Probabilistic Methods for Uncertainty Quantification in Computational Models

Khachik Sargsyan

Sandia National Laboratories
Livermore, CA

Schlumberger,
Inversion, Optimization & Uncertainty
SIG Webinar
April 19, 2016

Acknowledgements

H. Najm, B. Debusschere, C. Safta, X. Huan — Sandia National Laboratories, CA

R. Ghanem — USC
O. Knio — Duke
O. Le Maître — LIMSI-CNRS, Paris
Y. Marzouk — MIT
D. Ricciuto, P. Thornton – Oak Ridge National Lab

This work was supported by:

- DOE Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC)
- DOE, Biological and Environmental Research (BER)
- DOD, DARPA Enabling Quantification of Uncertainty in Physical Systems (EQUiPS) program

Sandia National Laboratories is a multi-program laboratory 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.

Outline

Background

- Ph.D. from U. of Michigan, Applied Math,, 2007
- 2007-present: working in UQ at Sandia National Labs

- US Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR)
- QUEST Institute (Quantification of Uncertainty in Extreme Scale CompuTations)
    - PI: Habib Najm
    - www.quest-scidac.org
    - Advanced UQ methods development
    - Reach out to application community
- SNL-CA: 5-10 staff members, $\sim$ 5 postdocs
    - Main research code: UQTk (www.sandia.gov/UQToolkit)
        - Lightweight C++/Python codebase
        - UQTk v3.0 to be posted soon

UQ Software Packages under QUEST

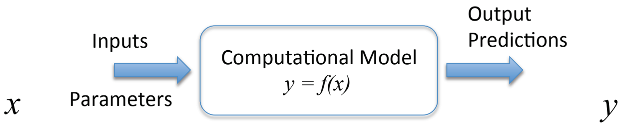
- The SciDAC Institute on Quantification of Uncertainty in Extreme Scale Computations (QUEST) is developing and maintaining a number of packages
  - `http://www.quest-scidac.org/`
  - DAKOTA: `http://dakota.sandia.gov/`
  - UQTk: UQ Toolkit `http://www.sandia.gov/UQToolkit/`
  - GPMSA: Gaussian Process Modeling and Sensitivity Analysis
  - QUESO: Bayesian inference `https://github.com/libqueso/queso/releases`
  - MUQ: MIT Uncertainty Quantification library `https://bitbucket.org/mituq/muq`
- Many packages are becoming available outside QUEST (UQLab, OpenTurns, SmartUQ, ChaosPy,...)

Uncertainty Quantification Toolkit (UQTk)

- A library of C++ and Python functions for propagation of uncertainty through computational models
- Mainly relies on Polynomial Chaos (PC) expansions for representing random variables and stochastic processes
- Target usage:
  - Rapid prototyping
  - Algorithmic research
  - Tutorials / educational
- Version 2.1 released under the GNU Lesser General Public License
  - C++ Tools for intrusive and non-intrusive UQ
  - Polynomial Chaos
  - Bayesian inference tools (various MCMC types)
  - Regression (polynomial, RBF, GP) tools
  - (Sparse) quadrature integration
  - Rosenblatt transformation
  - Python postprocessing and analysis tools
- Version 3.0 to be released very soon
- Available at `http://www.sandia.gov/UQToolkit`

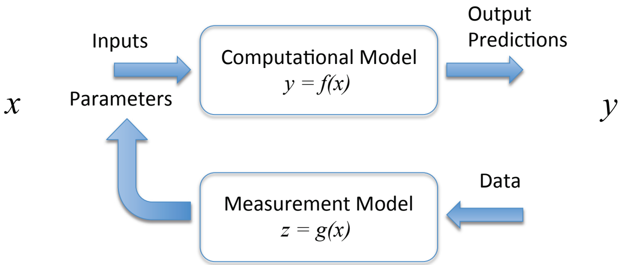
UQTk

Uncertainty Quantification and Computational Science



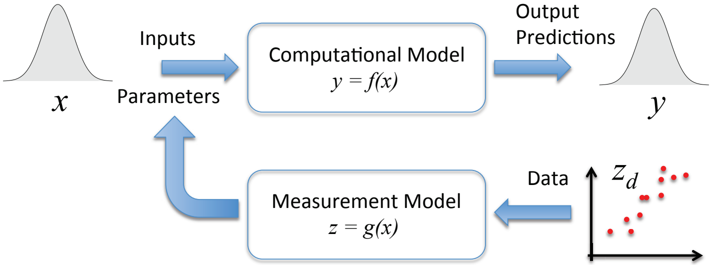
Forward problem

Uncertainty Quantification and Computational Science



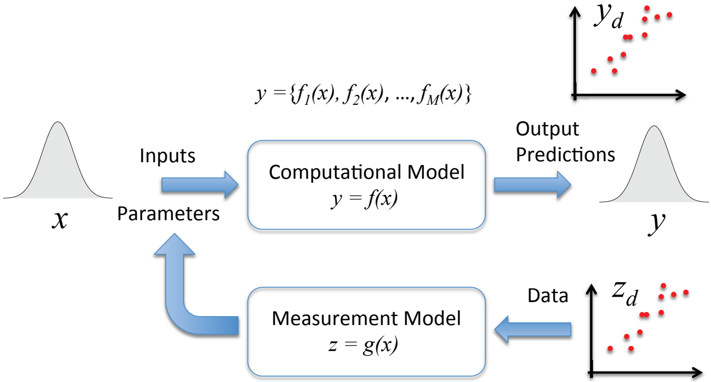
Inverse & Forward problems

Uncertainty Quantification and Computational Science



Inverse & Forward UQ

Uncertainty Quantification and Computational Science



Inverse & Forward UQ

Model validation & comparison, Hypothesis testing

The Case for Uncertainty Quantification

UQ needed for...

- Model predictions
- Model validation and comparison
- Confidence assessment
- Reliability analysis
- Dimensionality reduction
- Optimal design
- Decision support
- (Noisy) data assimilation

Uncertainty Sources

- Model parameters
- Initial/boundary conditions
- Model geometry/structure
- Lack of knowledge
- Data noise
- Intrinsic stochasticity
- Numerical errors, too

Outline

Polynomial Chaos – functional representation for RVs

$$U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$$

- First introduced by Wiener, 1938
- Revitalized by Ghanem and Spanos, 1991
- Convergent series if U has finite variance
- Selection of order p is a modeling choice
- Describes a r.v. U with a vector of *PC modes* $(u_0, u_1, \dots, u_p)$

- Standard r.v. ξ, standard orthogonal polynomials $\psi_k(\xi)$, *i.e.*

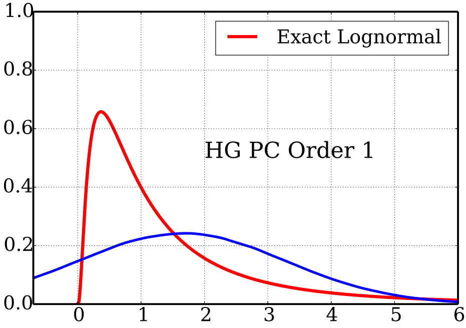
$$\int \psi_i(\xi)\psi_j(\xi)\pi_\xi(\xi)d\xi = \delta_{ij}||\psi_i||^2$$

PC Type	Domain	Density $\pi_\xi(\xi)$	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	$[0, +\infty)$	$\frac{\xi^\alpha e^{-\xi}}{\Gamma(\alpha+1)}$	Laguerre	$\alpha > -1$
Beta-Jacobi	$[-1, 1]$	$\frac{(1+\xi)^\alpha(1-\xi)^\beta}{2^{\alpha+\beta+1}B(\alpha+1,\beta+1)}$	Jacobi	$\alpha > -1, \beta > -1$

[Wiener, 1938; Ghanem & Spanos, 1991; Xiu & Karniadakis, 2002; Le Maître & Knio, 2010]

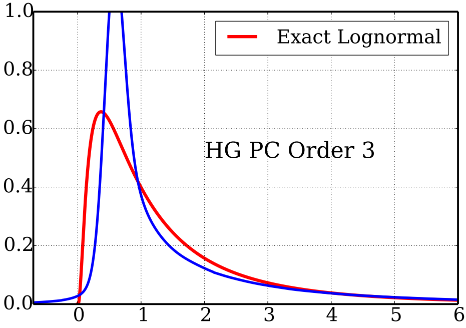
Construction of 1D PC $U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$

- Orthogonal projection: $u_k = \frac{1}{||\psi_k||^2}\langle U\psi_k\rangle$
- Need to compute integral $\langle U\psi_k\rangle = \int U(?)\psi_k(\xi)\pi_\xi(\xi)d\xi$
- Need a map $U \leftrightarrow \xi$
- If lucky, there is an explicit formula, e.g. lognormal $U = e^{\xi}$



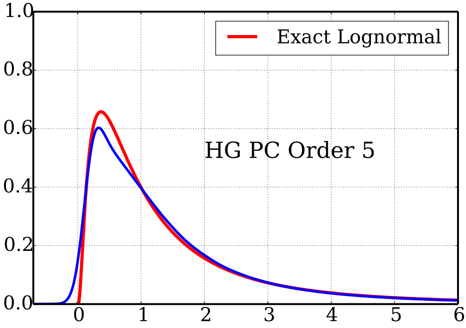
Construction of 1D PC $U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$

- Orthogonal projection: $u_k = \frac{1}{||\psi_k||^2}\langle U\psi_k\rangle$
- Need to compute integral $\langle U\psi_k\rangle = \int U(?)\psi_k(\xi)\pi_\xi(\xi)d\xi$
- Need a map $U \leftrightarrow \xi$
- If lucky, there is an explicit formula, e.g. lognormal $U = e^\xi$



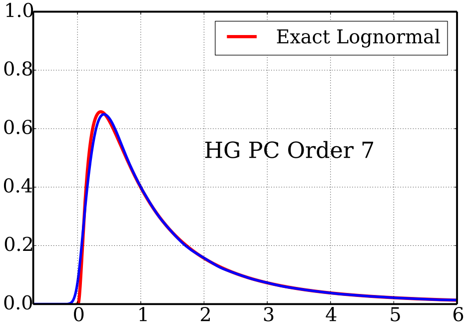
Construction of 1D PC $U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$

- Orthogonal projection: $u_k = \frac{1}{||\psi_k||^2} \langle U \psi_k \rangle$
- Need to compute integral $\langle U \psi_k \rangle = \int U(?) \psi_k(\xi) \pi_\xi(\xi) d\xi$
- Need a map $U \leftrightarrow \xi$
- If lucky, there is an explicit formula, e.g. lognormal $U = e^\xi$



Construction of 1D PC $U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$

- Orthogonal projection: $u_k = \frac{1}{||\psi_k||^2} \langle U \psi_k \rangle$
- Need to compute integral $\langle U \psi_k \rangle = \int U(?) \psi_k(\xi) \pi_\xi(\xi) d\xi$
- Need a map $U \leftrightarrow \xi$
- If lucky, there is an explicit formula, e.g. lognormal $U = e^{\xi}$



Construction of 1D PC $U \simeq \sum_{k=0}^{p} u_k \psi_k(\xi)$

- Orthogonal projection: $u_k = \frac{1}{||\psi_k||^2} \langle U\psi_k \rangle$
- Need to compute integral $\langle U\psi_k \rangle = \int U(?)\psi_k(\xi)\pi_\xi(\xi)d\xi$
- Need a map $U \leftrightarrow \xi$

- CDF transform helps:
  - $U = F_U^{-1}(\frac{\xi+1}{2})$ if ξ is Uniform, Legendre-Uniform PC
  - $U = F_U^{-1}(\Phi(\xi))$ if ξ is Normal, Gauss-Hermite PC

where $F_U(\cdot)$ is the Cumulative Distribution Function (CDF) of U.

[and $\Phi(\cdot)$ is CDF for standard normal]

Multivariate Polynomial Chaos

$$
\begin{cases}
U_1 = \sum_{k=0}^{K_1} u_{1k} \Psi_k(\xi_1, \dots, \xi_n) \\
U_2 = \sum_{k=0}^{K_2} u_{2k} \Psi_k(\xi_1, \dots, \xi_n) \\
\vdots \qquad\qquad \vdots \\
U_d = \sum_{k=0}^{K_d} u_{dk} \Psi_k(\xi_1, \dots, \xi_n)
\end{cases}
$$

- Multivariate polynomial $\Psi_k(\boldsymbol{\xi}) = \psi_{\alpha_1}(\xi_1) \cdots \psi_{\alpha_n}(\xi_n)$
- Usually $d = n$
- Construction non-trivial: e.g., capture
  - the PDF of U
  - select moments of U
  - some QoI $h(U)$
- Multivariate normal is a special case
- Multiindex $(\alpha_1, \dots, \alpha_n)$ selection, Truncation; see later
- Rosenblatt map (multivariate CDF transform)

Multivariate Polynomial Chaos

$$
\begin{cases}
U_1 = \sum_{k=0}^{K_1} u_{1k}\Psi_k(\xi_1,\dots,\xi_n) \\
U_2 = \sum_{k=0}^{K_2} u_{2k}\Psi_k(\xi_1,\dots,\xi_n) \\
\quad \vdots \qquad\qquad \vdots \\
U_d = \sum_{k=0}^{K_d} u_{dk}\Psi_k(\xi_1,\dots,\xi_n)
\end{cases}
$$

- Multivariate polynomial $\Psi_k(\boldsymbol{\xi}) = \psi_{\alpha_1}(\xi_1)\cdots\psi_{\alpha_n}(\xi_n)$
- Usually $d = n$
- Construction non-trivial: e.g., capture
  - the PDF of U
  - select moments of U
  - some QoI $h(U)$
- Multivariate normal is a special case
- Multiindex $(\alpha_1,\dots,\alpha_n)$ selection, Truncation; see later
- Rosenblatt map (multivariate CDF transform)

Fun example: $X = \xi_1^2 + \xi_2^2$ is exponential r.v. if ξ's are i.i.d. gaussians. However, no finite order 1D PC exists.

Essential Use of PC in UQ

$U \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$

Strategy:

- Represent model parameters/solution as random variables
- Construct PC for uncertain parameters
- Evaluate PC for model outputs

Advantages:

- Computational efficiency
- Utility
  - Moments: $\mathbb{E}[u] = u_0, \ \mathbb{V}[u] = \sum_{k=1}^{K} u_k^2 ||\Psi_k||^2, \ \ldots$
  - Global Sensitivities – fractional variances, Sobol' indices
  - Uncertainty propagation
  - Surrogate for forward model

Requirements:

- Finite variances (not a handicap in practice)
- Smooth forward functions

PC features: moment extraction

$$U \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

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

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

PC features: Global Sensitivity Analysis $U(\boldsymbol{\xi}) \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$

- Main effect sensitivity indices

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

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

PC features: Global Sensitivity Analysis $U(\boldsymbol{\xi}) \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$

- Main effect sensitivity indices

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

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

- Total effect sensitivity indices

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

$\mathbb{I}_i^T$ is the set of bases with ξ_i involved, including all its interactions.

PC features: Global Sensitivity Analysis $U(\boldsymbol{\xi}) \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$

- Main effect sensitivity indices

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

  - $\mathbb{I}_i$ 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

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

  - $\mathbb{I}_{ij}$ 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 features: uncertainty propagation

$$U \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

$$f(U) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

- Basic task: given PC for inputs, find PC for outputs.
- Input-output map can also be defined implicitly, via governing equations $G(f, U) = 0$.
- Two approaches
  - Intrusive: project governing equations
    - Results in set of equations for the PC modes
    - Requires redesign of computer code
    - PCEs for all uncertain variables in system
  - Non-intrusive: project outputs of interest
    - Sampling to evaluate projection operator
    - Can use existing code as black box
    - Only computes PCEs for quantities of interest

Non-intrusive Spectral Projection (NISP) PC UQ

$$U \simeq \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

$$f(U) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

- For any model output of interest $f(X)$:

$$f_k = \frac{\langle f \Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \frac{1}{||\Psi_k||^2} \int f(X(\boldsymbol{\xi}))\, \Psi_k(\boldsymbol{\xi}) \pi_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

- Evaluate projection integral *numerically*
- Relies on black-box utilization of the computational model
- Integral can be evaluated using
  - A variety of (Quasi) Monte Carlo methods
    - Slow convergence; $\sim$ indep. of dimensionality
  - Quadrature/Sparse-Quadrature methods
    - Fast convergence; depends on dimensionality

PC surrogate construction

- Build/presume PC for input parameter U

$$U(\boldsymbol{\xi}) = \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

with respect to multivariate standard polynomials.

PC surrogate construction

- Build/presume PC for input parameter U

$$U(\boldsymbol{\xi}) = \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

with respect to multivariate standard polynomials.

- E.g., uniform on an interval, or gaussian with known moments,

$$U = u_0 + u_1^T \boldsymbol{\xi}$$

PC surrogate construction

- Build/presume PC for input parameter U

$$U(\boldsymbol{\xi}) = \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

with respect to multivariate standard polynomials.

- If input parameters are uniform $U_i \sim$ Uniform$[a_i, b_i]$, then

$$U_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2}\, \xi_i.$$

PC surrogate construction

- Build/presume PC for input parameter U

$$U(\boldsymbol{\xi}) = \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

with respect to multivariate standard polynomials.

- Input parameters are represented via their cumulative distribution function (CDF) $F(\cdot)$, such that, with $\xi_i \sim$ Uniform$[-1, 1]$

$$U_i = F_{U_i}^{-1}\left(\frac{\xi_i + 1}{2}\right), \qquad \text{for } i = 1, 2, \ldots, d.$$

PC surrogate construction

- Build/presume PC for input parameter U

$$U(\boldsymbol{\xi}) = \sum_{k=0}^{K} u_k \Psi_k(\boldsymbol{\xi})$$

with respect to multivariate standard polynomials.

- Input parameters are represented via their cumulative distribution function (CDF) $F(\cdot)$, such that, with $\xi_i \sim$ Uniform$[-1, 1]$

$$U_i = F_{U_i}^{-1}\left(\frac{\xi_i + 1}{2}\right), \qquad \text{for } i = 1, 2, \ldots, d.$$

- Forward function $f(\cdot)$, output Z

$$Z = f(U(\boldsymbol{\xi})) \qquad Z = \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi}) \equiv f_s(\boldsymbol{\xi})$$

- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.

Outline

Inverse UQ – Estimation of Uncertain Parameters

Probabilistic setting

- Require joint PDF on input space
- Statistical inference – an inverse problem

Bayesian setting

- Given Constraints: PDF on uncertain inputs can be estimated using the Maximum Entropy principle
  - MaxEnt Methods
- Given Data: PDF on uncertain inputs can be estimated using Bayes formula
  - Bayesian Inference

Bayes formula for Parameter Inference

- Collected data: $\{(x_i, y_i)\}_{i=1}^N$
- Data model: $y_i = f(x_i; \lambda) + \epsilon_i$
- Bayes formula:

$$\underbrace{\boxed{p(\lambda|y)}}_{\text{Posterior}} = \frac{\overbrace{\boxed{p(y|\lambda)}}^{\text{Likelihood}}\ \overbrace{\boxed{p(\lambda)}}^{\text{Prior}}}{\underbrace{\boxed{p(y)}}_{\text{Evidence}}}$$

- Prior: knowledge of λ prior to data
- Likelihood: forward model and measurement noise
- Posterior: combines information from prior and data
- Evidence: normalizing constant for present context

The Prior

- Prior $p(\lambda)$ comes from
  - Physical constraints
  - Prior data/knowledge
- Types of *uninformative* priors
  - Improper prior
  - Objective prior
  - Maxent prior
  - Reference prior
  - Jeffreys prior

$$\underbrace{p(\lambda|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|\lambda)}^{\text{Likelihood}}\ \overbrace{p(\lambda)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- It can be chosen to impose *regularization*
- Unknown aspects of the prior can be added to the rest of the parameters as hyperparameters
- The choice of prior can be crucial if data is not informative
- When there is sufficient information in the data, the data can overrule the prior

Construction of the Likelihood $p(y|\lambda)$

- Requires a presumed error model
- Data model: $y_i = f(x_i; \lambda) + \epsilon_i$

$$\underbrace{p(\lambda|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|\lambda)}^{\text{Likelihood}}\ \overbrace{p(\lambda)}^{\text{Prior}}}{\underbrace{p(y)}_{\text{Evidence}}}$$

- Model this error as a random variable, e.g.
  - Error is due to instrument measurement noise
  - Instrument has Gaussian errors, with no bias
  - Measurements are independent

$$\epsilon \sim N(0, \sigma^2)$$

- For any given λ, this implies

$$y_i|\lambda, \sigma \sim N(f(x_i; \lambda), \sigma^2)$$

or

$$p(y|\lambda, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(y_i - f(x_i; \lambda))^2}{2\sigma^2}\right)$$

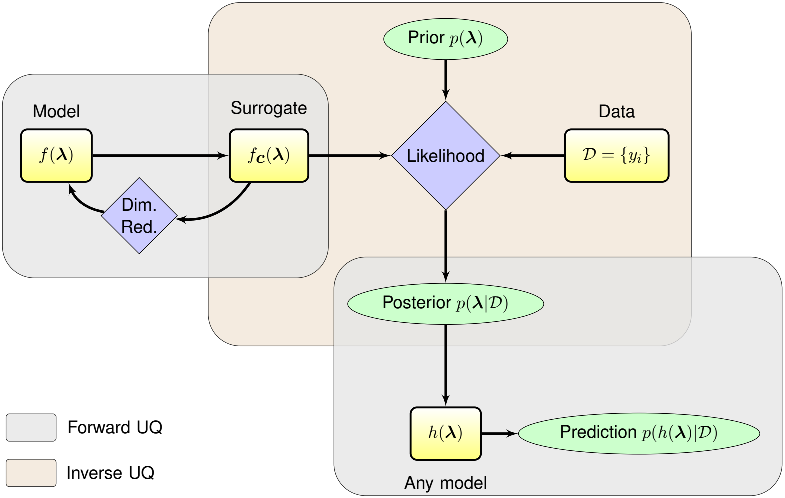
Exploring the Posterior

- Given any sample λ, the un-normalized posterior probability can be easily computed

$$\underbrace{p(\lambda|y)}_{\text{Posterior}} \propto \underbrace{p(y|\lambda)}_{\text{Likelihood}} \underbrace{p(\lambda)}_{\text{Prior}}$$

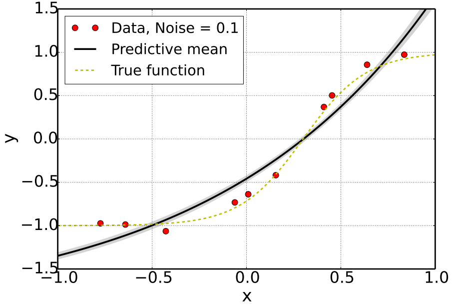
- Explore posterior w/ Markov Chain Monte Carlo (MCMC)
  - Metropolis-Hastings algorithm:
    - Random walk with proposal PDF & rejection rules
  - Computationally intensive, $\mathcal{O}(10^5)$ samples
  - Each sample: evaluation of the forward model
    - Surrogate models
- Evaluate moments/marginals from the MCMC statistics

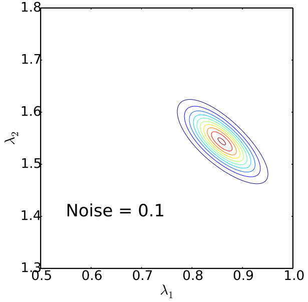
Forward and Inverse UQ in a workflow



Bayesian inference: Noise$\uparrow \Rightarrow$ Posterior uncertainty$\uparrow$

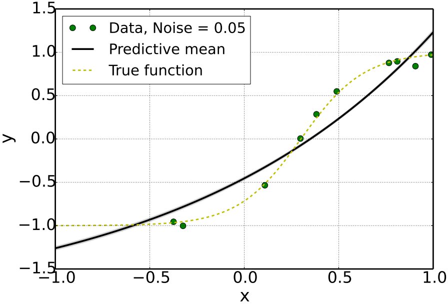
- True model $y = \tanh(3x - 0.9)$
- Increasing data noise level
- Calibrating $f(x; \lambda) = \lambda_1 e^{\lambda_0 x} - 2$
- Larger data noise $\Rightarrow$ larger posterior uncertainty

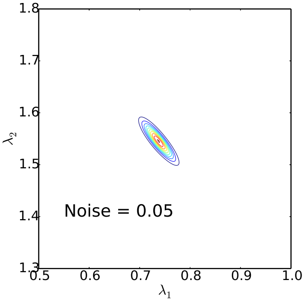




Bayesian inference: Noise↑ ⇒ Posterior uncertainty↑

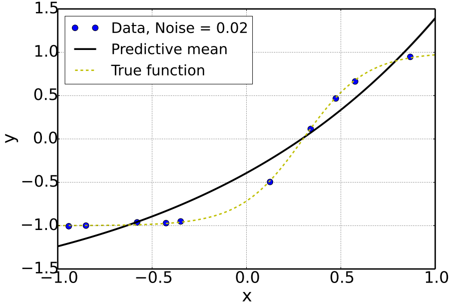
- True model $y = \tanh(3x - 0.9)$
- Increasing data noise level
- Calibrating $f(x;\lambda) = \lambda_1 e^{\lambda_0 x} - 2$
- Larger data noise ⇒ larger posterior uncertainty

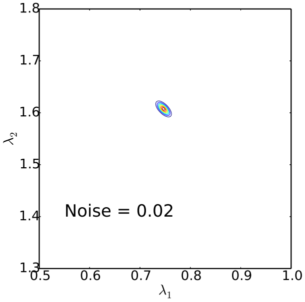




Bayesian inference: Noise$\uparrow \Rightarrow$ Posterior uncertainty$\uparrow$

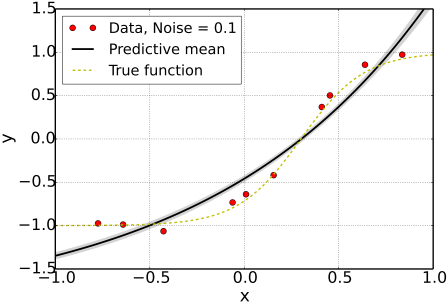
- True model $y = \tanh(3x - 0.9)$
- Increasing data noise level
- Calibrating $f(x;\lambda) = \lambda_1 e^{\lambda_0 x} - 2$
- Larger data noise $\Rightarrow$ larger posterior uncertainty

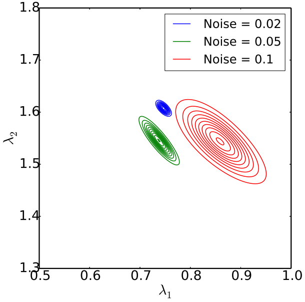




Bayesian inference: Noise$\uparrow$ $\Rightarrow$ Posterior uncertainty$\uparrow$

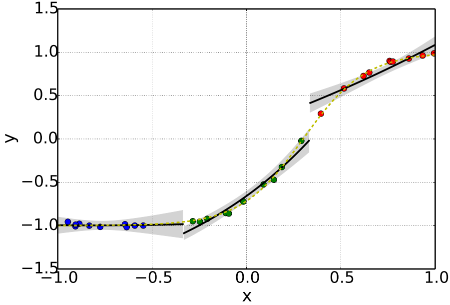
- True model $y = \tanh(3x - 0.9)$
- Increasing data noise level
- Calibrating $f(x; \lambda) = \lambda_1 e^{\lambda_0 x} - 2$
- Larger data noise $\Rightarrow$ larger posterior uncertainty

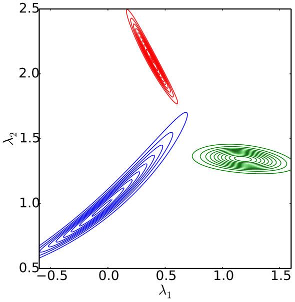




Bayesian inference: Data range $\Rightarrow$ Correlation

- True model $y = \tanh(3x - 0.9)$
- Collecting data at different locations
- Calibrating $f(x;\lambda) = \lambda_1 e^{\lambda_0 x} - 2$
- Correlation structure can change drastically





Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
  - Large number of input parameters
  - Dense spatial/temporal grid
  - PC truncation is a challenge
  - Low-rank (tensor) representations
  - **Sparse learning, (Bayesian) compressive sensing**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
  - UQ studies seriously hindered
  - Need surrogates with few model simulations
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
  - Polynomial representation not good enough
  - Quadrature integration fails

  - Stochastic domain decomposition
  - Data clustering/classification
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
  - Bayesian inference is prior-dominated
  - Lack of parameter identifiability
  - Bayesian methods do quantify lack-of-data uncertainty
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
  - Quadrature and sparse quadrature methods fail
  - Averaged quantities
  - Bayesian regression
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
  - Models are not perfect
  - Can not be ignored during parameter estimation
  - Additive model error as a Gaussian Process
  - **Embedded model error**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
  - Hard to sample from
  - Hard to interpret sensitivities
  - Rosenblatt transformation
- Low-Probability (Tail) Events
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
  - PC inaccurate in capturing regions of low probability
  - Use targeted PC germs ξ with fat tails
- Time Dynamics

Laundry List of Challenges/Issues (incomplete)

- **High-Dimensionality**
- Expensive Models
- Non-Linear Models, Discontinuities, Bimodalities
- Scarce Data
- Intrinsic Stochasticity
- **Model Errors**
- Input Correlations
- Low-Probability (Tail) Events
- Time Dynamics
  - Large amplification of phase errors over long time horizon
  - Chaotic dynamics

  - Increase order with time to retain accuracy
  - Ad-hoc corrections
  - Look at averaged quantities

Outline

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- Computationally expensive model simulations, data sparsity
  - Need to build accurate surrogates with as few training runs as possible
- High-dimensional input space
  - Too many samples needed to cover the space
  - Too many terms in the polynomial expansion

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- Computationally expensive model simulations, data sparsity
  - Need to build accurate surrogates with as few training runs as possible
- High-dimensional input space
  - Too many samples needed to cover the space
  - Too many terms in the polynomial expansion

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- Computationally expensive model simulations, data sparsity
  - Need to build accurate surrogates with as few training runs as possible
- High-dimensional input space
  - Too many samples needed to cover the space
  - Too many terms in the polynomial expansion

Alternative methods to obtain PC coefficients

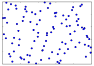
$$Z = f(U(\boldsymbol{\xi})) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

- Projection $\qquad f_k = \frac{\langle f(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \rangle}{||\Psi_k||^2}$

  The integral $\langle f(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \rangle = \int f(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \pi_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}$ is estimated by...

  - Monte-Carlo

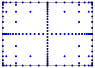
$$\frac{1}{N} \sum_{j=1}^{N} f(\boldsymbol{\xi}_j) \Psi_k(\boldsymbol{\xi}_j)$$



many(!) random samples

  - Quadrature

$$\sum_{j=1}^{Q} f(\boldsymbol{\xi}_j) \Psi_k(\boldsymbol{\xi}_j) w_j$$



samples at quadrature

Alternative methods to obtain PC coefficients

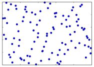
$$Z = f(U(\boldsymbol{\xi})) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

- Projection $\quad f_k = \frac{\langle f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle}{||\Psi_k||^2}$

  The integral $\langle f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle = \int f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\pi_{\boldsymbol{\xi}}(\boldsymbol{\xi})d\boldsymbol{\xi}$ is estimated by...

  - Monte-Carlo

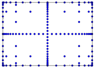
$$\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{\xi}_j)\Psi_k(\boldsymbol{\xi}_j)$$



many(!) random samples

  - Quadrature

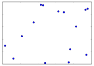
$$\sum_{j=1}^{Q} f(\boldsymbol{\xi}_j)\Psi_k(\boldsymbol{\xi}_j)w_j$$



samples at quadrature

- Bayesian regression

$$P(f_k|f(\boldsymbol{\xi}_j)) \propto P(f(\boldsymbol{\xi}_j)|f_k)P(f_k)$$



any (number of) samples

Alternative methods to obtain PC coefficients

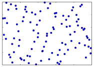
$$Z = f(U(\boldsymbol{\xi})) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

- Projection $\quad f_k = \frac{\langle f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle}{||\Psi_k||^2}$

  The integral $\langle f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle = \int f(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\pi_{\boldsymbol{\xi}}(\boldsymbol{\xi})d\boldsymbol{\xi}$ is estimated by...

  - Monte-Carlo

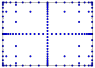
$$\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{\xi}_j)\Psi_k(\boldsymbol{\xi}_j)$$



many(!) random samples

  - Quadrature

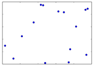
$$\sum_{j=1}^{Q} f(\boldsymbol{\xi}_j)\Psi_k(\boldsymbol{\xi}_j)w_j$$



samples at quadrature

- Bayesian regression

$$\underbrace{P(\boldsymbol{f}|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{P(\mathcal{D}|\boldsymbol{f})}_{\text{Likelihood}}\underbrace{P(\boldsymbol{f})}_{\text{Prior}}$$



any (number of) samples

Bayesian inference of PC surrogate

$$Z = f(\boldsymbol{\xi}) \simeq \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi}) \equiv f_s(\boldsymbol{\xi}) \qquad \overbrace{P(\boldsymbol{f}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{f})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{f})}^{\text{Prior}}$$

- Data consists of *training runs*

$$\mathcal{D} \equiv \{(\boldsymbol{\xi}_i, Z_i)\}_{i=1}^{N}$$

- Likelihood with a gaussian noise model with σ^2 fixed or inferred,

$$L(\boldsymbol{f}) = P(\mathcal{D}|\boldsymbol{f}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^{N} \exp\left(-\frac{(f_i - f_s(\boldsymbol{\xi}_i))^2}{2\sigma^2}\right)$$

- Prior on $\boldsymbol{f}$ is chosen to be conjugate, uniform or gaussian.
- Posterior is a *multivariate normal*

$$\boldsymbol{f} \quad \in \quad \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

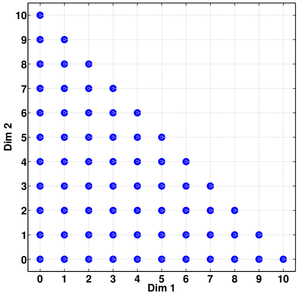
- The (uncertain) surrogate is *a gaussian process*

$$f_s(\boldsymbol{\xi}) = \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi}) = \boldsymbol{\Psi}(\boldsymbol{\xi})^T \boldsymbol{f} \quad \in \quad \mathcal{GP}(\boldsymbol{\Psi}(\boldsymbol{\xi})^T \boldsymbol{\mu}, \boldsymbol{\Psi}(\boldsymbol{\xi})\boldsymbol{\Sigma}\boldsymbol{\Psi}(\boldsymbol{\xi}')^T)$$

Bayesian inference of PC surrogate: high-d, low-data regime

$$Z = f(\boldsymbol{\xi}) \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

$$\Psi_k(\xi_1, \xi_2, ..., \xi_d) = \psi_{k_1}(\xi_1)\psi_{k_2}(\xi_2)\cdots\psi_{k_d}(\xi_d)$$



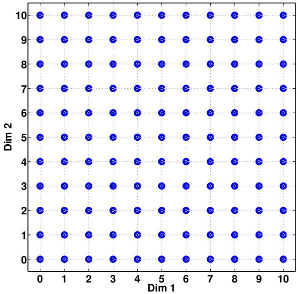
- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)

- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

$$Z = f(\boldsymbol{\xi}) \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

$$\Psi_k(\xi_1, \xi_2, ..., \xi_d) = \psi_{k_1}(\xi_1)\psi_{k_2}(\xi_2)\cdots\psi_{k_d}(\xi_d)$$

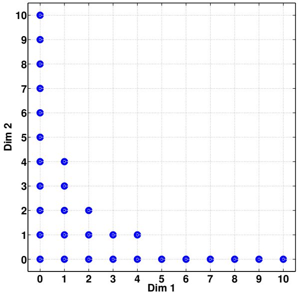


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

$$Z = f(\boldsymbol{\xi}) \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

$$\Psi_k(\xi_1, \xi_2, ..., \xi_d) = \psi_{k_1}(\xi_1)\psi_{k_2}(\xi_2)\cdots\psi_{k_d}(\xi_d)$$

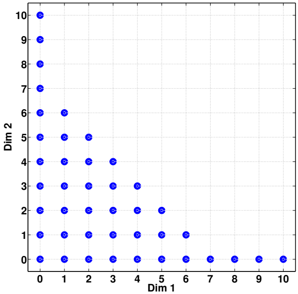


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

$$Z = f(\boldsymbol{\xi}) \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

$$\Psi_k(\xi_1, \xi_2, ..., \xi_d) = \psi_{k_1}(\xi_1)\psi_{k_2}(\xi_2)\cdots\psi_{k_d}(\xi_d)$$

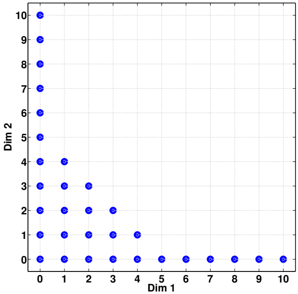


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

$$Z = f(\boldsymbol{\xi}) \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi})$$

$$\Psi_k(\xi_1, \xi_2, ..., \xi_d) = \psi_{k_1}(\xi_1)\psi_{k_2}(\xi_2)\cdots\psi_{k_d}(\xi_d)$$



- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

In a different language....

- N training data points $(\boldsymbol{\xi}_i, Z_i)$ and $K+1$ basis terms $\Psi_k(\cdot)$
- Projection matrix $\boldsymbol{P}^{N\times(K+1)}$ with $\boldsymbol{P}_{ik} = \Psi_k(\boldsymbol{x}_i)$
- Find regression weights $\boldsymbol{f} = (f_0, \ldots, f_K)$ so that

$$\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f} \quad \text{or} \quad Z_i \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi}_i)$$

- The number of polynomial basis terms grows fast; a p-th order, d-dimensional basis has a total of $K+1 = (p+d)!/(p!d!)$ terms.
- For limited data and large basis set ($N \leq K$) this is a sparse signal recovery problem $\Rightarrow$ need some regularization/constraints.

- Least-squares $\quad \mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2\}$
- The ‘sparsest’ $\quad \mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2 + \alpha||\boldsymbol{f}||_0\}$
- Compressive sensing $\quad \mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2 + \alpha||\boldsymbol{f}||_1\}$

In a different language....

- N training data points $(\boldsymbol{\xi}_i, Z_i)$ and $K+1$ basis terms $\Psi_k(\cdot)$
- Projection matrix $\boldsymbol{P}^{N\times(K+1)}$ with $\boldsymbol{P}_{ik} = \Psi_k(\boldsymbol{x}_i)$
- Find regression weights $\boldsymbol{f} = (f_0, \dots, f_K)$ so that

$$\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f} \quad \text{or} \quad Z_i \approx \sum_{k=0}^{K} f_k \Psi_k(\boldsymbol{\xi}_i)$$

- The number of polynomial basis terms grows fast; a p-th order, d-dimensional basis has a total of $K+1 = (p+d)!/(p!d!)$ terms.
- For limited data and large basis set ($N \le K$) this is a sparse signal recovery problem $\Rightarrow$ need some regularization/constraints.

- Least-squares $\quad$ *argmin*$_{\boldsymbol{c}}\{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2\}$
- The 'sparsest' $\quad$ *argmin*$_{\boldsymbol{c}}\{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2 + \alpha||\boldsymbol{f}||_0\}$
- Compressive sensing $\quad$ *argmin*$_{\boldsymbol{c}}\{||\boldsymbol{Z} - \boldsymbol{P}\boldsymbol{f}||_2 + \alpha||\boldsymbol{f}||_1\}$

Bayesian $\quad$ Likelihood $\quad$ Prior

Bayesian Compressive Sensing (BCS), or Relevance Vector Machine (RVM)

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{f}|\alpha) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2)p(\sigma_k^2|\alpha)d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|f_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{c} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

Bayesian Compressive Sensing (BCS), or Relevance Vector Machine (RVM)

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad\qquad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{f}|\alpha) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2) p(\sigma_k^2|\alpha) d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|f_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{c} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad\qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

Bayesian Compressive Sensing (BCS), or Relevance Vector Machine (RVM)

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{f}|\alpha) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2)p(\sigma_k^2|\alpha)d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|f_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{c} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

Bayesian Compressive Sensing (BCS), or Relevance Vector Machine (RVM)

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{f}|\alpha) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2) p(\sigma_k^2|\alpha) d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|f_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{c} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

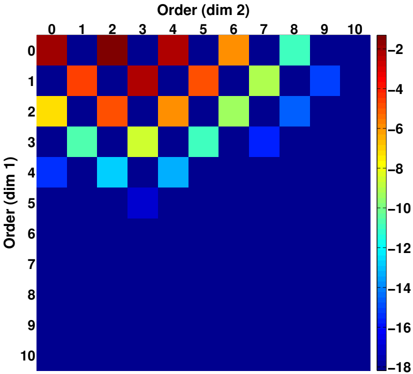
$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

- KEY: Some $\sigma_k^2 \to 0$, hence the corresponding basis terms are dropped.

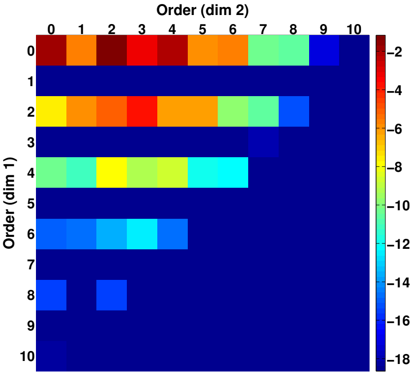
[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

BCS removes unnecessary basis terms

$f(x,y) = \cos(x + 4y)$

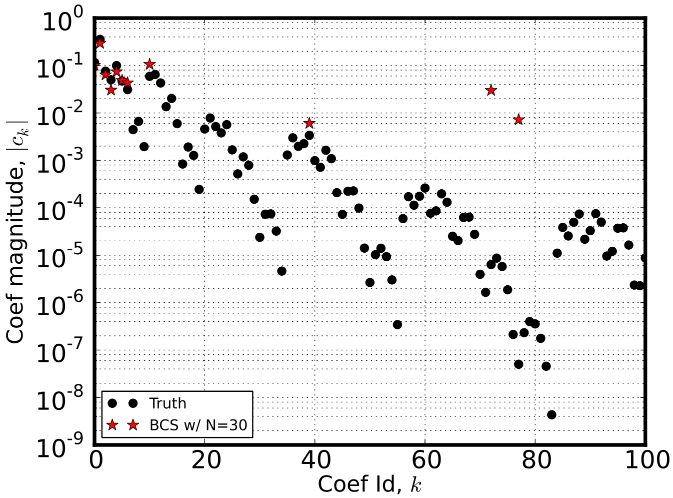


$f(x,y) = \cos(x^2 + 4y)$

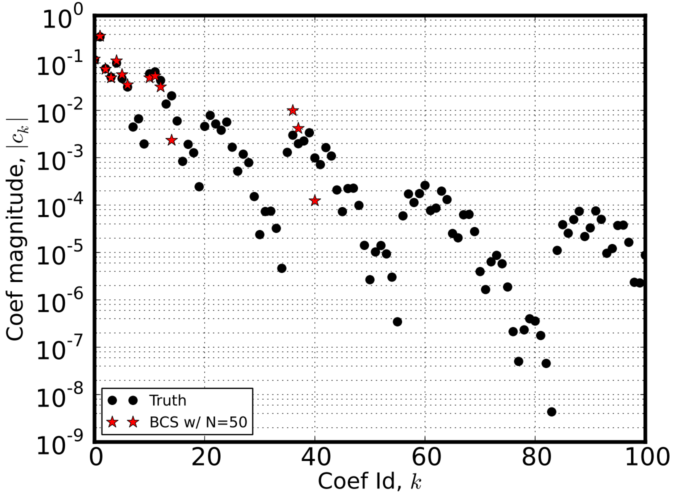


The square (i,j) represents the (log) spectral coefficient for the basis term $\psi_i(x)\psi_j(y)$.

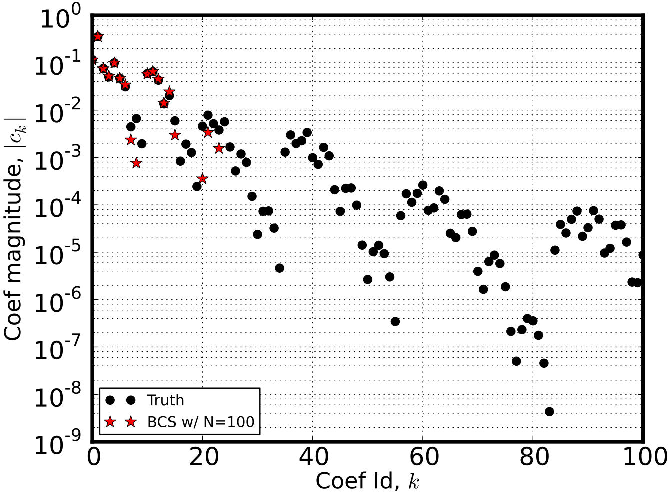
BCS recovers true PC coefficients with increased number of measurements



BCS recovers true PC coefficients with increased number of measurements



BCS recovers true PC coefficients with increased number of measurements



Bayesian Compressive Sensing

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad\qquad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{c}|\boldsymbol{\alpha}) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2) p(\sigma_k^2|\alpha) d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|f_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{f} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

  with

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{P}^T \boldsymbol{u} \qquad\qquad \boldsymbol{\Sigma} = \sigma^2 (\boldsymbol{P}^T \boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

- KEY: Some $\sigma_k^2 \to 0$, hence the corresponding basis terms are dropped.

Weighted Bayesian Compressive Sensing

- Dimensionality reduction by using hierarchical priors

$$p(f_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{f_k^2}{2\sigma_k^2}} \qquad p(\sigma_k^2|\alpha_k) = \frac{\alpha_k}{2} e^{-\frac{\alpha_k \sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\boldsymbol{c}|\boldsymbol{\alpha}) = \int \prod_{k=0}^{K-1} p(f_k|\sigma_k^2) p(\sigma_k^2|\alpha_k) d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha_k}}{2} e^{-\sqrt{\alpha_k}|f_k|}$$

- The parameter α_k can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha_k, \sigma^2$ and allows exact Bayesian solution

$$\boldsymbol{f} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{P}^T \boldsymbol{u} \qquad \boldsymbol{\Sigma} = \sigma^2 (\boldsymbol{P}^T \boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

- KEY: Some $\sigma_k^2 \to 0$, hence the corresponding basis terms are dropped.

Iteratively reweighting Compressive Sensing [Candes *et al.*, 2007]

Sparsest solution:	$min\|\|\boldsymbol{f}\|\|_0$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$
Compressive sensing:	$min\|\|\boldsymbol{f}\|\|_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$
Weighted compressive sensing:	$min\|\|\boldsymbol{W}\boldsymbol{f}\|\|_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

Iteratively reweighting Compressive Sensing [Candes *et al.*, 2007]

Sparsest solution: $min||\boldsymbol{f}||_0$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

Compressive sensing: $min||\boldsymbol{f}||_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

Weighted compressive sensing: $min||\boldsymbol{W}\boldsymbol{f}||_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

For sparse signals, $\boldsymbol{Z} = \boldsymbol{P}\boldsymbol{f}^s$, with $||\boldsymbol{f}^s||_0 = S < K$, ideal weights are

$$\boldsymbol{W} = diag\left(\frac{1}{|f_k^s|}\right)$$ [i.e., $W_{kk} = +\infty$ if $f_k^s = 0$]

In practice, the true signal coefficients are not known, so...

Iteratively reweighting Compressive Sensing [Candes *et al.*, 2007]

Sparsest solution: $min||\boldsymbol{f}||_0$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

Compressive sensing: $min||\boldsymbol{f}||_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

Weighted compressive sensing: $min||\boldsymbol{W}\boldsymbol{f}||_1$ such that $\boldsymbol{Z} \approx \boldsymbol{P}\boldsymbol{f}$

For sparse signals, $\boldsymbol{Z} = \boldsymbol{P}\boldsymbol{f}^s$, with $||\boldsymbol{f}^s||_0 = S < K$, ideal weights are

$$\boldsymbol{W} = diag\left(\frac{1}{|f_k^s|}\right)$$

[i.e., $W_{kk} = +\infty$ if $f_k^s = 0$]

In practice, the true signal coefficients are not known, so...

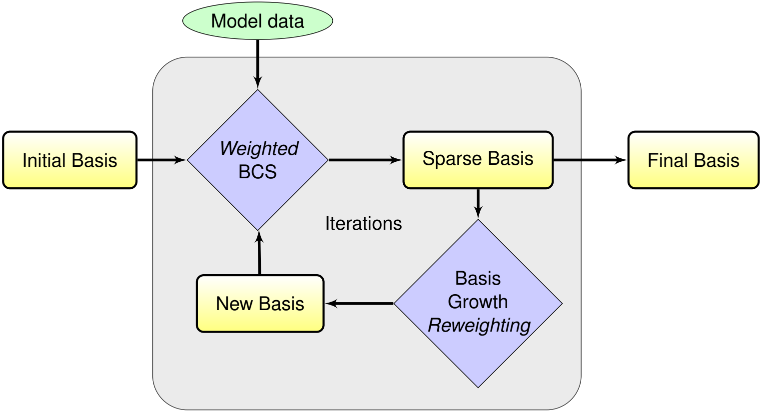
Iterative re-weighting

$$\boldsymbol{W}^{(i+1)} = diag\left(\frac{1}{|f_k^{(i)}| + \epsilon}\right)$$

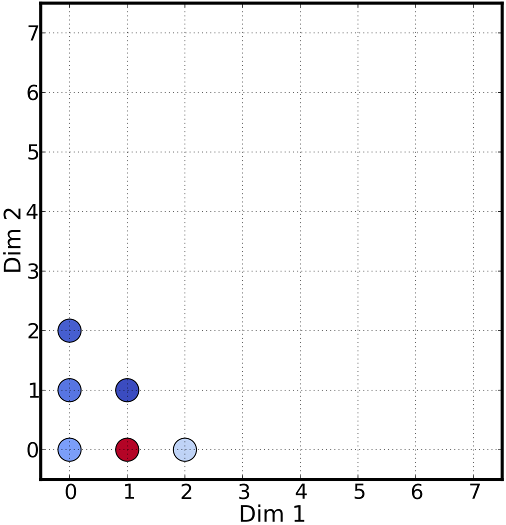
[$\epsilon \ll 1$ for stability]

Weighted Iterative BCS

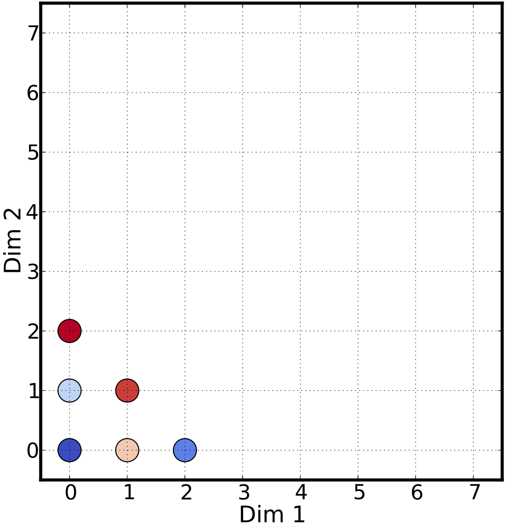
- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Sargsyan *et al.* 2014], [Jakeman *et al.* 2015].
- Combine basis growth and reweighting!



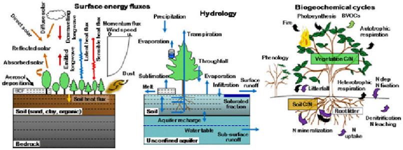
Basis set growth: simple anisotropic function

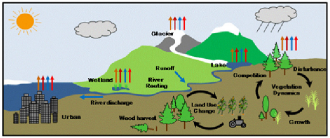


Basis set growth: ... added outlier term



The UQ Challenge for ACME Land Model



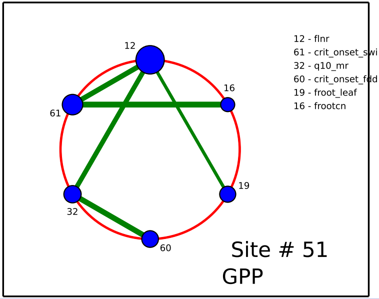


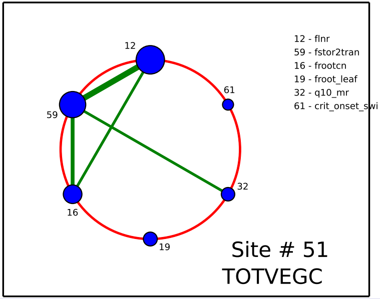
http://www.cesm.ucar.edu/models/clm/

- A single-site, 1000-yr simulation takes ~ 10 hrs on 1 CPU
- Involves ~ 70 input parameters; some dependent
- Non-smooth input-output relationship

Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

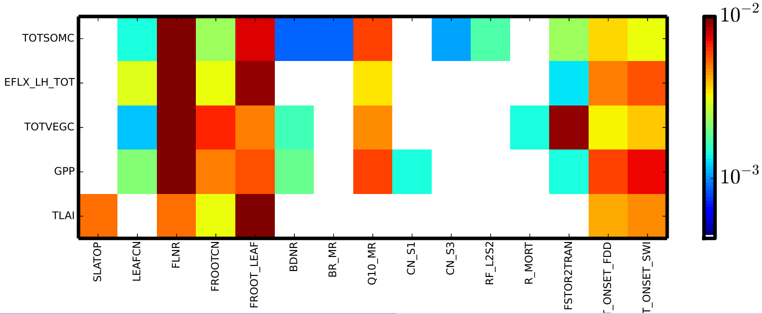
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
    - sampling in a reduced space
    - parameter calibration against experimental data





Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

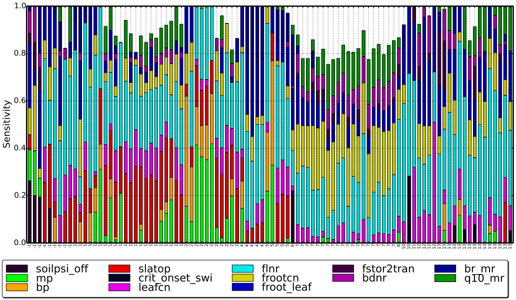
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data

- GPP
  gross primary
  productivity



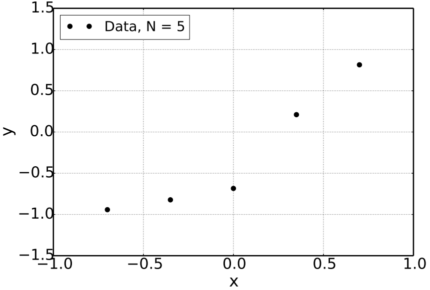
Outline

Main target: quantification of model error

Model error = deviation from 'truth', or from a higher-fidelity model

- Represent and estimate the error associated with
  - Simplifying assumptions, parameterizations
  - Mathematical formulation, theoretical framework
  - Numerical discretization
- ...will be useful for
  - Model validation
  - Model comparison
  - Scientific discovery and model improvement
  - Reliable computational predictions
- Inverse modeling context
  - Given experimental or higher-fidelity model data, estimate the model error

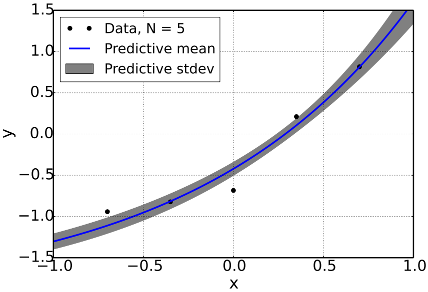
Motivation: can't afford ignoring model error



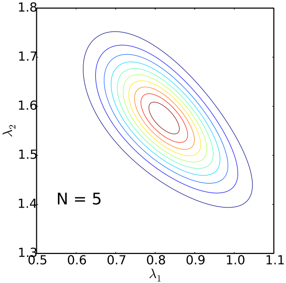
Model-data fit

- Given noisy data – Gaussian noise
- $y = g_{\text{true}}(x) + \epsilon$

Motivation: can't afford ignoring model error



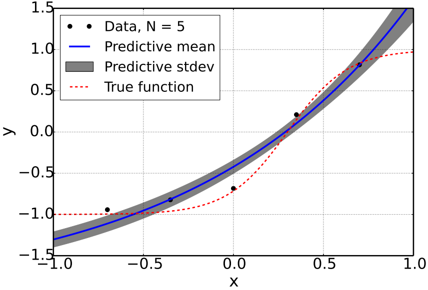
Model-data fit



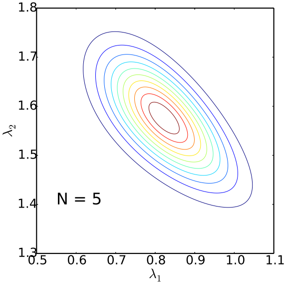
Posterior on parameters

- Employ Bayesian inference to fit an exponential model: $y_{\rm m} = f(x;\lambda)$
- Discrepancy between data and prediction presumed exclusively due to *i.i.d.* Gaussian data noise: $y = f(x;\lambda) + \epsilon_{\rm d}$
- Plotted:
  - Posterior density on the parameters
  - Preditive mean and standard deviation

Motivation: can't afford ignoring model error



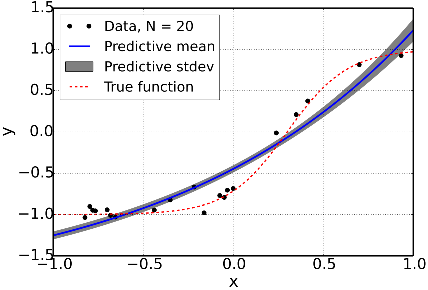
Model-data fit



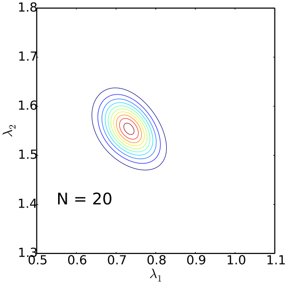
Posterior on parameters

- Employ Bayesian inference to fit an exponential model: $y_{\mathrm{m}} = f(x;\lambda)$
- Discrepancy between data and prediction presumed exclusively due to *i.i.d.* Gaussian data noise: $y = f(x;\lambda) + \epsilon_{\mathrm{d}}$
- True model $g(x)$ – dashed-red – differs from fit model $f(x,\lambda)$
- Actual discrepancy includes both data and model errors

Motivation: can't afford ignoring model error



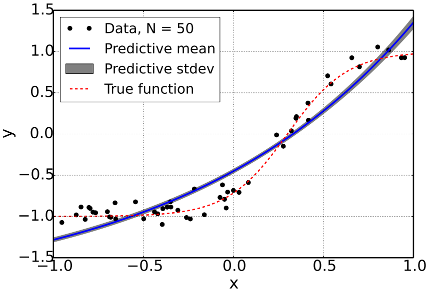
Model-data fit



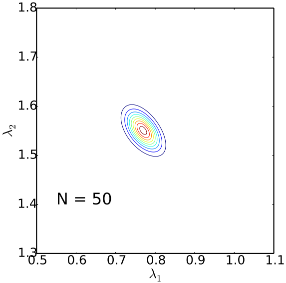
Posterior on parameters

- Increasing number of data points decreases posterior and predictive uncertainty
- We are increasingly sure about predictions based on the *wrong* model

Motivation: can't afford ignoring model error



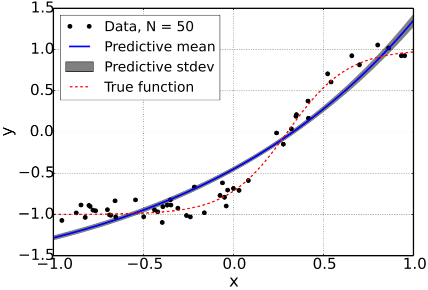
Model-data fit



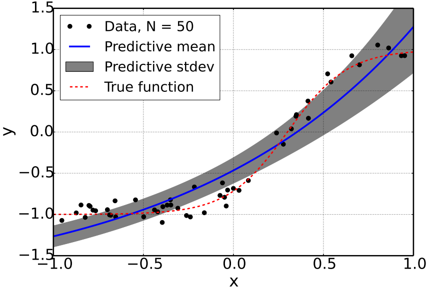
Posterior on parameters

- Increasing number of data points decreases posterior and predictive uncertainty
- We are increasingly sure about predictions based on the *wrong* model

Motivation: can't afford ignoring model error



Model-data fit



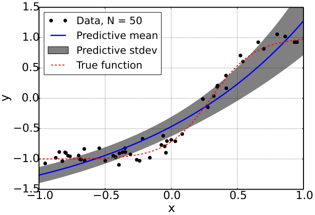
What we want

- If the model has structural uncertainty, more data leads to biased and overconfident results
- We want to quantify model-vs-truth discrepancy in a rigorous and systematic way
  - Cannot ignore model error

Model Error – Challenges with current methods

Total error budget

$$y_i = \underbrace{f(x_i;\lambda) + \delta(x_i)}_{\text{Truth } g(x_i)} + \epsilon_i$$



- Ignoring model error $\delta(x)$ leads to incorrect predictive errors
- Conventional statistical modeling (Kennedy and O'Hagan, 2001)
  - makes it difficult to disambiguate model/data errors
  - may violate physical constraints
  - not meaningful for prediction of other QoIs
- Issue is highlighted in model-to-model calibration ($\epsilon_i = 0$)
  - no a priori knowledge of the statistical structure of the discrepancy

Model Error – Key idea: probabilistic embedding

Cast input parameters λ as a random variable Λ

$$y_i = f(x_i;\lambda) + \delta(x_i) + \epsilon_i \longrightarrow y_i = f(x_i;\Lambda) + \epsilon_i$$

- Embed model error in specific submodel phenomenology
  - a modified transport or constitutive law
  - a modified formulation for a material property
  - turbulent model constants
- Allows placement of model error term in locations where key modeling assumptions and approximations are made
  - as a correction or high-order term
  - as a possible alternate phenomenology
- Naturally preserves model structure and physical constraints
- Disambiguates model/data errors

Model Error – Bayesian density estimation

$$y_i = f(x_i; \Lambda) + \epsilon_i$$

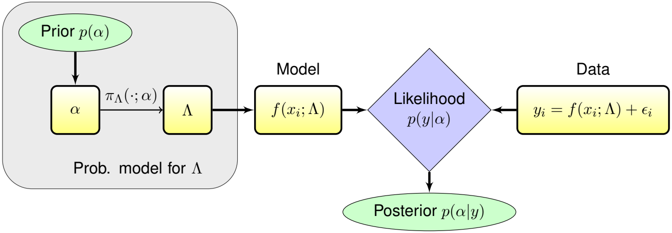
- Parametrise embedded random variable Λ:
  - PDF form $\pi_\Lambda(\cdot; \alpha)$
  - Polynomial Chaos (PC): $\Lambda = \sum_k \alpha_k \Psi_k(\xi)$
  - Multivariate Normal (MVN): $\begin{cases} \Lambda_1 = \alpha_{10} + \alpha_{11}\xi_1 \\ \Lambda_2 = \alpha_{20} + \alpha_{21}\xi_1 + \alpha_{22}\xi_2 \\ \quad\vdots \\ \Lambda_d = \alpha_{d0} + \alpha_{d1}\xi_1 + \alpha_{d2}\xi_2 + \cdots + \alpha_{dd}\xi_d \end{cases}$

- Inverse modeling context
  - Parameter estimation of $\lambda \Rightarrow$ PDF estimation of $\Lambda \Rightarrow$
    $\Rightarrow$ parameter estimation of α
  - Bayesian formulation

$$\underbrace{p(\alpha|y)}_{\text{Posterior}} \propto \underbrace{L_y(\alpha)}_{\text{Likelihood}} \underbrace{p(\alpha)}_{\text{Prior}}$$

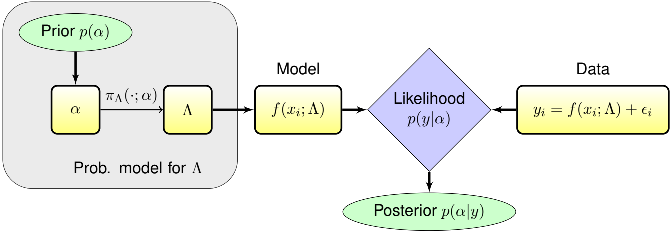
Model Error – Likelihood options

K. Sargsyan, H. Najm, and R. Ghanem, "On the Statistical Calibration of Physical Models".
International Journal for Chemical Kinetics, 47(4): pp 246–276, 2015.



Model Error – Likelihood options

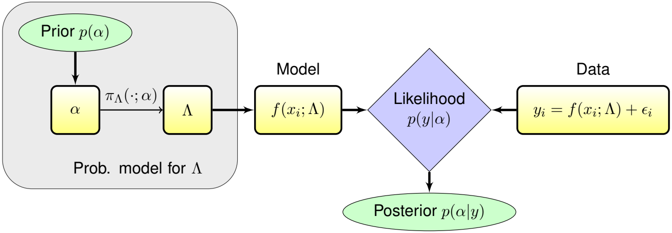
K. Sargsyan, H. Najm, and R. Ghanem, "On the Statistical Calibration of Physical Models".
International Journal for Chemical Kinetics, 47(4): pp 246–276, 2015.



- Full Likelihood: $L(\alpha) = p(y|\alpha) = p(y_1, \ldots, y_N|\alpha) = \pi(y)$
  - Degenerate if no data noise
  - Requires multivariate KDE or high-d integration
  - Gaussian approximation:
    $L(\alpha) \propto \exp\left(-\frac{1}{2}(y - \mu(\alpha))^T \Sigma^{-1}(\alpha)(y - \mu(\alpha))\right)$
  - NISP PC relieves the expense and provides easy access to mean $\mu(\alpha)$ and covariance $\Sigma(\alpha)$

Model Error – Likelihood options

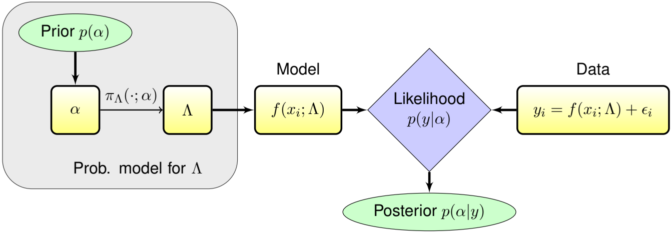
K. Sargsyan, H. Najm, and R. Ghanem, "On the Statistical Calibration of Physical Models". *International Journal for Chemical Kinetics*, 47(4): pp 246–276, 2015.



- Marginalized Likelihood: $L(\alpha) = p(y|\alpha) \approx \prod_{i=1}^N p(y_i|\alpha) = \prod_{i=1}^N \pi(y_i)$
  - Requires univariate KDE
  - Neglects built-in correlations
  - Gaussian approximation:
    $L(\alpha) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^N \Sigma_{ii}^{-1}(\alpha)(y_i - \mu_i(\alpha))^2\right)$

Model Error – Likelihood options

K. Sargsyan, H. Najm, and R. Ghanem, "On the Statistical Calibration of Physical Models". *International Journal for Chemical Kinetics*, 47(4): pp 246–276, 2015.



- Approximate Bayesian Computation (ABC): $L(\alpha) = \frac{1}{\epsilon} K\left(\frac{\rho(\mathcal{S}_\mathcal{M},\mathcal{S}_\mathcal{D})}{\epsilon}\right)$
  - Mean of $f(x_i;\Lambda)$ is "centered" on the data
  - The width of the distribution of $f(x_i;\Lambda)$ is consistent with the spread of the data around the nominal model prediction

$$L(\alpha) \propto \exp\left(-\frac{1}{2\epsilon^2}\sum_{i=1}^{N}\left[(\mu_i(\alpha)-y_i)^2 + \left(\sqrt{\Sigma_{ii}(\alpha)} - \gamma|\mu_i(\alpha)-y_i|\right)^2\right]\right)$$

Model Error – Predictions

$$f(x;\Lambda) = f\left(x; \sum_k \alpha_k \Psi_k(\xi)\right) = \sum_k f_k(x;\alpha)\Psi_k(\xi)$$

- Non-intrusive spectral projection (NISP) will be employed for
  - Likelihood computation
  - Posterior/pushed-forward predictions
  - Easy access to first two moments:

$$\mu(x;\alpha) = f_0(x;\alpha), \qquad \sigma^2(x;\alpha) = \sum_{k>0} f_k^2(x;\alpha)||\Psi_k||^2$$

- Predictive mean $\mathbb{E}[y(x) = \mathbb{E}_\alpha[\mu(x;\alpha)]$
- Decomposition of predictive variance

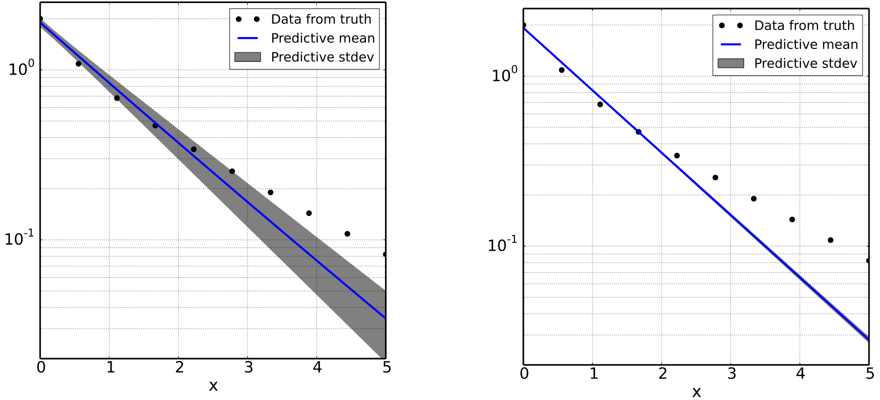
$$\mathbb{V}[y(x)] = \underbrace{\mathbb{E}_\alpha[\sigma^2(x;\alpha)]}_{\text{Model error}} + \underbrace{\mathbb{V}_\alpha[\mu(x;\alpha)] + \sigma_d^2}_{\text{Poserior/Data error}}$$

Predictions account for model error

Calibrating single-exponential models
with data from a double exponential model $g(x) = e^{-0.5x} + e^{-2x}$

Linear-exponential $f(x,\lambda) = e^{\lambda_1 + \lambda_2 x}$

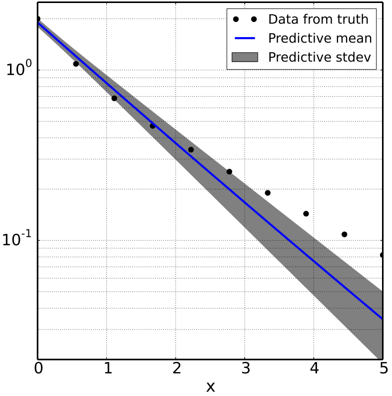
Additive Gaussian error



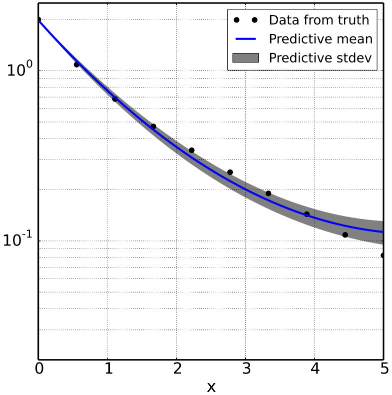
Predictions account for model error

Calibrating single-exponential models
with data from a double exponential model $g(x) = e^{-0.5x} + e^{-2x}$

Linear-exponential $f(x, \lambda) = e^{\lambda_1 + \lambda_2 x}$



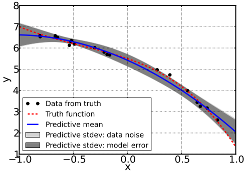
Quadratic-exponential $f_2(x, \lambda) = e^{\lambda_1 + \lambda_2 x + \lambda_3 x^2}$



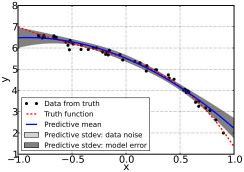
More data leads to ‘leftover’ model error

Calibrating a quadratic $f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2$
w.r.t. ‘truth’ $g(x) = 6 + x^2 - 0.5(x+1)^{3.5}$ measured with noise $\sigma = 0.1$.

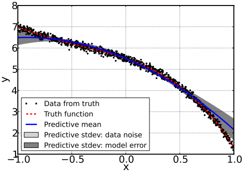
$N = 20$



$N = 50$

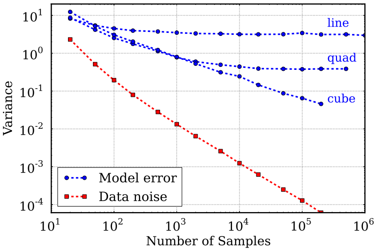


$N = 1000$



Summary of features:

- Well-defined model-to-model calibration
- Model-driven discrepancy correlations
- Respects physical constraints
- Disambiguates model and data errors
- Calibrated predictions of multiple QoIs



Chemistry problem – ABC

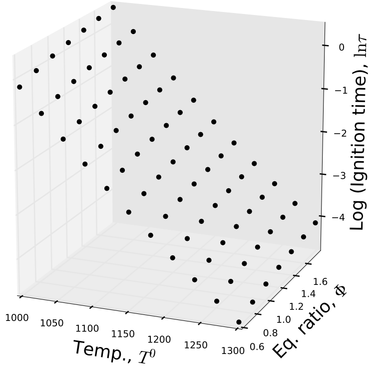
- Homogeneous ignition, methane-air mixture
- Single-step global reaction model calibrated against a detailed chemical kinetic model

- Data: ignition time; range of initial T & equivalence ratio
- Single-step model:

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$$

$$\begin{aligned} \mathfrak{R} &= [CH_4][O_2]k_f \\ k_f &= A\exp(-E/R^oT) \end{aligned}$$

- $(\ln A, E) = \sum_k \boldsymbol{\alpha}_k \Psi_k(\boldsymbol{\xi})$

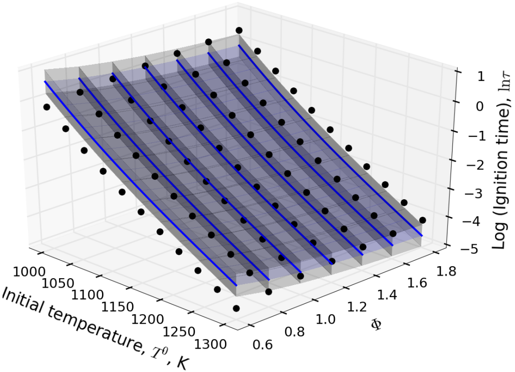


Quality of Uncertain Calibrated Model Predictions

Calibrated uncertain fit model is consistent with the detailed-model data.

Over the range of (T^0, Φ):

- MAP predictive mean ignition-time is centered on the data
- MAP predictive stdv is consistent with the scatter of the data



K. Sargsyan, H.N. Najm, and R. Ghanem
"On the Statistical Calibration of Physical Models"
Int. J. Chem. Kin., 47(4): 246-276, 2015

TransCom3 Experiment of CO_2 Flux Inversion

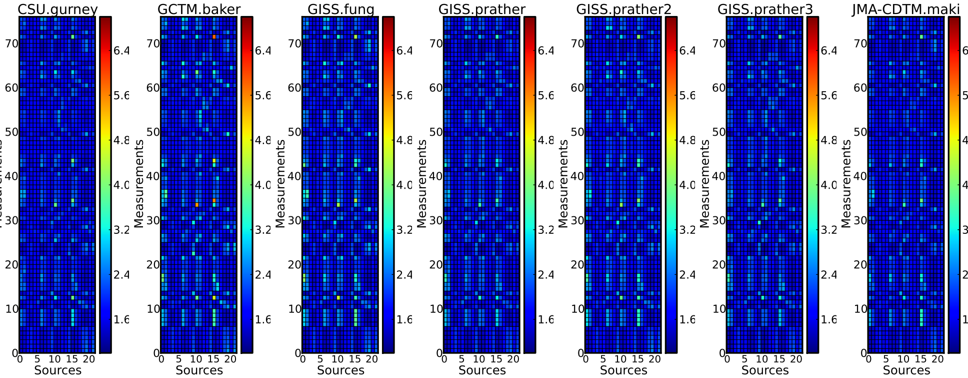
[Gurney *et al.*, Tellus B, 2003]

- Observations $\mathbf{d}$ at $N = 77$ sites around the world
- Inverse problem: find fluxes $\mathbf{s}$ at $M = 22$ locations
- Linearized 'response' model $\mathbf{R}$, such that $\mathbf{d} \approx \mathbf{R}\mathbf{s}$

$$\mathbf{d} = \mathbf{R}\mathbf{s} + \boldsymbol{\epsilon}_\mathbf{d}$$

- Model $\mathbf{R}$ is never perfect thus contaminating the inversion
- The inferred values of $\mathbf{s}$ compensate for model deficiencies
- $\boldsymbol{\epsilon}_\mathbf{d}$ is meant to capture data errors, but is 'entangled' with model errors

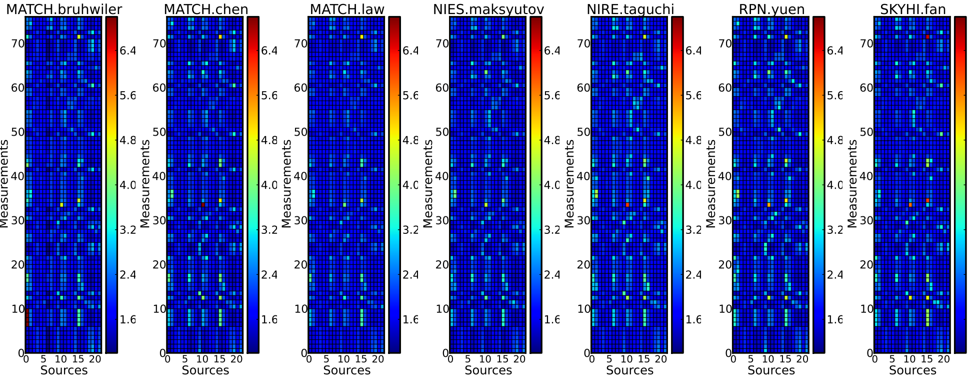
Consider 14 different response models $\mathbf{R}$



Infer fluxes $\mathbf{s}$, given measurements $\mathbf{d}$ to satisfy $\mathbf{d} \approx \mathbf{R}\mathbf{s}$

- Conventional additive Gaussian error (least-squares): $\mathbf{d} = \mathbf{R}\mathbf{s} + \xi$
- Embed probabilistic model for fluxes s: $\mathbf{d} = \mathbf{R}(\mu_\mathbf{s} + \mathbf{C}_\mathbf{s}\xi)$

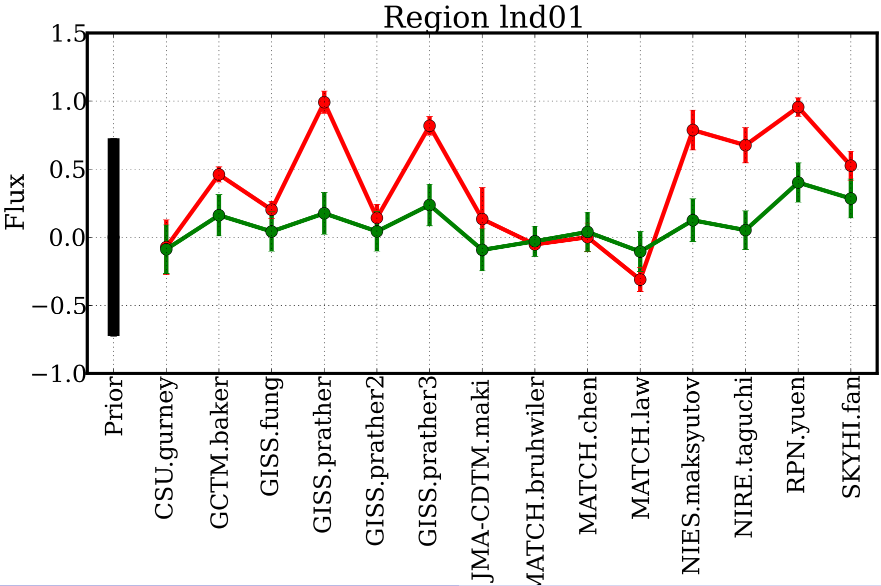
Consider 14 different response models $\mathbf{R}$



Infer fluxes $\mathbf{s}$, given measurements $\mathbf{d}$ to satisfy $\mathbf{d} \approx \mathbf{R}\mathbf{s}$

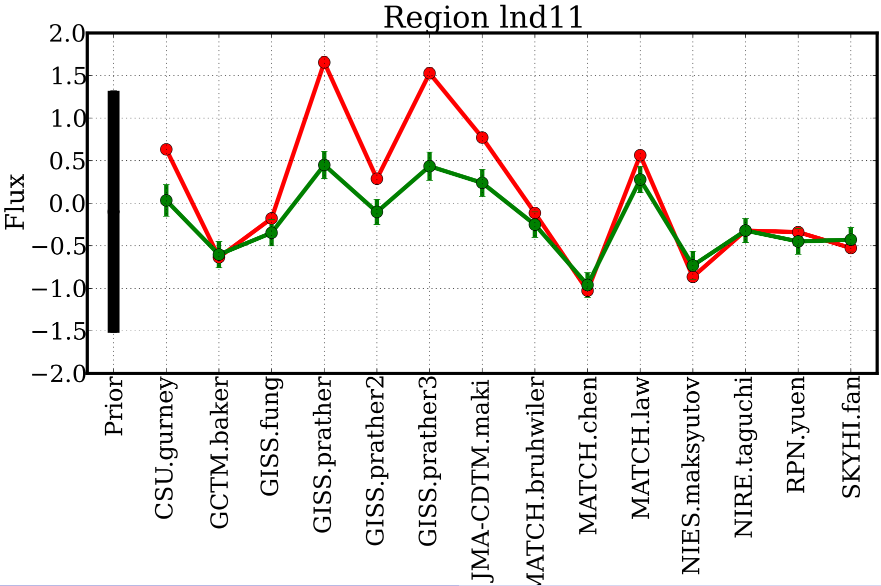
- Conventional additive Gaussian error (least-squares): $\mathbf{d} = \mathbf{R}\mathbf{s} + \xi$
- Embed probabilistic model for fluxes s: $\mathbf{d} = \mathbf{R}(\mu_\mathbf{s} + \mathbf{C}_\mathbf{s}\xi)$

Inferred fluxes show less variability across models

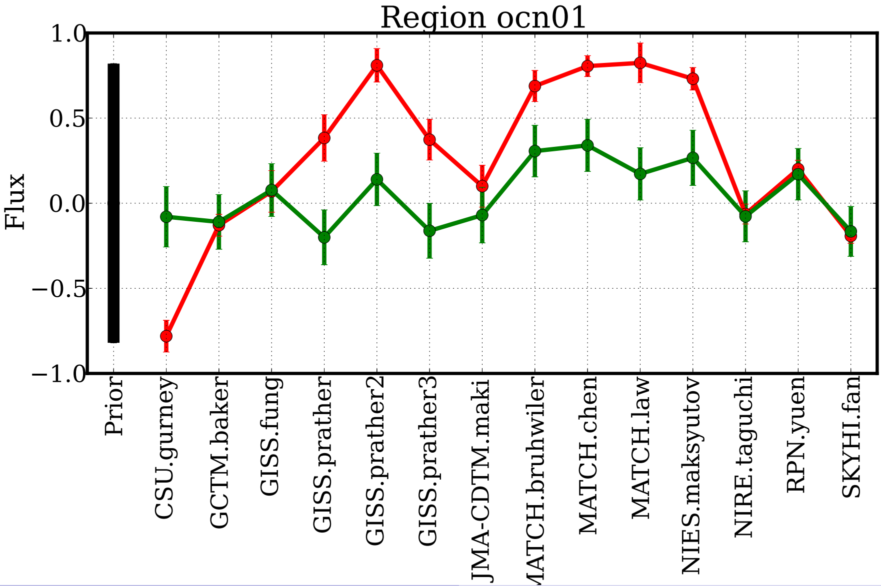


Inferred fluxes show less variability across models

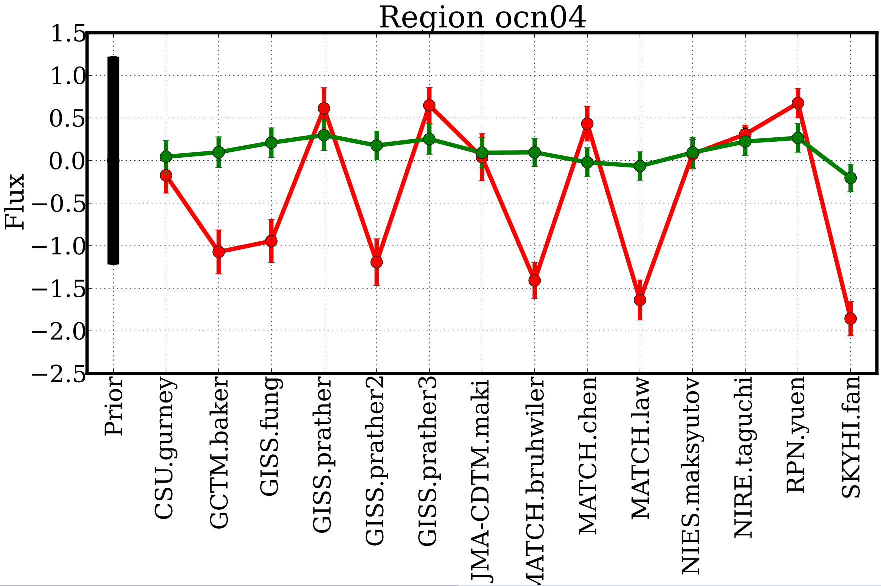
Inferred fluxes show less variability across models



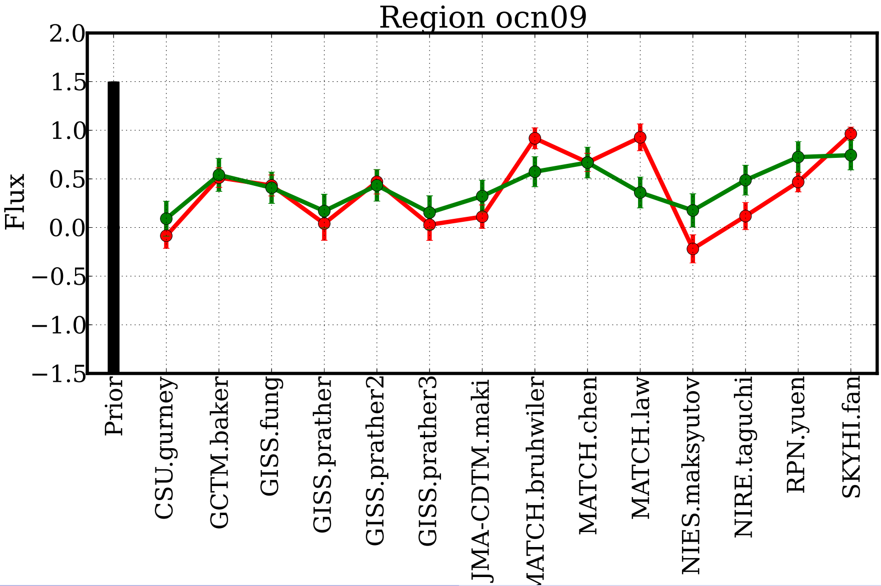
Inferred fluxes show less variability across models



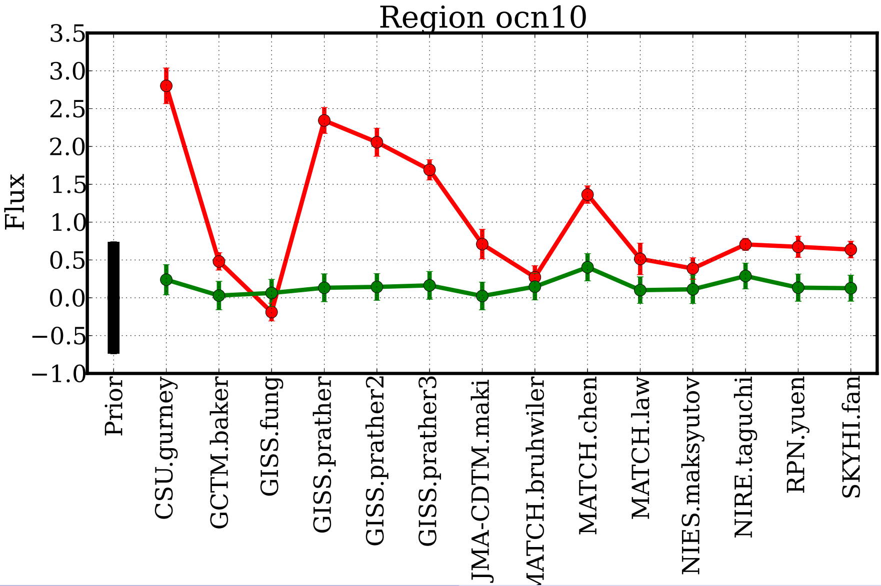
Inferred fluxes show less variability across models



Inferred fluxes show less variability across models



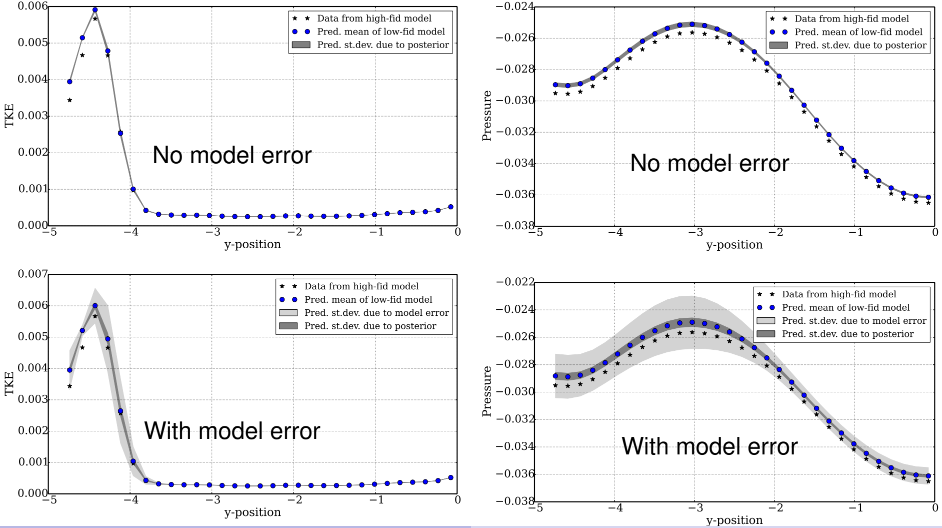
Inferred fluxes show less variability across models



Model error in LES: embed model err in Smagorinsky coefficient
Calibrate with TKE data, predict both TKE and Pressure

Pushed forward posterior

TKE Pressure



Summary

- Forward UQ: Polynomial Chaos representation of RVs
  - Non-intrusive spectral projection
  - Surrogate construction, Bayesian regression
  - **High-D challenge: sparse PC via Bayesian compressive sensing**

- Intrusive spectral projection
- Time/space-resolved processes (Karhunen-Loeve expansions)
- Non-polynomial regression (Radial Basis Functions, Gaussian Processes)
- Rosenblatt transform, Kernel Density Estimation
- Domain decomposition, multiwavelets

Summary

- Inverse UQ: Bayesian inference for parameter estimation
  - Bayesian parameter estimation
  - **Model error quantification: embedded model error approach**

- Markov chain Monte Carlo (MCMC) details
- Model plausibility theory: evidence, model selection, Bayes factors
- MaxEnt methods, data-free inference

Literature

Thank you!

General PC

N. Wiener, “Homogeneous Chaos”, *American Journal of Mathematics*, 60(4), 897-936, (1938).

Ghanem, R., Spanos, P., “Stochastic Finite Elements: A Spectral Approach”, Springer Verlag, (1991).

Xiu, D., Karniadakis, G., “The Wiener-Askey Polynomial Chaos for Stocahstic Differential Equations”, *SIAM J. Sci. Comp.*, 24(2), 619-644, (2002).

Le Maître, O., Knio, O., “Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics”, Springer-Verlag, (2010).

Najm, H., “Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics”, *Ann. Rev. Fluid Mech.*, 41(1):35-52, (2009).

Xiu, D., “Numerical Methods for Stochastic Computations: A Spectral Method Approach”, Princeton U. Press (2010).

Debusschere, B., Najm, H., Pébay, P., Knio, O., Ghanem, R., Le Maître, O., “Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Processes”, *SIAM J. Sci. Comp.*, 26(2):698-719, (2004).

Marzouk, Y., Najm, H., “Dimensionality Reduction and Polynomial Chaos Acceleration of Bayesian Inference in Inverse Problems”, *J. Comp. Phys.*, 228(6):1862-1902, (2009).

Bayesian compressive sensing

S. Ji, Y. Xue and L. Carin, “Bayesian Compressive Sensing”, *IEEE Trans. Signal Proc.*, 56(6), (2008).

K. Sargsyan, C. Safta, H. Najm, B. Debusschere, D. Ricciuto, P. Thornton, “Dimensionality reduction for complex models via Bayesian compressive sensing”, *Int. J. Uncertainty Quantification*, 4(1), 63-93, (2014).

Model error

M. Kennedy, M. and A. O’Hagan, “Bayesian calibration of computer models”, *Journal of the Royal Statistical Society*, Series B. 63, 425-464, (2001).

K. Sargsyan, H. N. Najm, R. Ghanem, “On the Statistical Calibration of Physical Models”, *Int. J. Chem. Kinetics*, 47(4), 246-276, (2015).

Additional Material (Core Dump)

Probabilistic Forward UQ & Polynomial Chaos Representation of Random Variables

With $y = f(x)$, x a random variable, estimate the RV y

- Can describe a RV in terms of its
  - density, moments, characteristic function, or
  - as a function on a probability space
- Constraining the analysis to RVs with finite variance
  - $\Rightarrow$ Represent RV as a spectral expansion in terms of orthogonal functions of standard RVs
    - Polynomial Chaos Expansion
- Enables the use of available functional analysis methods for forward UQ

Sensitivity indices are directly computable from PC

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

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

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

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

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

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

Variance contributions

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

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

Main effect sensitivities $\boxed{\xi_1}$ ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \boxed{c_7^2\langle\psi_2^2\rangle} + \; c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_9^2\langle\psi_2^2\rangle$$

Main effect sensitivities ξ_1 $\boxed{\xi_2}$ ξ_3

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

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

Main effect sensitivities $\xi_1 \quad \xi_2 \quad \boxed{\xi_3}$

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

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

Total sensitivities $\boxed{\xi_1}$ ξ_2 ξ_3

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

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

Total sensitivities ξ_1 $\boxed{\xi_2}$ ξ_3

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; \boxed{c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \; c_7^2\langle\psi_2^2\rangle \; + \; \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} + \boxed{c_9^2\langle\psi_2^2\rangle}$$

Total sensitivities ξ_1 ξ_2 $\boxed{\xi_3}$

Sensitivity indices are directly computable from PC

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

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) $\quad (\xi_1, \xi_3)$ $\quad (\xi_2, \xi_3)$

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

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

Joint sensitivities (ξ_1, ξ_2) $\boxed{(\xi_1, \xi_3)}$ (ξ_2, ξ_3)

Sensitivity indices are directly computable from PC

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

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

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

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

Variance contributions

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

$$+ \; c_4^2\langle\psi_2^2\rangle \; + \; c_5^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_6^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle \; + \; c_7^2\langle\psi_2^2\rangle \; + \; \boxed{c_8^2\langle\psi_1^2\rangle\langle\psi_1^2\rangle} \; + \; c_9^2\langle\psi_2^2\rangle$$

Joint sensitivities (ξ_1, ξ_2) (ξ_1, ξ_3) $\boxed{(\xi_2, \xi_3)}$

Other non-intrusive methods (stochastic collocation)

- Interpolation: Fit interpolant to samples
  - Oscillation concern in multi-D
- Regression: Estimate best-fit response surface
  - Least-squares
    - Sparsity via ℓ_1 constraints; compressive sensing
  - Bayesian inference
    - Sparsity via Laplace priors; Bayesian compressive sensing
  - Useful when quadrature methods are infeasible, e.g.:
    - Samples given *a priori*
    - Can't choose sample locations
    - Can't take enough samples
    - Forward model is noisy

PCE Construction for Noisy Functions

- Quadrature formulae presume a degree of smoothness
  - No convergence for a noisy function

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

- Sparse-Quadrature formulae are *ill-conditioned* and highly-sensitive to noise
  - No convergence with order
  - Error grows with increased dimensionality
- Options in the presence of noise:
  - RMS fitting for PC coefficients
  - Bayesian inference of PC coefficients

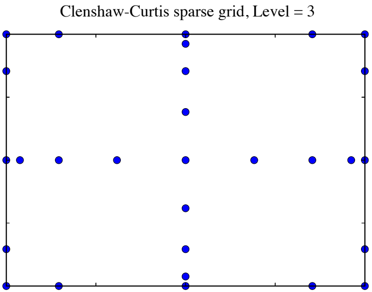
PC and High-Dimensionality

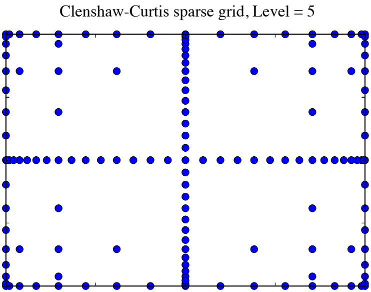
Dimensionality n of the PC basis: $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_n\}$

- $n \approx$ number of uncertain parameters
- $P + 1 = (n+p)!/n!p!$ grows fast with n

Impacts:

- Size of intrusive PC system
- Hi-D projection integrals $\Rightarrow$ large # non-intrusive samples
  - Sparse quadrature methods





PC coefficients via sparse regression

PCE:

$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(x)$$

with $x \in \mathbb{R}^n$, Ψ_k max order p, and $K = (p+n)!/p!/n!$

- N samples $(x_1, y_1), \ldots, (x_N, y_N)$
- Estimate K terms $c_0, \ldots, c_{K-1}$, s.t.

$$\min ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}||_2^2$$

where $\boldsymbol{y} \in \mathbb{R}^N$, $\boldsymbol{c} \in \mathbb{R}^K$, $\boldsymbol{A}_{ik} = \Psi_k(x_i)$, $\boldsymbol{A} \in \mathbb{R}^{N \times K}$

With $N << K \Rightarrow$ under-determined

- Need some form of regularization

Regularization – Compressive Sensing (CS)

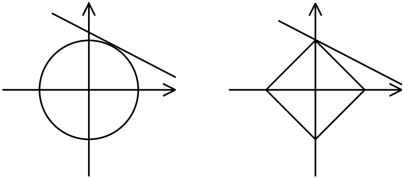
- ℓ_2-norm — Tikhonov regularization; Ridge regression:

$$\min \{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2 + \|\boldsymbol{c}\|_2^2\}$$

- ℓ_1-norm — Compressive Sensing; LASSO; basis pursuit

$$\begin{aligned} \min \{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2 + \|\boldsymbol{c}\|_1\} & \\ \min \{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2\} & \quad \text{subject to } \|\boldsymbol{c}\|_1 \leq \epsilon \\ \min \{\|\boldsymbol{c}\|_1\} & \quad \text{subject to } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2 \leq \epsilon \end{aligned}$$

$\Rightarrow$ discovery of sparse signals



Bayesian Regression

- Bayes formula
$$p(\boldsymbol{c}|D) \propto p(D|\boldsymbol{c})\pi(\boldsymbol{c})$$
- Bayesian regression: prior as a regularizer, *e.g.*
  - Log Likelihood $\Leftrightarrow$ $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2$
  - Log Prior $\Leftrightarrow$ $\|\boldsymbol{c}\|_p^p$
- Laplace sparsity priors $\pi(c_k|\alpha) = \frac{1}{2\alpha}e^{-|c_k|/\alpha}$
- LASSO (Tibshirani 1996) ... formally:
$$\min \left\{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}\|_2^2 + \lambda\|\boldsymbol{c}\|_1\right\}$$
  Solution $\sim$ the posterior mode of $\boldsymbol{c}$ in the Bayesian model
$$y \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{c}, I_N), \qquad c_k \sim \frac{1}{2\alpha}e^{-|c_k|/\alpha}$$
- Bayesian LASSO (Park & Casella 2008)

Bayesian Compressive Sensing (BCS)

- BCS (Ji 2008; Babacan 2010)— hierarchical priors:
  - Gaussian priors $\mathcal{N}(0, \sigma_k^2)$ on the c_k
  - Gamma priors on the σ_k^2

$\Rightarrow$ Laplace sparsity priors on the c_k

- Evidence maximization establishes ML estimates of the σ_k
  - many of which are found $\approx 0 \quad \Rightarrow \quad c_k \approx 0$
  - iteratively include terms that lead to the largest increase in the evidence
- iterative BCS (iBCS) (Sargsyan 2012):
  - adaptive iterative order growth
  - BCS on order-p Legendre-Uniform PC
  - repeat with order-$p+1$ terms added to surviving p-th order terms

Random Fields

- A random variable is a function on an event space Ω
  - No dependence on other coordinates –*e.g.* space or time
- A random field is a function on a product space $\Omega \times D$
  - *e.g.* sea surface temperature $T_{\mathrm{SS}}(z, \omega)$, $z \equiv (\boldsymbol{x}, t)$
- It is a more complex object than a random variable
  - A combination of an infinite number of random variables
- In many physical systems, uncertain field quantities, described by random fields:
  - are smooth, *i.e.*
  - they have an underlying *low dimensional structure*

  due to large correlation length-scales

Random Fields – KLE

- Smooth random fields can be represented with a small no. of stochastic degrees of freedom
- A random field $M(x, \omega)$ with
  - a mean function: $\mu(x)$
  - a continuous covariance function: $C(x_1, x_2) = \langle [M(x_1, \omega) - \mu(x_1)][M(x_2, \omega) - \mu(x_2)] \rangle$

  can be represented with the Karhunen-Loeve Expansion (KLE)

  $$M(x, \omega) = \mu(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \eta_i(\omega) \phi_i(x)$$

  where
  - λ_i and $\phi_i(x)$ are the eigenvalues and eigenfunctions of the covariance function $C(\cdot, \cdot)$
  - η_i are uncorrelated zero-mean unit-variance RVs
- KLE $\Rightarrow$ representation of random fields using PC

Intrusive PC UQ: A direct *non-sampling* method

- Given model equations: $\mathcal{M}(u(\boldsymbol{x},t);\lambda)=0$
- Express uncertain parameters/variables using PCEs

$$u=\sum_{k=0}^{P} u_k \Psi_k; \quad \lambda=\sum_{k=0}^{P} \lambda_k \Psi_k$$

- Substitute in model equations; apply Galerkin projection
- New set of equations: $\mathcal{G}(U(\boldsymbol{x},t),\Lambda)=0$
  - with $U=[u_0,\dots,u_P]^T$, $\Lambda=[\lambda_0,\dots,\lambda_P]^T$
- Solving this <u>deterministic</u> system <u>once</u> provides the full specification of uncertain model ouputs

Intrusive Galerkin PC ODE System

$$\boxed{\frac{du}{dt} = f(u;\lambda)}$$

$$\lambda = \sum_{i=0}^{P} \lambda_i \Psi_i \qquad u(t) = \sum_{i=0}^{P} u_i(t)\Psi_i$$

$$\boxed{\frac{du_i}{dt} = \frac{\langle f(u;\lambda)\Psi_i\rangle}{\langle \Psi_i^2 \rangle} \qquad i = 0, \ldots, P}$$

Say $f(u;\lambda) = \lambda u$, then

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

where the tensor $C_{pqi} = \langle \Psi_p \Psi_q \Psi_i \rangle / \langle \Psi_i^2 \rangle$ is readily evaluated

Intrusive PC UQ Pros/Cons

Cons:

- Reformulation of governing equations
- New discretizations
- New numerical solution method
  - Consistency, Convergence, Stability
  - Global vs. multi-element local PC constructions
- New solvers and model codes
  - Opportunities for automated code transformation
- New preconditioners

Pros:

- Tailored solvers <u>can</u> deliver superior performance

Model Evidence and Complexity

Let $\mathcal{M} = \{M_1, M_2, \ldots\}$ be a set of models of interest

- Parameter estimation from data is conditioned on the model

$$p(\theta|D, M_k) = \frac{p(D|\theta, M_k)\pi(\theta|M_k)}{p(D|M_k)}$$

Evidence (marginal likelihood) for M_k:

$$p(D|M_k) = \int p(D|\theta, M_k)\pi(\theta|M_k)\mathrm{d}\theta$$

Model evidence is useful for model selection

- Choose model with maximum evidence
- Compromise between fitting data and model complexity
  - Optimal complexity – Occam's razor principle
  - Avoid overfitting

Too much model complexity leads to overfitting

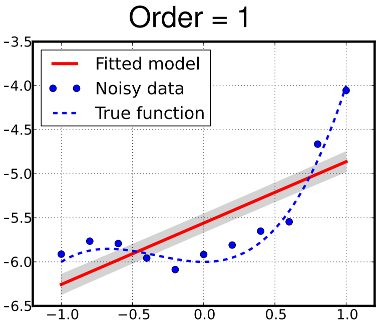
Data model: $i = 1, \dots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \dots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

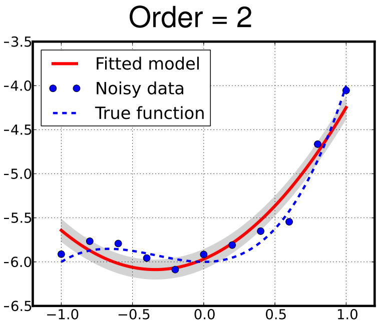
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

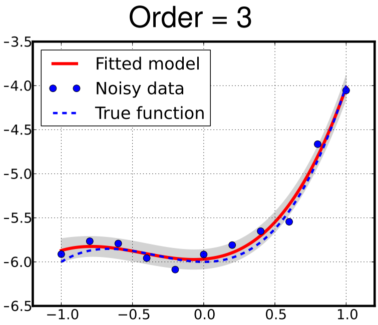
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

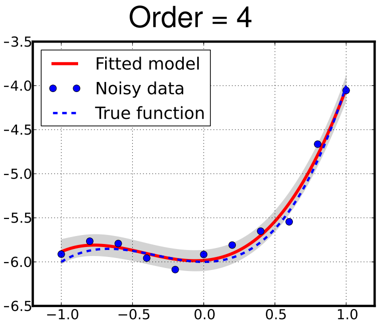
Data model: $\quad i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

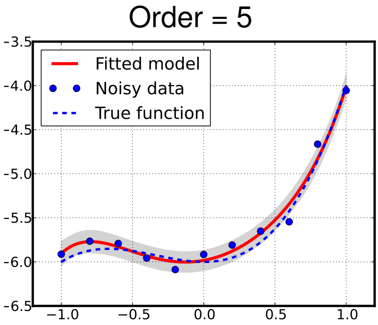
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

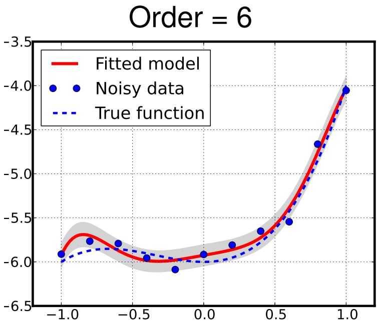
Data model: $i = 1, \ldots, N$

$$
\begin{aligned}
y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\
\epsilon_i &\sim N(0, s)
\end{aligned}
$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

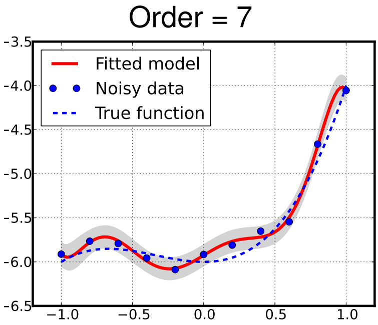
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

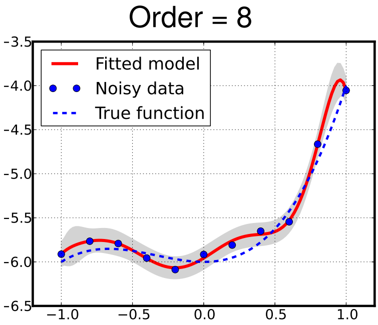
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

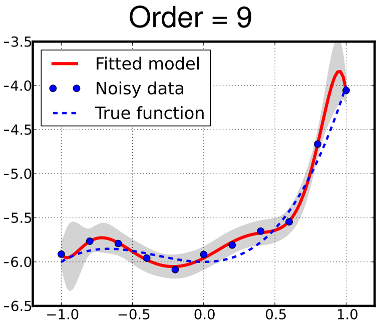
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Too much model complexity leads to overfitting

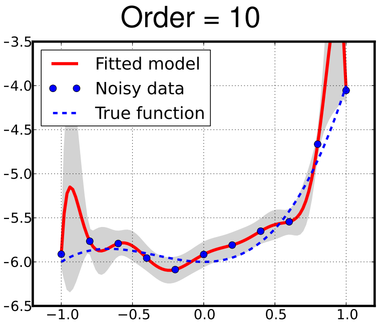
Data model: $i = 1, \ldots, N$

$$\begin{aligned} y_i &= x_i^3 + x_i^2 - 6 + \epsilon_i \\ \epsilon_i &\sim N(0, s) \end{aligned}$$

Bayesian regression with Legendre PCE fit models, order 1-10

$$y_m = \sum_{k=0}^{P} c_k \psi_k(x)$$

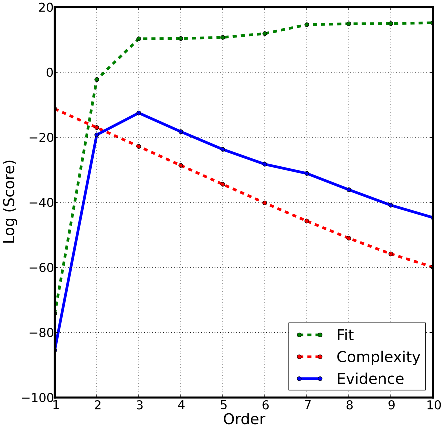
Uniform priors $\pi(c_k)$, $k = 0, \ldots, P$



Fitted model pushed-forward posterior versus the data

Evidence and Cross-Validation Error

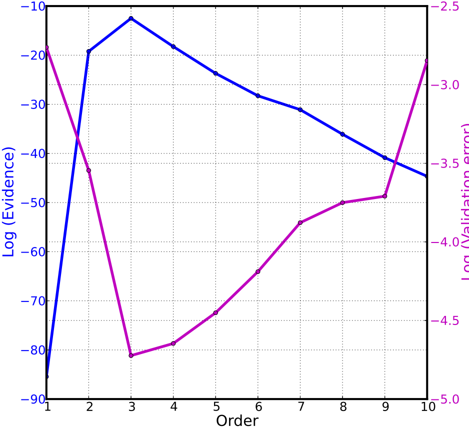
- Model evidence peaks at the true polynomial order of 3
- Cross validation error is equally minimal at order 3
- Models with optimal complexity are robust to cross validation



Log evidence: sum of two scores, balances complexity & fit

Evidence and Cross-Validation Error

- Model evidence peaks at the true polynomial order of 3
- Cross validation error is equally minimal at order 3
- Models with optimal complexity are robust to cross validation



Cross validation error and model evidence versus order

Challenges in PC UQ – High-Dimensionality

- Dimensionality n of the PC basis: $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_n\}$
  - number of degrees of freedom
  - $P + 1 = (n + p)!/n!p!$ grows fast with n
- Impacts:
  - Size of intrusive system
  - # non-intrusive (sparse) quadrature samples
- Generally $n \approx$ number of uncertain parameters
- Reduction of n:
  - Sensitivity analysis
  - Dependencies/correlations among parameters
  - Dominant eigenmodes of random fields
  - Manifold learning: Isomap, Diffusion maps
  - Sparsification: Compressed Sensing, LASSO

High dimensionality challenge – Forward UQ

Consider a forward model

$$y = f(x)$$

Let $x \in \mathbb{R}^n$ be uncertain, represented as a random vector,

$$x \sim p(x)$$

Estimate moments of y

$$\mathcal{M}^q = \int [f(x)]^q p(x) \mathrm{d}x$$

Forward UQ is an integration problem.

Integration in High Dimensions

- Monte Carlo (MC) methods
  - well suited for high-D integrals – convergence rate independent of dimensionality
  - nonetheless they require large numbers of samples for good accuracy
- Quadrature
  - Tensor product quadrature is useless in hi-D
    - Say m points in each of n dimensions: m^n points
  - Adaptive sparse quadrature
    - Much more feasible
    - Can beat MC – dep. on smoothness of integrand
  - Greedy algorithms
- Dimensionality reduction
  - Low rank and sparse representations
  - Global sensitivity analysis

High dimensionality challenge – Inverse UQ

- Bayesian inference in a computational setting relies on Markov Chain Monte Carlo (MCMC) methods
- MCMC: A random walk algorithm for generation of samples from the *posterior* density on model inputs
  - Moments are evaluated from the random samples
- Need many random sample evaluations of forward model
  - Employ model surrogates built via forward UQ
  - *Adaptive local surrogates*
- High dimensionality can lead to poor performance
  - local maxima
  - many directions uninformed by data
  - choice of proposal density
  - *Dimension-Adaptive Likelihood-Informed* MCMC

Bayesian inference – High Dimensionality Challenge

- Judgement on local/global posterior peaks is difficult
  - Multiple chains; Tempering
- Choosing a good starting point is very important
  - An initial optimization strategy is useful, albeit not trivial
- Choosing good MCMC proposals, and attaining good mixing
  - Likelihood-informed
    - Markov jump in those dimensions informed by data
    - Sample from prior in complement of dimensions
    - Adaptive proposal learning from MCMC samples
    - Log-Posterior Hessian $\Rightarrow$ local Gaussian approx.
    - Adaptive, Geometric, Langevin MCMC
  - Dimension independent
    - Proposal design: good MCMC performance in hiD
  - Literature: A. Stuart, M. Girolami, K. Law, T. Cui, Y. Marzouk

(Law 2014; Cui *et al.*, 2014,2015; Cotter *et al.*, 2013)

Curse of Dimensionality

- (Dim-adaptive) Sparse quadrature integration [Gerstner, 2003]
- High Dimensional Model Representation [Rabitz & Alis, 1999]
  - would not handle strong nonlinearities
  - tried cut-HDMR in a chemical kinetics context: fails!
- Proper Generalized Decomposition [Nuoy, 2010]

- Turn it into the *blessing of dimensionality* [Donoho, 2000]
- Compressive sensing in spectral methods [Doostan *et al.*, 2009]
- Bayesian compressive sensing [Ji *et al.*, 2008]

Curse of Dimensionality

- (Dim-adaptive) Sparse quadrature integration [Gerstner, 2003]
- High Dimensional Model Representation [Rabitz & Alis, 1999]
  - would not handle strong nonlinearities
  - tried cut-HDMR in a chemical kinetics context: fails!
- Proper Generalized Decomposition [Nuoy, 2010]

- Turn it into the *blessing of dimensionality* [Donoho, 2000]
- Compressive sensing in spectral methods [Doostan *et al.*, 2009]
- Bayesian compressive sensing [Ji *et al.*, 2008]

short answer: no free lunch

Challenges in PC UQ – Non-Linearity

- Bifurcative response at critical parameter values
  - Rayleigh-Bénard convection
  - Transition to turbulence
  - Chemical ignition
- Discontinuous $u(\lambda(\boldsymbol{\xi}))$
  - Failure of global PCEs in terms of smooth $\Psi_k()$
  - $\Leftrightarrow$ failure of Fourier series in representing a step function
- Local PC methods
  - Subdivide support of $\lambda(\boldsymbol{\xi})$ into regions of smooth $u \circ \lambda(\boldsymbol{\xi})$
  - Employ PC with compact support basis on each region
  - A spectral-element vs. spectral construction
  - Domain mapping

Discontinuities/Nonlinearities/Bifurcations

- Stochastic domain decomposition
  - Wiener-Haar expansions,
    Multiblock expansions,
    Multiwavelets, [Le Maître *et al*, 2004,2007]
  - also known as Multielement PC [Wan & Karniadakis, 2009]
- Data domain decomposition [Sargsyan *et al*, 2009,2010]
  - Data clustering, classification
  - Mixture PC expansions

- Adaptive setting helps
- Does not scale with dimensionality
- For expensive models, can not split much
- *Need a 'smart' domain decomposition*

Challenges in PC UQ – Time Dynamics

- Systems with limit-cycle or chaotic dynamics
- Large amplification of phase errors over long time horizon
- PC order needs to be increased in time to retain accuracy
- Time shifting/scaling remedies

- Futile to attempt representation of detailed turbulent velocity field $\boldsymbol{v}(\boldsymbol{x}, t; \lambda(\boldsymbol{\xi}))$ as a PCE
  - Fast loss of correlation due to energy cascade
  - Problem studied in 60's and 70's
- Focus on flow statistics, *e.g.* Mean/RMS quantities
  - Well behaved
  - Argues for non-intrusive methods with DNS/LES of turbulent flow

Model Complexity challenge

- If a single model run is a challenge then UQ is infeasible
- Most physical model output quantities of interest depend on only a "small" number of parameters, however:
  - Global sensitivity analysis itself requires many samples
  - Even after reduction of dimensionality to, say, 5 parameters, O(100) samples may be necessary
- Large number of independent samples
  - ideally suited for HPC
- Multifidelity UQ methods are useful – forward UQ
  - Use combinations of many low-resolution/low-fidelity runs with a few high-resolution/high-fidelity runs
- Parallel MCMC methods – inverse UQ

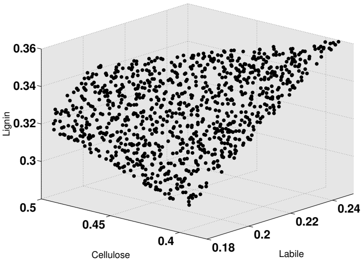
Data Scarcity Challenge

- Even in a “big-Data” context, it’s common to find no information in the data on many *big-model* parameters
  - Situation is typical in statistical inversion for field quantities
  - Bayesian inference of optimal random field constructions
  - Use adaptive MCMC methods that focus on data-informed parameters
- Usually, raw data is not published
  - Published “data” is essentially processed data products, being statistics on
    - the data, or functions of fitted model parameters
  - Use Maximum-Entropy and Approximate Bayesian Computation (ABC) methods – DFI
    - Discover posterior density on model parameters consistent with published statistics

Input correlations: Rosenblatt transformation

- Rosenblatt transformation maps any (not necessarily independent) set of random variables $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ 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 $\boldsymbol{\xi} = R^{-1}(\eta)$ ensures a well-defined quadrature integration to build PC [Sargsyan *et al.*, 2010]

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

- Caveat: if only samples of $\boldsymbol{\xi}$ are available, the conditional distributions are hard to evaluate accurately.