Analyzing Sparse-Grid UQ Techniques for Verification within Predictive Science

Clayton Webster*

Collaborators: Ivo Babuška, John Burkardt, Max Gunzburger, Fabio Nobile & Raul Tempone

*Computer Science Research Institute (CSRI), Sandia National Laboratories

Aug. 15th, 2007

Outline

Sandia National Laboratories
A Department of Energy National Laboratory

1. Background: Uncertainty Quantification (UQ)
2. Multivariate Integration
3. Sparse Grid Methods
4. Dimension-Adaptive (Anisotropic) Sparse Grids
5. Examples: Stochastic Collocation FEM
6. Concluding remarks

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94-AL85000.

Probabilistic or Stochastic models?

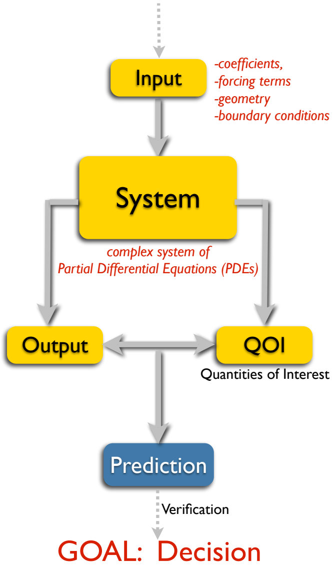
Many engineering applications (especially those predicting future events) are affected by a relatively large amount of uncertainty in the input data such as model coefficients, forcing terms, boundary conditions, geometry, etc.

- For example weather forecasting, i.e. hurricane paths.
- More generally, action from the surrounding environment.
- The model itself may contain an incomplete description of parameters, processes or fields (not possible or too costly to measure).
- There may be small, unresolved scales in the model that act as a kind of background noise (i.e. macro behavior from micro structure).

All these and many others introduce uncertainty in the model.

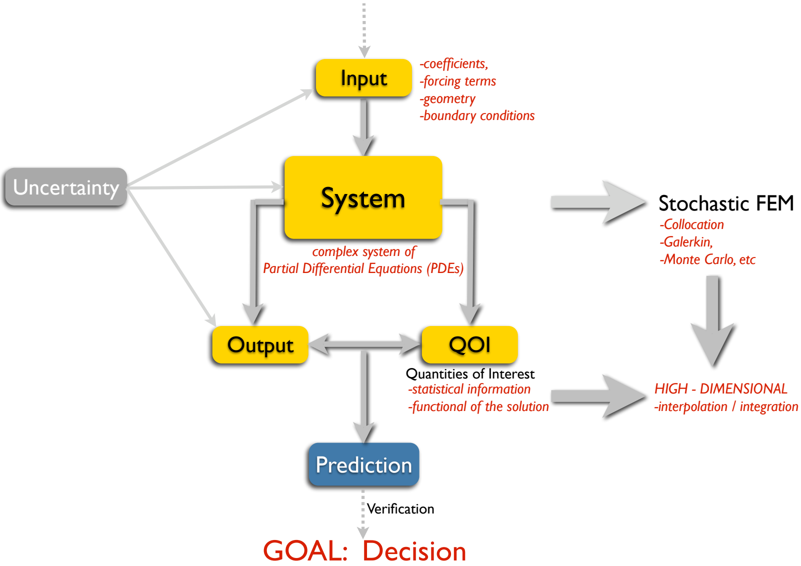
The Computational Model (SPDE)

From Real World to Predictions to Decisions



The Computational Model (SPDE)

From Real World to Predictions to Decisions



Stochastic formulation of uncertainty

Continuous solutions to SPDEs

Consider a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $D \subset \mathbb{R}^d$, $d = 1, 2, 3$, D convex, polygonal bounded.

$$\Downarrow$$

The input data (coefficients, forcing terms, geometries, boundary conditions) and the solution $u(\omega, x) \in \Omega \times D$ to your complex SPDE system are random (stochastic) functions (fields).

Goal of the computations

Example Quantities of Interest (QOI)

GOAL: to approximate statistical moments of u or some quantity of interest depending on u:

$$\mathbf{\Phi}_u = \langle \mathbf{\Phi}(u) \rangle := \mathbb{E}\left[\mathbf{\Phi}(u)\right] = \int_{\Omega} \int_{D} \Phi(u(\omega, x), \omega, x) dx \, \mathbb{P}(d\omega)$$

e.g. $\overline{u} = \mathbb{E}[u](x)$, OR $Var_u = \mathbb{E}[(\widetilde{u})^2](x)$, where $\widetilde{u} = u - \overline{u}$, OR

$$\mathbb{P}\left\{u \leq u_0\right\} = \mathbb{P}\left[\left\{\omega \in \Omega \, : \, u(\omega) \leq u_0\right\}\right] = \mathbb{E}\left[\chi_{\{u \leq u_0\}}\right]$$

With Non-Intrusive methods the **desired** QOI become expectations of some functional, that can be transformed into computing (high dimensional) integrals over a N-dimensional space Γ^N.

ASIDE: We may want to recover the stochastic response surface which can be accomplished via Stochastic Collocation (high dimensional interpolation over Γ^N).

Goal of the computations

Example Quantities of Interest (QOI)

Sandia National Laboratories

A Department of Energy National Laboratory

GOAL: to approximate statistical moments of u or some quantity of interest depending on u:

$$\Phi_u = \langle \Phi(u) \rangle := \mathbb{E}\left[\Phi(u)\right] = \int_\Omega \int_D \Phi(u(\omega, x), \omega, x)\, dx\, \mathbb{P}(d\omega)$$

e.g. $\overline{u} = \mathbb{E}[u](x)$, OR $Var_u = \mathbb{E}[(\widetilde{u})^2](x)$, where $\widetilde{u} = u - \overline{u}$, OR

$$\mathbb{P}\{u \leq u_0\} = \mathbb{P}\left[\{\omega \in \Omega \,:\, u(\omega) \leq u_0\}\right] = \mathbb{E}\left[\chi_{\{u \leq u_0\}}\right]$$

With Non-Intrusive methods the **desired** QOI become expectations of some functional, that can be transformed into computing (high dimensional) integrals over a N-dimensional space Γ^N.

ASIDE: We may want to recover the stochastic response surface which can be accomplished via Stochastic Collocation (high dimensional interpolation over Γ^N).

Multivariate integration

Computing high-dimensional integrals

Sandia National Laboratories

A Department of Energy National Laboratory

- $\Gamma_n \equiv Y_n(\Omega) \subset \mathbb{R}$, WLOG $\Gamma_n = [-1,1]$ bounded
- $\Gamma^N = \prod_{n=1}^{N} \Gamma_n \subset \mathbb{R}^N$ where $N \sim \mathcal{O}(10)$ – higher dimensions.
- $[Y_1, Y_2, \ldots, Y_N]$ have a joint probability density function $\rho : \Gamma^N \to \mathbb{R}_+$, with $\rho \in L^\infty(\Gamma^N)$, i.e. for $y \in \Gamma^N$

$$\mathbb{P}\left(Z \in \gamma \subset \Gamma^N\right) = \int_\gamma \rho(y)\, dy,$$

i.e. transform the measure $\mathbb{P}$ to $\mathbb{R}^N$.

Predicting the behavior of a physical system with random input information often requires the approximation of multi-dimensional integrals (expectations):

$$\mathscr{I}^{(N)}[u](x) = \int_{\Gamma^N} u(y,x)\rho(y)dy \quad (= \mathbb{E}[u](x))$$

where $y \in \Gamma^N$ and $x \in \overline{D}$.

Multivariate integration

Computing high-dimensional integrals

- $\Gamma_n \equiv Y_n(\Omega) \subset \mathbb{R}$, WLOG $\Gamma_n = [-1,1]$ bounded
- $\Gamma^N = \prod_{n=1}^{N} \Gamma_n \subset \mathbb{R}^N$ where $N \sim \mathcal{O}(10)$ – higher dimensions.
- $[Y_1, Y_2, \ldots, Y_N]$ have a joint probability density function $\rho : \Gamma^N \to \mathbb{R}_+$, with $\rho \in L^\infty(\Gamma^N)$, i.e. for $y \in \Gamma^N$

$$\mathbb{P}\left(Z \in \gamma \subset \Gamma^N\right) = \int_\gamma \rho(y)\, dy,$$

i.e. transform the measure $\mathbb{P}$ to $\mathbb{R}^N$.

Predicting the behavior of a physical system with random input information often requires the approximation of multi-dimensional integrals (expectations):

$$\mathscr{I}^{(N)}[u](x) = \int_{\Gamma^N} u(y,x)\rho(y)dy \quad (= \mathbb{E}[u](x))$$

where $y \in \Gamma^N$ and $x \in \overline{D}$.

General approaches

Direct vs. Interpolatory Quadrature

$$\mathscr{I}^{(N)}[u] = \int_{\Gamma^N} u(y)\rho(y)dy$$

1. **Direct Quadrature:** Monte Carlo, Quasi Monte Carlo, Latin Hypercube Sampling, Lattice Rules, etc.

$$\mathscr{I}^{(N)}[u] \approx \sum_{k=1}^{M} u(y_k)\rho(y_k)w_k \equiv \mathcal{Q}^{(N)}[u], \quad y_k \in \Gamma^N$$

2. **Interpolatory Quadrature:** Here we first want to approximate (recover) in $u : \Gamma^N \to \mathbb{R}$, i.e.

$$\mathcal{I}^{(N)}[u] = \sum_{k=1}^{M} u(y_k) l_k^{(N)}(y)$$

$$\Rightarrow \mathscr{I}^{(N)}[u] \approx \int_{\Gamma^N} \mathcal{I}^{(N)}[u]\rho(y)dy = \sum_{k=1}^{M} u(y_k)\rho(y_k) \underbrace{\int_{\Gamma^N} l_k^{(N)}(y)dy}_{w_k}$$

Sandia National Laboratories

A Department of Energy National Laboratory

General approaches

Direct vs. Interpolatory Quadrature

$$\mathscr{I}^{(N)}[u] = \int_{\Gamma^N} u(y)\rho(y)dy$$

1. **Direct Quadrature:** Monte Carlo, Quasi Monte Carlo, Latin Hypercube Sampling, Lattice Rules, etc.

$$\mathscr{I}^{(N)}[u] \approx \sum_{k=1}^{M} u(y_k)\rho(y_k)w_k \equiv \mathcal{Q}^{(N)}[u], \quad y_k \in \Gamma^N$$

2. **Interpolatory Quadrature:** Here we first want to approximate (recover) in $u : \Gamma^N \to \mathbb{R}$, i.e.

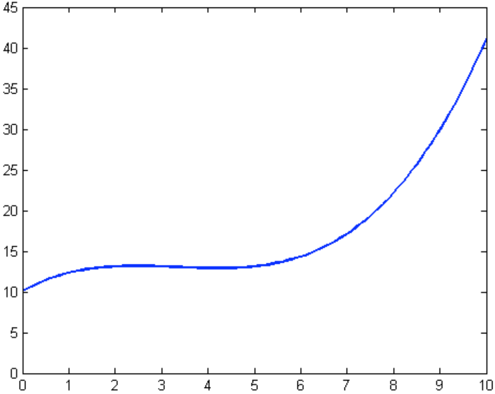
$$\mathcal{I}^{(N)}[u] = \sum_{k=1}^{M} u(y_k) l_k^{(N)}(y)$$

$$\Rightarrow \mathscr{I}^{(N)}[u] \approx \int_{\Gamma^N} \mathcal{I}^{(N)}[u]\rho(y)dy = \sum_{k=1}^{M} u(y_k)\rho(y_k) \underbrace{\int_{\Gamma^N} l_k^{(N)}(y)dy}_{w_k}$$

Interpolatory Quadrature

A simple example

A simple function to integrate

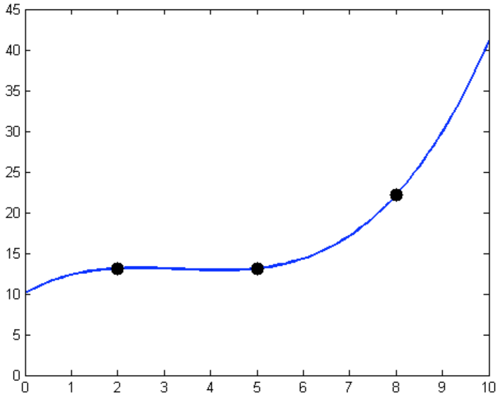


$u(y)$ is a given function

Interpolatory Quadrature

A simple example

Selected function values

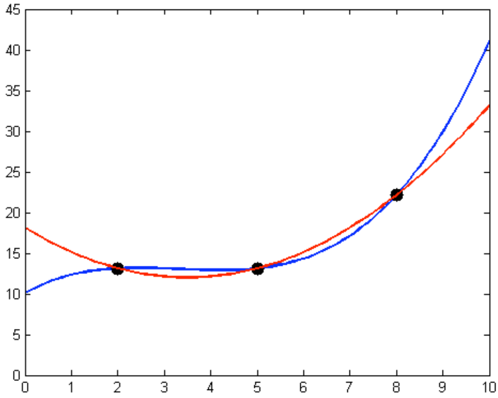


Evaluate $u(y)$ at M values $\{u(y_1), u(y_2), \ldots, u(y_M)\}$

Interpolatory Quadrature

A simple example

The interpolant

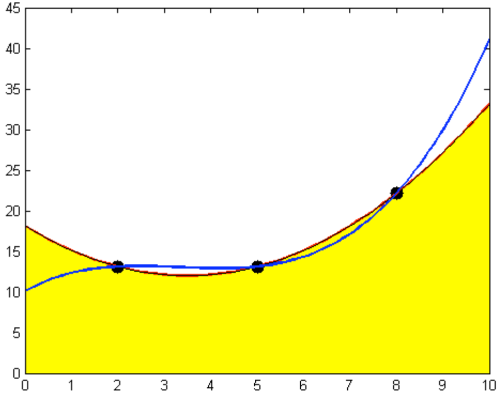


Determine the approximate polynomial $\mathcal{I}[u]$

Interpolatory Quadrature

A simple example

The area under the curve



Integrate the approximating polynomial EXACTLY

Full tensor product interpolation

The simplest multi-dimensional interpolant

Case $N = 1$: Let $i \in \mathbb{N}_+$ and $\{y_1^i, \ldots, y_{m_i}^i\} \subset [-1, 1]$.

$$\mathscr{U}^i(u)(y) = \sum_{j=1}^{m_i} u(y_j^i) \cdot l_j^i(y),$$

- $l_j^i \in \mathcal{P}_{m_i - 1}(\Gamma^1)$ are Lagrange poly's of degree $p_i = m_i - 1$.

Case $N > 1$: let $\mathbf{i} = (i_1, \ldots, i_N) \in \mathbb{N}_+^N$ be a multi-index.

$$\begin{aligned}\mathcal{I}_{\mathbf{i}}^{(N)} u(y) &= \left(\mathscr{U}^{i_1} \otimes \cdots \otimes \mathscr{U}^{i_N}\right)(u)(y) \\ &= \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_N=1}^{m_{i_N}} u\left(y_{j_1}^{i_1}, \ldots, y_{j_N}^{i_N}\right) \cdot \left(l_{j_1}^{i_1} \otimes \cdots \otimes l_{j_N}^{i_N}\right).\end{aligned}$$

- the interpolation requires $M = m_{i_1} \times \cdots \times m_{i_N}$ function evaluations (PDE solves).

Full tensor product interpolation

The simplest multi-dimensional interpolant

Sandia National Laboratories

A Department of Energy National Laboratory

Case $N = 1$: Let $i \in \mathbb{N}_+$ and $\{y_1^i, \dots, y_{m_i}^i\} \subset [-1, 1]$.

$$\mathscr{U}^i(u)(y) = \sum_{j=1}^{m_i} u(y_j^i) \cdot l_j^i(y),$$

- $l_j^i \in \mathcal{P}_{m_i-1}(\Gamma^1)$ are Lagrange poly's of degree $p_i = m_i - 1$.

Case $N > 1$: let $\mathbf{i} = (i_1, \dots, i_N) \in \mathbb{N}_+^N$ be a multi-index.

$$\begin{aligned}
\mathcal{I}_{\mathbf{i}}^{(N)} u(y) &= \left(\mathscr{U}^{i_1} \otimes \cdots \otimes \mathscr{U}^{i_N}\right)(u)(y) \\
&= \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_N=1}^{m_{i_N}} u\left(y_{j_1}^{i_1}, \dots, y_{j_N}^{i_N}\right) \cdot \left(l_{j_1}^{i_1} \otimes \cdots \otimes l_{j_N}^{i_N}\right).
\end{aligned}$$

- the interpolation requires $M = m_{i_1} \times \cdots \times m_{i_N}$ function evaluations (PDE solves).

Full tensor product interpolation

Complexity of an *isotropic* FTP interpolant

For functions with u with bounded mixed derivatives of order r:

$$\mathcal{W}_r^{(N)} := \left\{ u : \Gamma^N \to \mathbb{R}, \left\| \frac{\partial^{|s|_1} u}{\partial y_1^{s_1} \dots \partial y_N^{s_N}} \right\|_\infty < \infty, \; s_i \le r \right\}$$

The amount of work to achieve a prescribed accuracy ε using the standard *isotropic* tensor product approach:

$$\varepsilon(M) = \left\| u - \mathcal{I}^{(N)}[u] \right\|_\infty \approx \mathcal{O}\left(M^{-r/N} \right)$$

- depends (in a bad way!) exponentially on the dimension

Example: Take $m = 4$ point rule using an *isotropic* tensor product

N	1	2	3	4	5	10	50	100
$M = m^N$	4	16	64	256	1024	$\sim 10^6$	$\sim 10^{30}$	$\sim 10^{60}$

Full tensor product interpolation

Complexity of an *isotropic* FTP interpolant

For functions with u with bounded mixed derivatives of order r:

$$\mathcal{W}_r^{(N)} := \left\{ u : \Gamma^N \to \mathbb{R}, \left\| \frac{\partial^{|s|_1} u}{\partial y_1^{s_1} \dots \partial y_N^{s_N}} \right\|_\infty < \infty, \, s_i \le r \right\}$$

The amount of work to achieve a prescribed accuracy ε using the standard *isotropic* tensor product approach:

$$\varepsilon(M) = \left\| u - \mathcal{I}^{(N)}[u] \right\|_\infty \approx \mathcal{O}\left(M^{-r/N} \right)$$

- depends (in a bad way!) exponentially on the dimension

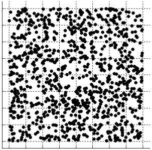
Example: Take $m = 4$ point rule using an *isotropic* tensor product

N	1	2	3	4	5	10	50	100
$M = m^N$	4	16	64	256	1024	$\sim 10^6$	$\sim 10^{30}$	$\sim 10^{60}$

Direct methods

Attempting to cope with the *curse of dimensionality*

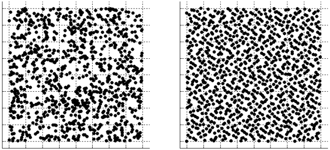
1. Monte Carlo methods:
   abscissas are (pseudo-) random numbers
2. Quasi-Monte Carlo methods:
   abscissas are low discrepancy sequences
3. Latin Hypercube Sampling:
   abscissas are chosen to ensure "good" spacing in each 1-dimensional component
4. Lattice rules:
   abscissas are "good" lattice points



Direct methods

Attempting to cope with the *curse of dimensionality*

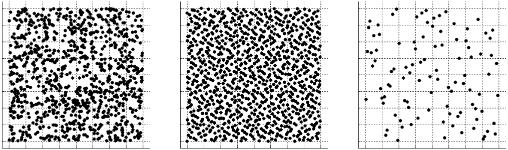
1. Monte Carlo methods:
   abscissas are (pseudo-) random numbers
2. Quasi-Monte Carlo methods:
   abscissas are low discrepancy sequences
3. Latin Hypercube Sampling:
   abscissas are chosen to ensure "good" spacing in each 1-dimensional component
4. Lattice rules:
   abscissas are "good" lattice points



Direct methods

Attempting to cope with the *curse of dimensionality*

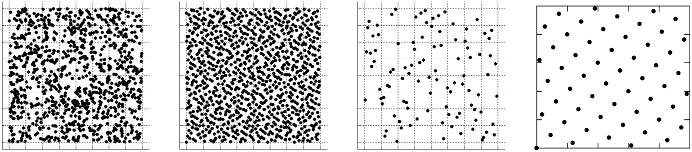
1. **Monte Carlo methods**:
   abscissas are (pseudo-) random numbers
2. **Quasi-Monte Carlo methods**:
   abscissas are low discrepancy sequences
3. **Latin Hypercube Sampling**:
   abscissas are chosen to ensure “good” spacing in each 1-dimensional component
4. **Lattice rules**:
   abscissas are “good” lattice points



Direct methods

Attempting to cope with the *curse of dimensionality*

1. Monte Carlo methods:
   abscissas are (pseudo-) random numbers
2. Quasi-Monte Carlo methods:
   abscissas are low discrepancy sequences
3. Latin Hypercube Sampling:
   abscissas are chosen to ensure "good" spacing in each 1-dimensional component
4. Lattice rules:
   abscissas are "good" lattice points



Complexity comparisons

Asymptotic error estimates

The amount of work to reach a required accuracy ε:

- Isotropic TP – $\varepsilon(M) = \mathcal{O}\left(M^{-r/N}\right)$,
- MC – $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$,
- QMC, LHS – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{N}\right)$,
- Lattice rules – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{(N+1)/2}\right)$,

Comments: even for moderate dimensions we suffer from the *curse of dimensionality* since ε grows exponentially in the number of random variables (dimensions) N.

Complexity comparisons

Asymptotic error estimates

The amount of work to reach a required accuracy ε:

- Isotropic TP – $\varepsilon(M) = \mathcal{O}\left(M^{-r/N}\right)$,
- MC – $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$,
- QMC, LHS – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{N}\right)$,
- Lattice rules – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{(N+1)/2}\right)$,

Comments: no dependence on the dimension N of the stochastic space but slow convergence.

Complexity comparisons

Asymptotic error estimates

The amount of work to reach a required accuracy ε:

- Isotropic TP – $\varepsilon(M) = \mathcal{O}\left(M^{-r/N}\right)$,
- MC – $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$,
- QMC, LHS – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{N}\right)$,
- Lattice rules – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{(N+1)/2}\right)$,

Comments: This is almost half an order better than the complexity of the MC approach. However, the dimension N enters through a logarithmic term and this dependence on N causes **SLOWER** convergence for large N.

Complexity comparisons

Asymptotic error estimates

The amount of work to reach a required accuracy ε:

- Isotropic TP – $\varepsilon(M) = \mathcal{O}\left(M^{-r/N}\right)$,
- MC – $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$,
- QMC, LHS – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{N}\right)$,
- Lattice rules – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{(N+1)/2}\right)$,

Comments: similar observations for the QMC and LHS approaches can be made here.

Complexity comparisons

Asymptotic error estimates

The amount of work to reach a required accuracy ε:

- Isotropic TP – $\varepsilon(M) = \mathcal{O}\left(M^{-r/N}\right)$,
- MC – $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$,
- QMC, LHS – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{N}\right)$,
- Lattice rules – $\varepsilon(M) = \mathcal{O}\left(M^{-1}(\log(M))^{(N+1)/2}\right)$,

NOTE: The convergence rate of the direct methods do not depend on the smoothness of the function. The first method which makes use of the smoothness of the integrand and at the same time reduces the *curse of the dimensionality* is the Sparse Grid method.

Sparse tensor product interpolation

The classical (*isotropic*) approach - S. Smolyak '63

- A **tough** competitor of the direct methods for coping with the *curse of dimensionality* is the Sparse Grid approach.
- Unlike MC, QMC, LHS, etc. the Sparse Grid approach can exploit the inherent smoothness of the function we are interpolating (integrating).

Application areas include:

- numerical solution of PDEs (Sparse Finite Differences)
- integral equations
- eigenvalue problems
- quadrature
- interpolation
- data compression
- data mining

The isotropic Sparse Grid Method

Sandia National Laboratories
A Department of Energy National Laboratory

Assume $N > 1$, $|\mathbf{i}| = i_1 + \cdots + i_N$, $\vartheta^i = \{y_1^i, \ldots, y_{m_i}^i\} \subset [-1,1]$:

$$\mathscr{U}^0 = 0, \quad \Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}.$$

$$X(w,N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

$$Y(w,N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; w - N + 1 \leq \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

The isotropic Sparse Grid Method

Assume $N > 1$, $|\mathbf{i}| = i_1 + \cdots + i_N$, $\vartheta^i = \{y_1^i, \ldots, y_{m_i}^i\} \subset [-1, 1]$:

$$\mathscr{U}^0 = 0, \quad \Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}.$$

$$X(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; \sum_{n=1}^N (i_n - 1) \leq w \right\}$$

$$Y(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; w - N + 1 \leq \sum_{n=1}^N (i_n - 1) \leq w \right\}$$

For integers $w \in \mathbb{N}$ the **isotropic** Smolyak algorithm is given by

$$\mathscr{A}(w, N) = \sum_{\mathbf{i} \in X(w,N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N} \right)$$

The isotropic Sparse Grid Method

Assume $N > 1$, $|\mathbf{i}| = i_1 + \cdots + i_N$, $\vartheta^i = \{y_1^i, \ldots, y_{m_i}^i\} \subset [-1, 1]$:

$$\mathscr{U}^0 = 0, \quad \Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}.$$

$$X(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

$$Y(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \; : \; w - N + 1 \leq \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

For integers $w \in \mathbb{N}$ the **isotropic** Smolyak algorithm is given by

$$\mathscr{A}(w, N) = \sum_{\mathbf{i} \in X(w,N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N} \right)$$

$$\mathscr{A}(w, N) = \sum_{\mathbf{i} \in Y(w,N)} (-1)^{w+N-|\mathbf{i}|} \binom{N-1}{w+N-|\mathbf{i}|} \cdot \left(\mathscr{U}^{i_1} \otimes \cdots \otimes \mathscr{U}^{i_N} \right).$$

The isotropic Sparse Grid Method

Assume $N > 1$, $|\mathbf{i}| = i_1 + \cdots + i_N$, $\vartheta^i = \{y_1^i, \ldots, y_{m_i}^i\} \subset [-1, 1]$:

$$\mathscr{U}^0 = 0, \quad \Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}.$$

$$X(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \, : \, \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

$$Y(w, N) := \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \, : \, w - N + 1 \leq \sum_{n=1}^{N} (i_n - 1) \leq w \right\}$$

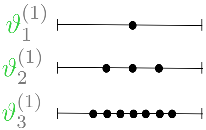
To compute the interpolant $\mathscr{A}(w, N)(u)$ sample the "sparse grid"

$$\mathscr{H}(w, N) = \bigcup_{\mathbf{i} \in Y(w,N)} \left(\vartheta^{i_1} \times \cdots \times \vartheta^{i_N} \right)$$

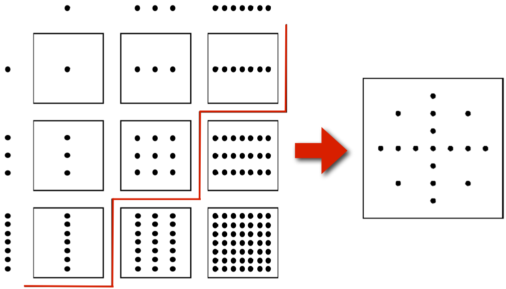
Example: $N = 2$ Sparse grid

Nested rules minimize the amount of work

Nested equidistant grids with $m_i = 1, 3$ and 7:



Grids $\vartheta_i^{(1)} \otimes \vartheta_j^{(1)}$ for $i, j \leq 3$ and the *isotropic* sparse grid $\mathscr{H}(3, 2)$:



Asymptotic Accuracy

Sparse grids vs. tensor products

Definition

The asymptotic accuracy of an interpolation (quadrature) rule is determined by the highest **total degree** $\mathbf{p}$ for which we can guarantee that all monomials will be interpolated (integrated) exactly.

Suppose $\mathscr{U}^i_{max}$ can interpolate a polynomial of degree $\mathbf{p}$ then: N-dimensional sparse grid interpolant can **exactly** recover a polynomial of **total degree** $\mathbf{p}$, i.e. $\sum_{|\mathbf{i}|\leq \mathbf{p}} c_{\mathbf{i}}\, y_1^{i_1} \dots y_N^{i_N}$

- For $N = 2$, monomials of **total degree** $\mathbf{p} = 4$ are exactly y_1^4, $y_1^3 y_2$, $y_1^2 y_2^2$, $y_1 y_2^3$ and y_2^4

N-dimensional tensor product interpolant can **exactly** recover product of monomials of **degree** $\mathbf{p}$, i.e. $\prod_{i=1}^N c_i y_i^{\mathbf{p}}$.

Asymptotic Accuracy

Sparse grids vs. tensor products

Definition

The asymptotic accuracy of an interpolation (quadrature) rule is determined by the highest **total degree** $\mathbf{p}$ for which we can guarantee that all monomials will be interpolated (integrated) exactly.

Suppose $\mathscr{U}^i_{max}$ can interpolate a polynomial of degree $\mathbf{p}$ then:
N-dimensional sparse grid interpolant can **exactly** recover a polynomial of **total degree** $\mathbf{p}$, i.e. $\sum_{|\mathbf{i}|\leq \mathbf{p}} c_{\mathbf{i}}\, y_1^{i_1} \dots y_N^{i_N}$

- For $N = 2$, monomials of **total degree** $\mathbf{p} = 4$ are exactly $y_1^4,\ y_1^3 y_2,\ y_1^2 y_2^2,\ y_1 y_2^3$ and y_2^4

N-dimensional tensor product interpolant can **exactly** recover product of monomials of **degree** $\mathbf{p}$, i.e. $\prod_{i=1}^N c_i y_i^{\mathbf{p}}$.

Useless monomials

Sparse grids vs. tensor products

0					1				
1				y_1		y_2			
2			y_1^2		$y_1 y_2$		y_2^2		
3		y_1^3		$y_1^2 y_2$		$y_1 y_2^2$		y_2^3	
4	y_1^4		$y_1^3 y_2$		$y_1^2 y_2^2$		$y_1 y_2^3$		y_2^4
5		$y_1^4 y_2$		$y_1^3 y_2^2$		$y_1^2 y_2^3$		$y_1 y_2^4$	
6			$y_1^4 y_2^2$		$y_1^3 y_2^3$		$y_1^2 y_2^4$		
7				$y_1^4 y_2^3$		$y_1^3 y_2^4$			
8					$y_1^4 y_2^4$				

Monomials up to 4th degree. Those below the line are the useless monomials we capture (using tensor products) and **are not needed**...they don't help the asymptotic accuracy.

Types of sparse grids

Based on 1-dimensional sequences of abscissas

Sandia National Laboratories
A Department of Energy National Laboratory

Let m_i be the number of points used by 1-dimensional rule. Nested abscissas, i.e. $\vartheta^i \subset \vartheta^{i+1}$, produce nested grids, i.e. $\mathscr{H}(w,N) \subset \mathscr{H}(w+1,N)$

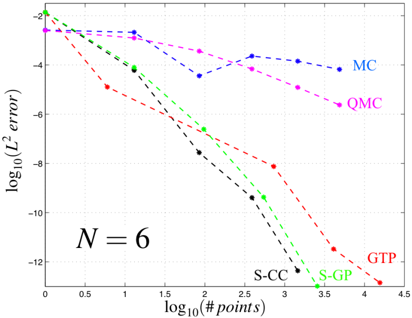
1. Newton-Cotes: nested equidistant abscissas by taking $m_1 = 1$ and $m_i = 2^{i-1} + 1$, for $i > 1$. Maximum degree of exactness $m_i - 1$. Can have (highly) negative weights causing numerical inaccuracies.
2. Clenshaw-Curtis: nested (same growth as above) Chebyshev points with maximum $m_i - 1$ degree of exactness.
3. Gauss: non-nested abscissas with $2m_i - 1$ degree of exactness.
4. Gauss-Patterson ('68): recursive expansion of the Kronrod ('65) scheme: • start with 3-point Gauss-Legendre formula
   • extend to $m_i = 2^i - 1$ and maintain the Kronrod properties
   • $3 \cdot 2^{i-1} - 1$ degree of exactness

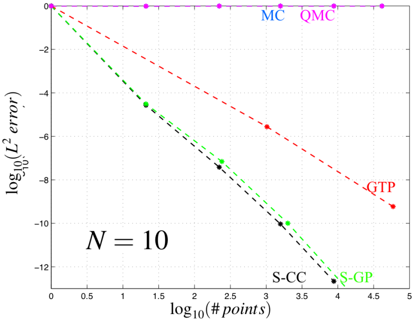
A simple example

Consider the following N-dimensional Genz integral:

$$\int_{\mathbb{R}^N} \exp\left(-\sum_{i=1}^{N} {c_i}^2 \left(y_i - w_i\right)^2\right) d\mathbf{y}$$

where c_i, w_i are sampled randomly from $U(0,1)$



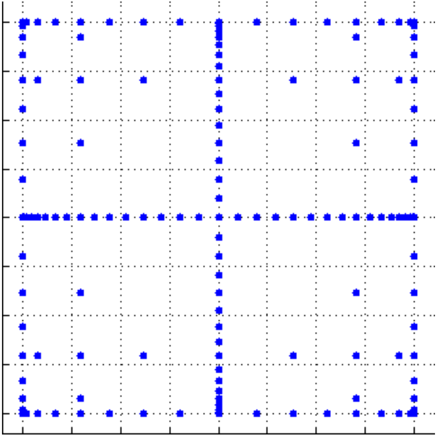


Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$$\Rightarrow |\mathbf{i}| \leq w + N = 7$$



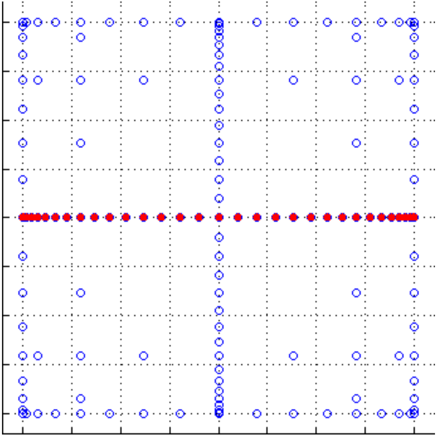
$\mathscr{A}(5, 2)$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$\Rightarrow |\mathbf{i}| \leq w + N = 7$



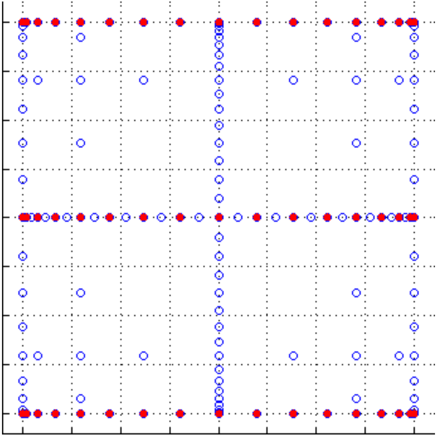
$\mathbf{i} = (6, 1) \Rightarrow (33 \times 1)$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$$\Rightarrow |\mathbf{i}| \leq w + N = 7$$



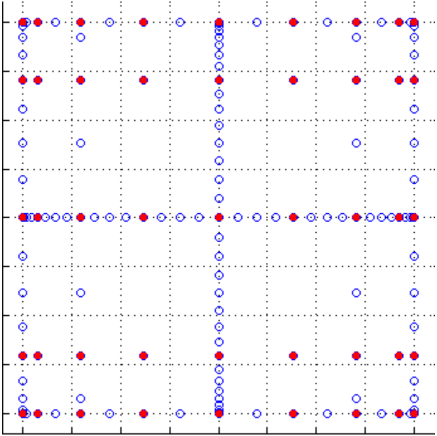
$$\mathbf{i} = (5, 2) \Rightarrow (17 \times 3)$$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$$\Rightarrow |\mathbf{i}| \leq w + N = 7$$

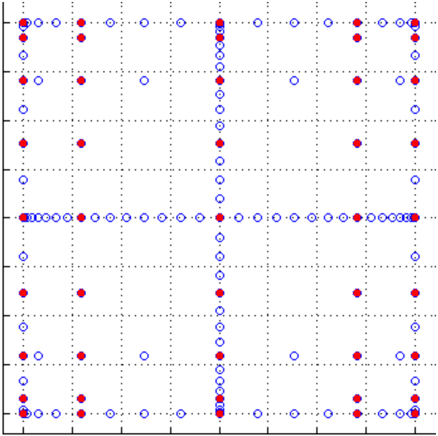


$$\mathbf{i} = (4, 3) \Rightarrow (9 \times 5)$$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$$N = 2 \text{ isotropic sparse grid: } w = 5$$
$$\Rightarrow |\mathbf{i}| \le w + N = 7$$



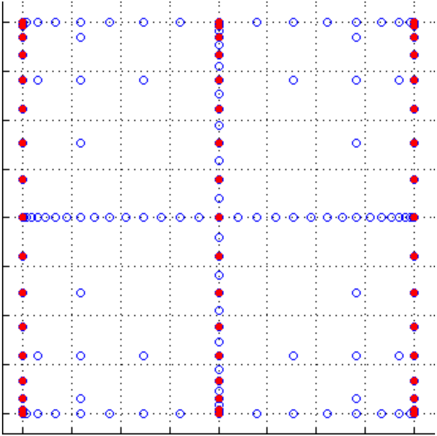
$$\mathbf{i} = (3, 4) \Rightarrow (5 \times 9)$$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$\Rightarrow |\mathbf{i}| \leq w + N = 7$



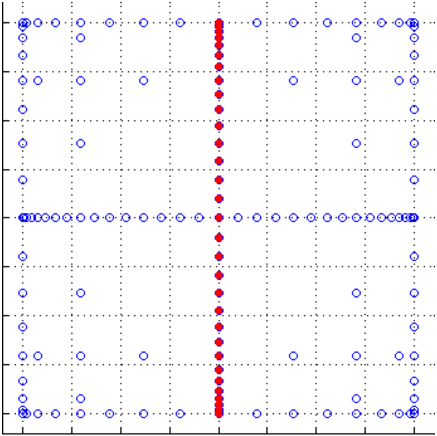
$\mathbf{i} = (2, 5) \Rightarrow (3 \times 17)$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

$$\Rightarrow |\mathbf{i}| \le w + N = 7$$



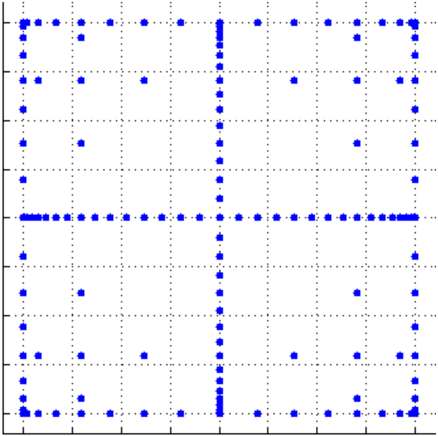
$$\mathbf{i} = (1, 6) \Rightarrow (1 \times 33)$$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

$N = 2$ isotropic sparse grid: $w = 5$

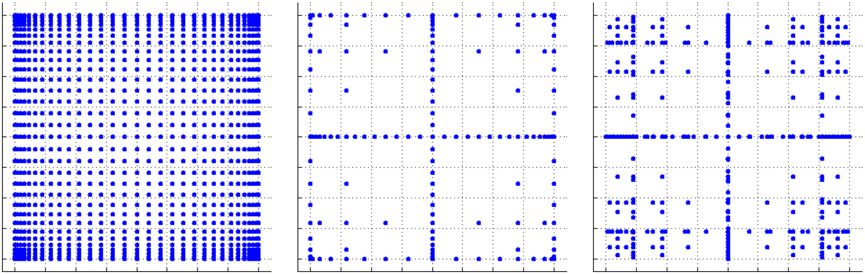
$$\Rightarrow |\mathbf{i}| \leq w + N = 7$$



$\mathscr{A}(5, 2)$

Generating Smolyak sparse grids: $N = 2$

Using Clenshaw-Curtis abscissas

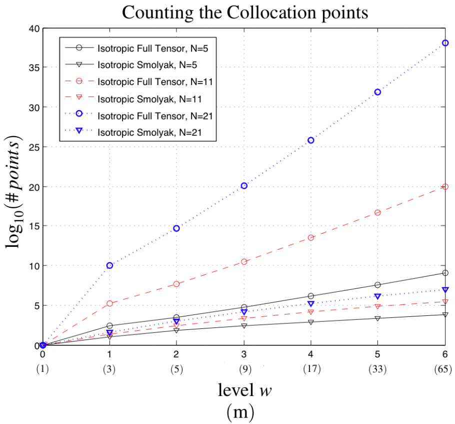


Isotropic FT — Smolyak C-C — Smolyak Gauss

For $N = 2$ and $w = 5$ we compare the full tensor product grid using the Clenshaw-Curtis abscissas with the isotropic Smolyak sparse grids utilizing the Clenshaw-Curtis abscissas and the Gaussian abscissas.

The curse of dimensionality

The need for sparse grids



For Γ^N with $N = 5, 11$ and 21 we plot the log of the number of distinct Clenshaw-Curtis points used by the *isotropic* sparse grid and the isotropic tensor grid vs. the level w(or the maximum number of points m employed in each direction).

Complexity of C-C sparse grids:

Why you really use this stuff.....

Recall: convergence of *isotropic* the FTP: $\varepsilon(M) \approx \mathcal{O}\left(M^{-r/N}\right)$

What about the *isotropic* sparse grid approach?

[Novak, Ritter, '95]

$$\varepsilon(M) = \|u - \mathscr{A}(w, N)u\|_\infty \approx \mathcal{O}\left(M^{-r}\left(\log M\right)^{(N-1)(r+1)}\right)$$

- "Exploits the smoothness r of the function (as opposed to MC, QMC, LHS, etc.)"
- "Improves upon the QMC convergence rate assuming $r > 1$"
- "Breaks the curse of dimensionality"

Complexity of C-C sparse grids:

Why you really use this stuff.....

Sandia National Laboratories

A Department of Energy National Laboratory

Recall: convergence of *isotropic* the FTP: $\varepsilon(M) \approx \mathcal{O}\left(M^{-r/N}\right)$

What about the *isotropic* sparse grid approach?

[Novak, Ritter, '95]

$$\varepsilon(M) = \|u - \mathscr{A}(w, N)u\|_\infty \approx \mathcal{O}\left(M^{-r} (\log M)^{(N-1)(r+1)}\right)$$

- "Exploits the smoothness r of the function (as opposed to MC, QMC, LHS, etc.)"
- "Improves upon the QMC convergence rate assuming $r > 1$"
- "Breaks the curse of dimensionality"

Complexity of C-C sparse grids:

Why you really use this stuff.....

Sandia National Laboratories
A Department of Energy National Laboratory

Recall: convergence of *isotropic* the FTP: $\varepsilon(M) \approx \mathcal{O}\left(M^{-r/N}\right)$

What about the *isotropic* sparse grid approach?

[Novak, Ritter, '95]

$$\varepsilon(M) = \|u - \mathscr{A}(w,N)u\|_\infty \approx \mathcal{O}\left(M^{-r}\left(\log M\right)^{(N-1)(r+1)}\right)$$

- "Exploits the smoothness r of the function (as opposed to MC, QMC, LHS, etc.)"
- "Improves upon the QMC convergence rate assuming $r > 1$"
- "Breaks the curse of dimensionality"

This is a Terrible Upper Bound and is simply a cute result: Does it really exploit smoothness? Where are the real improvements over MC, QMC, etc and is it even competitive with tensor products? What if the function is analytic?

A simple experiment

$N = 11$, $r = 10$ and $M = 1000$

$$\varepsilon_{SG} \approx \mathcal{O}\left(M^{-r} (\log M)^{(N-1)(r+1)} \right) \sim 10^{-30}\, 3^{(11)\,(10)} \sim 3 \times 10^{22}$$

$$\varepsilon_{TP} \approx \mathcal{O}\left(M^{-r/N} \right) \sim 2 \times 10^{-3}$$

$$\varepsilon_{MC} \approx \mathcal{O}\left(M^{-1/2} \right) \sim 3 \times 10^{-2}$$

$$\varepsilon_{QMC} \approx \mathcal{O}\left(M^{-1} (\log M)^{N} \right) \sim 2 \times 10^{2}$$

[Nobile, Tempone, Webster, '07]

$$\varepsilon_{SG}(M) = \|u - \mathscr{A}(w, N)u\|_{\infty} \approx \mathcal{O}\left(M^{-\sigma/(\log(2N))} \right)$$

- $\sigma = r$ when $u \in \mathcal{W}_r^{(N)}$ (bdd mixed derivatives of order r)
- σ is related to the distance to the nearest singularity of an analytically continued function.

A simple experiment

$N = 11$, $r = 10$ and $M = 1000$

$$\varepsilon_{SG} \approx \mathcal{O}\left(M^{-r}\,(\log M)^{(N-1)(r+1)}\right) \sim 10^{-30}\,3^{(11)\,(10)} \sim 3 \times 10^{22}$$

$$\varepsilon_{TP} \approx \mathcal{O}\left(M^{-r/N}\right) \sim 2 \times 10^{-3}$$
$$\varepsilon_{MC} \approx \mathcal{O}\left(M^{-1/2}\right) \sim 3 \times 10^{-2}$$
$$\varepsilon_{QMC} \approx \mathcal{O}\left(M^{-1}(\log M)^{N}\right) \sim 2 \times 10^{2}$$

[Nobile, Tempone, Webster, '07]

$$\varepsilon_{SG}(M) = \|u - \mathscr{A}(w, N)u\|_{\infty} \approx \mathcal{O}\left(M^{-\sigma/(\log(2N))}\right)$$

- $\sigma = r$ when $u \in \mathcal{W}_r^{(N)}$ (bdd mixed derivatives of order r)
- σ is related to the distance to the nearest singularity of an analytically continued function.

A simple experiment

$N = 11$, $r = 10$ and $M = 1000$

$$\varepsilon_{SG} \approx \mathcal{O}\left(M^{-r}\,(\log M)^{(N-1)(r+1)}\right) \sim 10^{-30}\,3^{(11)\,(10)} \sim 3 \times 10^{22}$$

$$\varepsilon_{TP} \approx \mathcal{O}\left(M^{-r/N}\right) \sim 2 \times 10^{-3}$$
$$\varepsilon_{MC} \approx \mathcal{O}\left(M^{-1/2}\right) \sim 3 \times 10^{-2}$$
$$\varepsilon_{QMC} \approx \mathcal{O}\left(M^{-1}(\log M)^{N}\right) \sim 2 \times 10^{2}$$

[Nobile, Tempone, Webster, '07]

$$\varepsilon_{SG}(M) = \|u - \mathscr{A}(w, N)u\|_{\infty} \approx \mathcal{O}\left(M^{-\sigma/(\log(2N))}\right)$$

- $\sigma = r$ when $u \in \mathcal{W}_r^{(N)}$ (bdd mixed derivatives of order r)
- σ is related to the distance to the nearest singularity of an analytically continued function.

A simple experiment

$N = 11$, $r = 10$ and $M = 1000$

$$\varepsilon_{SG} \approx \mathcal{O}\left(M^{-r}\,(\log M)^{(N-1)(r+1)}\right) \sim 10^{-30}\,3^{(11)\,(10)} \sim 3 \times 10^{22}$$

$$\varepsilon_{TP} \approx \mathcal{O}\left(M^{-r/N}\right) \sim 2 \times 10^{-3}$$

$$\varepsilon_{MC} \approx \mathcal{O}\left(M^{-1/2}\right) \sim 3 \times 10^{-2}$$

$$\varepsilon_{QMC} \approx \mathcal{O}\left(M^{-1}(\log M)^{N}\right) \sim 2 \times 10^{2}$$

[Nobile, Tempone, Webster, '07]

$$\varepsilon_{SG}(M) = \|u - \mathscr{A}(w, N)u\|_{\infty} \approx \mathcal{O}\left(M^{-\sigma/(\log(2N))}\right)$$

- $\sigma = r$ when $u \in \mathcal{W}_r^{(N)}$ (bdd mixed derivatives of order r)
- σ is related to the distance to the nearest singularity of an analytically continued function.

$$\varepsilon_{SG} \approx \mathcal{O}\left(M^{-r/(\log(2N))}\right) \sim 1 \times 10^{-10}$$

Dimension-Adaptive sparse grids

An anisotropic appraoch [Nobile, Tempone, Webster, '07]

Basic idea: Consider the general class of simplices $\mathbf{i} \cdot \boldsymbol{\alpha} \leq q$ where

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_N) \in \mathbb{R}_+^N \text{ with } \underline{\alpha} := \min_{1 \leq n \leq N} \alpha_n$$

is a N-dimensional weight vector for each stochastic direction.

For $w \in \mathbb{N}$, the **anisotropic** Smolyak algorithm is given by:

$$\mathscr{A}_{\boldsymbol{\alpha}}(w, N) = \sum_{\mathbf{i} \in X_{\boldsymbol{\alpha}}(w,N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N} \right)$$

$$X_{\boldsymbol{\alpha}}(w, N) = \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \, : \, \sum_{n=1}^{N} (i_n - 1)\alpha_n \leq w\underline{\alpha} \right\}.$$

Q: How does one construct $\boldsymbol{\alpha}$ for solving SPDEs?

A: We have an optimal strategy that relies on either *a priori* estimates or *a posteriori* approaches.

Dimension-Adaptive sparse grids

An anisotropic appraoch [Nobile, Tempone, Webster, '07]

Basic idea: Consider the general class of simplices $\mathbf{i} \cdot \boldsymbol{\alpha} \leq q$ where

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathbb{R}_+^N \text{ with } \underline{\alpha} := \min_{1 \leq n \leq N} \alpha_n$$

is a N-dimensional weight vector for each stochastic direction. For $w \in \mathbb{N}$, the **anisotropic** Smolyak algorithm is given by:

$$\mathscr{A}_{\boldsymbol{\alpha}}(w, N) = \sum_{\mathbf{i} \in X_{\boldsymbol{\alpha}}(w,N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right)$$

$$X_{\boldsymbol{\alpha}}(w, N) = \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \;:\; \sum_{n=1}^{N} (i_n - 1)\alpha_n \leq w\underline{\alpha} \right\}.$$

Q: How does one construct α for solving SPDEs?

A: We have an optimal strategy that relies on either *a priori* estimates or *a posteriori* approaches.

Dimension-Adaptive sparse grids

An anisotropic appraoch [Nobile, Tempone, Webster, '07]

Basic idea: Consider the general class of simplices $\mathbf{i} \cdot \boldsymbol{\alpha} \leq q$ where

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathbb{R}_+^N \text{ with } \underline{\alpha} := \min_{1 \leq n \leq N} \alpha_n$$

is a N-dimensional weight vector for each stochastic direction. For $w \in \mathbb{N}$, the **anisotropic** Smolyak algorithm is given by:

$$\mathscr{A}_{\boldsymbol{\alpha}}(w, N) = \sum_{\mathbf{i} \in X_{\boldsymbol{\alpha}}(w,N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N} \right)$$

$$X_{\boldsymbol{\alpha}}(w, N) = \left\{ \mathbf{i} \in \mathbb{N}_+^N, \mathbf{i} \geq \mathbf{1} \ : \sum_{n=1}^{N} (i_n - 1)\alpha_n \leq w\underline{\alpha} \right\}.$$

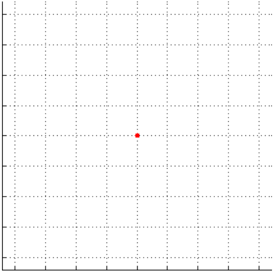
Q: How does one construct $\boldsymbol{\alpha}$ for solving SPDEs?

A: We have an optimal strategy that relies on either *a priori* estimates or *a posteriori* approaches.

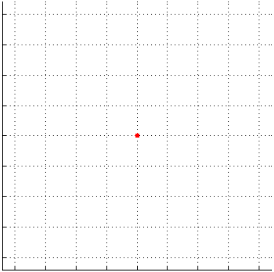
Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

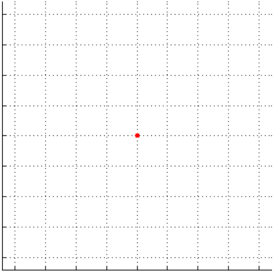
$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 0, 2)$



$\alpha_2/\alpha_1 = 1$



$\alpha_2/\alpha_1 = 1.5$



$\alpha_2/\alpha_1 = 2$

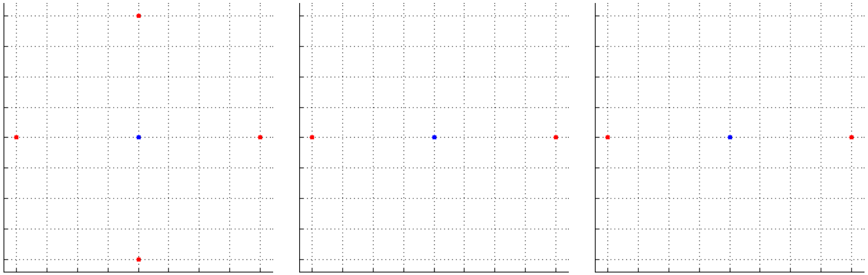
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 1, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

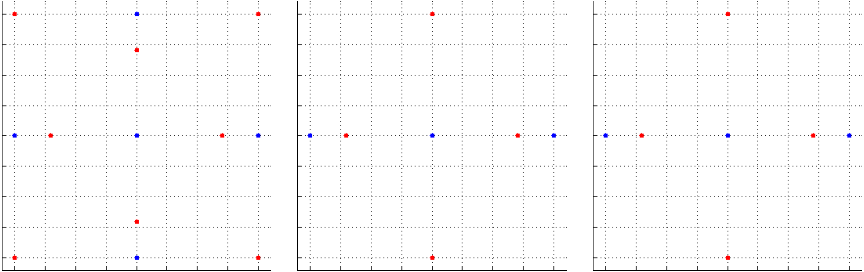
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 2, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

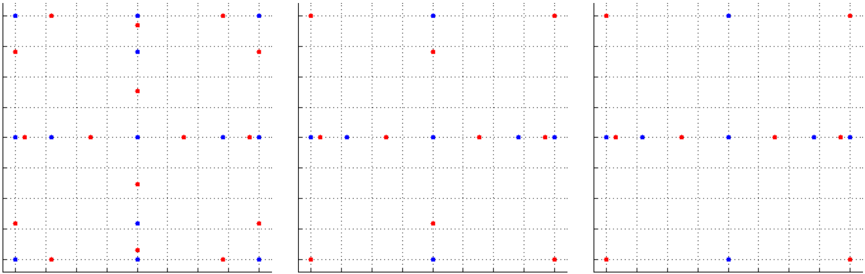
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 3, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

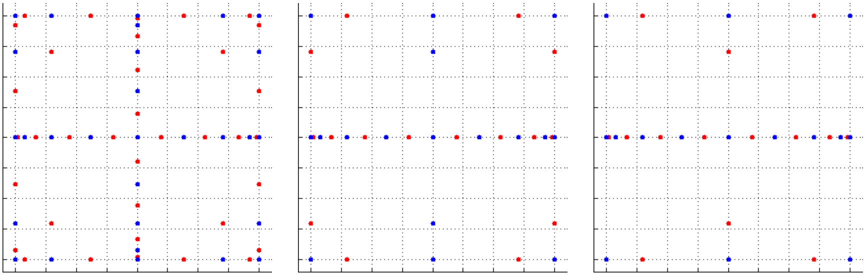
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\alpha}(w = 4, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

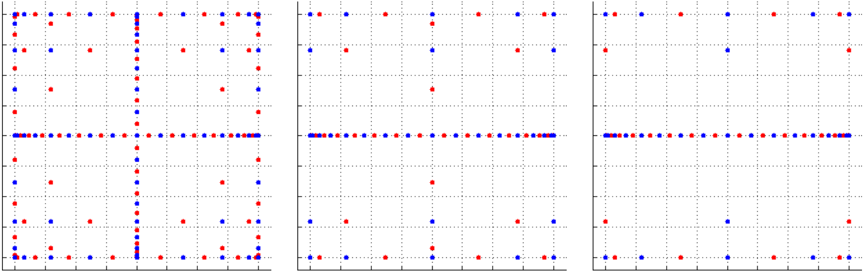
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\alpha}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\alpha} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\alpha} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 5, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

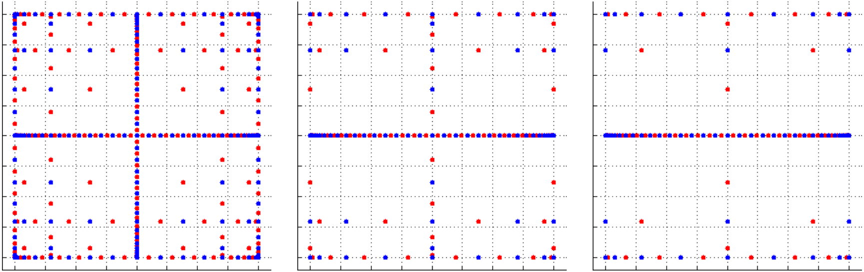
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 6, 2)$



$\alpha_2/\alpha_1 = 1$ $\alpha_2/\alpha_1 = 1.5$ $\alpha_2/\alpha_1 = 2$

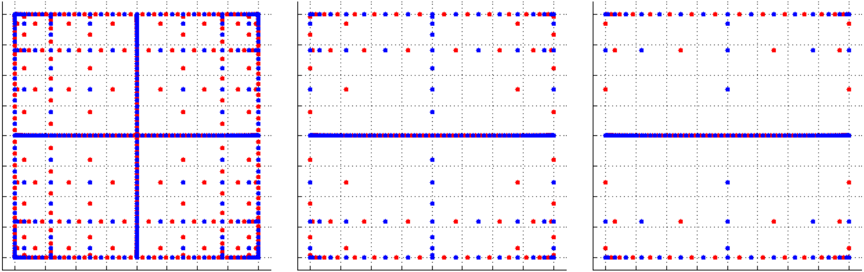
$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated anisotropic sparse grids

Clenshaw-Curtis abscissas, $N = 2$

$N = 2$ anisotropic sparse grid: $\mathscr{H}_{\boldsymbol{\alpha}}(w = 7, 2)$



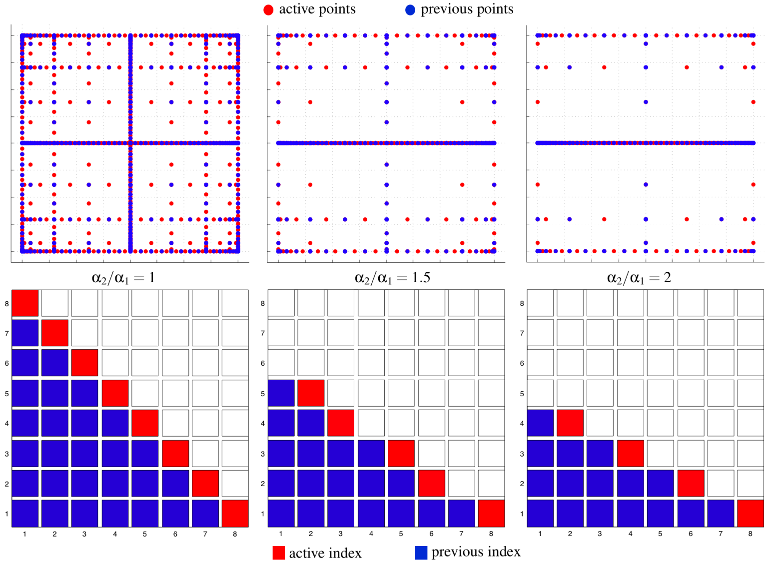
$\alpha_2/\alpha_1 = 1$ $\quad\quad$ $\alpha_2/\alpha_1 = 1.5$ $\quad\quad$ $\alpha_2/\alpha_1 = 2$

$$\varepsilon_{ASG} \approx \mathcal{O}\left(M^{-c_{\boldsymbol{\alpha}}\sigma/\mathscr{G}(N)}\right) \quad \text{(Approx. behavior)}$$

For highly isotropic problems $\mathscr{G}(N) \approx \log(2N)$ and $c_{\boldsymbol{\alpha}} \approx 1$. For highly anisotropic problems $\mathscr{G}(N) < \log(2N)$ and $c_{\boldsymbol{\alpha}} >> 1$.

Generated C-C anisotropic sparse grids

Correspondng indices $(i_1, i_2) \in X_\alpha(7,2)$



Examples

Using Stochastic Collocation FEM (SCFEM)

- decouples computations as Monte Carlo does by using a Lagrange interpolating basis for the polynomial approximation to the probability space (as opposed to Polynomial Chaos)
- treats efficiently the case of non independent random variables introducing an auxiliary density

$$\hat{\rho}(y) = \prod_{n=1}^{N} \hat{\rho}_n(y_n), \quad \forall y \in \Gamma^N, \qquad \text{and s.t.} \quad \left\| \frac{\rho}{\hat{\rho}} \right\|_{L^\infty(\Gamma^N)} < \infty,$$

- deals with unbounded random variables in the input data,
- essentially preserves the convergence speed to the stochastic Galerkin FEM.
- effectively handle problems that depend on random input data described by a moderately large number of random variables with the use of sparse collocation tensor product techniques

Numerical examples

The problems we consider are:

$$\begin{cases} -\nabla \cdot (a_N(\omega,\cdot)\nabla u_N(\omega,\cdot)) &= f_N(\omega,\cdot) \quad \text{in } \Omega \times D \\ u_N(\omega,\cdot) &= 0 \quad \text{on } \Omega \times \partial D \end{cases}$$

$$\begin{cases} -\nabla \cdot (a_N(\omega,\cdot)\nabla u_N(\omega,\cdot)) + (u_N(\omega,\cdot))^2 &= f_N(\omega,\cdot) \quad \text{in } \Omega \times D \\ u_N(\omega,\cdot) &= 0 \quad \text{on } \Omega \times \partial D \end{cases}$$

with $D = \left\{\mathbf{x} = (x,z) \in \mathbb{R}^2 \,:\, 0 \le x, z \le 1\right\}$ and a deterministic load $f_N(\omega, x, z) = \cos(x)\sin(z)$

$$\log(a_N(\omega,x) - 0.5) = 1 + Y_1(\omega)\left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{n=2}^{N} \beta_n\, \varphi_n(x)\, Y_n(\omega).$$

$$\beta_n := \left(\sqrt{\pi}L\right)^{1/2} \exp\left(\frac{-\left(\lfloor \frac{n}{2} \rfloor \pi L\right)^2}{8}\right), \qquad \text{if } n > 1$$

$$\varphi_n(x) := \begin{cases} \sin\left(\lfloor \frac{n}{2} \rfloor \pi x\right), & \text{if } n \text{ even,} \\ \cos\left(\lfloor \frac{n}{2} \rfloor \pi x\right), & \text{if } n \text{ odd} \end{cases}.$$

Numerical examples

The problems we consider are:

$$\begin{cases} -\nabla \cdot (a_N(\omega,\cdot)\nabla u_N(\omega,\cdot)) = f_N(\omega,\cdot) & \text{in } \Omega \times D \\ u_N(\omega,\cdot) = 0 & \text{on } \Omega \times \partial D \end{cases}$$

$$\begin{cases} -\nabla \cdot (a_N(\omega,\cdot)\nabla u_N(\omega,\cdot)) + (u_N(\omega,\cdot))^2 = f_N(\omega,\cdot) & \text{in } \Omega \times D \\ u_N(\omega,\cdot) = 0 & \text{on } \Omega \times \partial D \end{cases}$$

with $D = \left\{\mathbf{x} = (x,z) \in \mathbb{R}^2 \, : \, 0 \le x, z \le 1\right\}$ and a deterministic load $f_N(\omega,x,z) = \cos(x)\sin(z)$

$$\log(a_N(\omega,x) - 0.5) = 1 + Y_1(\omega)\left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{n=2}^{N} \beta_n \, \varphi_n(x)\, Y_n(\omega).$$

$$\beta_n := \left(\sqrt{\pi}L\right)^{1/2} \exp\left(\frac{-\left(\lfloor \frac{n}{2} \rfloor \pi L\right)^2}{8}\right), \quad \text{if } n > 1$$

$$\varphi_n(x) := \begin{cases} \sin\left(\lfloor \frac{n}{2} \rfloor \pi x\right), & \text{if } n \text{ even}, \\ \cos\left(\lfloor \frac{n}{2} \rfloor \pi x\right), & \text{if } n \text{ odd} \end{cases}.$$

Computational goals

Expected value

- $\mathbb{E}[Y_n] = 0$ and $\mathbb{E}[Y_n Y_m] = \delta_{nm}$ for $n, m \in \mathbb{N}_+$, and are taken uniform in the interval $[-\sqrt{3}, \sqrt{3}]$.
- To resolve the noise $W_h(D) \equiv$ span of continuous piecewise quadratic functions over a uniform triangulation of D with maximum mesh size $h = 1/3N$. (Smallest Period $T = 4/N$) i.e. for $N = 11$ we have 4225 FE unknowns.

GOAL: Compute the an approximation to the expected value (extensions to higher moments, probabilities, etc. straight-forward)

$$\mathbb{E}[u_N](x) \approx \int_{\Gamma^N} \mathcal{I}^{(N)}[u_N^h(y,\cdot)](x)\rho(y)dy, \quad y \in \Gamma^N, x \in \overline{D}$$

Calculating the weighting parameters

A priori: $N = 11$ (Linear SPDE)

	$g(1)$	$g(2,3)$	$g(4,5)$	$g(6,7)$	$g(8,9)$	$g(10,11)$
$L = 1/2$	0.20	0.19	0.42	1.24	3.1	5.8
$L = 1/64$	0.79	0.62	0.62	0.62	0.62	0.62

Table: The $N = 11$ values of the function $g(n)$ constructed from *a priori* information. The components of $\boldsymbol{\alpha} = \mathrm{g}$ are used as the input information for the anisotropic Smolyak algorithm with correlation lengths $L = 1/2$ and $L = 1/64$.

To study the convergence of the anisotropic algorithm we consider a fixed dimension N and estimate the error:

$$\|\mathbb{E}[\epsilon]\|_{L^2(D)} \approx \|\mathbb{E}[\mathscr{A}_{\boldsymbol{\alpha}}(w, N)\pi_h u_N - \mathscr{A}_{\widehat{\boldsymbol{\alpha}}}(\overline{w}+1, N)\pi_h u_N]\|_{L^2(D)}$$

- $w = 0, 1, \ldots, \overline{w}$ and $\mathscr{A}_{\widehat{\boldsymbol{\alpha}}}(\overline{w}+1, N)\pi_h u_N$ is an enriched solution that is an approximation to the exact solution.

Calculating the weighting parameters

A priori: $N = 11$ (Linear SPDE)

	$g(1)$	$g(2,3)$	$g(4,5)$	$g(6,7)$	$g(8,9)$	$g(10,11)$
$L = 1/2$	0.20	0.19	0.42	1.24	3.1	5.8
$L = 1/64$	0.79	0.62	0.62	0.62	0.62	0.62

Table: The $N = 11$ values of the function $g(n)$ constructed from *a priori* information. The components of $\boldsymbol{\alpha} = \mathrm{g}$ are used as the input information for the anisotropic Smolyak algorithm with correlation lengths $L = 1/2$ and $L = 1/64$.

To study the convergence of the anisotropic algorithm we consider a fixed dimension N and estimate the error:

$$\|\mathbb{E}[\epsilon]\|_{L^2(D)} \approx \|\mathbb{E}[\mathscr{A}_{\boldsymbol{\alpha}}(w, N)\pi_h u_N - \mathscr{A}_{\widehat{\boldsymbol{\alpha}}}(\overline{w}+1, N)\pi_h u_N]\|_{L^2(D)}$$

- $w = 0, 1, \ldots, \overline{w}$ and $\mathscr{A}_{\widehat{\boldsymbol{\alpha}}}(\overline{w}+1, N)\pi_h u_N$ is an enriched solution that is an approximation to the exact solution.

Calculating the weighting parameters

A posteriori selection: $N = 11$ (Nonlinear SPDE)

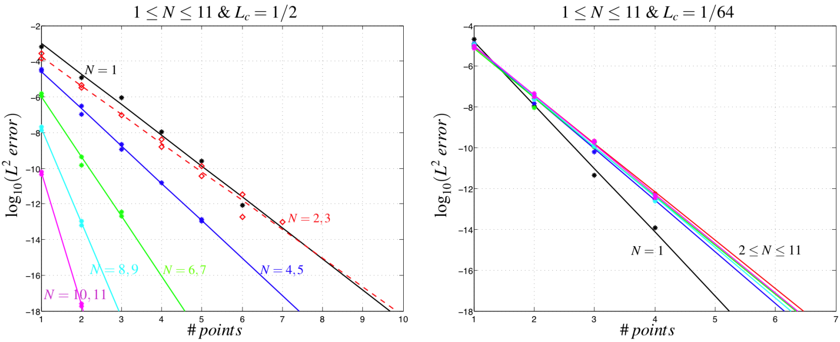
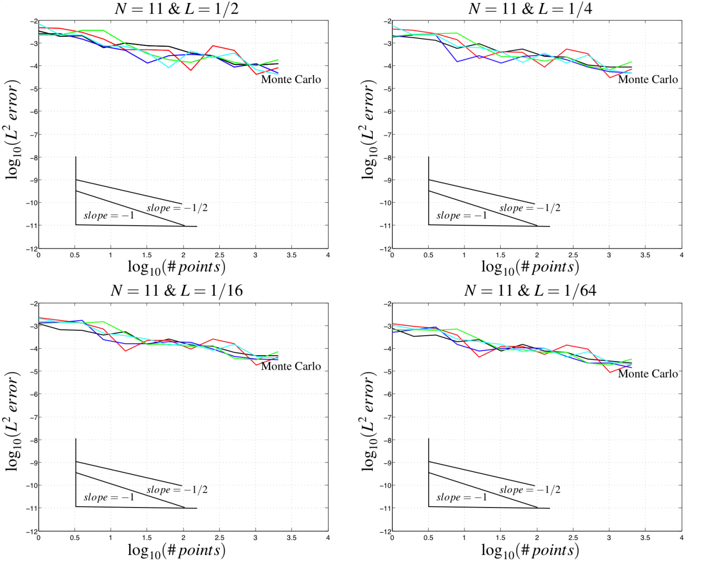


Figure: A linear least square approximation to fit $\log_{10}(\|E[\varepsilon_n]\|_{L^2(D)})$ versus p_n. For $n = 1, 2, \ldots, N = 11$ we plot: on the left, the highly anisotropic case $L_c = 1/2$ and on the right, the isotropic case $L_c = 1/64$.

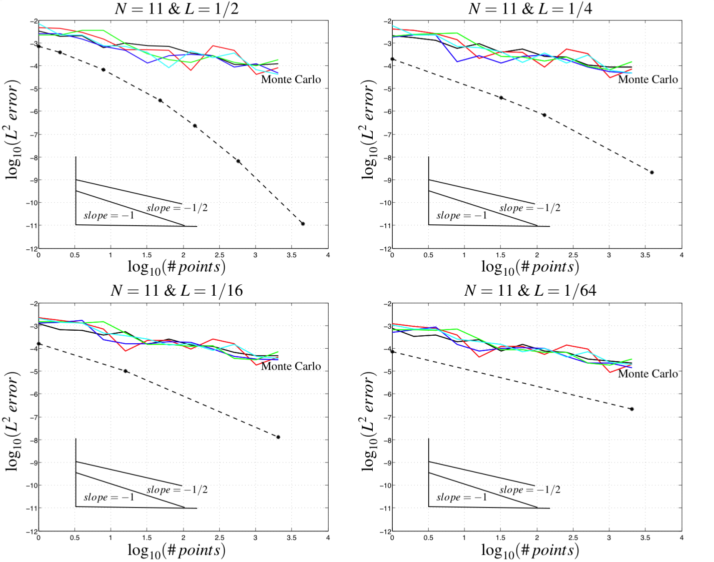
Convergence Comparisons I

$N = 11$ random variables



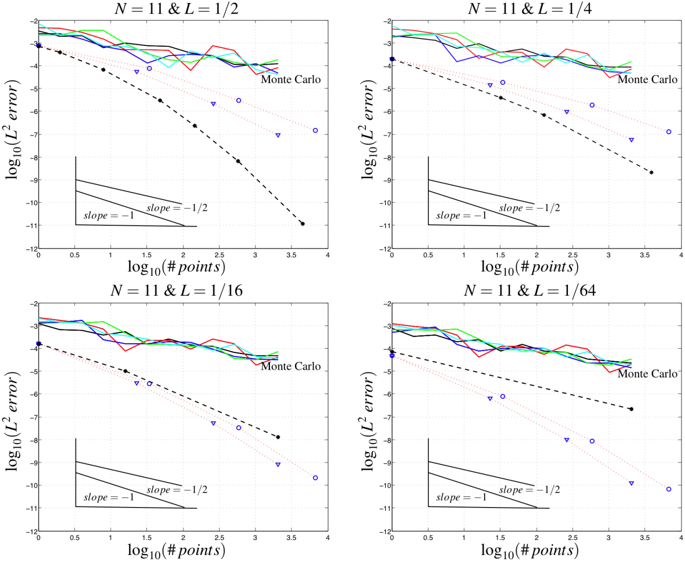
Convergence Comparisons I

$N = 11$ random variables



Convergence Comparisons I

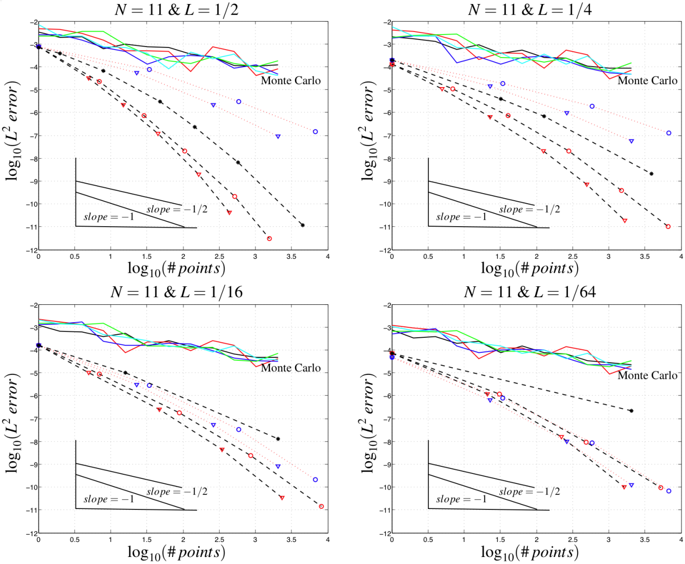
$N = 11$ random variables



Sandia National Laboratories
A Department of Energy National Laboratory

Convergence Comparisons I

$N = 11$ random variables



Convergence Comparisons II

$N = 11$ random variables

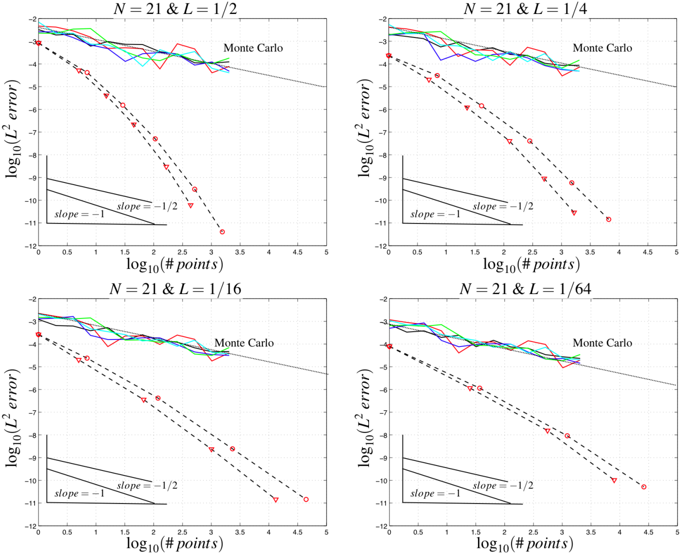
Sandia National Laboratories
A Department of Energy National Laboratory

L	AS	AF	IS	MC
$1/2$	50	252	2512	$5.0e+09$
$1/4$	158	1259	3981	$2.0e+09$
$1/16$	199	1958	501	$1.6e+09$
$1/64$	316	199530	360	$1.3e+09$

Table: For Γ^N, with $N = 11$, we compare the number of deterministic solutions required by the Anisotropic Smolyak (AS) using Clenshaw-Curtis abscissas, Anisotropic Full Tensor product method (AF) using Gaussian abscissas, Isotropic Smolyak (IS) using Clenshaw-Curtis abscissas and the Monte Carlo (MC) method using random abscissas, to reduce the original error by a factor of 10^4.

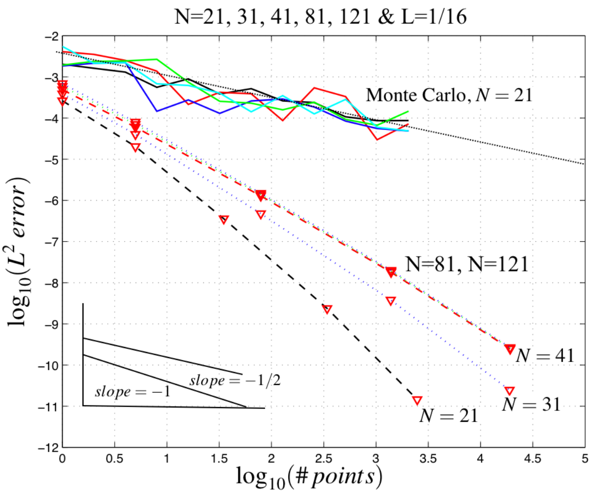
Convergence Comparisons III

$N = 21$ random variables



Convergence Comparisons IV: Nonlinear

$N = 21, 31, 41, 81, 121$ random variables (*A posteriori approach*)



Sandia National Laboratories
A Department of Energy National Laboratory

Concluding remarks

DAKOTA 4.1 - August 31st (including Sparse Grids)

- High-dimensional integration is a characteristic of modern uncertainty quantification. Accurate Monte Carlo-type results take too long and tensor product methods suffer from the *curse of dimensionality*.
- We have developed and analyzed several novel (non-instrusive) numerical methods for solving SPDEs. In particular, the *Sparse grid Stochastic Collocation FEM* yields:
  - uncoupled deterministic problems (fully parallelizable)
  - reduces considerably the *curse of dimensionality.*
  - The analysis reveals that if u has an analytic extension w.r.t. the noise $\Longrightarrow$ at least an algebraic convergence of the "probability error" w.r.t. the total number of collocation (sample) points
- Non-smooth solutions: Anisotropic detection combined with adaptivity within each dimension (RV) is in progress.

Concluding remarks

DAKOTA 4.1 - August 31st (including Sparse Grids)

Sandia National Laboratories
A Department of Energy National Laboratory

- High-dimensional integration is a characteristic of modern uncertainty quantification. Accurate Monte Carlo-type results take too long and tensor product methods suffer from the *curse of dimensionality*.
- We have developed and analyzed several novel (non-instrusive) numerical methods for solving SPDEs. In particular, the *Sparse grid Stochastic Collocation FEM* yields:
  - uncoupled deterministic problems (fully parallelizable)
  - reduces considerably the *curse of dimensionality.*
  - The analysis reveals that if u has an analytic extension w.r.t. the noise $\Longrightarrow$ at least an algebraic convergence of the "probability error" w.r.t. the total number of collocation (sample) points
- Non-smooth solutions: Anisotropic detection combined with adaptivity within each dimension (RV) is in progress.

Concluding remarks

DAKOTA 4.1 - August 31st (including Sparse Grids)

- High-dimensional integration is a characteristic of modern uncertainty quantification. Accurate Monte Carlo-type results take too long and tensor product methods suffer from the *curse of dimensionality*.
- We have developed and analyzed several novel (non-instrusive) numerical methods for solving SPDEs. In particular, the *Sparse grid Stochastic Collocation FEM* yields:
  - uncoupled deterministic problems (fully parallelizable)
  - reduces considerably the *curse of dimensionality.*
  - The analysis reveals that if u has an analytic extension w.r.t. the noise $\Longrightarrow$ at least an algebraic convergence of the "probability error" w.r.t. the total number of collocation (sample) points
- Non-smooth solutions: Anisotropic detection combined with adaptivity within each dimension (RV) is in progress.

References

- F. Nobile, R. Tempone and **C. Webster**
  An anisotropic sparse grid stochastic collocation method for PDEs with random input data, preprint, submitted SIAM J. Numer. Anal., Jan. 2007.
- I. Babuška, F. Nobile, R. Tempone and **C. Webster**
  Stochastic collocation methods for nonlinear PDEs with random input data, in progress.
- F. Nobile, R. Tempone and **C. Webster**
  A sparse grid stochastic collocation method for PDEs with random input data, preprint, submitted SIAM J. Numer. Anal., May 2006.
- I. Babuška, F. Nobile and R. Tempone.
  A stochastic collocation method for elliptic PDEs with random input data, SIAM J. Numer. Anal., 2006.