \Psi_k \rangle &= \langle \Psi_j \Psi_k \Psi_i \rangle = \\ \langle \Psi_k \Psi_i \Psi_j \rangle &= \langle \Psi_k \Psi_j \Psi_i \rangle \end{aligned}$$

- with other not-reported $\langle \Psi_i \Psi_j \Psi_k \rangle$ zero

1D 4^{th}-Order C_{ijk} Example : Legendre polynomials

$\langle \Psi_i \Psi_j \Psi_k \rangle$	value
$\langle \Psi_0 \Psi_0 \Psi_0 \rangle$	1
$\langle \Psi_0 \Psi_1 \Psi_1 \rangle$	1/3
$\langle \Psi_0 \Psi_2 \Psi_2 \rangle$	1/5
$\langle \Psi_0 \Psi_3 \Psi_3 \rangle$	1/7
$\langle \Psi_0 \Psi_4 \Psi_4 \rangle$	1/9
$\langle \Psi_1 \Psi_1 \Psi_2 \rangle$	2/15
$\langle \Psi_1 \Psi_2 \Psi_3 \rangle$	3/35
$\langle \Psi_1 \Psi_3 \Psi_4 \rangle$	4/63
$\langle \Psi_2 \Psi_2 \Psi_2 \rangle$	2/35
$\langle \Psi_2 \Psi_2 \Psi_4 \rangle$	2/35
$\langle \Psi_2 \Psi_3 \Psi_3 \rangle$	4/105
$\langle \Psi_2 \Psi_4 \Psi_4 \rangle$	$\approx$0.029
$\langle \Psi_3 \Psi_3 \Psi_4 \rangle$	$\approx$0.026
$\langle \Psi_4 \Psi_4 \Psi_4 \rangle$	$\approx$0.018

k	$\langle \Psi_k^2 \rangle$
0	1
1	1/3
2	1/5
3	1/7
4	1/9

- $C_{ijk} = \langle \Psi_i \Psi_j \Psi_k \rangle \, / \, \langle \Psi_k^2 \rangle$
- and,

$$\begin{aligned} \langle \Psi_i \Psi_j \Psi_k \rangle &= \langle \Psi_i \Psi_k \Psi_j \rangle = \\ \langle \Psi_j \Psi_i \Psi_k \rangle &= \langle \Psi_j \Psi_k \Psi_i \rangle = \\ \langle \Psi_k \Psi_i \Psi_j \rangle &= \langle \Psi_k \Psi_j \Psi_i \rangle \end{aligned}$$

- with other not-reported $\langle \Psi_i \Psi_j \Psi_k \rangle$ zero

Pseudo-Spectral Construction–1

$$w = \lambda u^2 v, \qquad u = \sum_{k=0}^{P} u_k \Psi_k, \qquad \text{similarly for } \lambda \ \& \ v$$

Spectral:

$$\begin{aligned} w_i &= \left\langle \lambda u^2 v \right\rangle_i \\ &= \sum_{j=0}^{P} \sum_{k=0}^{P} \sum_{l=0}^{P} \sum_{m=0}^{P} \lambda_j u_k u_l v_m \left\langle \Psi_j \Psi_k \Psi_l \Psi_m \right\rangle_i , \quad i = 0, \ldots, P \end{aligned}$$

- The corresponding tensor of basis product expectations becomes too large to pre-compute and store

Pseudo-Spectral: Project each PC product onto a $(P+1)$-polynomial before proceeding further, thus:

Pseudo-Spectral Construction–2

$$\tilde{w} = uv \quad : \quad \tilde{w}_i = \langle uv \rangle_i = \sum_{j=0}^{P} \sum_{k=0}^{P} u_k v_j \langle \Psi_k \Psi_j \rangle_i , \quad i = 0, \ldots, P$$

$$\hat{w} = u\tilde{w} \quad : \quad \hat{w}_i = \langle u\tilde{w} \rangle_i = \sum_{j=0}^{P} \sum_{k=0}^{P} u_k \tilde{w}_j \langle \Psi_k \Psi_j \rangle_i , \quad i = 0, \ldots, P$$

$$w = \lambda\hat{w} \quad : \quad w_i = \langle \lambda\hat{w} \rangle_i = \sum_{j=0}^{P} \sum_{k=0}^{P} \lambda_k \hat{w}_j \langle \Psi_k \Psi_j \rangle_i , \quad i = 0, \ldots, P$$

- Aliasing errors
- Efficiency, and convenience

[Debusschere *et al.*, *SIAM J. Sci. Comp.*, 2005.]

Intrusive propagation through non-polynomial functions

Addition, subtraction, and product allow (pseudo-)spectral evaluation of all polynomial functions

How to propagate PC expansions ($\{u_k\} \Rightarrow \{v_k\}$) through transcendental functions

$$v = \frac{1}{u}, \quad v = \ln u, \quad \text{or} \quad v = e^u$$

- Use local polynomial approximations, e.g. Taylor series
  - E.g. $e^u = 1 + \frac{u}{1!} + \frac{u^2}{2!} + \frac{u^3}{3!} + \ldots$
  - Issues:
    - Convergence issues
    - High-order PC multiplications lead to aliasing
    - Instabilities
- Write system of equations for output
- Integration approach
- Borchardt-Gauss Algorithm: Arithmetic-Geometric Mean (AGM) series

[Debusschere et al., SISC 2005, McKale, Texas Tech, M.S. Thesis, 2011]

Inversion and division can be done without Taylor series

- Assume three stochastic variables u, v, and w

$$w = \frac{u}{v} \Rightarrow \qquad vw = u$$

- Mode k of the stochastic product

$$\langle vw \rangle_k = \sum_{m=0}^{P} \sum_{l=0}^{P} C_{klm} v_l \cdot w_m = u_k$$

- System of $P+1$ linear algebraic equations in w_m with known u_k and v_l,

$$V_{km} = \sum_{l=0}^{P} C_{klm} v_l, \quad \mathbf{Vw} = \mathbf{u}$$

- More robust than Taylor series expansion for $1/u$
- What about the condition number of $\mathbf{V}$?

Integration approach for non-polynomial functions

- Consider the ODE $\frac{du}{dx} = u$, with solution $u = e^x$
- e^x can be obtained from

$$du = udx \quad \Rightarrow \quad e^x - e^{x_o} = \int_{x_o}^{x} u \, \mathrm{d}x$$

- Similarly for e^{-x^2}, and $\ln(x)$

$$e^{-x^2} - e^{-x_o^2} = \int_{x_o}^{x} -2xu \, \mathrm{d}x, \qquad \ln(x) - \ln(x_o) = \int_{x_o}^{x} \frac{\mathrm{d}x}{x}$$

- Agrees well with directly sampled pdf if PC order is high enough to resolve pdf of solution

A More General Integration approach for irrational functions

- To evaluate $u(x)$, $x = \sum_{k=0}^{P} x_k \Psi_k$, $u = \sum_{k=0}^{P} u_k \Psi_k$,
  - use a deterministic IC x_a such that $u(x_a)$ is known
  - express $\dot{u} = \mathrm{d}u/\mathrm{d}x = f(u, x)$;
    ... **require**: f is a rational function
    ... ensures that $(\dot{u})_k$ are found from the u_k and x_k coeffs
  - evaluate the integral:

$$u_k(x_b) - u_k(x_a) = \sum_{j=0}^{P} \int_{(x_a)_j}^{(x_b)_j} \sum_{i=0}^{P} C_{ijk} (\dot{u})_i \mathrm{d}x_j$$

- ok for e^x, e^{x^2}, and $\ln(x)$, with $\dot{u} = u$, $2xu$, and $1/x$ resp.
  - but not for $e^{\sin x}$, with $\dot{u} = u \cos x$
- CPU-intensive, only slightly more robust than Taylor series

Pseudo-spectral overloading of operations

- Construction allows for a general representation using pseudo-spectral (PS) overloaded operations.
  - E.g. multiplication operation '$*$'

$$w = \lambda * u * u * v$$

  - Each deterministic function multiplication is transformed into a corresponding polynomial chaos product
- Potential meta-code: take a general deterministic code function $F(u)$, produce a pseudo-spectral stochastic function $\tilde{F}(\tilde{u})$
- Possibility of transforming legacy deterministic code into corresponding pseudo-spectral stochastic code.
- UQToolkit: contains library of utilities for operations on random variables represented with PCEs

Uncertainty Quantification Toolkit (UQTk)

- A library of C++ and Matlab functions for propagation of uncertainty through computational models
- Mainly relies on spectral Polynomial Chaos Expansions (PCEs) for representing random variables and stochastic processes
- Target usage:
  - Rapid prototyping
  - Algorithmic research
  - Tutorials / Educational
- Version 1.0 released under the GNU Lesser General Public License
- Downloadable from `http://www.sandia.gov/UQToolkit/`

UQTk contents and development/release plans

- Currently released (`http://www.sandia.gov/UQToolkit/`)
  - C++ Tools for intrusive UQ with PCEs
- Under production, planned release Fall 2012
  - C++ Tools for non-intrusive UQ
  - Matlab tools for intrusive and non-intrusive UQ
  - Karhunen-Loève decomposition
  - Bayesian inference tools
  - Many more examples and documentation
- Under development
  - Adding support for multiwavelet based stochastic domain decompositions
  - Support for arbitrary basis types

PCEs in the UQToolkit

```
// Initialize PC class
int ord = 5;   // Order of PCE
int dim = 1;   // Number of uncertain parameters
PCSet myPCSet("ISP",ord,dim,"LU"); // Legendre-Uniform PCEs

// Initialize PC class
int ord = 5;   // Order of PCE
int dim = 1;   // Number of uncertain parameters
PCSet myPCSet("NISP",ord,dim,"LU"); // Legendre-Uniform PCEs
```

- Currently support Wiener-Hermite, Legendre-Uniform, and Gamma-Laguerre (limited), Jacobi-Beta (development version)
- PCSet class initializes PC basis type and pre-computes information needed for working with PC expansions

Operations on PCEs in the UQToolkit

```
// PC coefficients in double*
double* a = new double[npc];
double* b = new double[npc];
double* c = new double[npc];

// Initialization
a[0] = 2.0;
a[1] = 0.1;
...
// Perform some arithmetic
myPCSet.Subtract(a,b,c);
myPCSet.Prod(a,b,c);
myPCSet.Exp(a,c);
myPCSet.Log(a,c);
```

```
// PC coefficients in Arrays
Array1D<double> aa(npc,0.e0);
Array1D<double> ab(npc,0.e0);
Array1D<double> ac(npc,0.e0);

// Initialization
aa(0) = 2.0;
aa(1) = 0.1;
...
// Perform arithmetic
myPCSet.Subtract(aa,ab,ac);
myPCSet.Prod(aa,ab,ac);
myPCSet.Exp(aa,ac);
myPCSet.Log(aa,ac);
```

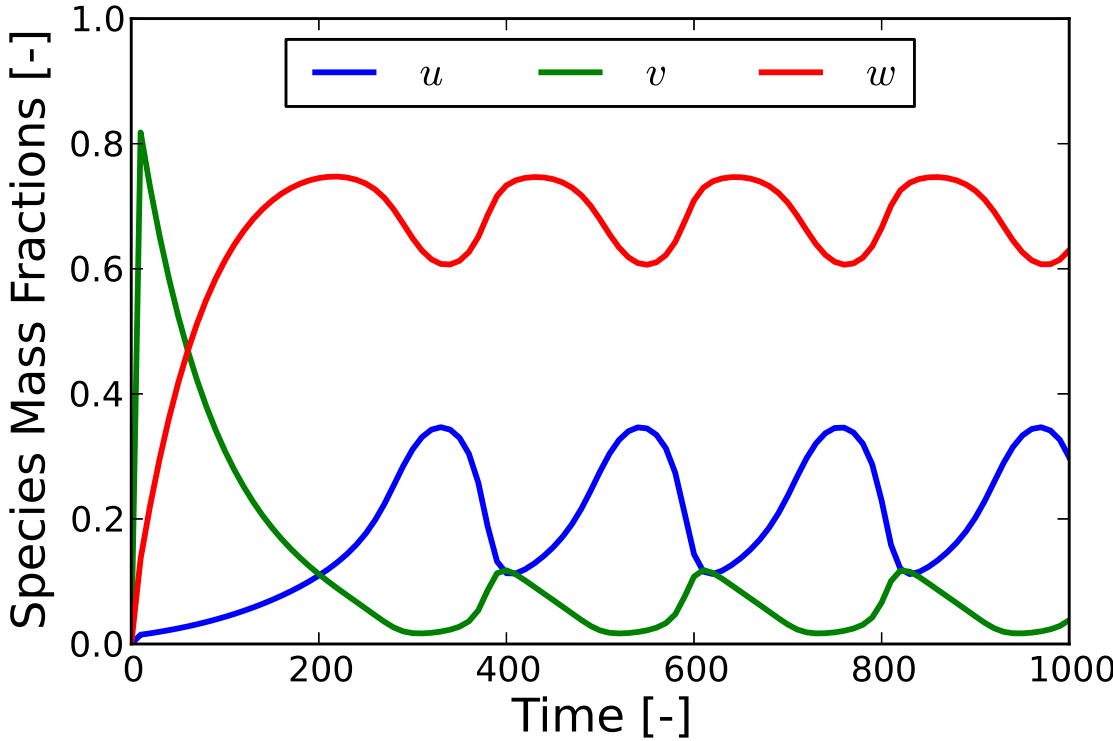
- PC coefficients are either stored in `double*` vectors or in more advanced custom `Array1D<double>` classes
- Functions can take either data type as argument

Surface Reaction Model

3 ODEs for a monomer (u), dimer (v), and inert species (w) adsorbing onto a surface out of gas phase.

$$\frac{du}{dt} = az - cu - 4duv$$
$$\frac{dv}{dt} = 2bz^2 - 4duv$$
$$\frac{dw}{dt} = ez - fw$$
$$z = 1 - u - v - w$$

$$u(0) = v(0) = w(0) = 0.0$$



Oscillatory behavior for $b \in [20.2, 21.2]$

$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad d = 1.0 \quad e = 0.36 \quad f = 0.016$

(Vigil *et al.*, Phys. Rev. E., 1996; Makeev *et al.*, J. Chem. Phys., 2002)

Surface Reaction Model: Intrusive Spectral Propagation (ISP) of Uncertainty

- Assume PCE for uncertain parameter b and for the output variables, u, v, w
- Substitute PCEs into the governing equations
- Project the governing equations onto the PC basis functions
  - Multiply with Ψ_k and take the expectation
- Apply pseudo-spectral approximations where necessary
- UQTk elementary operations

Surface Reaction Model: Specify PCEs for inputs and outputs

Represent uncertain inputs with PCEs with known coefficients:

$$b = \sum_{i=0}^{P} b_i \Psi_i(\xi)$$

Represent all uncertain variables with PCEs with unknown coefficients:

$$u = \sum_{i=0}^{P} u_i \Psi_i(\xi) \quad v = \sum_{i=0}^{P} v_i \Psi_i(\xi) \quad w = \sum_{i=0}^{P} w_i \Psi_i(\xi) \quad z = \sum_{i=0}^{P} z_i \Psi_i(\xi)$$

Surface Reaction Model: Substitute PCEs into governing equations and project onto basis functions

$$
\begin{aligned}
\frac{\mathrm{d}u}{\mathrm{d}t} &= az - cu - 4duv \\
\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} u_i\Psi_i &= a\sum_{i=0}^{P} z_i\Psi_i - c\sum_{i=0}^{P} u_i\Psi_i - 4d\sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \\
\left\langle \Psi_k \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} u_i\Psi_i \right\rangle &= \left\langle a\Psi_k \sum_{i=0}^{P} z_i\Psi_i \right\rangle - \left\langle c\Psi_k \sum_{i=0}^{P} u_i\Psi_i \right\rangle \\
&- \left\langle 4d\Psi_k \sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \right\rangle
\end{aligned}
$$

Surface Reaction Model: Reorganize terms

$$\frac{\mathrm{d}}{\mathrm{d}t} u_k \left\langle \Psi_k^2 \right\rangle = a z_k \left\langle \Psi_k^2 \right\rangle - c u_k \left\langle \Psi_k^2 \right\rangle - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \left\langle \Psi_i \Psi_j \Psi_k \right\rangle$$

$$\frac{\mathrm{d}}{\mathrm{d}t} u_k = a z_k - c u_k - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} u_k = a z_k - c u_k - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j C_{ijk}$$

- Triple products $C_{ijk} = \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle}$ can be pre-computed and stored for repeated use

Surface Reaction Model: Substitute PCEs into governing equations and project onto basis functions

$$
\begin{aligned}
\frac{\mathrm{d}v}{\mathrm{d}t} &= 2bz^2 - 4duv \\
\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} v_i\Psi_i &= 2\sum_{h=0}^{P} b_h\Psi_h \sum_{i=0}^{P} z_i\Psi_i \sum_{j=0}^{P} z_j\Psi_j - 4d\sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \\
\left\langle \Psi_k \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} v_i\Psi_i \right\rangle &= \left\langle 2\Psi_k \sum_{h=0}^{P} b_h\Psi_h \sum_{i=0}^{P} z_i\Psi_i \sum_{j=0}^{P} z_j\Psi_j \right\rangle \\
&- \left\langle 4d\Psi_k \sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \right\rangle
\end{aligned}
$$

Surface Reaction Model: Reorganize terms

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k \left\langle \Psi_k^2 \right\rangle = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j \left\langle \Psi_h \Psi_i \Psi_j \Psi_k \right\rangle - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \left\langle \Psi_i \Psi_j \Psi_k \right\rangle$$

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j \frac{\left\langle \Psi_h \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle} - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j D_{hijk} - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j C_{ijk}$$

- Pre-computing and storing the quad product D_{hijk} becomes cumbersome
- Use pseudo-spectral approach instead

Surface Reaction Model: Pseudo-Spectral approach for products

- Introduce auxiliary variable $g = z^2$

$$\begin{aligned} g &= z^2 \\ f = 2bz^2 &= 2bg \end{aligned}$$

$$\begin{aligned} g_k &= \sum_{i=0}^{P}\sum_{j=0}^{P} z_i z_j C_{ijk} \\ f_k &= 2\sum_{i=0}^{P}\sum_{j=0}^{P} b_i g_j C_{ijk} \end{aligned}$$

- Limits the complexity of computing product terms
  - Higher products can be computed by repeated use of the same binary product rule
- Does introduce errors if order of PCE is not large enough

Surface Reaction Model: UQTk implementation

```
// Build du/dt = a*z - c*u - 4.0*d*u*v
aPCSet.Multiply(z,a,dummy1);            // dummy1 = a*z
aPCSet.Multiply(u,c,dummy2);            // dummy2 = c*u
aPCSet.SubtractInPlace(dummy1,dummy2); // dummy1 = a*z - c*u
aPCSet.Prod(u,v,dummy2);                // dummy2 = u*v
aPCSet.MultiplyInPlace(dummy2,4.e0*d); // dummy2 = 4.0*d*u*v
aPCSet.Subtract(dummy1,dummy2,dudt);    // dudt = a*z - c*u - 4.0*d*u*v
```

- All operations are replaced with their equivalent intrusive UQ counterparts
- Results in a set of coupled ODEs for the PC coefficients
  - u, v, w, z represent vector of PC coefficients
- This set of equations is integrated to get the evolution of the PC coefficients in time

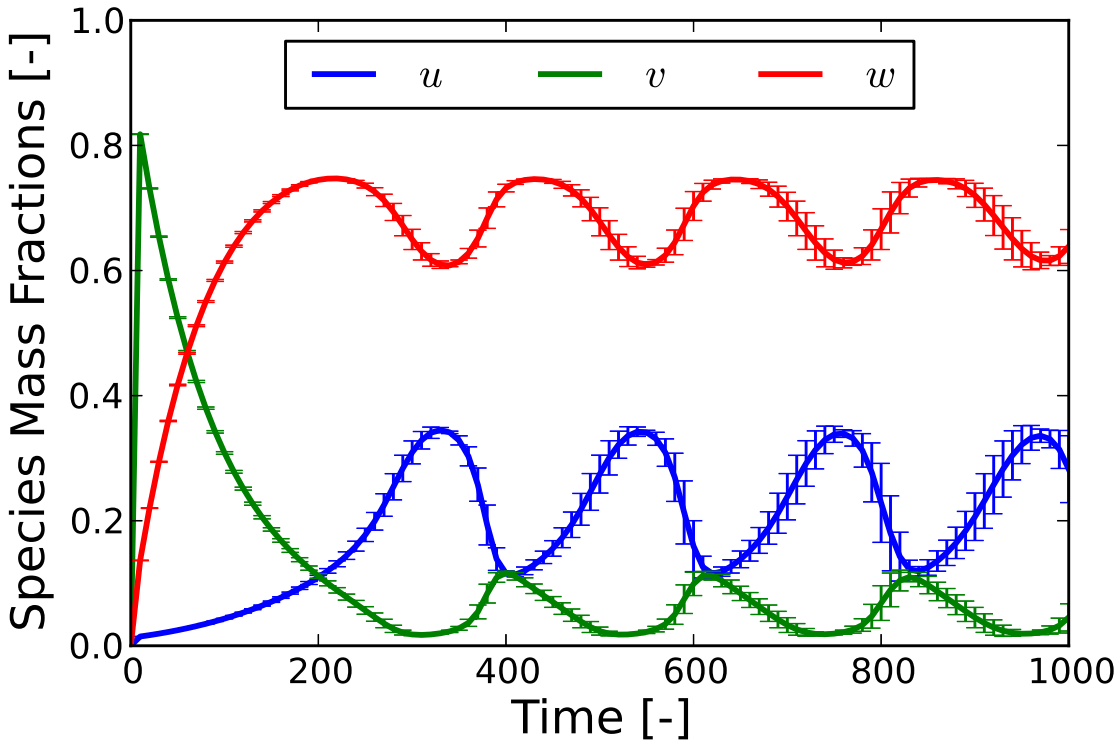
Surface Reaction Model: Second equation implementation

```
// Build dv/dt = 2.0*b*z*z - 4.0*d*u*v
aPCSet.Prod(z,z,dummy1);                 // dummy1 = z*z
aPCSet.Prod(dummy1,b,dummy2);            // dummy2 = b*z*z
aPCSet.Multiply(dummy2,2.e0,dummy1);     // dummy1 = 2.0*b*z*z
aPCSet.Prod(u,v,dummy2);                 // dummy2 = u*v
aPCSet.MultiplyInPlace(dummy2,4.e0*d); // dummy2 = 4.0*d*u*v
aPCSet.Subtract(dummy1,dummy2,dvdt);     // dvdt = 2.0*b*z*z - 4.0*d*u*v

// Build dw/dt = e*z - f*w
aPCSet.Multiply(z,e,dummy1);             // dummy1 = e*z
aPCSet.Multiply(w,f,dummy2);             // dummy2 = f*w
aPCSet.Subtract(dummy1,dummy2,dwdt);     // dwdt = e*z - f*w
```

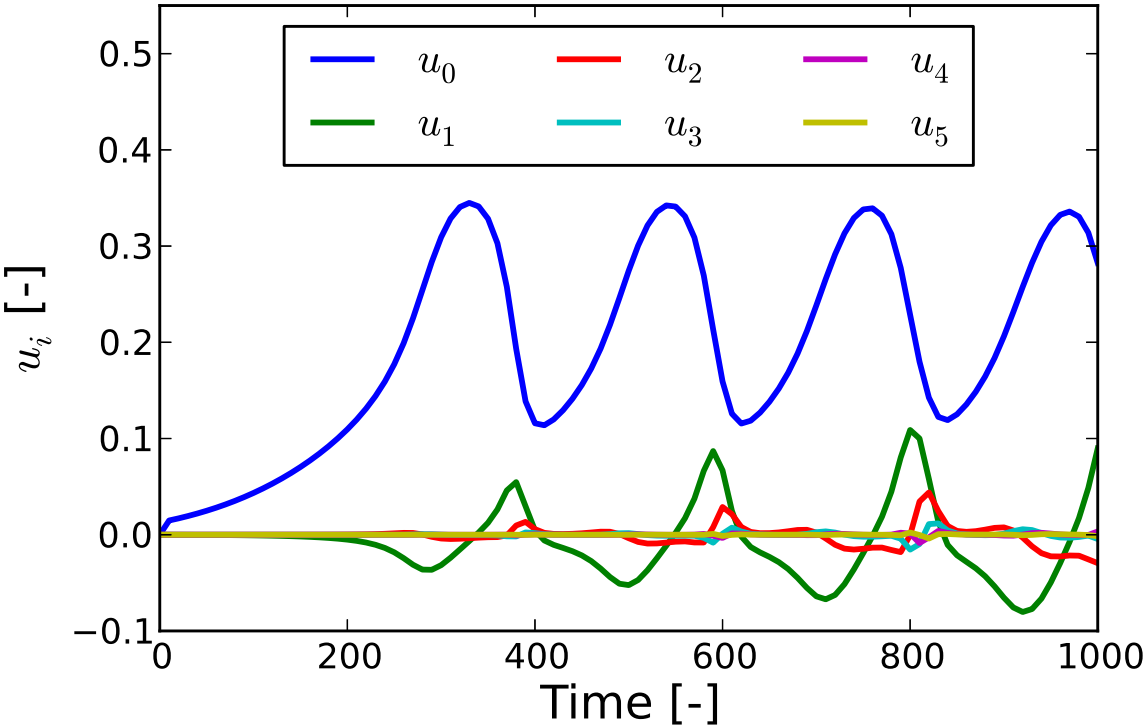
- Dummy variables used where needed to build the terms in the equations
- Data structure is currently being enhanced to provide the operation result as the function return value
  - Will allow more elegant inline replacement of operators with their stochastic counterparts

Surface Reaction Model: ISP results



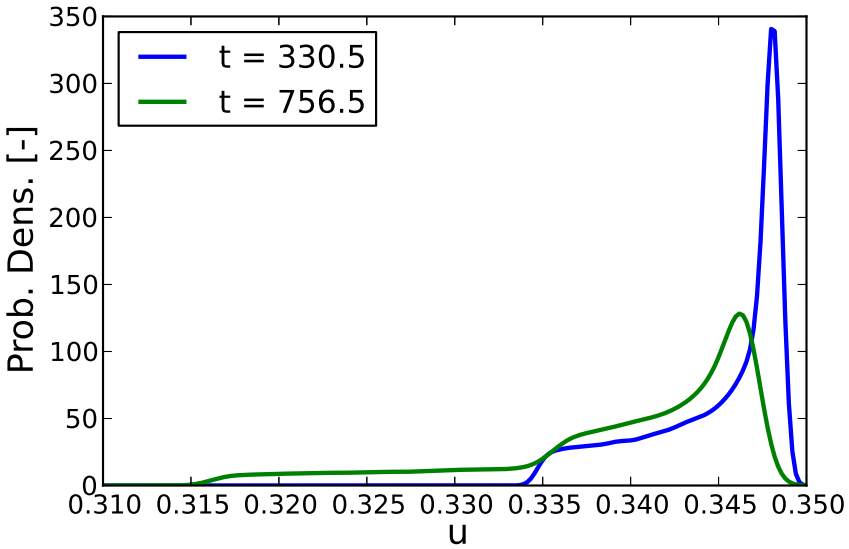
- Assume 0.5% uncertainty in *b* around nominal value
- Legendre-Uniform intrusive PC
- Mean and standard deviation for *u*, *v*, and *w*
- Uncertainty grows in time

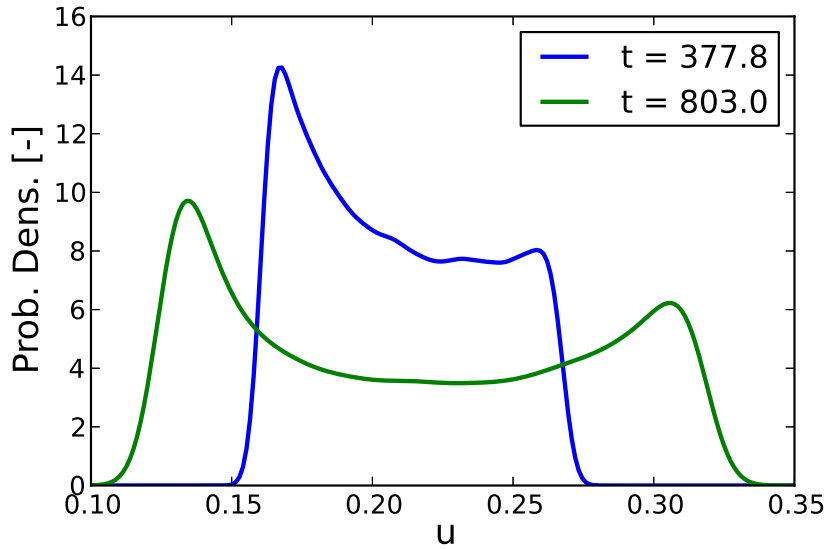
Surface Reaction Model: ISP results



- Modes of *u*
- Modes decay with higher order
- Amplitudes of oscillations of higher order modes grow in time

Surface Reaction Model: ISP results: PDFs





- Pdfs of u at maximum mean (left) and maximum standard deviation (right)
- Distributions get broader and multimodal as time increases
  - Effect of accumulating uncertainty in phase of oscillation

Spectral UQ: Incompressible Flow - Stochastic Projection Method

- $(P+1)$ Galerkin-Projected Mom./Cont. Eqns, $q = 0, \ldots, P$:

$$\begin{aligned} \frac{\partial \boldsymbol{v}_q}{\partial t} + \nabla \cdot \langle \boldsymbol{v}\boldsymbol{v} \rangle_q &= -\nabla p_q + \frac{1}{\mathrm{Re}} \nabla \cdot \left\langle \mu[(\nabla \boldsymbol{v}) + (\nabla \boldsymbol{v})^T] \right\rangle_q \\ \nabla \cdot \boldsymbol{v}_q &= 0 \end{aligned}$$

- Projection: for $q = 0, \ldots, P$:

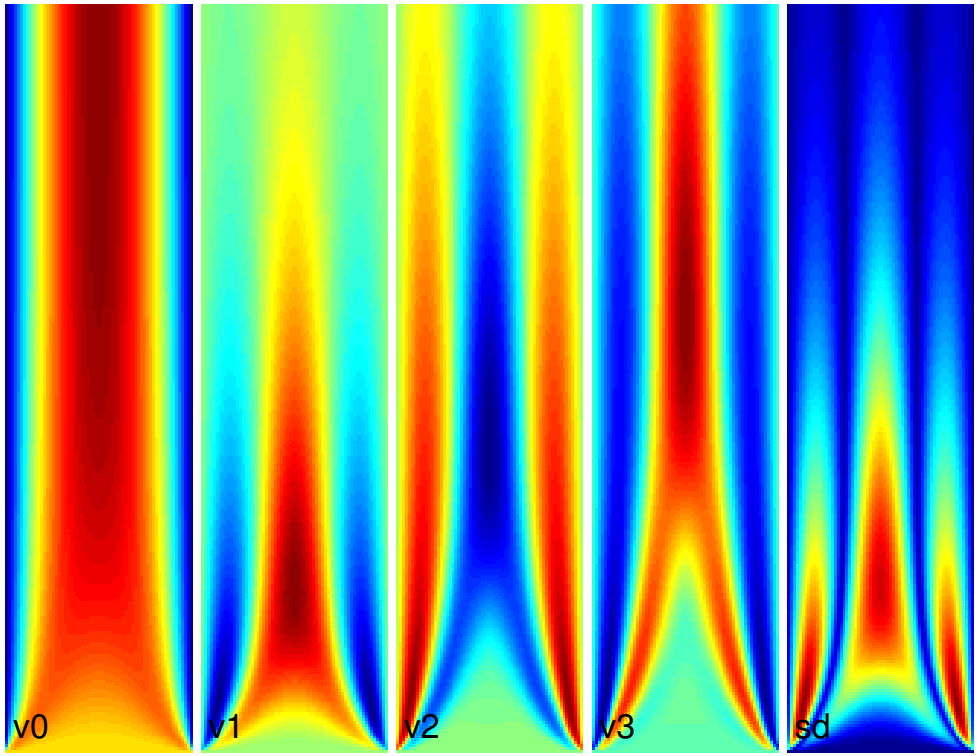
$$\begin{aligned} \frac{\tilde{\boldsymbol{v}}_q - \boldsymbol{v}_q^n}{\Delta t} &= C_q^n + D_q^n \\ \nabla^2 p_q &= -\frac{1}{\Delta t} \nabla \cdot \tilde{\boldsymbol{v}}_q \\ \frac{\boldsymbol{v}_q^{n+1} - \tilde{\boldsymbol{v}}_q}{\Delta t} &= -\nabla p_q \end{aligned}$$

- $P+1$ *decoupled* Poisson Eqns for the pressure modes

[Le Maître *et al.*, *J. Comp. Phys.*, 2001.]

Laminar 2D Channel Flow with Uncertain Viscosity

- Incompressible flow
- Gaussian viscosity PDF
  - $\nu = \nu_0 + \nu_1 \xi$
- Streamwise velocity
  - $v = \sum_{i=0}^{P} v_i \Psi_i$
  - v_0: mean
  - v_i: i-th order mode
  - $\sigma^2 = \sum_{i=1}^{P} v_i^2 \left\langle \Psi_i^2 \right\rangle$



v_0 v_1 v_2 v_3 σ

Non-Intrusive propagation of uncertainty - Projection

Galerkin Projection

$$u_k = \frac{\langle u\Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \frac{1}{\langle \Psi_k^2 \rangle} \int u\Psi_k(\xi) w(\xi) d\xi, \quad k = 0, \ldots, P$$

Evaluate projection integrals numerically

- Pick samples of uncertain parameters, e.g. b
- Run deterministic forward model for each of the sampled input parameter values b_i
- Integration depends on sampling approach
  - Random Sampling: $\langle u\Psi_k \rangle = \frac{1}{N_s} \sum_{i=1}^{N_s} u(b_i)$
  - Quadrature: $\langle u\Psi_k \rangle = \sum_{i=1}^{N_q} q_i u(b_i)$

Reconstruct uncertain model output

$$u(x, t; \theta) = \sum_{k=0}^{P} u_k(x, t)\Psi_k(\xi(\theta))$$

Random Sampling approaches

- Evaluate integral through sampling

$$\int u\Psi_k(\xi)w(\xi)d\xi = \frac{1}{N_s}\sum_{i=1}^{N_s} u(\xi_i)$$

- Samples are drawn according to the distribution of ξ
  - Monte-Carlo (MC)
  - Latin-Hypercube-Sampling (LHS)
- Pros:
  - Can be easily made fault tolerant
  - Sometimes random samples is all we have
- Cons: slow convergence, but less dependent on number of stochastic dimensions

Quadrature approaches

- Numerically evaluate integrals in Galerkin projection

$$\int u\Psi_k(\xi)w(\xi)d\xi = \sum_{i=1}^{N_q} q_i u(\xi_i)$$

- Gauss quadrature rules are very efficient
  - ξ are quadrature points, with corresponding weights q_i
  - N_q quadrature points can integrate polynomial of order $2N_q - 1$ exactly
  - Gauss-Hermite and Gauss-Legendre quadrature tailored to specific choices of the weight function $w(\xi)$
  - As a rule of thumb, $p + 1$ quadrature points are needed for Galerkin projection of PCE of order p
    - If both u and Ψ_k are of order p, then integrand is of order $2p$
    - $2p \leq 2N_q - 1$ or $N_q \geq p + \frac{1}{2}$
    - Only exact if u is indeed a polynomial of order $\leq p$
- Pros:
  - Can use existing codes as black box to evaluate $u(\xi_i)$
  - Embarrassingly parallel
- Cons: Tensor product rule for d dimensions requires N_q^d samples

Non-Intrusive propagation of uncertainty - Collocation

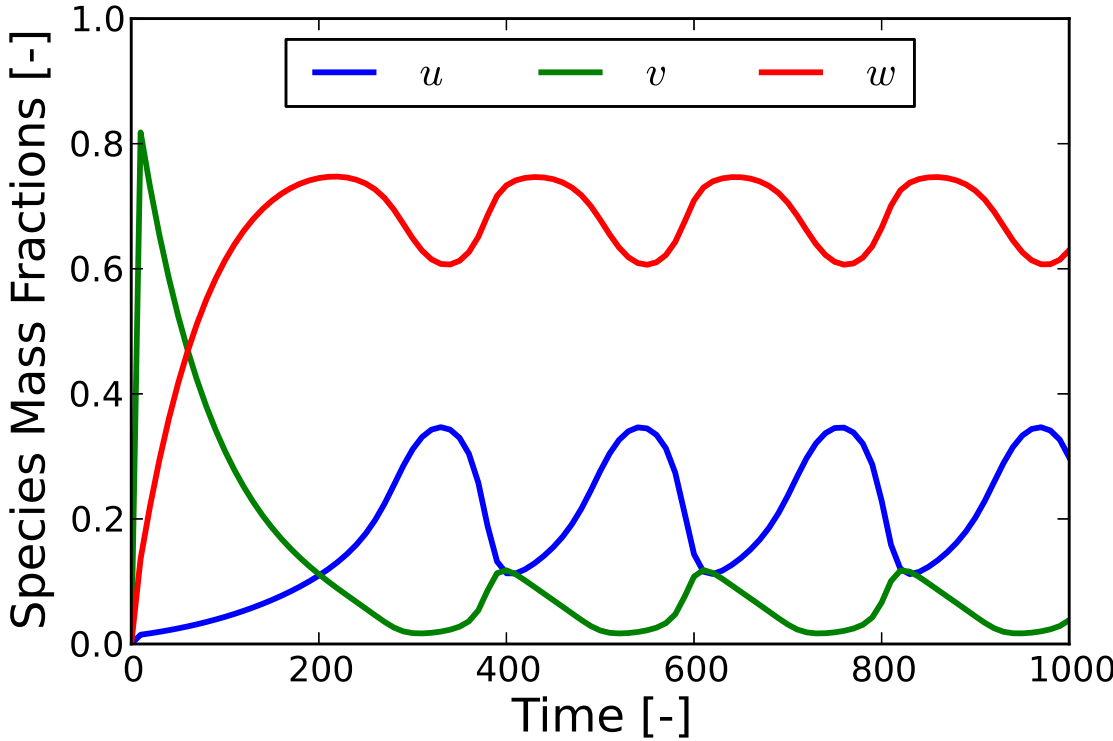
- Collocation techniques minimize errors at sample points
  - $\sum_{k=0}^{P} u_k \Psi_k(\xi_i) = u(\xi_i) \ , \quad i = 1, \ldots, N_c$
  - Can use interpolation, e.g. Lagrange interpolants
  - Or use regression approaches: $P + 1$ degrees of freedom to fit N_c points
- Pros: can position points where most accuracy desired
- Cons:

Surface Reaction Model

3 ODEs for a monomer (u), dimer (v), and inert species (w) adsorbing onto a surface out of gas phase.

$$\frac{du}{dt} = az - cu - 4duv$$
$$\frac{dv}{dt} = 2bz^2 - 4duv$$
$$\frac{dw}{dt} = ez - fw$$
$$z = 1 - u - v - w$$

$$u(0) = v(0) = w(0) = 0.0$$



Oscillatory behavior for $b \in [20.2, 21.2]$

$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad d = 1.0 \quad e = 0.36 \quad f = 0.016$

(Vigil *et al.*, Phys. Rev. E., 1996; Makeev *et al.*, J. Chem. Phys., 2002)

Surface Reaction Model: NISP implementation in UQTk

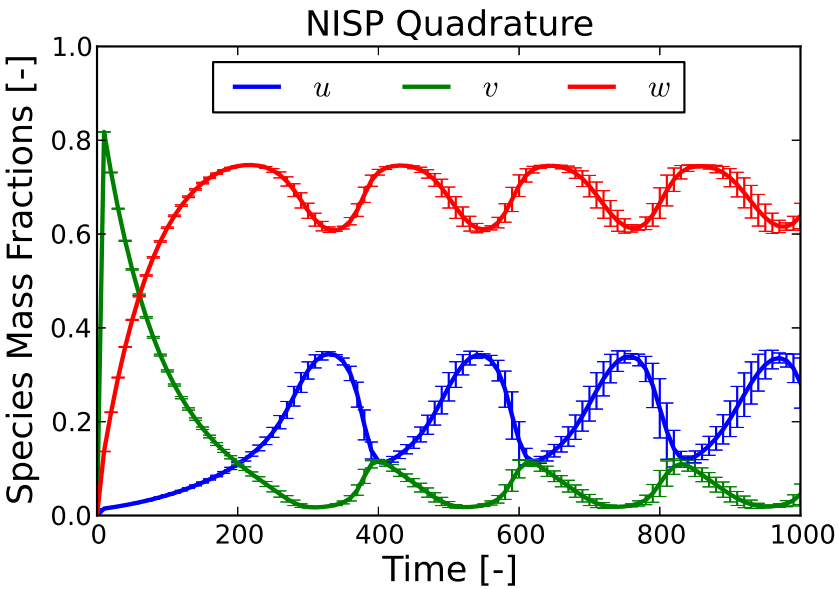
Quadrature:

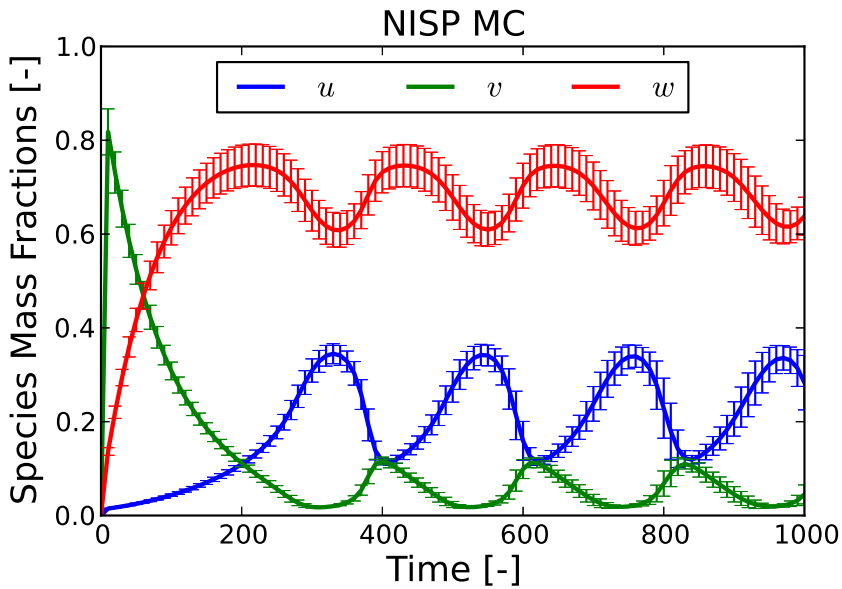
```
// Get the quadrature points
int nQdpts=myPCSet.GetNQuadPoints();
double* qdpts=new double[nQdpts];
myPCSet.GetQuadPoints(qdpts);
...
// Evaluate parameter at quad pts
for(int i=0;i<nQdpts;i++){
  bval[i]=myPCSet.EvalPC(b,&qdpts[i]);
}
...
// Run model for all samples
for(int i=0;i<nQdpts;i++){
  u_val[i] = ...
}
// Spectral projection
myPCSet.GalerkProjection(u_val,u);
myPCSet.GalerkProjection(v_val,v);
myPCSet.GalerkProjection(w_val,w);
```

Monte-Carlo Sampling:

```
// Get the sample points
int nSamples=1000;
Array2D<double> samPts(nSamples,dim);
myPCSet.DrawSampleVar(samPts);
...
// Evaluate parameter at sample pts
for(int i=0;i<nSamples;i++){
  ... // select samPt from samPts
  bval[i]=myPCSet.EvalPC(b,&samPt)
}
...
// Run model for all samples
for(int i=0;i<nSamples;i++){
  u_val[i] = ...
}
// Spectral projection
myPCSet.GalerkProjectionMC(samPts,u_val,u);
myPCSet.GalerkProjectionMC(samPts,v_val,v);
myPCSet.GalerkProjectionMC(samPts,w_val,w);
```

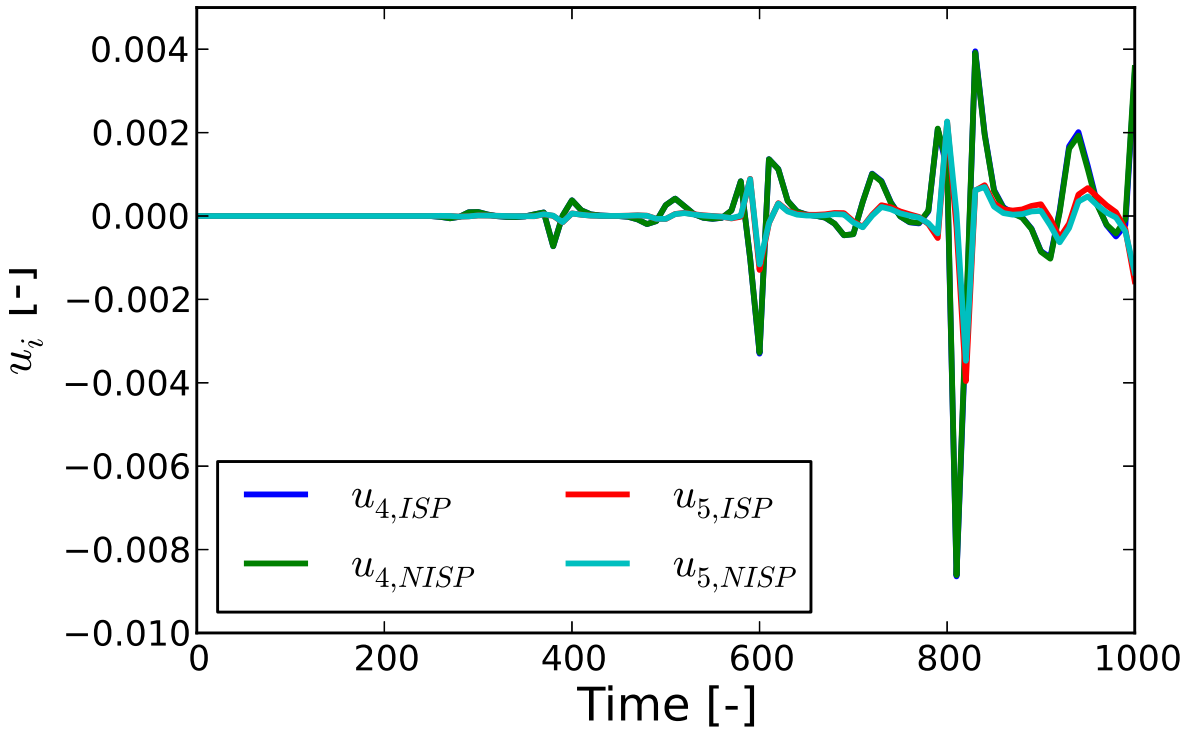
Surface Reaction Model: NISP results





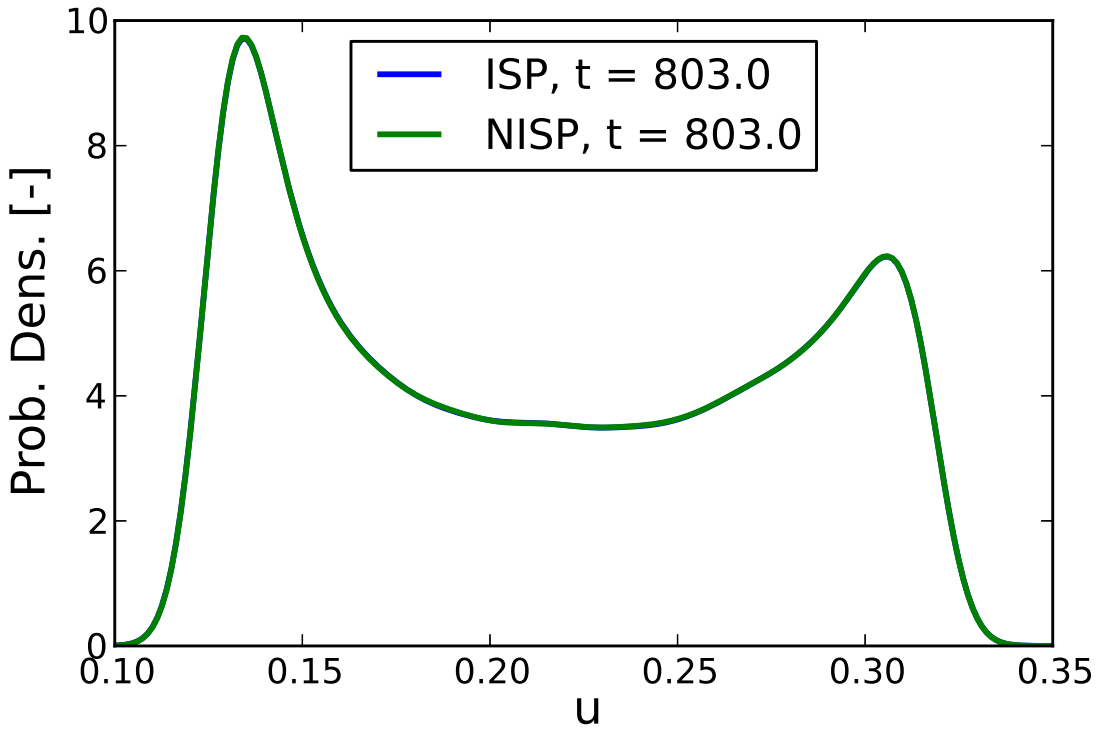
- Mean and standard deviation for u, v, and w
- Quadrature approach agrees well with ISP approach using 6 quadrature points
- Monte Carlo sampling approach converges slowly
  - With a 1000 samples, results are quite different from ISP and NISP

Surface Reaction Model: Comparison ISP and NISP



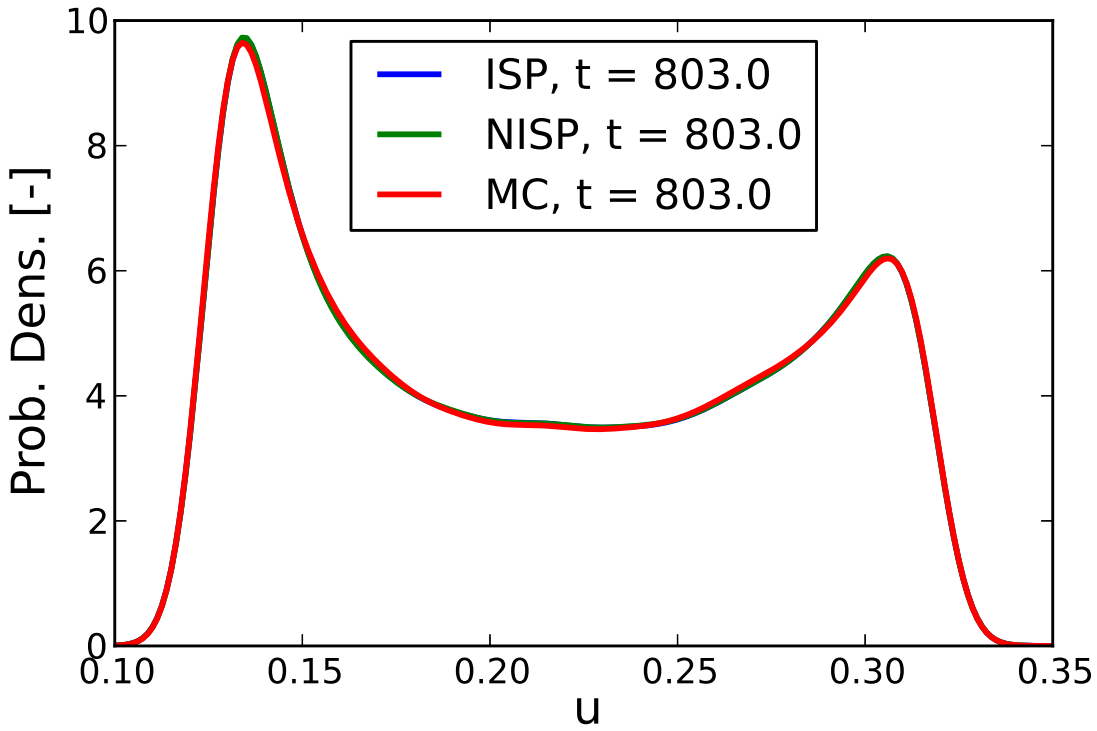
- Lower order modes agree perfectly
- Very small differences in higher order modes
  - Difference increases with time

Surface Reaction Model: Comparison ISP and NISP



- All pdf's based on 50K samples each and evaluated with Kernel Density Estimation (KDE)
- No difference in PDFs of sampled PCEs between NISP and ISP

Surface Reaction Model: Comparison ISP, NISP, and MC



- All pdf's based on 50K samples each and evaluated with Kernel Density Estimation (KDE)
- Good agreement between intrusive, non-intrusive projection, and Monte Carlo sampling

ISP pros and cons

- Pros:
  - Elegant
  - One time solution of system of equations for the PC coefficients fully characterizes uncertainty in all variables at all times
  - Tailored solvers can (potentially) take advantage of new hardware developments
- Cons:
  - Often requires re-write of the original code
  - Reformulated system is factor (P+1) larger than the original system and can be challenging to solve
  - Challenges with increasing time-horizon for ODEs
- Many efforts in the community to automate ISP
  - UQToolkit: `http://www.sandia.gov/UQToolkit/`
  - Sundance: `http://www.math.ttu.edu/~klong/Sundance/html/`
  - Stokhos: `http://trilinos.sandia.gov/packages/stokhos/`
  - ...

NISP pros and cons

- Pros:
  - Easy to use as wrappers around existing codes
  - Embarassingly parallel
- Cons:
  - Most methods suffer from curse of dimensionality $N_q = n^{N_d}$
- Many development efforts for smarter sampling approaches and dimensionality reduction
  - (Adaptive) Sparse Quadrature approaches
  - Compressive Sensing
  - ...
- Sampling methods have found very wide spread use in the community
  - DAKOTA: `http://dakota.sandia.gov/`
  - ...

Sensitivity Analysis

- Obtaining global sensitivity analysis from PCEs
  - Identify dominant sources of uncertainty
  - Attribution

PC postprocessing: global sensitivity information is readily obtained from PCE

$$g(x_1, \ldots, x_d) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

- Global sensitivity analysis $\equiv$ Variance decomposition
- Total variance

$$Var[g(\boldsymbol{x})] = \sum_{k>0} c_k^2 ||\Psi_k||^2$$

- Main effect sensitivity indices

$$S_i = \frac{Var[\mathbb{E}(g(\boldsymbol{x}|x_i)]}{Var[g(\boldsymbol{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 ||\Psi_k||^2}{\sum_{k>0} c_k^2 ||\Psi_k||^2}$$

$\mathbb{I}_i$ is the set of bases with only x_i involved. S_i is the uncertainty contribution that is due to i-th parameter only.

- Joint sensitivity indices

$$S_{ij} = \frac{Var[\mathbb{E}(g(\boldsymbol{x}|x_i, x_j)]}{Var[g(\boldsymbol{x})]} - S_i - S_j = \frac{\sum_{k \in \mathbb{I}_{ij}} c_k^2 ||\Psi_k||^2}{\sum_{k>0} c_k^2 ||\Psi_k||^2}$$

$\mathbb{I}_{ij}$ is the set of bases with only x_i and x_j involved. S_{ij} is the uncertainty contribution that is due to (i, j) parameter pair.

PC postprocessing: sampling-based approaches

$$g(x_1, \ldots, x_d) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

- In some cases, need to resort to Monte-Carlo estimation, e.g.
  - Piecewise-PC with irregular subdomains
  - Output transformations, e.g. build PC for $\log g(\boldsymbol{x})$, but inquire sensitivity with respect to $g(\boldsymbol{x})$
- A brute-force sampling of $Var[\mathbb{E}(g(\boldsymbol{x}|x_i)]$ is extremely inefficient.
- Tricks are available, given a single set of sampled input [Saltelli, 2002]. E.g., use

$$\mathbb{E}[g(\boldsymbol{x}|x_i)^2] = \mathbb{E}[g(\boldsymbol{x}|x_i)g(\boldsymbol{x}'|x_i)] = \frac{1}{N-1}\sum_{r=1}^{N} g(\boldsymbol{x}^{(r)})g(\tilde{\boldsymbol{x}}^{(r)}),$$

  where $\tilde{\boldsymbol{x}}$ is $\boldsymbol{x}'$ with i-th element replaced by x_i.
- Similar formulae available for joint sensitivity indices.

- Con: as all Monte-Carlo algorithms, converges slowly.
- Pro: sampling is cheap.

Challenges for PCE-based Uncertainty Quantification

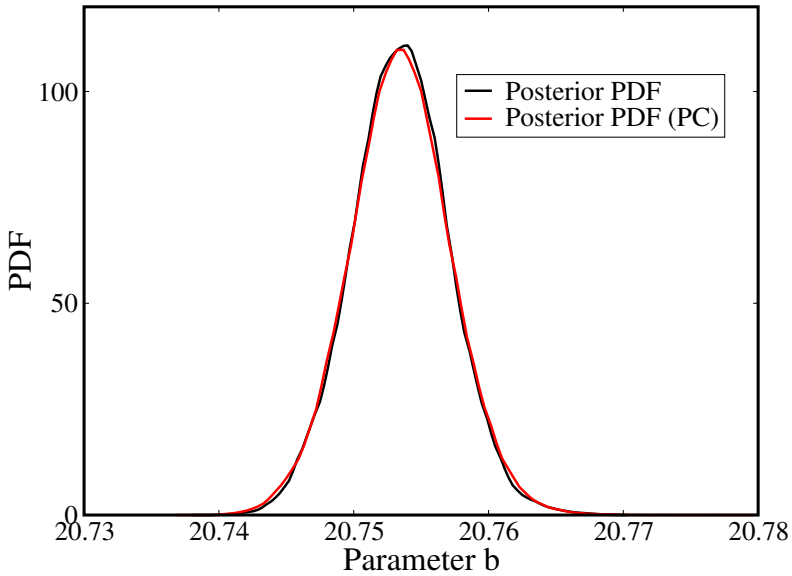
- Representing input variables with arbitrary distributions
- Systems with high-dimensional uncertainty
- Systems with long time horizon / oscillatory behavior
- Nonlinearities in governing equations for intrusive UQ
- Physical constraints in uncertain quantities
- Systems with non-smooth behavior – discontinuities
- Systems with inherent stochasticity

Various approaches have been developed to tackle these challenges ...

Obtaining PCEs for uncertain inputs

- Characterizing PCEs for uncertain inputs is a really difficult problem
- Inputs specified in a variety of ways
  - Probability density function
  - Samples
  - Expert opinion (e.g. “about 3.5”)
- Often obtained from inverse problem solution
  - See session on Bayesian inference
  - Generally provides many samples of the uncertain inputs

Inverse CDF Mapping for 1D RVs



- Consider random variable a with CDF $F(\cdot)$
  - Either specified or constructed from samples with KDE
- CDF transformation $F(a) = \eta$ maps random variable a to uniform$[0, 1]$ random variable η.
- $\eta = \Phi(\xi)$ maps uniform η to normal RV ξ
- The inverse CDF enables NISP projection

$$a = \sum_{k=0}^{P} a_k \Psi_k(\xi) \qquad a_k \propto \langle a \Psi_k(\xi) \rangle = \int \underbrace{F^{-1}(\Phi(\xi))}_{a} \Psi_k(\xi) d\xi$$

Constructing an *n*D PCE for a RV with a given PDF

- Given RV $z \in \mathbb{R}$ with PDF: $g(z)$, define:

$$z = \sum_{i=0}^{P} z_i \Psi_i(\xi_1, \xi_2, \ldots, \xi_n), \qquad P + 1 = \frac{(n+p)!}{n!p!}$$

- No general procedure
- Can choose $\{n, p\}$ and the mode strengths by ensuring accurate capture of
  - the PDF $g(z)$
  - select moments of z
  - some observable of interest $\phi(z)$

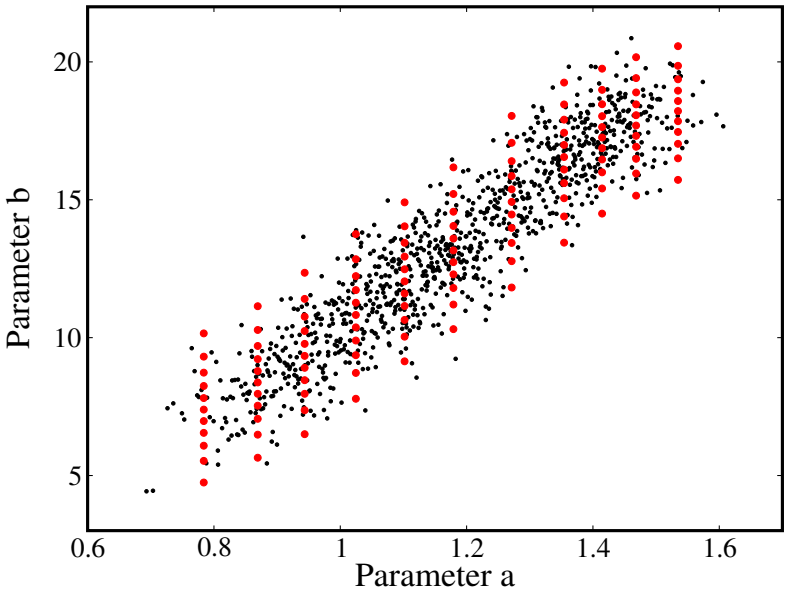
Rosenblatt for Multi-D RVs

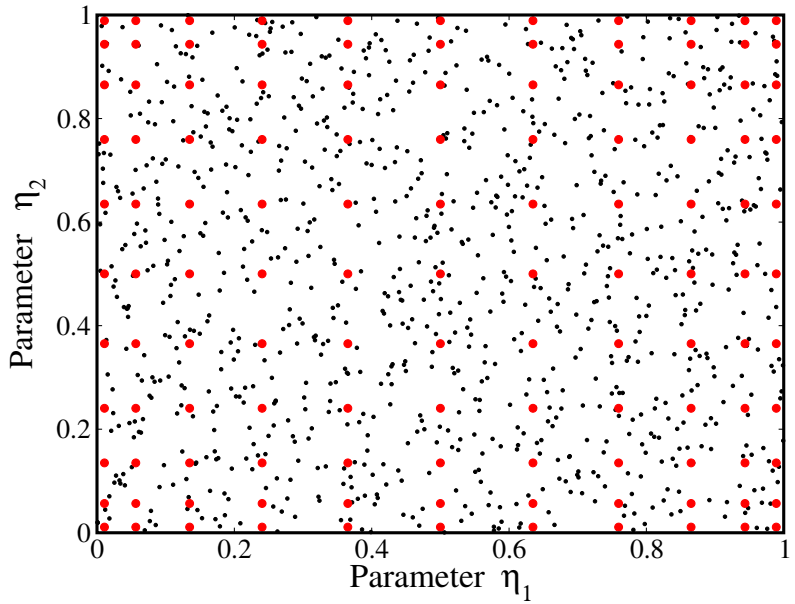
- Rosenblatt transformation maps any (not necessarily independent) set of random variables $(\lambda_1, \ldots, \lambda_n)$ to uniform i.i.d.'s $\{\eta_i\}_{i=1}^n$ (Rosenblatt, 1952).

$$\begin{aligned} \eta_1 &= F_1(\lambda_1) \\ \eta_2 &= F_{2|1}(\lambda_2|\lambda_1) \\ &\vdots \\ \eta_n &= F_{n|n-1,\ldots,1}(\lambda_n|\lambda_{n-1}, \ldots, \lambda_1) \end{aligned}$$

- Rosenblatt transformation is a multi-D generalization of 1D CDF mapping.
- Conditional CDFs are harder to evaluate in high dimensions

Projection of Rosenblatt transformed vars onto PCEs





- NISP projection is enabled by inverse Rosenblatt transformation $(a, b) = R^{-1}(\xi_1, \xi_2)$ ensures a well-defined quadrature integration

$$a = \sum_{k=0}^{P} a_k \Psi_k(\xi) \qquad a_k \propto \int \underbrace{R_a^{-1}(\xi)}_{a} \Psi_k(\xi) d\xi$$

$$b = \sum_{k=0}^{P} b_k \Psi_k(\xi) \qquad b_k \propto \int \underbrace{R_b^{-1}(\xi)}_{b} \Psi_k(\xi) d\xi$$

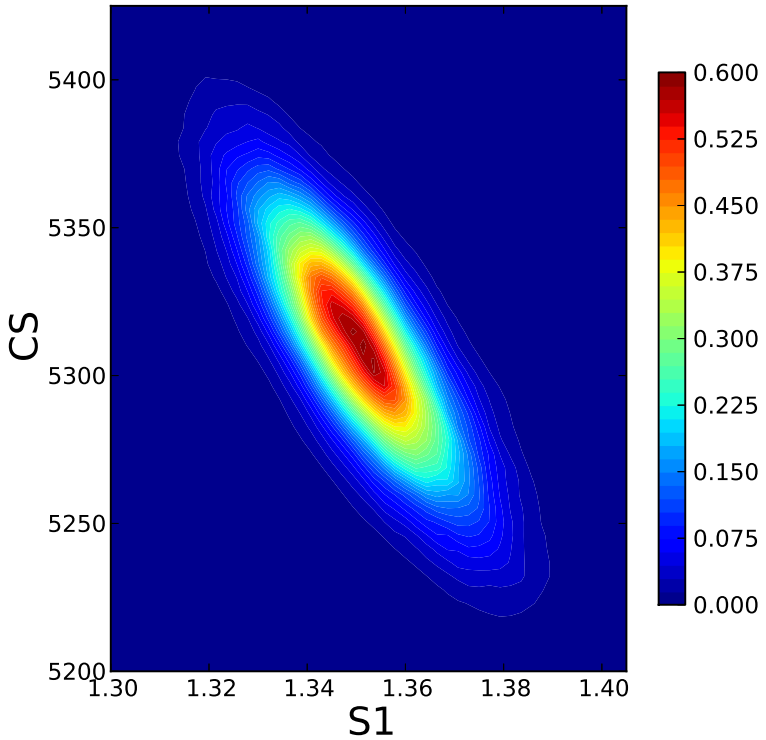
Multivariate Normal Approximation

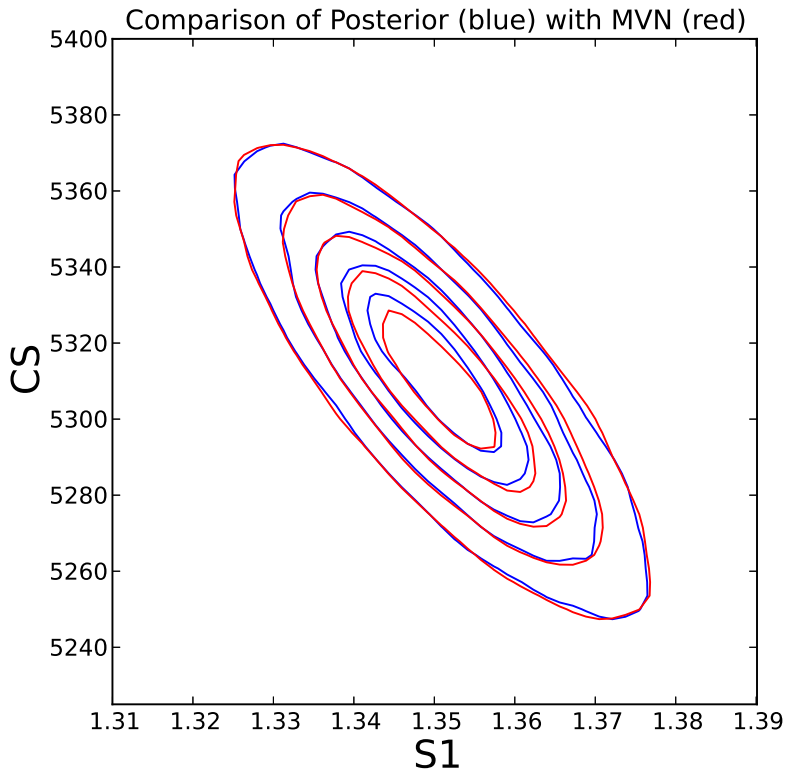
- Many distributions are unimodal and somewhat shaped like Gaussians
- MultiVariate Normal (MVN) approximations capture the mean and correlation structure of the random variables
- Easy to extract from a set of samples
  - In 1D: just compute mean and standard deviation: $u = u_0 + u_1 \xi$
  - Multi-D: Cholesky factorization of covariance

$$\begin{aligned} C &= LL^T \\ u &= L\xi \end{aligned}$$

```
# Compute mean parameter values
par_mean = numpy.mean(samples,axis=0)
# Compute the covariance
par_cov = numpy.cov(samples,rowvar=0)
# Compute the Cholesky Decomposition
chol_lower = numpy.linalg.cholesky(par_cov)
```

MVN approximation of Bayesian posterior from MCMC samples





$$S_1 = 1.351 + 0.01367\xi_1$$
$$CS = 5310 - 26.25\xi_1 + 20.26\xi_2$$

Data Decomposition Approaches to Handle Multi-Modalities

- Multimodal, dependent random variables
- Clustering and Rosenblatt transformation to represent data sets with PCEs
- PC mixture model

Karhunen-Loève (KL) Expansions

- Assume stochastic process $F(x,\theta) : D \times \Theta \to \mathbb{R}$ an L^2 random field on D
- With covariance function $Cov(x,y)$
- F can be written as

$$F(x,\theta) = \langle F(x,\theta) \rangle_\theta + \sum_{k=1}^{\infty} \sqrt{\lambda_k} f_k(x) \xi_k$$

- $f_k(x)$: eigenfunctions of $Cov(x,y)$
- λ_k: corresponding eigenvalues, all positive
- ξ_k: uncorrelated random variables, unit variance
  - Samples are obtained by projecting realizations of F onto f_k
  - Generally not independent
    - Special case: for Gaussian F, ξ_k are i.i.d. normal random variables
- The KLE is *optimal*: of all possible orthonormal bases for $L^2(\Theta \times D)$ the above $\{f(x)\}$ minimize the mean-square error in a finite linear representation of $F(\cdot)$.

KL Expansions - Numerical Approach - 1

- Covariance Matrix, $Cov(x, y) = \langle F(x, \theta)F(y, \theta)\rangle_\theta$:
  - specified analytically
  - estimated from samples
- Estimate eigenvalyes and eigenvectors for the Fredholm equation of second kind:

$$\int Cov(x, y)f(y)dy = \lambda f(x)$$

  - ...using the Nystrom algorithm:

$$\sum_{i=1}^{N_p} w_i Cov(x, y_i)f(y_i) = \lambda f(x)$$

    where w_i are the weights for the quadrature rule that uses N_p points y_i where realizations are provided.
  - Further manipulation leads to the eigenvalue problem

$$Ag = \lambda g$$

    where $A = WKW$ and $g = Wf$, with W being the diagonal matrix $W_{ii} = \sqrt{w_i}$ and $K_{ij} = Cov(x_i, y_j)$. Solutions consist of pairs of eigenvalues λ_k and eigenmodes $f_k = W^{-1}g_k$.

KL Expansions - Numerical Approach - 2

- Samples of random variables ξ_k are obtained by projecting realizations of the random process F on the eigenmodes f_k

$$\xi_k|_{\theta_l} = \left\langle F(x,\theta_l) - \langle F(x,\theta)\rangle_\theta \, , f_k(x)\right\rangle_x / \sqrt{\lambda_k}$$

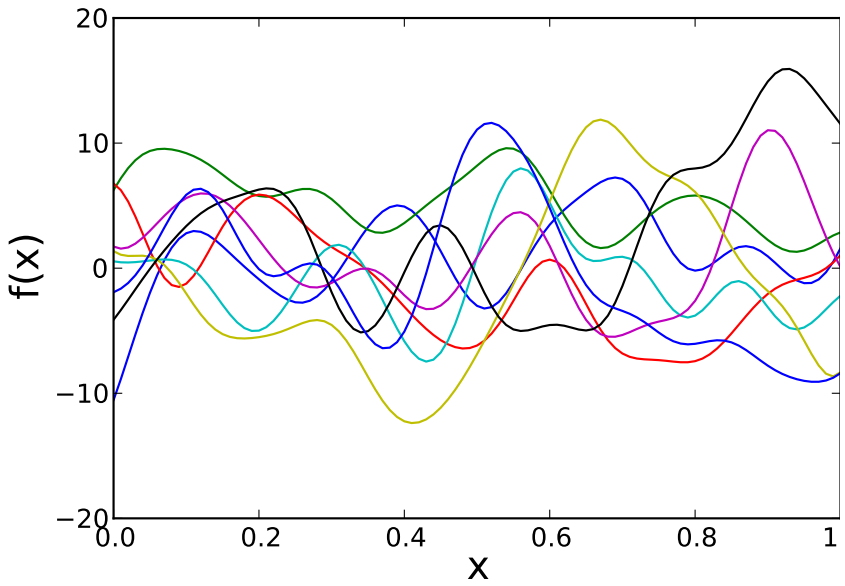
  - ... or numerically

$$\xi_k|_{\theta_l} = \sum_{i=1}^{N_p} w_i \left(F(x_i,\theta_l) - \langle F(x_i,\theta)\rangle_\theta\right) f_k(x_i)/\sqrt{\lambda_k}$$

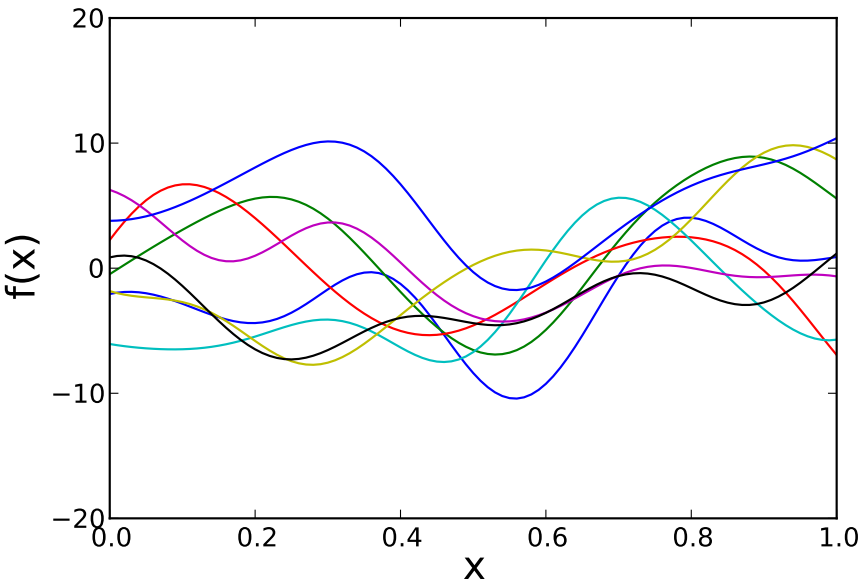
- If Gaussian process: automatically have first order WH PCE
- If not, same approaches as for converting RVs to PCEs applied to KL RVs

1D Gaussian Process: Realizations

$\delta = 0.1$



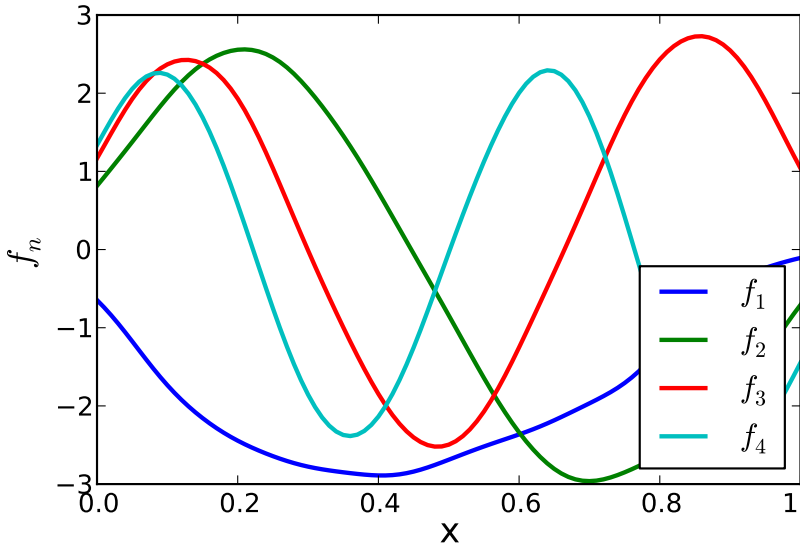
$\delta = 0.2$



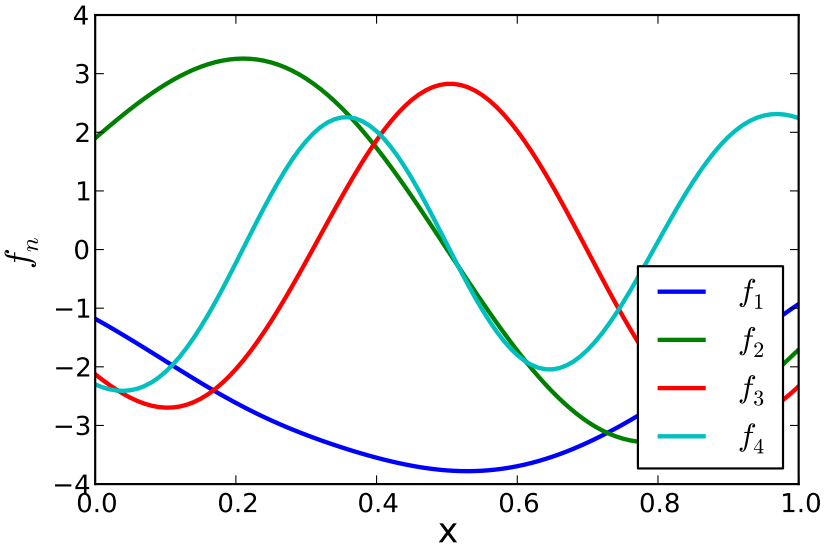
- Covariance $Cov(x_1, x_2) = exp(-(x_1 - x_2)^2/\delta^2)$
- Sample realizations are noisier as correlation length decreases

1D Gaussian Process: KL modes

$\delta = 0.1$



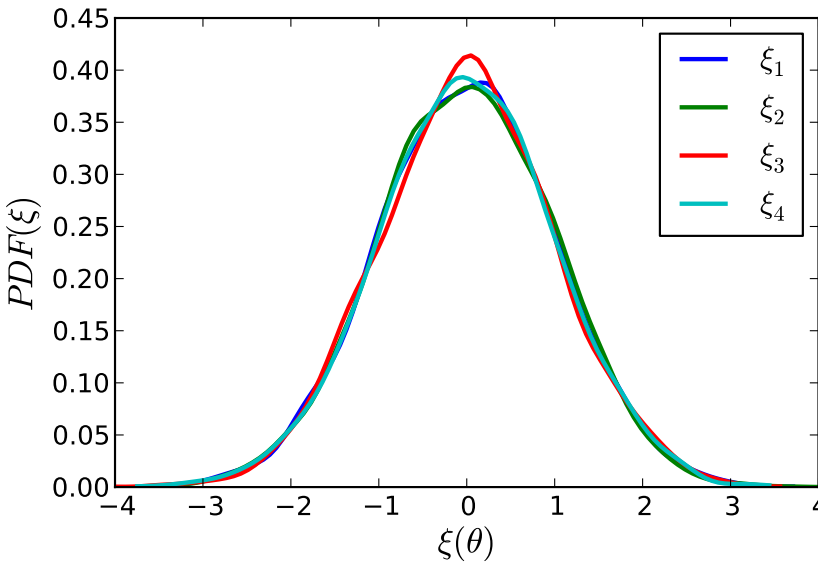
$\delta = 0.2$



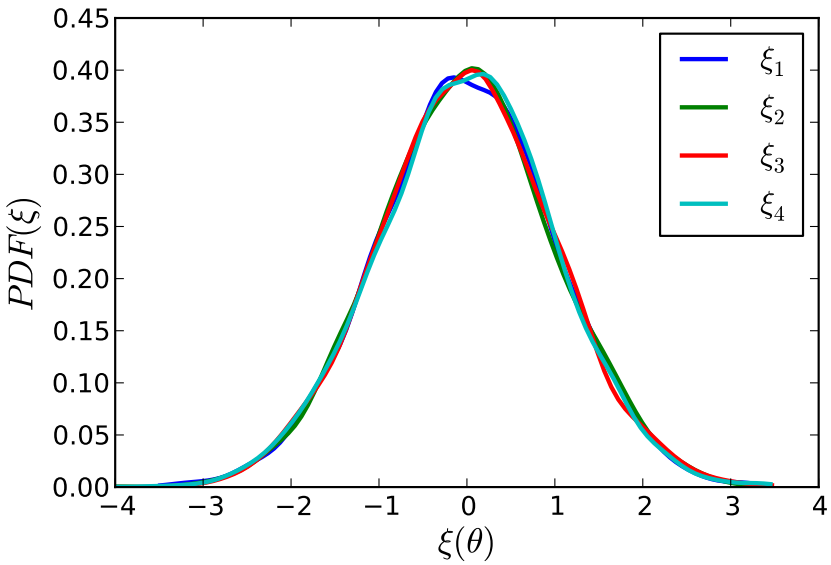
- Eigenmodes of the covariance matrix
  - Data covariance matrix constructed from 4096 Gaussian process realizations
- Higher modes are more oscillatory

1D Gaussian Process: KL random variables

$\delta = 0.1$

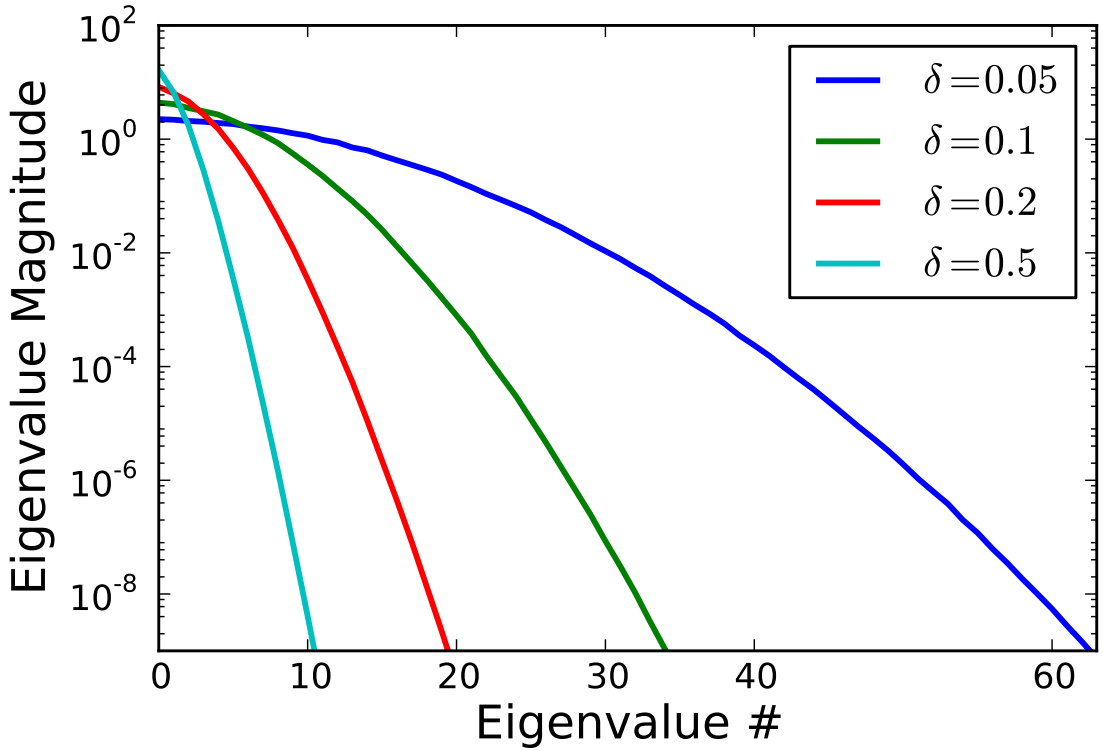


$\delta = 0.2$



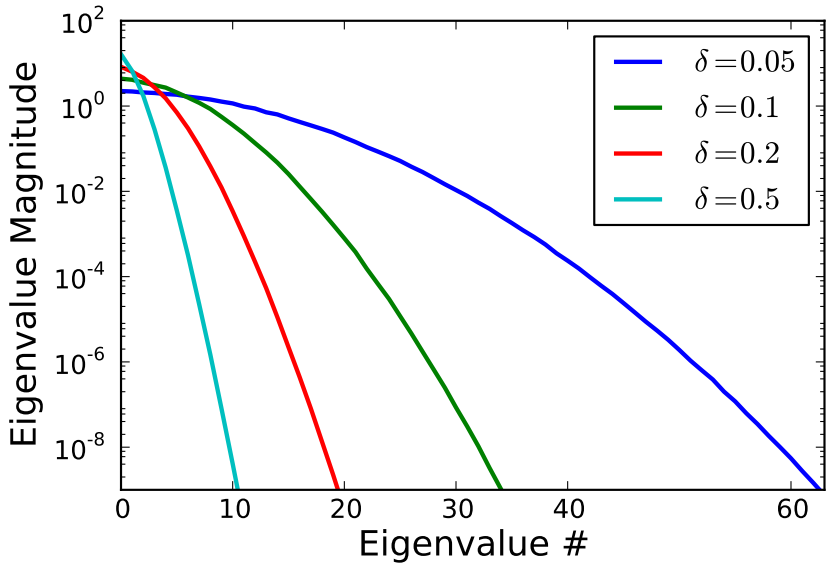
- Random variables obtained by projecting realizations onto KL modes
- Uncorrelated by construction
  - Also independent due to nature of Gaussian Process

1D Gaussian Process: Eigenvalue spectrum



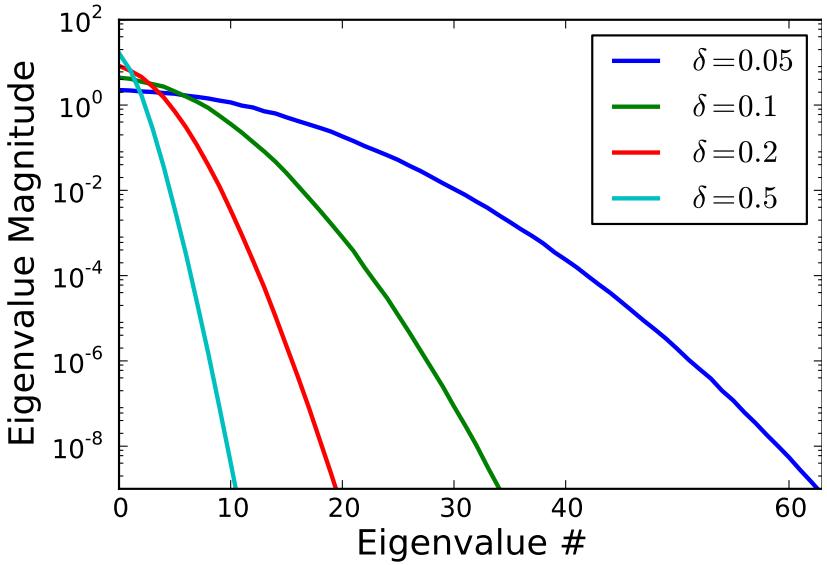
- Eigenvalue spectrum decays more slowly as correlation length decreases
  - More oscillatory modes needed to represent fluctuations in x
- KL expansion generally is truncated after enough modes are included to capture a specified fraction of the total variance

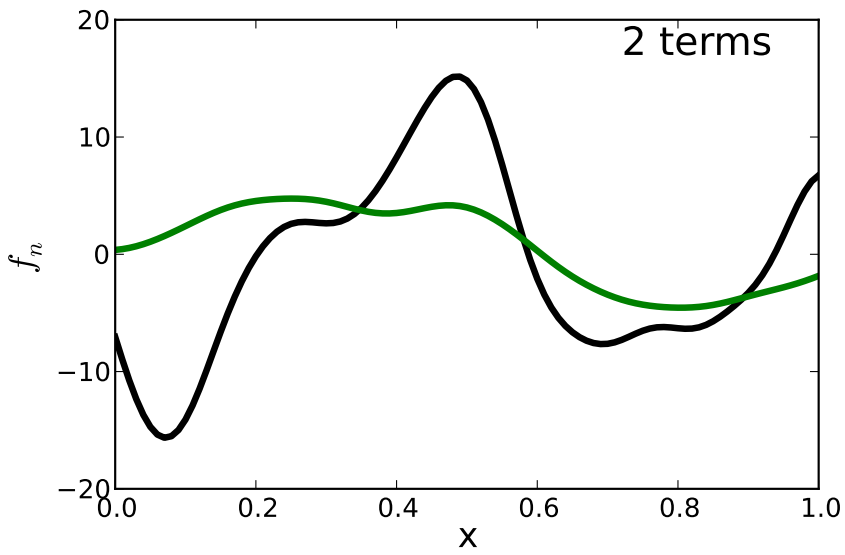
1D Gaussian Process: Reconstructed realizations



- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

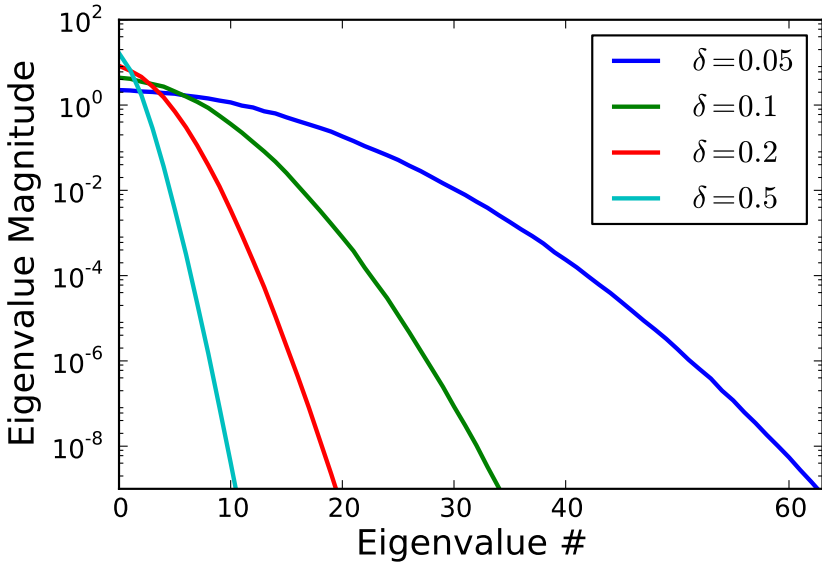
1D Gaussian Process: Reconstructed realizations

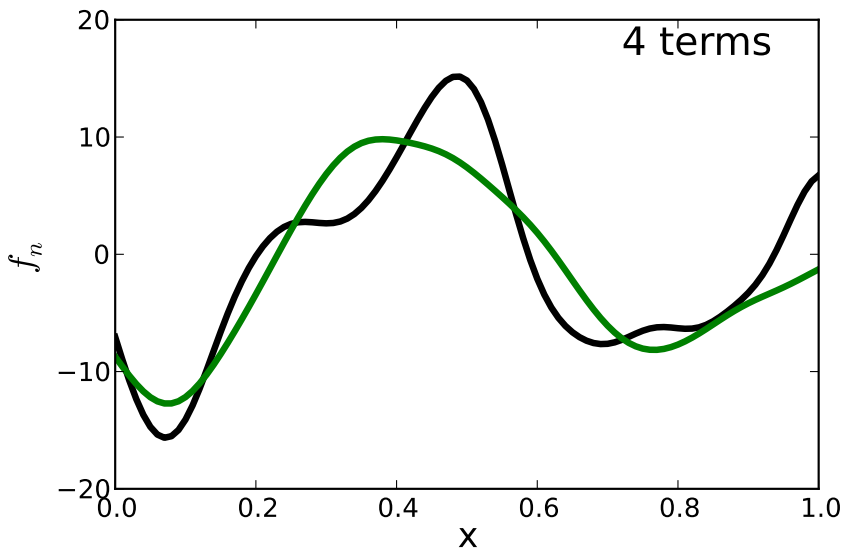




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

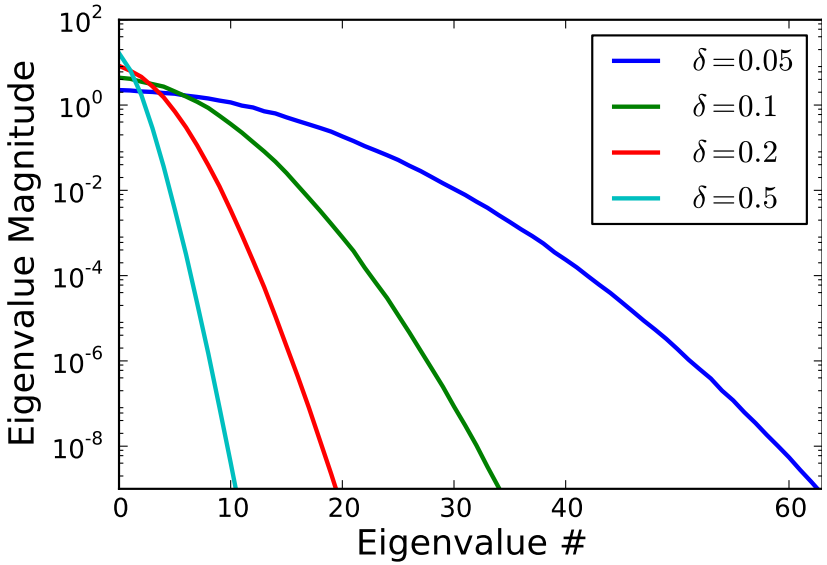
1D Gaussian Process: Reconstructed realizations

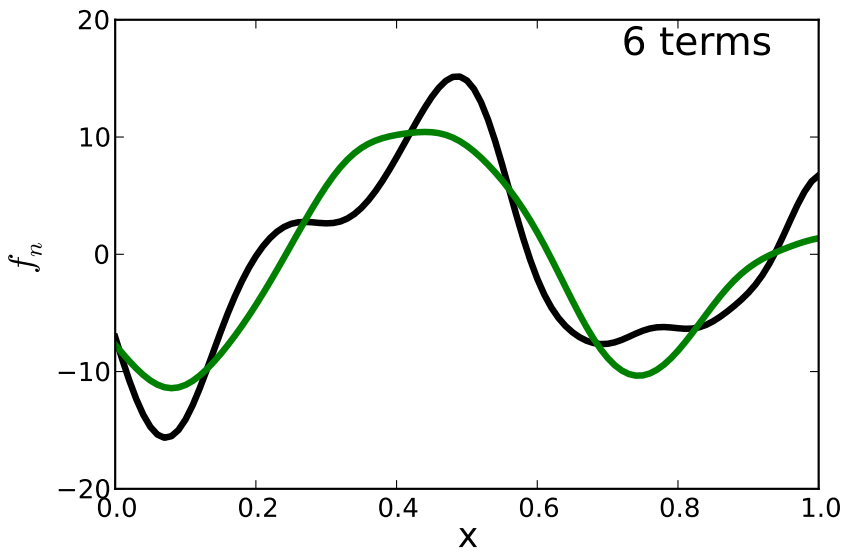




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

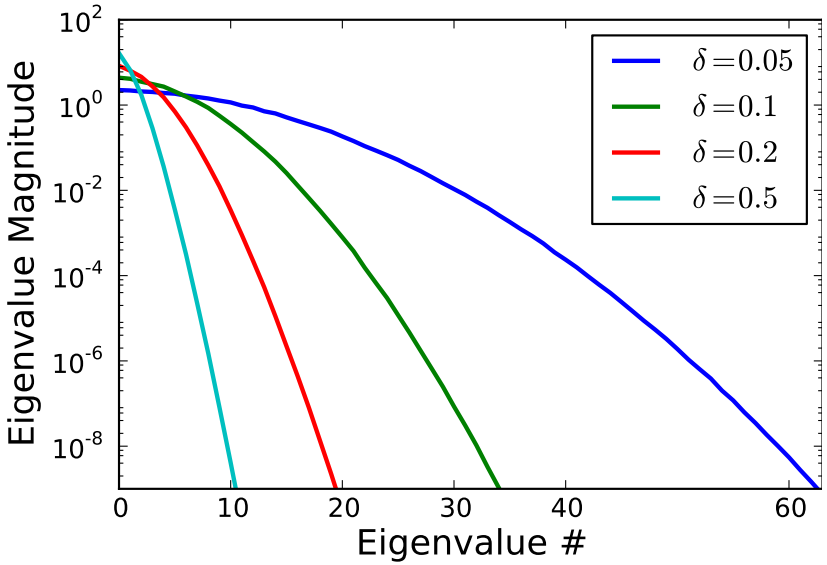
1D Gaussian Process: Reconstructed realizations

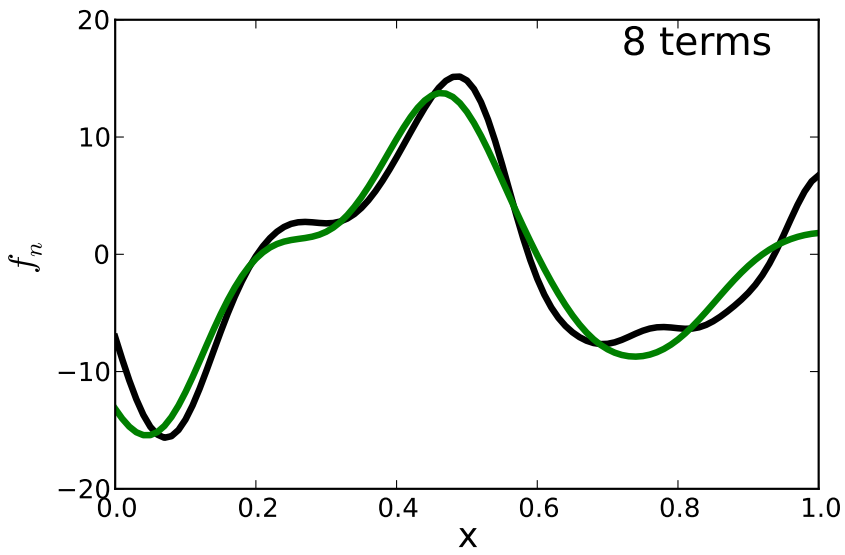




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

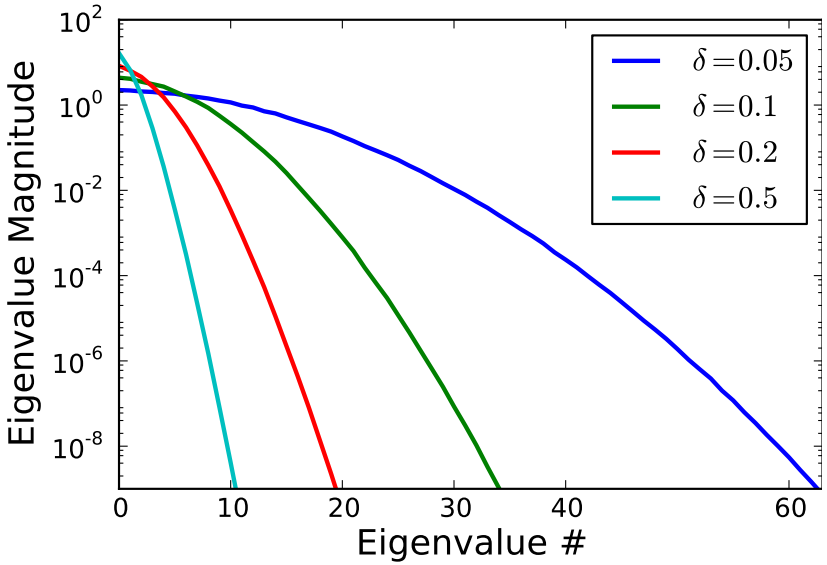
1D Gaussian Process: Reconstructed realizations

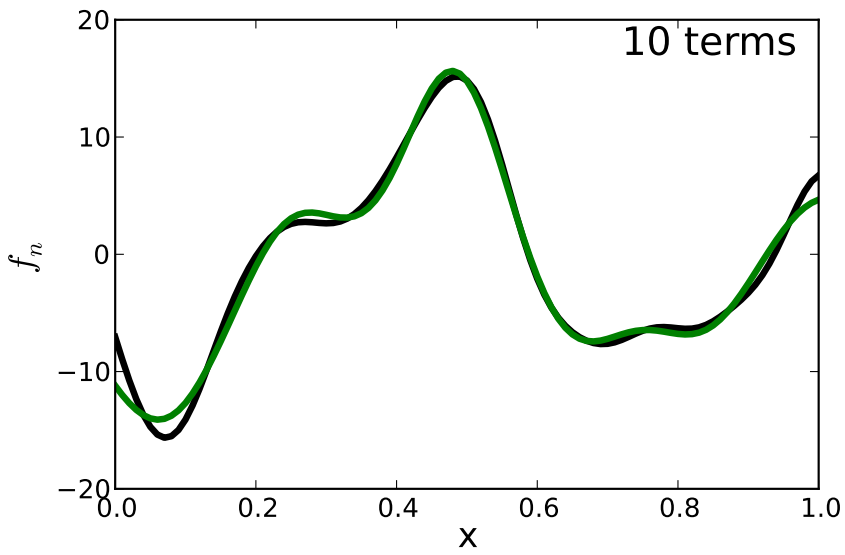




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

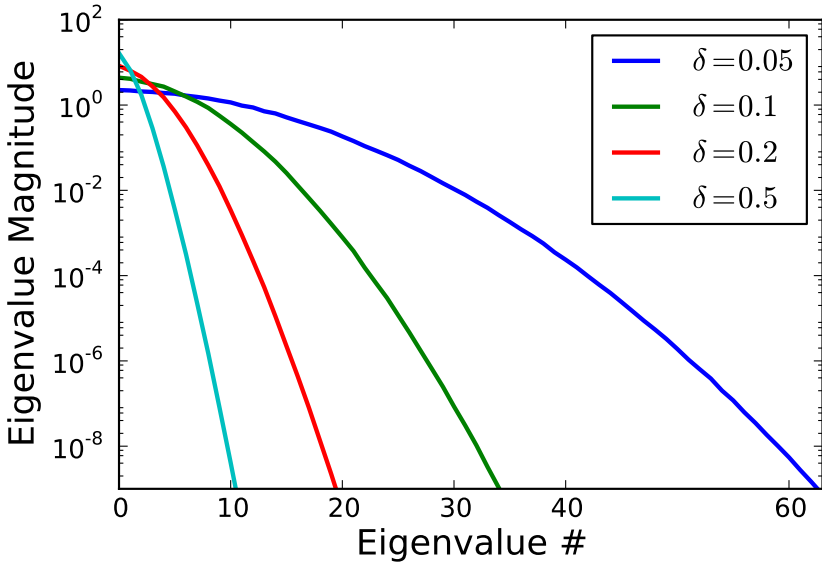
1D Gaussian Process: Reconstructed realizations

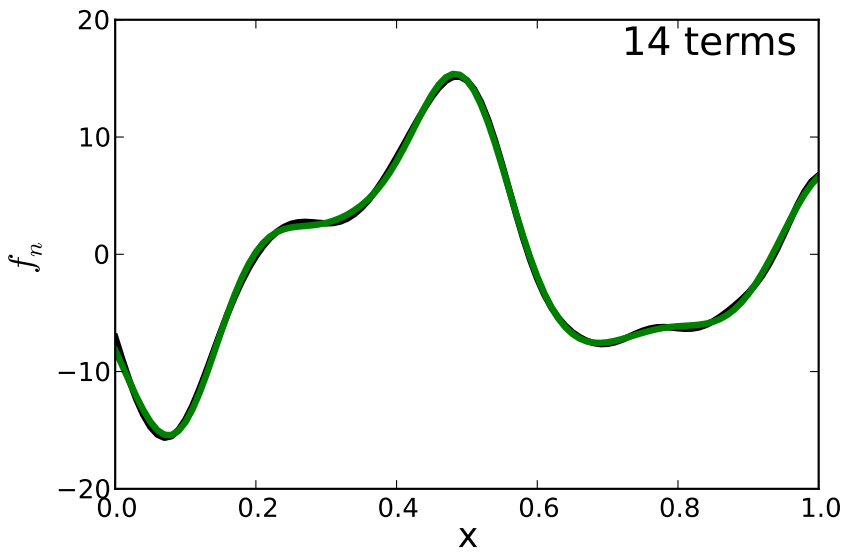




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

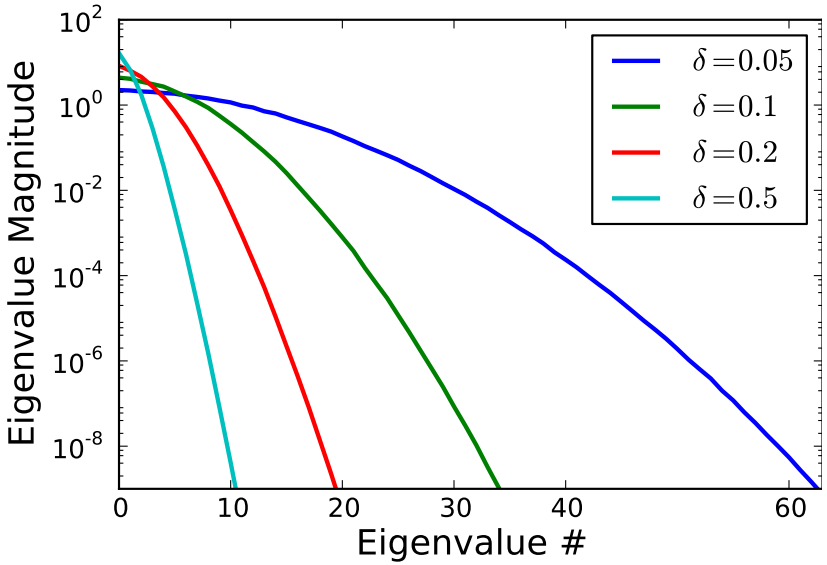
1D Gaussian Process: Reconstructed realizations

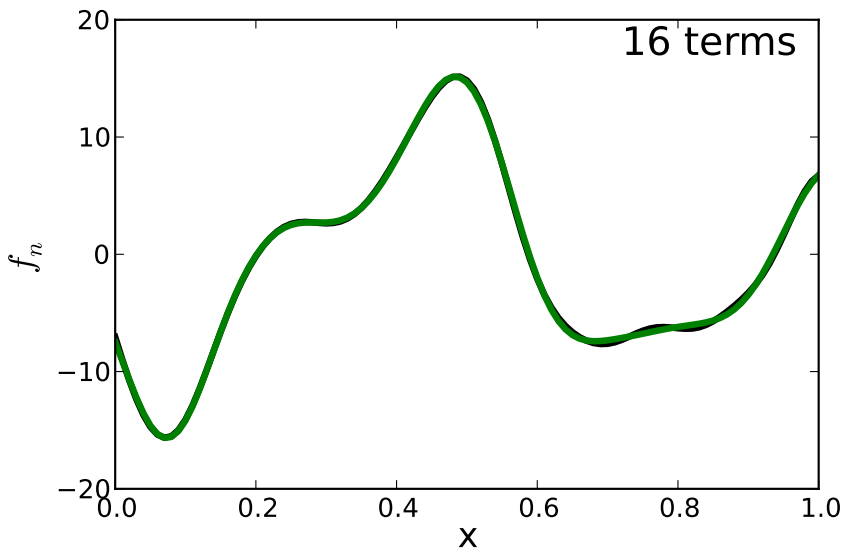




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

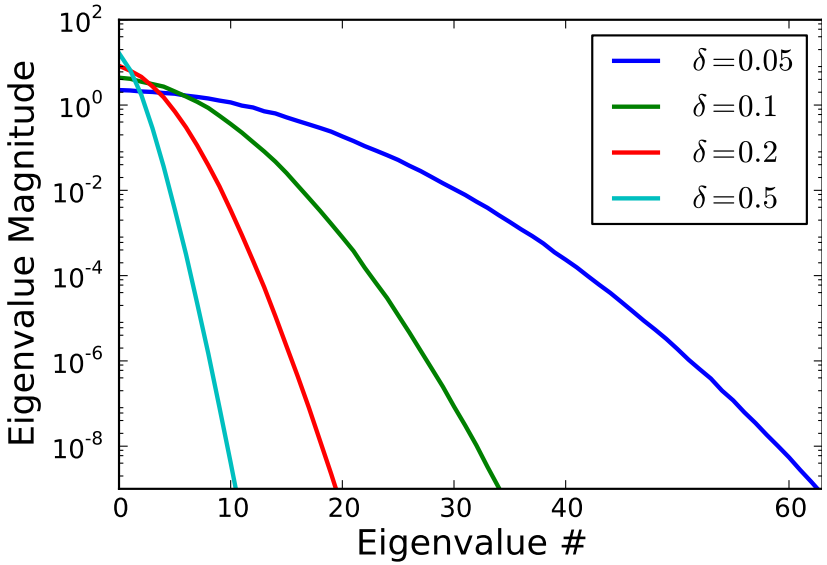
1D Gaussian Process: Reconstructed realizations

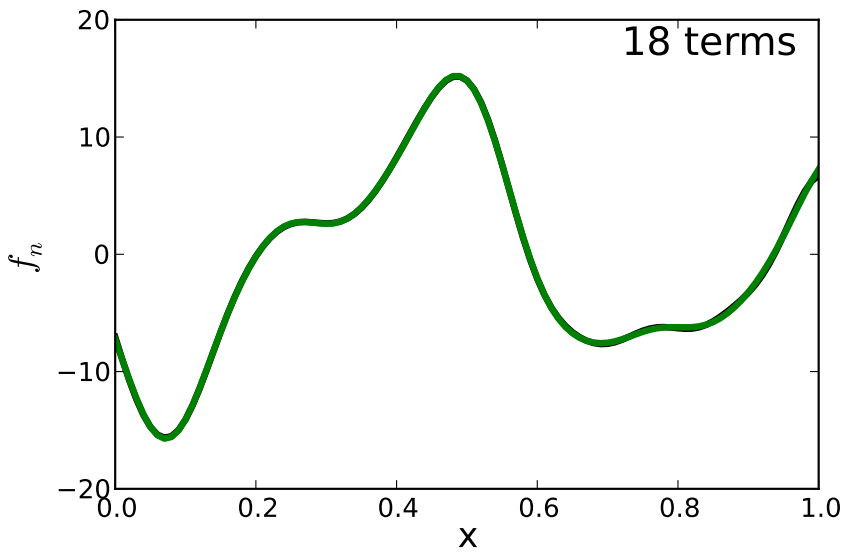




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

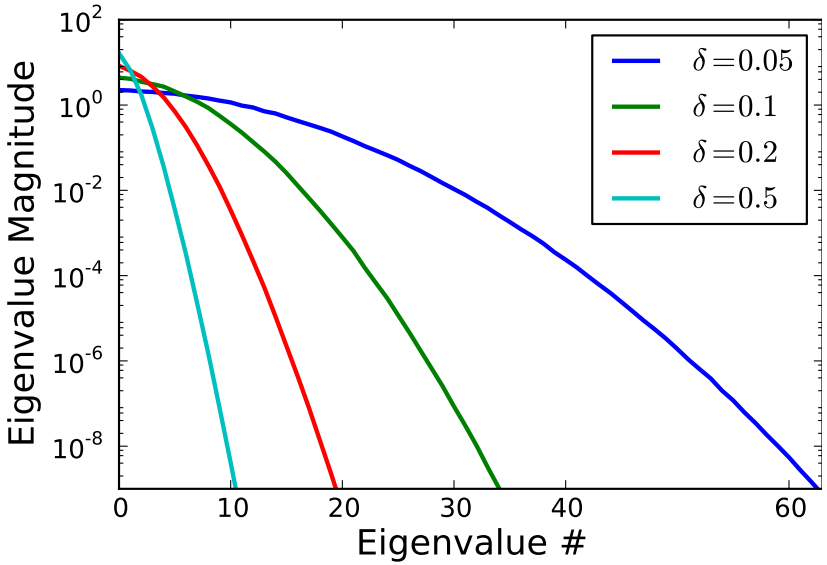
1D Gaussian Process: Reconstructed realizations

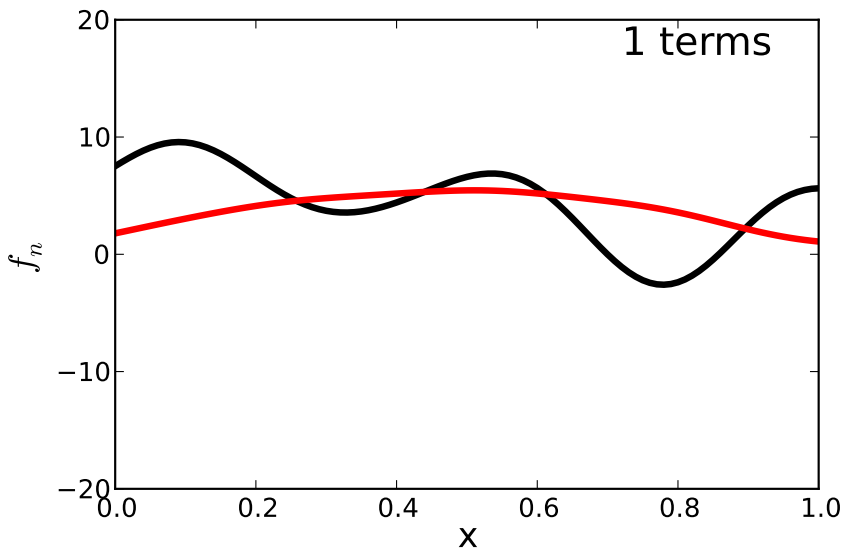




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

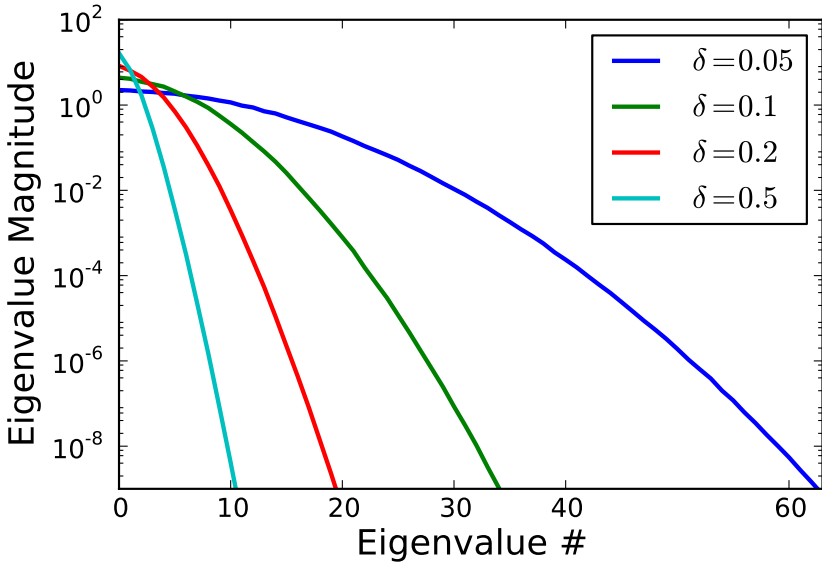
1D Gaussian Process: Reconstructed realizations

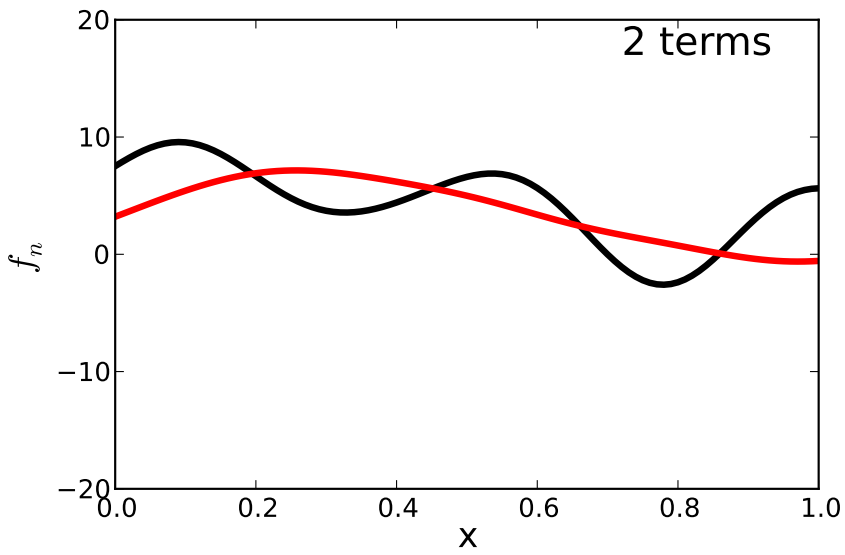




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

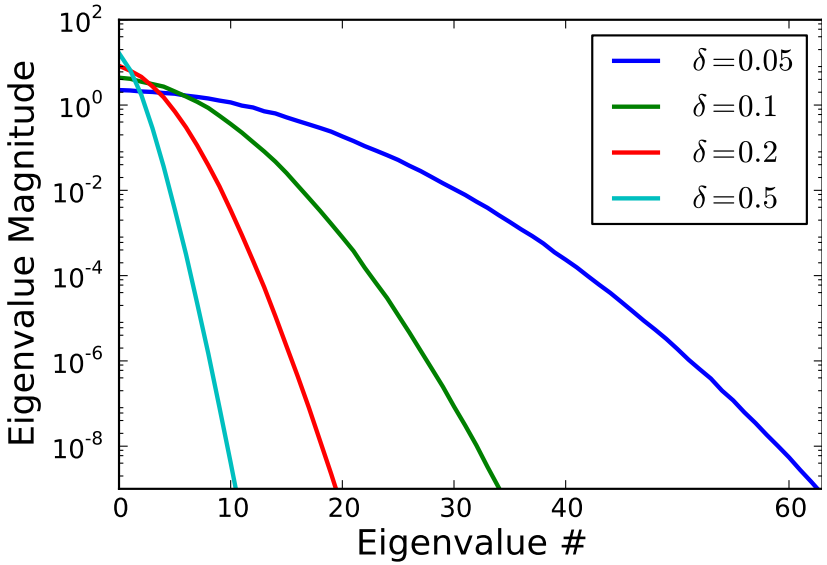
1D Gaussian Process: Reconstructed realizations

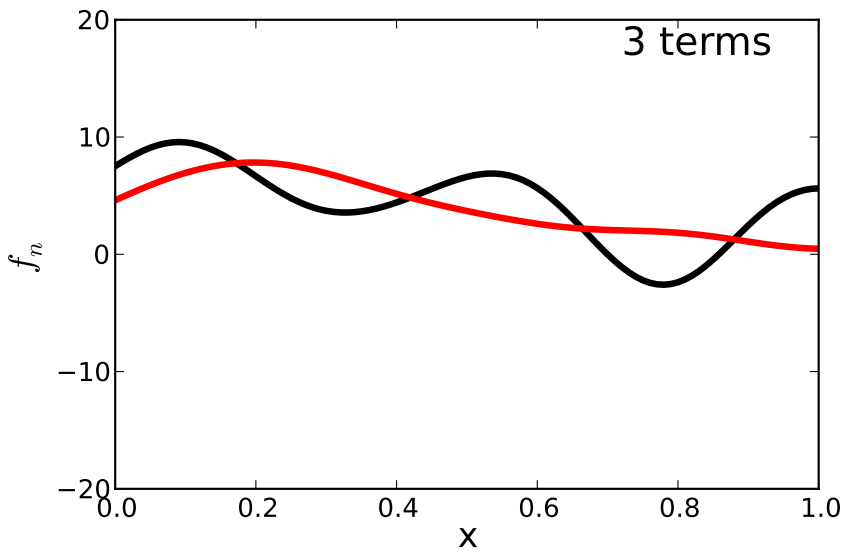




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

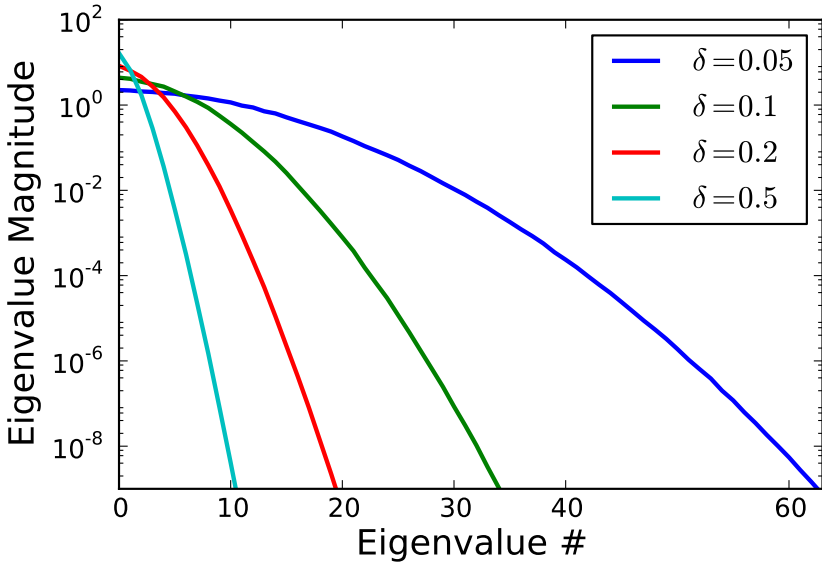
1D Gaussian Process: Reconstructed realizations

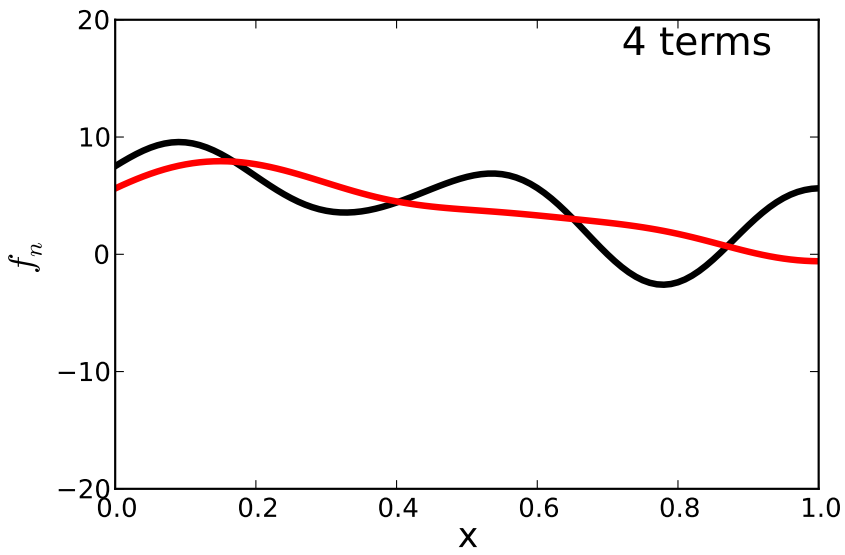




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

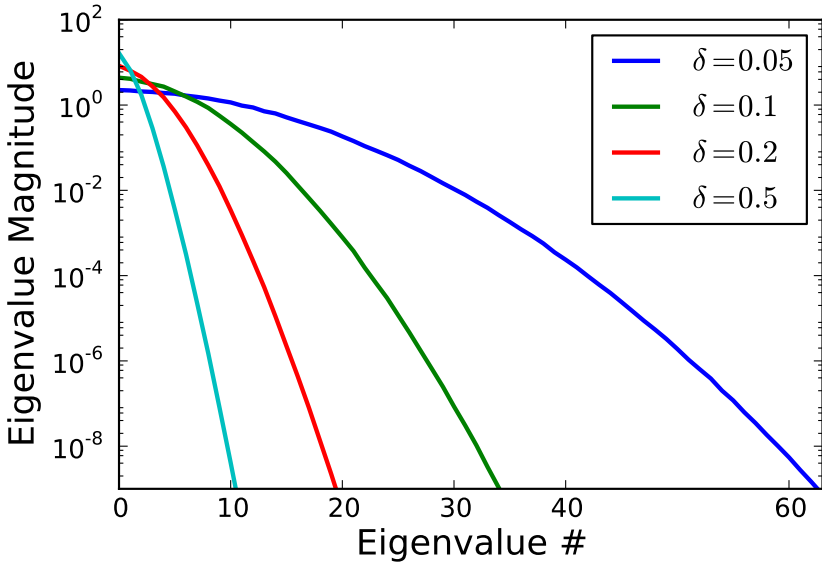
1D Gaussian Process: Reconstructed realizations

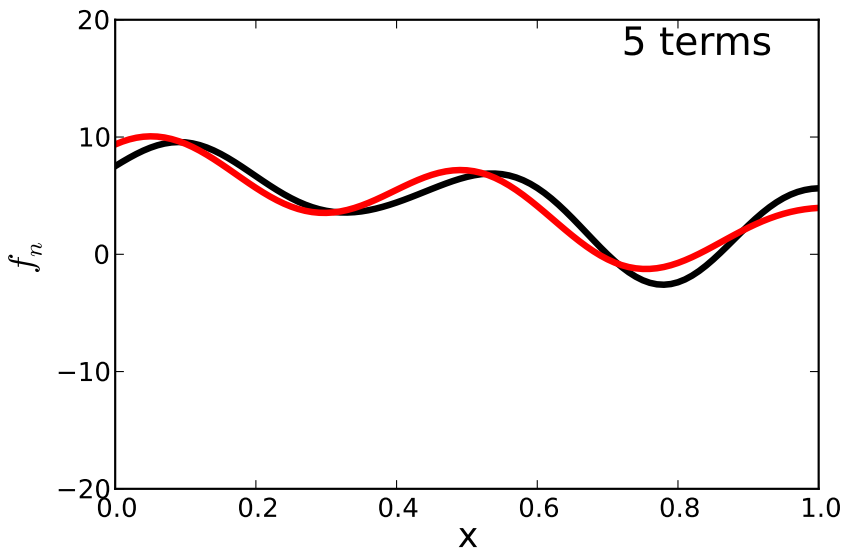




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

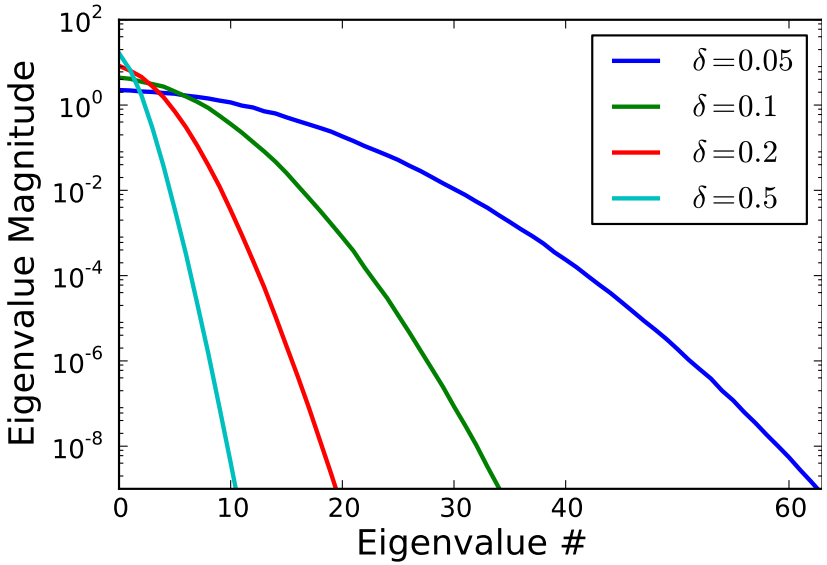
1D Gaussian Process: Reconstructed realizations

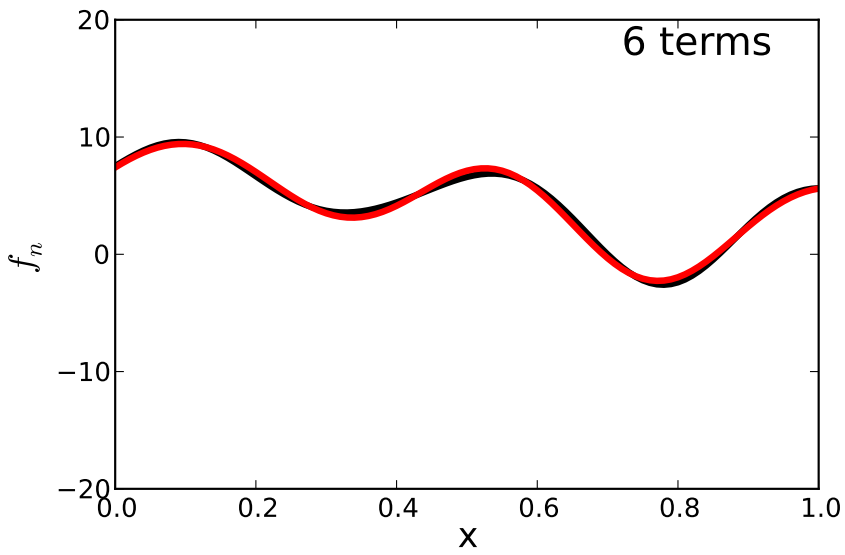




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

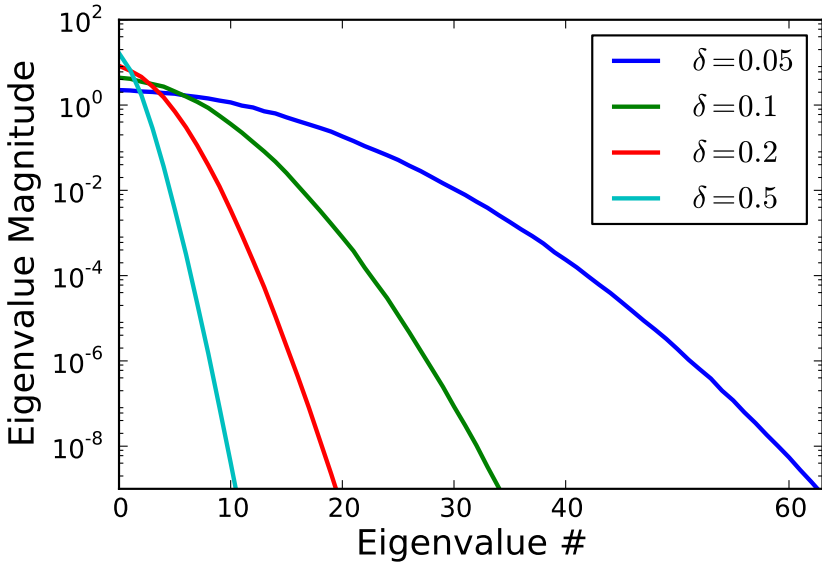
1D Gaussian Process: Reconstructed realizations

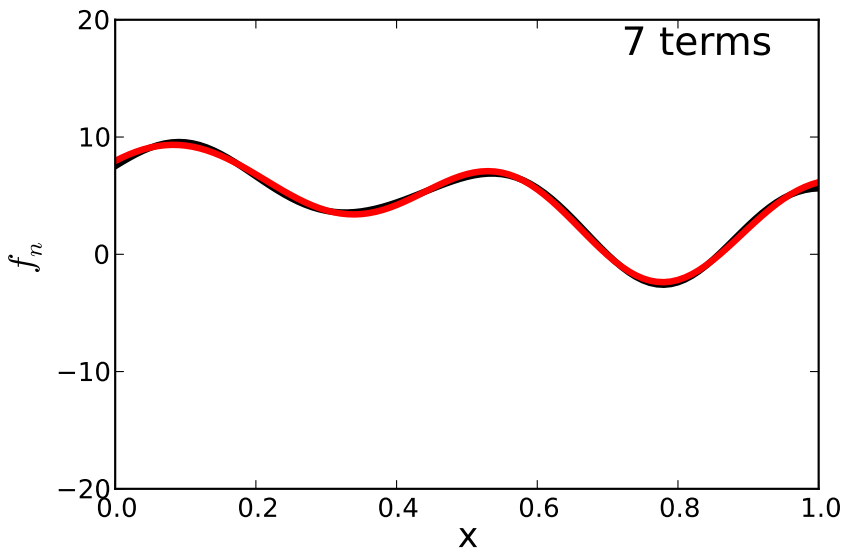




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

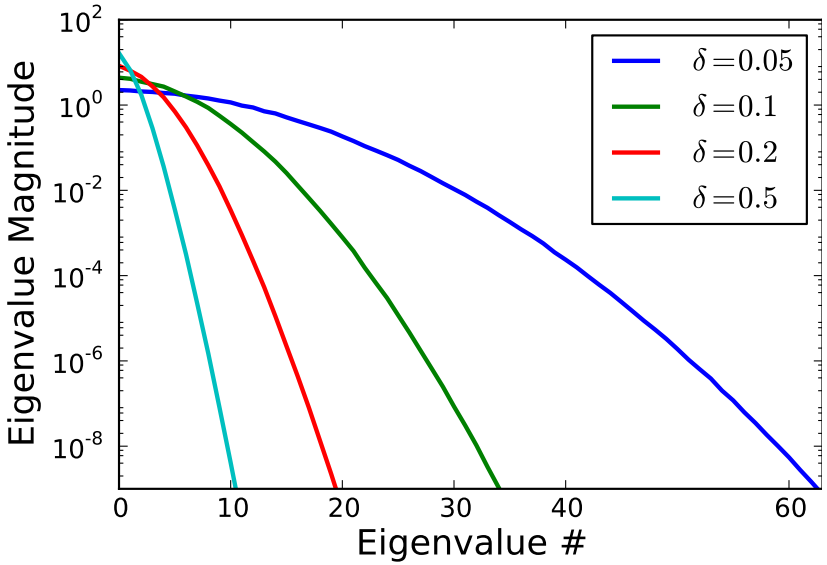
1D Gaussian Process: Reconstructed realizations

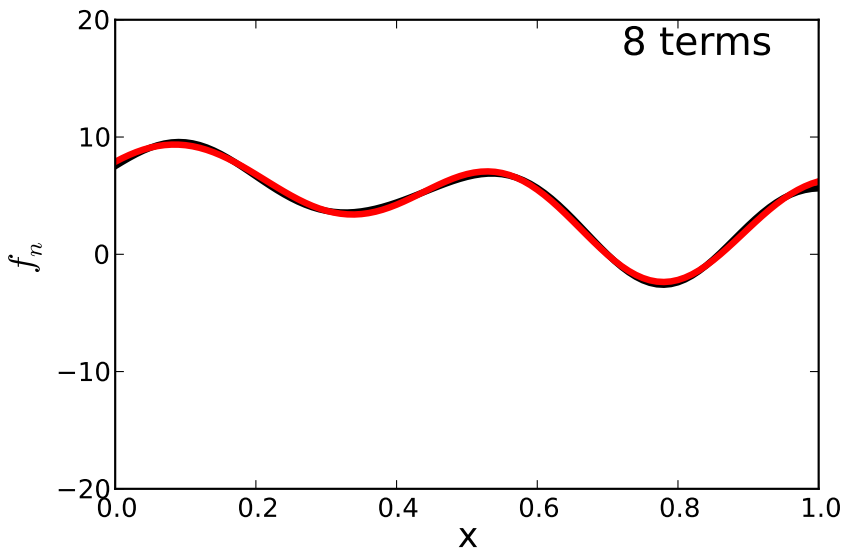




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

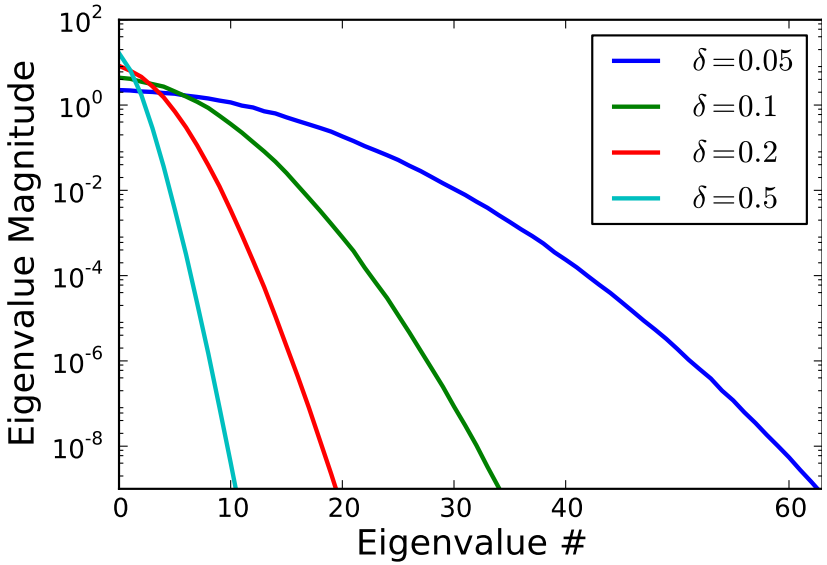
1D Gaussian Process: Reconstructed realizations

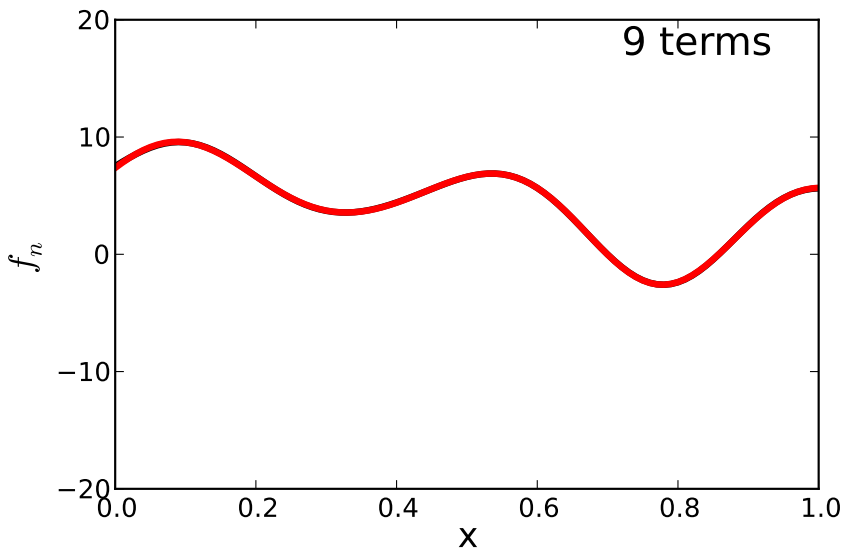




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

1D Gaussian Process: Reconstructed realizations

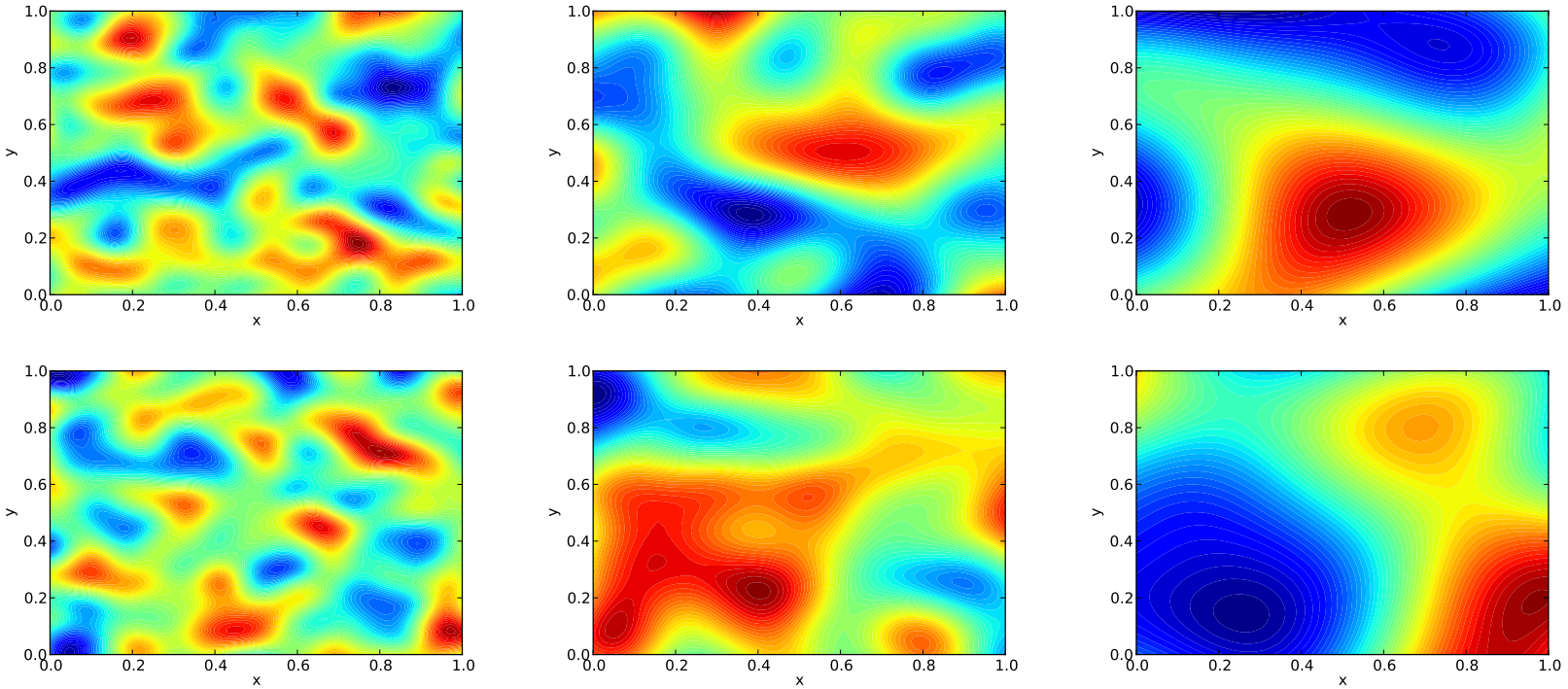




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

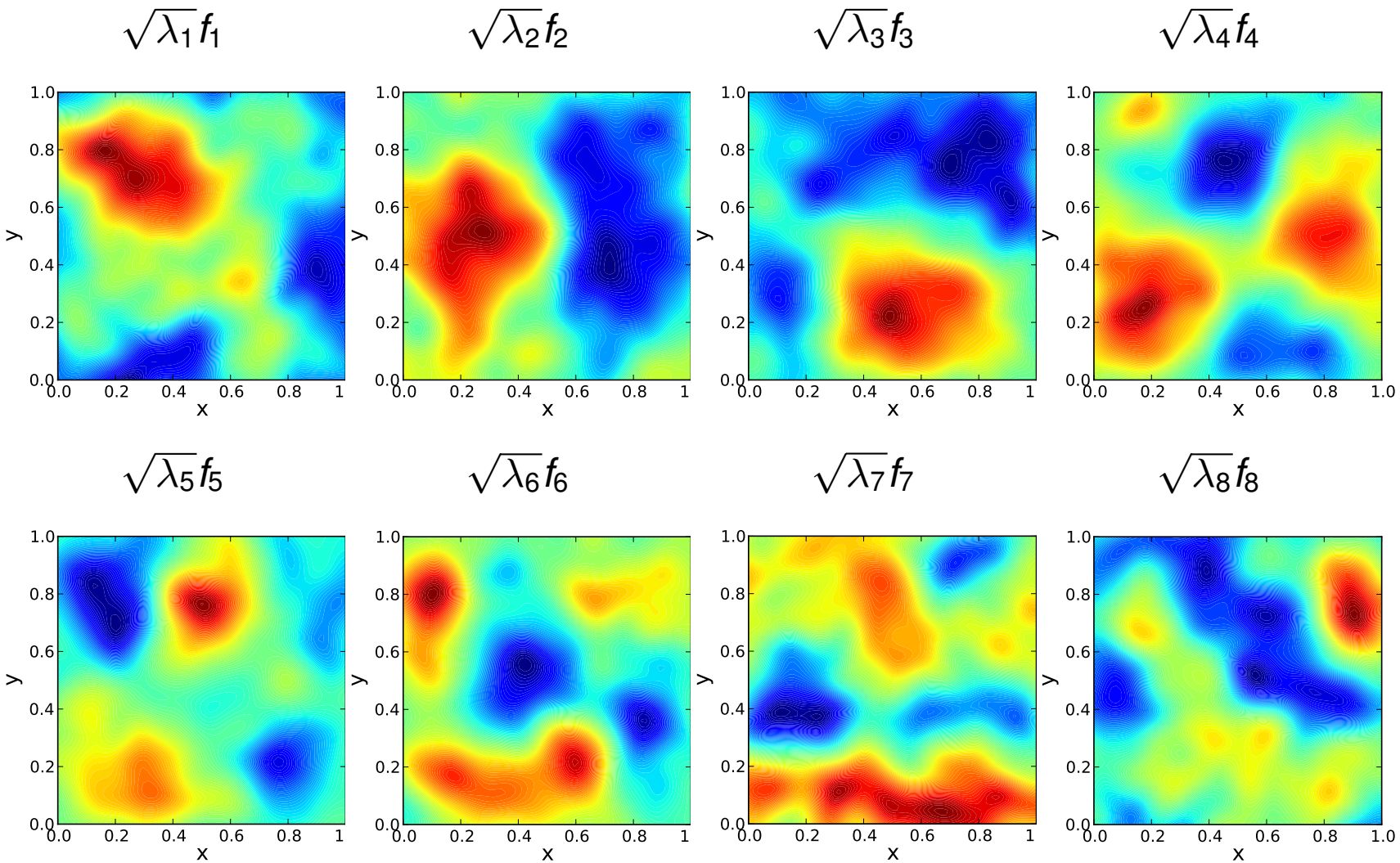
KL of 2D Gaussian Process

$\delta = 0.1$ $\delta = 0.2$ $\delta = 0.5$

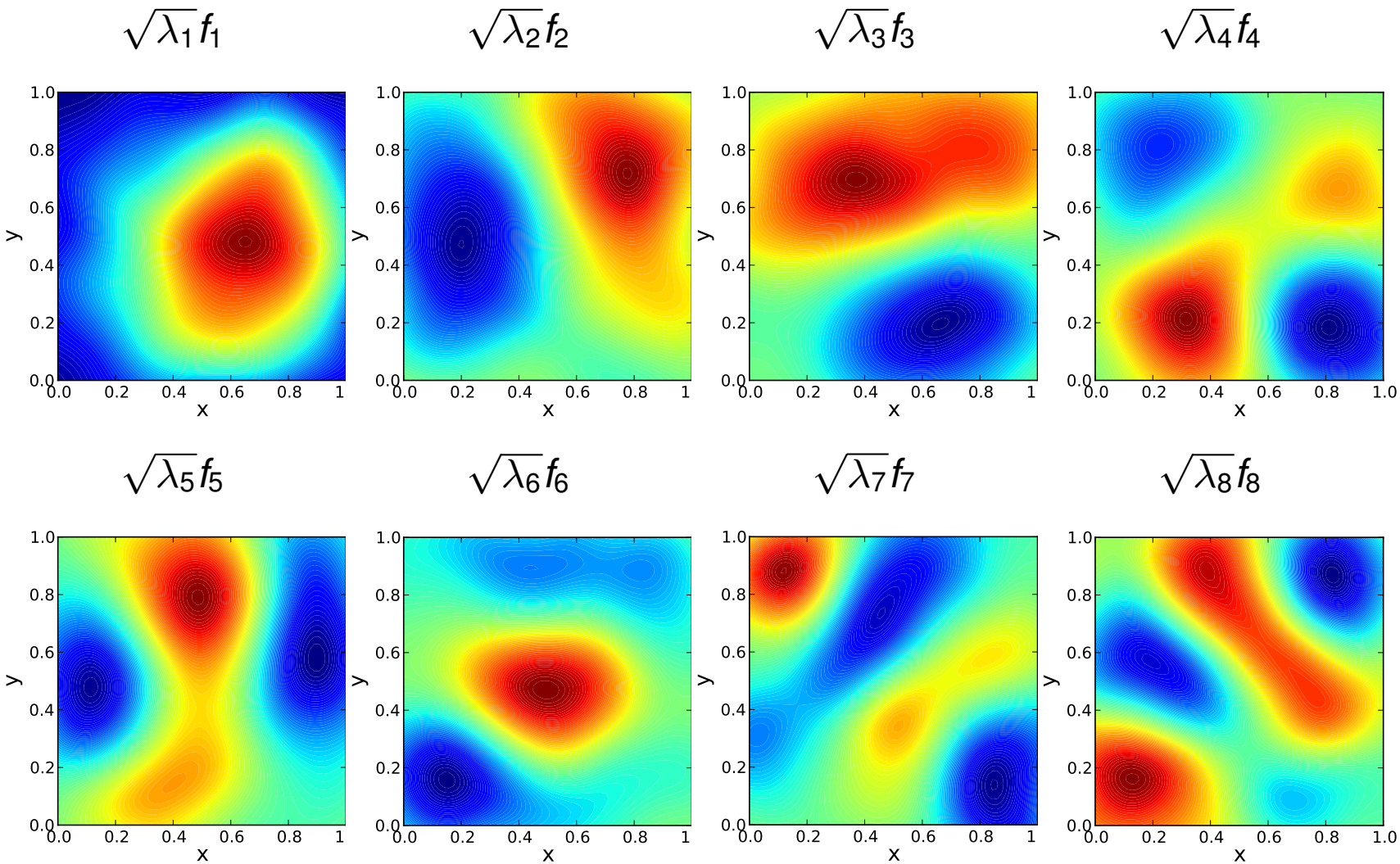


- 2D Gaussian Process with covariance:
  $Cov(x_1, x_2) = exp(-||x_1 - x_2||^2/\delta^2)$
- Realizations are smoother as covariance length δ increases

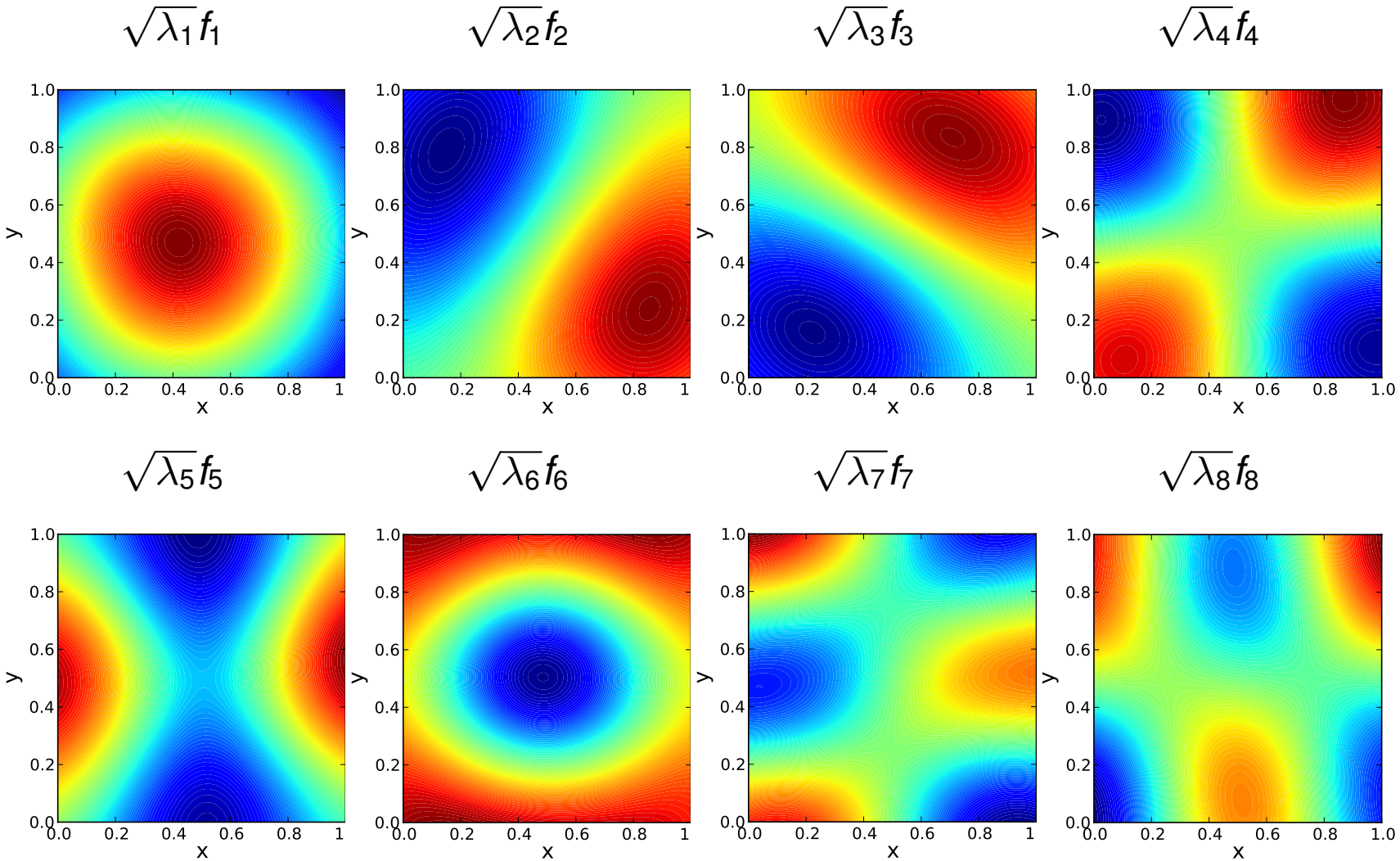
2D KL - Modes for $\delta = 0.1$



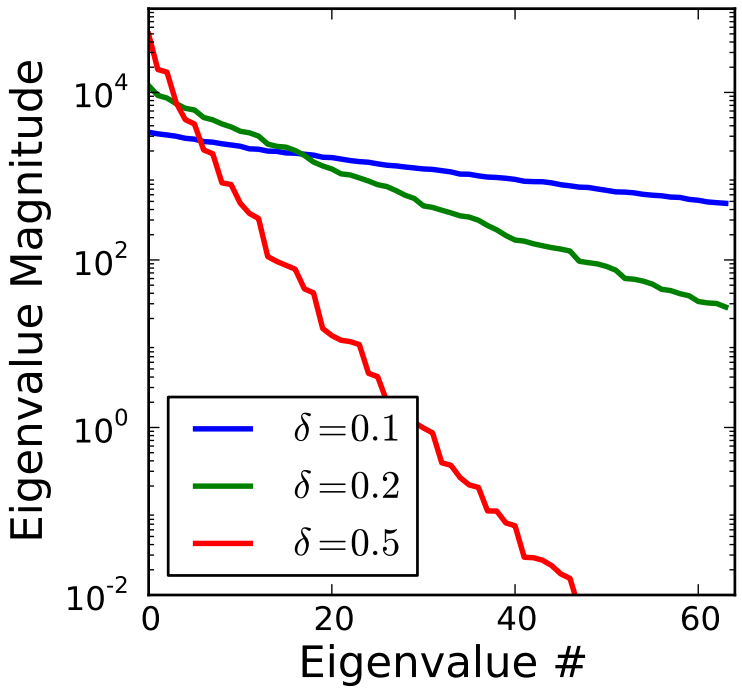
2D KL - Modes for $\delta = 0.2$



2D KL - Modes for $\delta = 0.5$

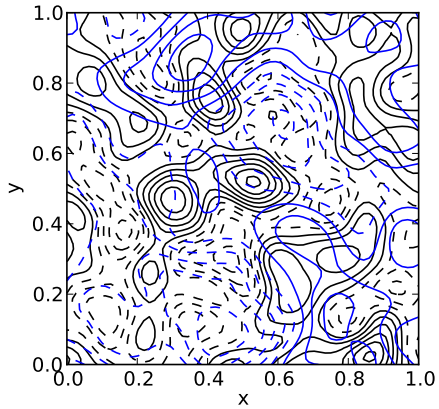


2D KL - eigenvalue spectrum

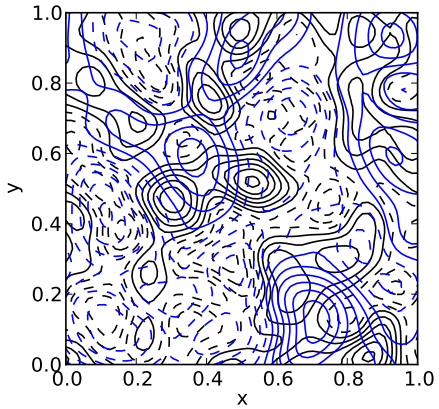


$\delta = 0.1$

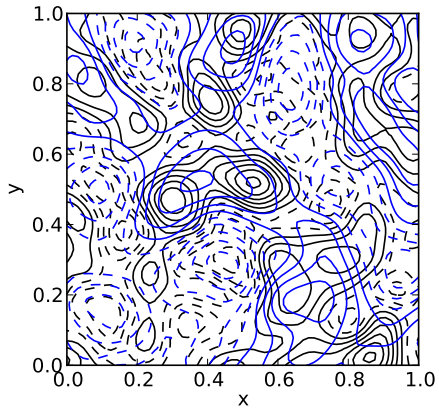
4 terms



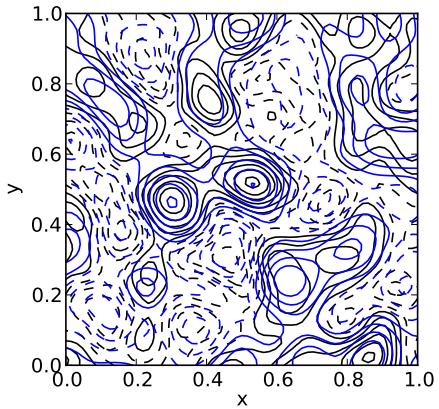
16 terms



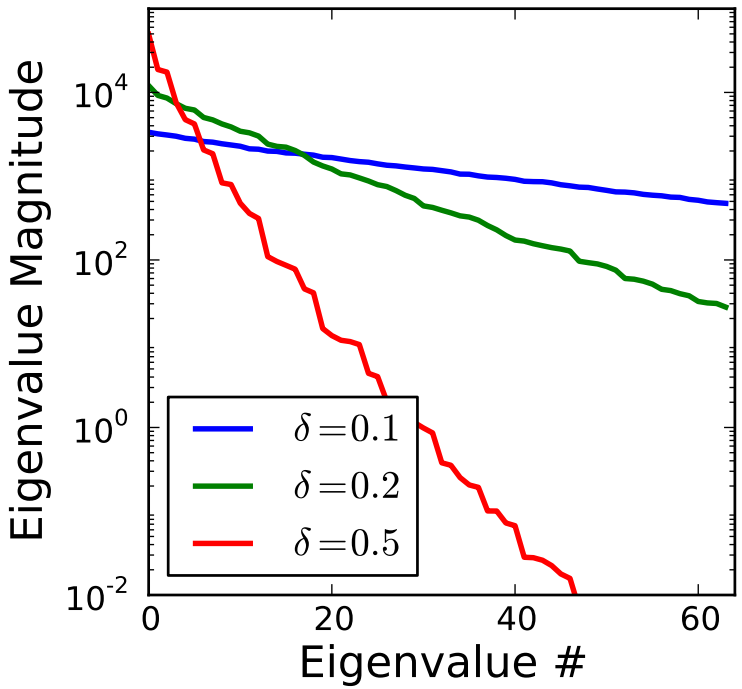
32 terms



64 terms

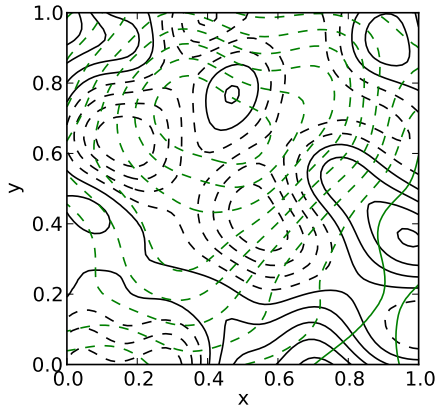


2D KL - eigenvalue spectrum

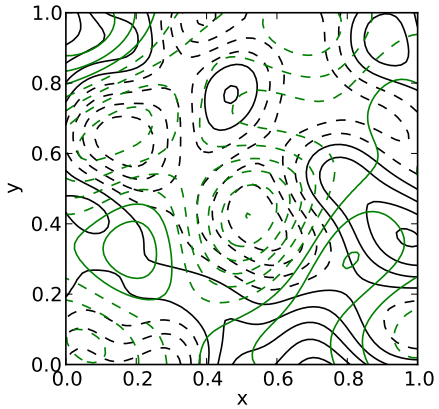


$\delta = 0.2$

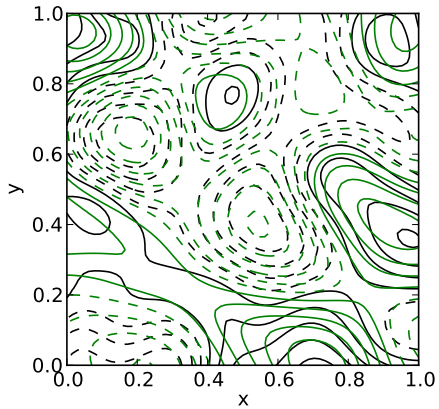
4 terms



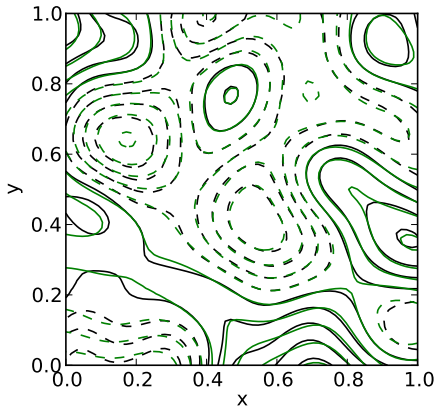
16 terms



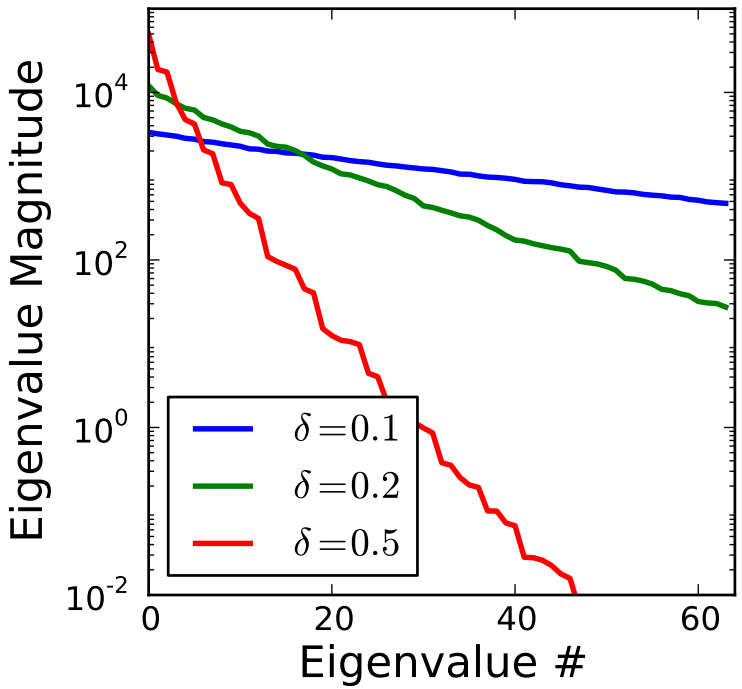
32 terms



64 terms

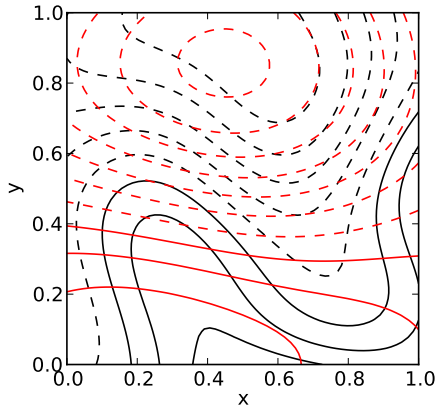


2D KL - eigenvalue spectrum

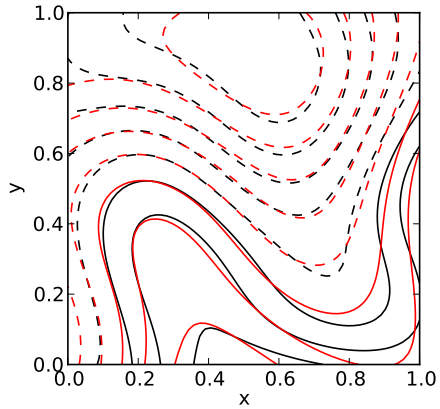


$\delta = 0.5$

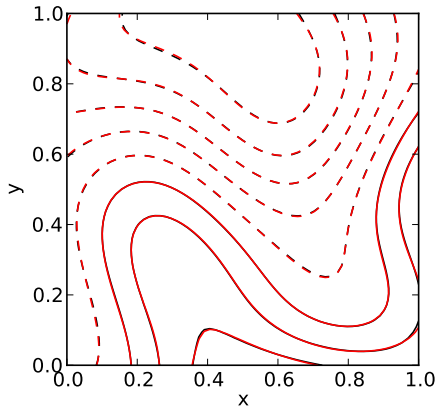
4 terms



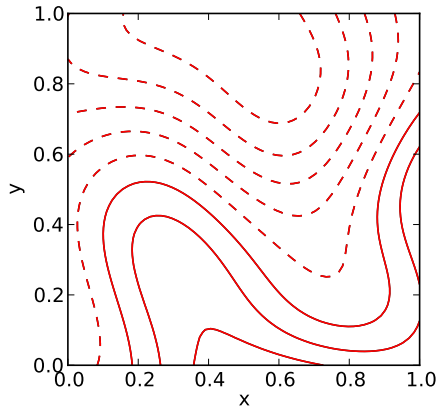
16 terms



32 terms



64 terms



Other approaches

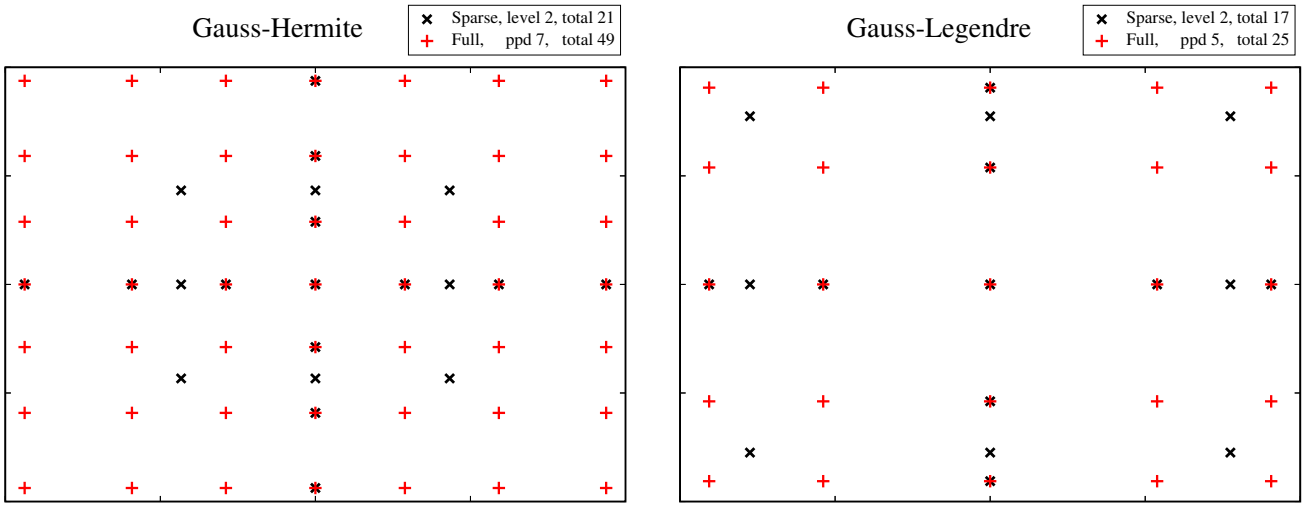
- Domain decomposition approaches with projection
- Inference-based approaches for PCEs

Sparse Quadrature Approaches for High-Dimensional Systems

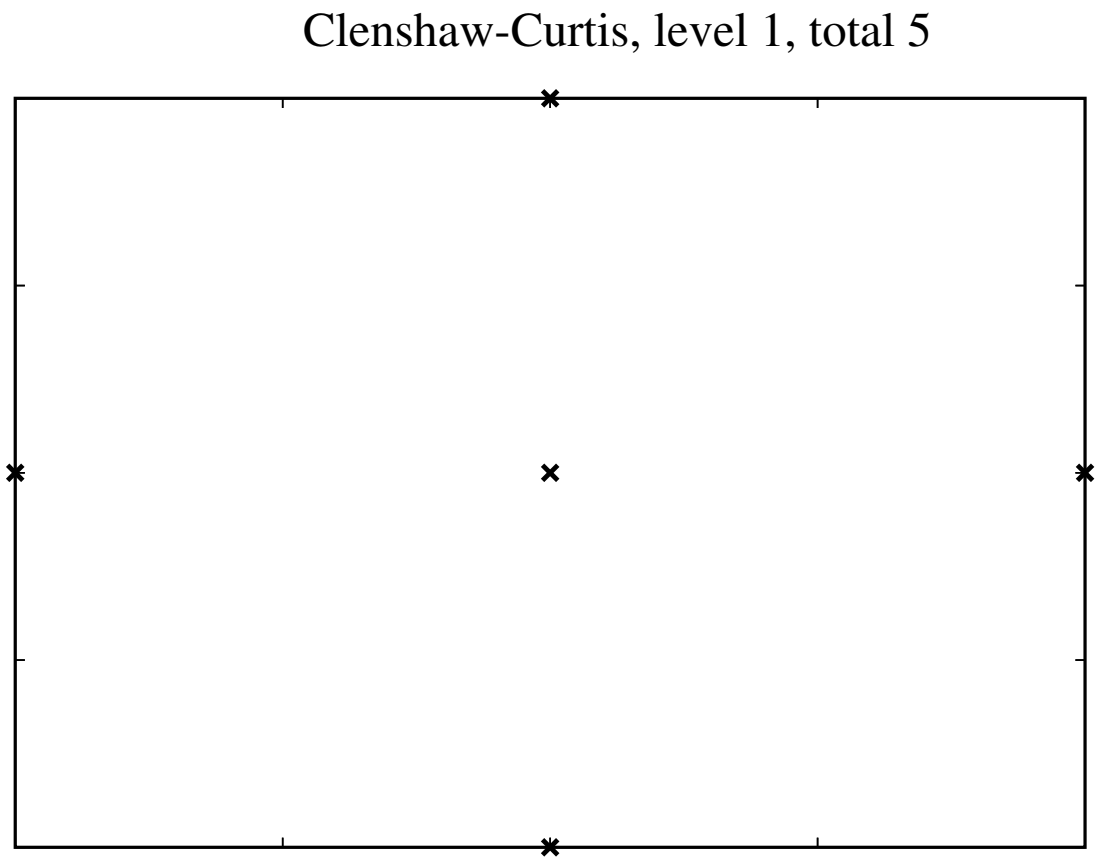
- Need for sparse quadrature
- Sparse quadrature grids
- Application to surface reaction example

Sparse quadrature drastically reduces the number of function evaluations

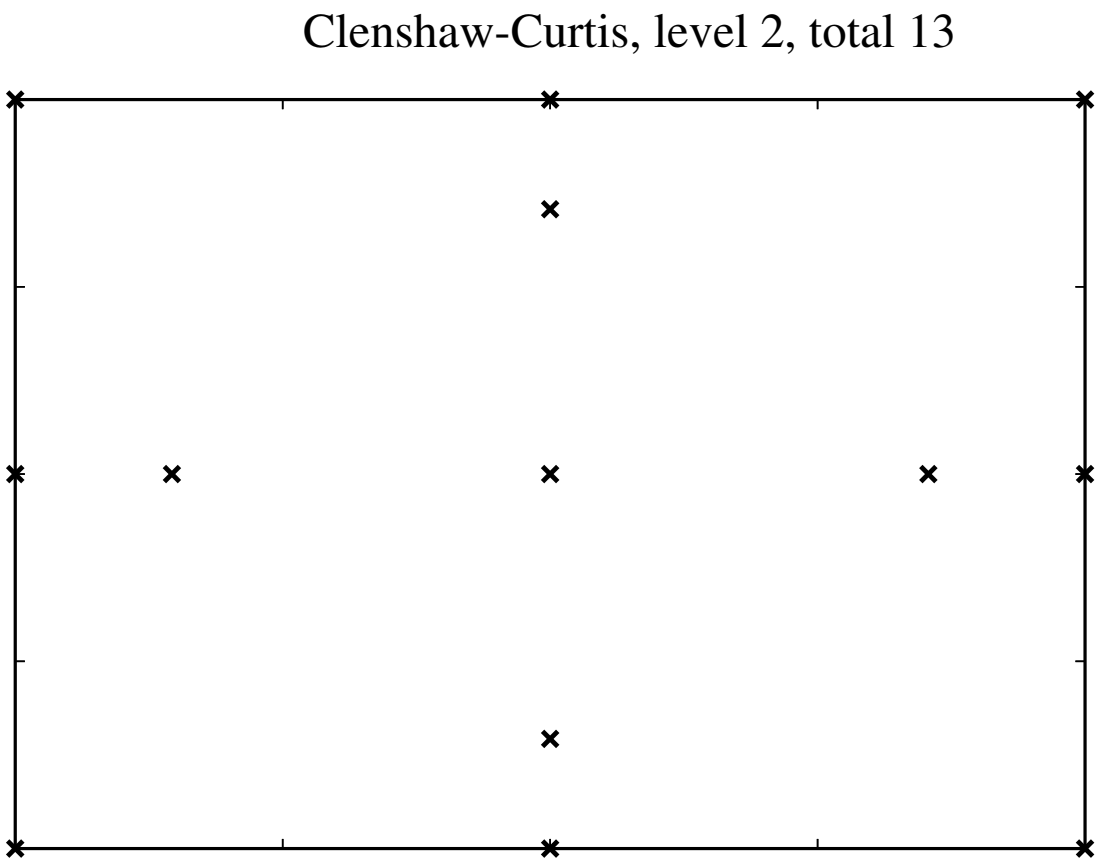
- Define the **precision** as the highest order of a polynomial that is integrated *exactly* by the quadrature rule.
- Gaussian quadratures are optimal in 1d
  - N points achieve the highest possible precision of $2N - 1$.
- In multi-d, full product quadrature is wasteful:
  - a 5 ppd (point per dimension) rule is of precision $P = 9$, but it integrates a polynomial $x^9 y^9$ exactly.
- Sparse grids are built to achieve maximal precision with fewest possible points



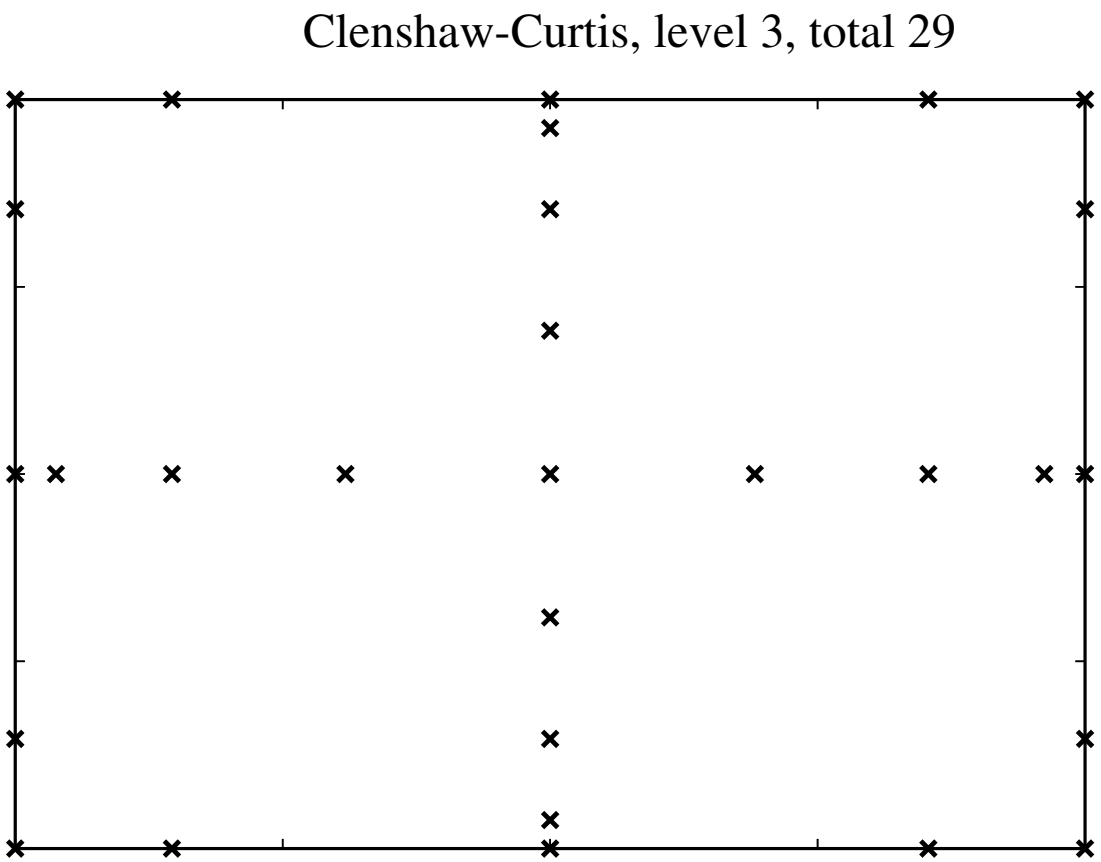
Clenshaw-Curtis nested quadrature rules allow reuse of function evalutions



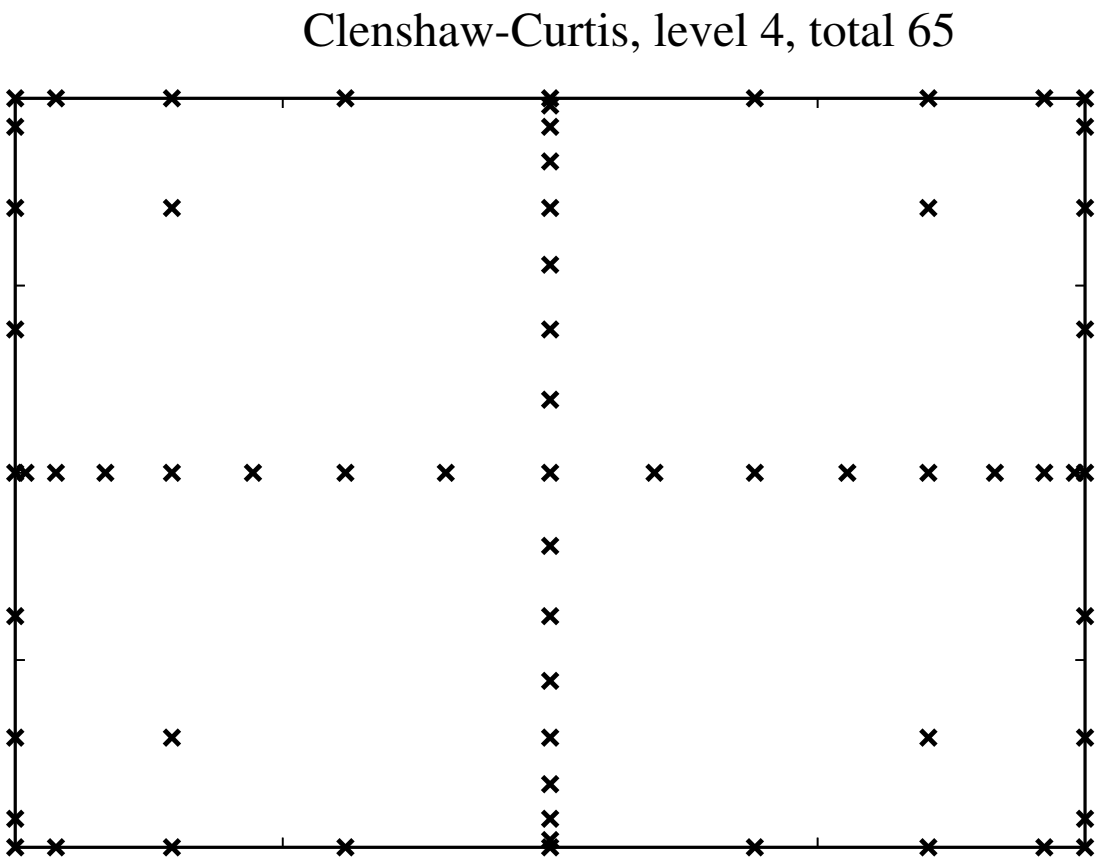
Clenshaw-Curtis nested quadrature rules allow reuse of function evalutions



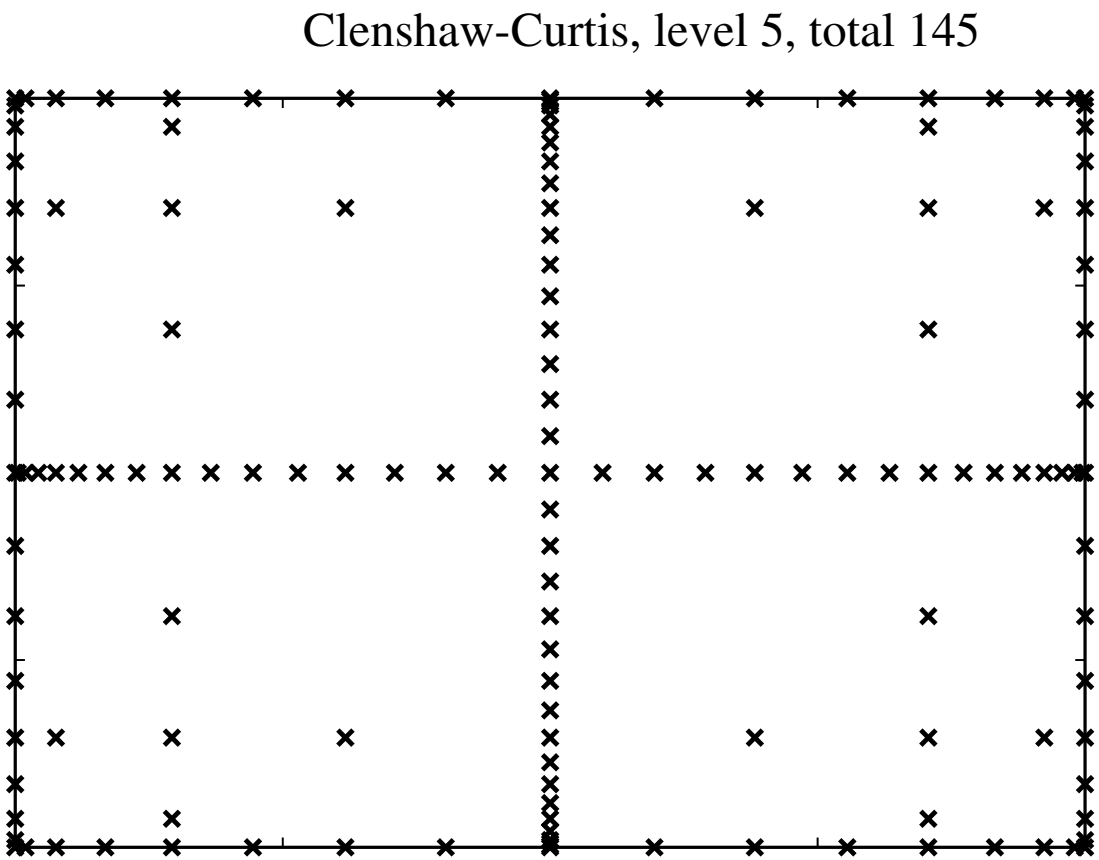
Clenshaw-Curtis nested quadrature rules allow reuse of function evalutions



Clenshaw-Curtis nested quadrature rules allow reuse of function evalutions



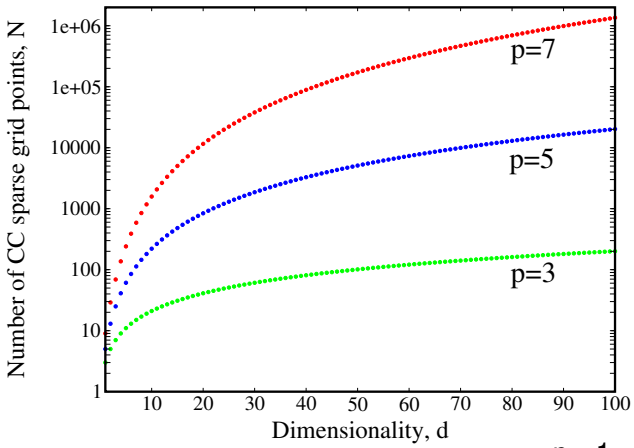
Clenshaw-Curtis nested quadrature rules allow reuse of function evalutions



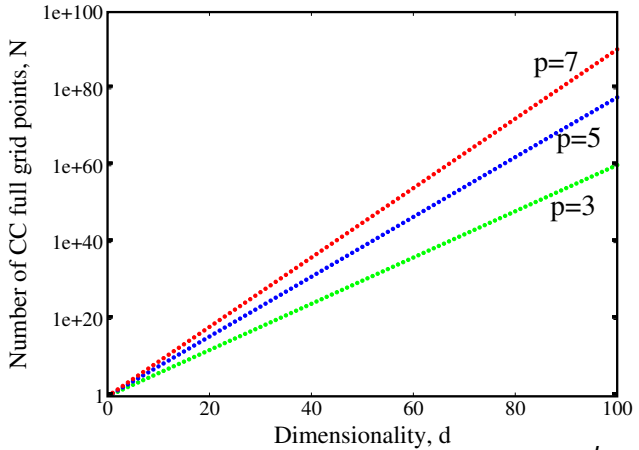
The number of function evaluations is drastically reduced compared to full quadrature

Table: The number of Clenshaw-Curtis sparse grid quadrature points for various levels and dimensionalities.

Level L	Precision $p = 2L - 1$	N $(d = 2)$	N $(d = 5)$	N $(d = 10)$	$N(d)$ General
1	1	1	1	1	1
2	3	5	11	21	$1 + 2d$
3	5	13	61	221	$1 + 2d + 2d^2$
4	7	29	241	1581	$1 + \frac{14}{3}d + 2d^2 + \frac{4}{3}d^3$
5	9	65	801	8801	$1 + \frac{20}{3}d + \frac{22}{3}d^2 + \frac{4}{3}d^3 + \frac{2}{3}d^4$
6	11	145	2433	41265	$\cdots$
7	13	321	6993	171425	$\cdots$
8	15	705	19313	652065	$\cdots$



Sparse grid: polynomial growth, $\mathcal{O}(d^{\frac{p-1}{2}})$



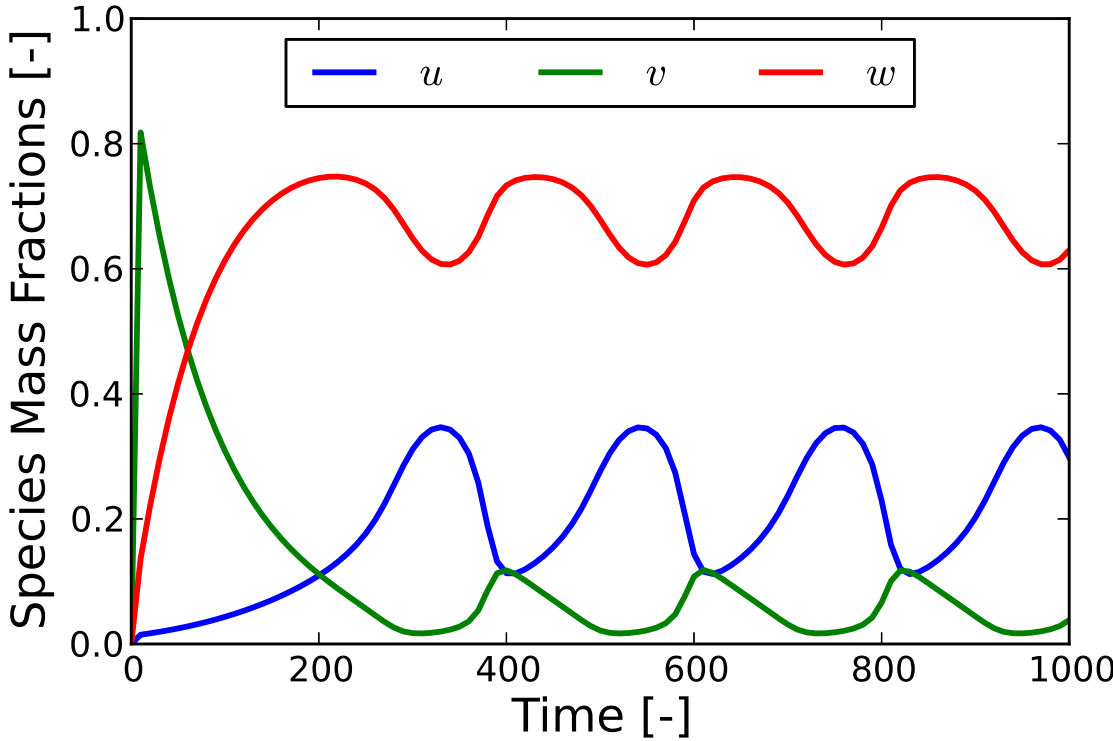
Full grid: exponential growth, $(p + 1)^d$

Surface Reaction Model

3 ODEs for a monomer (u), dimer (v), and inert species (w) adsorbing onto a surface out of gas phase.

$$\frac{du}{dt} = az - cu - 4duv$$
$$\frac{dv}{dt} = 2bz^2 - 4duv$$
$$\frac{dw}{dt} = ez - fw$$
$$z = 1 - u - v - w$$

$$u(0) = v(0) = w(0) = 0.0$$

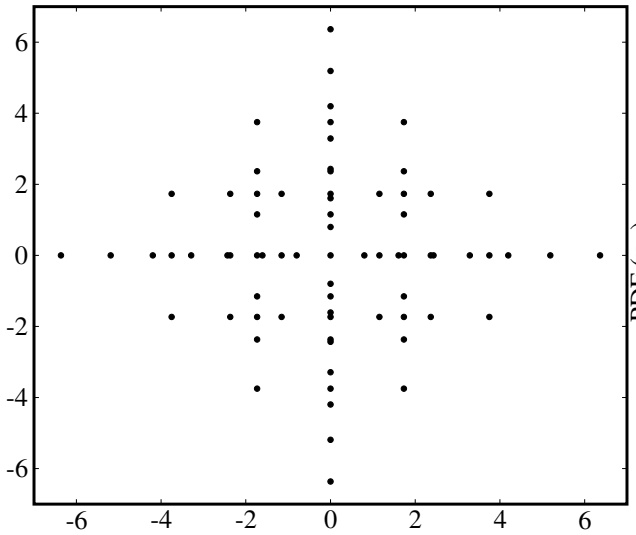


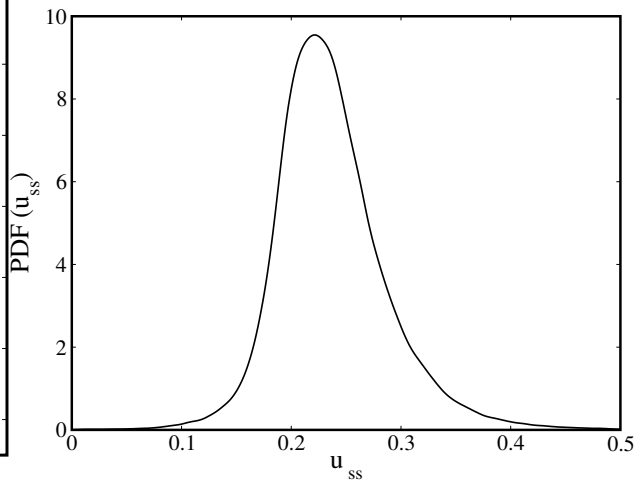
Oscillatory behavior for $b \in [20.2, 21.2]$

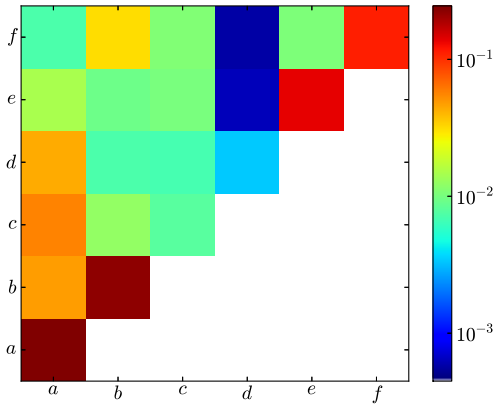
$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad d = 1.0 \quad e = 0.36 \quad f = 0.016$

(Vigil *et al.*, Phys. Rev. E., 1996; Makeev *et al.*, J. Chem. Phys., 2002)

Surface Reaction Model: 6d results







- Output observable: time averaged u at steady state u_{ss}
- Assume all input parameters have Gaussian distributions with $\sigma/\mu = 0.01$, i.e. 1% deviation.
- 6-d, level 3 Gauss-Hermite sparse quadrature point set includes 713 distinct points (a 2-d case is plotted)
- Output PDF is generated by $100K$ samples of a third order PC
- Variance-based sensitivity information comes for free with the PC expansion

Advantages and caveats of sparse quadrature approaches

- Pro: number of required samples scales much more gracefully with number of dimensions than full tensor product quadrature rule
- Caveats:
  - Function to be integrated needs to be smooth
  - Due to negative quadrature weights, integrating a noisy positive function can give a negative answer
  - For very high dimensions, even sparse quadrature is too expensive

Taking Advantage of Sparsity in the System

- For really high dimensional systems, even sparse quadrature requires too many function evaluations
  - For 80-dimensional climate land model, $L = 4$ requires $\approx 10^6$ points
- Such systems can only be tackled with dimensionality reduction and/or adaptive order
  - Sensitivity analysis
  - High Dimensional Model Representation (HDMR)
  - Adaptive sparse quadrature approaches
- More generally, use only the basis terms needed to represent the physics / information in the system / data
  - (Bayesian) Compressive Sensing (CS) approaches
- If information content is sparse, it can be represented at reasonable cost
  - If not, you need to pay the price

Challenges with Oscillatory / Long Time Horizon Systems

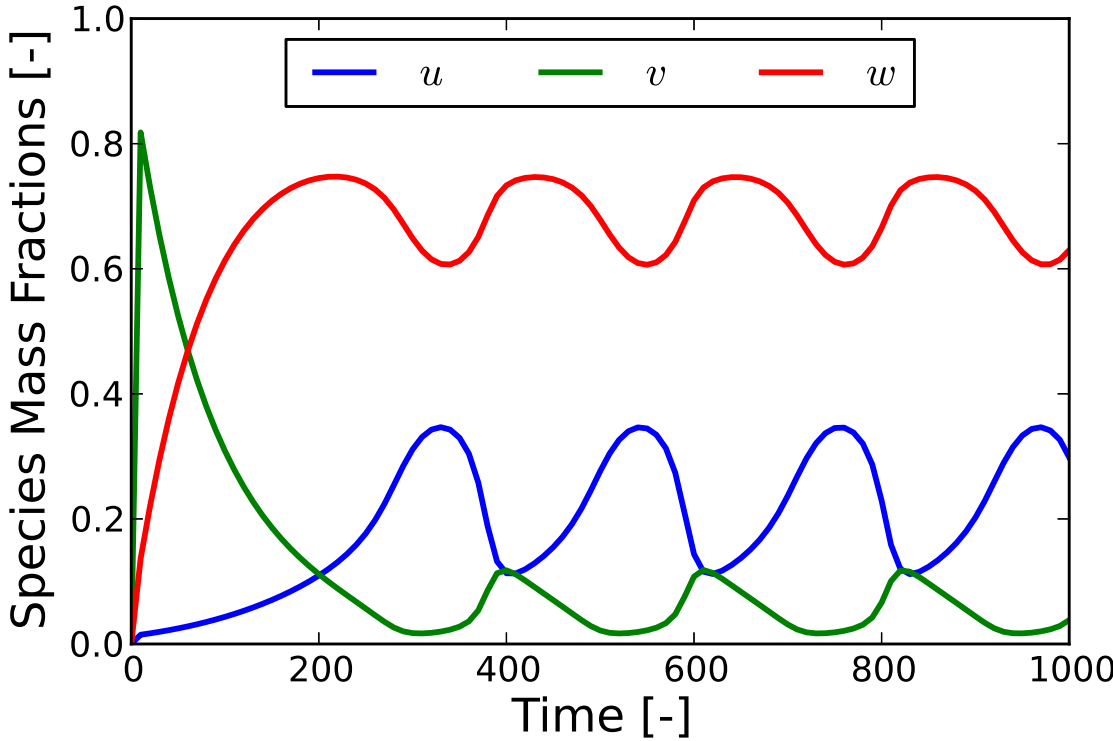
- Issue with oscillations / long time horizon
- Importance of choice of observables

Surface Reaction Model

3 ODEs for a monomer (u), dimer (v), and inert species (w) adsorbing onto a surface out of gas phase.

$$\frac{du}{dt} = az - cu - 4duv$$
$$\frac{dv}{dt} = 2bz^2 - 4duv$$
$$\frac{dw}{dt} = ez - fw$$
$$z = 1 - u - v - w$$

$$u(0) = v(0) = w(0) = 0.0$$

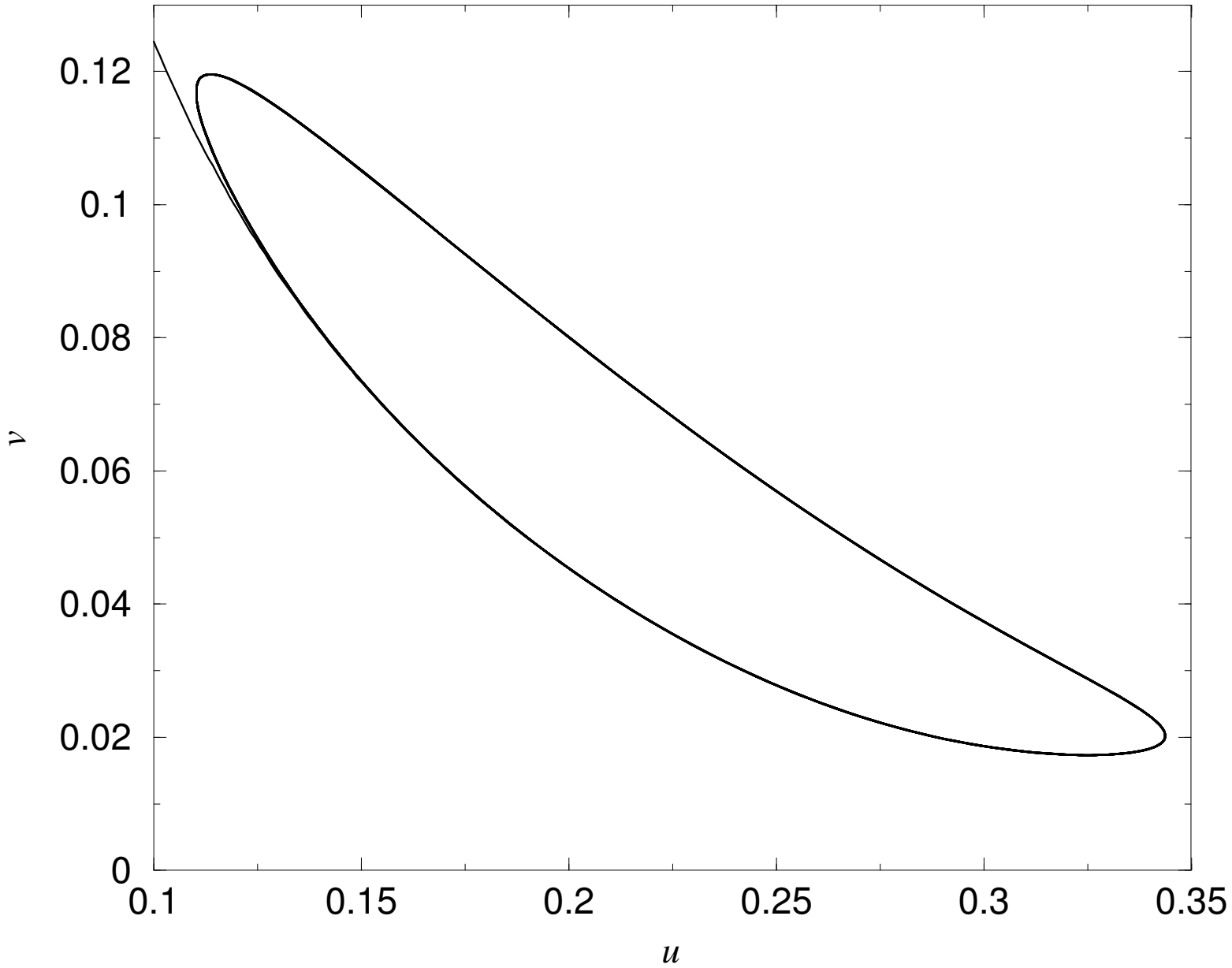


Oscillatory behavior for $b \in [20.2, 21.2]$

$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad d = 1.0 \quad e = 0.36 \quad f = 0.016$

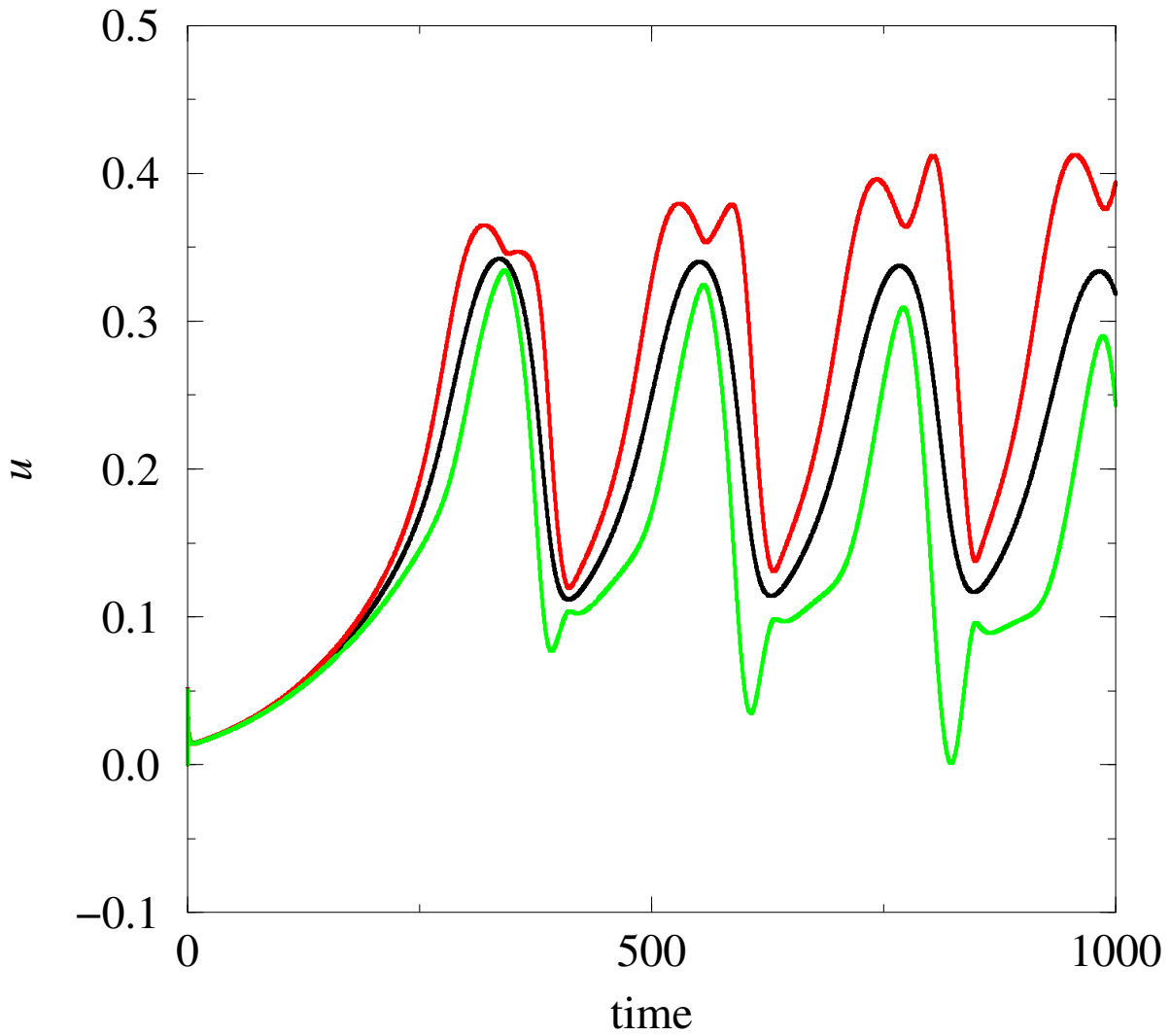
(Vigil *et al.*, Phys. Rev. E., 1996; Makeev *et al.*, J. Chem. Phys., 2002)

Limit Cycle Orbit in Phase Space, v vs u

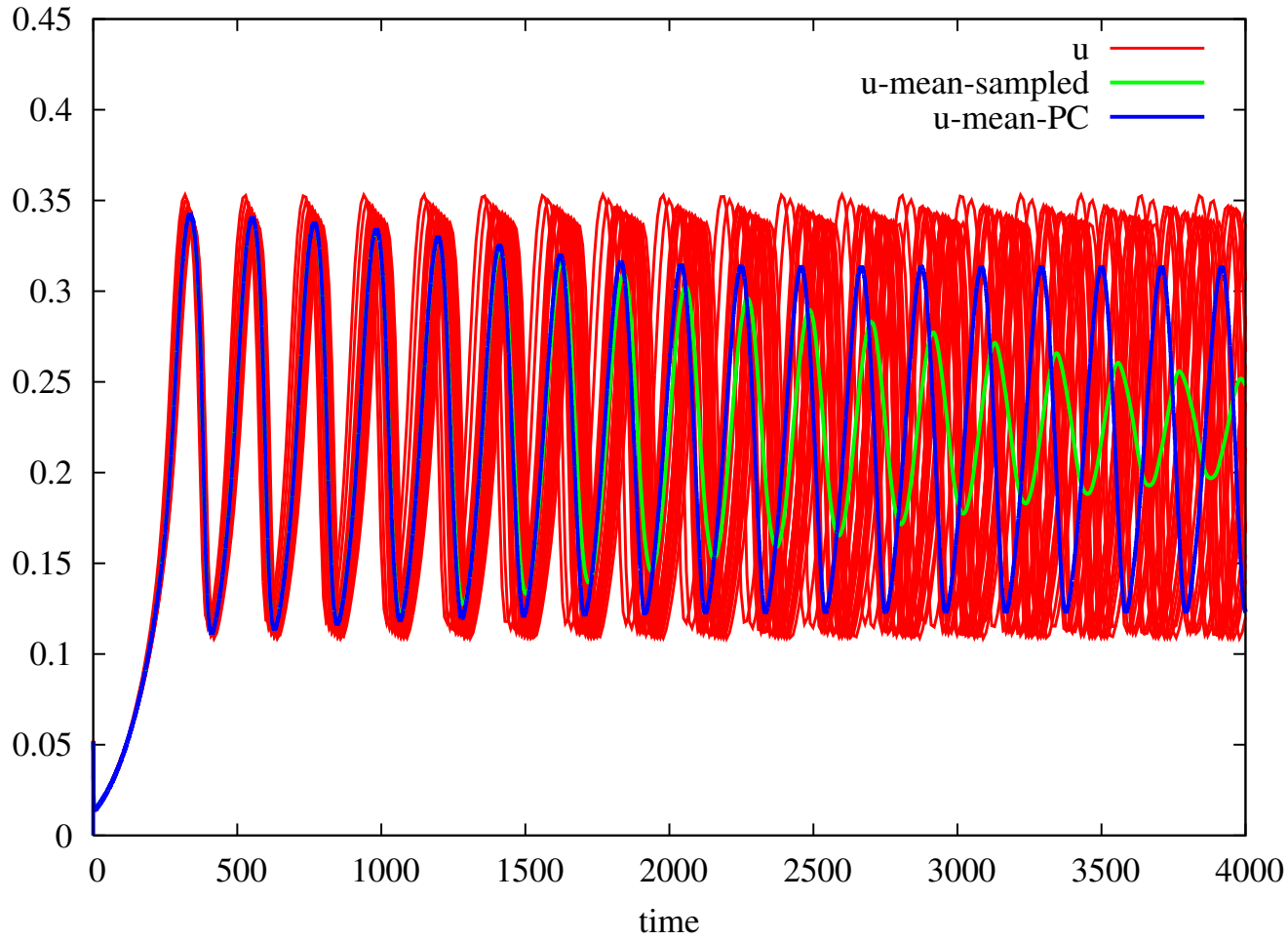


4*th*-order Intrusive PC UQ

- Uncertain b
- Wiener-Hermite PC
- Mean and $\pm 3\sigma$ bounds



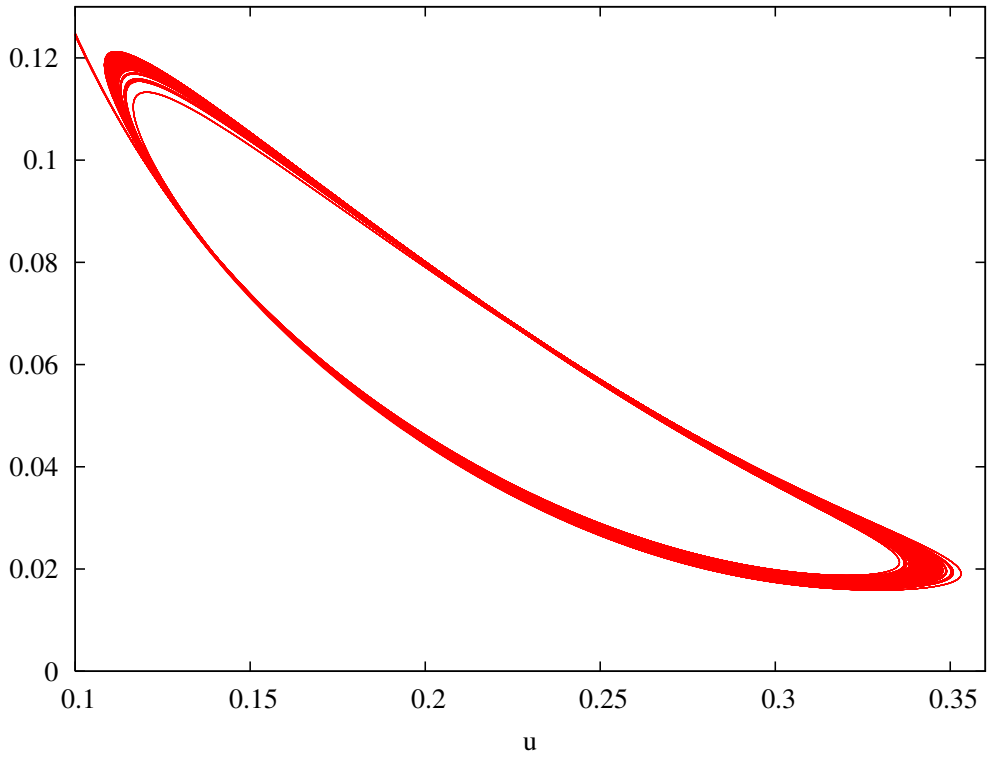
Growth of Phase Errors in Time



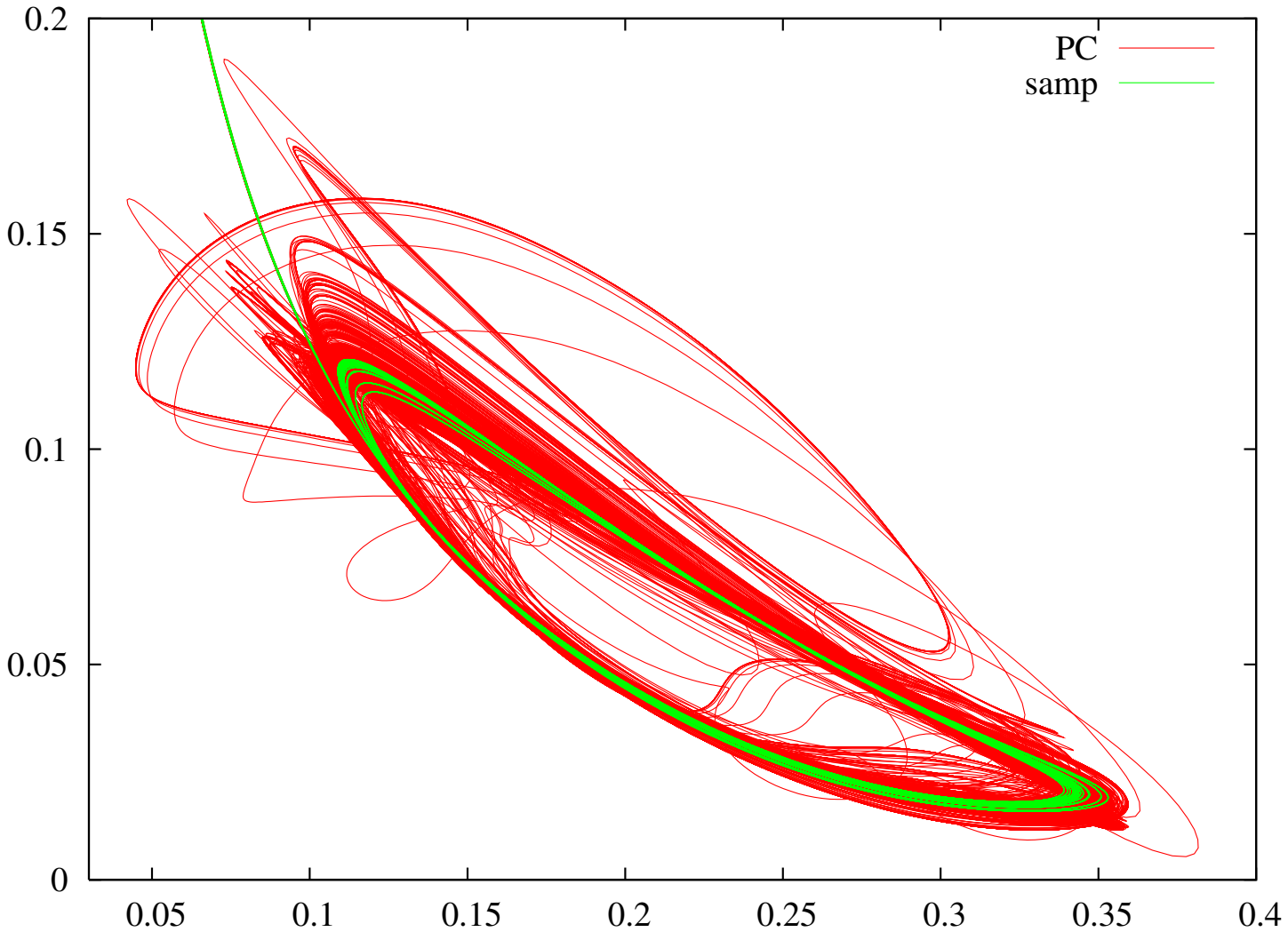
- PC mean deviates away from the sampled-mean in time
- Large phase variances

Scatter in State Space

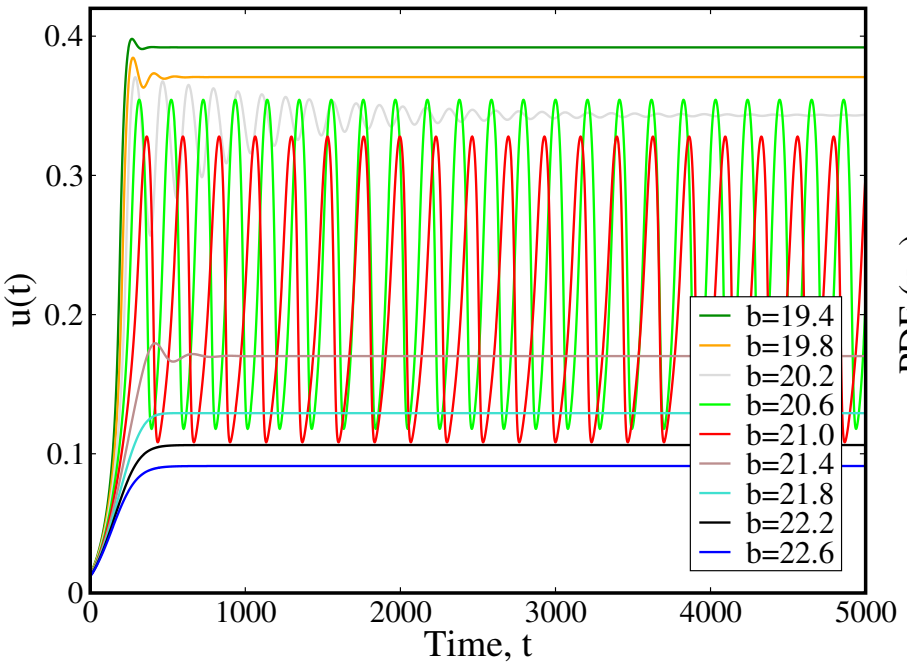
- Much better behaved uncertainty in state space
- Phase variances are not of interest per se
- Knowledge of the uncertainty in the orbit details are more of interest than the detailed phase variance errors

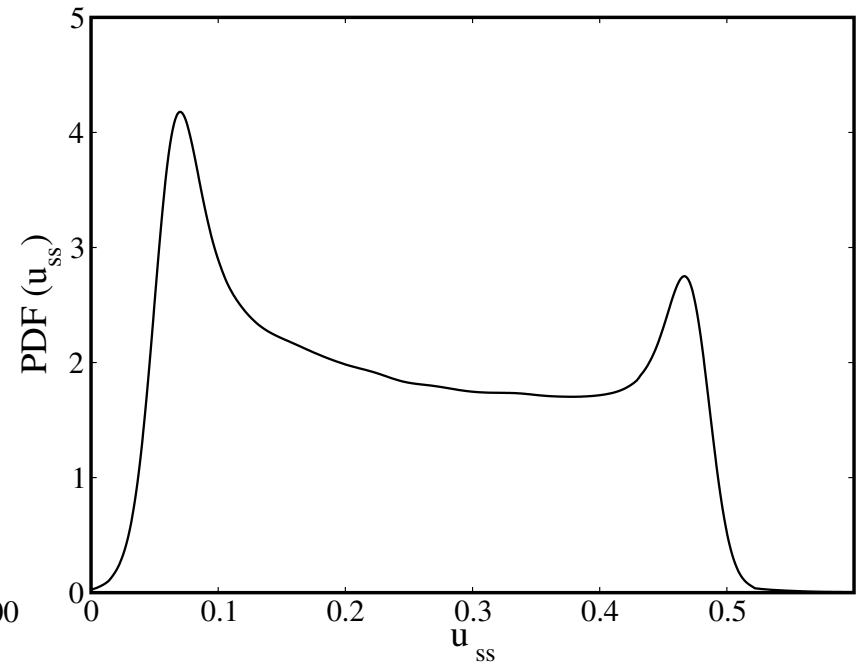


PC vs. Sampling Limit Cycle Orbit in Phase Space, *v* vs *u*



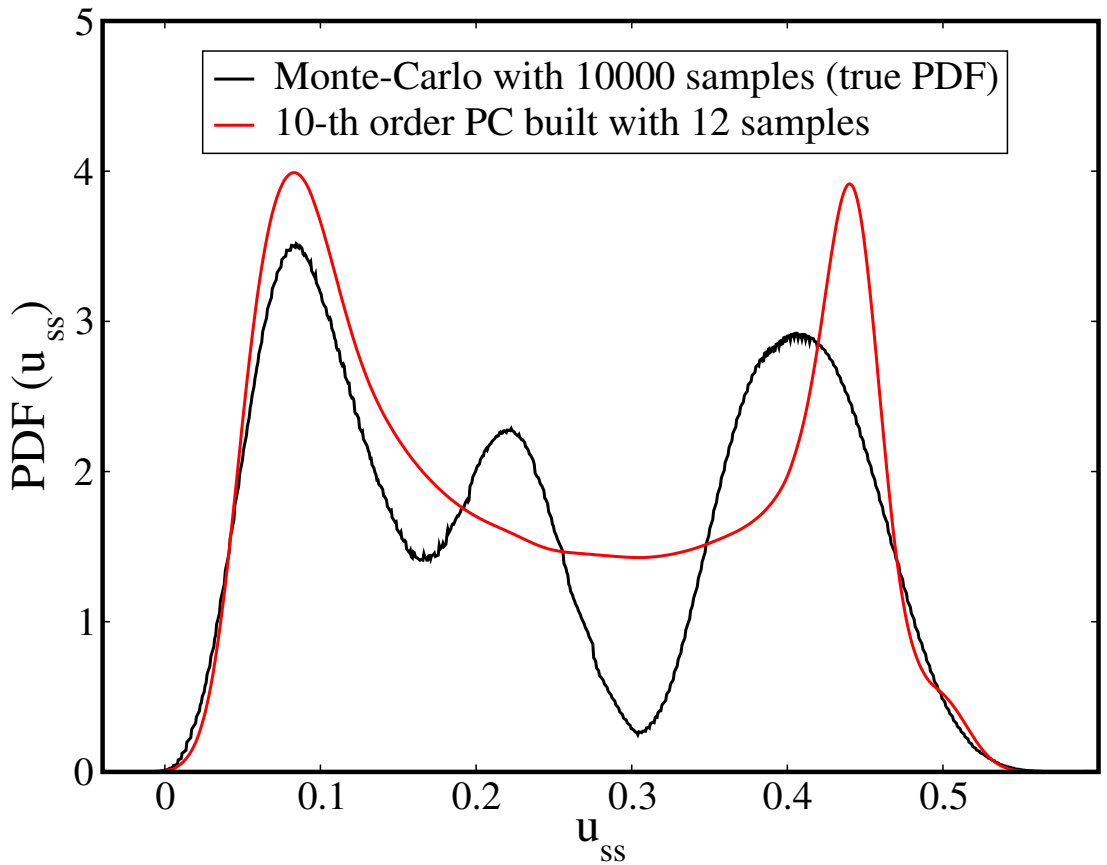
Surface Reaction Model: Uncertainty in Time Average of u





- Variation of b leads to different qualitative behaviors
- Output observable: average over time of u at steady state: u_{ss}
  - Representative, e.g. of expected coverage in catalytic system
- Varied the parameter b by 10% around its nominal value
- Output PDF is generated by $100K$ samples of a 9-th order PC

Global PC expansions fail to capture multimodality



- In principle, PC-based uncertainty propagation requires much fewer function evaluation.
- However, the accuracy of the PC expansion needs to be properly estimated.
- E.g., multimodal variables are not well-captured, even with a high PC order.

Domain Decomposition Approaches to Handle Nonlinearities

- A PCE is essentially a polynomial approximation of a random variable as a function in stochastic space
- Global PCEs fail to represent very non-linear functions over large domains
  - E.g. Gibbs oscillations around discontinuities
  - As order increases, more oscillations
- Piecewise representations can alleviate this
  - Lower order approximations over subsections of stochastic domain
  - Continuity of representation across subdomain boundaries is not required
    - Boundary is area of zero measure
    - Allows easy representation of discontinuities
- Application: intrusive UQ in thermal ignition model

UQ in constant-pressure ignition

M reactions in N species, with mass fractions Y_i:

$$\begin{aligned} \frac{\mathrm{d}Y_i}{\mathrm{d}t} &= \frac{w_i}{\rho}, \quad i = 1, \cdots, N \\ \frac{\mathrm{d}T}{\mathrm{d}t} &= \frac{w_T}{\rho c_p} \end{aligned}$$

with $w_T = -\sum_{i=1}^{N} h_i w_i$ and $w_i = \sum_{k=1}^{M} \nu_{ik} \mathcal{R}_k$

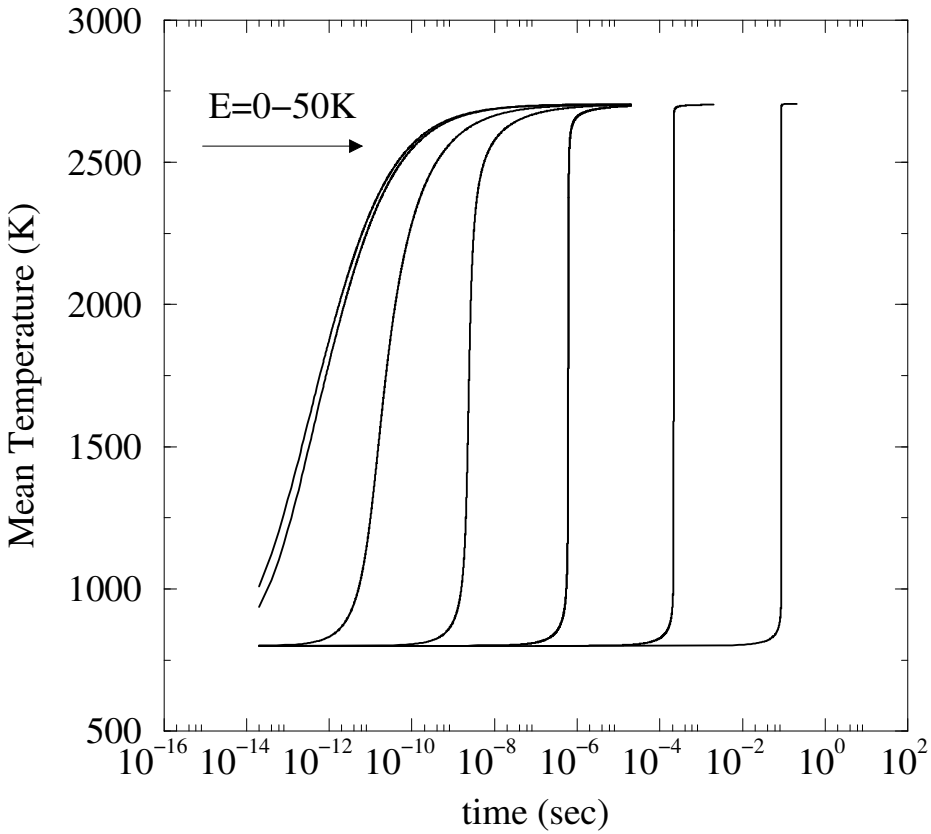
Example: $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$
Stoichiometric coefficients: $\nu_{ik} = \{-1, -2, 1, 2\}$
Reaction rate of progress: $\mathcal{R}_k = [CH_4][O_2]^2 \, A_k \, T^{n_k} \, e^{-E_k/T}$

- Quantify reaction-rate pre-exponential (A_k) uncertainty with multiplicative factor F_k :

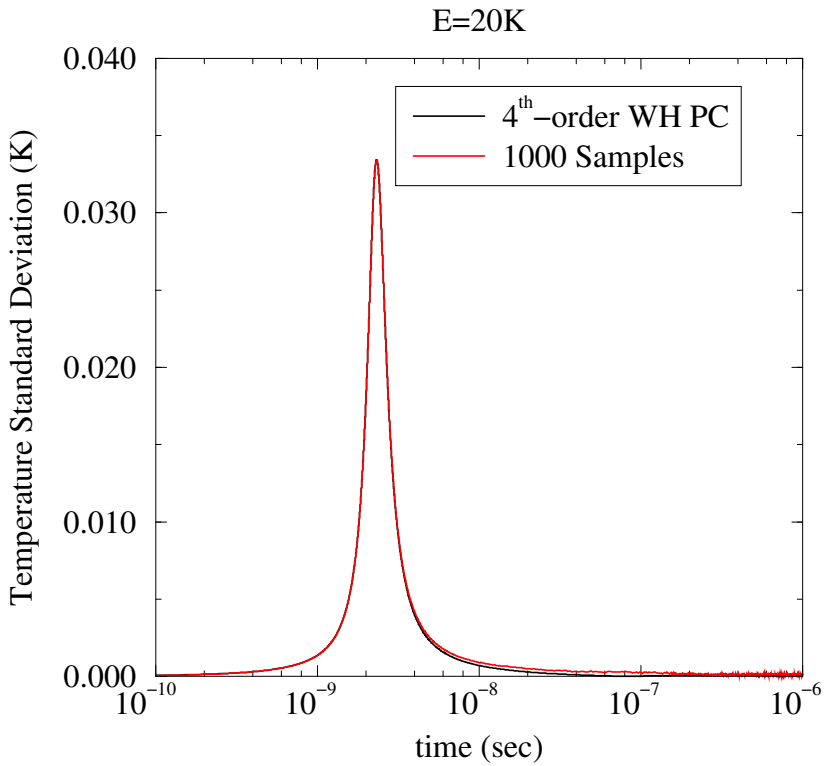
$$P\left(\frac{A_k}{F_k} < A_k < F_k A_k\right) = 0.95$$

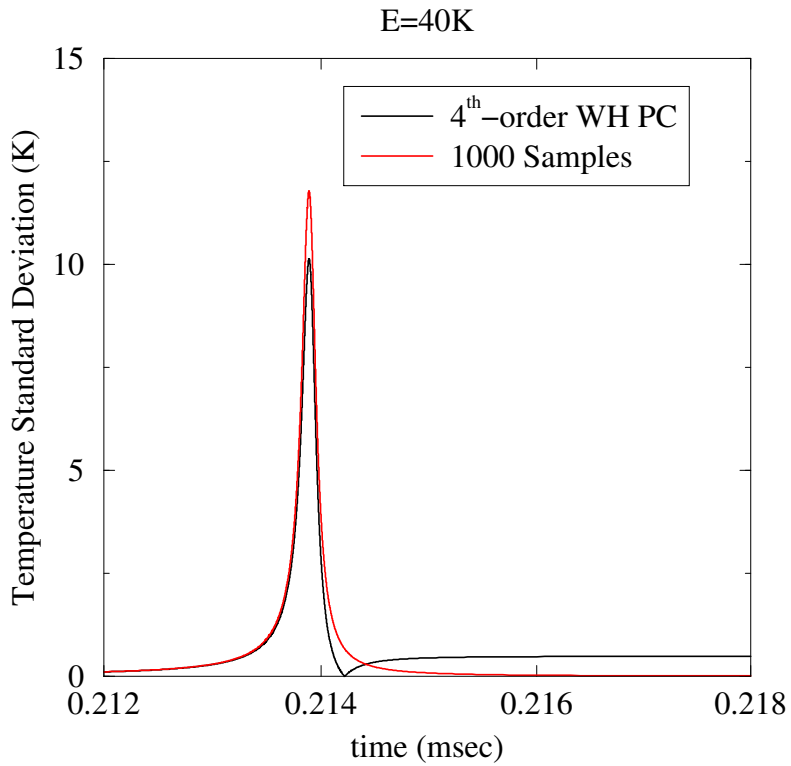
Large activation energy (E_k) exponentials lead to *very* fast changes in species concentrations and temperature

- Methane-air ignition — Global single-step irreversible mechanism
- Initial $T = 800K$
- Stoichiometric
- $p = 1$ atm (constant)



Increased E_k leads to higher peak dT/dt and higher consequence of small uncertainties in reaction rate constants

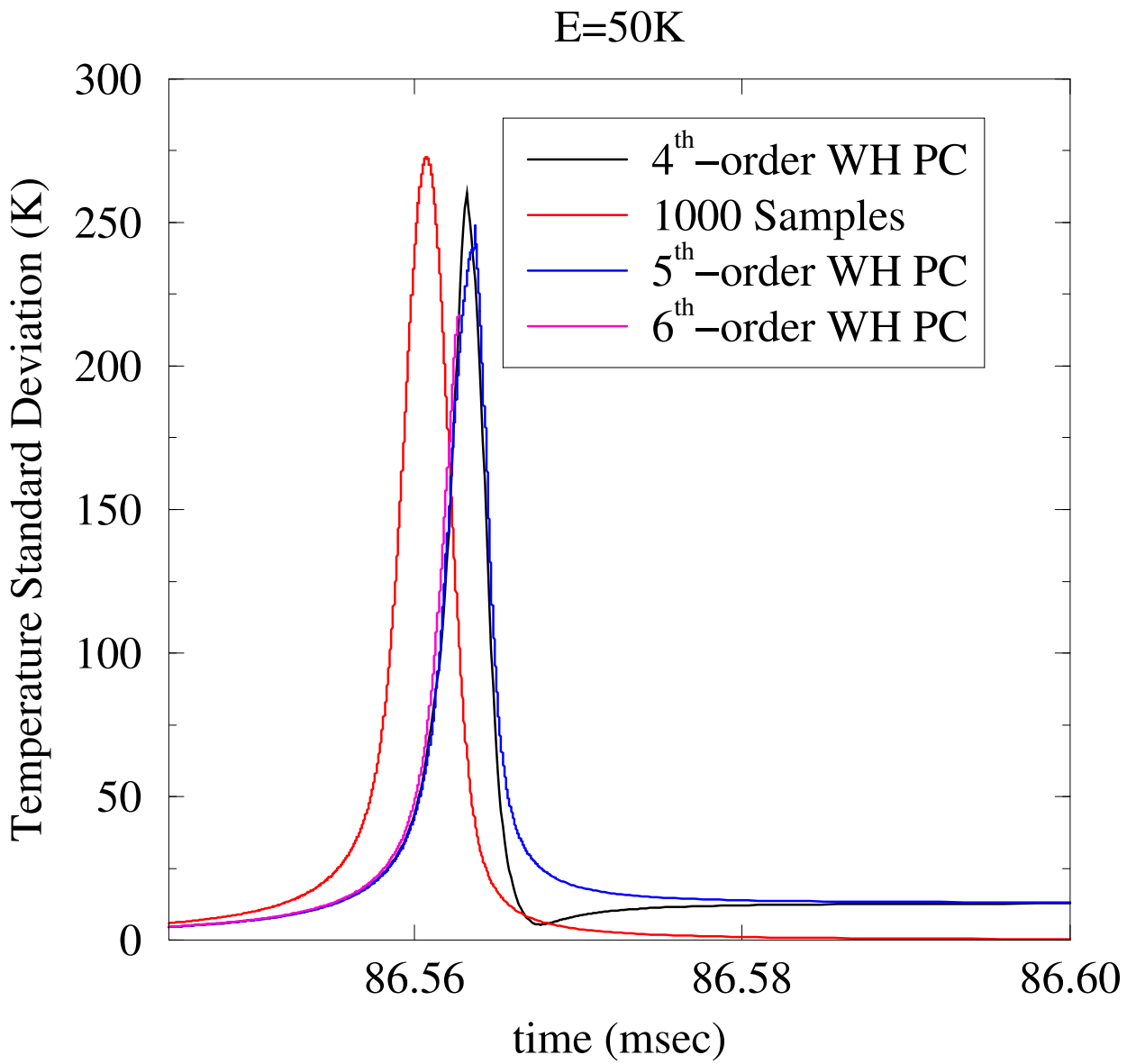




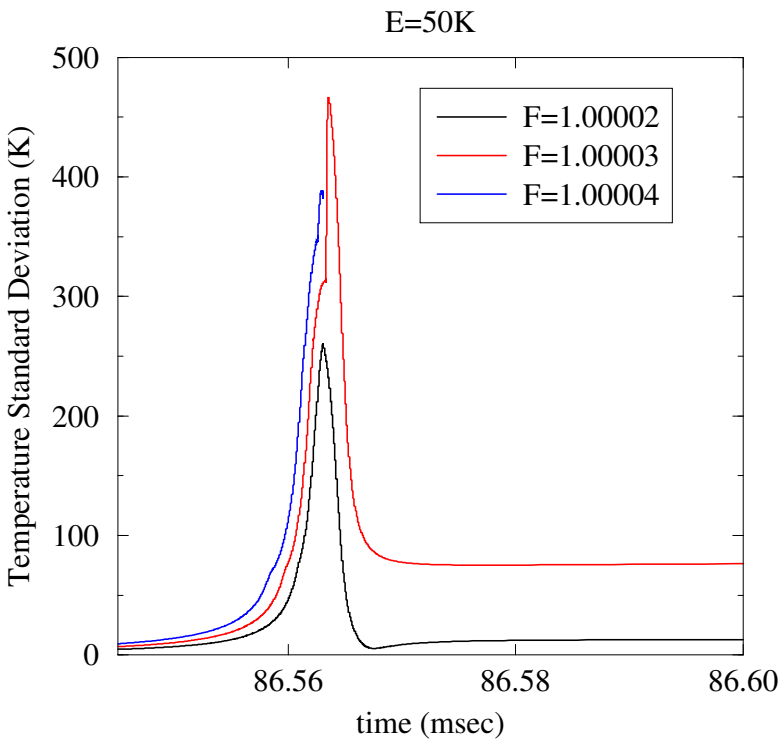
- $A_k = A_k(\xi)$, 1-D Wiener-Hermite
- PC UQ captures sampled stochastic behavior at low E_k
  - with miniscule uncertainty in A_k ($F = 1.00002$, COV=10^{-5}).
- Unphysical effects observed at high activation energy

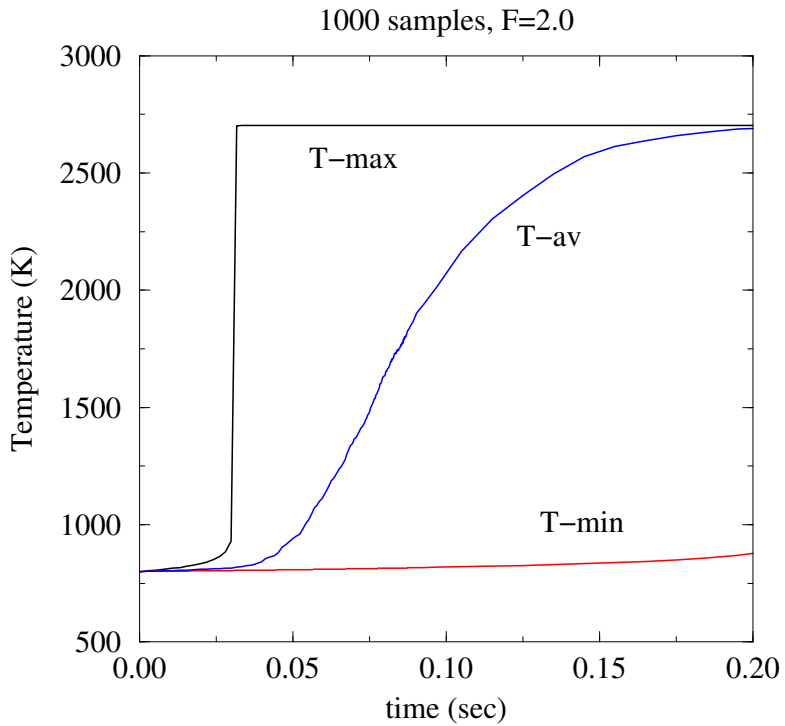
Errors increase for realistic E_k

- $E_k = 50$ K
- $F_k = 1.00002$
- Max T-stdv $\approx$ 300K
- Non-zero uncertainty in T for $t \to \infty$
- Increased PC order does not resolve the problem



Increasing A_k COV towards minimally-practical levels leads to failed time integration





- Unrealistic to expect a WH PC expansion in 1D to capture expected PDFs at realistic reaction-rate parametric uncertainties
- Need increased dimensionality of the PCEs, using multiple ξ's for each uncertain parameter, for increased accuracy and stability

Uncertainty Quantification with Multiwavelets

- An uncertain field quantity $u(\mathbf{x}, t, \theta)$ is expressed using PC

$$u = \sum_{k=0}^{P} u_k(\mathbf{x}, t)\Psi_k(\xi_1, \ldots, \xi_N)$$

- Introduce $\zeta_i = p(\xi_i)$: CDF of ξ_i, where ζ_i is on [0,1]

$$u = g(\xi_1, \ldots, \xi_N) = f(\zeta_1, \ldots, \zeta_N)$$

- Represent $f(\zeta)$ using N-D multiwavelets (Alpert, 1993)

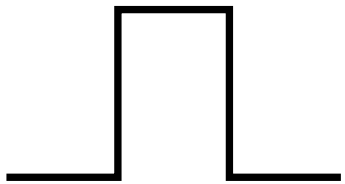
$$u = \sum_{\lambda=0}^{Q} \tilde{u}_\lambda(\mathbf{x}, t)\, \mathcal{W}_\lambda(\zeta_1, \ldots, \zeta_N)$$

Le Maître, Ghanem, Knio, and Najm, *J. Comp. Phys.*, 197:28-57 (2004)
Le Maître, Najm, Ghanem, and Knio, *J. Comp. Phys.*, 197:502-531 (2004)

Haar-Wavelets

Haar scaling functions

$$\phi^w(y) = \begin{cases} 1, & 0 \le y < 1 \\ 0 & \text{otherwise} \end{cases}$$

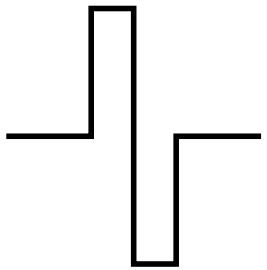


Scaled Haar functions, scaling factor j, and sliding factor k:

$$\phi^w_{jk}(y) = 2^{j/2}\phi^w(2^j y - k)$$

Haar function (mother wavelet)

$$\psi^w(y) \equiv \frac{1}{\sqrt{2}}\phi^w_{1,0}(y) - \frac{1}{\sqrt{2}}\phi^w_{1,1}(y) = \begin{cases} 1, & 0 \le y < \frac{1}{2}, \\ -1, & \frac{1}{2} \le y < 1, \\ 0, & \text{otherwise.} \end{cases}$$



Wavelet family

$$\psi^w_{j,k}(y) = 2^{j/2}\psi^w(2^j y - k), \quad j = 0, 1, \ldots \text{ and } k = 0, \ldots, 2^j - 1$$

Wiener-Haar Construction

The set of $\psi^w_{j,k}(y)$ is an orthonormal system

Any function $f \in L^2([0,1])$ can be arbitrarily well approximated by the sum of its mean and a finite linear combination of the $\psi^w_{j,k}(y)$.

The wavelet set $W_{j,k}(\xi(\theta)) \equiv \psi^w_{j,k}(p(\xi))$ forms a basis for the space of L^2 random processes.

$$X(\xi(\theta)) = X_o + \sum_{j=0}^{\infty} \sum_{k=0}^{2^j-1} X^w_{j,k} \psi^w_{j,k}(p(\xi)) = \sum_{\lambda} X_\lambda W_\lambda(\xi(\theta))$$

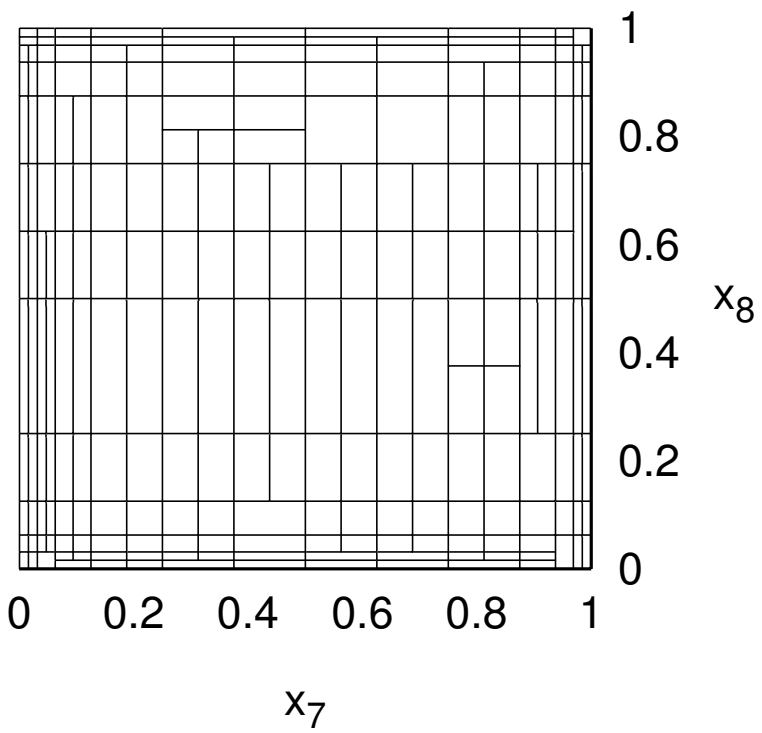
Multidimensional $\{\xi_1, \xi_2, \ldots, \xi_N\}$,

$$\mathcal{W} = \prod_{k=1}^{N} W_{\lambda_k}(\xi_k)$$

Multidimensional Multiwavelet Construction

- Wiener-Haar PC able to represent uncertainty in systems exhibiting bifurcations depending on parameter values
- Poor convergence relative to PC constructions with smooth global bases on smooth functions
- Use multiwavelet construction (Alpert, 1993) employing higher order polynomials instead of the Haar-functions
- For efficient multidimensional construction, use
  - Block-decomposition of the stochastic space
  - A local MW construction on each block employing
    - Scaled Legendre polynomials on $[0, 1]$
    - First level Multiwavelet details
  - Adaptive resolution in each dimension on each block

Block decomposition in a 2D parametric space

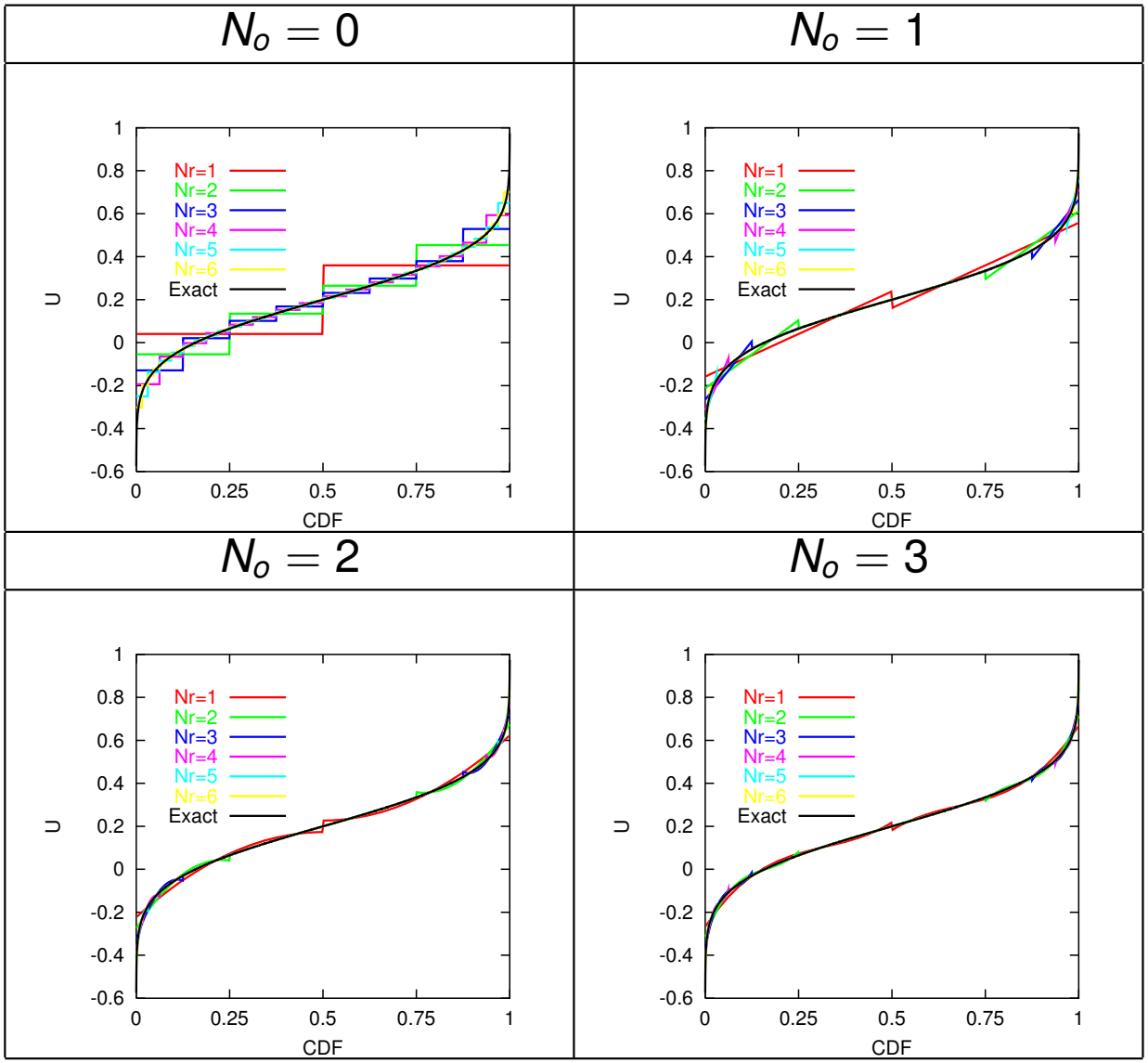


Computational Advantages with MRA UQ

- Adaptive partitioning allows optimal combinations of resolution and order
  - over space-time
- UQ problem can be done separately on each sub-domain
  - This can be done
    - intrusively or
    - non-intrusively

    on each sub-domain
  - Presents advantages in a massively parallel context

Multi-Wavelet Representation of Model Problem Gaussian IC



- IC $U_0 = 0.2$, $U_1 = 0.1$
- N_o: MW order
- N_r: MRA resolution
- Local representation using low-order polynomial Multi-Wavelets
- Increased N_o leads to faster convergence to exact solution with increasing N_r

Adaptive Partition of the Random Parameter Space

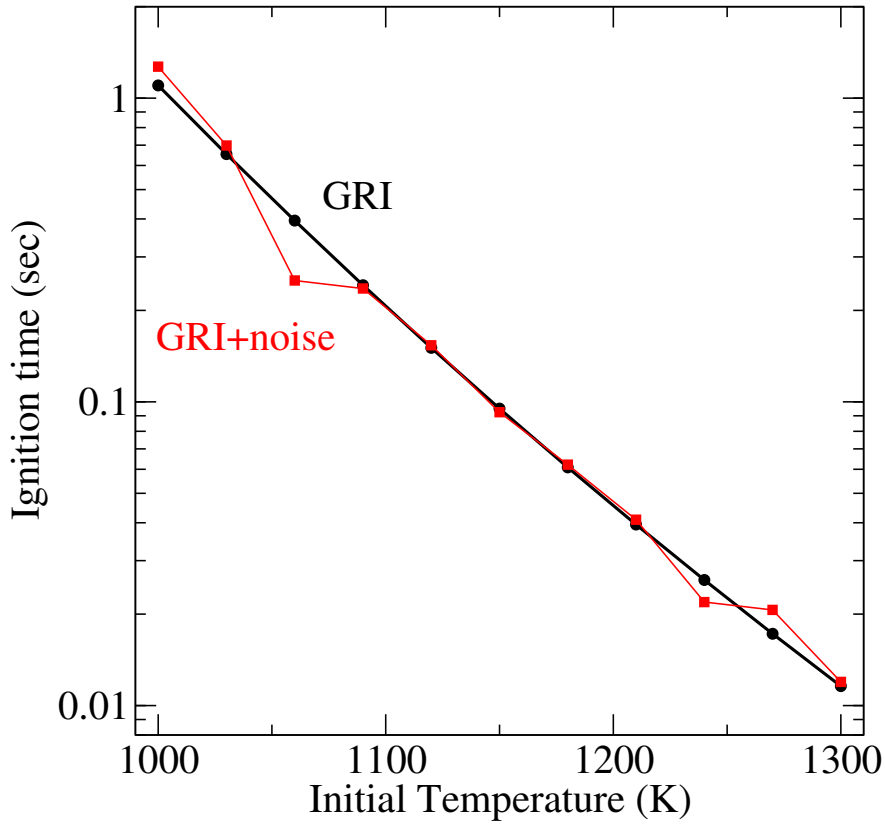
- For multiple stochastic dimensions, the computational cost of the MW spectral products can become prohibitive
- Use adaptive partitioning of the space of random parameters
- On each sub-domain define a local scaled random basis
- Construct a local MW expansion with up-to only 1^{st}-level details
  - Local Wiener-Legendre projection of order N_o, plus 1D details
- Combine local block-statistics to arrive at global statistics
- Adaptively refine block decomposition in each stoch dimension
  - Use 1D details variance to guide refinement in each dimension

Le Maître *et al. J. Comp. Phys.*, 197:502-531 (2004)

Generate ignition “data" using a detailed model+noise

- Ignition using a detailed chemical model for methane-air chemistry
- Ignition time versus Initial Temperature
- Multiplicative noise error model
- 11 data points:

$$\begin{aligned} d_i &= t_{\mathrm{ig},i}^{\mathrm{GRI}}(1+\sigma\epsilon_i) \\ \epsilon &\sim N(0,1) \end{aligned}$$

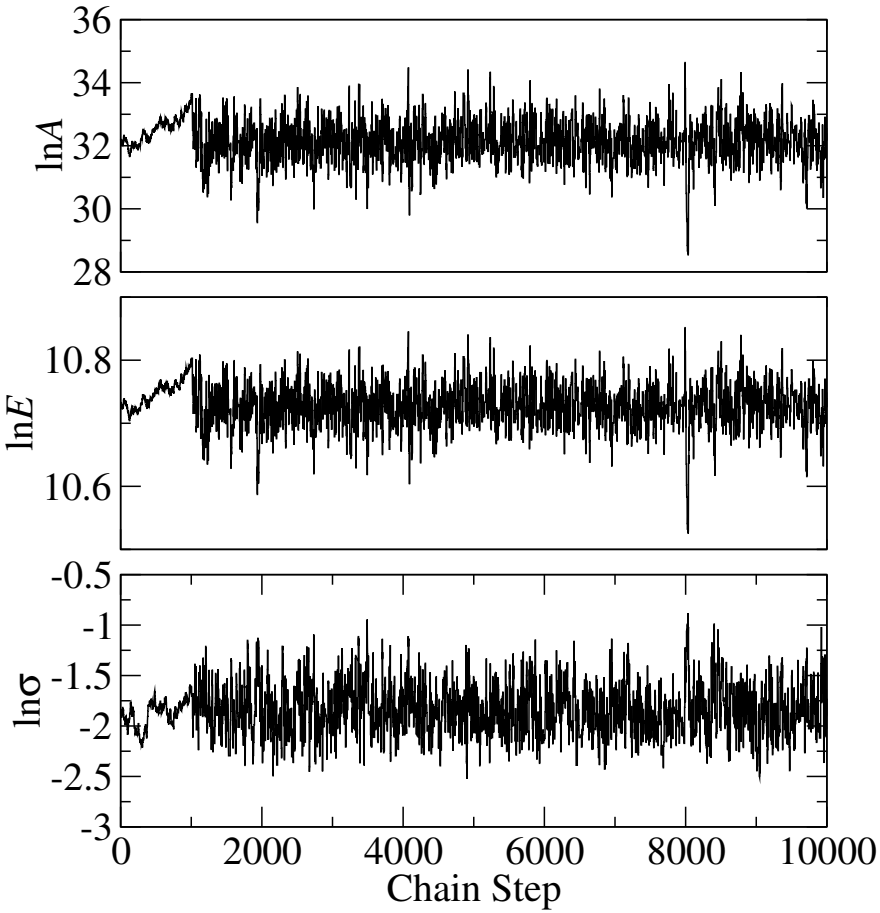


Fitting with a simple chemical model

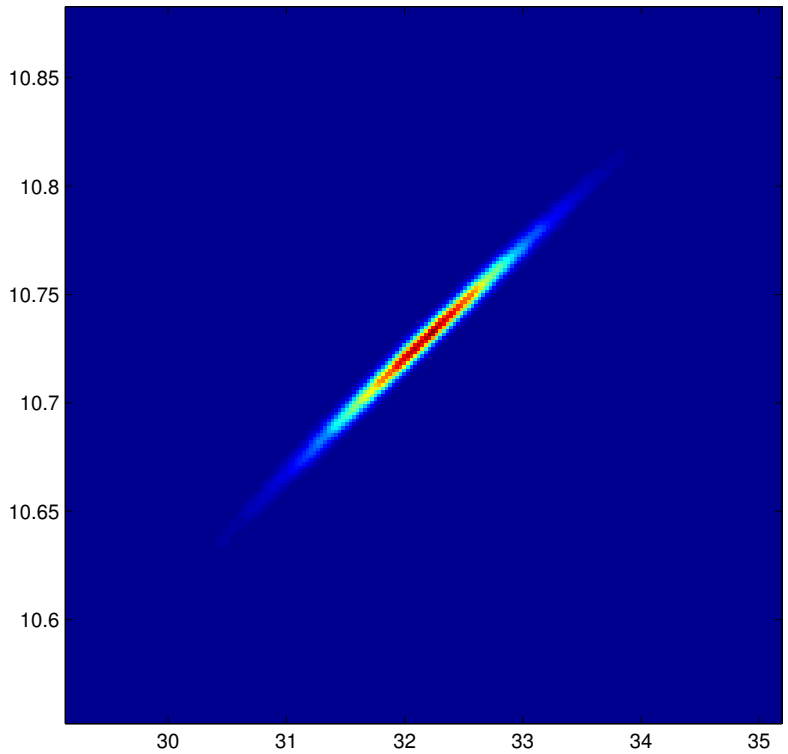
- Fit a global single-step irreversible chemical model

$$\begin{aligned} CH_4 + 2O_2 &\rightarrow CO_2 + 2H_2O \\ \Re &= [CH_4][O_2]k_f \\ k_f &= A\exp(-E/R^oT) \end{aligned}$$

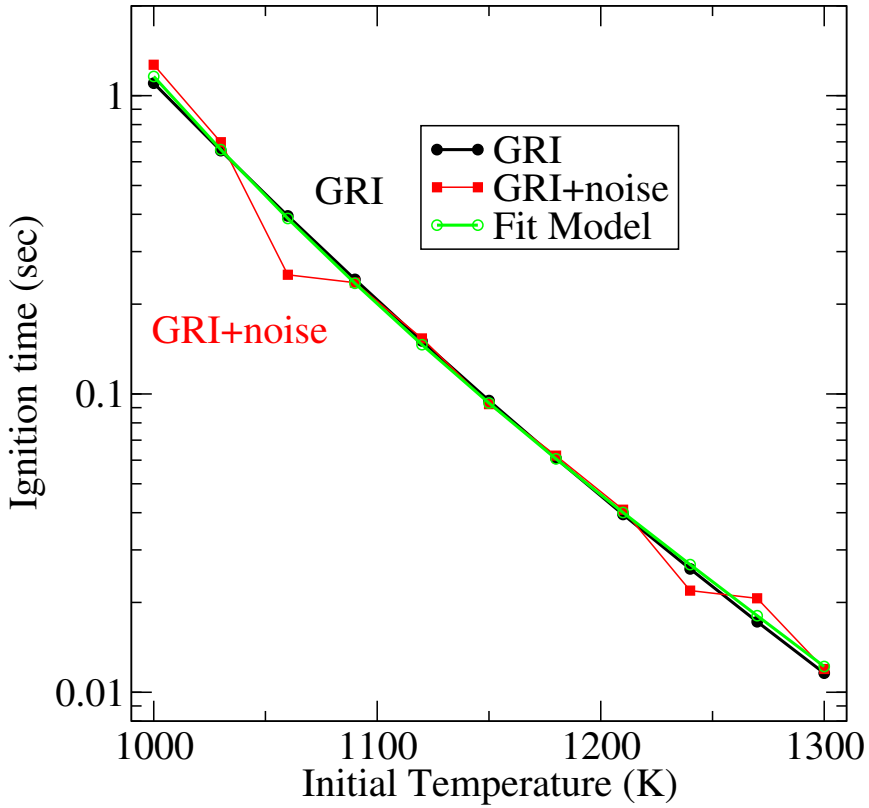
- Infer 3-D parameter vector $(\ln A, \ln E, \ln \sigma)$
- Good mixing with adaptive MCMC when start at MLE



Bayesian Inference Posterior and Nominal Prediction

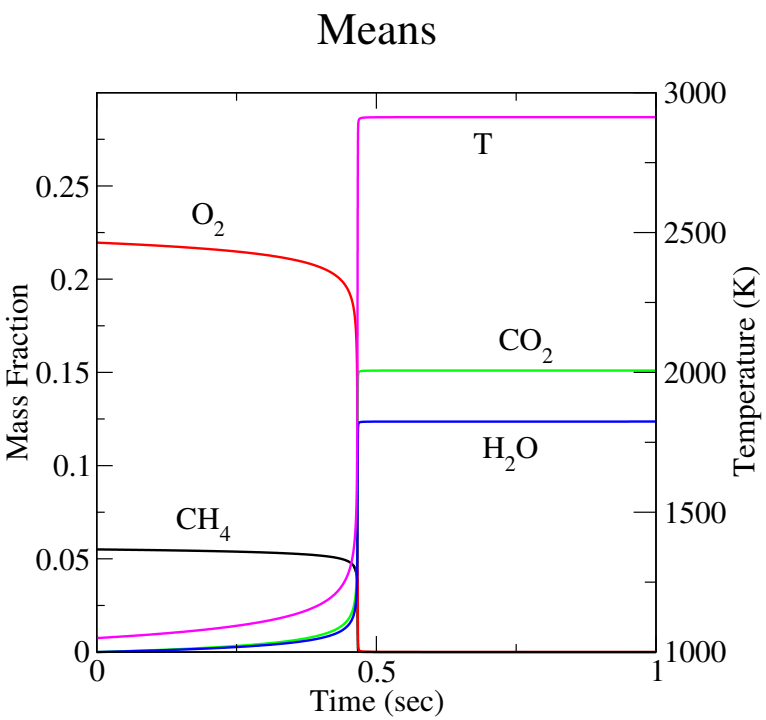


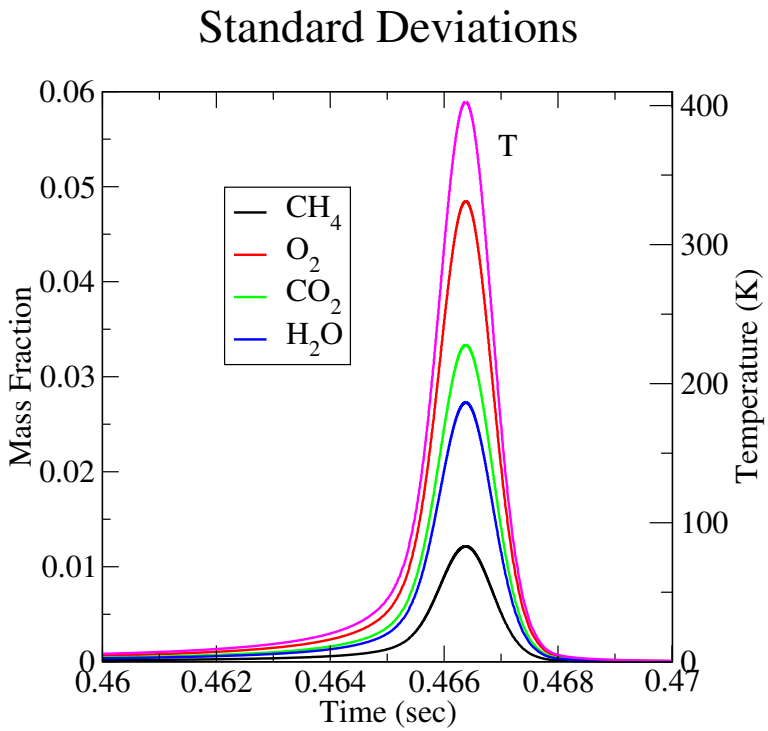
Marginal joint posterior on $(\ln A, \ln E)$ exhibits strong correlation



Nominal fit model is consistent with the true model

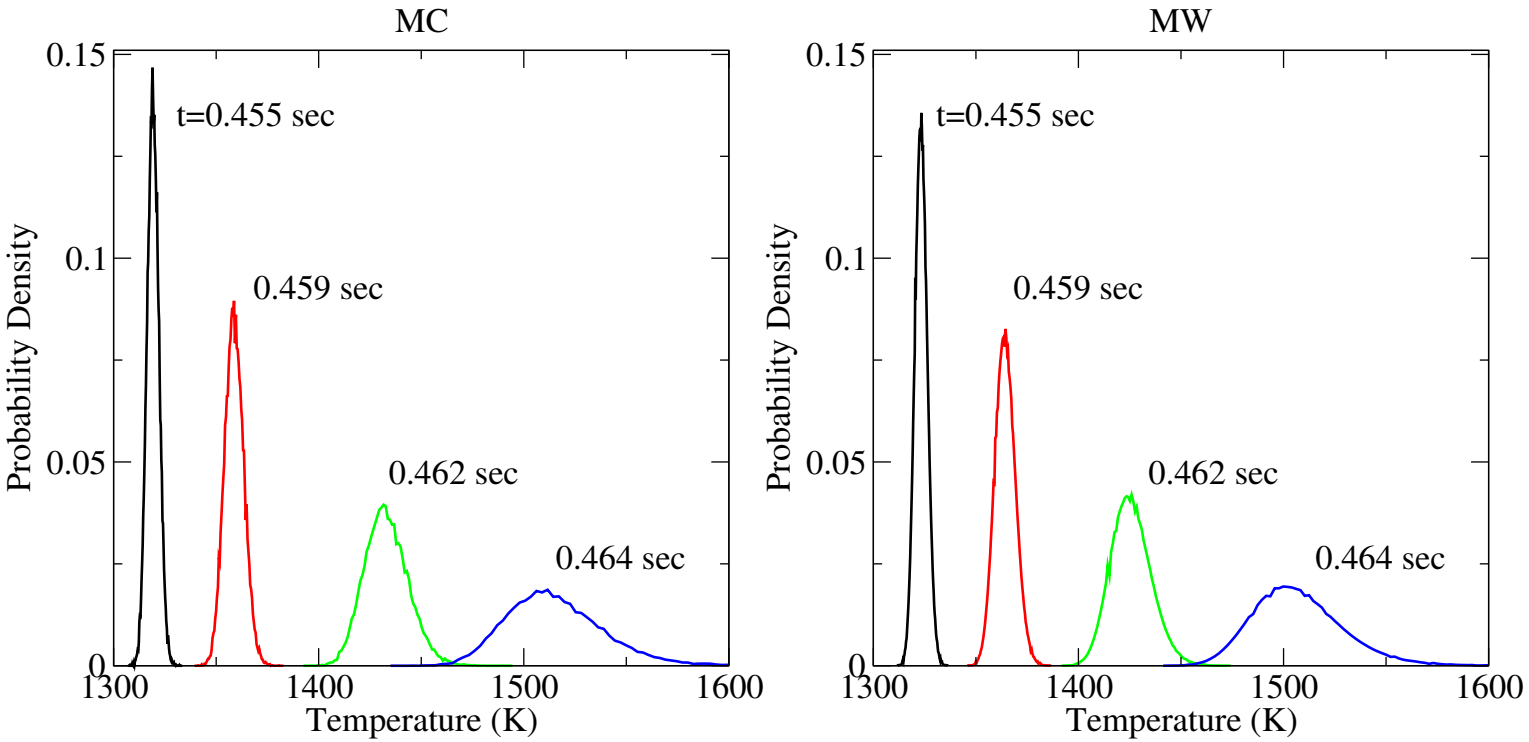
Correlation Slope χ and Chemical Ignition





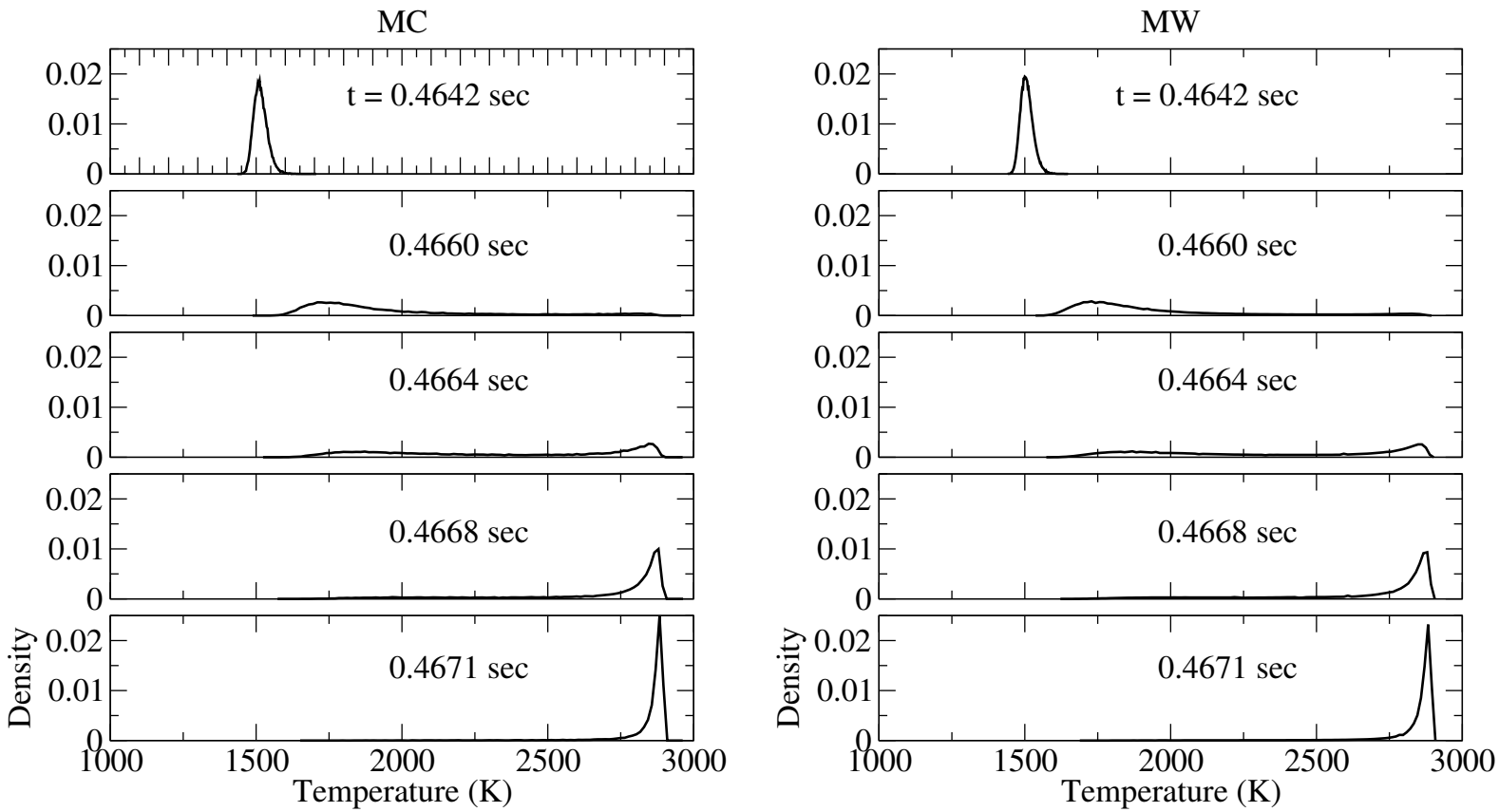
- 4^{th} Order Multiwavelet PC, Multiblock, Adaptive
- $\sigma_{T,\max} \sim 400$ K during ignition transient, $\chi \sim 0.03$

Time evolution of Temperature PDFs in preheat stage



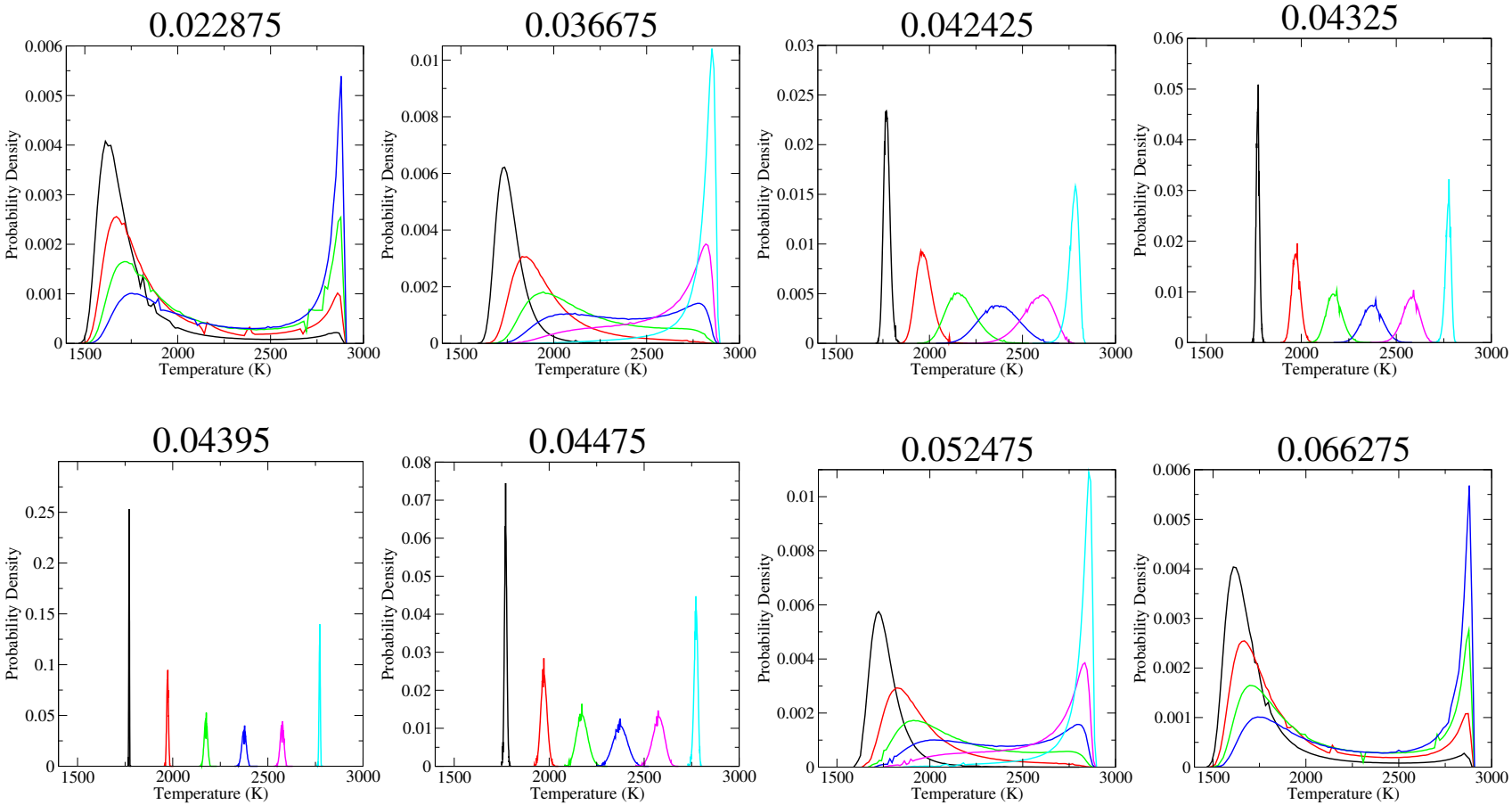
- Similar results from MC (20K samples) and MW PC
- Increased uncertainty, and long high–T PDF tails, in time

Evolution of Temp. PDF – Fast Ignition Transient



- Transition from unimodal to bimodal PDFs
- Leakage of probability mass from pre-heat PDF high–T tail

Time evolution of Temperature PDFs for different χ



- Bimodal solution PDFs for high uncertainty growth
- Unimodal for low uncertainty growth, with $\chi \approx 0.044$

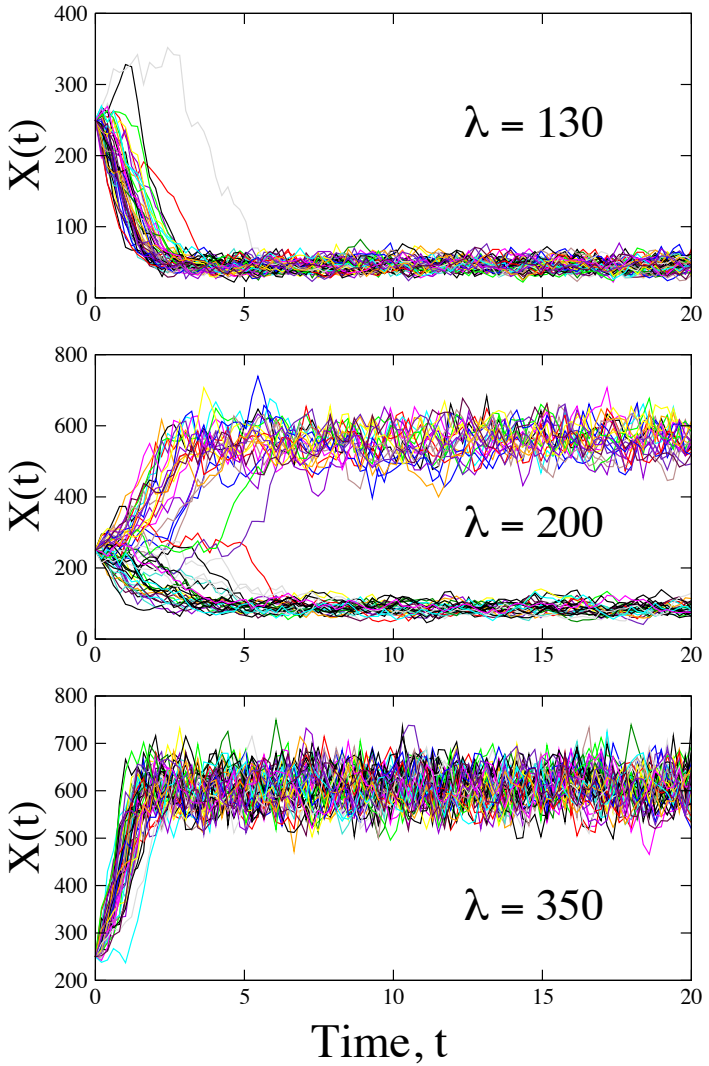
Data Decomposition Approaches to Handle Multi-Modalities

- Alternative approach when faced with samples of multi-modal random variables
  - Separate data into multiple sets that are easier to represent
  - Represent each with global PCE
  - Overall results is a PC mixture model
    - Generalization of Gaussian mixture models
- Application: Karhunen-Loève applied to output state of Schlögl model

Schlögl Model is a prototype bistable model

- Reactions
$$A + 2X \underset{a_2}{\overset{a_1}{\rightleftarrows}} 3X$$
$$B \underset{a_4}{\overset{a_3}{\rightleftarrows}} X \quad .$$
- Propensities
$a_1 = k_1 A X(X-1)/2,$
$a_2 = k_2 X(X-1)(X-2)/6,$
$a_3 = k_3 B,$
$a_4 = k_4 X.$
- Nominal parameters

$k_1 A$	0.03
k_2	0.0001
$k_3 B = \lambda$	200
k_4	3.5
A	10^5
B	$2 \cdot 10^5$
$X(0)$	250



Polynomial Chaos expansion represents any random variable as a polynomial of a standard random variable

- Truncated PCE: finite dimension n and order p

$$X(\theta) \simeq \sum_{k=0}^{P} c_k \Psi_k(\eta)$$

  with the number of terms $P + 1 = \frac{(n+p)!}{n!p!}$.

- $\eta = (\eta_1, \cdots, \eta_n)$ standard i.i.d. r.v.
  Ψ_k standard orthogonal polynomials
  c_k spectral modes.

- Most common standard Polynomial-Variable pairs:
  (continuous) Gauss-Hermite, Legendre-Uniform,
  (discrete) Poisson-Charlier.

[Wiener, 1938; Ghanem & Spanos, 1991; Xiu & Karniadakis, 2002]

Galerkin Projection is typically needed

PC expansion: $X(\boldsymbol{\theta}) \simeq \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\eta}) = g_{\mathcal{D}}(\boldsymbol{\eta})$

Orthogonal projection: $c_k = \frac{\langle X(\boldsymbol{\theta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$

- Intrusive Spectral Projection (ISP)
  - ⋆ Direct projection of governing equations
  - ⋆ Leads to deterministic equations for PC coefficients
  - ∗ No explicit governing equation for SRNs
- Non-intrusive Spectral Projection (NISP)
  - ⋆ Sampling based
  - ⋆ No explicit evolution equation for X needed
  - ∗ Galerkin projection not well-defined for SRNs

Karhunen-Loève decomposition reduces stochastic process to a finite number of random variables

- KL decomposition:

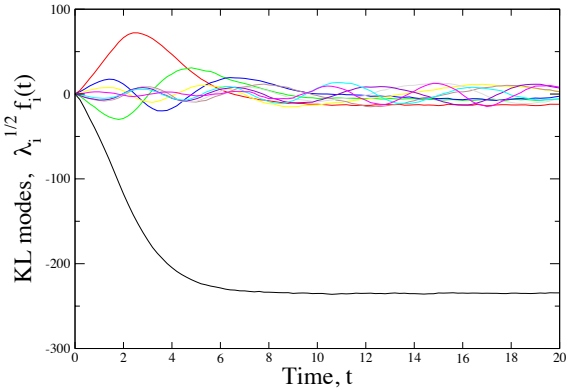
$$X(t,\theta) = \bar{X}(t) + \sum_{n=1}^{\infty} \xi_n(\theta)\sqrt{\lambda_n} f_n(t)$$

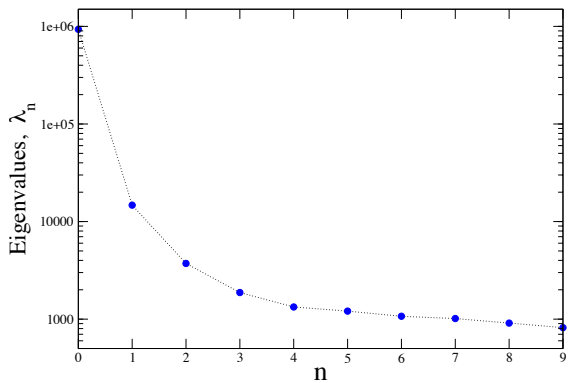
- Uncorrelated, zero-mean KL variables:

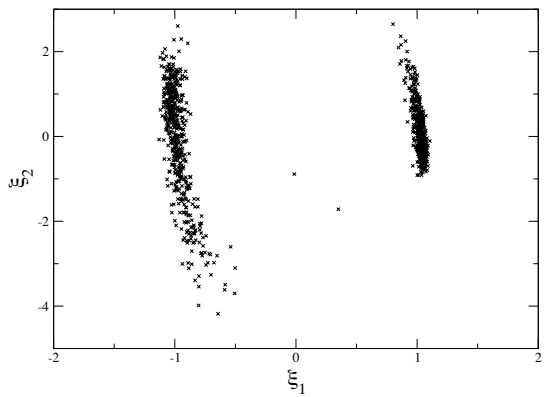
$$\langle \xi_n \rangle = 0, \qquad \langle \xi_n \xi_m \rangle = \delta_{nm}$$

- SSA(continuum) $\longleftrightarrow$ KL(discrete)

$$X(t) \longleftrightarrow \boldsymbol{\xi} = (\xi_1, \xi_2, \dots)$$

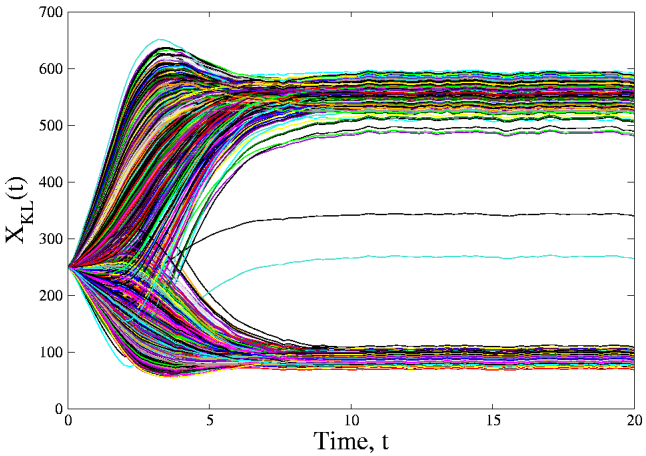




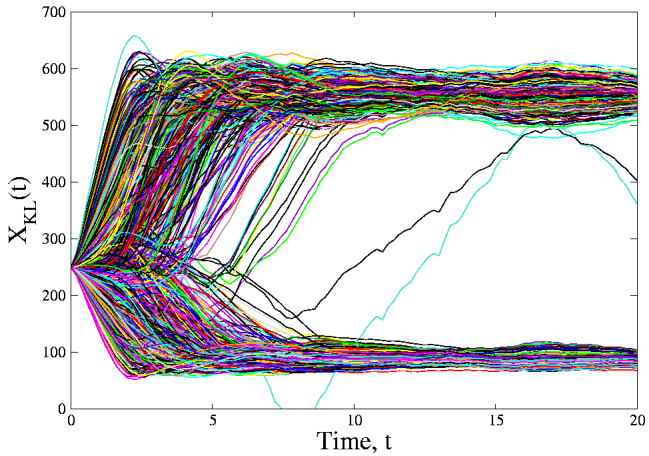


K-L decomposition captures each realization

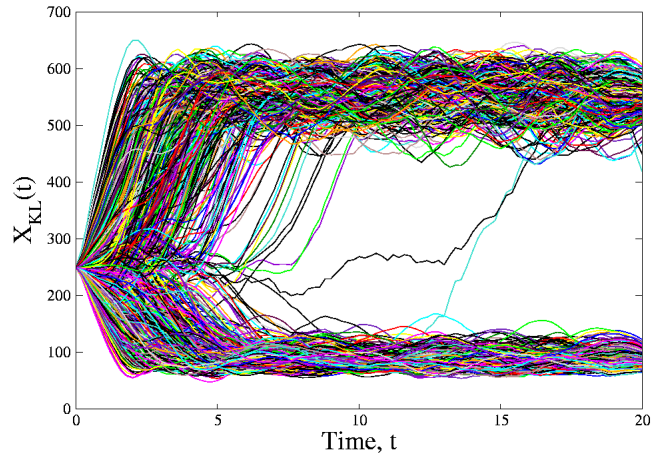
KL decomposition with 2 modes



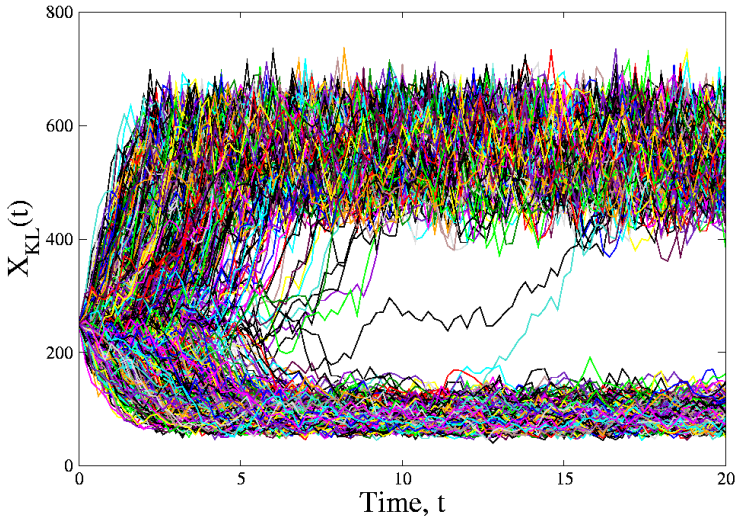
KL decomposition with 5 modes



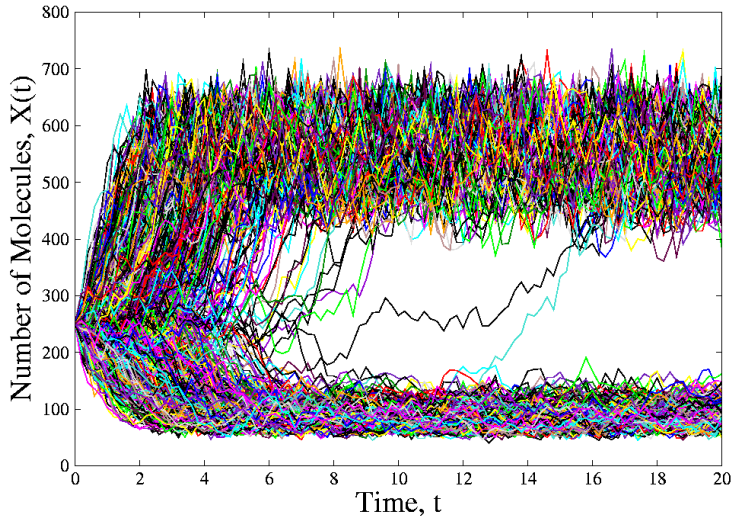
KL decomposition with 10 modes



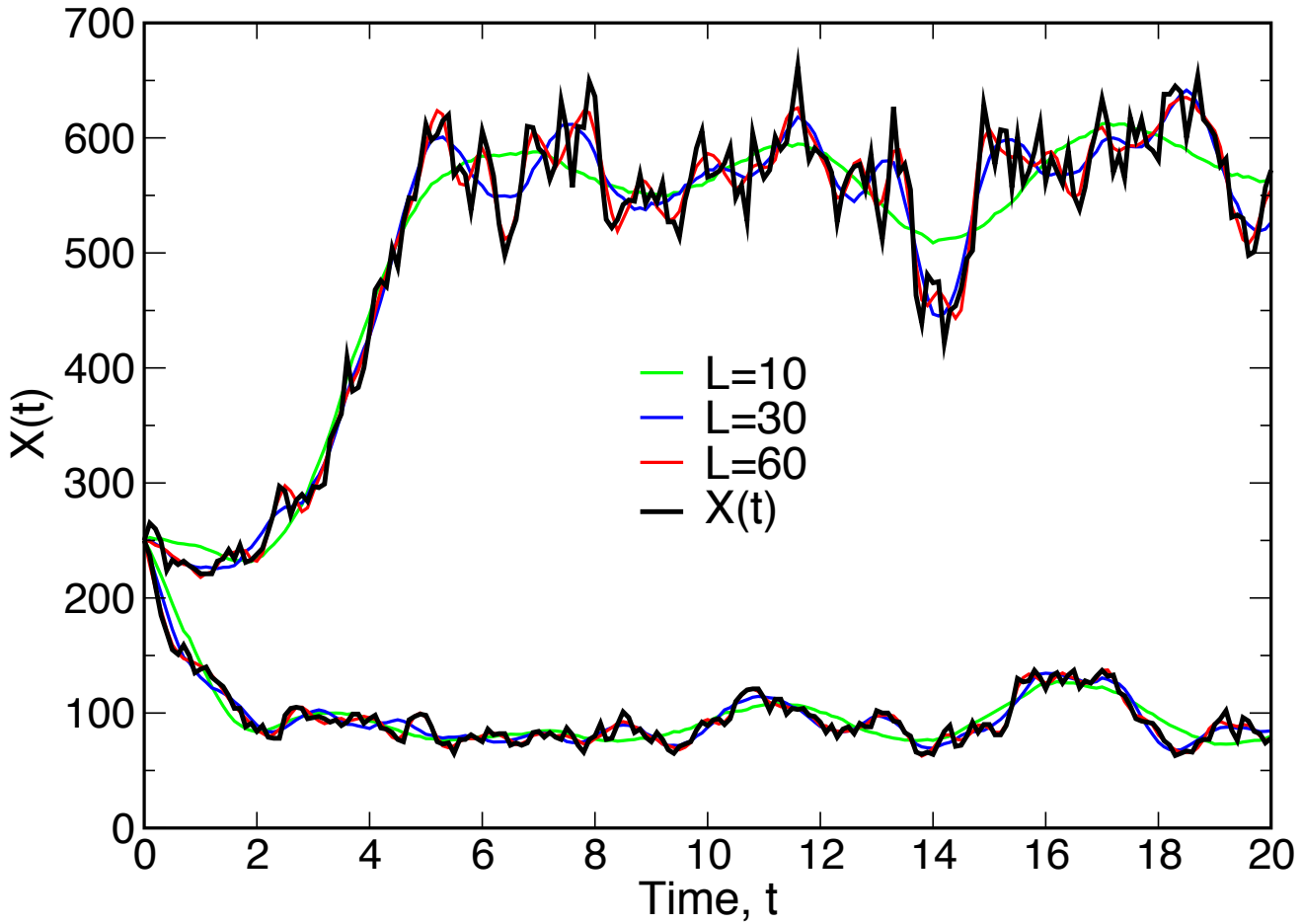
KL decomposition with 100 modes



SSA Realizations



K-L decomposition captures each realization



PC expansion of a random vector

$$\xi = \sum_{k=0}^{P} c_k \Psi_k(\eta)$$

Galerkin projection

$$c_k = \frac{\langle \xi \Psi_k(\eta) \rangle}{\langle \Psi_k^2(\eta) \rangle}$$

is not well-defined,
since ξ and η do not belong to the same stochastic space.

PC expansion of a random vector

$$\xi = \sum_{k=0}^{P} c_k \Psi_k(\eta)$$

Galerkin projection

$$c_k = \frac{\langle \xi \Psi_k(\eta) \rangle}{\langle \Psi_k^2(\eta) \rangle}$$

is not well-defined,
since ξ and η do not belong to the same stochastic space.

Need a map $\xi \leftrightarrow \eta$.

Rosenblatt transformation

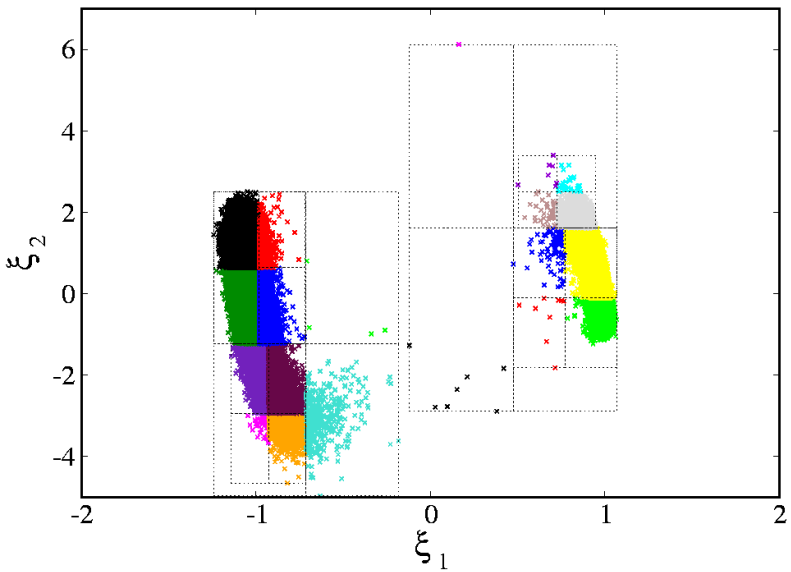
- Rosenblatt transformation maps any (not necessarily independent) set of random variables $(\xi_1, \ldots, \xi_n)$ to uniform i.i.d.'s $\{\eta_i\}_{i=1}^n$ (Rosenblatt, 1952).

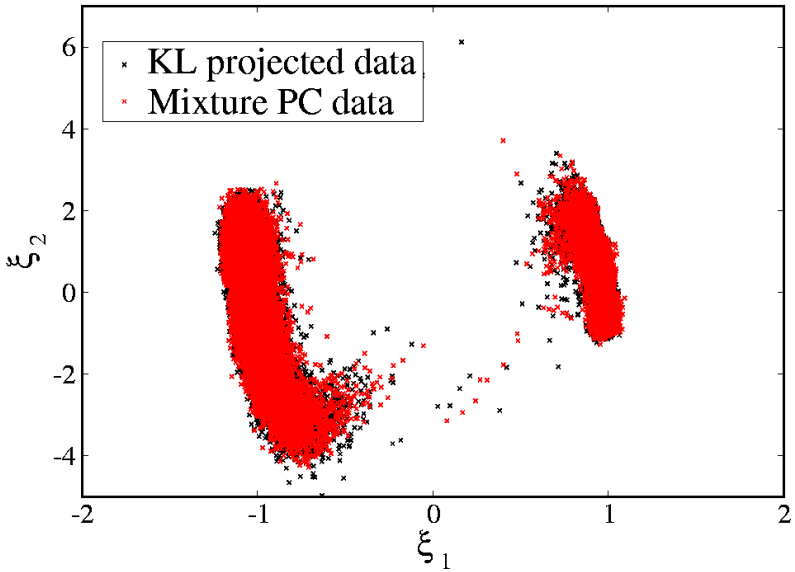
$$\begin{aligned}
\eta_1 &= F_1(\xi_1) \\
\eta_2 &= F_{2|1}(\xi_2|\xi_1) \\
\eta_3 &= F_{3|2,1}(\xi_3|\xi_2, \xi_1) \\
&\vdots \\
\eta_n &= F_{n|n-1,\ldots,1}(\xi_n|\xi_{n-1}, \ldots, \xi_1)
\end{aligned}$$

- Inverse Rosenblatt transformation $\boldsymbol{\xi} = R^{-1}(\boldsymbol{\eta})$ ensures a well-defined quadrature integration

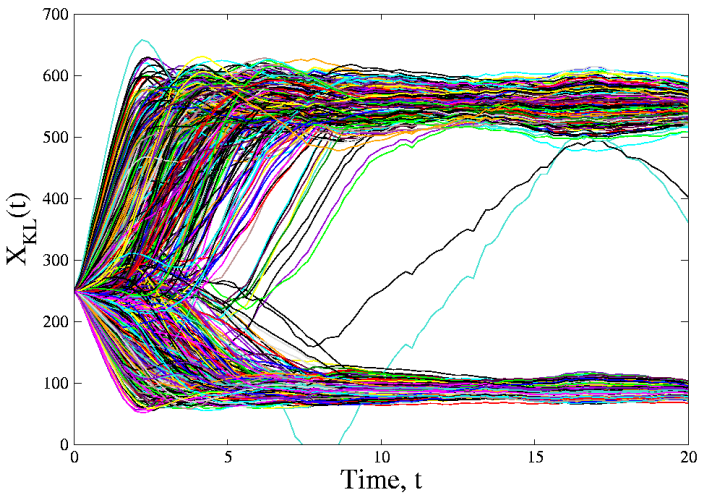
$$\langle \xi_i \Psi_k(\boldsymbol{\eta}) \rangle = \int R^{-1}(\boldsymbol{\eta})_i \Psi_k(\boldsymbol{\eta}) d\boldsymbol{\eta}$$

KL+PC+Data Partitioning represent the dynamics of a bimodal process

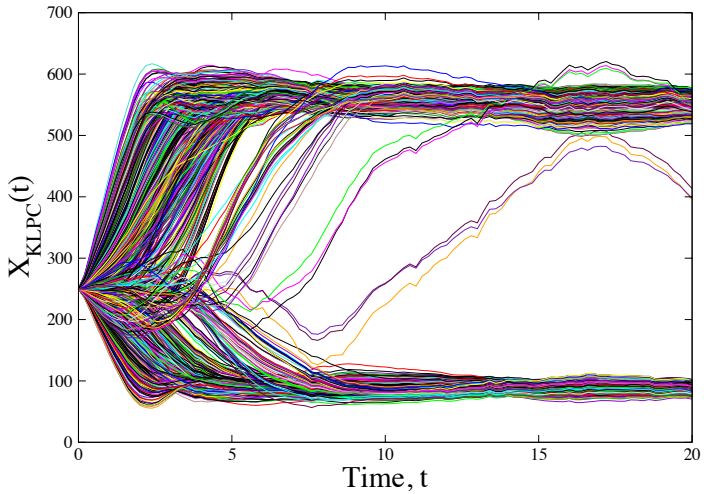




KL decomposition with 5 modes



KL-PC representation, 5 KL modes, 3rd PC order



UQ in Systems with Inherent Stochastic Noise

- Some systems have intrinsic uncertainty
  - Stochastic reaction networks
  - Macroscale quantities extracted from atomistic methods with sampling
- Galerkin projection is challenged by such systems
  - No deterministic governing equation for intrusive UQ
  - Quadrature methods, especially sparse methods, are challenged by noise in function evaluations
- Use Bayesian regression to infer PC coefficients

Summary

- Polynomial Chaos Expansions offer a convenient way to represent random variables
- Both intrusive and non-intrusive approaches are available to propagate uncertain model inputs to its outputs
- While conceptually straightforward, many challenges remain in terms of efficiency and accuracy

Further Reading

- N. Wiener, “Homogeneous Chaos", *American Journal of Mathematics*, 60:4, pp. 897-936, 1938.
- M. Rosenblatt, “Remarks on a Multivariate Transformation", *Ann. Math. Statist.*, 23:3, pp. 470-472, 1952.
- R. Ghanem and P. Spanos, “Stochastic Finite Elements: a Spectral Approach", Springer, 1991.
- O. Le Maître and O. Knio, “Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics", Springer, 2010.
- D. Xiu, “Numerical Methods for Stochastic Computations: A Spectral Method Approach", Princeton U. Press, 2010.
- O.G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, “On the convergence of generalized polynomial chaos expansions,” ESAIM: M2AN, 46:2, pp. 317-339, 2011.
- D. Xiu and G.E. Karniadakis, “The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations", *SIAM J. Sci. Comput.*, 24:2, 2002.
- B. Debusschere, H. Najm, P. Pébay, O. Knio, R. Ghanem and O. Le Maître, “Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Proecesses", *SIAM J. Sci. Comp.*, 26:2, 2004.
- S. Ji, Y. Xue and L. Carin, “Bayesian Compressive Sensing", *IEEE Trans. Signal Proc.*, 56:6, 2008.
- K. Sargsyan, B. Debusschere, H. Najm and O. Le Maître, “Spectral representation and reduced order modeling of the dynamics of stochastic reaction networks via adaptive data partitioning". *SIAM J. Sci. Comp.*, 31:6, 2010.
- K. Sargsyan, B. Debusschere, H. Najm and Y. Marzouk, “Bayesian inference of spectral expansions for predictability assessment in stochastic reaction networks". *J. Comp. Theor. Nanosc.*, 6:10, 2009.