Adaptive Basis Selection and Dimensionality Reduction with Bayesian Compressive Sensing

Khachik Sargsyan

Sandia National Laboratories, Livermore, CA
Transportation Energy Center
Reacting Flow Research Department

SIAM UQ Conference,
Raleigh, NC, April 3, 2012

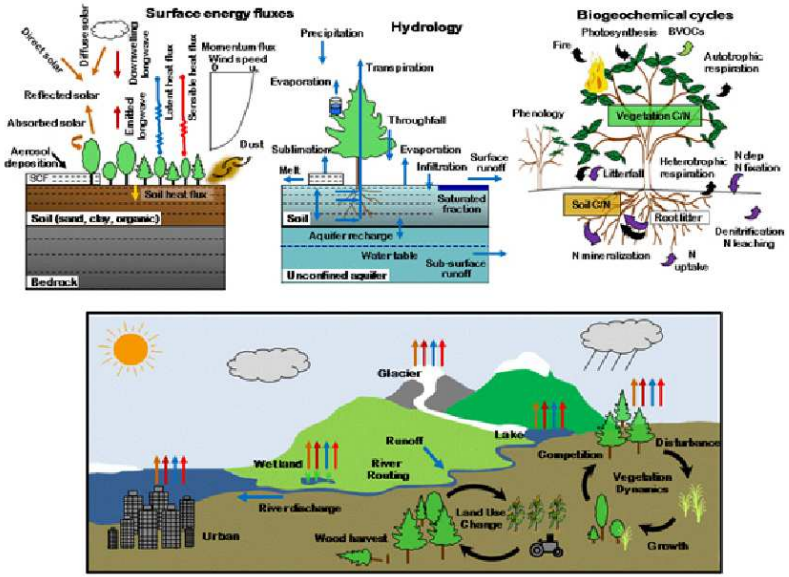
Acknowledgements

Cosmin Safta, SNL
Bert Debusschere, SNL
Habib Najm, SNL
Robert Berry, formerly SNL
Daniel Ricciuto, ORNL
Peter Thornton, ORNL

- DOE, Biological and Environmental Research,
- DOE, Advanced Scientific Computing Research.

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.

Application of Interest: Community Land Model



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

- Nested computational grid hierarchy
- A single-site, 1000-yr simulation takes ~ 10 hrs on 1 CPU
- Involves ~ 80 input parameters; some correlated
- Strongly nonlinear input-output relationship

[MS 59, Climate UQ, Wed 5-6pm, D. Ricciuto, C. Safta]

Challenges we tackle

Construct surrogate for a complex model

- 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
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Strongly non-linear forward function

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter inference

Random variables represented by Polynomial Chaos

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

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

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

- Typical truncation rule: total-order p, $k_1 + k_2 + \ldots k_d \leq p$.
  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.

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

Random variables represented by Polynomial Chaos

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

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

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

- Typical truncation rule: total-order p, $k_1 + k_2 + \ldots k_d \leq p$.
  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.

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

Random variables represented by Polynomial Chaos

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

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

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

- Typical truncation rule: total-order p, $k_1 + k_2 + \ldots k_d \leq p$.
  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.

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

Random variables represented by Polynomial Chaos

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

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

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

- Typical truncation rule: total-order p, $k_1 + k_2 + \ldots k_d \leq p$.
  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.

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

PC surrogate construction

- Build/presume PC for input parameter λ

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

PC surrogate construction

- Build/presume PC for input parameter λ

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

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

$$\lambda = \lambda_0 + \lambda_1 \eta$$

PC surrogate construction

- Build/presume PC for input parameter λ

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

- If input parameters are uniform $\lambda_i \sim \text{Uniform}[a_i, b_i]$, then

$$\lambda_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2}\,\eta_i.$$

PC surrogate construction

- Build/presume PC for input parameter λ

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

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

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

PC surrogate construction

- Build/presume PC for input parameter λ

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

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

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

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

$$u = f(\boldsymbol{\lambda}(\boldsymbol{\eta})) \qquad u = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g(\boldsymbol{\eta})$$

- For optimization, inverse problems, the surrogate $g(\boldsymbol{\eta})$ can replace the expensive forward function $f(\boldsymbol{\lambda}(\boldsymbol{\eta}))$
- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.

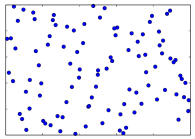
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

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



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

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

samples at quadrature

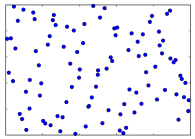
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

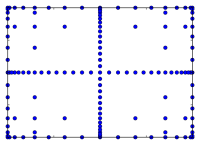
$$\frac{1}{N}\sum_{j=1}^{N} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)$$



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

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



samples at quadrature

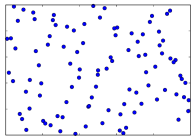
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

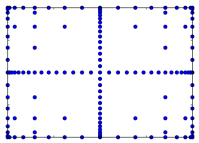
$$\frac{1}{N}\sum_{j=1}^{N} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)$$



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

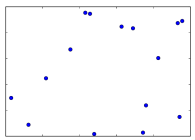
$$\sum_{j=1}^{Q} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)w_j$$



samples at quadrature

- *Bayesian inference*

$$P(c_k|u(\boldsymbol{\eta}_j)) \propto P(u(\boldsymbol{\eta}_j)|c_k)P(c_k)$$



any (number of) samples

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g_{\boldsymbol{c}}(\boldsymbol{\eta})$$

$$\overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g_{\boldsymbol{c}}(\boldsymbol{\eta}) \qquad \overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

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

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

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g_{\boldsymbol{c}}(\boldsymbol{\eta}) \qquad \overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

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

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

- Prior on $\boldsymbol{c}$ is chosen to be conjugate, uniform or gaussian.

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g_{\boldsymbol{c}}(\boldsymbol{\eta}) \qquad \overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

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

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

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

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

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g_{\boldsymbol{c}}(\boldsymbol{\eta}) \qquad \overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

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

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

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

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

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

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

In a different language....

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

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

- 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.
- Tikhonov regularization $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_2\}$
- Lasso regression $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$ subject to $||\boldsymbol{c}||_1 \leq \alpha$
- Compressive sensing $\quad$ with PC [Doostan and Owhadi, 2011]

In a different language....

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

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

- 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.
- Tikhonov regularization $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_2\}$
- Lasso regression $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$ subject to $||\boldsymbol{c}||_1 \leq \alpha$
- Compressive sensing $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

In a different language....

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

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

- 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.
- Tikhonov regularization $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_2\}$
- Lasso regression $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$ subject to $||\boldsymbol{c}||_1 \leq \alpha$
- Compressive sensing $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

Bayesian: Likelihood, Prior

Bayesian Compressive Sensing (BCS)

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- 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

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

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

Bayesian Compressive Sensing (BCS)

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- 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

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

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

Bayesian Compressive Sensing (BCS)

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- 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

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

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

Bayesian Compressive Sensing (BCS)

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- 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

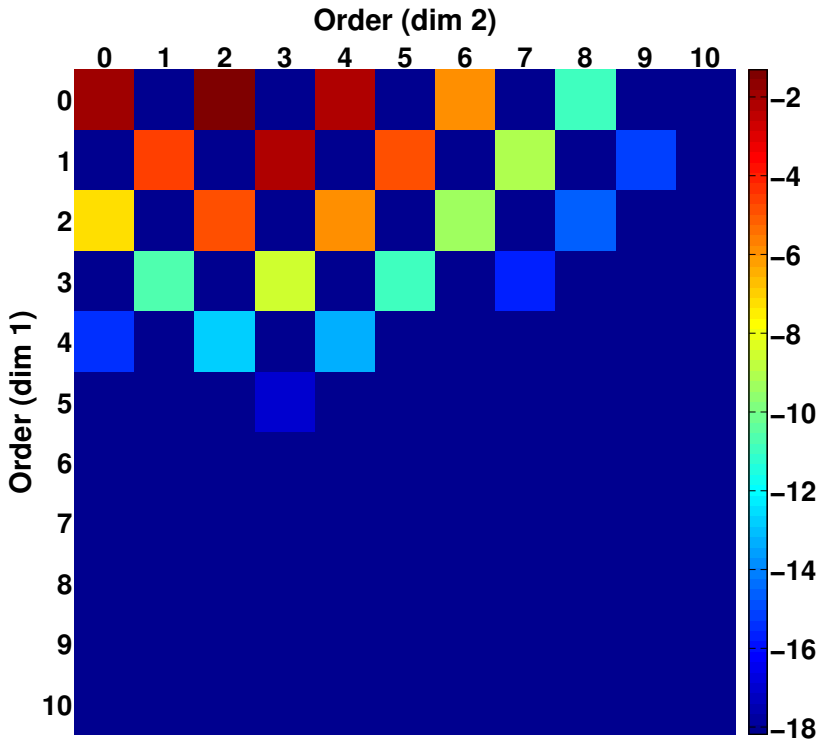
$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

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

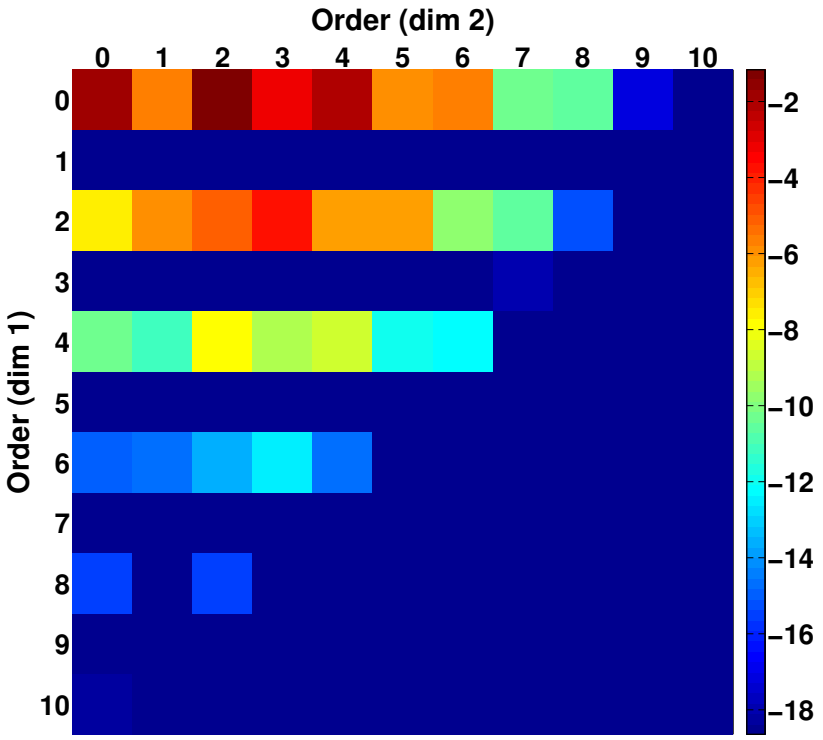
[Ji *et al.*, 2008; Babacan *et al.*, 2010]

BCS removes unnecessary basis terms

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



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



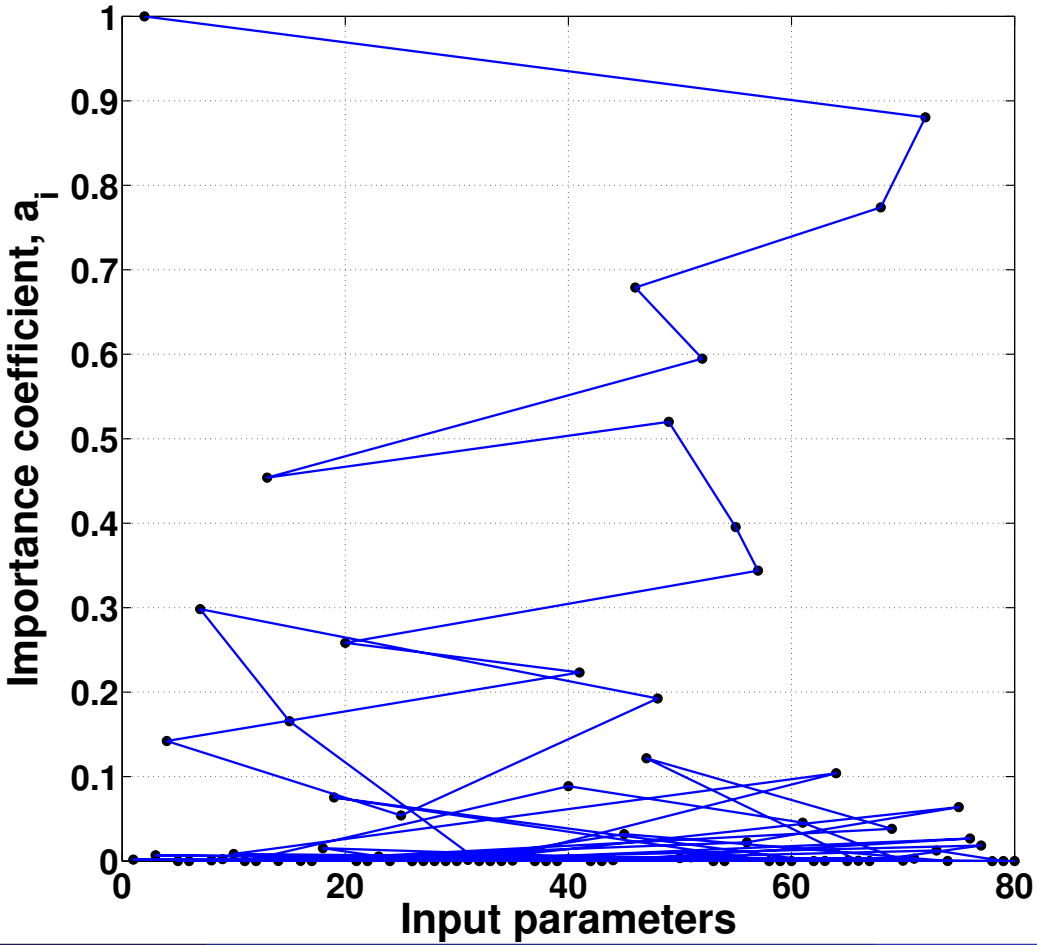
The square (i,j) represents the (log) spectral coefficient for the basis term $\psi_i(x)\psi_j(y)$.

BCS picks the most important dimensions

Consider test function

$$f(\boldsymbol{x}) = \exp\left(\sum_{i=1}^{d} a_i x_i\right)$$

Dimensional importance coefficients set to $a_i = (i/d)^{10}$ and shuffle.



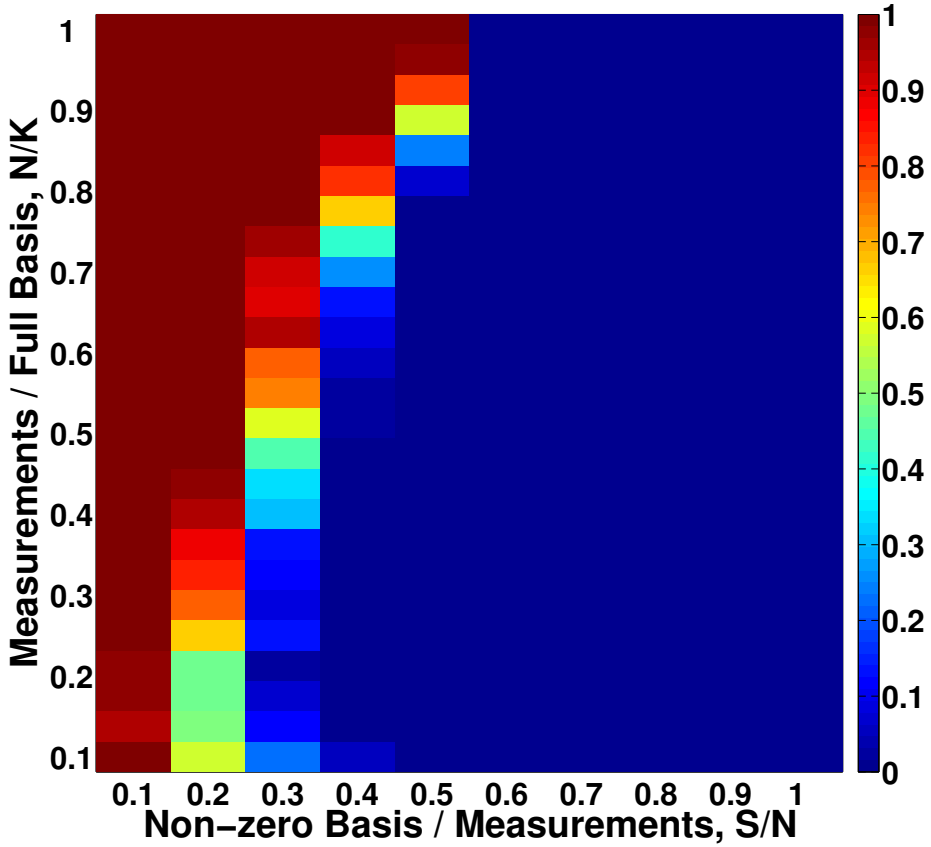
Success rate grows with more data and 'sparser' model

Consider test function

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

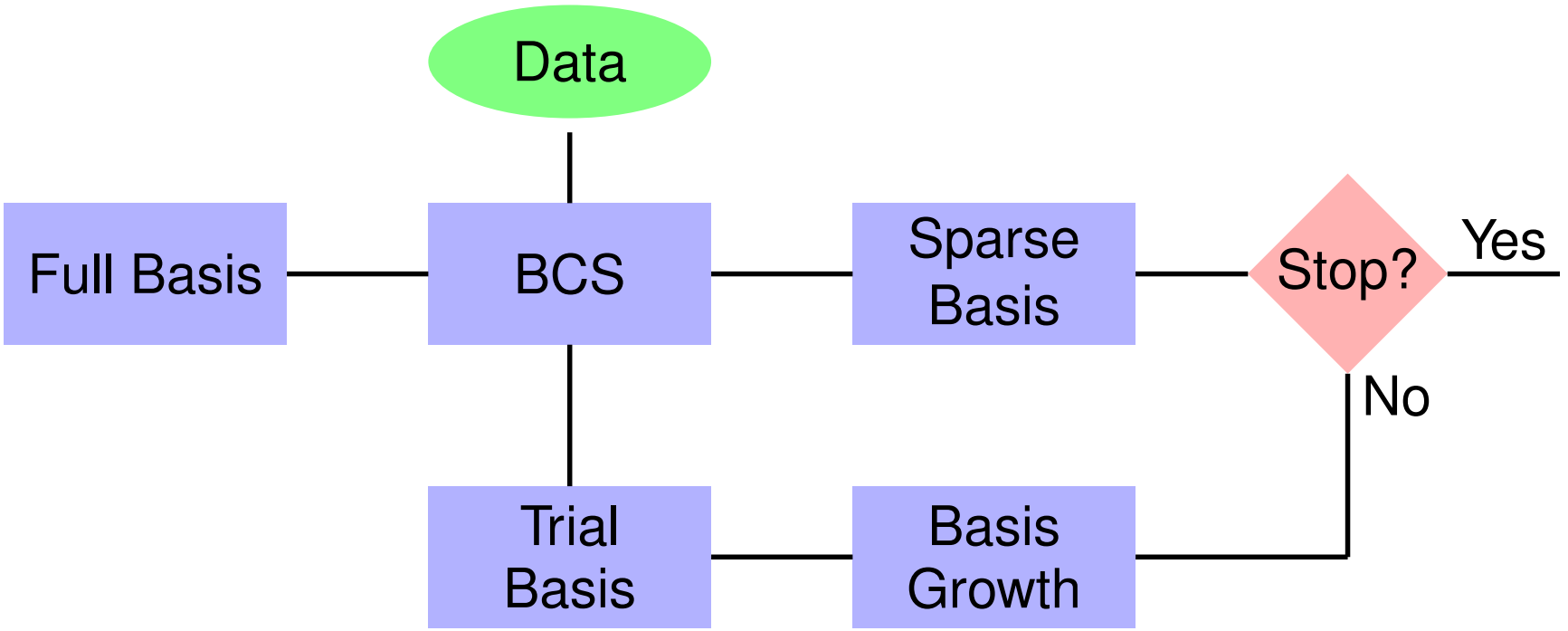
where only S coefficients c_k are non-zero. Typical setting is

$$S < N < K$$



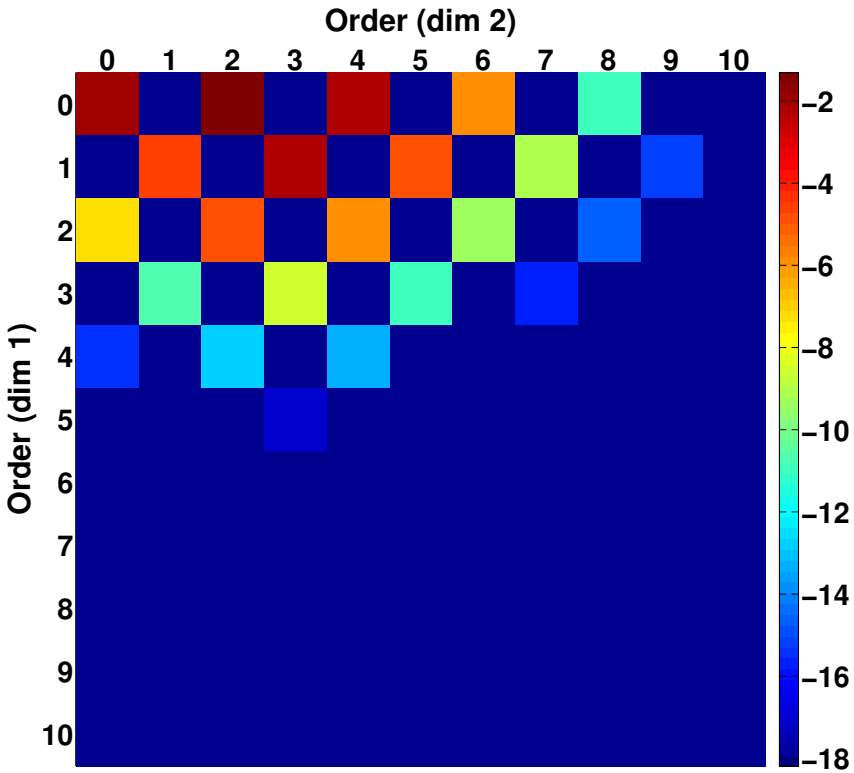
Iterative Bayesian Compressive Sensing (iBCS)

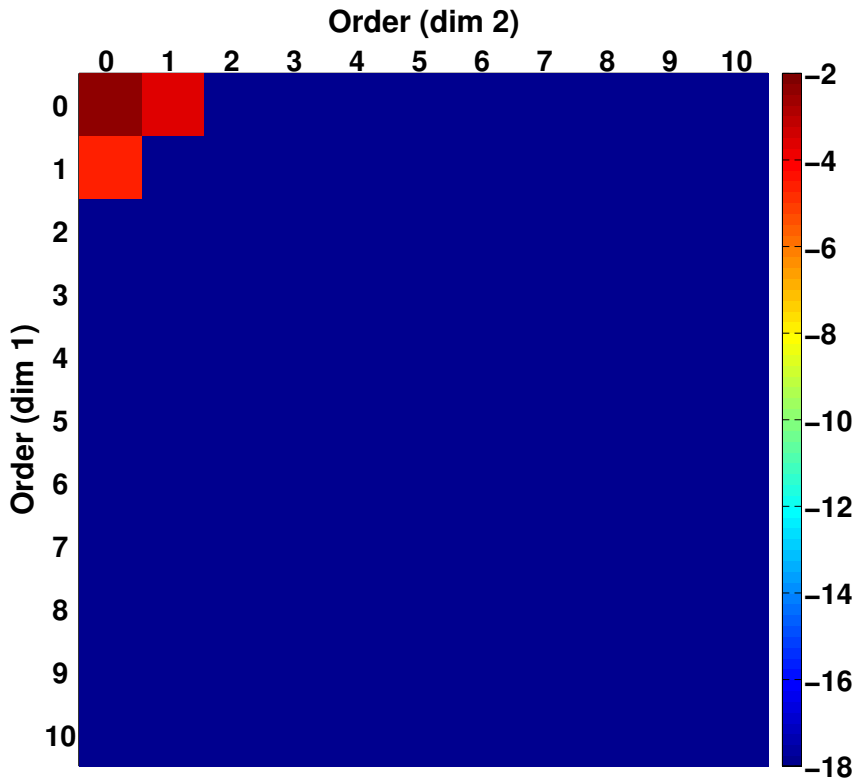
- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [S. *et al.* 2012].



Basis set growth

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



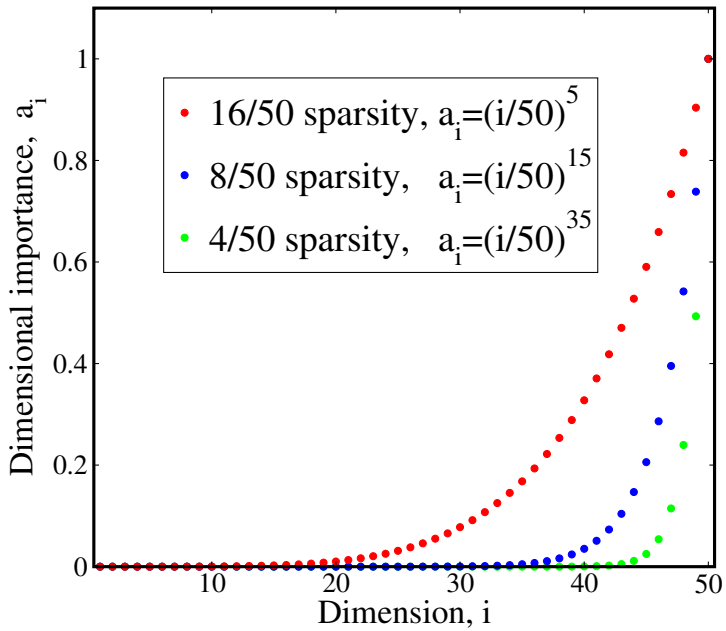


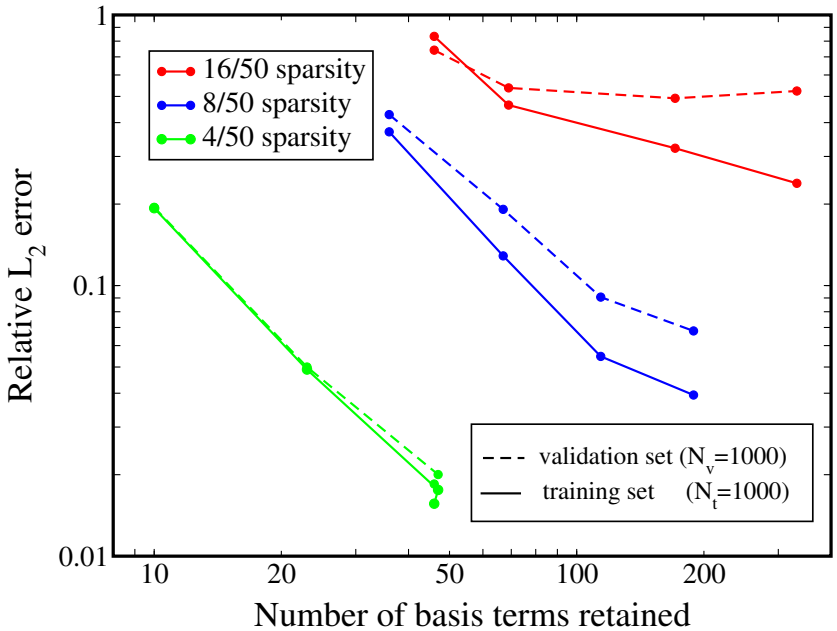
The fewer dimensions matter, the better

$$f(\boldsymbol{x}) = \exp\left(\sum_{i=1}^{d} a_i x_i\right)$$

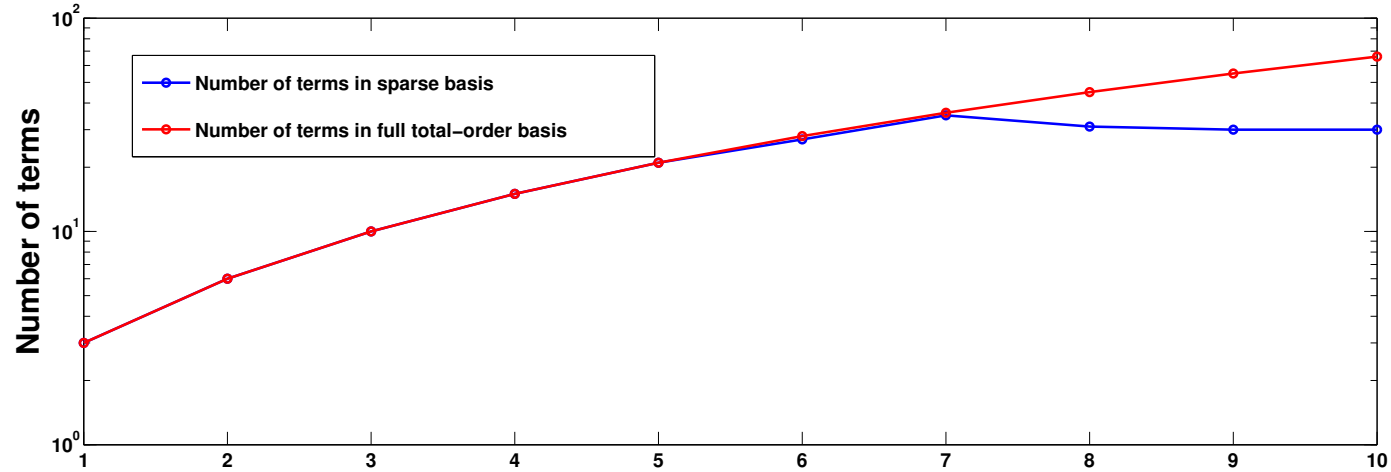
Dimensionality importance coefficients are chosen so that 90% of energy is in a small subset of dimensions

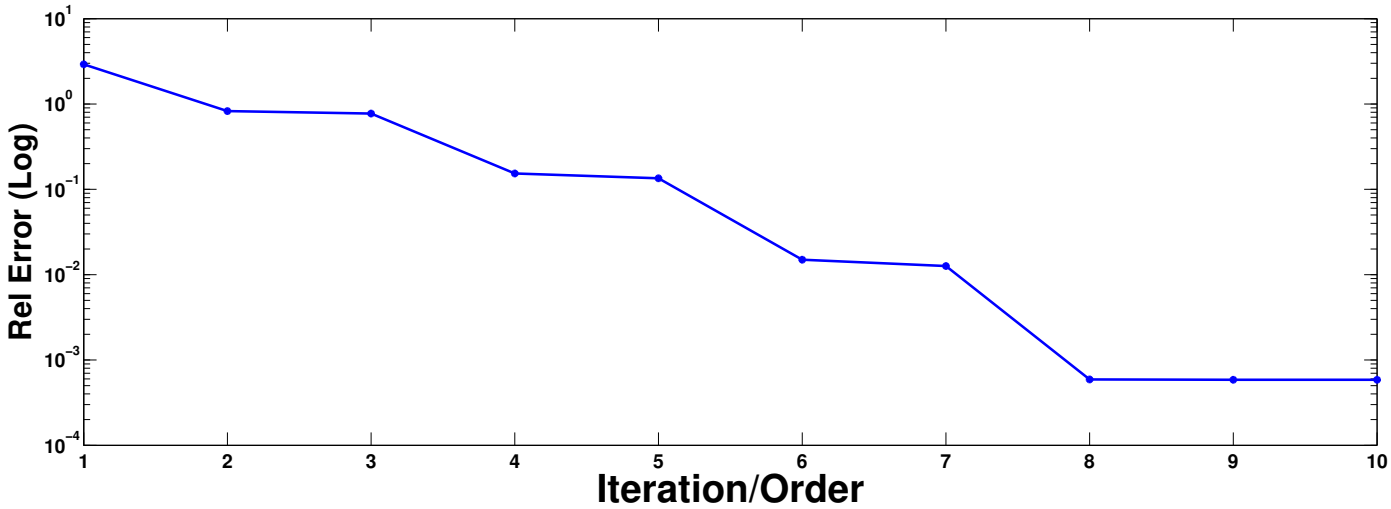
Validation error increase indicates overfitting. $N_t = 1000$ training runs are sufficient if ~ 10 dimensions matter.



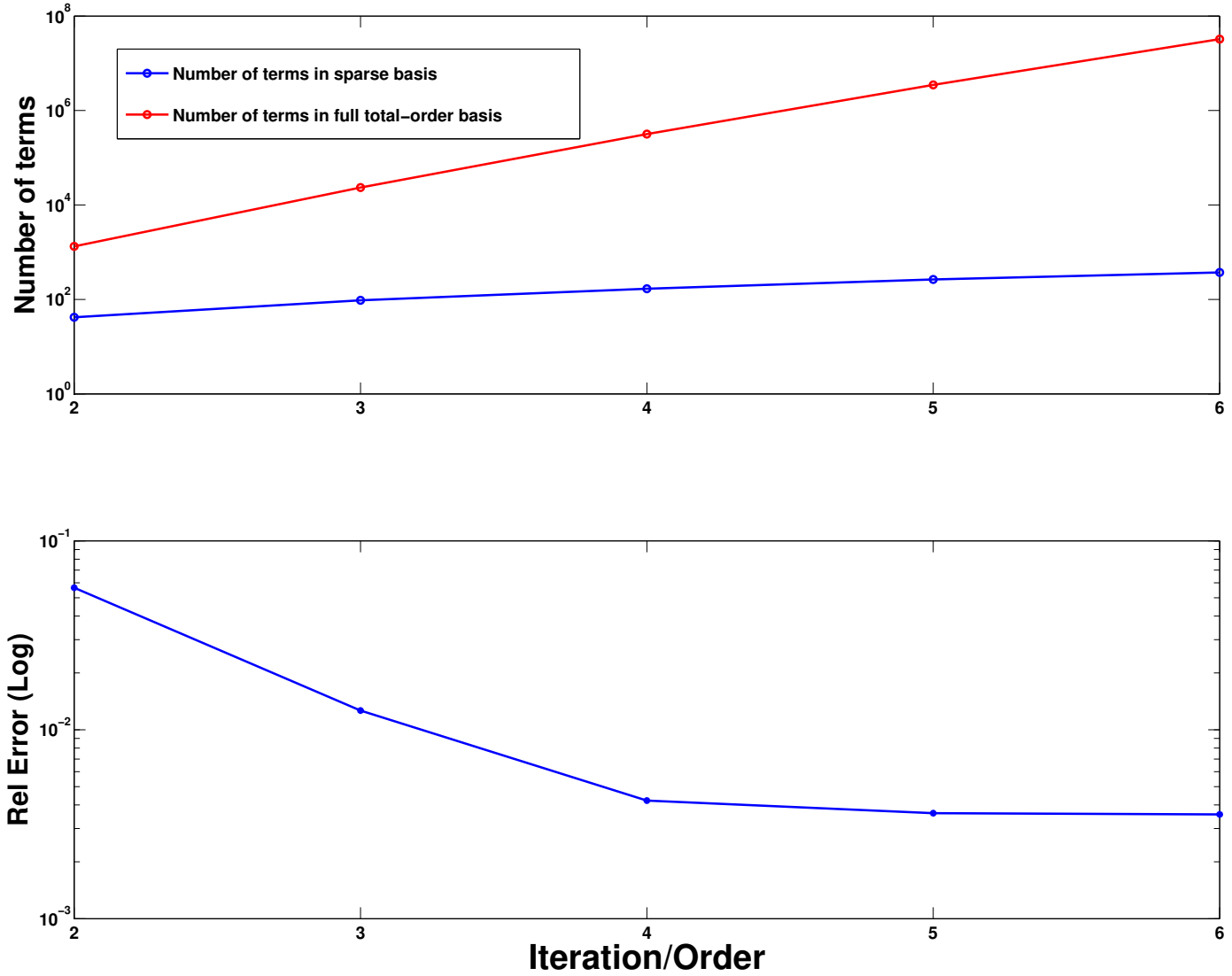


iBCS leads to reasonable accuracy with significant dimensionality reduction





iBCS leads to reasonable accuracy with significant dimensionality reduction



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; S. *et al*, 2011]
- Data domain decomposition,
  - Mixture PC expansions [S. *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 $g_i(\boldsymbol{x}) = \sum_k c_{ik} \Psi_k(\boldsymbol{x})$ using each dataset individually ($i = 1, 2$)
- Declare a surrogate

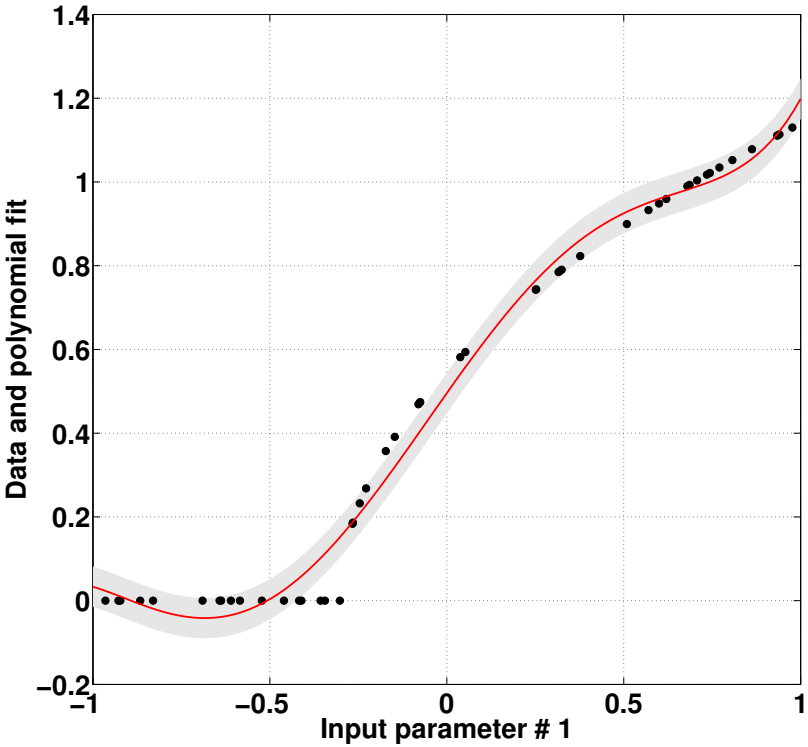
$$g_s(\boldsymbol{x}) = \begin{cases} g_1(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_1 \\ g_2(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_2 \end{cases}$$

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

- Caveat: the sensitivity information is harder to obtain.

Illustration of piecewise PC construction

Global 5-th order surrogate fails



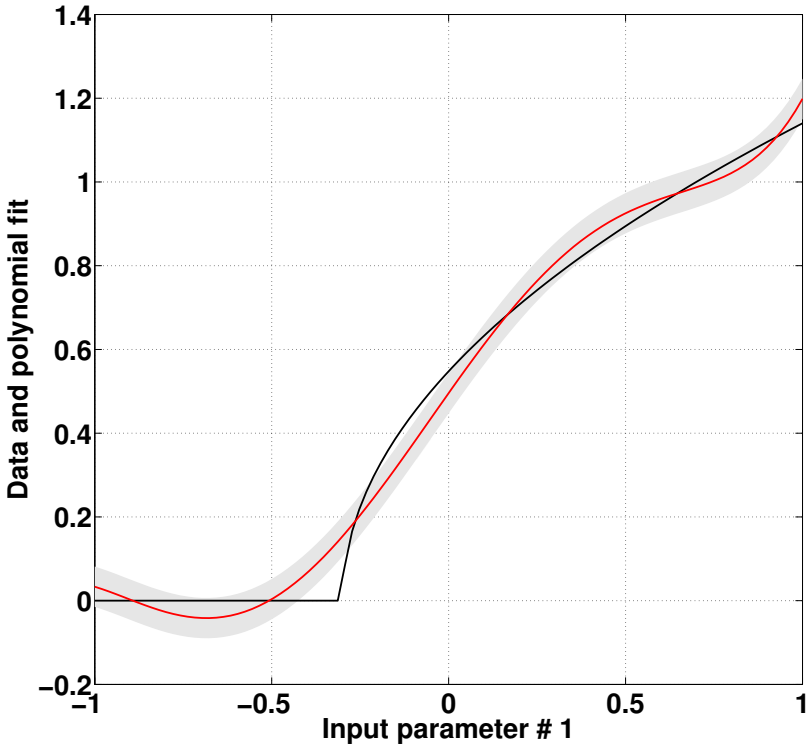
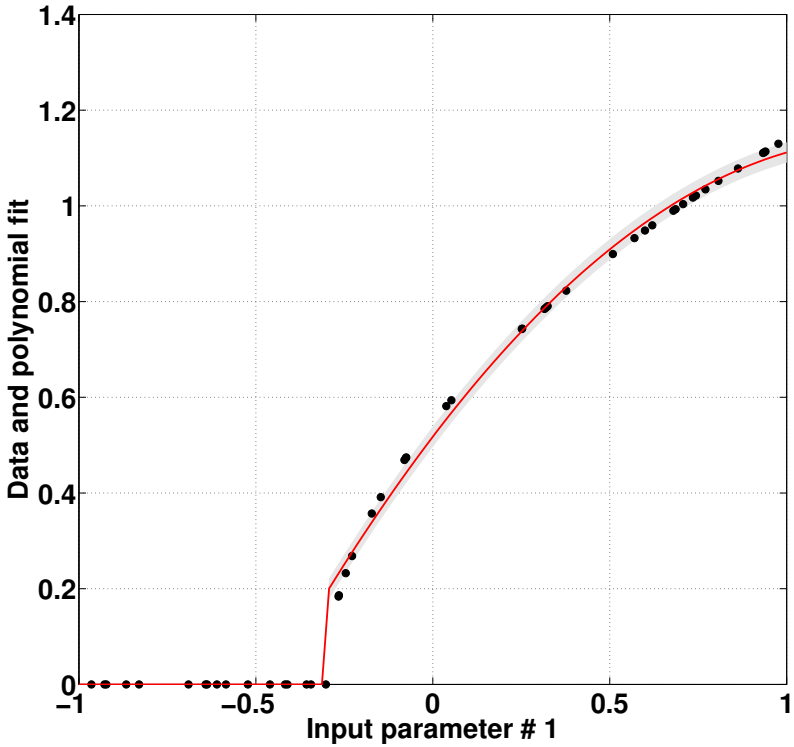


Illustration of piecewise PC construction

Piecewise 2-nd order surrogate



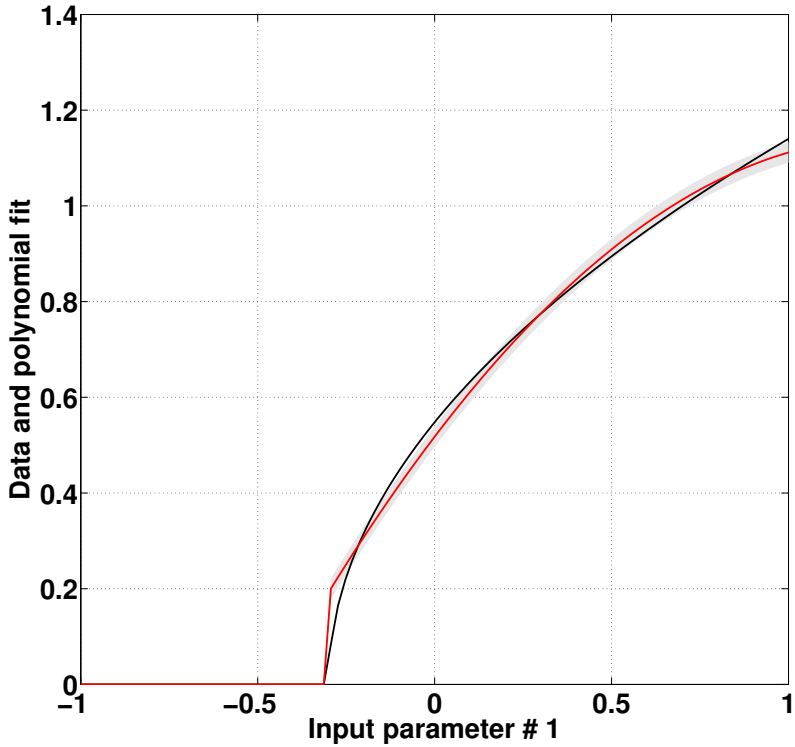
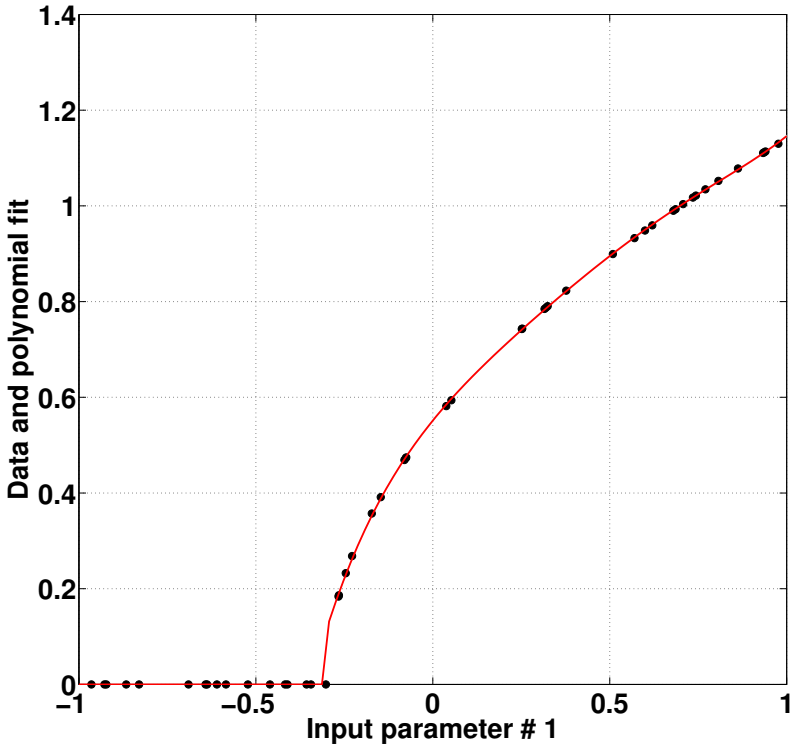


Illustration of piecewise PC construction

Piecewise 5-th order surrogate



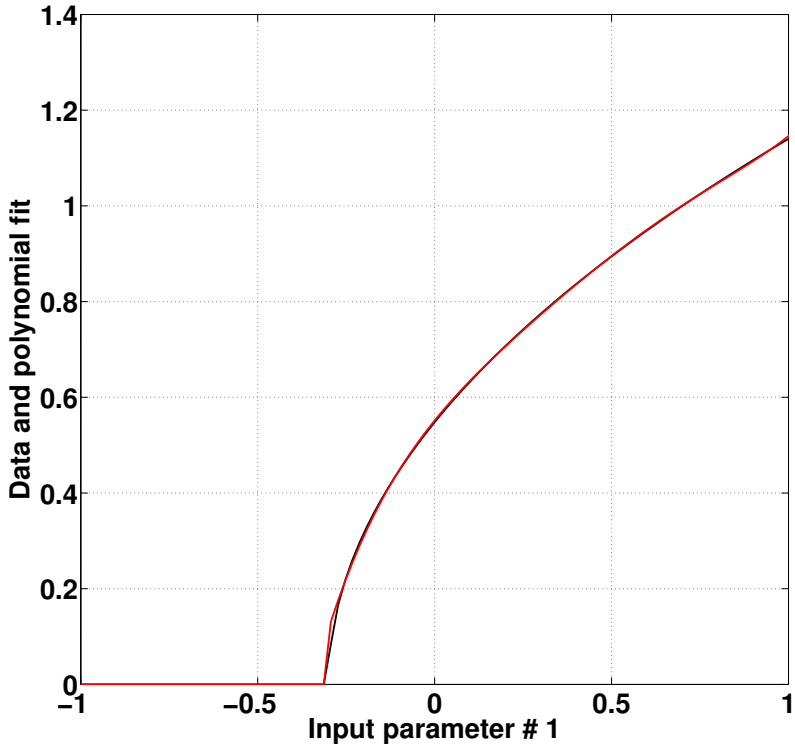
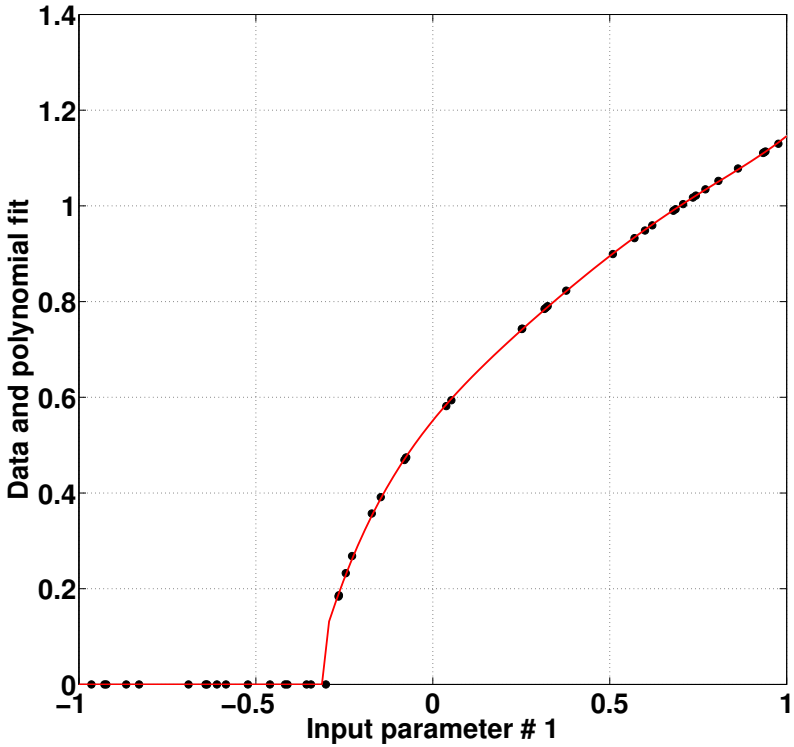


Illustration of piecewise PC construction

Piecewise 5-th order surrogate



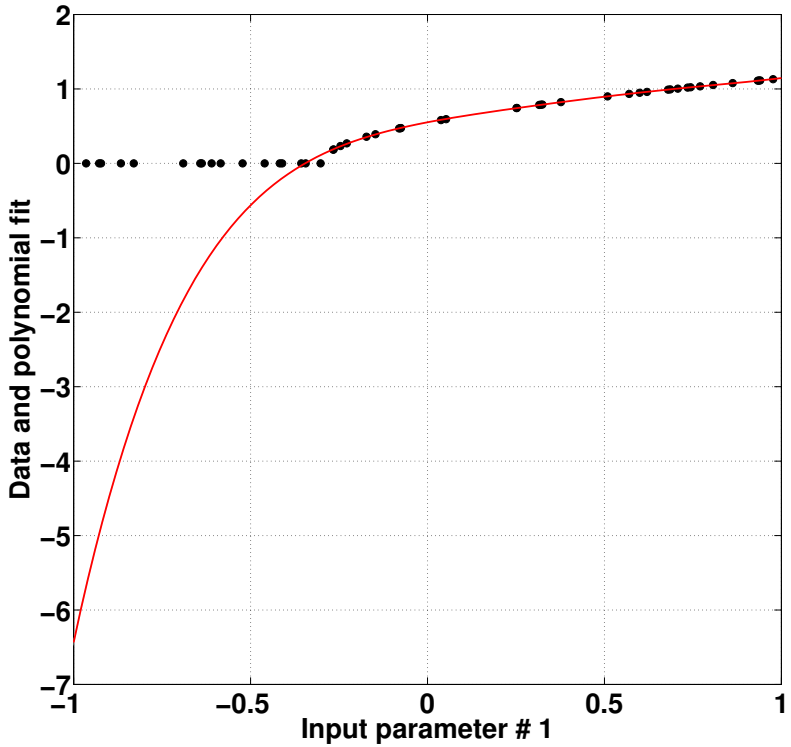
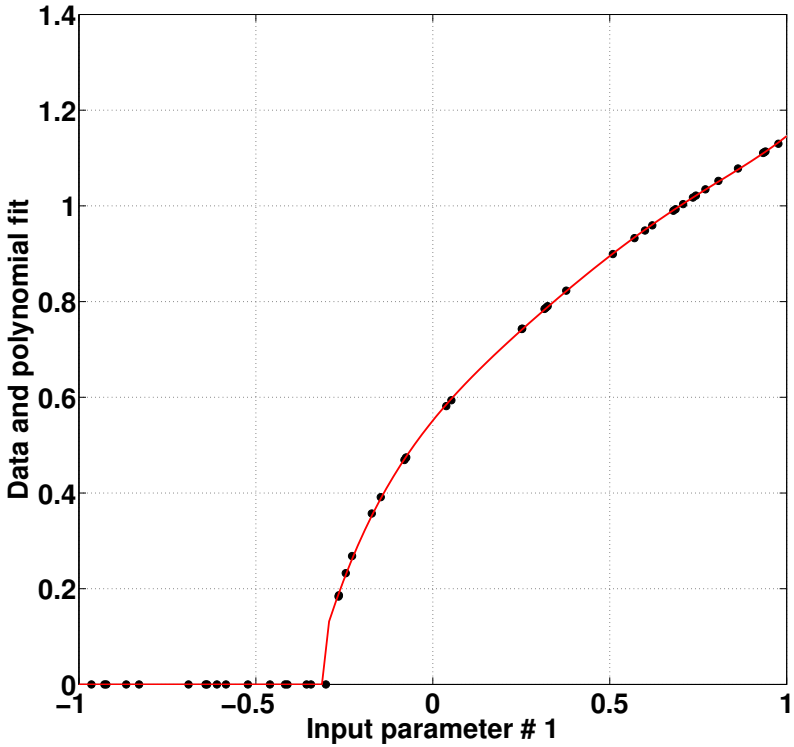
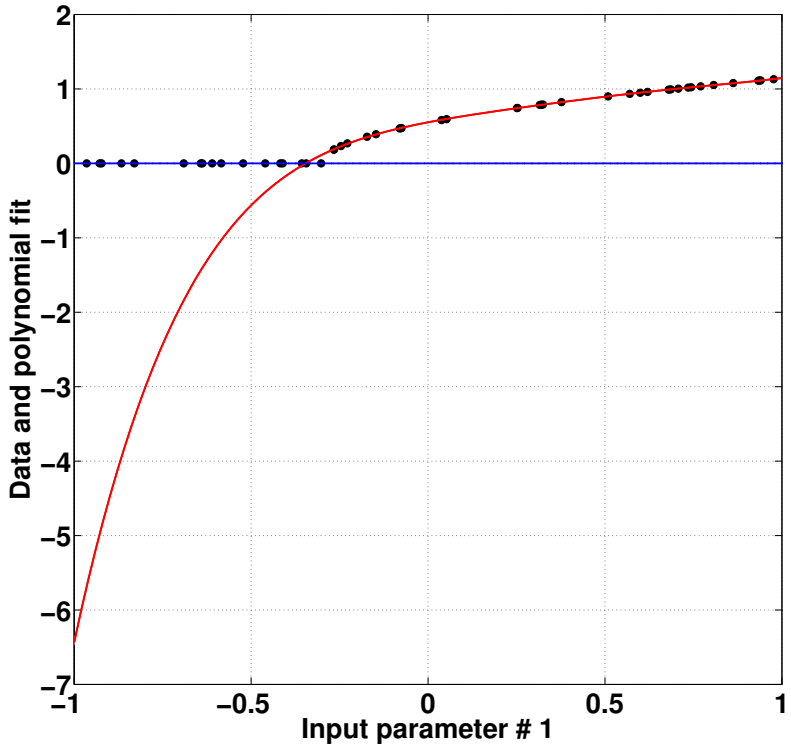


Illustration of piecewise PC construction

Piecewise 5-th order surrogate





Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

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

Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

- Joint sensitivity indices

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

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

Sensitivity information comes free with PC surrogate,

but not with piecewise PC

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

- Main effect sensitivity indices

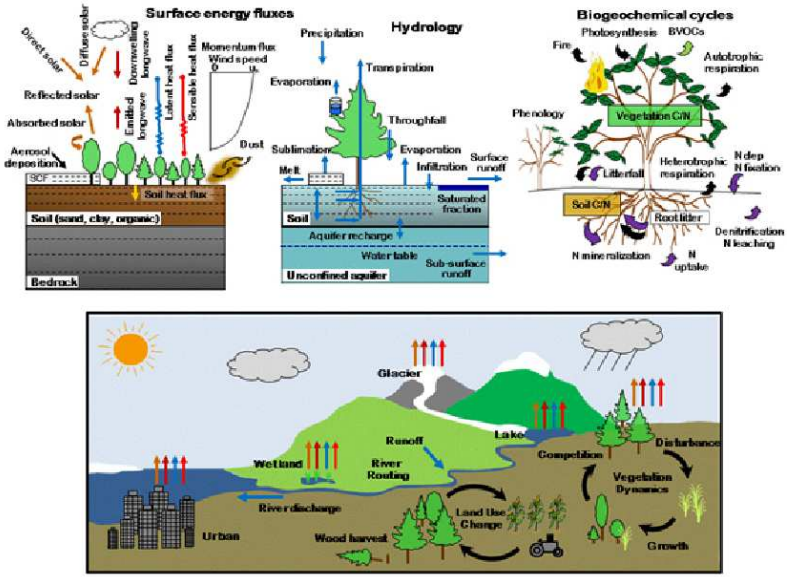
$$S_i = \frac{Var[\mathbb{E}(g(\boldsymbol{x}|x_i)]}{Var[g(\boldsymbol{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 ||\Psi_k||^2}{\sum_{k>0} c_k^2 ||\Psi_k||^2}$$

- Joint sensitivity indices

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

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

Application of Interest: Community Land Model



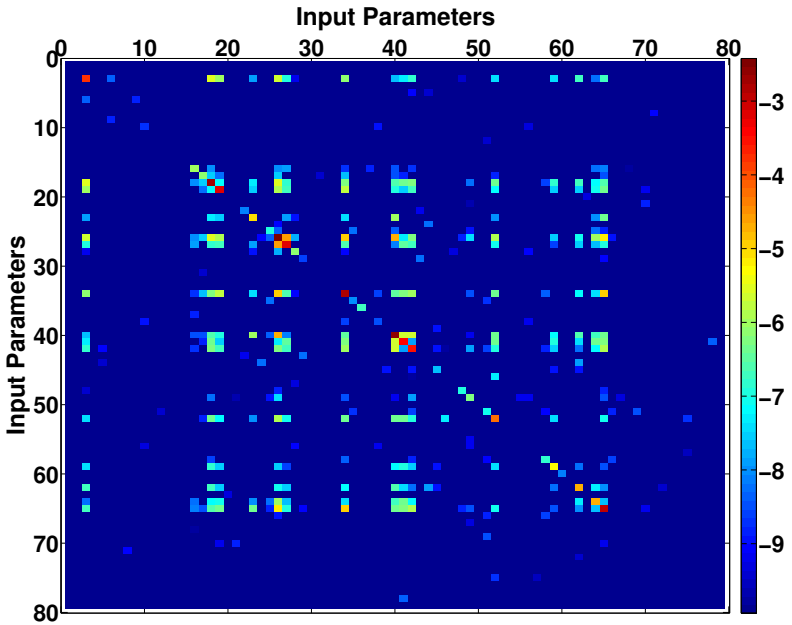
http://www.cesm.ucar.edu/models/clm/

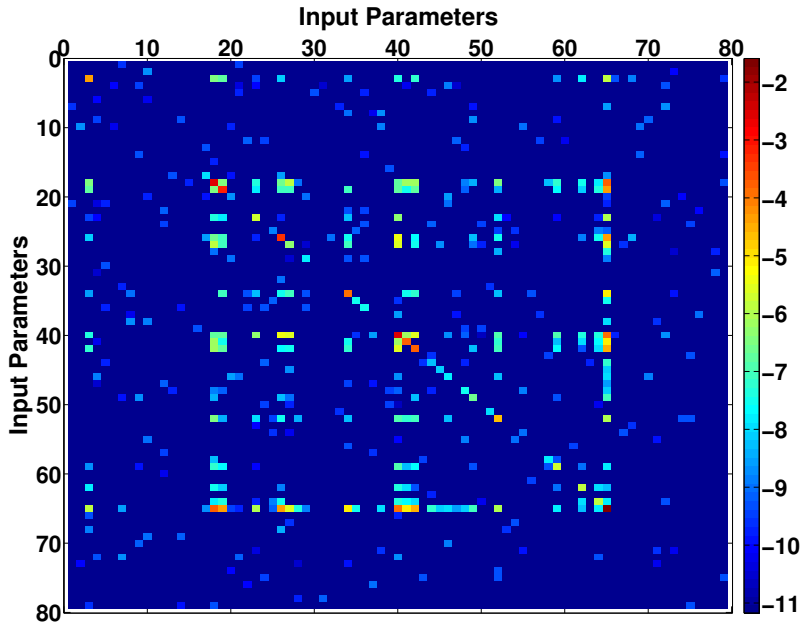
- Nested computational grid hierarchy
- A single-site, 1000-yr simulation takes ~ 10 hrs on 1 CPU
- Involves ~ 80 input parameters; some correlated
- Strongly nonlinear input-output relationship

[MS 59, Climate UQ, Wed 5-6pm, D. Ricciuto, C. Safta]

Sparse PC surrogate for Community Land Model: main effect and joint sensitivity indices

- First order information : rank input parameters
- Second order information : most influential input couplings
- About 200 out of 3200 terms retained
- Sparse PC can be used for parameter calibration against experimental data





[MS 59, Climate UQ, Wed 5-6pm, D. Ricciuto, C. Safta]

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 signal processing community
  - We implemented *iterative* Bayesian CS algorithm that reduces dimensionality and increases order on-the-fly.
- Nonlinear behavior
  - Data clustering and classification-driven piecewise PC

Literature

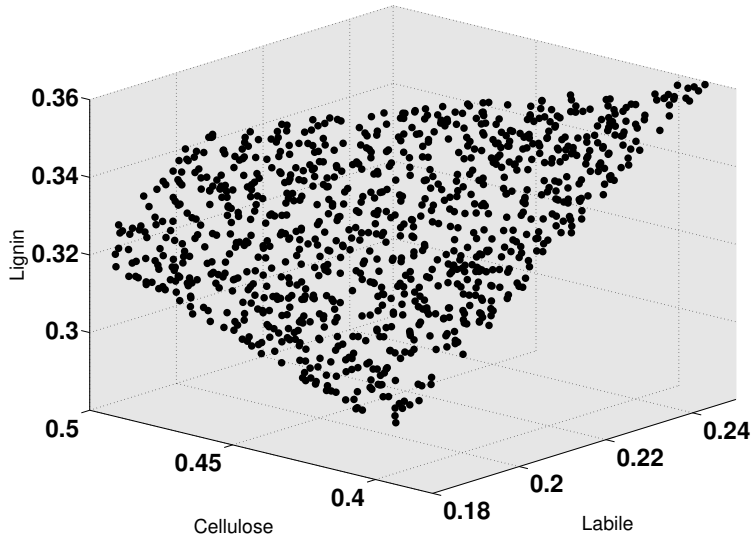
- S. Ji, Y. Xue and L. Carin, “Bayesian compressive sensing”, *IEEE Trans. Signal Proc.*, 56:6, 2008.
- S. Babacan, R. Molina and A. Katsaggelos, “Bayesian compressive sensing using Laplace priors”, *IEEE Trans. Image Proc.*, 19:1, 2010.
- A. Saltelli, “Making best use of model evaluations to compute sensitivity indices”,*Comp Phys Comm*, 145,2002.

- K. Sargsyan, C. Safta, B. Debusschere and H. Najm, “Multiparameter spectral representation of competence dynamics in Bacillus Subtilis”. *Submitted to IEEE Trans. Comp. Biol. and Bioinformatics*, 2012.
- K. Sargsyan, C. Safta, R. Berry, J. Ray, B. Debusschere and H. Najm, “Efficient uncertainty quantification methodologies for high-dimensional climate land models”, Sandia Report, SAND2011-8757, Nov. 2011.
- K. Sargsyan, B. Debusschere, H. Najm and O. Le Maître, “Spectral representation and reduced order modeling of the dynamics of stochastic reaction networks via adaptive data partitioning”. *SIAM J. Sci. Comp.*, 31:6, 2010.

Input correlations: Rosenblatt transformation

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

$$
\begin{aligned}
\eta_1 &= F_1(\lambda_1) \\
\eta_2 &= F_{2|1}(\lambda_2|\lambda_1) \\
\eta_3 &= F_{3|2,1}(\lambda_3|\lambda_2, \lambda_1) \\
&\vdots \\
\eta_d &= F_{d|d-1,\ldots,1}(\lambda_d|\lambda_{d-1}, \ldots, \lambda_1)
\end{aligned}
$$

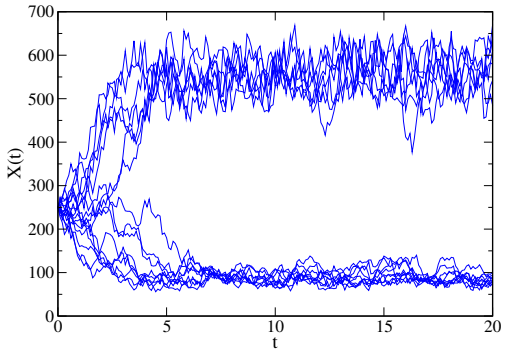


- Inverse Rosenblatt transformation $\boldsymbol{\lambda} = R^{-1}(\boldsymbol{\eta})$ ensures a well-defined input PC construction [S. *et al.*, 2010]

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

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

Stochastic systems: quadrature methods choke



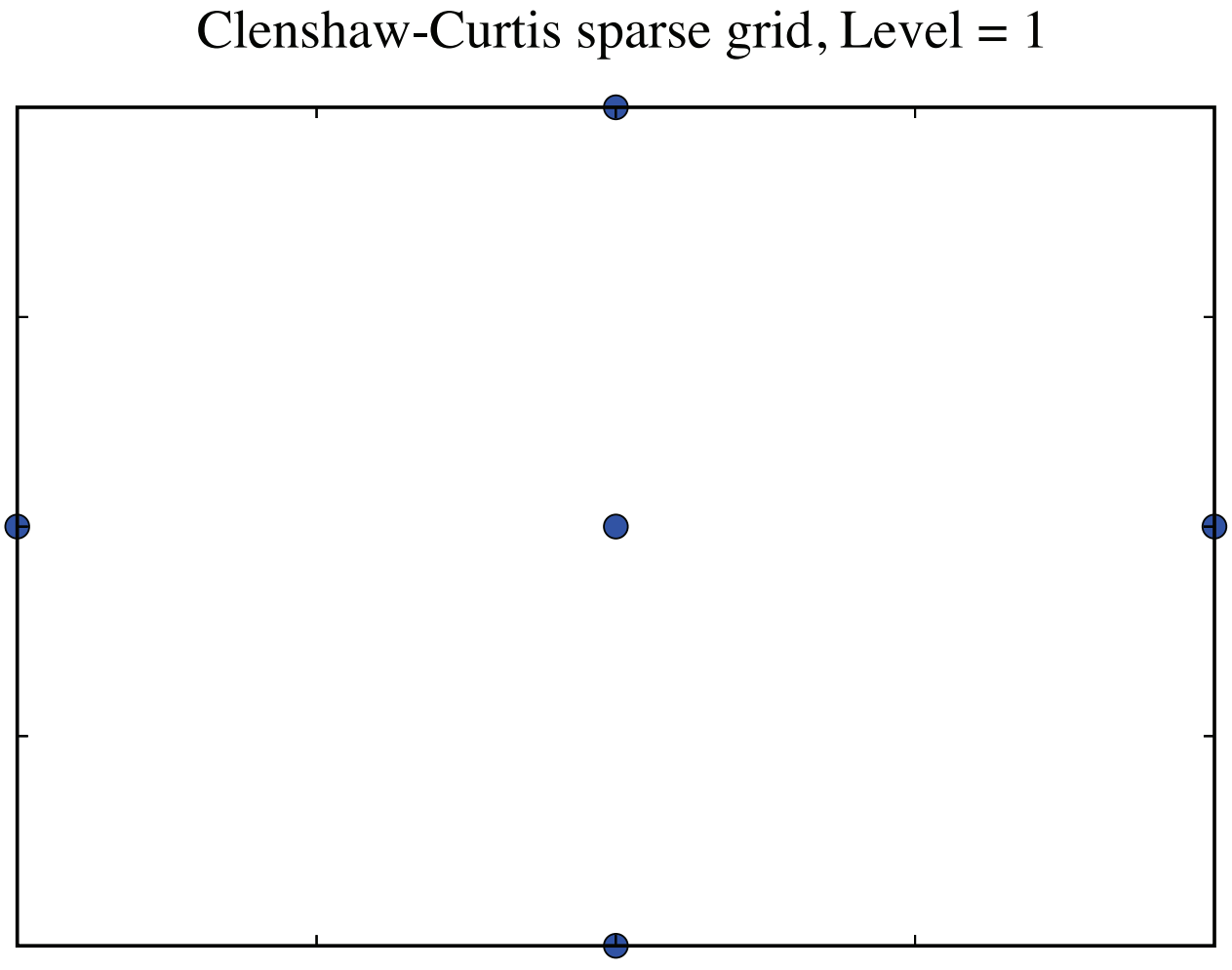
- Stochastic chemical kinetics
  [Gillespie, 1977]
- Climate buzzword: stochastic physics
  [Palmer & WIlliams, 2009]

- Quadrature formulae presume a degree of smoothness

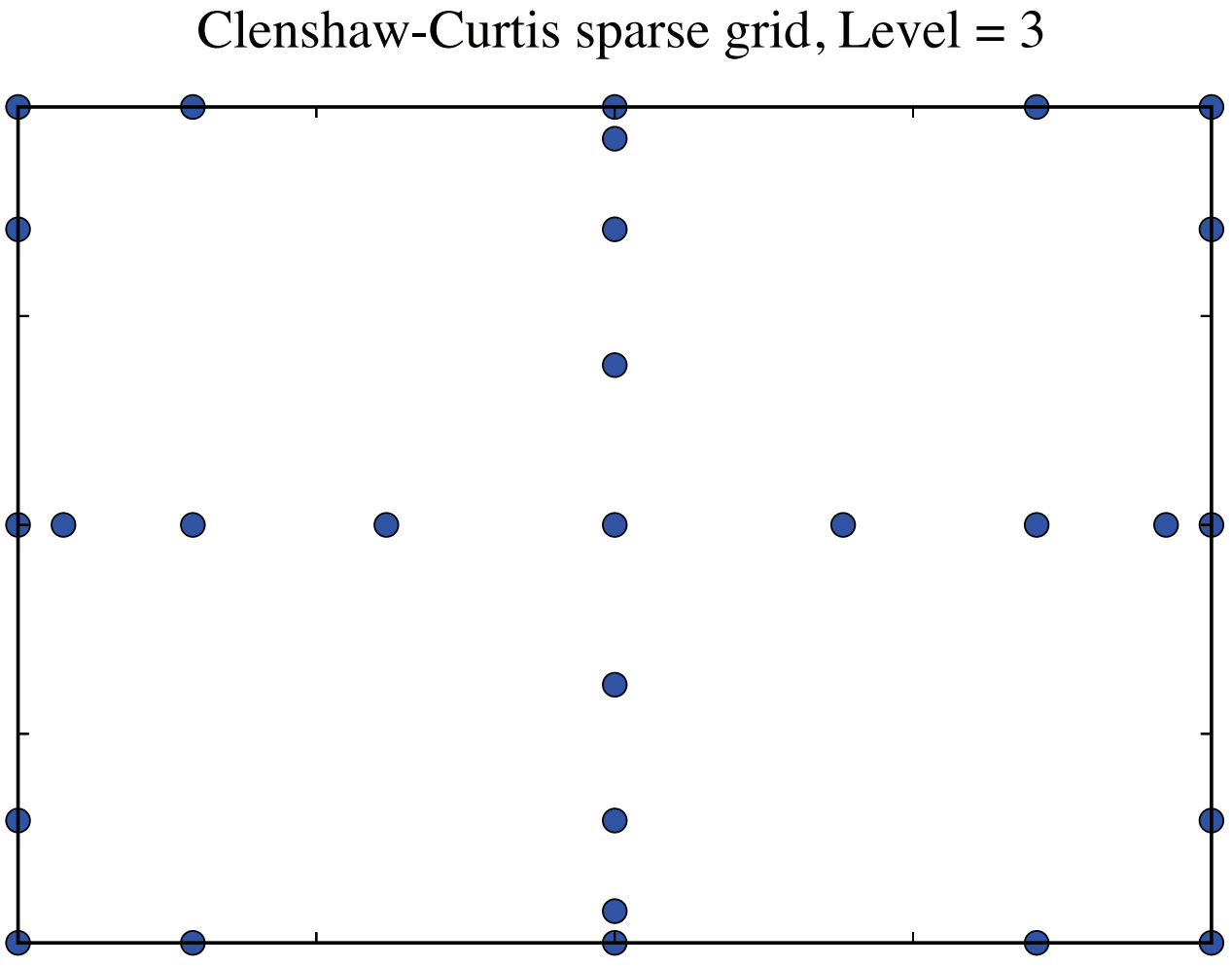
$$u_k = \frac{1}{\left\langle \Psi_k^2 \right\rangle} \int u(\lambda(\boldsymbol{\eta}))\, \Psi_k(\boldsymbol{\eta}) \pi(\boldsymbol{\eta}) d\boldsymbol{\eta} \approx \sum_* u(\boldsymbol{\lambda}(\boldsymbol{\eta}_*)) \Psi_k(\boldsymbol{\eta}_*) w_*$$

- 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

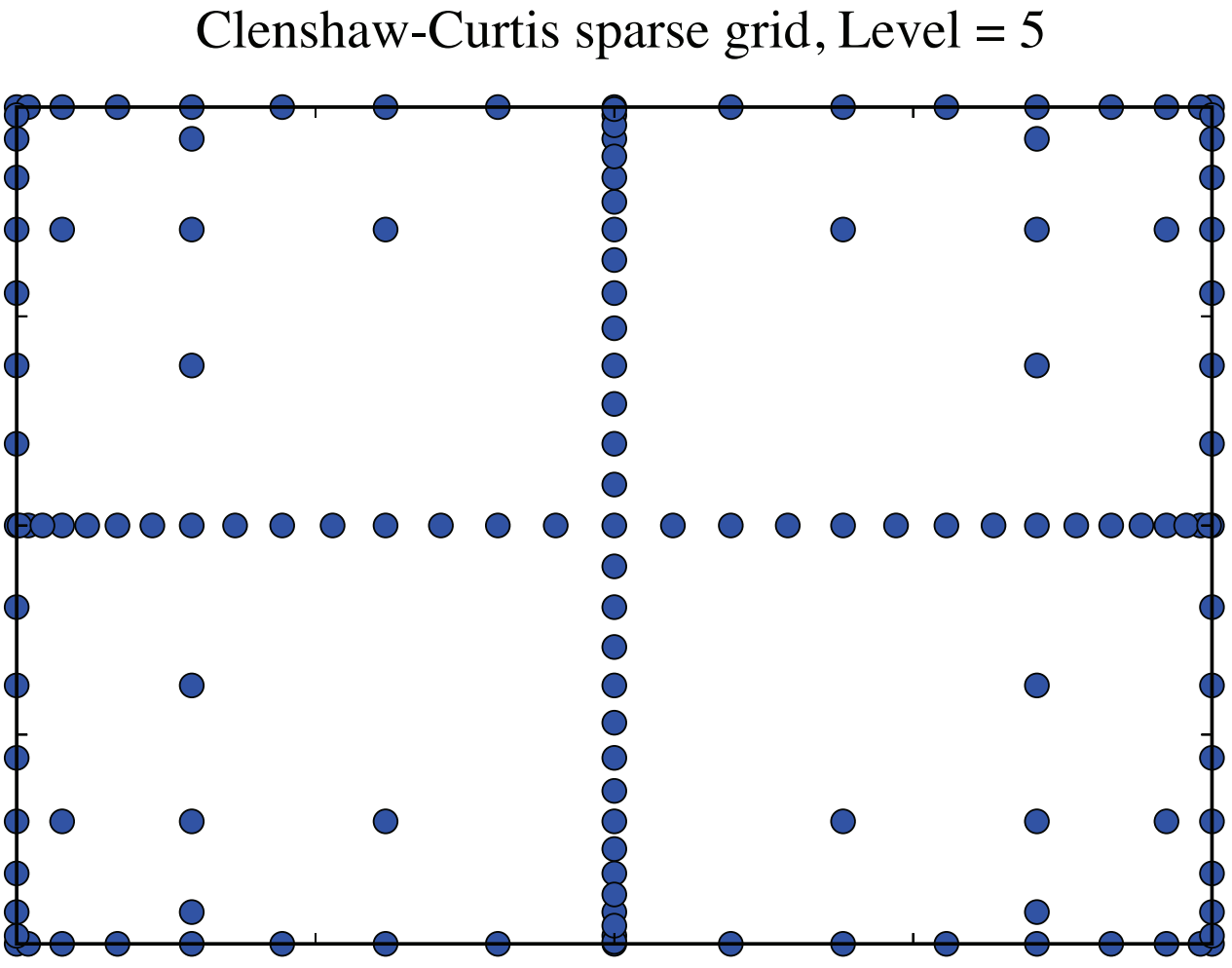
Sparse quadrature integration well-suited for high-dimensional *smooth* integrands



Sparse quadrature integration well-suited for high-dimensional *smooth* integrands



Sparse quadrature integration well-suited for high-dimensional *smooth* integrands



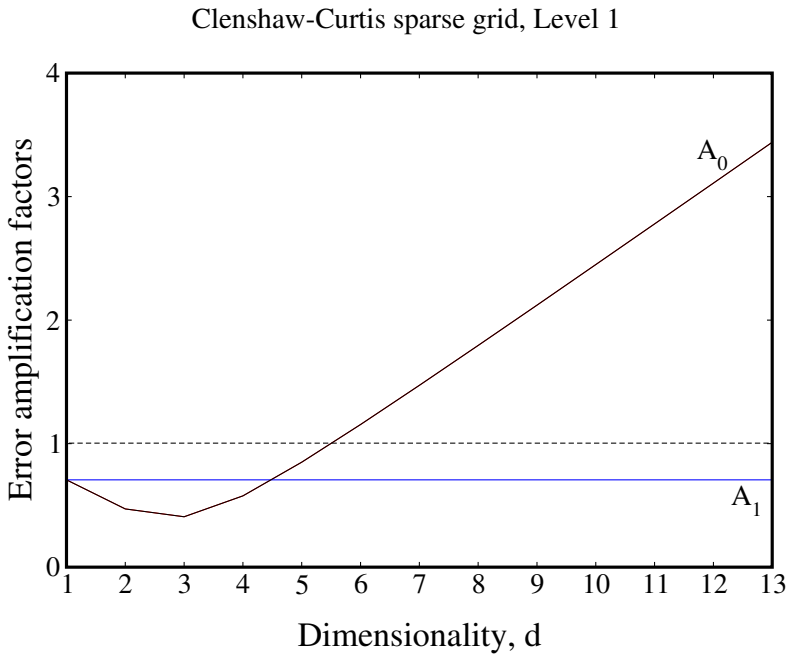
Sparse quadrature integration fails for noisy integrands

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

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

$$c_k \approx \frac{1}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle} \sum_{j=1}^{Q} Y(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j) w_j$$

$\text{Noise}_Y \sim \sigma \Longrightarrow \text{Error}_{c_k} \sim A_k \sigma$



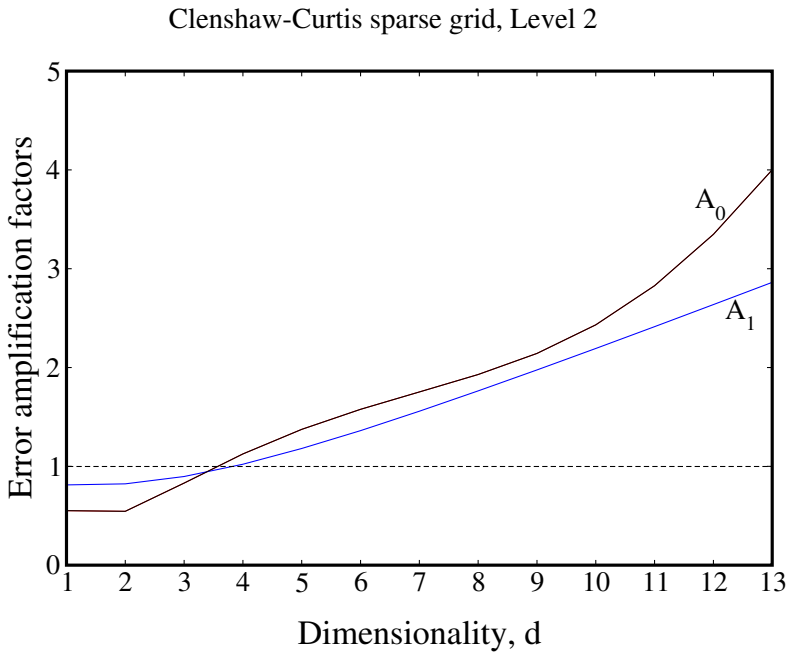
- amplification factor A_k grows with dimensionality
  - CC, level 1: $A_0 = \frac{1}{3}\sqrt{(d-3)^2 + \frac{d}{2}}, \qquad A_1 = \frac{1}{\sqrt{2}}.$
- blame the negative weights.
- for full quadrature, $\frac{1}{n^{d/2}} \leq A_0 \leq 1$, no amplification!

Sparse quadrature integration fails for noisy integrands

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

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

$$c_k \approx \frac{1}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle} \sum_{j=1}^{Q} Y(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j) w_j$$



$\text{Noise}_Y \sim \sigma \Longrightarrow \text{Error}_{c_k} \sim A_k \sigma$

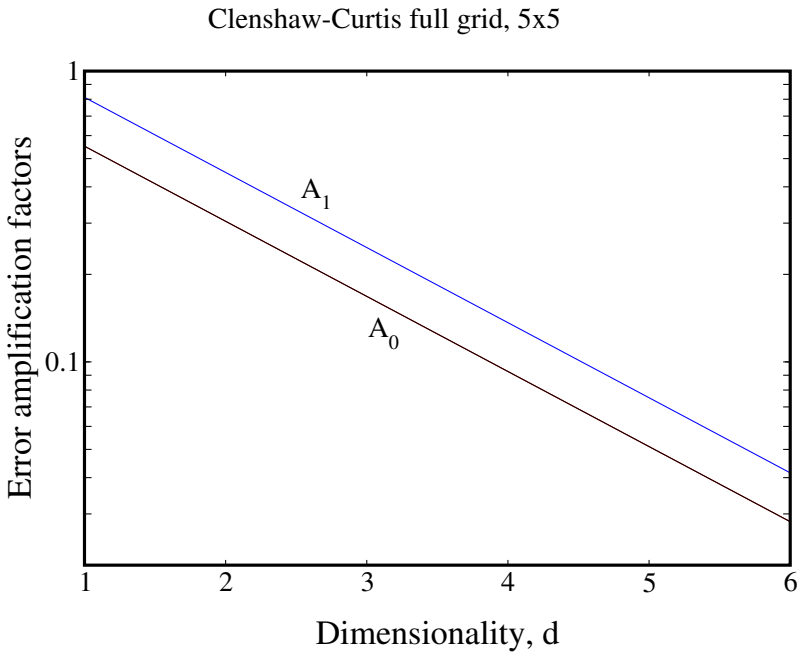
- amplification factor A_k grows with dimensionality
  - CC, level 1: $A_0 = \frac{1}{3}\sqrt{(d-3)^2 + \frac{d}{2}}, \qquad A_1 = \frac{1}{\sqrt{2}}.$
- blame the negative weights.
- for full quadrature, $\frac{1}{n^{d/2}} \leq A_0 \leq 1$, no amplification!

Sparse quadrature integration fails for noisy integrands

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

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

$$c_k \approx \frac{1}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle} \sum_{j=1}^{Q} Y(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j) w_j$$



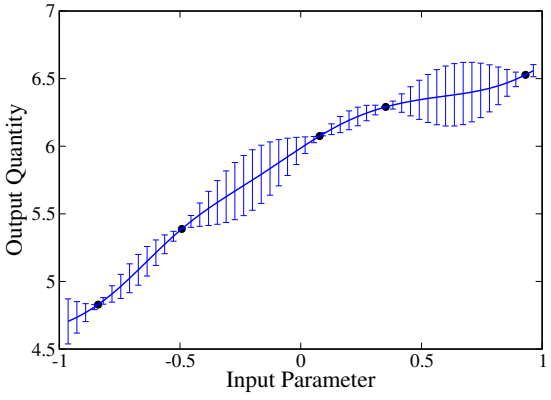
$\text{Noise}_Y \sim \sigma \Longrightarrow \text{Error}_{c_k} \sim A_k \sigma$

- amplification factor A_k grows with dimensionality
  - CC, level 1: $A_0 = \frac{1}{3}\sqrt{(d-3)^2 + \frac{d}{2}}, \qquad A_1 = \frac{1}{\sqrt{2}}.$
- blame the negative weights.
- for full quadrature, $\frac{1}{n^{d/2}} \leq A_0 \leq 1$, no amplification!

Limited Data

Both observational experiments and computer model simulations are expensive.

- Need to infer functional representation based on limited number of model runs/experiments.
  - Interpolation (kriging)
  - Gaussian Process emulation to assess the lack-of-knowledge [O'Hagan]
  - Extended to stochastic model setting



- *Bayesian* experimental design
  - What are the best locations to take observations?
  - At which parameter sets to run climate models to gain maximal information?