

# *High-Dimensional Sparse Surrogate Construction via Bayesian Compressive Sensing*

K. Chowdhary<sup>1</sup>,  
K. Sargsyan<sup>1</sup>, C. Safta<sup>1</sup>, D. Ricciuto<sup>2</sup>,  
B. Debusschere<sup>1</sup>, H. Najm<sup>1</sup>, P. Thornton<sup>2</sup>

<sup>1</sup>Sandia National Laboratories  
Livermore, CA

<sup>2</sup>Oak Ridge National Laboratory  
Oak Ridge, TN

*Sponsored by DOE, Biological and Environmental Research,  
under Accelerated Climate Modeling for Energy (ACME).*

*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

- **Surrogates** needed for complex models
- **Polynomial Chaos (PC)** surrogates do well with uncertain inputs
- **Bayesian regression** provide results with uncertainty certificate
- **Compressive sensing** ideas deal with high-dimensionality

# Surrogate construction: scope and challenges

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

- Global sensitivity analysis
  - Optimization
  - Forward uncertainty propagation
  - Input parameter calibration
  - ...
- 
- 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  $f(\lambda)$  to enable

- Global sensitivity analysis
  - Optimization
  - Forward uncertainty propagation
  - Input parameter calibration
  - ...
- 
- 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  $f(\lambda)$  to enable

- Global sensitivity analysis
  - Optimization
  - Forward uncertainty propagation
  - Input parameter calibration
  - ...
- 
- 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

# Polynomial Chaos surrogate

- Build/presume PC for input parameter  $\lambda$

$$\lambda(\mathbf{x}) = \sum_{k=0}^{K-1} \mathbf{a}_k \Psi_k(\mathbf{x})$$

# Polynomial Chaos surrogate

- Build/presume PC for input parameter  $\lambda$

$$\lambda(\mathbf{x}) = \sum_{k=0}^{K-1} \mathbf{a}_k \Psi_k(\mathbf{x})$$

- E.g., gaussian with known moments  $\mu_i, \sigma_i$ ,

$$\lambda_i = \mu_i + \sigma_i x_i$$

# Polynomial Chaos surrogate

- Build/presume PC for input parameter  $\lambda$

$$\lambda(\mathbf{x}) = \sum_{k=0}^{K-1} \mathbf{a}_k \Psi_k(\mathbf{x})$$

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

$$\lambda_i = F_{\lambda_i}^{-1} \left( \frac{x_i + 1}{2} \right), \quad \text{for } i = 1, 2, \dots, d.$$



# Polynomial Chaos surrogate

- Build/presume PC for input parameter  $\lambda$

$$\lambda(\mathbf{x}) = \sum_{k=0}^{K-1} \mathbf{a}_k \Psi_k(\mathbf{x})$$

- If input parameters are uniform in  $[a_i, b_i]$ , then

$$\lambda_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} x_i$$

# Polynomial Chaos surrogate

- Build/presume PC for input parameter  $\lambda$

$$\lambda(\mathbf{x}) = \sum_{k=0}^{K-1} \mathbf{a}_k \Psi_k(\mathbf{x})$$

- If input parameters are uniform in  $[a_i, b_i]$ , then

$$\lambda_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} x_i$$

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

$$u = f(\lambda(\mathbf{x})) \quad u = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x}) \equiv g(\mathbf{x})$$

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

# Alternative methods to obtain PC coefficients

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

- Projection  $c_k = \frac{\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral  $\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int u(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$  can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

# Alternative methods to obtain PC coefficients

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

- Projection  $c_k = \frac{\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral  $\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int u(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$  can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

- Bayesian regression

$$P(c_k | u(\mathbf{x}_j)) \propto P(u(\mathbf{x}_j) | c_k) P(c_k)$$



any (number of) samples

# Alternative methods to obtain PC coefficients

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

- Projection  $c_k = \frac{\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral  $\langle u(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int u(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$  can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q u(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

- Bayesian regression

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



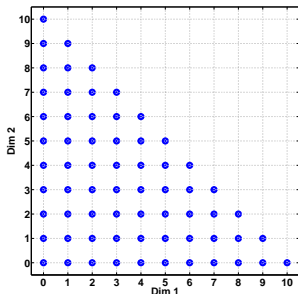
any (number of) samples

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

$$y = u(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- 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

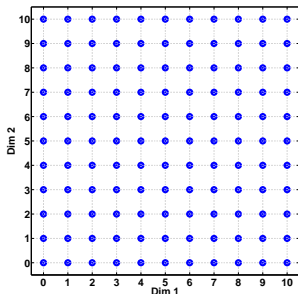


# Bayesian inference of PC surrogate: high-d, low-data regime

$$y = u(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- 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

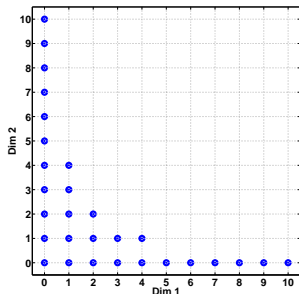


# Bayesian inference of PC surrogate: high-d, low-data regime

$$y = u(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- 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



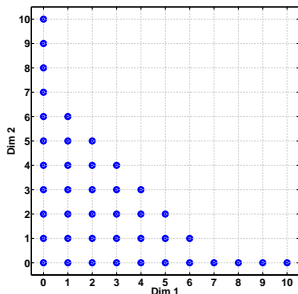


# Bayesian inference of PC surrogate: high-d, low-data regime

$$y = u(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- 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

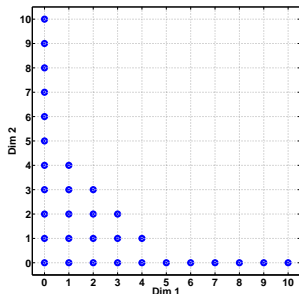


# Bayesian inference of PC surrogate: high-d, low-data regime

$$y = u(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- 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  $(\mathbf{x}_n, u_n)$  and  $K$  basis terms  $\Psi_k(\cdot)$
- Projection matrix  $\mathbf{P}^{N \times K}$  with  $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights  $\mathbf{c} = (c_0, \dots, c_{K-1})$  so that

$$\mathbf{u} \approx \mathbf{P}\mathbf{c}$$

or

$$u_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

- The number of polynomial basis terms grows fast; a  $p$ -th order,  $d$ -dimensional basis has a total of  $K = (p + d)! / (p!d!)$  terms.
- For limited data and large basis set ( $N < K$ ) this is a sparse signal recovery problem  $\Rightarrow$  need some regularization/constraints.
- Least-squares  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 \}$
- The 'sparsest'  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$
- Compressive sensing  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$

# In a different language....

- $N$  training data points  $(\mathbf{x}_n, u_n)$  and  $K$  basis terms  $\Psi_k(\cdot)$
- Projection matrix  $\mathbf{P}^{N \times K}$  with  $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights  $\mathbf{c} = (c_0, \dots, c_{K-1})$  so that

$$\mathbf{u} \approx \mathbf{P}\mathbf{c}$$

or

$$u_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

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

- Least-squares  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 \}$

- The 'sparsest'  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$

- Compressive sensing  $\text{argmin}_{\mathbf{c}} \{ \|\mathbf{u} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$   
Bayesian Likelihood Prior

# Weighted Bayesian Compressive Sensing

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- Evidence maximization dictates values for  $\sigma_k^2$ ,  $\alpha_k$ ,  $\sigma^2$  and allows exact Bayesian solution

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

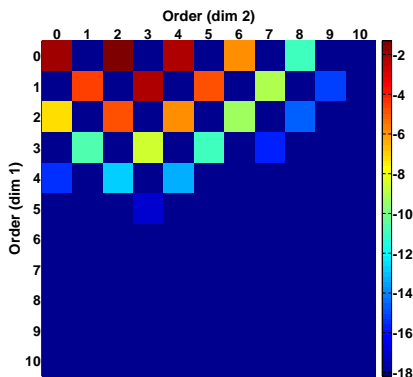
with

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

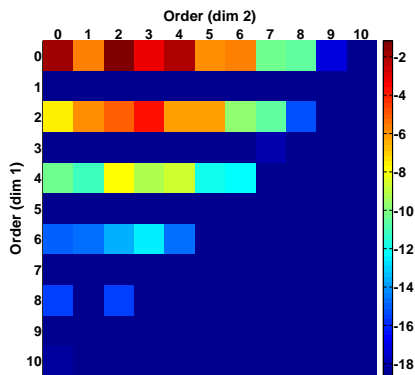
- KEY: Some  $\sigma_k^2 \rightarrow 0$ , hence the corresponding basis terms are dropped.

# 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  $\psi_i(x)\psi_j(y)$ .

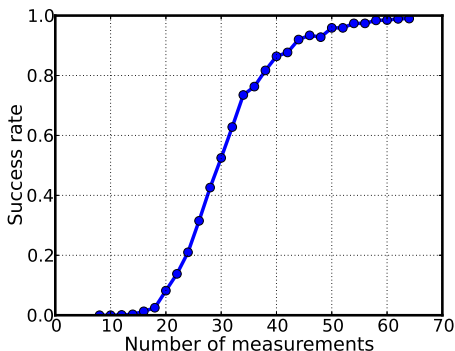
## Success rate grows with more data and 'sparser' model

Consider test function

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

where only  $S$  coefficients  $c_k$  are non-zero. Typical setting is

$$S < N < K$$



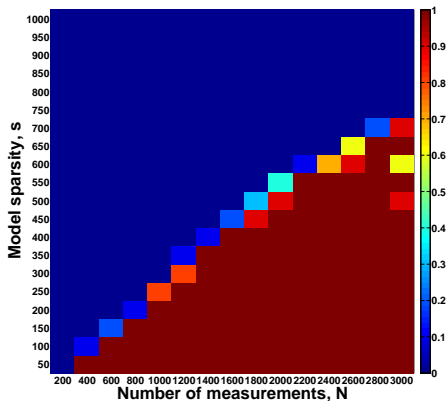
## Success rate grows with more data and 'sparser' model

Consider test function

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

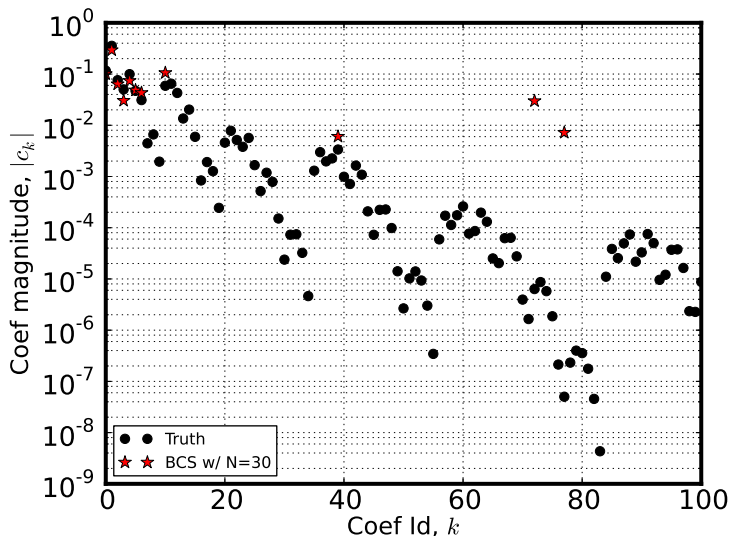
where only  $S$  coefficients  $c_k$  are non-zero. Typical setting is

$$S < N < K$$

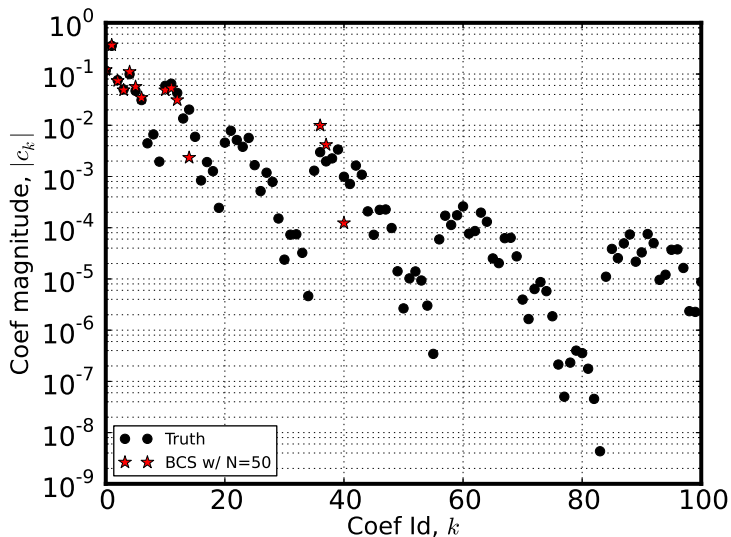




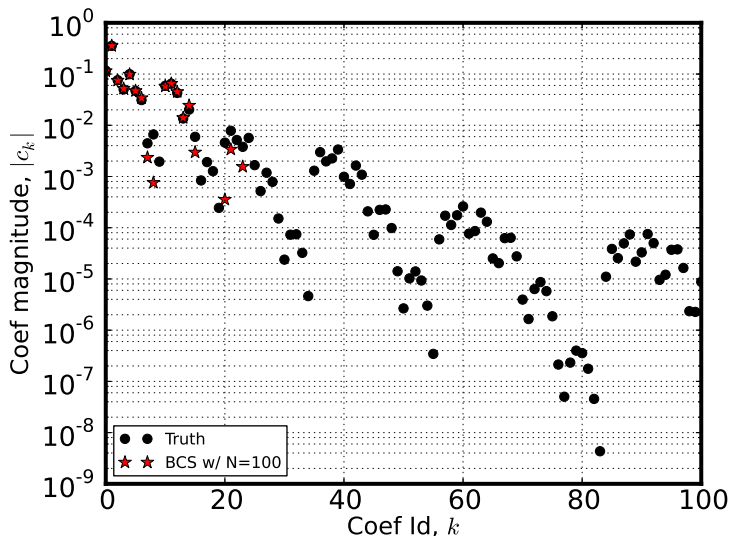
# BCS recovers true PC coefficients with increased number of measurements



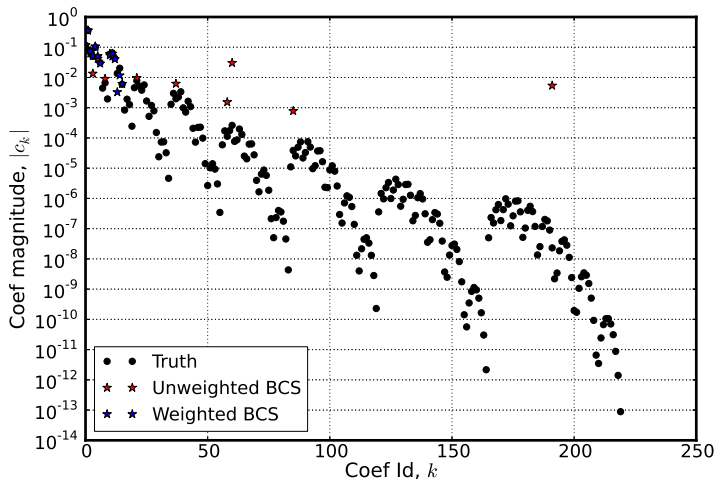
# BCS recovers true PC coefficients with increased number of measurements



# BCS recovers true PC coefficients with increased number of measurements



# WBCS recovers true coefficients better



$$f(\mathbf{x}) = x_0 \cos \left( e + \sum_{i=1}^9 x_i / i \right)$$

Sparsest solution:  $\min \|c\|_0$  such that  $u \approx Pc$

Compressive sensing:  $\min \|c\|_1$  such that  $u \approx Pc$

Weighted compressive sensing:  $\min \|Wc\|_1$  such that  $u \approx Pc$

Sparsest solution:  $\min \|c\|_0$  such that  $u \approx Pc$

Compressive sensing:  $\min \|c\|_1$  such that  $u \approx Pc$

Weighted compressive sensing:  $\min \|Wc\|_1$  such that  $u \approx Pc$

For sparse signals,  $u = Pc^s$ , with  $\|c^s\|_0 = S < K$ , ideal weights are

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

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

Sparsest solution:  $\min \|c\|_0$  such that  $u \approx Pc$

Compressive sensing:  $\min \|c\|_1$  such that  $u \approx Pc$

Weighted compressive sensing:  $\min \|Wc\|_1$  such that  $u \approx Pc$

For sparse signals,  $u = Pc^s$ , with  $\|c^s\|_0 = S < K$ , ideal weights are

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

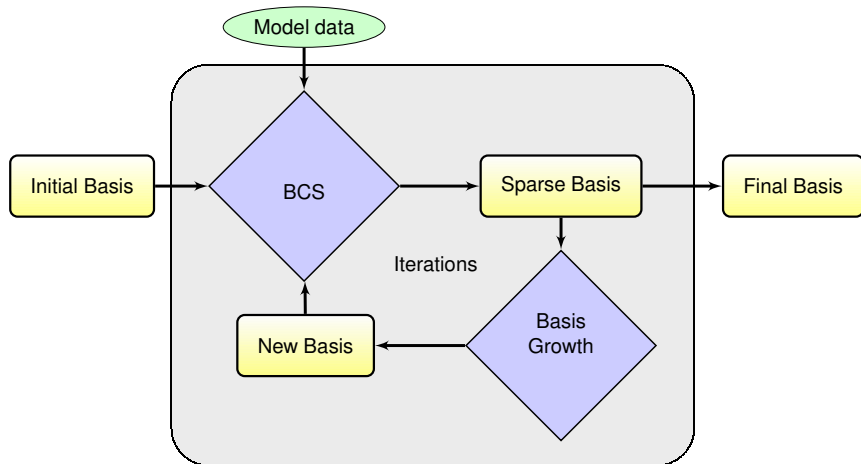
In practice, the true signal coefficients are not known, so...

Iterative re-weighting

$$W^{(i+1)} = \text{diag} \left( \frac{1}{|c_k^{(i)}| + \epsilon} \right) \quad [\epsilon \ll 1 \text{ for stability}]$$

# Iterative Bayesian Compressive Sensing (iBCS)

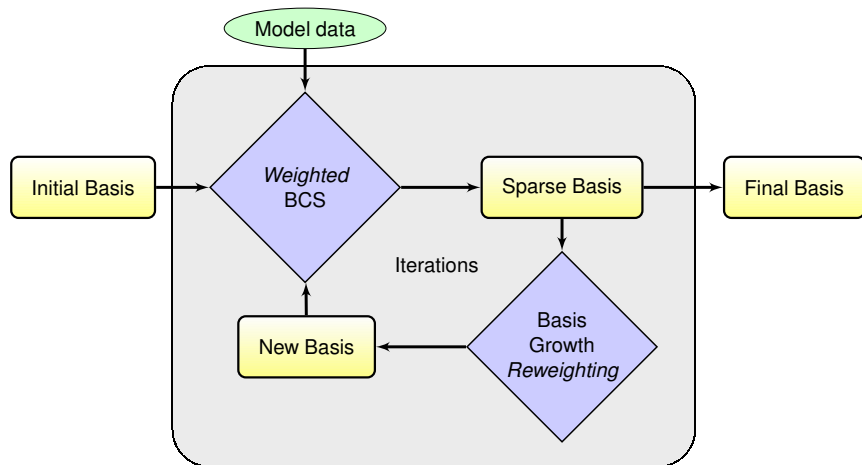
- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Sargsyan *et al.* 2014], [Jakeman *et al.* 2015].





# Iterative Bayesian Compressive Sensing (iBCS)

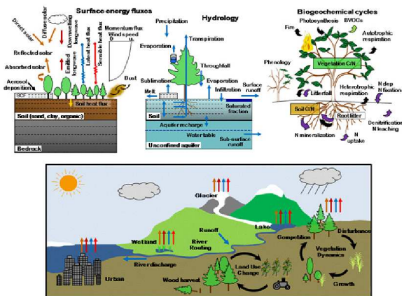
- Combine basis growth and reweighting!



# Basis set growth: simple anisotropic function

# Basis set growth: ... added outlier term

# Application of Interest: Community Land Model

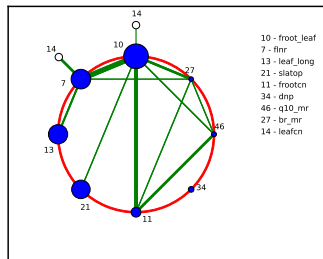
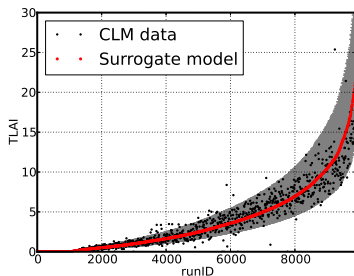


<http://www.cesm.ucar.edu/models/clm/>

- Nested computational grid hierarchy
- A single-site, 1000-yr simulation takes  $\sim 10$  hrs on 1 CPU
- Involves  $\sim 50$  input parameters; some dependent
- Non-smooth input-output relationship

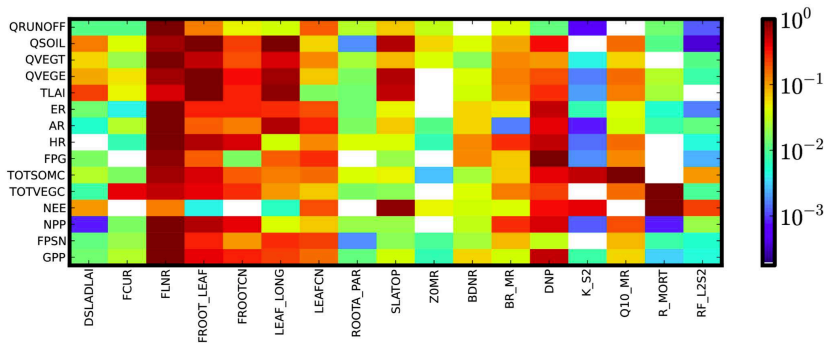
# Sparse PC surrogate for the Community Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 50-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



# Sparse PC Surrogate for the Community Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 50-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



# Summary

- **Surrogate** models are necessary for complex models
    - Replace the full model for both forward and inverse UQ
  - Uncertain inputs
    - **Polynomial Chaos** surrogates well-suited
  - Limited training dataset
    - **Bayesian** methods handle limited information well
  - Curse of dimensionality
    - The hope is that not too many dimensions matter
    - Compressive sensing (CS) ideas ported from machine learning
    - We implemented **iteratively reweighting Bayesian CS** algorithm that reduces dimensionality and increases order on-the-fly.
- 

- Open issues
  - Computational design. What is the best sampling strategy?
  - Overfitting still present. Cross-validation techniques help.

# Literature

- M. Tipping, “Sparse Bayesian learning and the relevance vector machine”, *J Machine Learning Research*, 1, pp. 211-244, 2001.
- S. Ji, Y. Xue and L. Carin, “Bayesian compressive sensing”, *IEEE Trans. Signal Proc.*, 56:6, 2008.
- S. Babacan, R. Molina and A. Katsaggelos, “Bayesian compressive sensing using Laplace priors”, *IEEE Trans. Image Proc.*, 19:1, 2010.
- E. J. Candes, M. Wakin and S. Boyd. “Enhancing sparsity by reweighted  $\ell_1$  minimization”, *J. Fourier Anal. Appl.*, 14 877-905, 2007.
- A. Saltelli, “Making best use of model evaluations to compute sensitivity indices”, *Comp Phys Comm*, 145, 2002.
- K. Sargsyan, C. Safta, H. Najm, B. Debusschere, D. Ricciuto and P. Thornton, “Dimensionality reduction for complex models via Bayesian compressive sensing”, *Int J for Uncertainty Quantification*, 4(1), pp. 63-93, 2014.
- J. Jakeman, M. Eldred and K. Sargsyan, “Enhancing  $\ell_1$ -minimization estimates of polynomial chaos expansions using basis selection”, *J Comp Phys*, in press, 2015, see ArXiv.



# Random variables represented by Polynomial Chaos

$$X \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta})$$

- $\boldsymbol{\eta} = (\eta_1, \dots, \eta_d)$  standard i.i.d. r.v.  
 $\Psi_k$  standard polynomials, orthogonal w.r.t.  $\pi(\boldsymbol{\eta})$ .

$$\Psi_k(\eta_1, \eta_2, \dots, \eta_d) = \psi_{k_1}(\eta_1) \psi_{k_2}(\eta_2) \cdots \psi_{k_d}(\eta_d)$$

- Typical truncation rule: total-order  $p$ ,  $k_1 + k_2 + \dots + k_d \leq p$ .  
Number of terms is  $K = \frac{(d+p)!}{d!p!}$ .
- Essentially, a parameterization of a r.v. by deterministic spectral modes  $c_k$ .
- Most common standard Polynomial-Variable pairs:  
(continuous) Gauss-Hermite, Legendre-Uniform,  
(discrete) Poisson-Charlier.

# Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x}) \equiv g\mathbf{c}(\mathbf{x}) \quad \underbrace{P(\mathbf{c}|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{P(\mathcal{D}|\mathbf{c})}_{\text{Likelihood}} \underbrace{P(\mathbf{c})}_{\text{Prior}}$$

- Data consists of *training runs*

$$\mathcal{D} \equiv \{(\mathbf{x}_i, u_i)\}_{i=1}^N$$

- Likelihood with a gaussian noise model with  $\sigma^2$  fixed or inferred,

$$L(\mathbf{c}) = P(\mathcal{D}|\mathbf{c}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^N \exp\left(-\frac{(u_i - g\mathbf{c}(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

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

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

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

$$\sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x}) = \boldsymbol{\Psi}(\mathbf{x})^T \mathbf{c} \in \mathcal{GP}(\boldsymbol{\Psi}(\mathbf{x})^T \boldsymbol{\mu}, \boldsymbol{\Psi}(\mathbf{x}) \boldsymbol{\Sigma} \boldsymbol{\Psi}(\mathbf{x}')^T)$$

## Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

$\mathbb{I}_i$  is the set of bases with only  $x_i$  involved

# Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

- Joint sensitivity indices

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

$\mathbb{I}_{ij}$  is the set of bases with only  $x_i$  and  $x_j$  involved

# Sensitivity information comes free with PC surrogate, but not with piecewise PC

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

- Main effect sensitivity indices

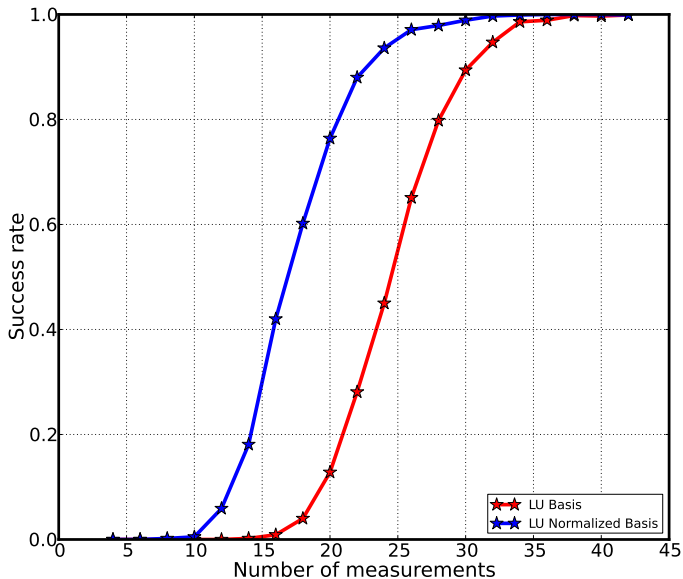
$$S_i = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i))] }{\text{Var}[g(\mathbf{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

- Joint sensitivity indices

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

- For piecewise PC, need to resort to Monte-Carlo estimation  
[\[Saltelli, 2002\]](#).

# Basis normalization helps the success rate



## Input correlations: Rosenblatt transformation

- Rosenblatt transformation maps any (not necessarily independent) set of random variables  $\lambda = (\lambda_1, \dots, \lambda_d)$  to uniform i.i.d.'s  $\{x_i\}_{i=1}^d$  [Rosenblatt, 1952].

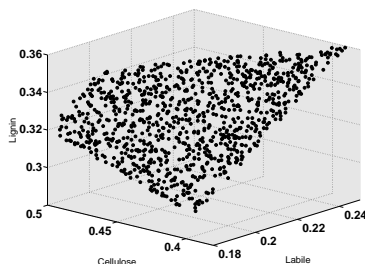
$$x_1 = F_1(\lambda_1)$$

$$x_2 = F_{2|1}(\lambda_2|\lambda_1)$$

$$x_3 = F_{3|2,1}(\lambda_3|\lambda_2, \lambda_1)$$

$\vdots$

$$x_d = F_{d|d-1,\dots,1}(\lambda_d|\lambda_{d-1}, \dots, \lambda_1)$$



- Inverse Rosenblatt transformation  $\lambda = R^{-1}(\mathbf{x})$  ensures a well-defined input PC construction

$$\lambda_i = \sum_{k=0}^{K-1} \lambda_{ik} \Psi_k(\mathbf{x})$$

- Caveat: the conditional distributions are often hard to evaluate accurately.

# Strong discontinuities/nonlinearities challenge global polynomial expansions

- Basis enrichment [Ghosh & Ghanem, 2005]
- Stochastic domain decomposition
  - Wiener-Haar expansions,  
Multiblock expansions,  
Multiwavelets, [Le Maître *et al*, 2004,2007]
  - also known as Multielement PC [Wan & Karniadakis, 2009]
- Smart splitting, discontinuity detection  
[Archibald *et al*, 2009; Chantrasmi, 2011; Sargsyan *et al*, 2011; Jakeman *et al*, 2012]
- Data domain decomposition,
  - Mixture PC expansions [Sargsyan *et al*, 2010]
- Data clustering, classification,
  - Piecewise PC expansions



# Piecewise PC expansion with classification

- Cluster the training dataset into non-overlapping subsets  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , where the behavior of function is smoother
- Construct global PC expansions  $g_i(\mathbf{x}) = \sum_k c_{ik} \Psi_k(\mathbf{x})$  using each dataset individually ( $i = 1, 2$ )
- Declare a surrogate

$$g_s(\mathbf{x}) = \begin{cases} g_1(\mathbf{x}) & \text{if } \mathbf{x} \in^* \mathcal{D}_1 \\ g_2(\mathbf{x}) & \text{if } \mathbf{x} \in^* \mathcal{D}_2 \end{cases}$$

\* Requires a classification step to find out which cluster  $\mathbf{x}$  belongs to. We applied Random Decision Forests (RDF).

- Caveat: the sensitivity information is harder to obtain.