Polynomial Chaos Based Uncertainty Quantification

SAND2015-0639PE

Lecture 3: Characterization of Random Variables and Processes

Bert Debusschere

bjdebus@sandia.gov
Sandia National Laboratories,
Livermore, CA

DATE – LOCATION

Acknowledgements

Cosmin Safta Khachik Sargsyan Kenny Chowdhary Habib Najm	Sandia National Laboratories Livermore, CA
Omar Knio Justin Winokur	Duke University, Raleigh, NC
Roger Ghanem	University of Southern California Los Angeles, CA
Olivier Le Maître	LIMSI-CNRS, Orsay, France
and many others ...	

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC) and Applied Mathematics Research (AMR) programs.

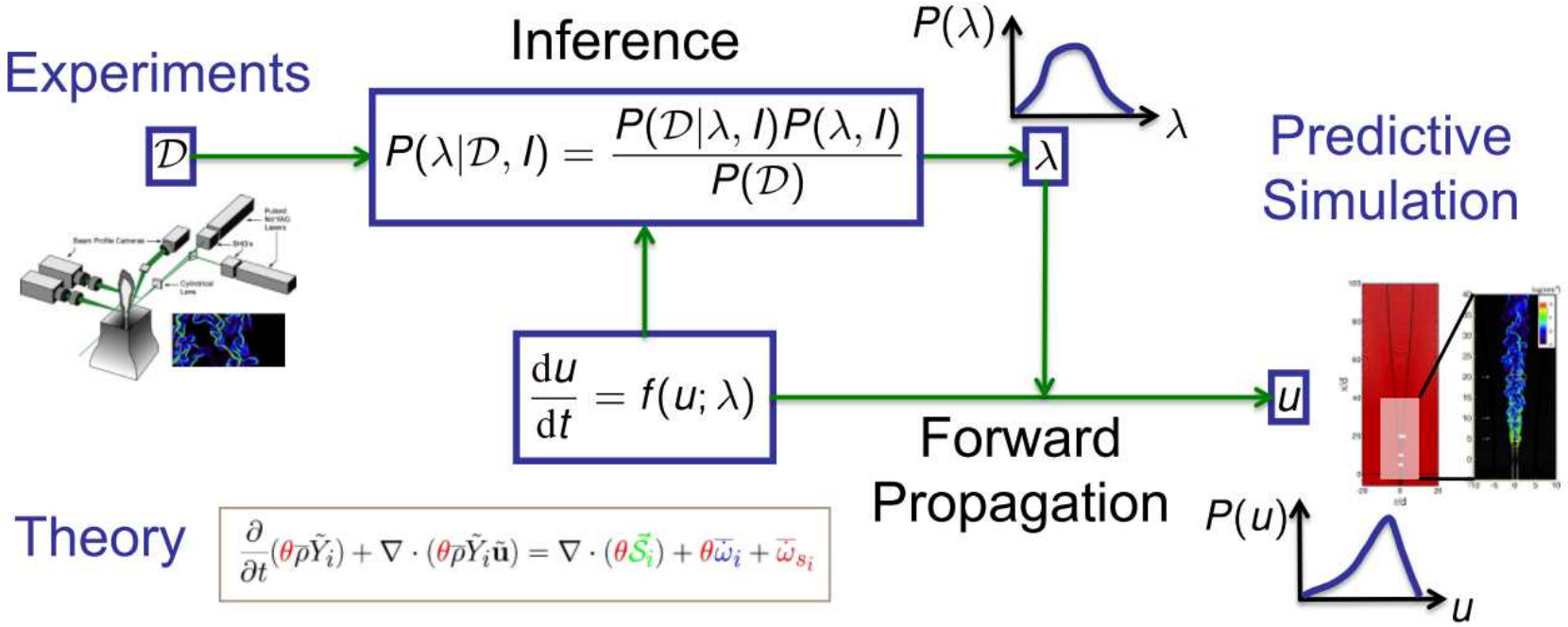
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Goals of This Tutorial

- 4 Lectures
  - Lecture 1: Context and Fundamentals
  - Lecture 2: Forward Propagation
  - Lecture 3: Characterization
    - Sensitivity Analysis
    - Representing Arbitrary Random Variables
    - Representing Random Fields
  - Lecture 4: Bayesian Inference – UQ Software

Outline

Sensitivity analysis gives insight into key sources of uncertainty



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

PC Postprocessing: global sensitivity information is readily obtained from PCE

$$g(\xi_1, \ldots, \xi_d) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

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

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

PC Postprocessing: Main Effect and Joint Sensitivity Indices

- Main effect sensitivity indices

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

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

- Joint sensitivity indices

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

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

PC Postprocessing: Total Effect Sensitivity Indices

- Total effect sensitivity indices

$$T_i = 1 - \frac{Var[\mathbb{E}(g(\boldsymbol{\xi}|\xi_{-i})]}{Var[g(\boldsymbol{\xi})]} = \frac{\sum_{k \in \mathbb{I}_i^T} c_k^2 ||\Psi_k||^2}{\sum_{k>0} c_k^2 ||\Psi_k||^2}$$

  - The notation ξ_{-i} indicates terms that do not have ξ_i in them
  - $\mathbb{I}_i^T$ is the set of bases with ξ_i involved, including all its interactions
  - The sum of all T_i is usually > 1

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$

$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$

$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Total sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Total sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Total sensitivities ξ_1 ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities (ξ_1, ξ_2) (ξ_1, ξ_3) (ξ_2, ξ_3)

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities (ξ_1, ξ_2) (ξ_1, ξ_3) (ξ_2, ξ_3)

Sensitivity indices are directly computable from PC

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

Consider dimensionality $d = 3$, total order $p = 2$,
number of PC terms $P + 1 = (d + p)!/(d!p!) = 10$.

$$g(\xi_1, \xi_2, \xi_3) = c_0 + c_1\psi_1(\xi_1) + c_2\psi_1(\xi_2) + c_3\psi_1(\xi_3) +$$
$$+ c_4\psi_2(\xi_1) + c_5\psi_1(\xi_1)\psi_1(\xi_2) + c_6\psi_1(\xi_1)\psi_1(\xi_3) + c_7\psi_2(\xi_2) + c_8\psi_1(\xi_2)\psi_1(\xi_3) + c_9\psi_2(\xi_3)$$

Variance contributions

$$Var(g) = 0 + c_1^2\langle\psi_1^2\rangle + c_2^2\langle\psi_1^2\rangle + c_3^2\langle\psi_1^2\rangle +$$
$$+ c_4^2\langle\psi_2^2\rangle + c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_7^2\langle\psi_2^2\rangle + c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle + c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities (ξ_1, ξ_2) (ξ_1, ξ_3) (ξ_2, ξ_3)

PC Postprocessing: Sampling-Based Approaches

$$g(\xi_1, \ldots, \xi_d) = \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi})$$

- In some cases, need to resort to Monte-Carlo estimation, e.g.
  - Piecewise-PC with irregular subdomains
  - Output transformations, e.g. build PC for $\log g(\boldsymbol{\xi})$, but inquire sensitivity with respect to $g(\boldsymbol{\xi})$
- A brute-force sampling of $Var[\mathbb{E}(g(\boldsymbol{\xi}|\xi_i)]$ is extremely inefficient.

PC Postprocessing: Sampling-Based Approaches

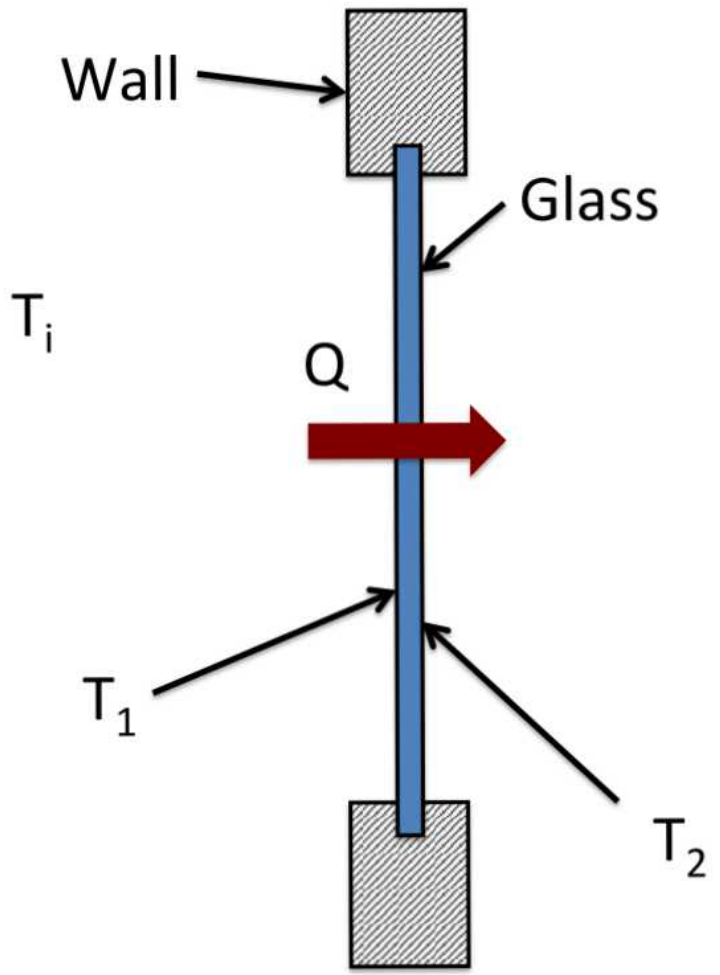
- Tricks are available, given a single set of sampled input [Saltelli, 2002]. E.g., use

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

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

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

Heat Transfer through a Window



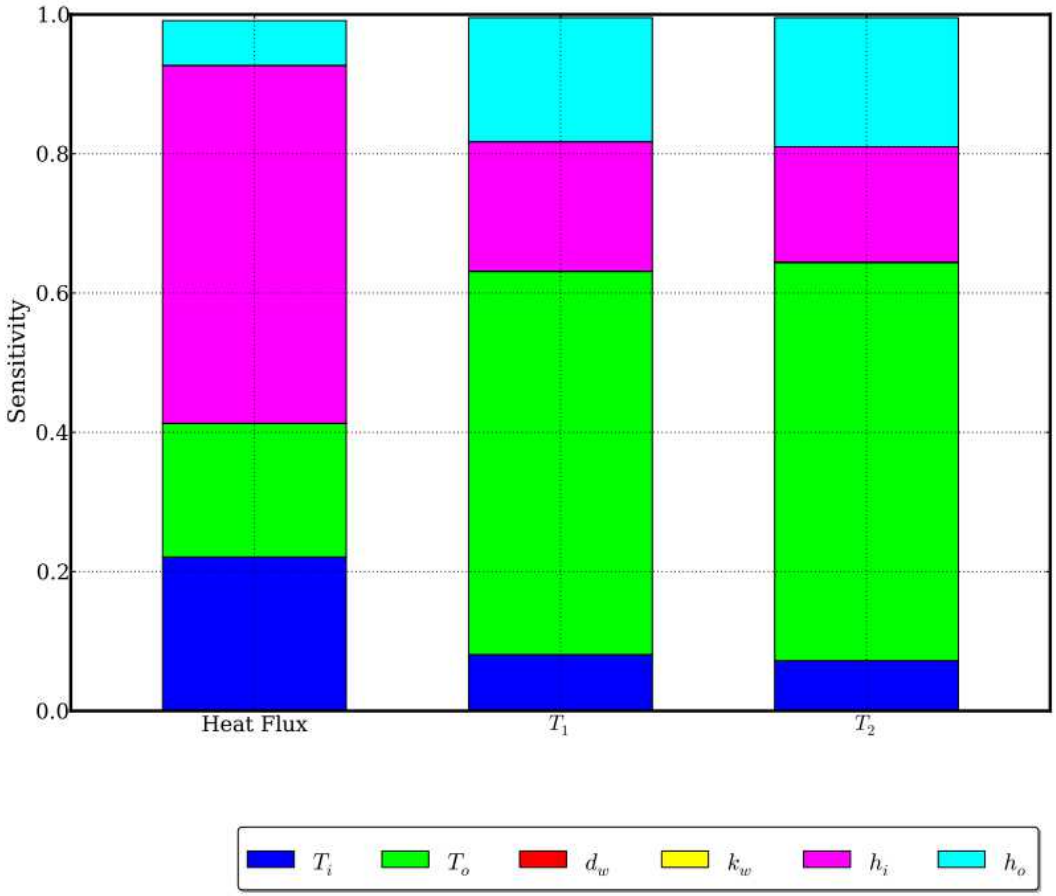
$$h_i(T_i - T_1) = k_w \frac{(T_1 - T_2)}{d_w}$$

$$k_w \frac{(T_1 - T_2)}{d_w} = h_o(T_2 - T_o)$$

6 Uncertain, Gaussian parameters

$$T_i = 293\text{K}, \sigma = 0.5\%$$
$$T_o = 273\text{K}, \sigma = 0.5\%$$
$$d_w = 0.01\text{m}, \sigma = 1\%$$
$$k_w = 1\text{W/mK}, \sigma = 5\%$$
$$h_i = 2\text{W/m}^2\text{K}, \sigma = 15\%$$
$$h_o = 6\text{W/m}^2\text{K}, \sigma = 15\%$$

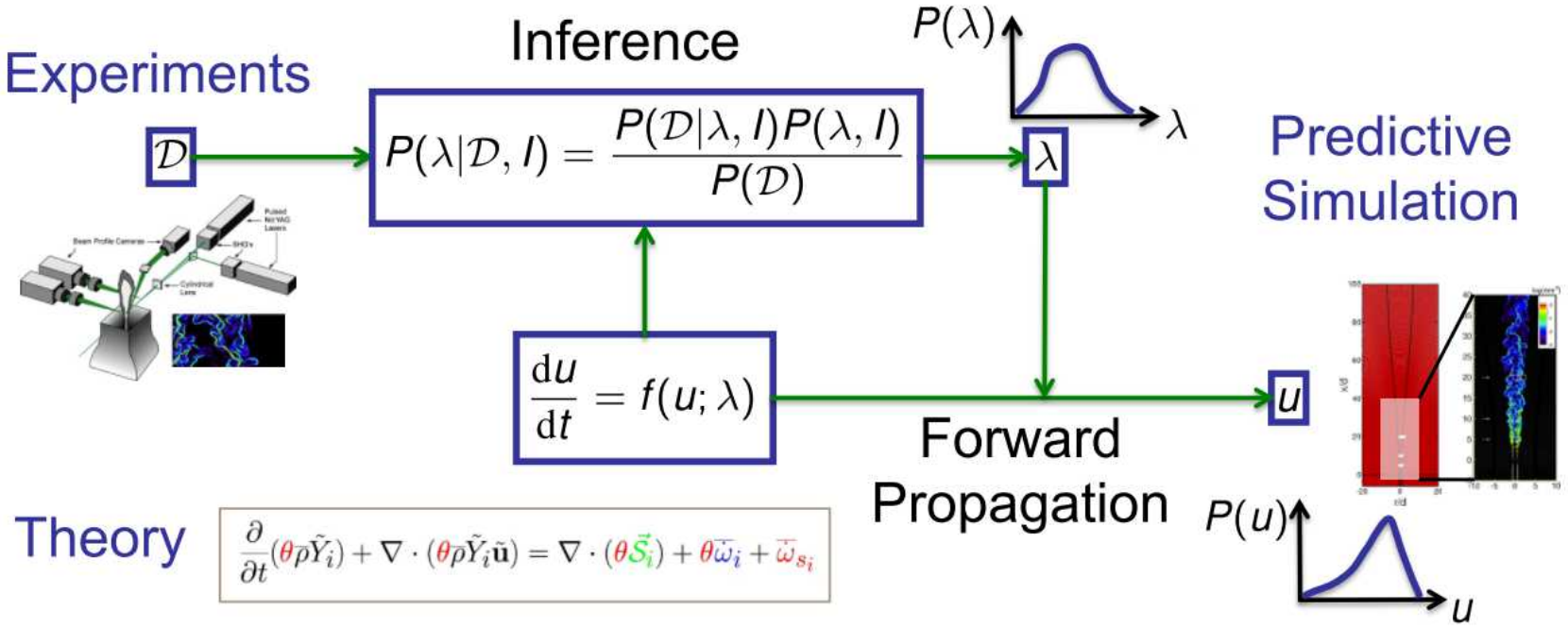
Outputs are most sensitive to ambient temperatures and convective heat transfer coefficients



- Main effect sensitivities
  - Sum to 1 only if coupling terms do not matter
- k_w has minimal contribution due to its low uncertainty

Outline

Obtaining PCEs for Uncertain Inputs

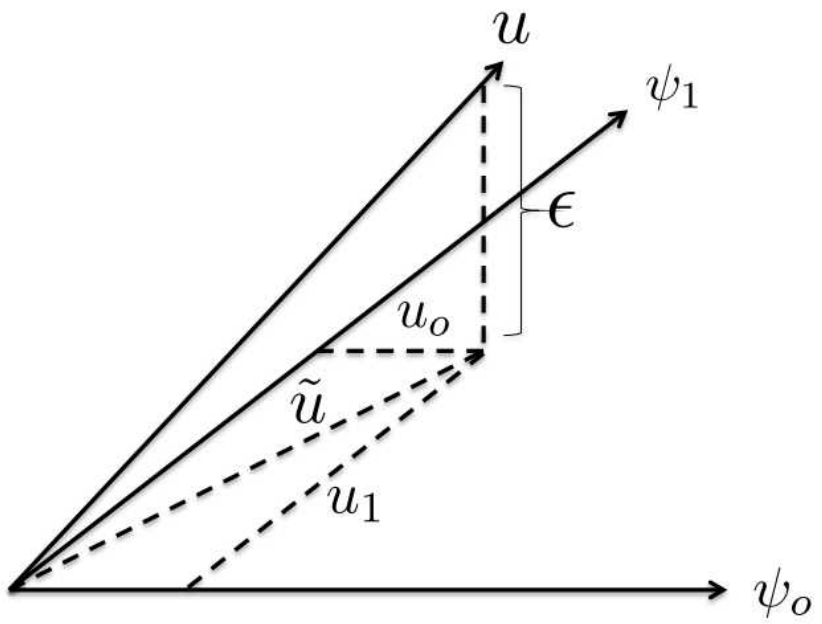


- Representation of uncertain inputs λ with PCEs

Obtaining PC coefficients for arbitrary random variables is not trivial

- Characterizing PCEs for uncertain inputs is a really difficult problem
- Inputs specified in a variety of ways, and often incomplete
  - Probability density function
  - Samples, *e.g.* from inverse problem solution
  - Expert opinion (*e.g.* “about 3.5”)
- Particular case of a random variable specified by a PDF is generally tractable

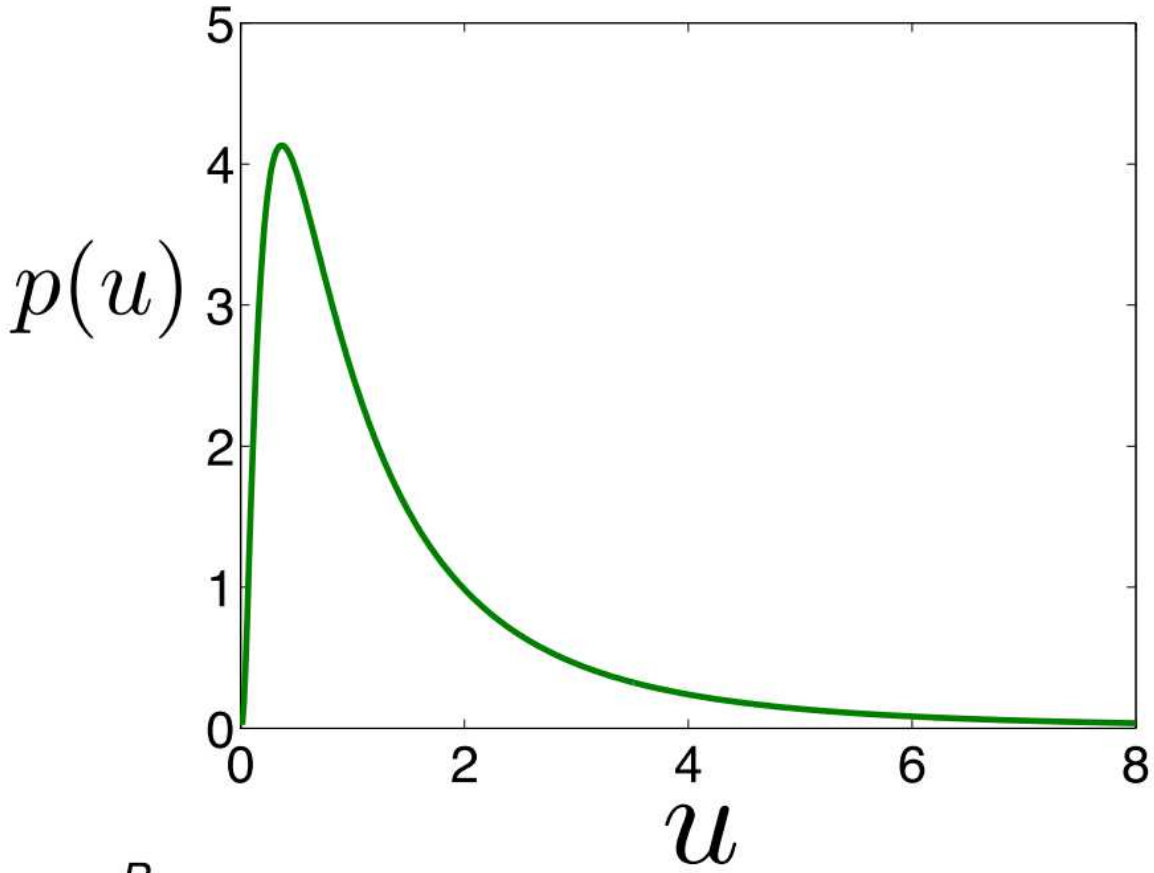
Orthogonality enables a Galerkin projection to determine the PC coefficients



$$u \approx \sum_{k=0}^{P} u_k \Psi_k \qquad \Rightarrow \qquad \langle \Psi_i u \rangle = \sum_{k=0}^{P} u_k \langle \Psi_i \Psi_k \rangle = u_i \langle \Psi_i^2 \rangle$$

$$\Rightarrow \qquad u_i = \frac{\langle u \Psi_i \rangle}{\langle \Psi_i^2 \rangle}$$

Galerkin projection requires functional relationship between random variable and germ of PCE



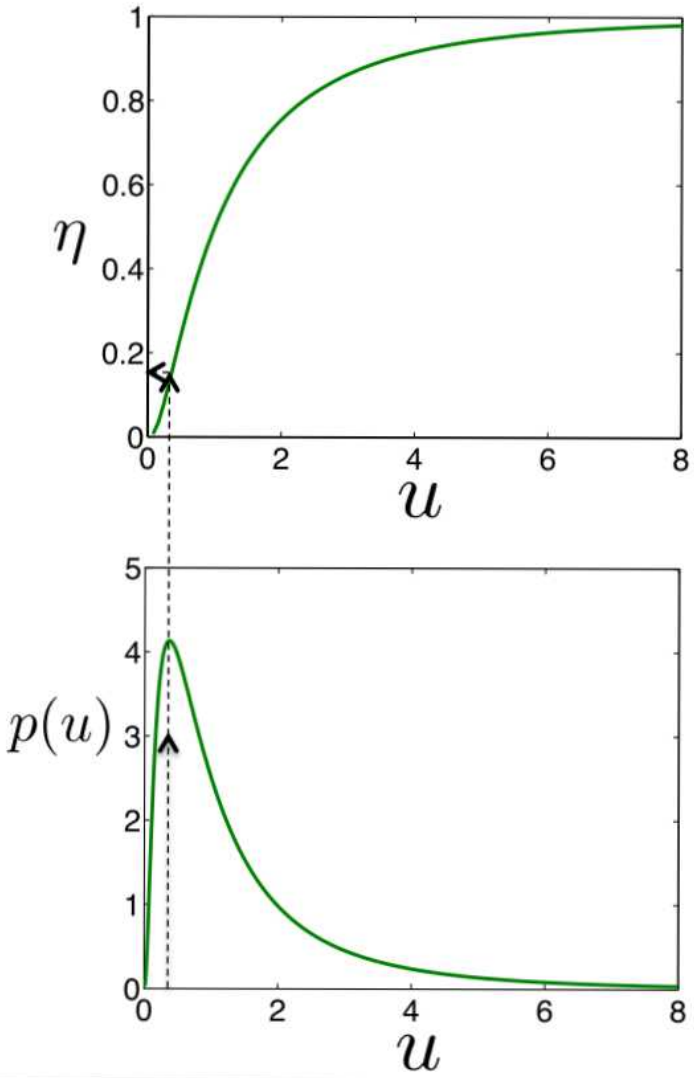
$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi) \qquad \Rightarrow \qquad u_i = \frac{\langle u\, \Psi_i(\xi) \rangle}{\langle \Psi_i^2 \rangle}$$

Cumulative Distribution Function (CDF) maps arbitrary random variable to a uniform random variable

- Consider u with PDF $p(u)$
- CDF of u is given by

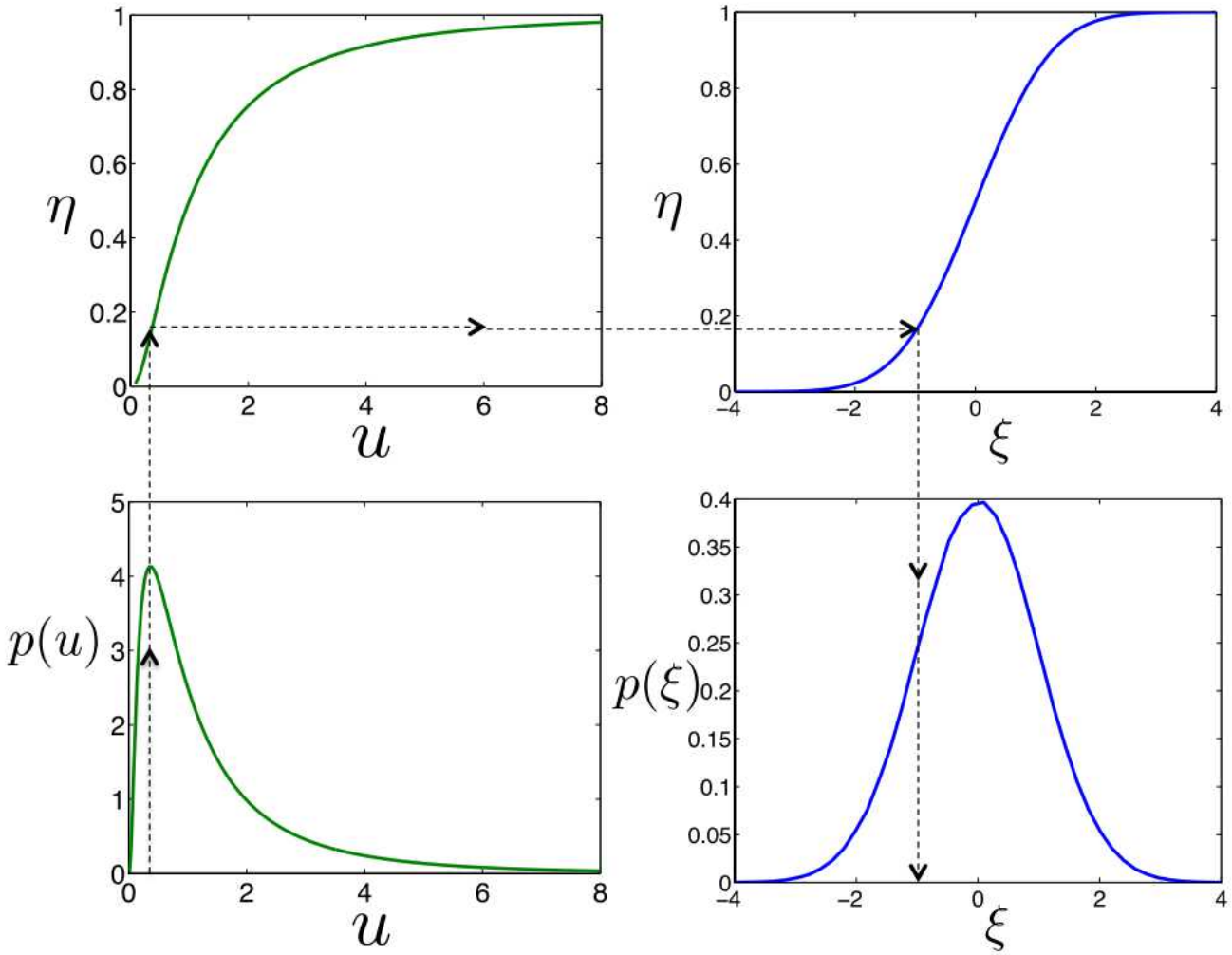
$$F(u) = \int_{-\infty}^{u} p(s)ds$$

- $F(u)$ maps u to η, uniform on $[0, 1]$



Inverse CDF mapping enables Galerkin Projection

- $\eta = F(u)$
- $\eta = \Phi(\xi)$ maps uniform η to normal RV ξ
- $u = F^{-1}(\Phi(\xi))$

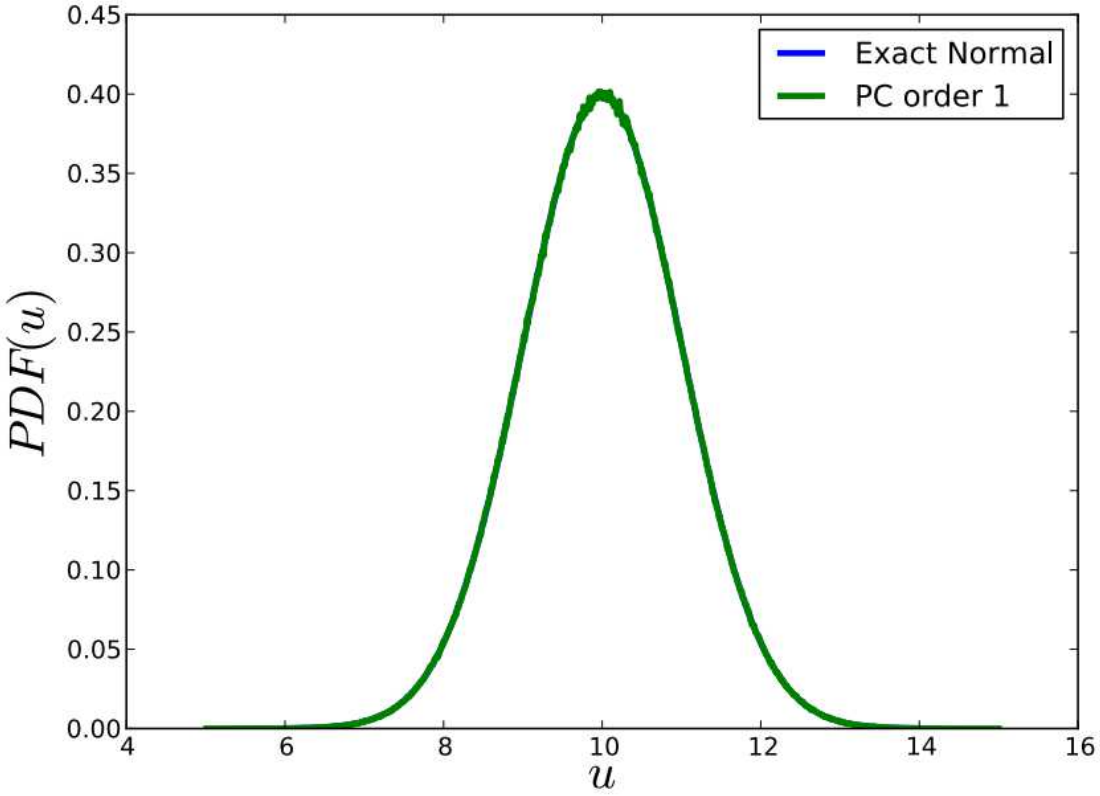


$$\langle u\, \Psi_i(\xi) \rangle = \int \underbrace{F^{-1}(\Phi(\xi))}_{u} \Psi_i(\xi) w(\xi) d\xi$$

PC Illustration: PC Expansion for a Normal RV

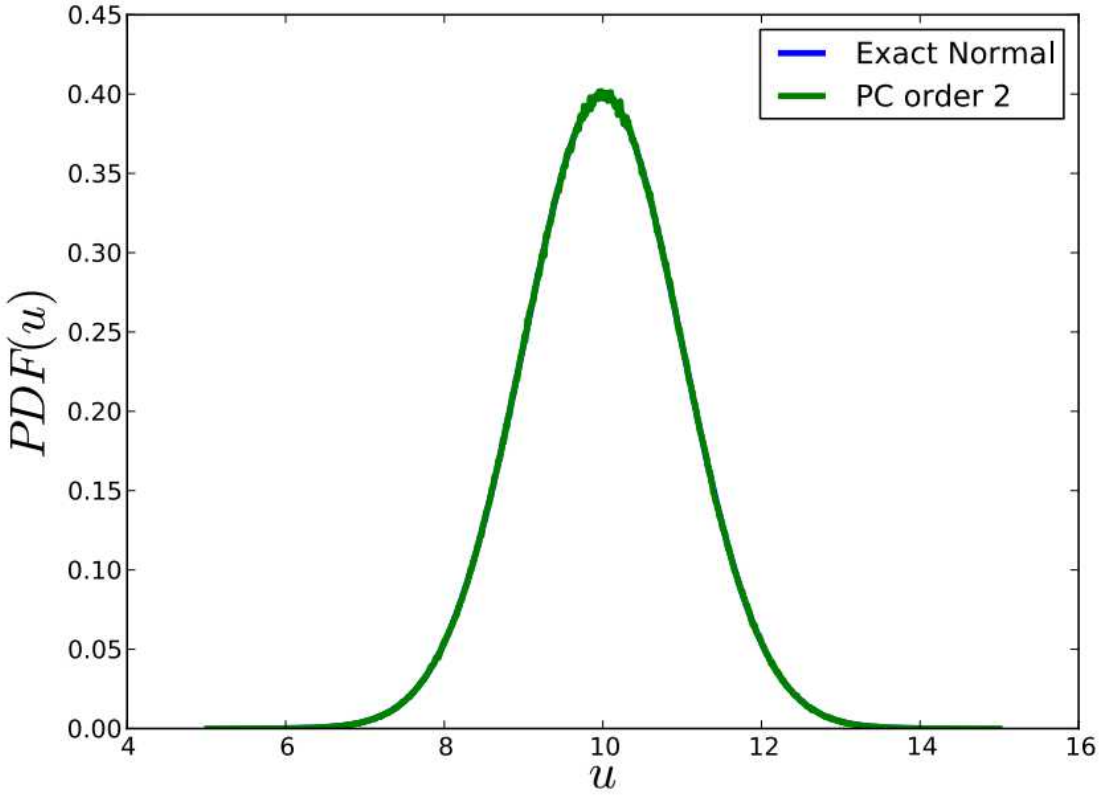
- Wiener-Hermite PCE constructed for a Normal RV
- PCE-sampled PDF superposed on true PDF
- Order = 1

$$
\begin{aligned}
u &= \sum_{k=0}^{P} u_k \Psi_k(\xi) \\
&= u_0 + u_1 \xi
\end{aligned}
$$



PC Illustration: PC Expansion for a Normal RV

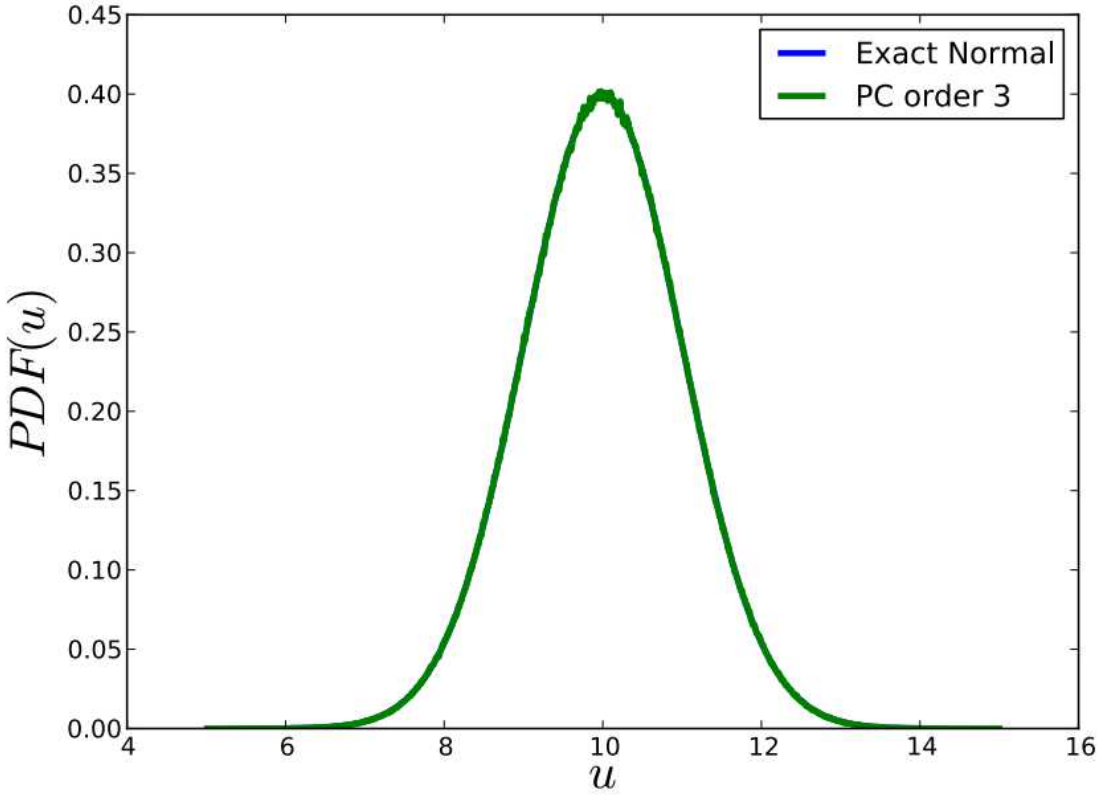
- Wiener-Hermite PCE constructed for a Normal RV
- PCE-sampled PDF superposed on true PDF
- Order = 2



$$
\begin{aligned}
u &= \sum_{k=0}^{P} u_k \Psi_k(\xi) \\
&= u_0 + u_1 \xi + u_2(\xi^2 - 1)
\end{aligned}
$$

PC Illustration: PC Expansion for a Normal RV

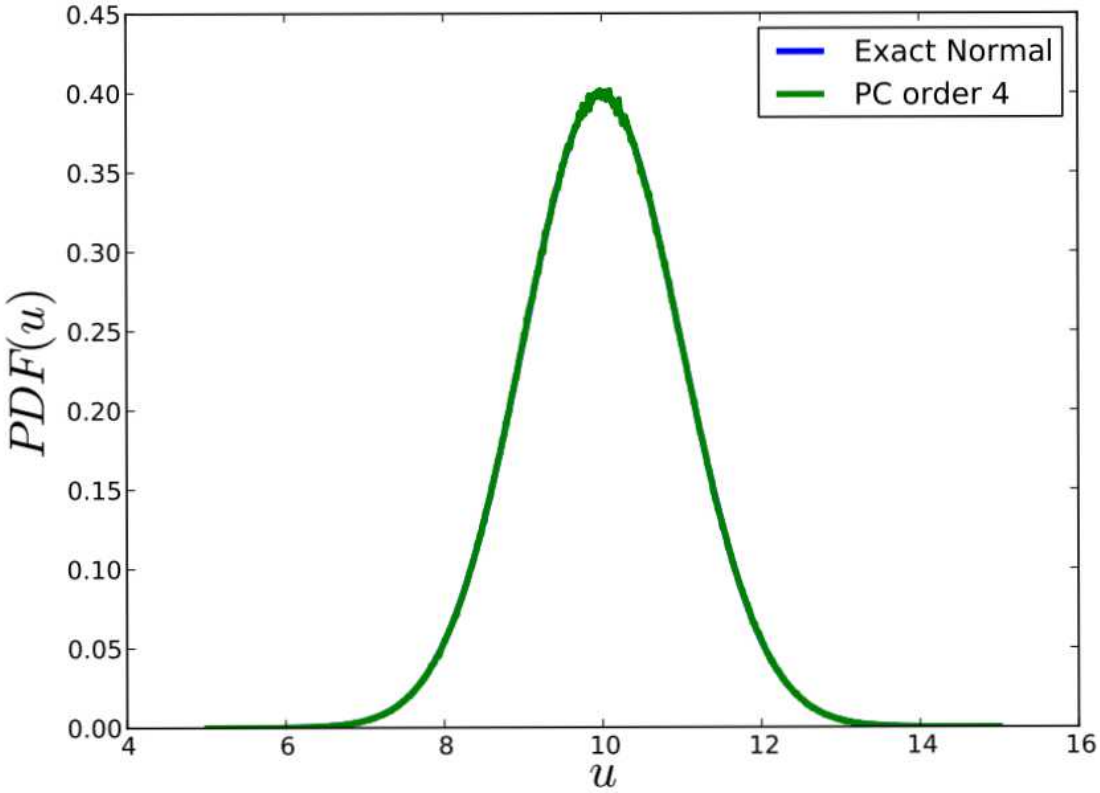
- Wiener-Hermite PCE constructed for a Normal RV
- PCE-sampled PDF superposed on true PDF
- Order = 3



$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$
$$= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi)$$

PC Illustration: PC Expansion for a Normal RV

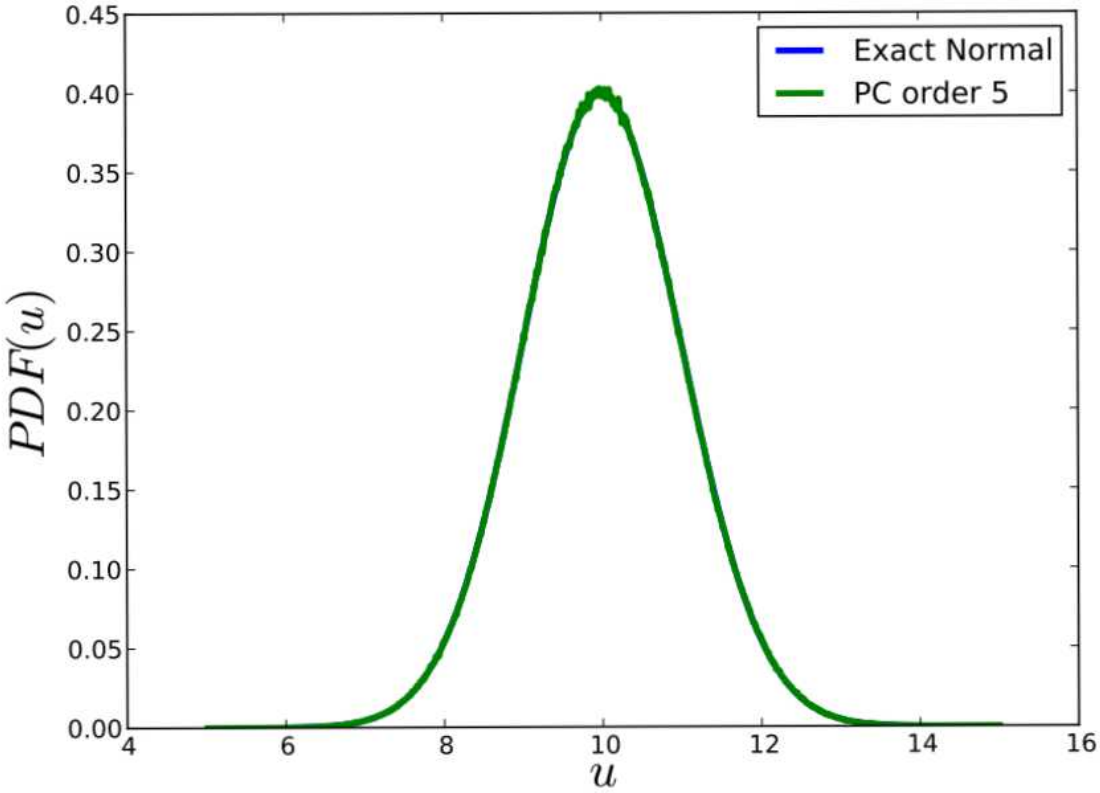
- Wiener-Hermite PCE constructed for a Normal RV
- PCE-sampled PDF superposed on true PDF
- Order = 4



$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$
$$= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi) + u_4(\xi^4 - 6\xi^2 + 3)$$

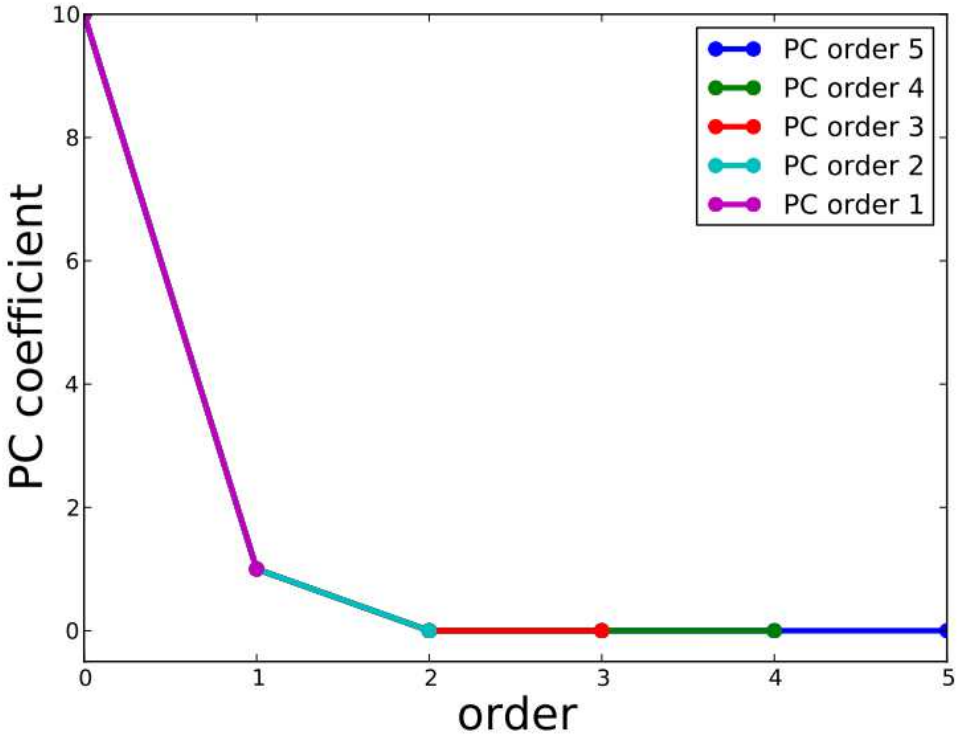
PC Illustration: PC Expansion for a Normal RV

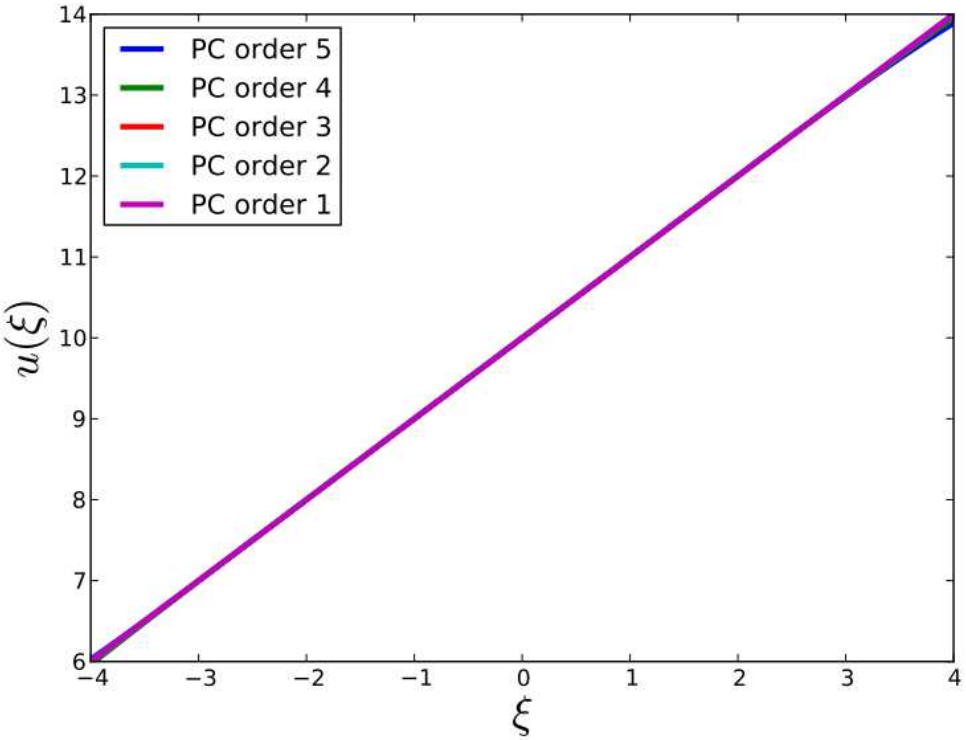
- Wiener-Hermite PCE constructed for a Normal RV
- PCE-sampled PDF superposed on true PDF
- Order = 5



$$
\begin{aligned}
u &= \sum_{k=0}^{P} u_k \Psi_k(\xi) \\
&= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi) + u_4(\xi^4 - 6\xi^2 + 3) \\
&\quad + u_5(\xi^5 - 10\xi^3 + 15\xi)
\end{aligned}
$$

PC Illustration: WH PCE for a Normal RV



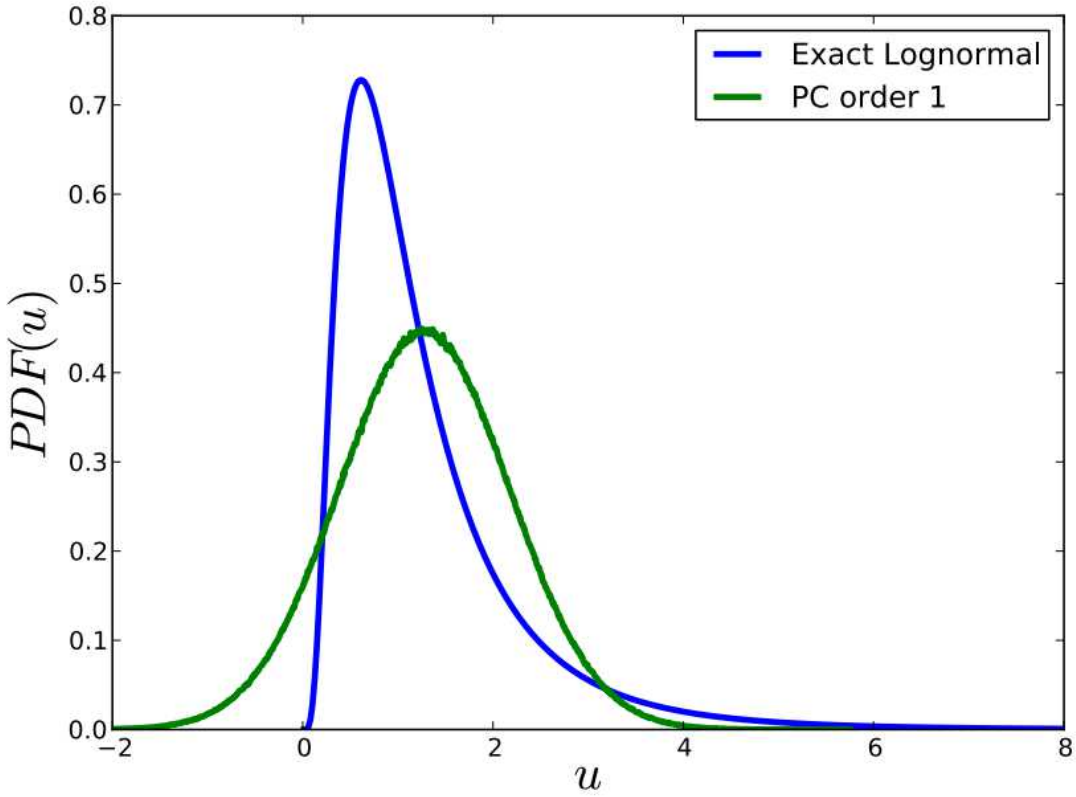


- First order Wiener-Hermite PCE exact for a normal RV
- Linear function of ξ
- Higher order terms are negligible

PC Illustration: WH PCE for a Lognormal RV

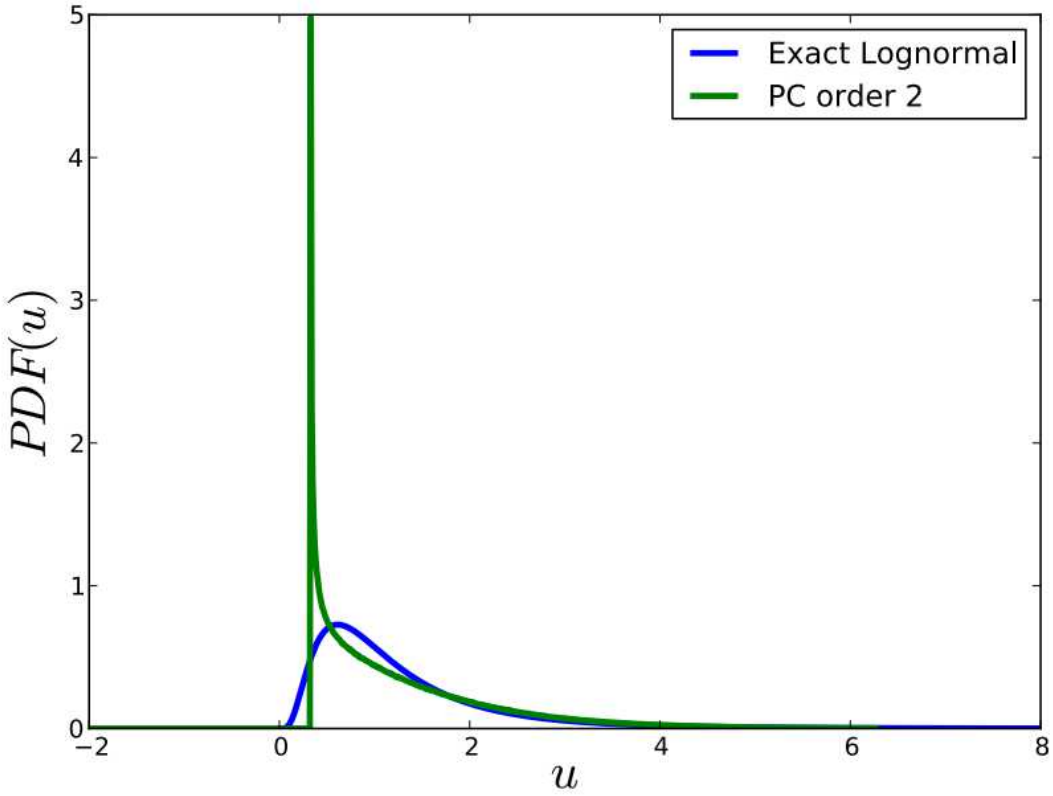
- Wiener-Hermite PCE constructed for a Lognormal RV
- PCE-sampled PDF superposed on true PDF
- Order = 1

$$
\begin{aligned}
u &= \sum_{k=0}^{P} u_k \Psi_k(\xi) \\
&= u_0 + u_1 \xi
\end{aligned}
$$



PC Illustration: WH PCE for a Lognormal RV

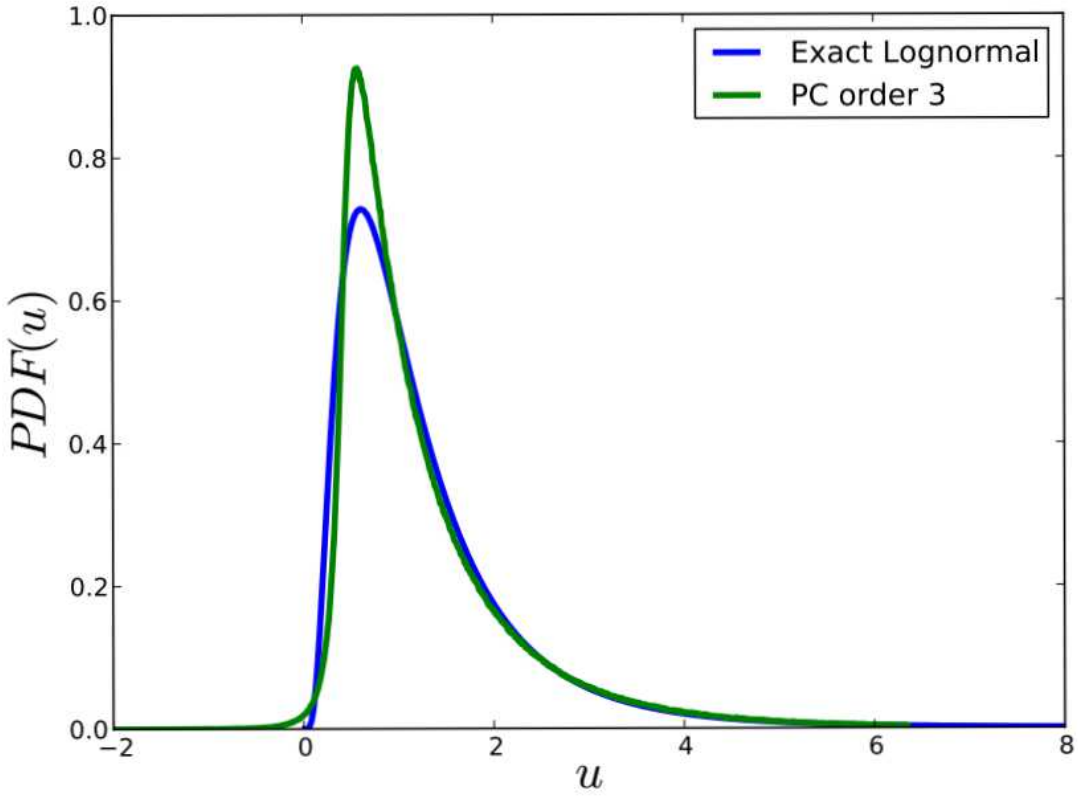
- Wiener-Hermite PCE constructed for a Lognormal RV
- PCE-sampled PDF superposed on true PDF
- Order = 2



$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$
$$= u_0 + u_1\xi + u_2(\xi^2 - 1)$$

PC Illustration: WH PCE for a Lognormal RV

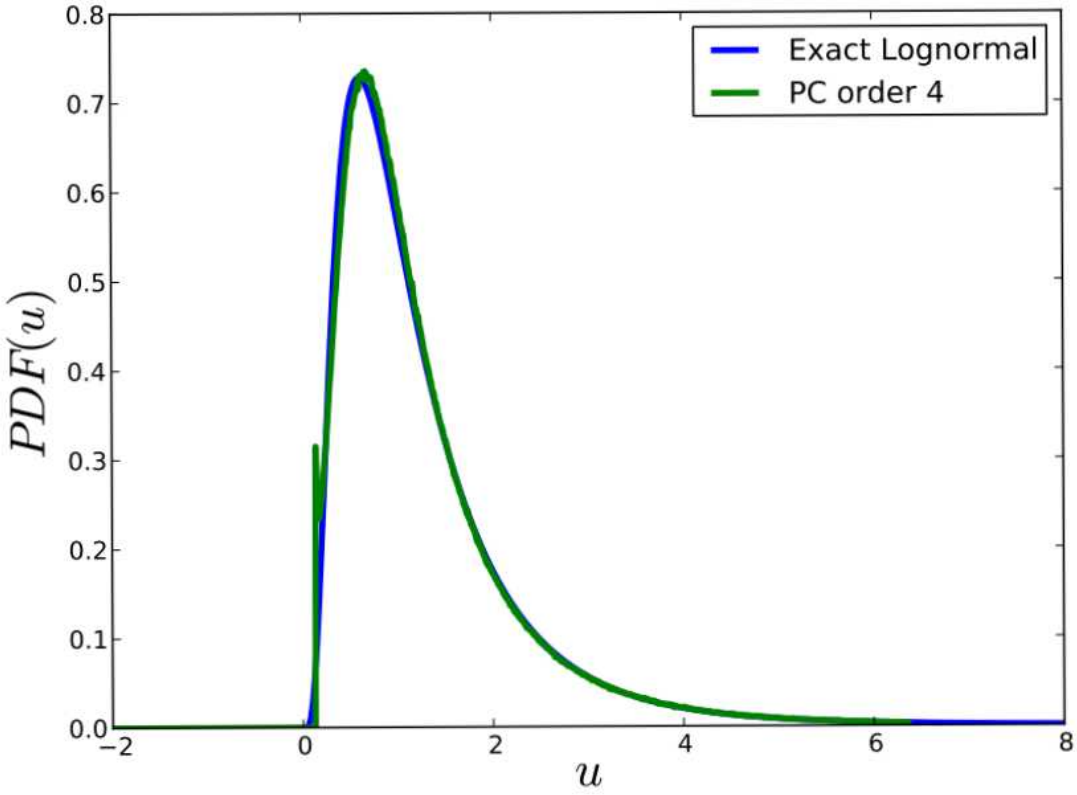
- Wiener-Hermite PCE constructed for a Lognormal RV
- PCE-sampled PDF superposed on true PDF
- Order = 3



$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$
$$= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi)$$

PC Illustration: WH PCE for a Lognormal RV

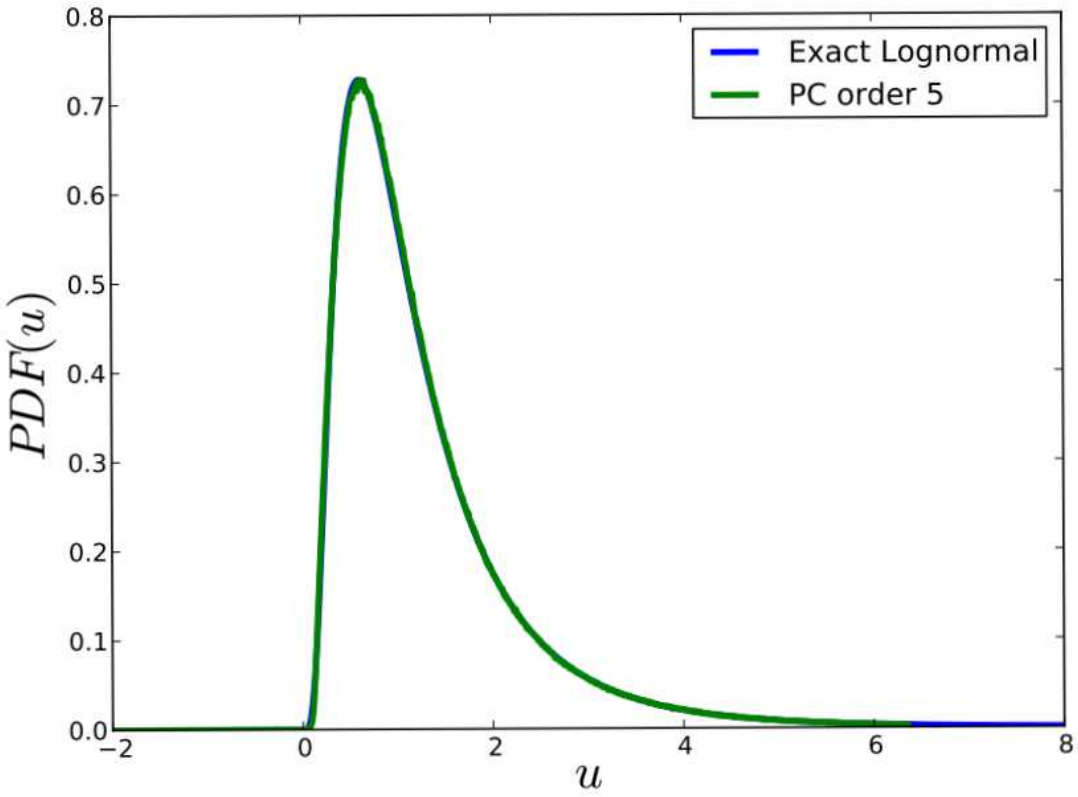
- Wiener-Hermite PCE constructed for a Lognormal RV
- PCE-sampled PDF superposed on true PDF
- Order = 4



$$u = \sum_{k=0}^{P} u_k \Psi_k(\xi)$$
$$= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi) + u_4(\xi^4 - 6\xi^2 + 3)$$

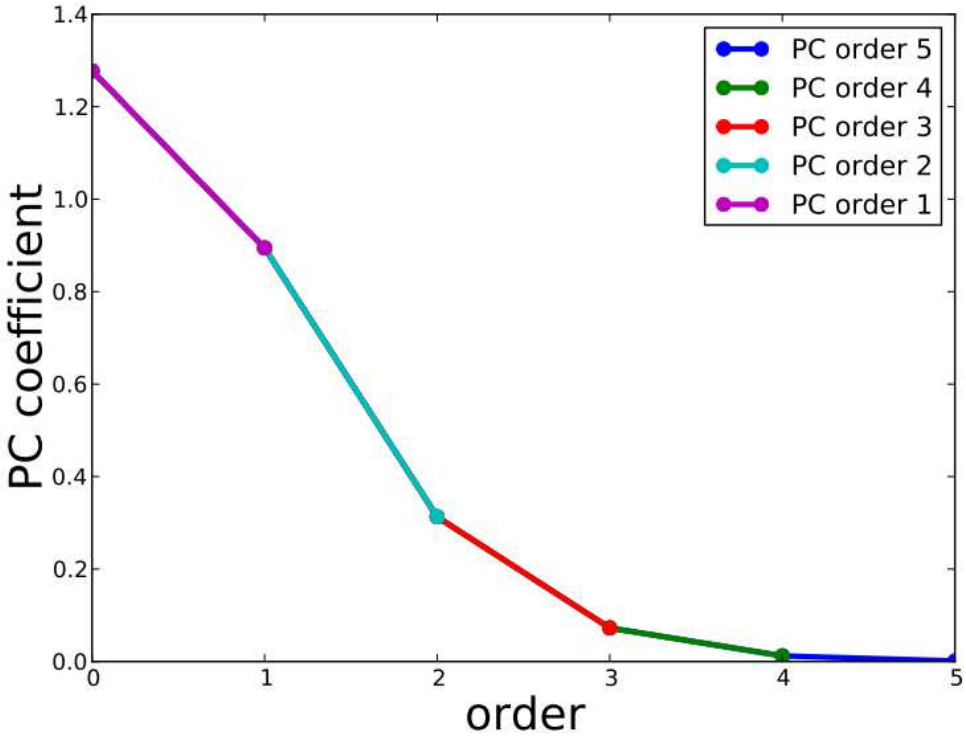
PC Illustration: WH PCE for a Lognormal RV

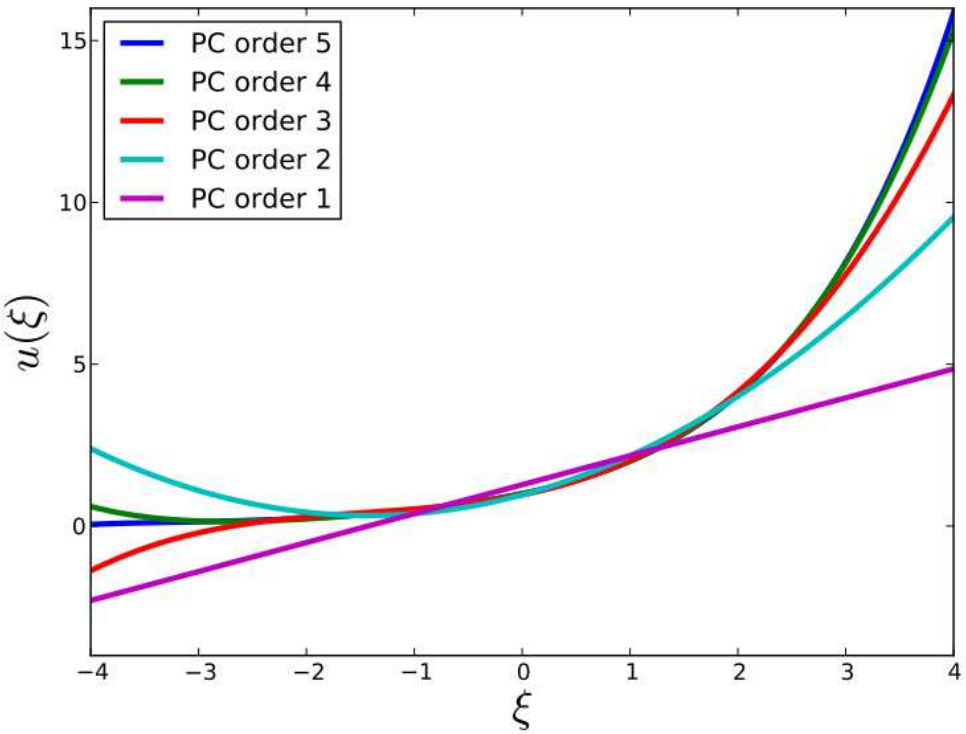
- Wiener-Hermite PCE constructed for a Lognormal RV
- PCE-sampled PDF superposed on true PDF
- Order = 5



$$
\begin{aligned}
u &= \sum_{k=0}^{P} u_k \Psi_k(\xi) \\
&= u_0 + u_1\xi + u_2(\xi^2 - 1) + u_3(\xi^3 - 3\xi) + u_4(\xi^4 - 6\xi^2 + 3) \\
&\quad + u_5(\xi^5 - 10\xi^3 + 15\xi)
\end{aligned}
$$

PC Illustration: WH PCE for a Lognormal RV





- Fifth-order Wiener-Hermite PCE represents the given Lognormal well
- Higher order terms are negligible

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
- Construct PCE as model choice to represent what is known about RV
- 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 $\phi(z)$

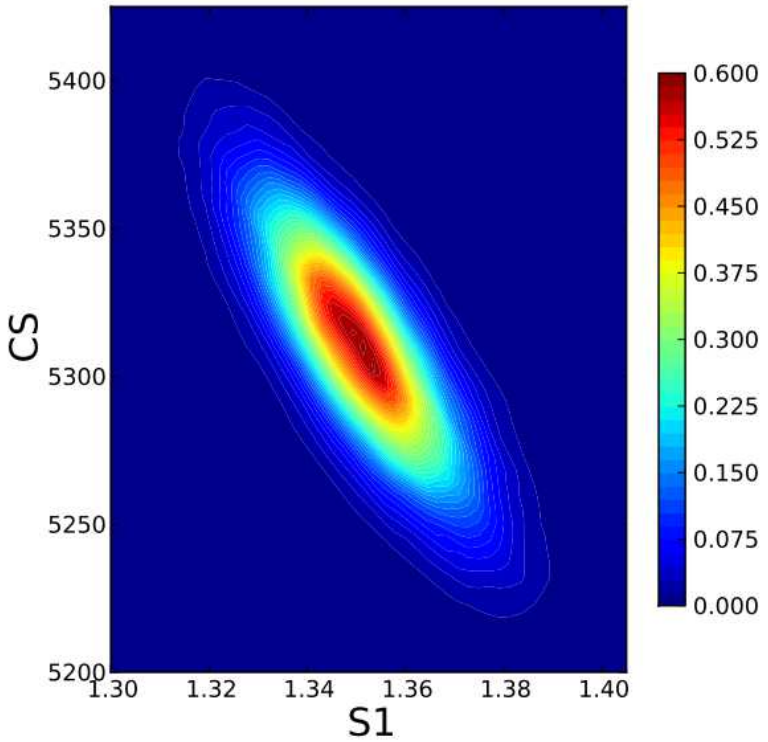
Multivariate Normal Approximation (MVN)

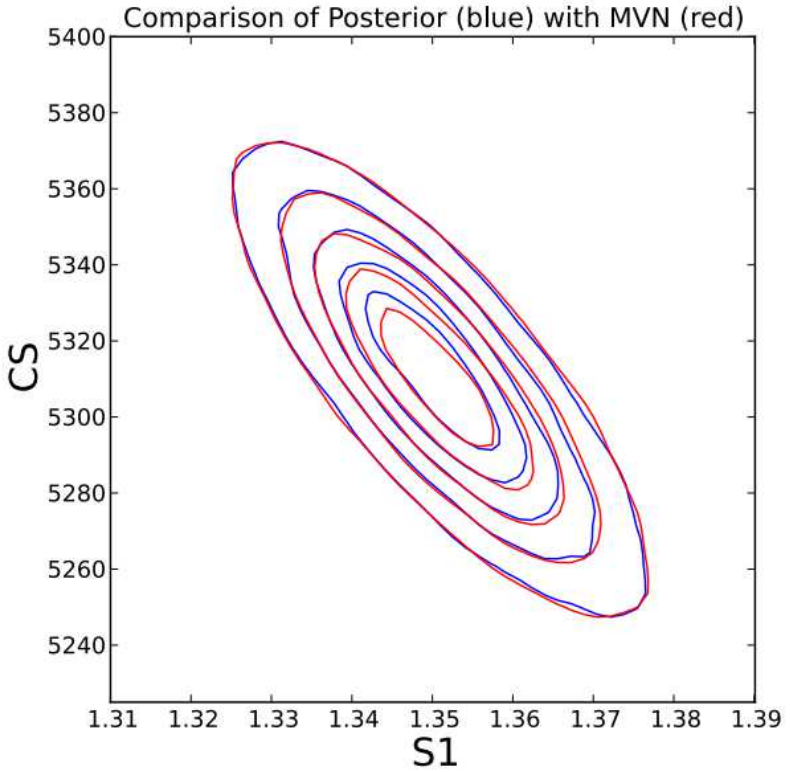
- Many distributions are unimodal and somewhat shaped like Gaussians
- MultiVariate Normal approximations capture mean and correlation structure of the random variables
- Easy to extract from a set of samples
  - In 1D: just compute mean and standard deviation: $u = u_0 + u_1 \xi$
  - Multi-D: Cholesky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

$$C = LL^T$$
$$u = L\xi$$

```
# 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 Distribution from Samples





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

Rosenblatt Transformation for Multi-D RVs

- Assume samples of multi-D RVs are (*e.g.* from MCMC sampling of posterior parameter distribution)
- Rosenblatt transformation maps any (not necessarily independent) set of random variables $(\lambda_1, \ldots, \lambda_d)$ to uniform i.i.d.'s $\{\eta_i\}_{i=1}^d$ (Rosenblatt, 1952).

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

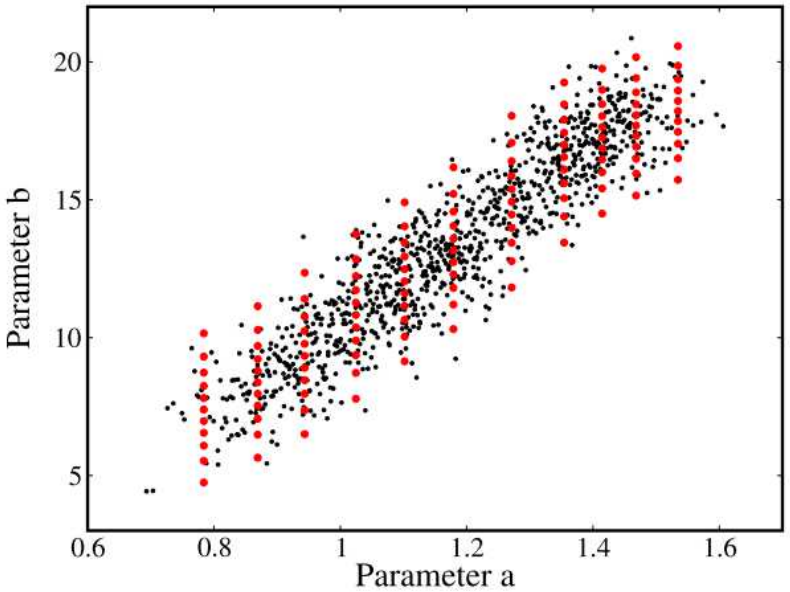
- Rosenblatt transformation is a multi-D generalization of 1D CDF mapping.
- Conditional CDFs are harder to evaluate in high dimensions

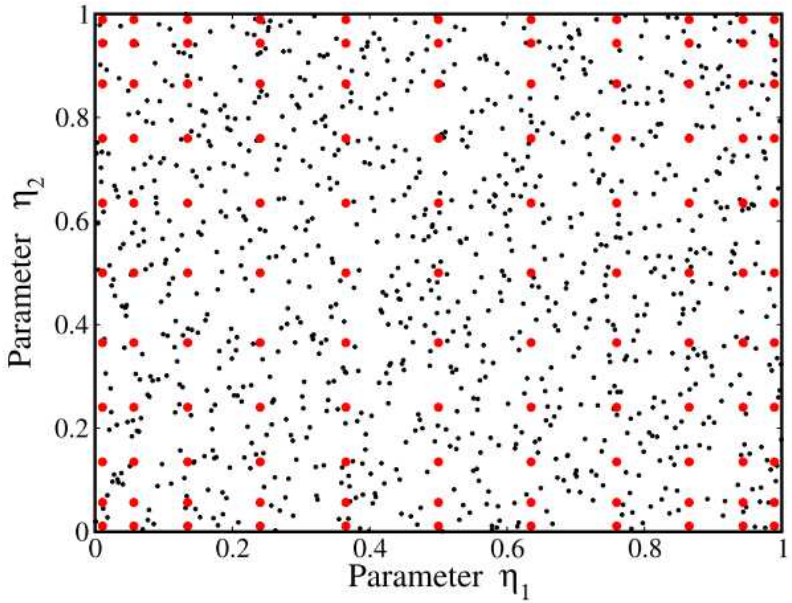
Rosenblatt transformation of a given sample set requires KDE

- Given samples $\{\boldsymbol{\lambda}^{(n)}\}_{n=1}^{N}$ of the random variable $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$
- Kernel Density Estimation (KDE) is needed to compute conditional CDFs

$$
\begin{aligned}
\eta_k &= F_{k|k-1,\ldots,1}(\lambda_k|\lambda_{k-1},\ldots,\lambda_1) = \\
&= \int_{\infty}^{\lambda_k} p_{k|k-1,\ldots,1}(\lambda_k^{'}|\lambda_{k-1},\ldots,\lambda_1)d\lambda_k^{'} = \int_{\infty}^{\lambda_k} \frac{p_{k,\ldots,1}(\lambda_k^{'},\lambda_{k-1},\ldots,\lambda_1)}{p_{k-1,\ldots,1}(\lambda_{k-1},\ldots,\lambda_1)} d\lambda_k^{'} \\
&\approx \frac{1}{h\sqrt{2\pi}} \int_{\infty}^{\lambda_k} \frac{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_k^{'}-\lambda_k^{(n)})^2}{2h^2}\right)}{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_{k-1}-\lambda_{k-1}^{(n)})^2}{2h^2}\right)} d\lambda_k^{'} \\
&= \int_{\infty}^{\lambda_k} \frac{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_{k-1}-\lambda_{k-1}^{(n)})^2}{2h^2}\right) \times \frac{1}{h\sqrt{2\pi}} \exp\left(-\frac{(\lambda_k^{'}-\lambda_k^{(n)})^2}{2h^2}\right)}{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_{k-1}-\lambda_{k-1}^{(n)})^2}{2h^2}\right)} d\lambda_k^{'} \\
&= \frac{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_{k-1}-\lambda_{k-1}^{(n)})^2}{2h^2}\right) \times \Phi\left(\frac{\lambda_k-\lambda_k^{(n)}}{h}\right)}{\sum_{n=1}^{N} \exp\left(-\frac{(\lambda_1-\lambda_1^{(n)})^2+\cdots+(\lambda_{k-1}-\lambda_{k-1}^{(n)})^2}{2h^2}\right)},
\end{aligned}
$$

Rosenblatt transformation enables Galerkin projection





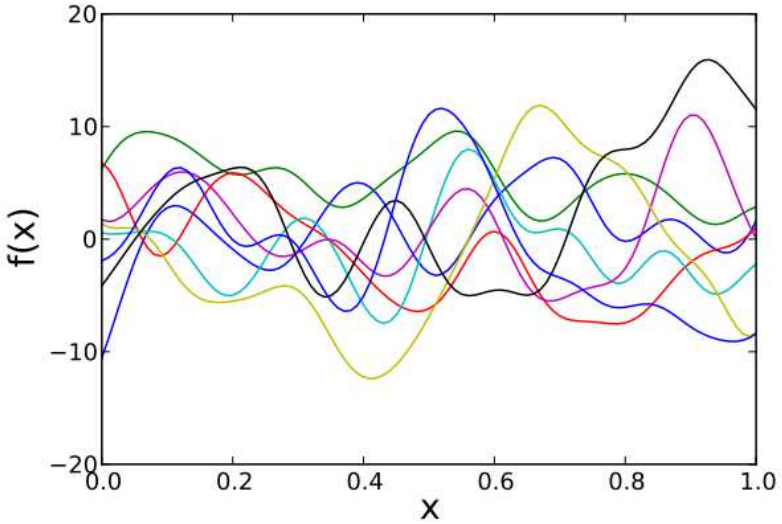
$(a, b) = R^{-1}(\xi_1, \xi_2)$ ensures a well-defined quadrature integration

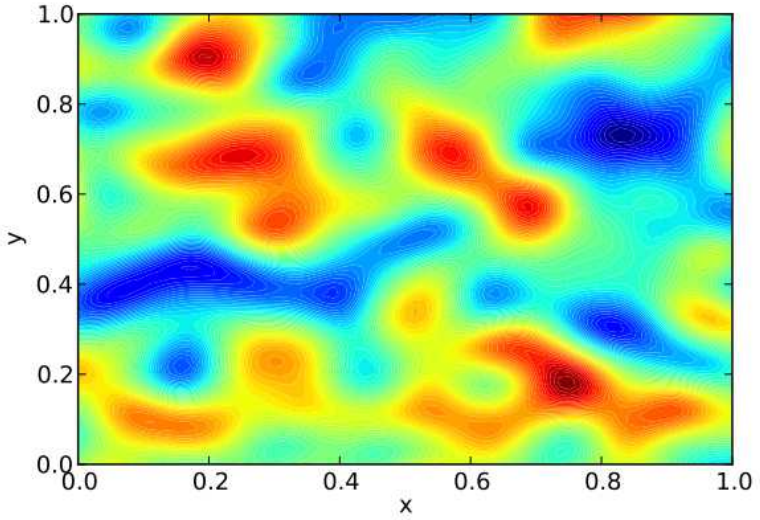
$$a = \sum_{k=0}^{P} a_k \Psi_k(\xi) \qquad a_k \propto \int \underbrace{R_a^{-1}(\xi)}_{a} \Psi_k(\xi) w(\xi) d\xi$$

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

Outline

Compact Representations of Random Fields





- Sometimes an uncertain input is a random field (also called stochastic process)
  - Friction coefficient along a wall
  - Thermal conductivity in heterogeneous material
  - Permeability in groundwater flow simulations
- Have infinitely many degrees of freedom
- To model effect of random fields on computations, need compact representation that captures key aspects of random field

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 obtained by projecting realizations of F onto f_k
- Generally not independent
  - 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 $\{f(x)\}$ 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 eigenvalues and eigenvectors for the Fredholm equation of second kind:

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

KL Expansions - Numerical Approach - 2

- Nystrom algorithm for Fredholm equation

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

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 - 3

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

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

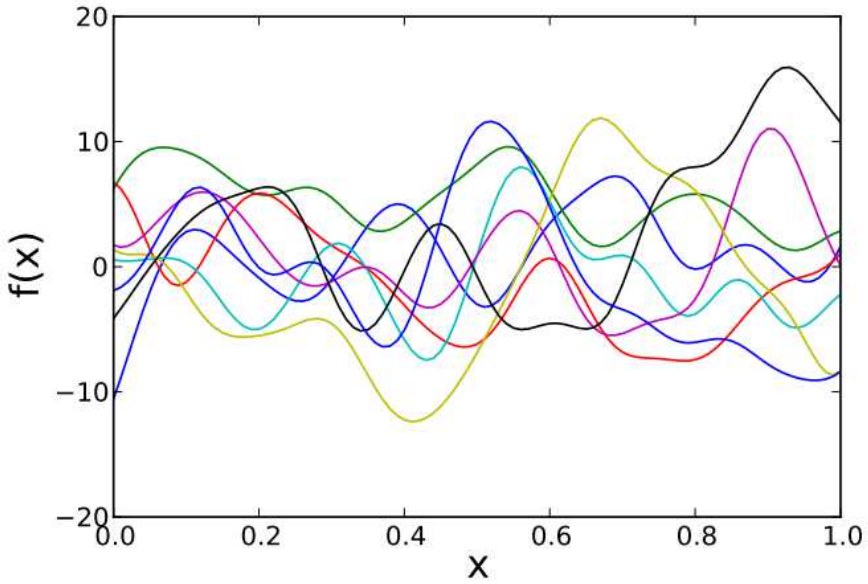
  - ... or numerically

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

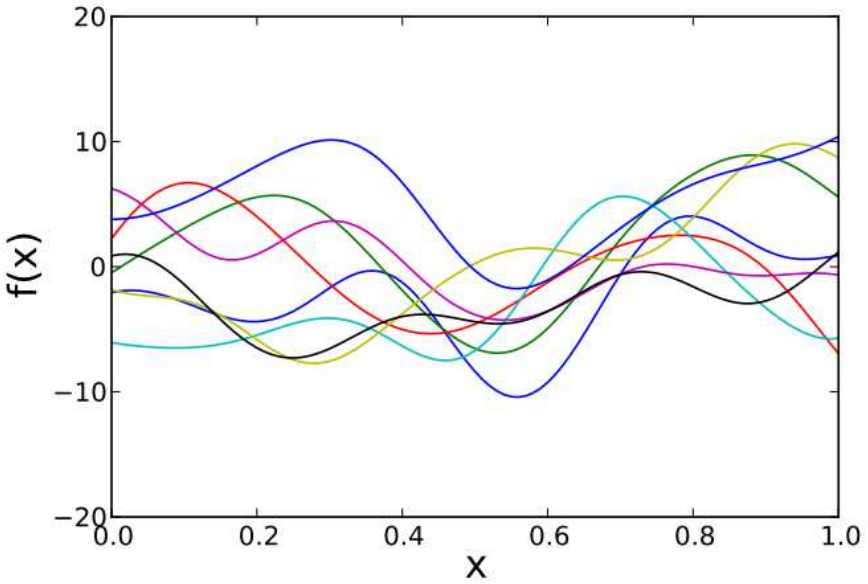
- If Gaussian process: automatically have first order WH PCE
- If not, same approaches as for converting RVs to PCEs applied to KL RVs

1D Gaussian Process: Realizations

$\delta = 0.1$



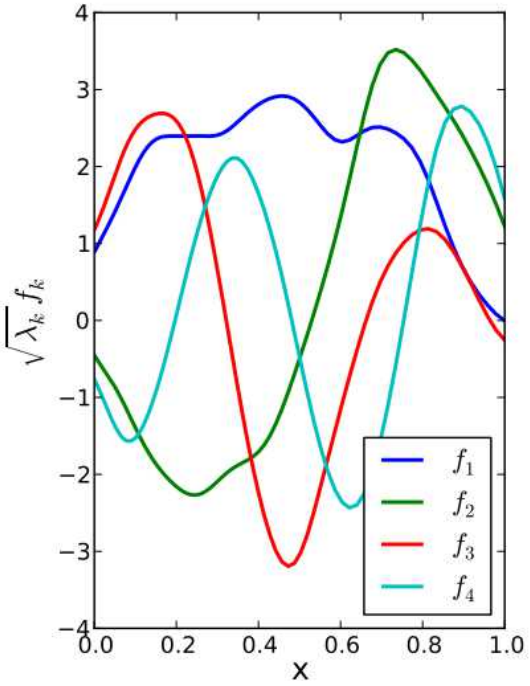
$\delta = 0.2$



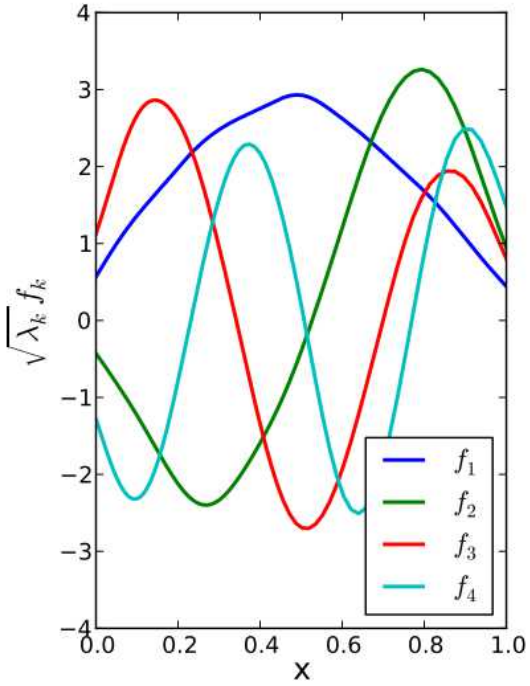
- Covariance $Cov(x_1, x_2) = exp(-(x_1 - x_2)^2/\delta^2)$
- Sample realizations are noisier as correlation length decreases

1D Gaussian Process: KL Modes for $\delta = 0.10$

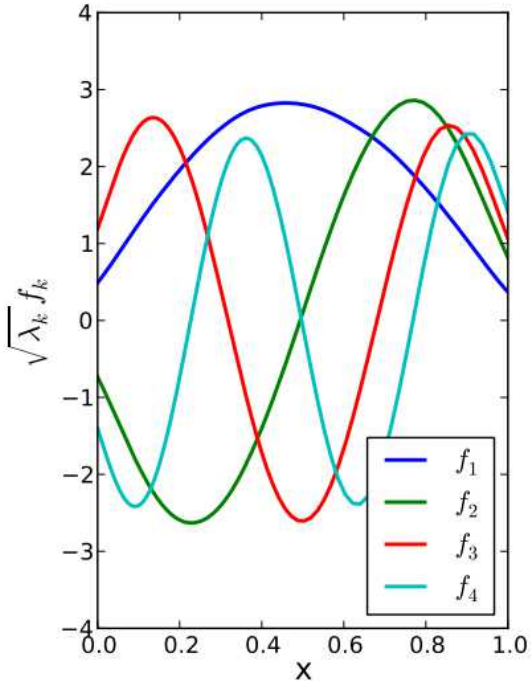
2^9 realizations



2^{13} realizations



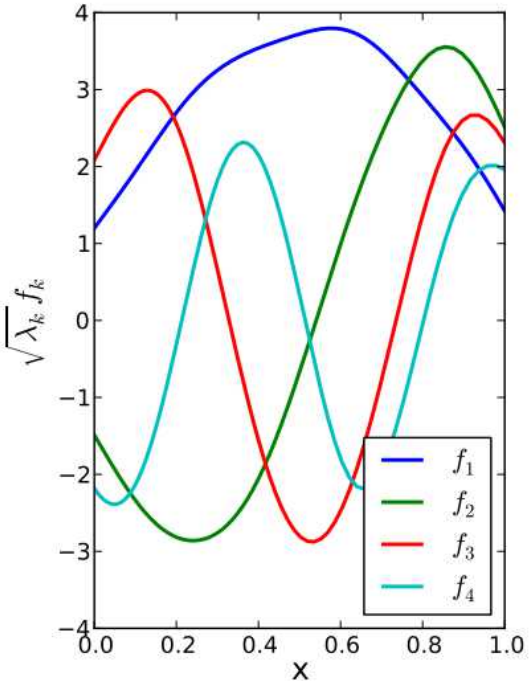
2^{17} realizations



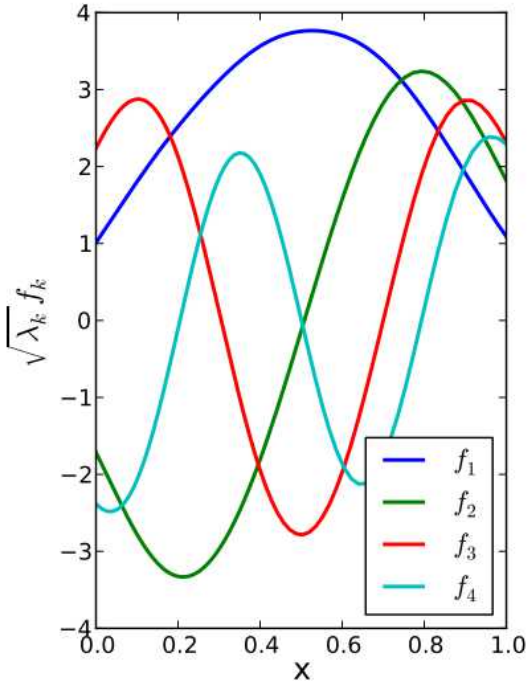
- Eigenmodes of the covariance matrix
  - Data covariance matrix constructed from $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$ Gaussian process realizations
- Higher modes are more oscillatory

1D Gaussian Process: KL Modes for $\delta = 0.20$

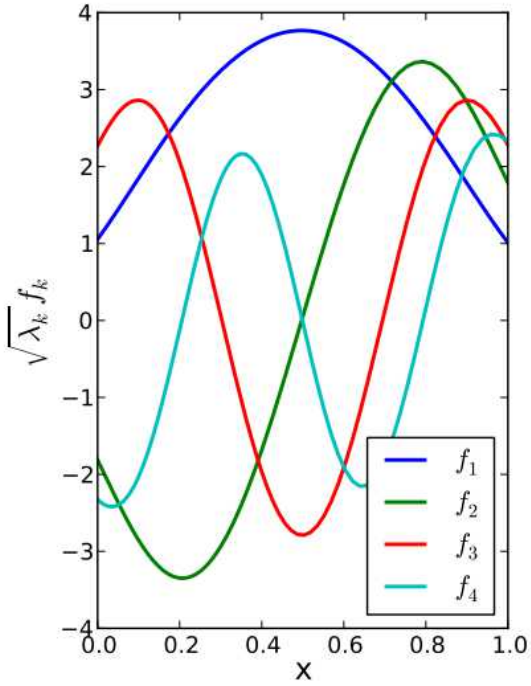
2^9 realizations



2^{13} realizations



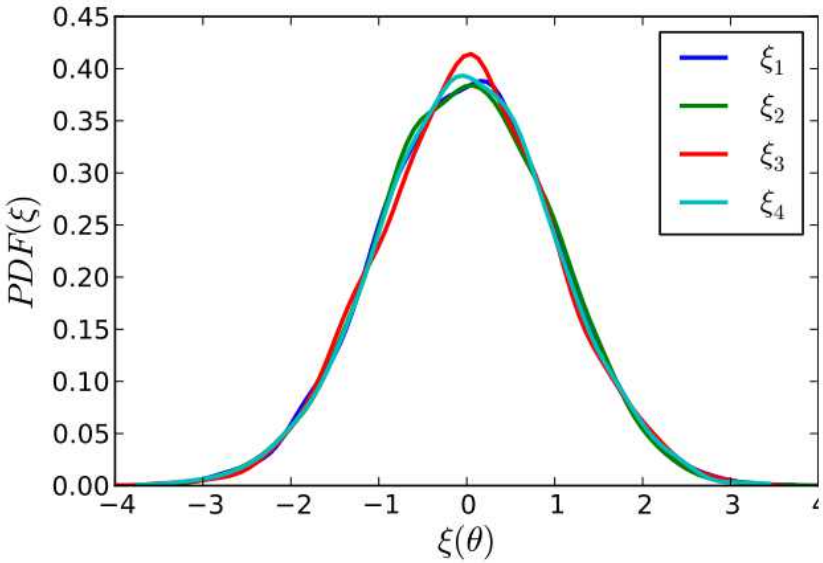
2^{17} realizations



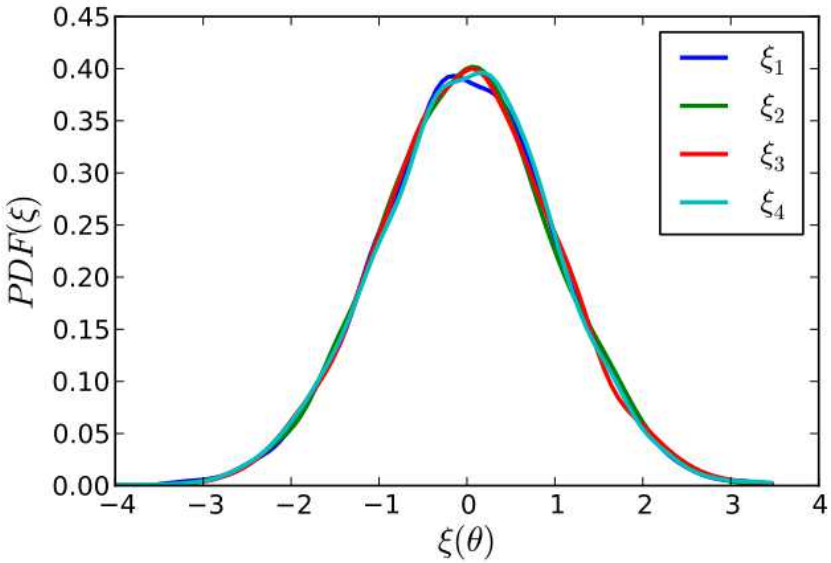
- Eigenmodes of the covariance matrix
  - Data covariance matrix constructed from $2^9 = 512$, $2^{13} = 8192$, and $2^{17} = 131072$ Gaussian process realizations
- Higher modes are more oscillatory

1D Gaussian Process: KL Random Variables

$\delta = 0.1$

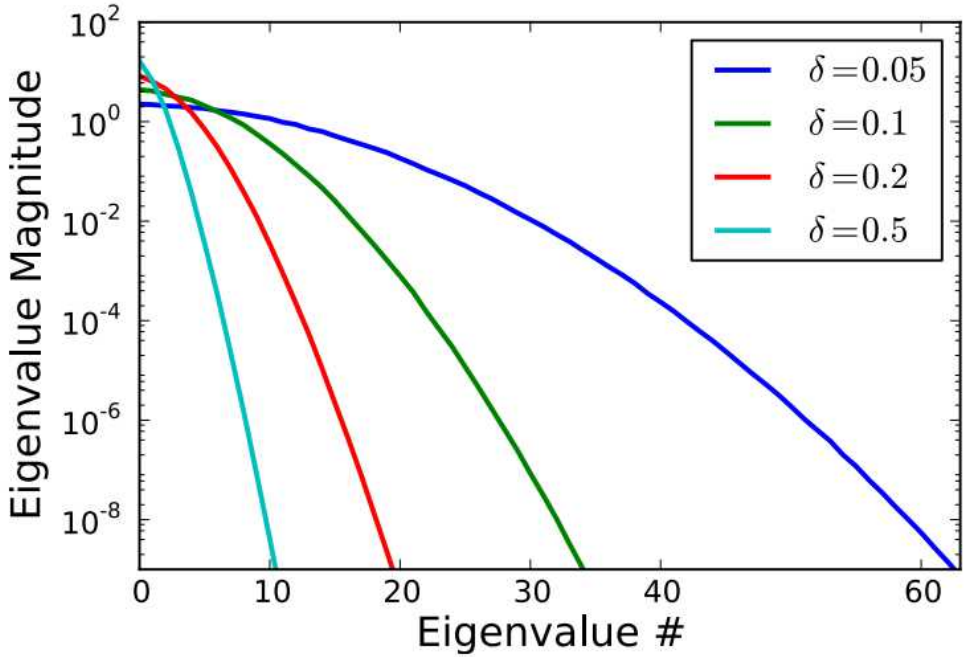


$\delta = 0.2$



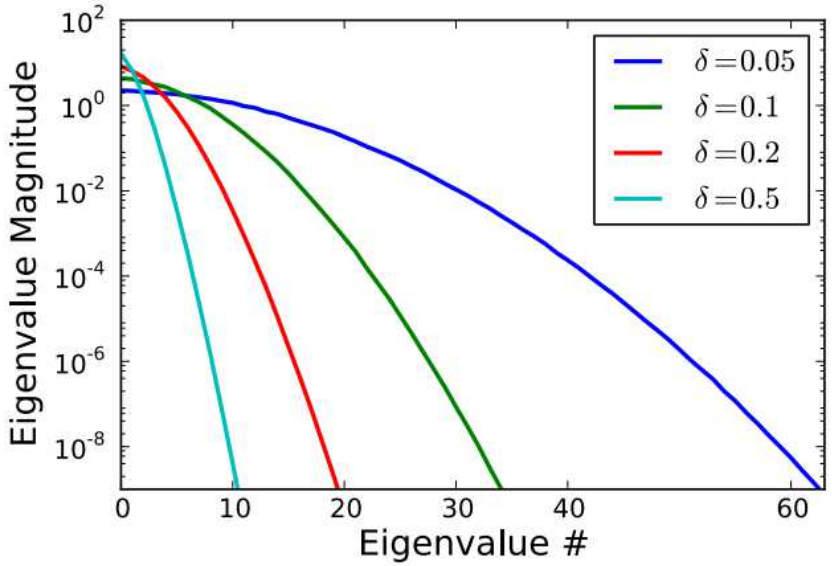
- Random variables obtained by projecting realizations onto KL modes
- Uncorrelated by construction
  - Also independent due to nature of Gaussian Process

1D Gaussian Process: Eigenvalue Spectrum



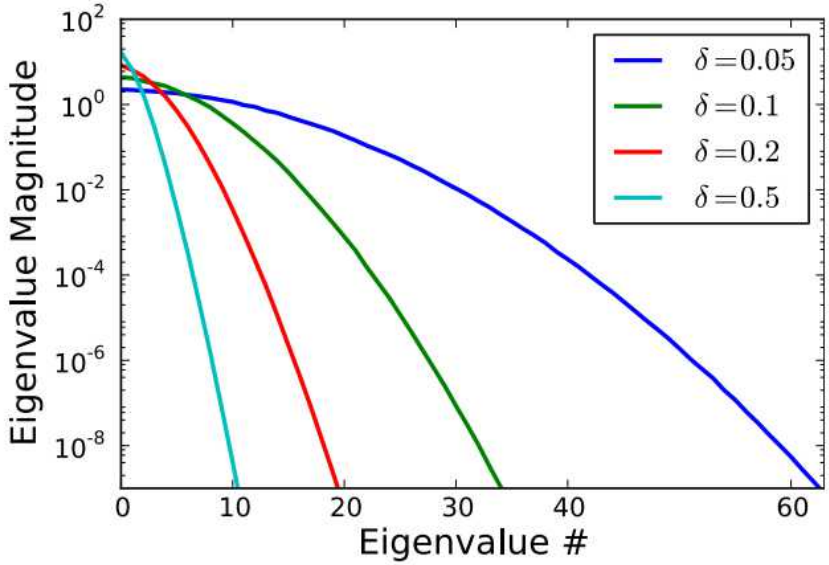
- Eigenvalue spectrum decays more slowly as correlation length decreases
  - More oscillatory modes needed to represent fluctuations in x
- KL expansion generally is truncated after enough modes are included to capture a specified fraction of the total variance

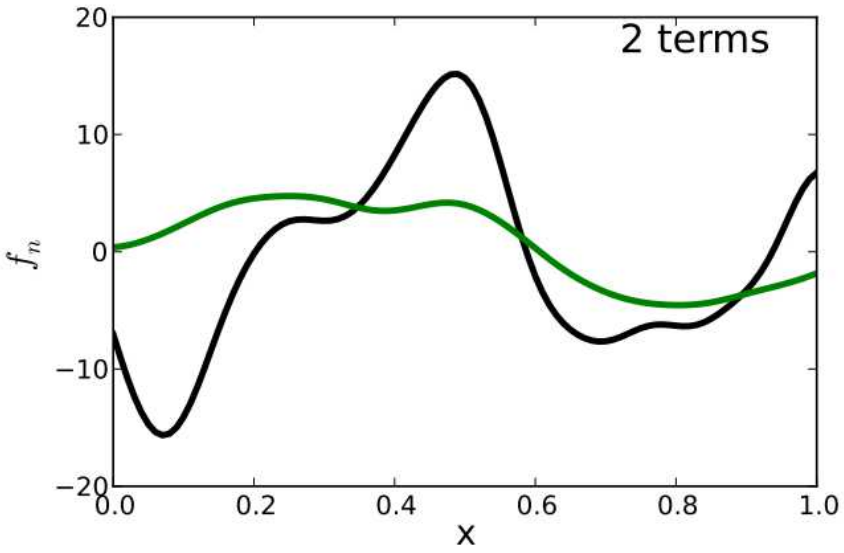
1D Gaussian Process: Reconstructed Realizations



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

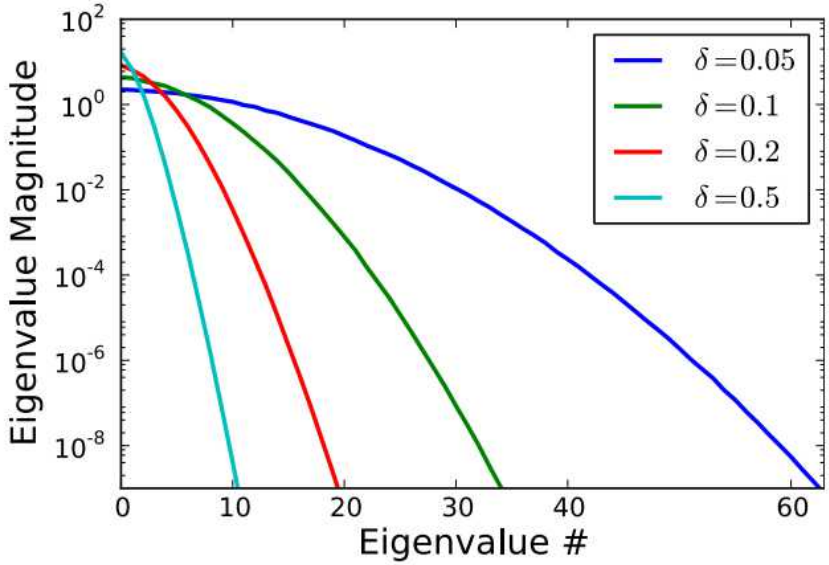
1D Gaussian Process: Reconstructed Realizations

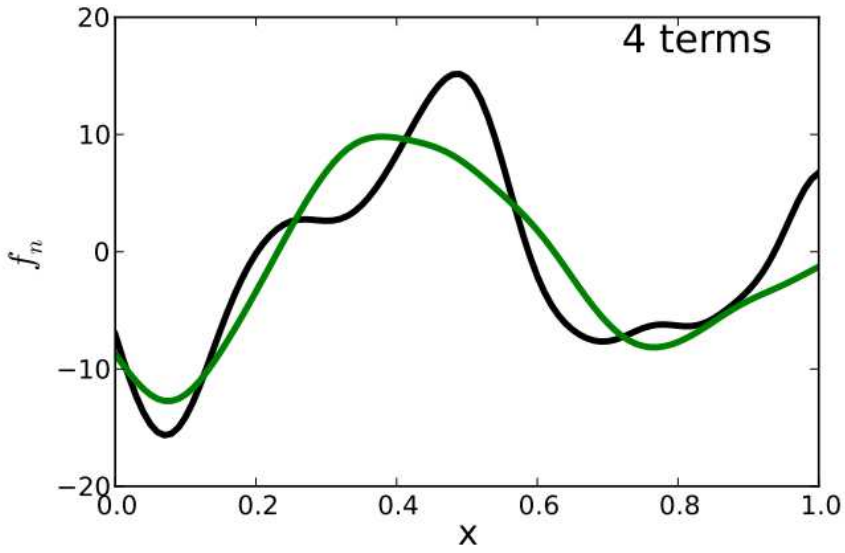




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

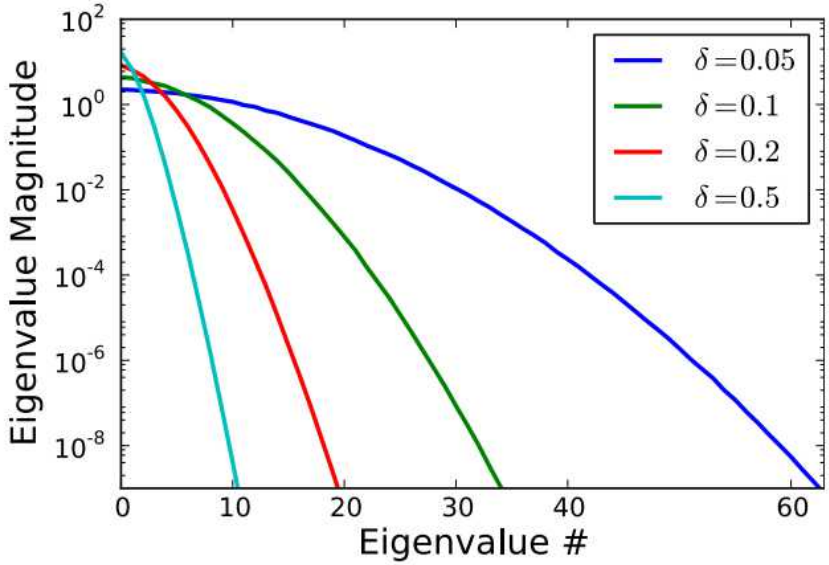
1D Gaussian Process: Reconstructed Realizations

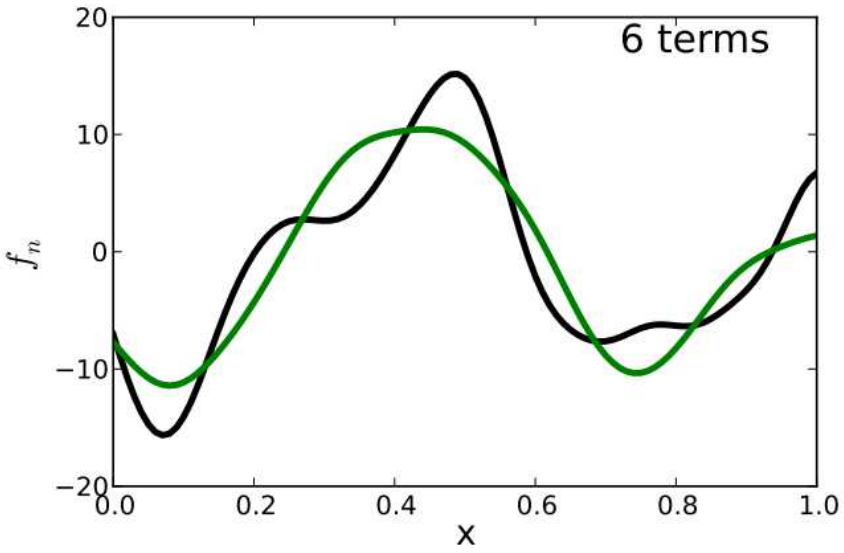




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

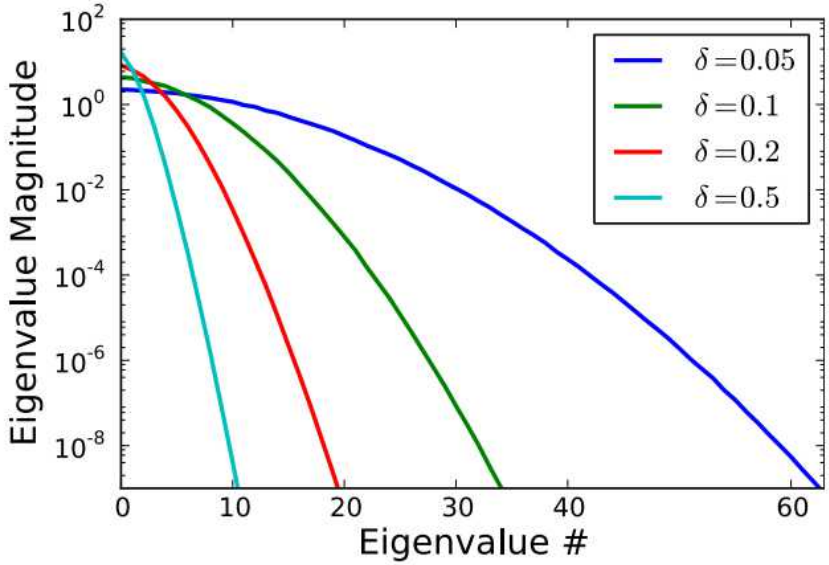
1D Gaussian Process: Reconstructed Realizations

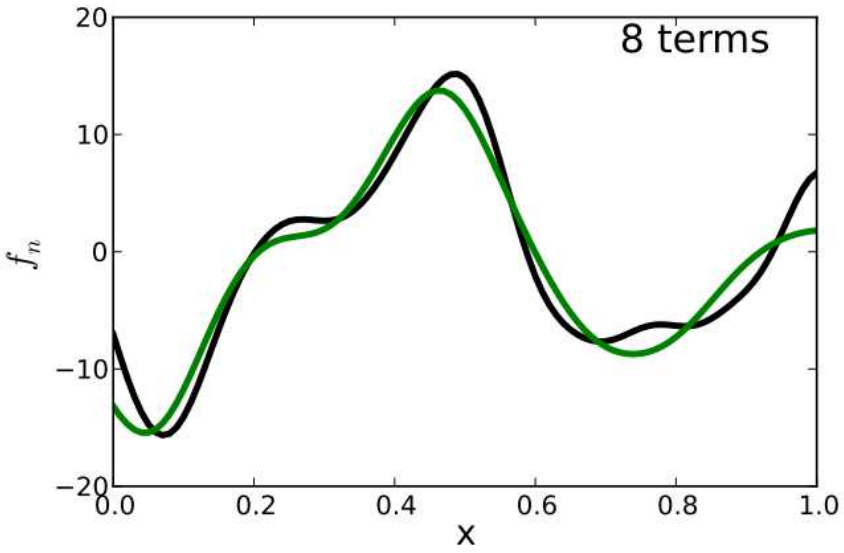




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

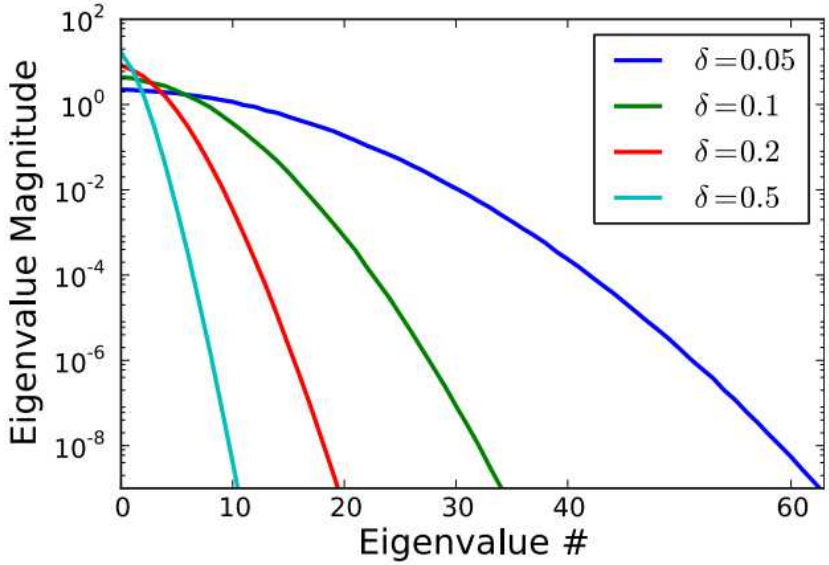
1D Gaussian Process: Reconstructed Realizations

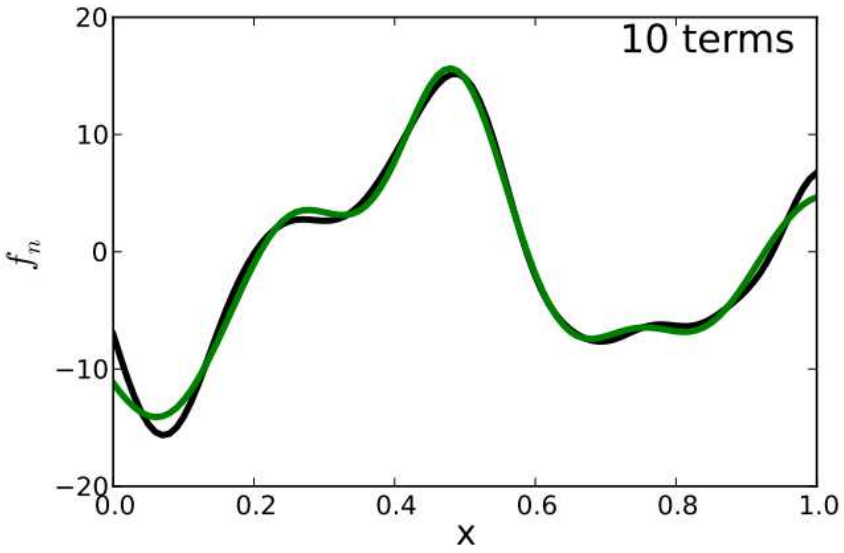




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

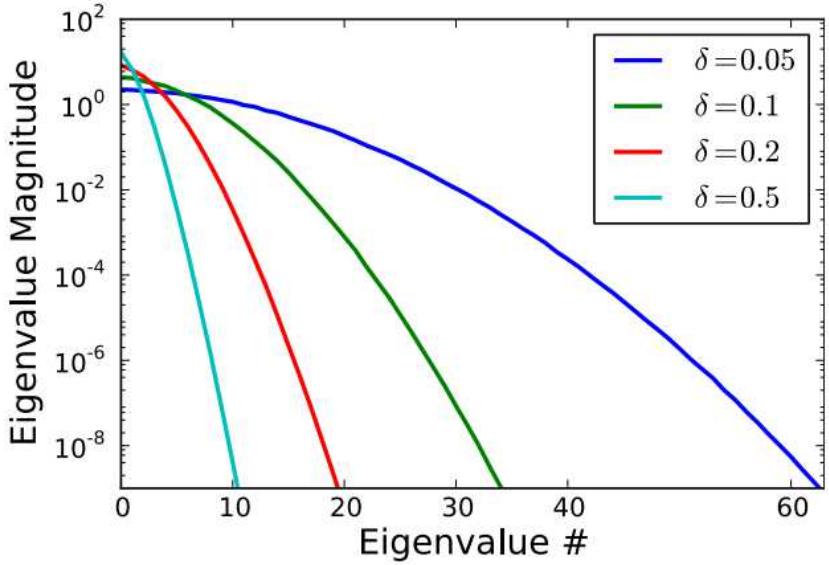
1D Gaussian Process: Reconstructed Realizations

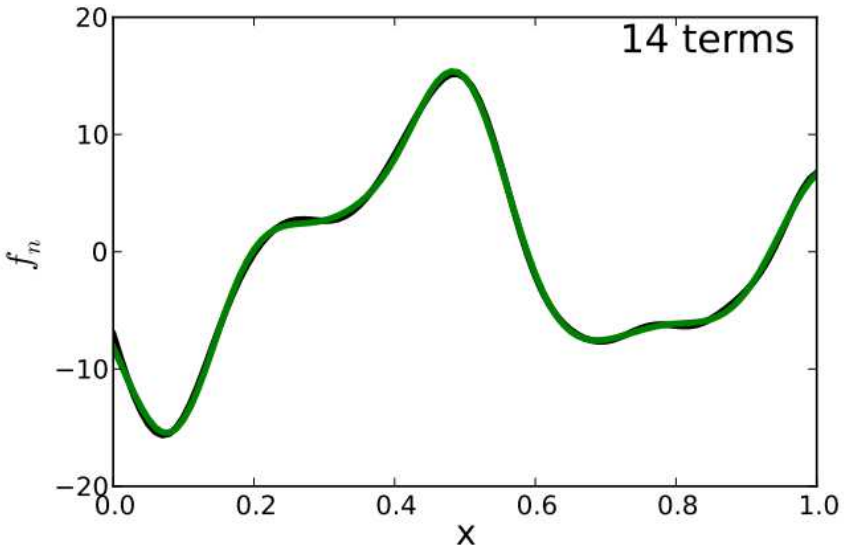




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

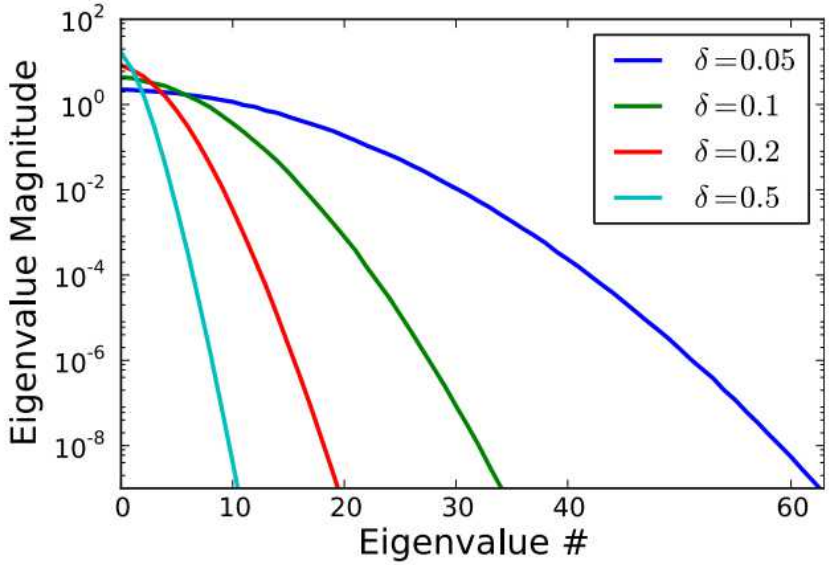
1D Gaussian Process: Reconstructed Realizations

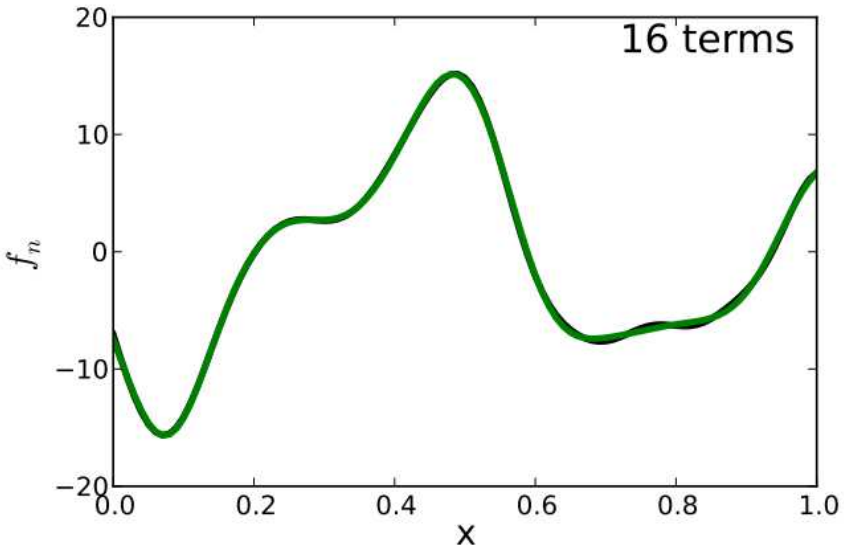




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

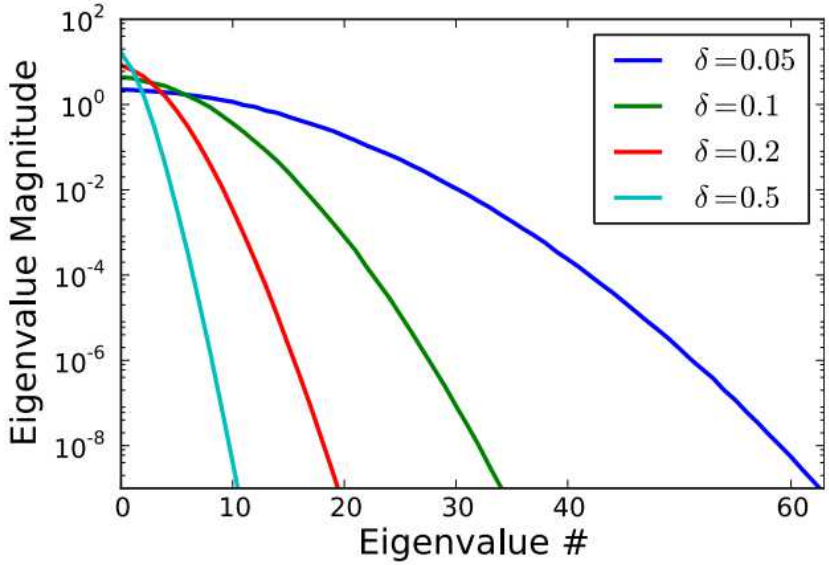
1D Gaussian Process: Reconstructed Realizations

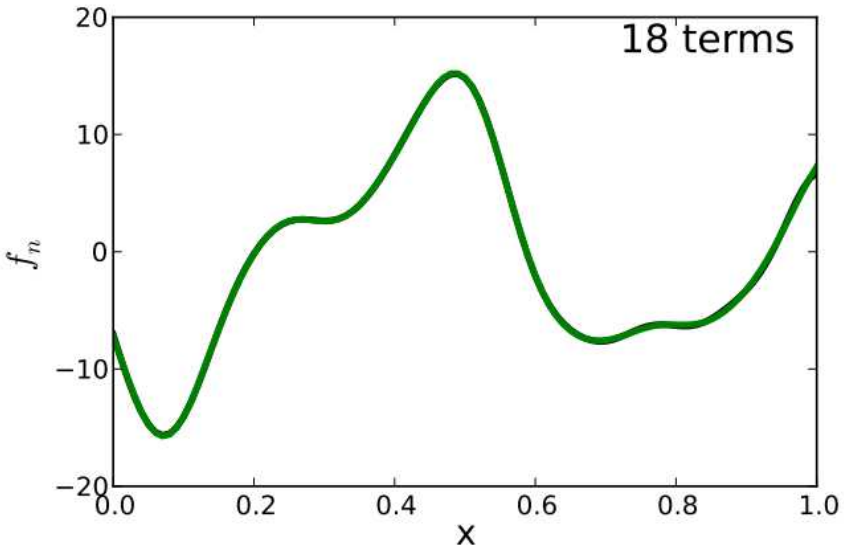




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

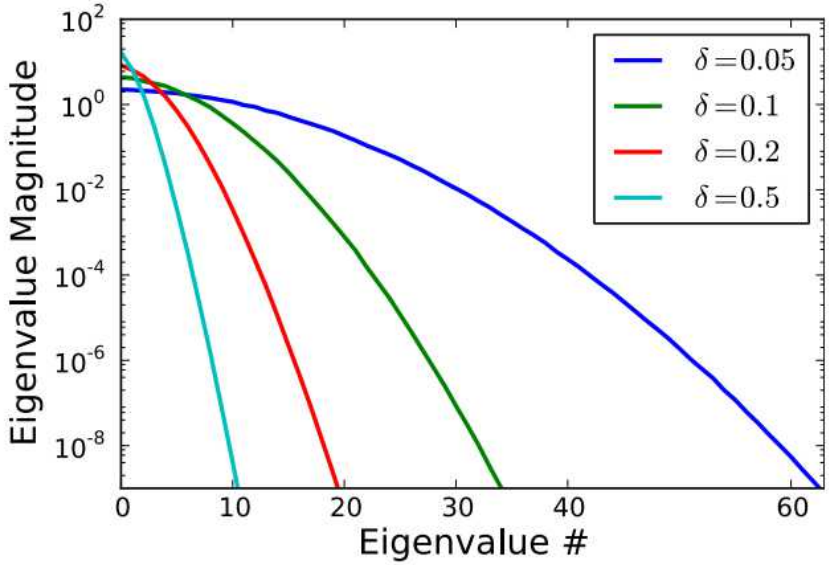
1D Gaussian Process: Reconstructed Realizations

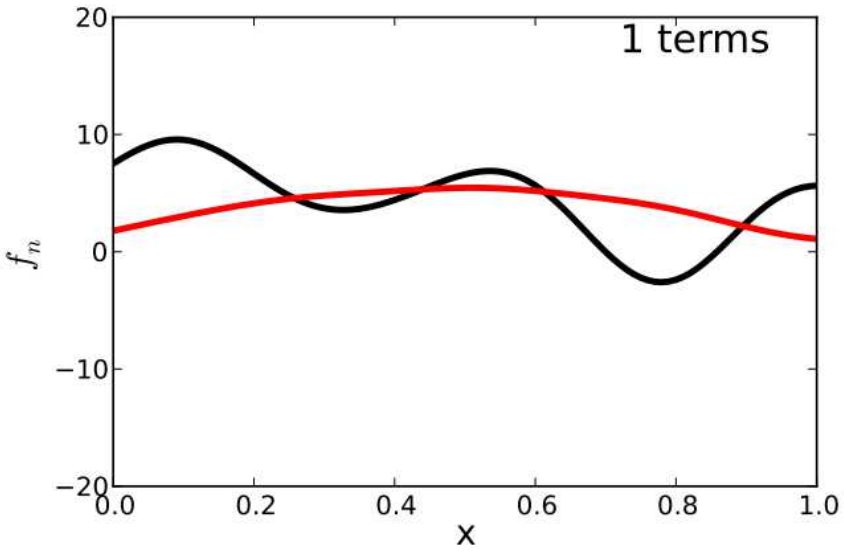




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

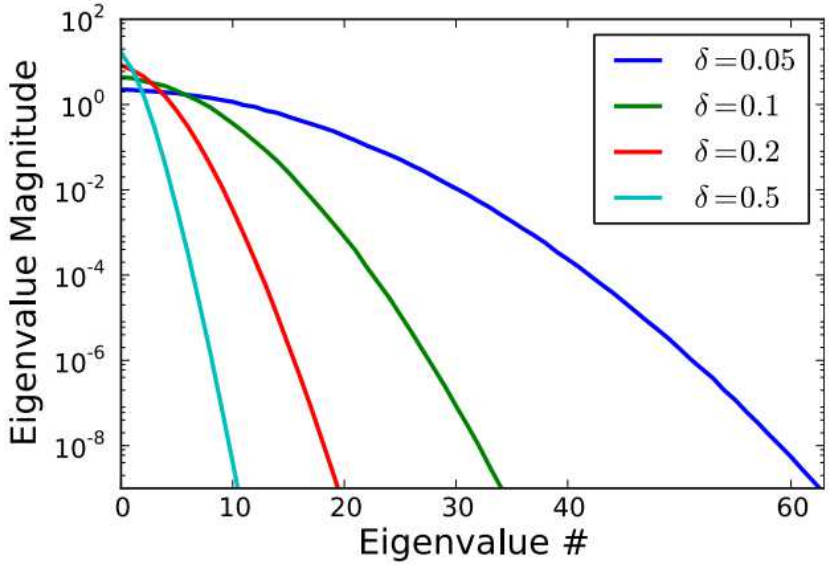
1D Gaussian Process: Reconstructed Realizations

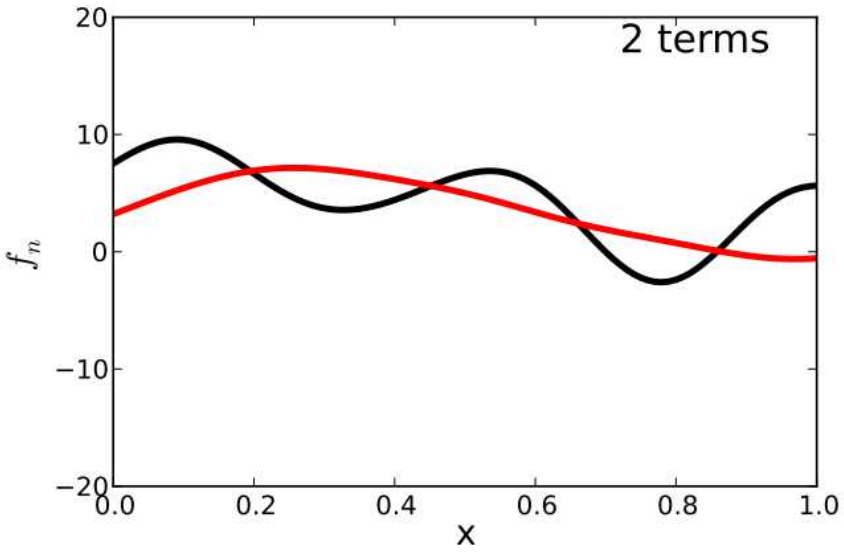




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

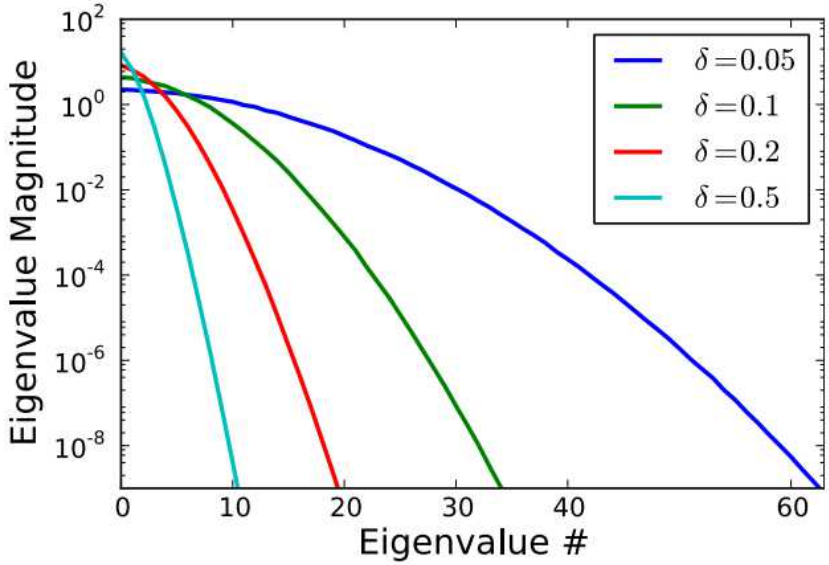
1D Gaussian Process: Reconstructed Realizations

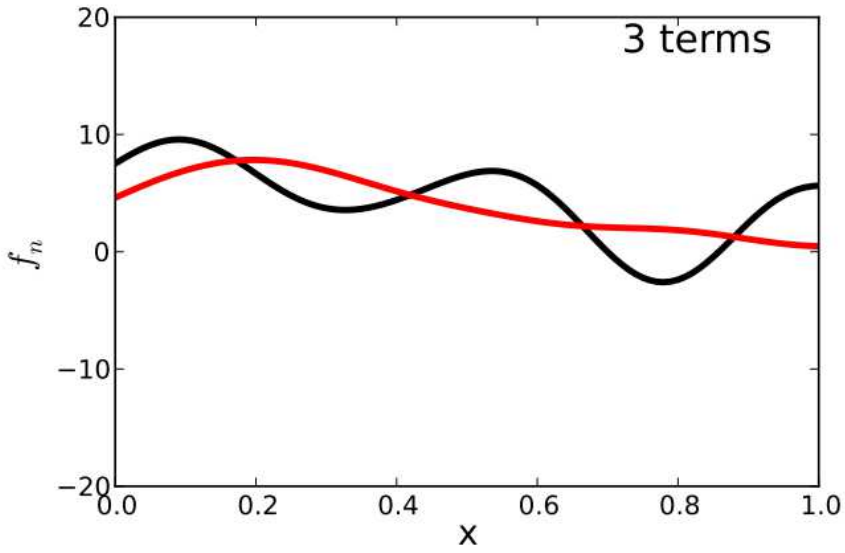




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

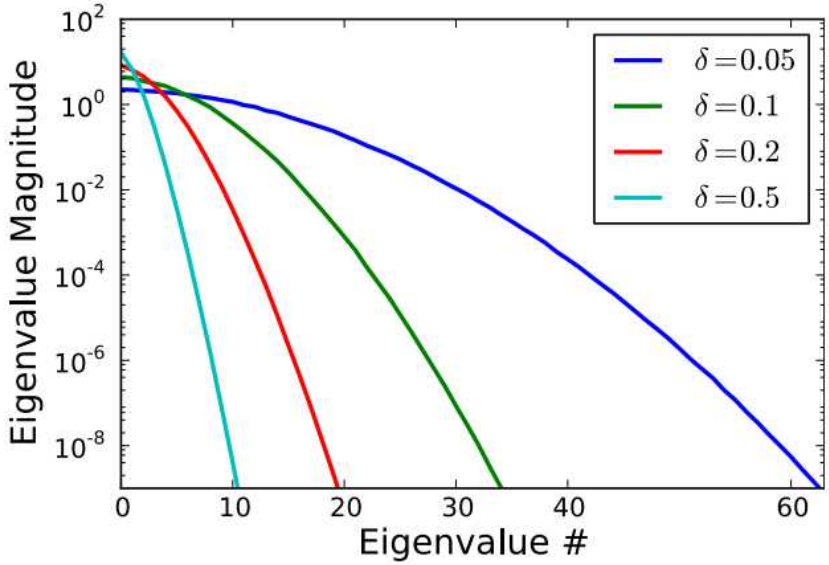
1D Gaussian Process: Reconstructed Realizations

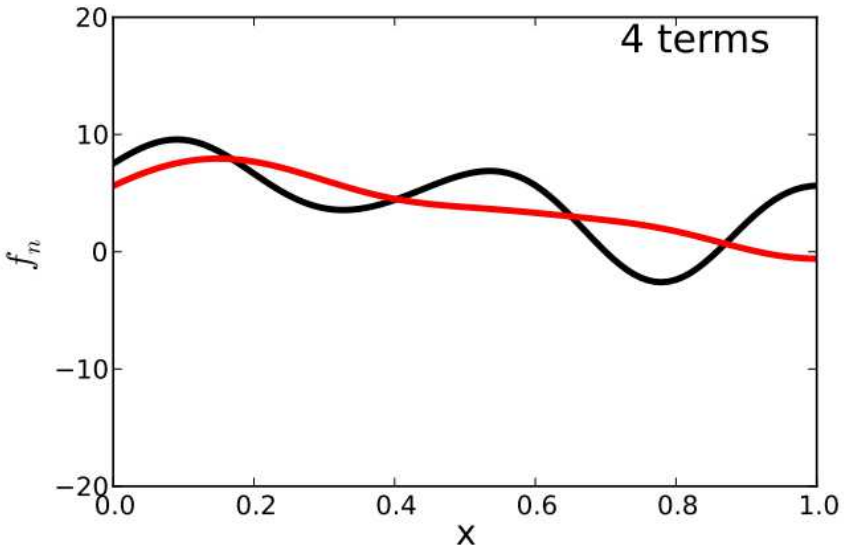




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

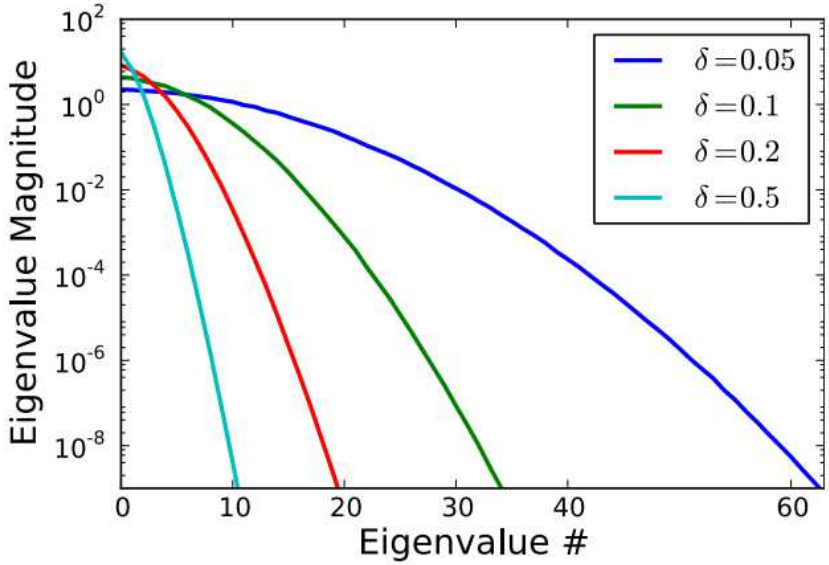
1D Gaussian Process: Reconstructed Realizations

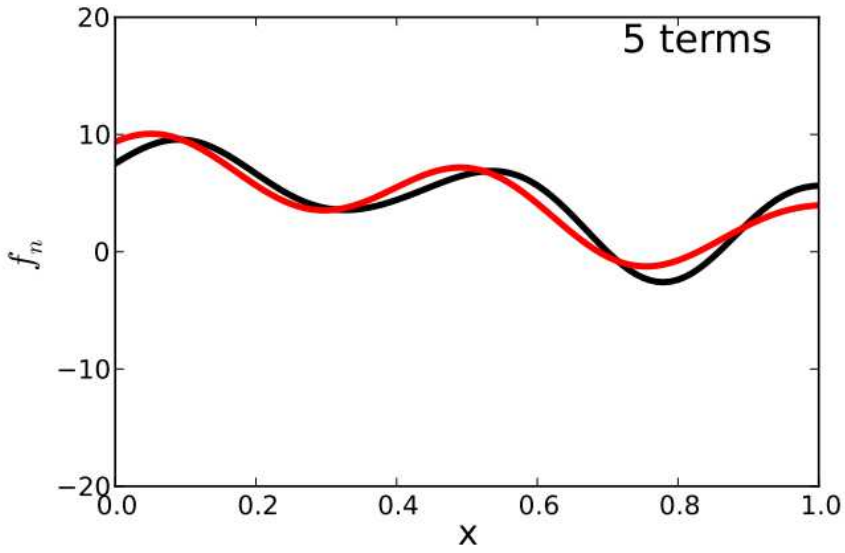




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

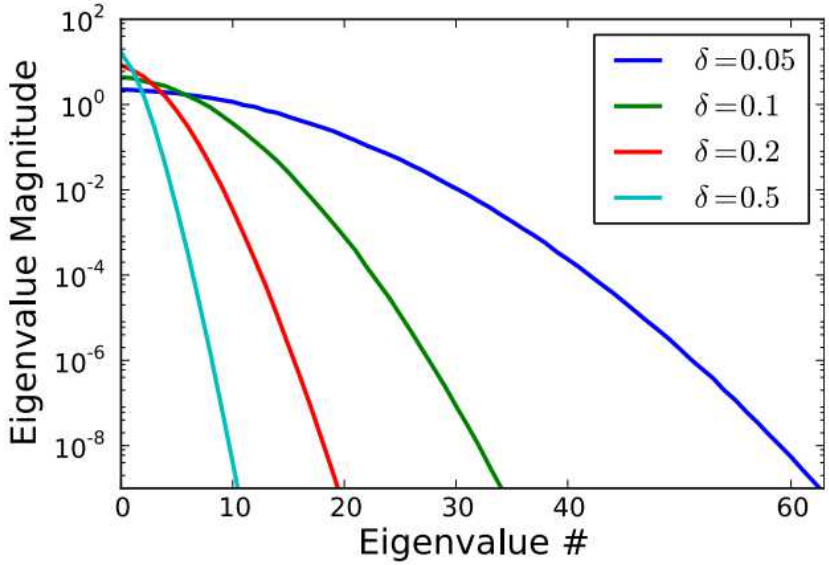
1D Gaussian Process: Reconstructed Realizations

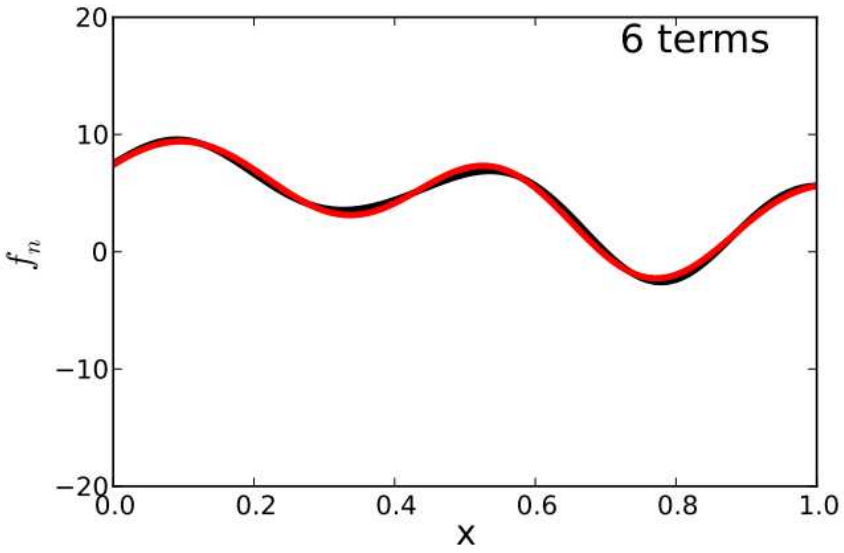




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

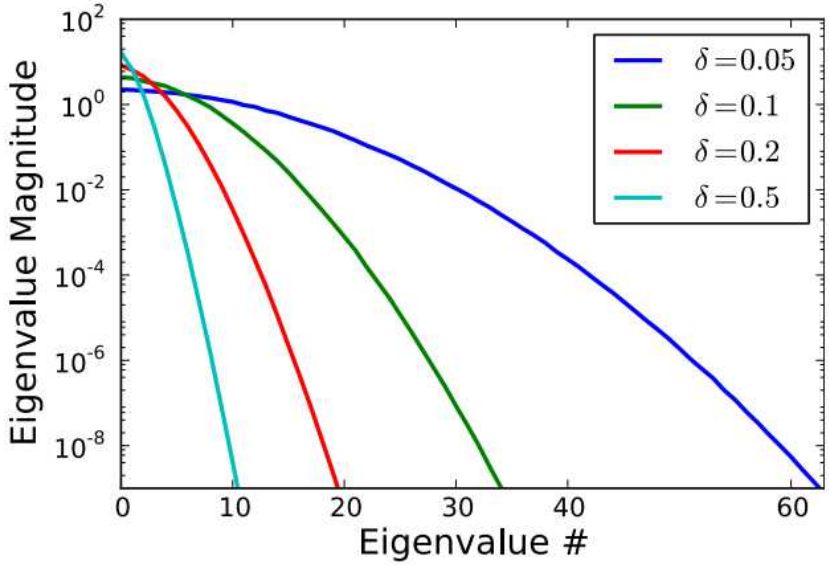
1D Gaussian Process: Reconstructed Realizations

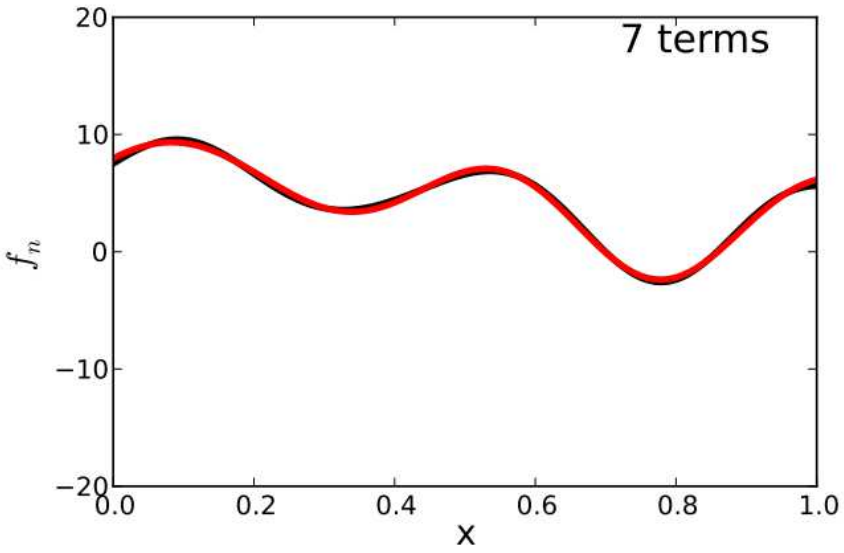




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

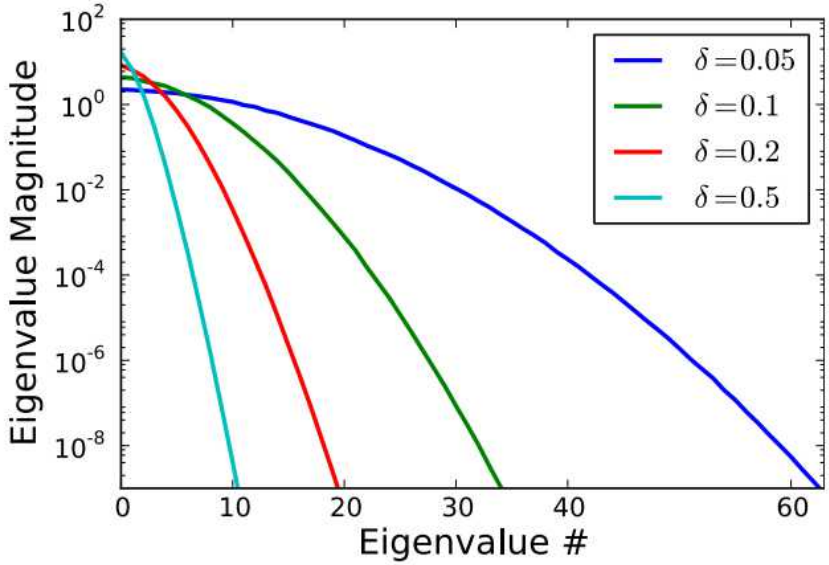
1D Gaussian Process: Reconstructed Realizations

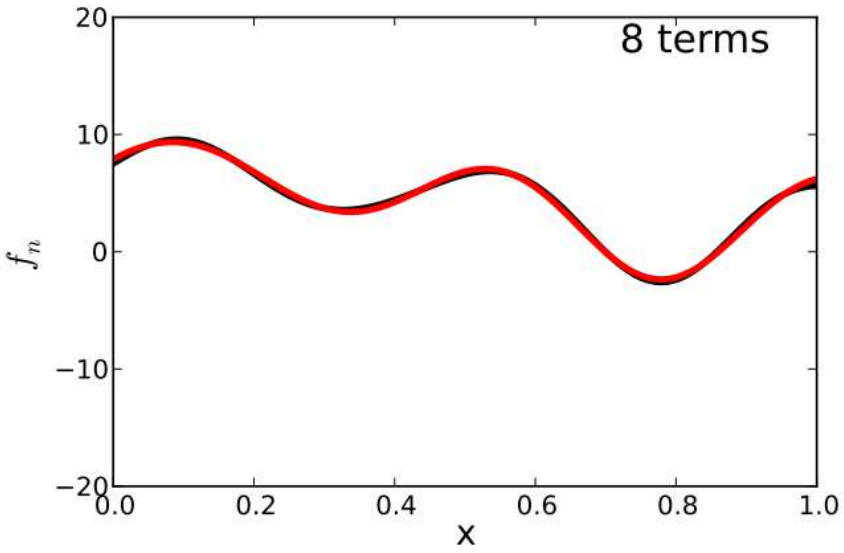




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

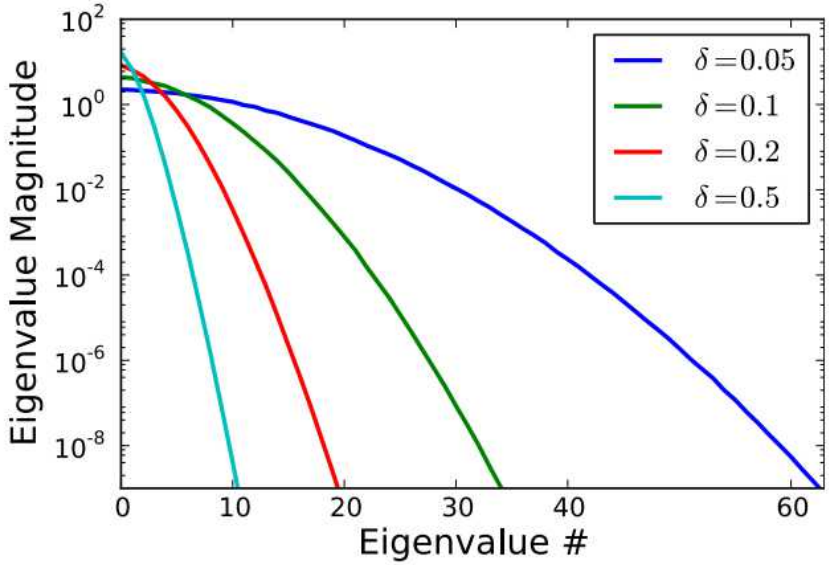
1D Gaussian Process: Reconstructed Realizations

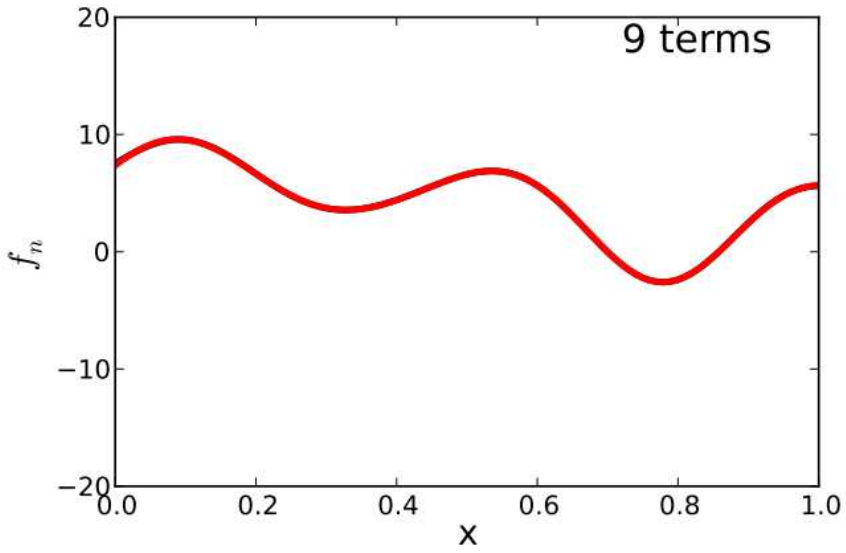




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

1D Gaussian Process: Reconstructed Realizations

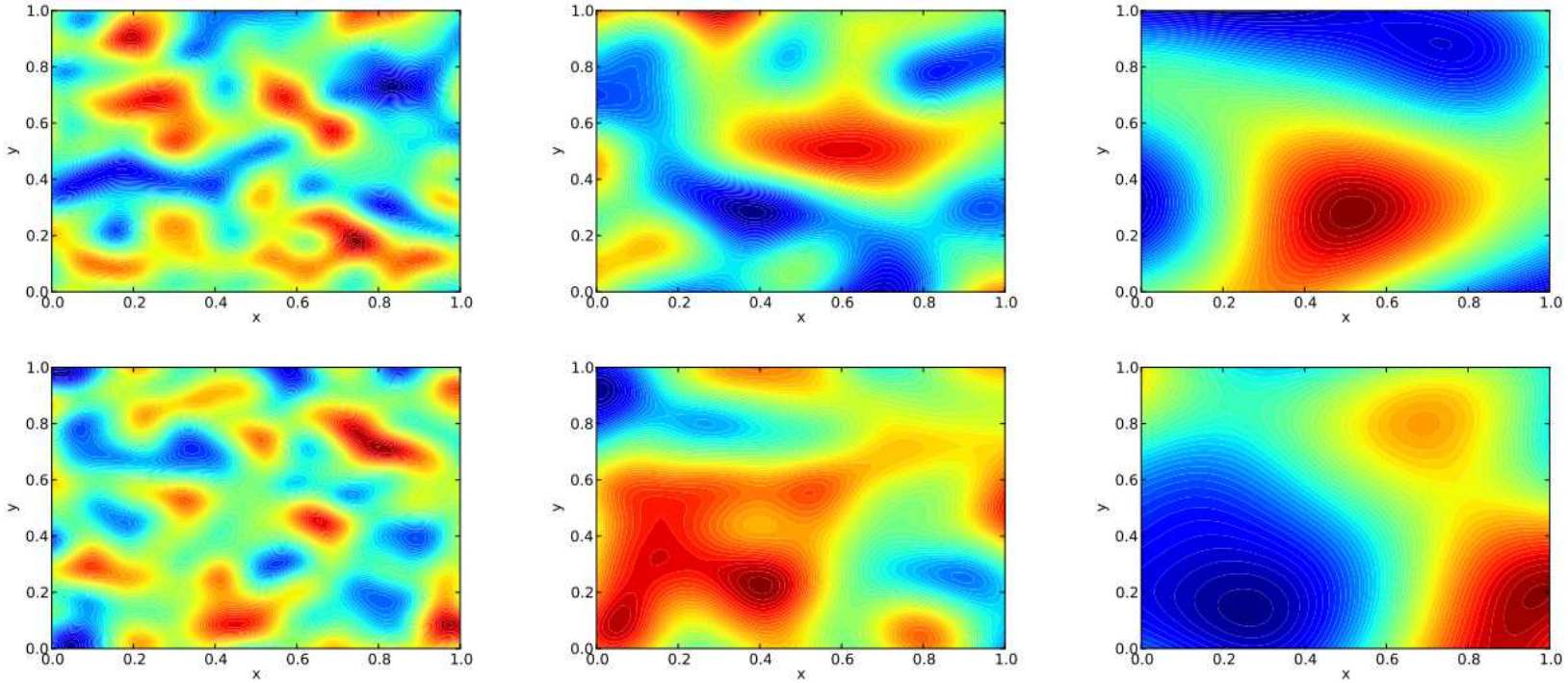




- Large scale features can be resolved with small number of modes
- Smaller scale features require higher modes

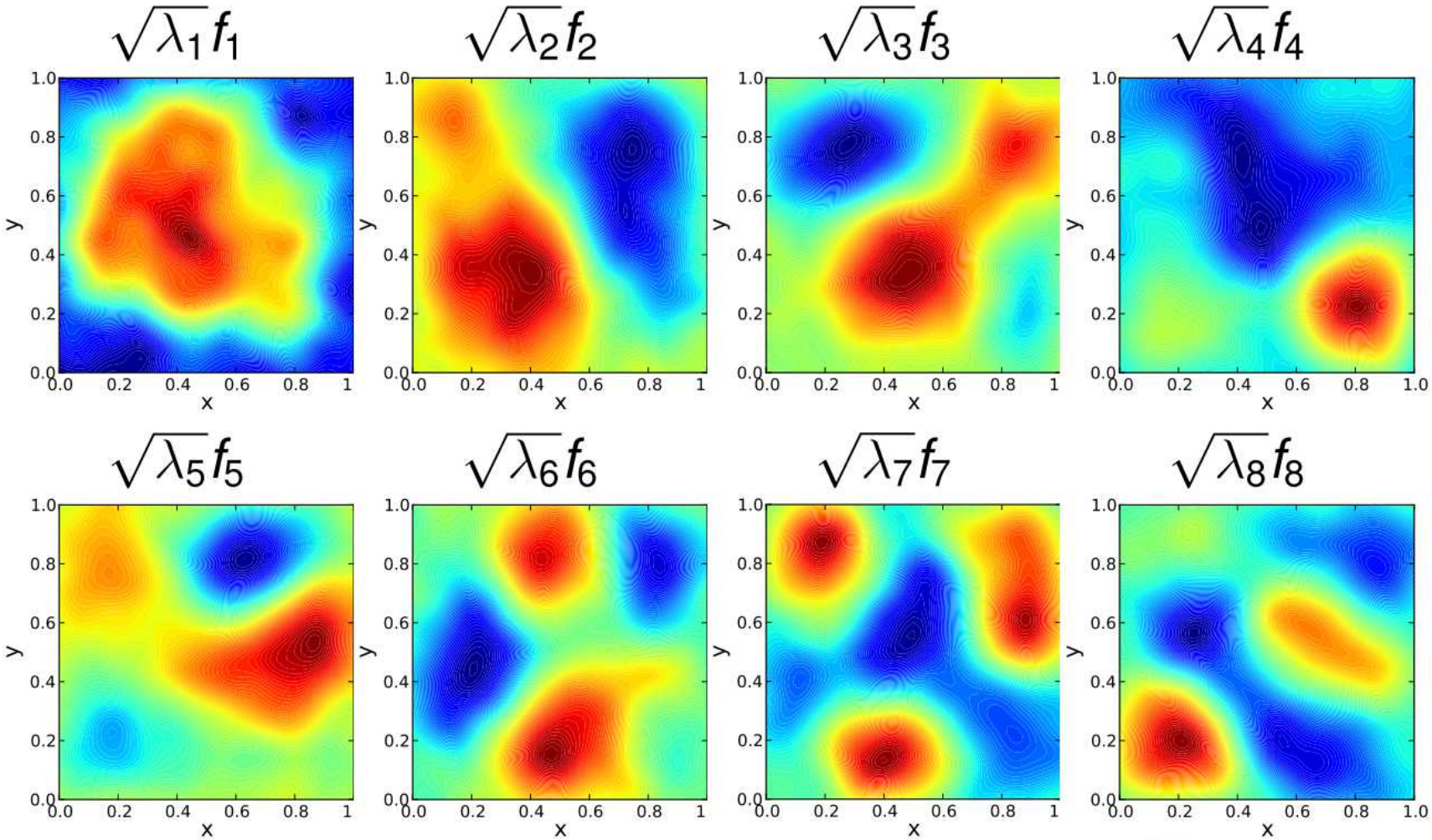
KL of 2D Gaussian Process

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



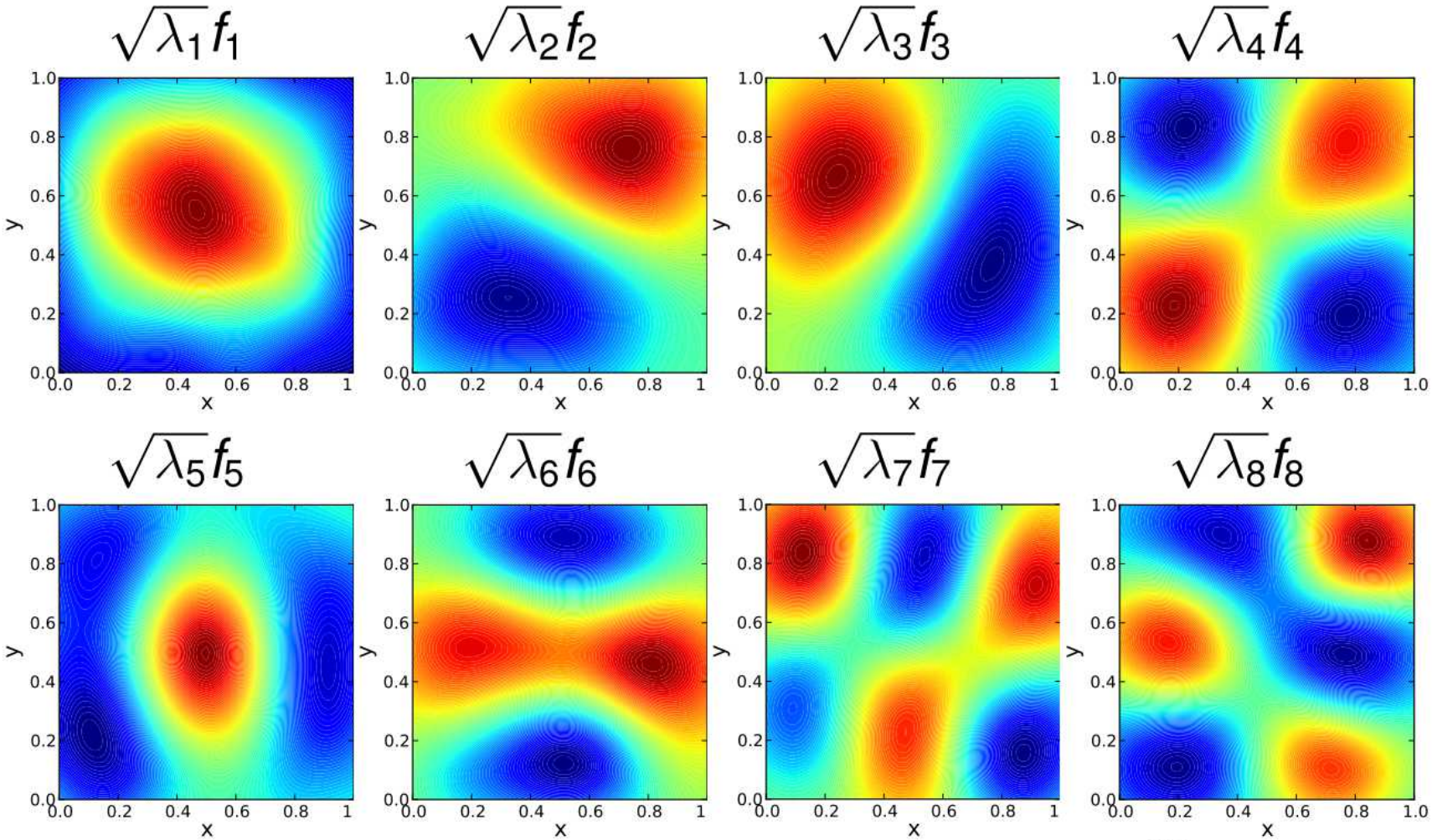
- 2D Gaussian Process with covariance: $Cov(x_1, x_2) = exp(-||x_1 - x_2||^2/\delta^2)$
- Realizations are smoother as covariance length δ increases

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



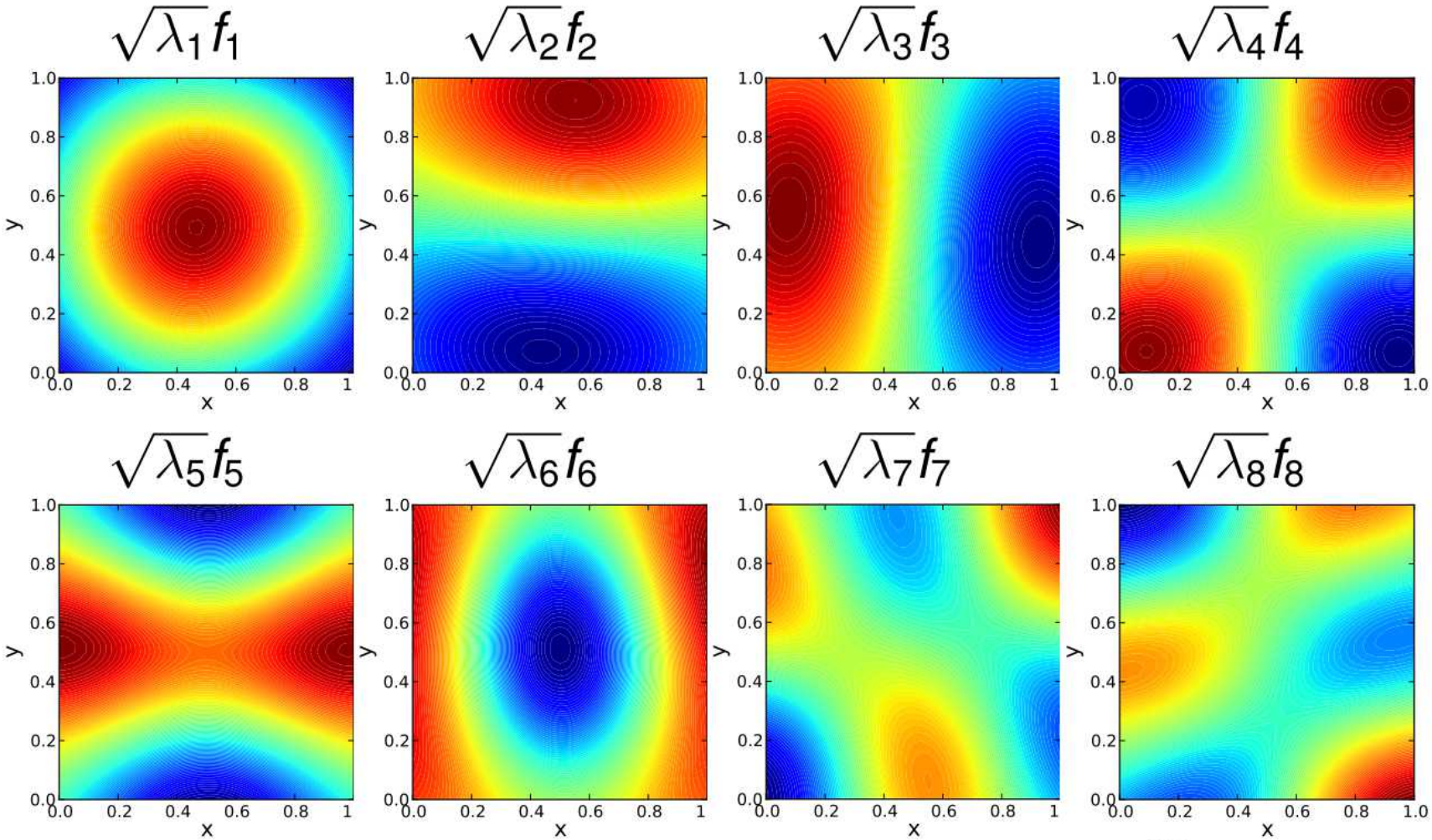
- Data covariance matrix constructed from $2^{12} = 4096$ Gaussian process realizations

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



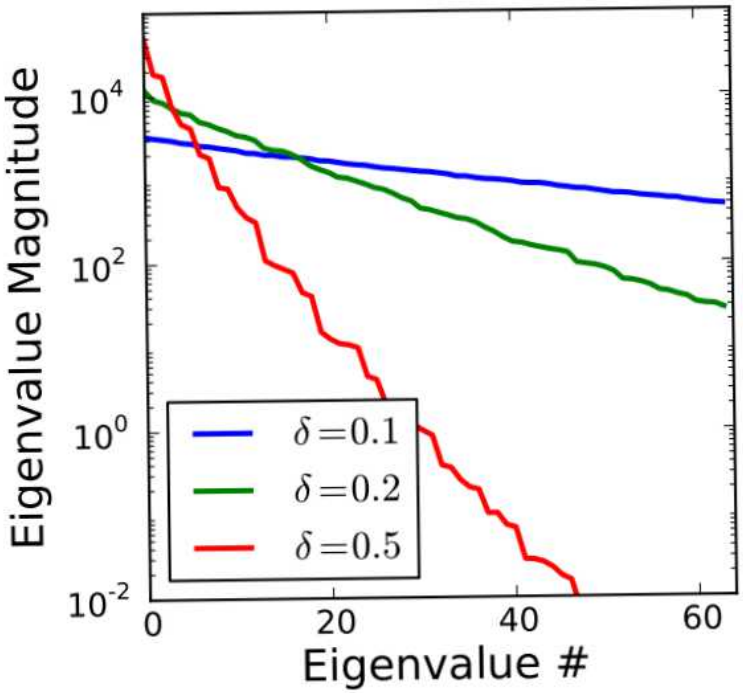
- Data covariance matrix constructed from $2^{12} = 4096$ Gaussian process realizations

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



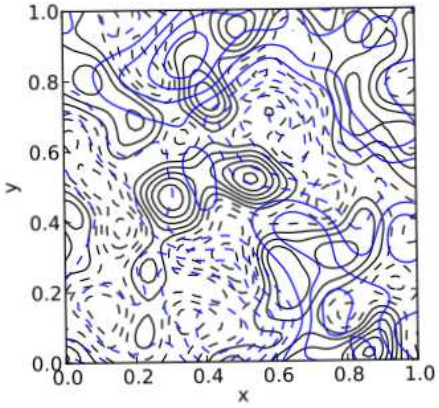
- Data covariance matrix constructed from $2^{12} = 4096$ Gaussian process realizations

2D KL - Eigenvalue Spectrum

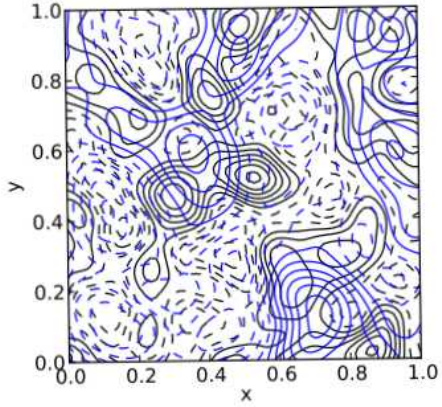


$\delta = 0.1$

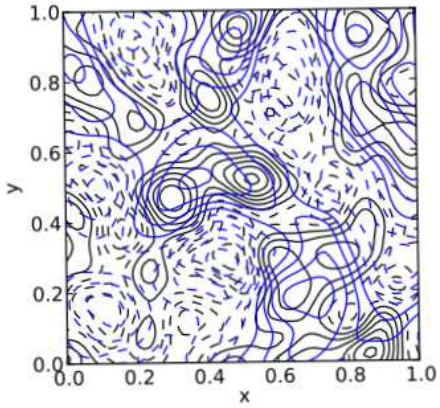
4 terms



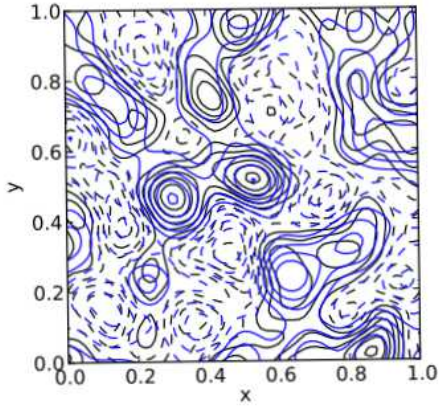
16 terms



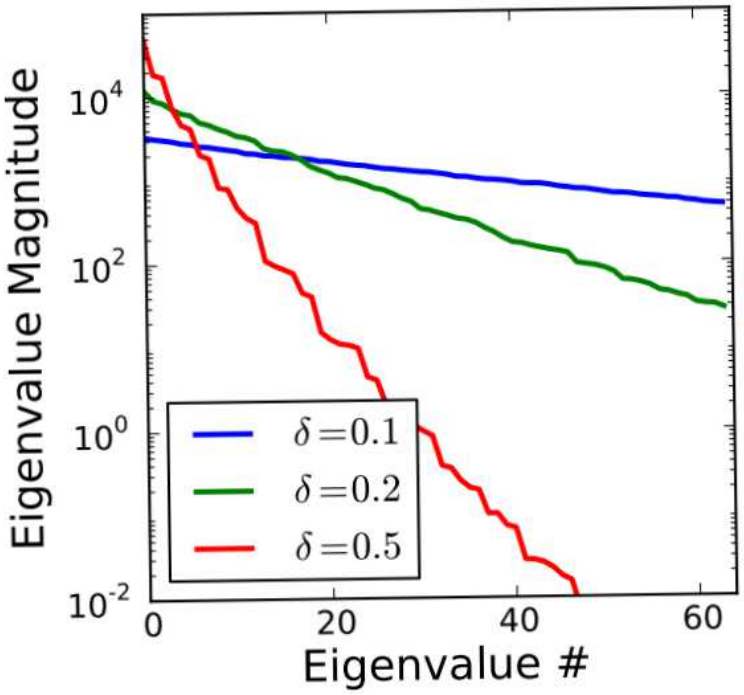
32 terms



64 terms

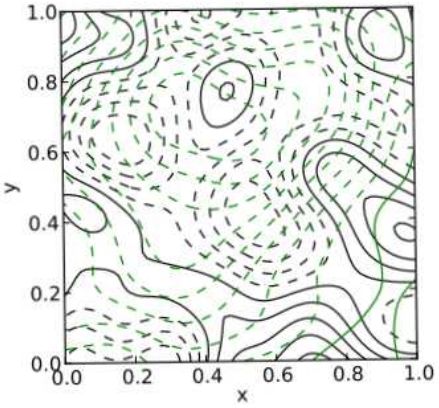


2D KL - Eigenvalue Spectrum

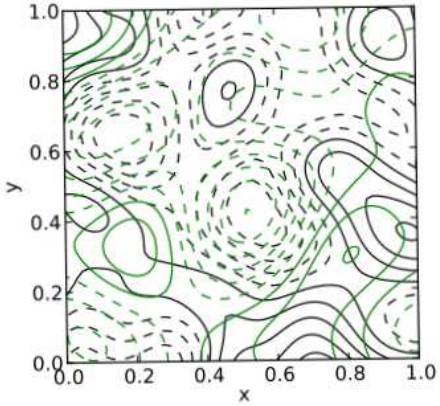


$\delta = 0.2$

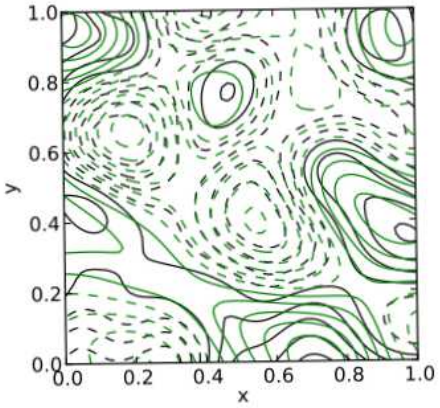
4 terms



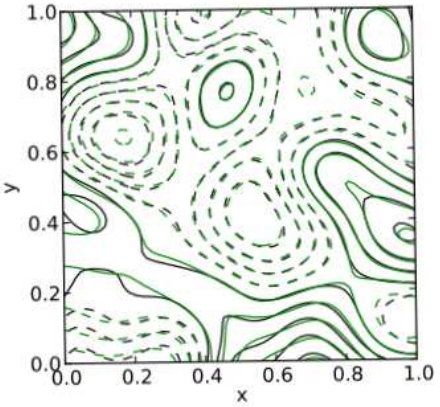
16 terms



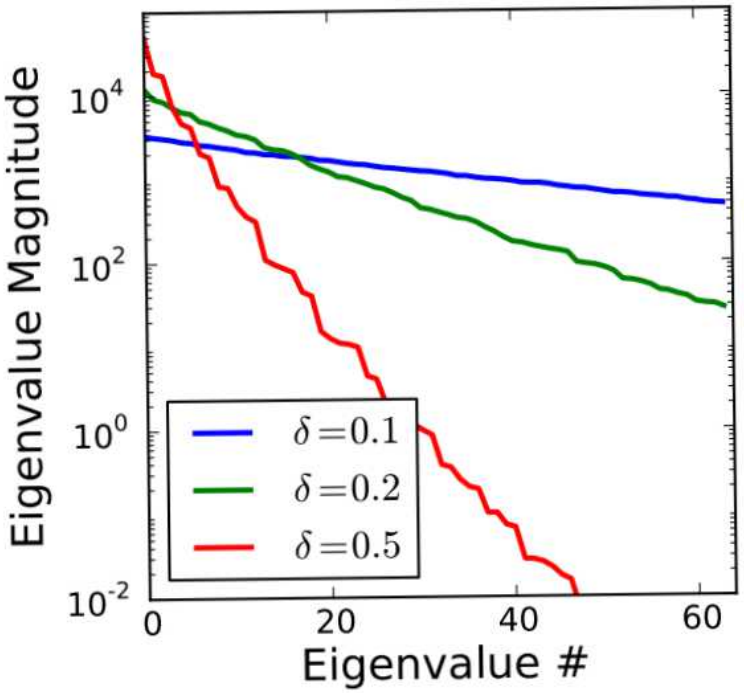
32 terms



64 terms

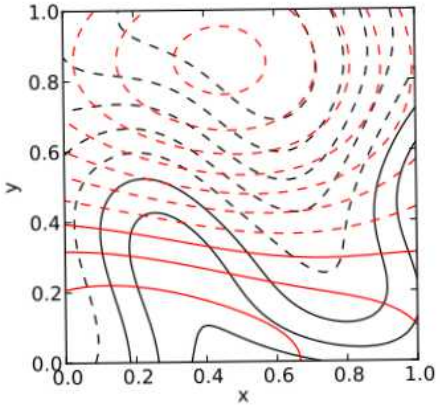


2D KL - Eigenvalue Spectrum

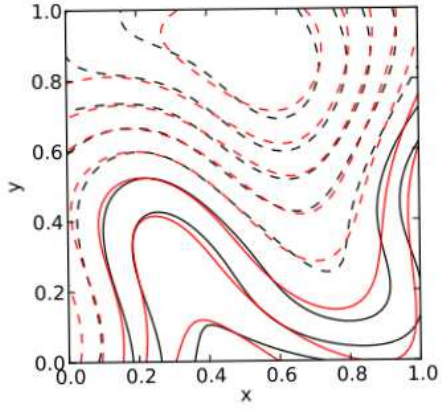


$\delta = 0.5$

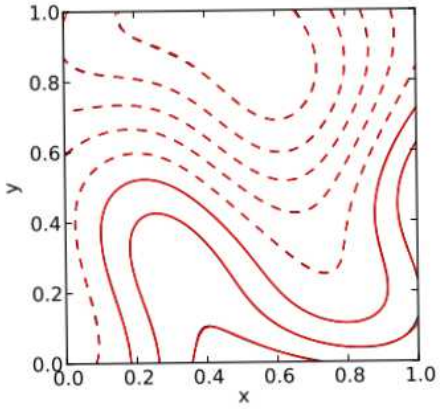
4 terms



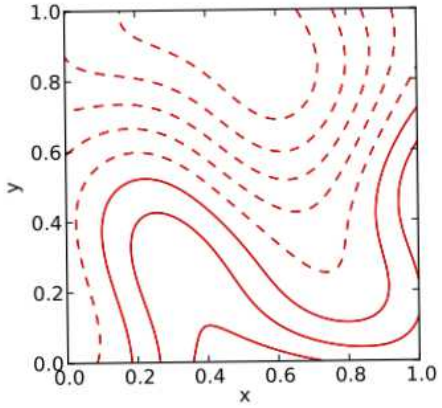
16 terms



32 terms



64 terms



KL: Summary

- The KL expansion allows a compact, finite dimensional representation of random fields
- Dimensionality reduction is strongest if eigenvalue spectrum decays rapidly
  - Long correlation length in random field realizations
  - Relatively narrow range of feature sizes
- A potential drawback is that many samples are needed to compute the covariance matrix accurately
  - 1000's of realizations were needed to get smooth eigenmodes in 2D examples shown earlier
- Often, a covariance function is assumed and the associated eigenmodes are used to parametrize an unknown random field

Further Reading

References on Polynomial Chaos Fundamentals

- N. Wiener, “Homogeneous Chaos", *American Journal of Mathematics*, 60:4, pp. 897-936, 1938.
- R. Ghanem and P. Spanos, “Stochastic Finite Elements: a Spectral Approach", Springer, 1991.
- 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.

References on PCE-Based Forward Uncertainty Propagation

- 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.
- 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 Processes", *SIAM J. Sci. Comp.*, 26:2, 2004.
- B. J. Debusschere, H. N. Najm, A. Matta, O. M. Knio, R. G. Ghanem, and O. P. Le Maître, “Protein labeling reactions in electrochemical microchannel flow: Numerical simulation and uncertainty propagation,” Phys. Fluids, vol. 15, no. 8, p. 2238, 2003.

Quadrature References

- P. R. Conrad and Y. M. Marzouk, “Adaptive Smolyak Pseudospectral Approximations," SIAM J. Sci. Comput., vol. 35, no. 6, pp. A2643–A2670, 2013.
- J. Winokur, P. Conrad, I. Sraj, O. Knio, A. Srinivasan, W. C. Thacker, Y. Marzouk, and M. Iskandarani, “A priori testing of sparse adaptive polynomial chaos expansions using an ocean general circulation model database," Comput Geosci, 2013.
- T. Gerstner and M. Griebel, “Numerical integration using sparse grids," Numer Algor, vol. 18, pp. 209–232, Jan. 1998.

Sensitivity Analysis – Characterization

- A. Saltelli and S. Tarantola and F. Campolongo and M. Ratto, “Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models”. John Wiley & Sons, 2004.
- M. Rosenblatt, “Remarks on a Multivariate Transformation", *Ann. Math. Statist.*, 23:3, pp. 470-472, 1952.

Bayesian Inference

- R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal, “Markov chain monte carlo in practice: A roundtable discussion," The American Statistician, vol. 52, no. 2, pp. 93–100, 1998.
- D.S. Sivia, “Data Analysis: A Bayesian Tutorial,” Oxford Science, 1996.