Uncertainty Quantification using deterministic solvers and collocation based approaches in high stochastic dimensions

Prof. Nicholas Zabaras
Materials Process Design and Control Laboratory
Sibley School of Mechanical and Aerospace Engineering
101 Frank H. T. Rhodes Hall
Cornell University
Ithaca, NY 14853-3801

Email: zabaras@cornell.edu
URL: http://mpdc.mae.cornell.edu/
Tel: (607) 255 9104

OA Statistical Forum 2010, Eglin, November 8th, 2010
Outline of the presentation

- Stochastic analysis
- Issues with gPC and ME-gPC
- Collocation based approaches
- Basics of stochastic collocation
- Issues with existing stochastic sparse grid collocation method (Smolyak algorithm)
- Adaptive sparse grid collocation method
- Some numerical examples
Thermal transport through polycrystalline and functionally graded materials (~ 2.5 million DOF)

Analysis of physical phenomena in random media

Multiple sources of uncertainty involved: initial conditions, property variations (topology), boundary conditions, noise, fluctuations, etc.

Realistic analysis must consider these aspects
Problem definition

- Define a complete probability space \((\Omega, F, P)\). We are interested to find a stochastic function \(u: \Omega \times D \rightarrow \mathbb{R}\) such that for P-almost everywhere (a.e.) \(\omega \in \Omega\), the following equation hold:

\[
L(x, \omega; u) = f(x, \omega), \quad \forall x \in D
\]

\[
B(x, u) = g(x), \quad \forall x \in \partial D
\]

where \(x = (x_1, x_2, \ldots, x_d)\) are the coordinates in \(\mathbb{R}^d\), \(L\) is (linear/nonlinear) differential operator, and \(B\) is a boundary operators.

- In the most general case, the operator \(L\) and \(B\) as well as the driving terms \(f\) and \(g\), can be assumed random.

- In general, we require an infinite number of random variables to completely characterize a stochastic process. This poses a numerical challenge in modeling uncertainty in physical quantities that have spatio-temporal variations, hence necessitating the need for a reduced-order representation.

- By using Karhunen-Loève expansion, the random input can be characterized by a set of random variables.
1. Interpreting random variables

2. Distribution of the random variable
   
   Ex. inlet velocity, inlet temperature

\[ \theta = \theta_0 \left(1 + 0.1\xi\right) \]

3. Correlated data
   
   Ex. Presence of impurities, porosity
   
   Usually represented with a correlation function
   
   We specifically concentrate on this.

4. High dimensionality arises from random fields!
Representing Randomness 2

1. Representation of random process
   - Karhunen-Loève, Polynomial Chaos expansions

2. Infinite dimensions to finite dimensions
   - depends on the covariance

Karhunen-Loève expansion
Based on the spectral decomposition of the covariance kernel of the stochastic process

\[ w(x, t, \theta) = \overline{w}(x, t) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i(\theta) \phi_i(x, t) \]

- Need to know covariance
- Converges uniformly to any second order process

Set the number of stochastic dimensions, N

Dependence of variables

\[ \theta(\omega, x) = \theta(Y^1(\omega), ..., Y^N(\omega), x) \]

Pose the (N+d) dimensional problem
Karhunen-Loéve expansion

\[ X(x, t, \omega) = \bar{X}(x, t) + \sum_{i=1}^{\infty} X_i(x, t)Y_i(\omega) \]

\[ \rightarrow \text{Stochastic process} \quad \downarrow \text{Mean function} \]

\[ \rightarrow \text{ON random variables} \quad \downarrow \text{Deterministic functions} \]

- Deterministic functions \( \sim \) eigen-values, eigenvectors of the covariance function
- Orthonormal random variables \( \sim \) type of stochastic process
- In practice, we truncate (KL) to first \( N \) terms

\[ X(x, t, \omega) = \text{fn}(x, t, Y_1, \ldots, Y_N) \]
The finite-dimensional noise assumption

- By using the Doob-Dynkin lemma, the solution of the problem can be described by the same set of random variables, i.e.
  \[ u(x, \omega) = u(x, Y_1(\omega), \ldots, Y_N(\omega)) \]

- So the original problem can be restated as: Find the stochastic function \( u : \Gamma \times D \to \mathbb{R} \) such that
  \[
  L(x, Y; u) = f(x, Y), \quad \forall x \in D \times \Gamma \\
  B(x, Y; u) = g(x, Y), \quad \forall x \in \partial D \times \Gamma
  \]

- In this work, we assume that \( \{Y_i(\omega)\}_{i=1}^{N} \) are independent random variables with probability density function \( \rho_i \). Let \( \Gamma_i \) be the image of \( Y_i \). Then
  \[
  \rho(Y) = \prod_{i=1}^{N} \rho_i(Y_i), \quad \forall Y \in \Gamma
  \]
  is the joint probability density of \( Y = (Y_1, \ldots, Y_N) \) with support
  \[
  \Gamma \equiv \prod_{i=1}^{N} \Gamma_i \in \mathbb{R}^N
  \]
Uncertainty analysis techniques

- Monte-Carlo: Simple to implement, computationally expensive
- Perturbation, Neumann expansions: Limited to small fluctuations, tedious for higher order statistics
- Sensitivity analysis, method of moments: Probabilistic information is indirect, small fluctuations
- Spectral stochastic uncertainty representation: Basis in probability and functional analysis, Can address second order stochastic processes, Can handle large fluctuations, derivations are general
- Stochastic collocation: Results in decoupled equations
A stochastic process = spatially, temporally varying random function

\[ X : (x, t, \Omega) \rightarrow \mathbb{R} \]
Generalized polynomial chaos (gPC)

Generalized polynomial chaos expansion is used to represent the stochastic output in terms of the input 

\[ X(x, t, \omega) = f_n(x, t, Y_1, \ldots, Y_N) \]

Askey polynomials ~ type of input stochastic process

Usually, Hermite, Legendre, Jacobi etc.
Disadvantages:

1) Coupled nature of the formulation makes implementation nontrivial$^1$.

2) Substantial effort into coding the stochastics: Intrusive

3) As the order of expansion ($p$) is increased, the number of unknowns ($P$) increases factorially with the stochastic dimension ($N$): combinatorial explosion$^{2,3}$.

$$P + 1 = \frac{(N + p)!}{N! P!}$$

4) Furthermore, the *curse of dimensionality* makes this approach particularly difficult for large stochastic dimensions


Failure of gPC

- Gibbs phenomenon: Due to its global support, it fails to capture stochastic discontinuities in the stochastic space.

- Therefore, we need some method which can resolve discontinuities/non-smooth conditions locally. This has motivated the so called multi-element gPC method. However, the truth is that these methods don’t work in more than a few random dimensions!
**Stochastic Collocation based framework**

A stochastic function in 2 dimensions is described as:

$$f : \mathbb{R}^N \rightarrow \mathbb{R}$$

Stochastic function in 2 dimensions is sampled at a finite set of points:

$$\Theta_M = \{Y^{(i)}\}_{i=1}^M$$

Use basis function to get an approximate representation:

$$\tilde{f}(Y) = \sum_{i=1}^M f(Y^{(i)})a_i(Y^{(i)})$$

Function value at any point is simply:

$$\tilde{f}(\xi)$$

Spatial domain is approximated using a FE, FD, or FV discretization.

Stochastic domain is approximated using multidimensional interpolating functions.
From one-dimension to higher dimensions

- Denote the one dimensional interpolation formula as
  \[ U^i(f) = \sum_{j=1}^{m_i} f(Y^i_j) \cdot a_j^i \]
  with the set of support nodes: \( X^i = \{ Y^i_j \mid Y^i_j \in [0, 1] \text{ for } j = 1, 2, \ldots, m_i \} \)

- In higher dimension, a simple case is the tensor product formula
  \[ (U^{i_1} \otimes \cdots \otimes U^{i_N})(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_N=1}^{m_{i_N}} f(Y^{i_1}_{j_1}, \ldots, Y^{i_N}_{j_N}) \cdot (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_N}^{i_N}) \]

For instance, if \( M=10 \) dimensions and we use \( k \) points in each direction

<table>
<thead>
<tr>
<th>Number of points in each direction, ( k )</th>
<th>Total number of sampling points</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1024</td>
</tr>
<tr>
<td>3</td>
<td>59049</td>
</tr>
<tr>
<td>4</td>
<td>1.05x10^6</td>
</tr>
<tr>
<td>5</td>
<td>9.76x10^6</td>
</tr>
<tr>
<td>10</td>
<td>1x10^{10}</td>
</tr>
</tbody>
</table>

This quickly becomes impossible to use.

One idea is only to pick the more important points from the tensor product grid.
Extensively used in statistical mechanics

Provides a way to construct interpolation functions based on minimal number of points

Provides a way to progress naturally from univariate interpolation to multivariate interpolation.

First construct one dimensional interpolation functions

Construct the required multi-dimensional as a linear combination of products of these 1D functions

How to choose the degree and coefficients of these products?

Smolyak (1963) came up with a set of rules to construct such products\(^1\).

---

Conventional sparse grid collocation (CSGC)

In higher dimensions, not all the points are equally important.

Some points contribute less to the accuracy of the solution (e.g. points where the function is very smooth, regions where the function is linear, etc.). Discard the points that contribute less: **SPARSE GRID COLLOCATION**
Conventional sparse grid collocation

- Both piecewise linear as well as polynomial functions, L, can be used to construct the full tensor product formula above.
- The Smolyak construction starts from this full tensor product representation and discards some multi-indices to construct the minimal representation of the function.
- Define the differential interpolant, \( \Delta \) as the difference between two interpolating approximations
  \[
  \Delta^i = U^i - U^{i-1}, \quad \text{with} \quad U^0 = 0
  \]
- The sparse grid interpolation of the function \( f \) is given by
  \[
  A_{q,N}(f) = \sum_{||i|| \leq q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N})(f)
  \]

where \( ||i|| = i_1 + \cdots + i_d \) is the multi-index representation of the support nodes. \( N \) is the dimension of the function \( f \) and \( q \) is an integer (\( q > N \)). \( k = q - N \) is called the depth of the interpolation. As \( q \) is increased more and more points are sampled.
Conventional **sparse grid collocation**

- The interpolant can be represented in a recursive manner as
  \[
  A_{q,N}(f) = \sum_{|i| \leq q} (\Delta^{i_1} \otimes \cdots \Delta^{i_N})(f) = A_{q-1,N}(f) + \sum_{|i|=q} (\Delta^{i_1} \otimes \cdots \Delta^{i_N})(f) \quad \text{with} \quad A_{N-1,N} = 0
  \]

- Can increase the accuracy of the interpolant based on Smolyak’s construction **WITHOUT** having to discard previous results.

- Can build on previous interpolants due to the recursive formulation.

- To compute the interpolant \( A_{q,N} \), one only needs the function values at the sparse grid point given by:
  \[
  H_{q,N} = \bigcup_{q-N+1 \leq |i| \leq q} (X^{i_1} \times \cdots \times X^{i_N})
  \]

  One could select the sets \( X^i \), in a nested fashion such that \( X^i \subset X^{i+1} \).

- Similar to the interpolant representation in a recursive form, the sparse grid points can be represented as:
  \[
  H_{q,M} = H_{q-1,M} \cup \Delta H_{q,M}
  \]
  with \( \Delta H_{q,M} = \bigcup_{|i|=q} (X^{i_1}_\Delta \times \cdots \times X^{i_M}_\Delta) \) and \( X^i_\Delta = X^i \setminus X^{i-1} \).
Choice of collocation points and nodal basis functions

- One of the choice is the Clenshaw-Curtis grid at non-equidistant extrema of the Chebyshev polynomials. (D. Xiu etc, 2005, F. Nobile etc, 2006)

\[ m_i = \begin{cases} 
1, & \text{if } i = 1 \\
2^{i-1} + 1, & \text{if } i > 1 
\end{cases} \]

\[ y_{ij}^i = \begin{cases} 
(-\cos(\pi(j - 1)/(m_i - 1)) + 1)/2, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1 \\
0.5, & \text{for } j = 1, \text{ if } m_i = 1.
\end{cases} \]

- By doing so, the resulting sets are nested. The corresponding univariate nodal basis functions are Lagrangian characteristic polynomials:

\[ a_j^i = \begin{cases} 
1, & \text{for } i = 1, \text{ and} \\
\prod_{k=1}^{m_i} \frac{Y - y_{jk}^i}{y_{ji}^i - y_{jk}^i}, & \text{for } i > 1 \text{ and } j = 1, \ldots, m_i.
\end{cases} \]
Choice of collocation points and nodal basis functions

- The advantage of Clenshaw-Curtis grid:
  1. It is a polynomial interpolation, so it has fast convergence rate.
  2. The nodes are nested so it will result in very few collocation points than that of non-nested grid.

- However, there are also two problems with Clenshaw-Curtis grid:
  1. The basis function is of global support, so it will fail when there is discontinuity in the stochastic space.
  2. The nodes are pre-determined, so it is not suitable for implementation of local adaptivity. However, it is suitable for implementation of dimensional adaptivity.

- Therefore, we proposed to equal-distant node with piecewise linear interpolation to resolve the discontinuity in the stochastic space.
Choice of collocation points and nodal basis functions

In the context of incorporating adaptivity, we have used the Newton-Cotes grid using equidistant support nodes and use the linear hat function as the univariate nodal basis.

\[ m_i = \begin{cases} 
1, & \text{if } i = 1 \\
2^{i-1} + 1, & \text{if } i > 1 
\end{cases} \]

\[ Y^i_j = \begin{cases} 
\frac{j - 1}{m_i - 1}, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1, \\
0.5, & \text{for } j = 1, \text{ if } m_i = 1.
\]

Furthermore, by using the linear hat function as the univariate nodal basis function, one ensures a local support in contrast to the global support of Lagrange polynomials. This ensures that discontinuities in the stochastic space can be resolved. The piecewise linear basis functions can be defined as

\[ a^i_1 = 1 \quad \text{for } i = 1, \text{ and} \]

\[ a^i_j = \begin{cases} 
1 - (m_i - 1) \cdot |Y - Y^i_j|, & \text{if } |Y - Y^i_j| < 1/(m_i - 1), \\
0, & \text{otherwise,} 
\end{cases} \]

for \( i > 1 \) and \( j = 1, \ldots, m_i \).
Choice of collocation points and nodal basis functions

- The advantage of Newton-Cotes grid:
  1. It is a piecewise linear interpolation of local support, so it can resolve the stochastic discontinuity.
  2. The nodes are also nested so it will result in very few collocation points than that of non-nested grid.
  3. It is suitable for both implementations of local and dimensional adaptivity.

- However, the significant drawback of this method is its slow convergence rate due to its linear interpolation. It may need much more points to achieve the same error as that of Clenshaw-Curtis grid.

- Nevertheless, both Clenshaw-Curtis and Newton-Cotes are nested grids, which enables us to reformulate the original Smolyak algorithm into a hierarchical formulation.
Comparison of the two grids

Clenshaw-Curtis grid

Newton-Cotes grid
Sparse Grids

$H_{5,2}$
Number of points = 29
Number of points full grid = 81

$H_{7,2}$
Number of points = 145
Number of points full grid = 1089

$H_{6,3}$
Number of points = 69
Number of points full grid = 729

$H_{8,3}$
Number of points = 441
Number of points full grid = 35937
The number of sampling points in M dimensions is given by the cardinality of the set:

\[ H_{q,M} = \bigcup_{q-M+1 \leq i \leq q} (X^{i_1} \times \cdots \times X^{i_M}) \]

The plot shows the number of sampling points as a function of the depth of interpolation for \( M = 2, 3, 5 \).

It is straightforward to come up with bounds on \( N^{1,2} \).

\[ 2^{q-M+1} \leq N \leq \frac{2^q q^M}{(M-1)!} \]


Error estimate

Depending on the order of the one-dimensional interpolant, one can come up with error bounds on the approximating quality of the interpolant\(^1,2\).

If piecewise linear 1D interpolating functions are used to construct the sparse interpolant, the error estimate is given by:

\[
\|f - A_{q,N}(f)\|_\infty = O(M^{-2} |\log_2 M|^{3(N-1)})
\]

where \(N\) is the number of sampling points,

If 1D polynomial interpolation functions are used, the error bound is:

\[
\|f - A_{q,N}(f)\|_\infty = O(M^{-k} |\log_2 M|^{(k+2)(N+1)+1})
\]

assuming that \(f\) has \(k\) continuous derivatives

---

Numerical illustration of the sparse grid algorithm.

Approximate a smooth anisotropic function $f$ in $[-1,1]^2$.

Investigate interpolation accuracy as $\rho$ increases. As $\rho$ increases the function becomes steeper in one direction.

Error defined as deviation of interpolant from actual function

$$e = \max(| f - A_{q,2} |)$$
Numerical illustration: 1

Figure plots the error vs number of sampling points for increasing anisotropy\(^1\).

For \( \rho = 1000 \), the function is very anisotropic. The sparse collocation method uses 3329 points to construct an approximate solution with an error of \( 3 \times 10^{-2} \).

The full tensor product method uses 263169 points to get the same level of accuracy.

---

**Numerical illustration: 2**

\[ f(x, y) = e^{-x^2 + \text{sign}(y)} \]

Numerical illustration of the sparse grid algorithm.

Approximate a discontinuous function \( f \) in \([-1,1]^2\).

Function has a discontinuity in the \( y \) direction on the \( y = 0 \) line.

Investigate how the sparse grid algorithm approximates this function.
**Numerical illustration: 2**

**Error vs number of sampling points**

![Graph](image)

**Issues:**

Notice that the Smolyak method uniformly samples both dimensions.

Can the number of sampling points be further reduced by choosing points adaptively in different directions based on the behavior of the function?

Can this be done on-the-fly? **ADAPTIVITY**
Motivation of HDMR

- Conventional and adaptive collocation methods are not suitable for high-dimensional problems due to their weakly dependence on the dimensionality (logarithmic) in the error estimate.

- Although ASGC can alleviate this problem to some extent, its performance depends on the regularity of the problem and the method is only effective when some random dimensions are more important than others.

- In e.g. random heterogeneous media we often deal with a very small correlation length and this results in a rather high-dimensional stochastic space with nearly the same weights along each dimension. In this case, all the previously developed stochastic methods are obviously not applicable.

- These modeling issues for high-dimensional stochastic problems motivate the use of the High Dimensional Model Representation (HDMR) technique.
The finite-dimensional noise assumption

- The solution of the problem of interest is described by a set of random variables, i.e.
  \[ u(x, \omega) = u(x, Y_1(\omega), \cdots, Y_N(\omega)) \]

- The SPDE problem is stated as: Find the stochastic function \( u : \Gamma \times D \to \mathbb{R} \) such that
  \[ L(x, Y; u) = f(x, Y), \quad \forall x \in D \times \Gamma \]
  \[ B(x, Y; u) = g(x, Y), \quad \forall x \in \partial D \times \Gamma \]

- In this work, we assume that \( \{Y_i(\omega)\}_{i=1}^N \) are independent random variables with probability density function \( \rho_i \). Let \( \Gamma_i \) be the image of \( Y_i \). Then
  \[ \rho(Y) = \prod_{i=1}^N \rho_i(Y_i), \quad \forall Y \in \Gamma \]
  is the joint probability density of \( Y = (Y_1, \cdots, Y_N) \) with support
  \[ \Gamma = \prod_{i=1}^N \Gamma_i \in \mathbb{R}^N \]
**Stochastic collocation based framework**

Function value at any point is simply a stochastic function in 2 dimensions.

Need to represent this function.

Sample the function at a finite set of points \( \Theta_M = \{Y^{(i)}\}_{i=1}^M \).

Use polynomials (Lagrange polynomials) to get a approximate representation:

\[
\mathcal{I}f(Y) = \sum_{i=1}^{M} f(Y^{(i)}) a_i(Y^{(i)})
\]

Spatial domain is approximated using a FEM discretization.

Stochastic domain is approximated using multidimensional interpolating functions.

Function value at any point is simply \( \mathcal{I}f(\xi) \).
Choice of collocation points and nodal basis functions

In the context of incorporating adaptivity, we use the Newton-Cotes grid with equidistant support nodes and the linear hat function as the univariate nodal basis.

\[ m_i = \begin{cases} 
1, & \text{if } i = 1 \\
2^{i-1} + 1, & \text{if } i > 1
\end{cases} \]

\[ Y_j^i = \begin{cases} 
\frac{j - 1}{m_i - 1}, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1, \\
0.5, & \text{for } j = 1, \text{ if } m_i = 1.
\end{cases} \]

In this manner, one ensures a local support and that discontinuities in the stochastic space can be resolved. The piecewise linear basis functions is defined as

\[ a_1^i = 1 \quad \text{for } i = 1, \text{ and} \]

\[ a_j^i = \begin{cases} 
1 - (m_i - 1) \cdot |Y - Y_j^i|, & \text{if } |Y - Y_j^i| < 1/(m_i - 1), \\
0, & \text{otherwise,}
\end{cases} \]

for \( i > 1 \) and \( j = 1, \ldots, m_i \).
Let our basic 1D interpolation scheme be summarized as

\[ U^i(f) = \sum_{j=1}^{k_i} f(Y^i_j) \cdot a^i_j \]

In higher dimensions, a simple case is the tensor product formula

\[ (U^{i_1} \otimes \cdots \otimes U^{i_N})(f) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_N=1}^{k_N} f(Y^{i_1}_{j_1}, \ldots, Y^{i_N}_{j_N}) \cdot (a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_N}_{j_N}) \]

Using the 1D formula, the sparse interpolant \( A_{q,N} \), where \( q \) is the depth of sparse grid interpolation \((q \geq 0, q \in \mathbb{N}_0)\) and \( N \) is the number of stochastic dimensions, is given by the Smolyak algorithm as

\[ A_{q,N}(f) = A_{q-1,N}(f) + \Delta A_{q,N}(f) \quad \Delta^i = U^i - U^{i-1} \quad A_{-1,N} = 0 \]

\[ U^0 = 0 \quad \|i\| = i_1 + \cdots + i_N \]

\[ A_{q,N}(f) = \sum_{\|i\| \leq N+q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}) A_{q,N}(f) = \sum_{\|i\| \leq N+q} \sum_{j \in B_i} \left( a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_N}_{j_N} \right) \]

Here, we define the hierarchical surplus as:

\[ w^i_j = f(Y^{i_1}_{j_1}, \ldots, Y^{i_N}_{j_N}) - A_{\|i\|-1,N}(f)(Y^{i_1}_{j_1}, \ldots, Y^{i_N}_{j_N}) \]
Nodal basis versus hierarchical basis

Nodal basis

Hierarchical basis

\[ f = f(Y_1^3) a_1^3 + f(Y_2^3) a_2^3 + f(Y_3^3) a_3^3 + f(Y_4^3) a_4^3 + f(Y_5^3) a_5^3 \]

\[ f = w_1^1 a_1^1 + w_1^2 a_1^2 + w_2^2 a_2^2 + w_1^3 a_3^3 + w_2^3 a_3^3 \]
Hierarchical Integration

- The *mean* of the random solution can be evaluated as follows:

\[
\mathbb{E}[f(x)] = \sum_{|i| \leq N+q, j \in B_i} w_j^i(x) \cdot \int_{\Gamma} a_j^i(Y) dY \cdot \int_0^1 a_j^i(Y) dY = \begin{cases} 
1, & \text{if } i = 1, \\
\frac{1}{4}, & \text{if } i = 2, \\
2^{1-i}, & \text{otherwise}.
\end{cases}
\]

- Denoting \( \int_{\Gamma} a_j^i(Y) dY = I_j^i \), we rewrite the mean as

\[
\mathbb{E}_q[f(x)] = \sum_{|i| \leq N+q, j \in B_i} w_j^i(x) \cdot I_j^i
\]

- To obtain the variance of the solution, we need to first obtain an approximate expression for \( u^2 \)

\[
u^2(x, Y) = \sum_{|i| \leq N+q, j \in B_i} v_j^i(x) \cdot a_j^i(Y)
\]

\[
\text{Var}[u(x)] = \mathbb{E}[u^2(x)] - (\mathbb{E}[u(x)])^2
\]

\[
= \sum_{|i| \leq N+q, j \in B_i} v_j^i(x) \cdot I_j^i - \left( \sum_{|i| \leq N+q, j \in B_i} w_j^i(x) \cdot I_j^i \right)^2
\]
Adaptive sparse grid collocation (ASGC)\textsuperscript{1}

Let us first revisit the 1D hierarchical interpolation

- For smooth functions, the hierarchical surpluses tend to zero as the interpolation level increases.

- Finite discontinuities are indicated by the magnitude of the hierarchical surplus.

- The bigger the magnitude is, the stronger the underlying discontinuity is.

- Therefore, the hierarchical surplus is a natural candidate for error control and adaptivity. If the hierarchical surplus is \textit{larger} than a pre-defined value (threshold), we simply add the $2N$ neighboring points to the current point.

\textsuperscript{1} X. Ma, N. Zabaras, An hierarchical adaptive sparse grid collocation method for the solution of stochastic differential equations, JCP, 228 (2009) 3084-3113.
Definition of the error indicator

- The mean of the random solution can be evaluated as follows:

\[
E[f(x)] = \sum_{\|i\| \leq N+q} \sum_{j \in B_i} w_j^i(x) \cdot \int_{\Gamma} a_j^i(Y) dY
\]

\[
\int_{0}^{1} a_j^i(Y) dY = \begin{cases} 
1, & \text{if } i = 1, \\
\frac{1}{4}, & \text{if } i = 2, \\
2^{1-i}, & \text{otherwise.}
\end{cases}
\]

- Denoting \( \int_{\Gamma} a_j^i(Y) dY = I_j^i \), we rewrite the mean as

\[
E_q[f(x)] = \sum_{\|i\| \leq N+q} \sum_{j \in B_i} w_j^i(x) \cdot I_j^i
\]

- We now define the error indicator as follows:

\[
\gamma_j^i = \frac{\|w_j^i(x) \cdot I_j^i\|_{L_2}}{\|E_{\|i\| - N - 1}\|_{L_2}}
\]

- In addition to the surpluses, this error indicator incorporates information from the basis functions. This forces the error to decrease to a sufficient small value for a large interpolation level. This error indicator guarantees that the refinement would stop at a certain interpolation level.
Algorithm 1 Adaptive sparse grid interpolation

Set level of Smolyak construction $q = 0$.
Construct the first level adaptive sparse grid $\mathcal{G}_{N,N}$.

- Calculate the function value at the point $(0.5, \ldots, 0.5)$.
- Generate the $2N$ neighbor points and add them to the active index set.
- Set $q = q + 1$.

while $q \leq q_{\text{max}}$ and the active index set is not empty do

- Copy the points in the active index set to an old index set and clear the active index set.
- Calculate in parallel the hierarchical surplus of each point in the old index set according to

$$w_j^i = f(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}) - \mathcal{A}_{q-1,N}(f)(Y_{j_1}^{i_1}, \ldots, Y_{j_N}^{i_N}).$$

Here, we use all of the existing collocation points in the current adaptive sparse grid $\mathcal{G}_{N+q-1,N}$. This allows us to evaluate the surplus for each point from the old index set in parallel.

- For each point in the old index set, if $\gamma_j^i \geq \varepsilon$
  - Generate $2N$ neighbor points of the current active point.
  - Add them to the active index set.

- Add the points in the old index set to the existing adaptive sparse grid $\mathcal{G}_{q-1,N}$. Now the adaptive sparse grid becomes $\mathcal{G}_{q,N}$.
- $q = q + 1$.

end while
Adaptive sparse grid interpolation

Ability to detect and reconstruct steep gradients

\[ f(x, y) = \frac{1}{|10^{-3} - x^2 - y^2| + 10^{-3}} \]

\[ f(x, y) = \frac{1}{|0.3 - x^2 - y^2| + 10^{-1}} \]
Simple extension of the concepts detailed before to the solution of a stochastic partial differential equation.

Given the stochastic partial differential equation

$$L(u : x, t, Y) = 0$$

Can construct an accurate approximation of the unknown variables $u$ by estimating the function values at the sparse grid positions.

This involves solving a set of deterministic equations

$$L\left(u(Y^{(i)}): x, t, Y^{(i)}\right) = 0, \ i = 1, ..., N$$
Solution Methodology

**PREPROCESSING**
Choose an appropriate algorithm, adaptive or conventional, based on number of stochastic dimensions, d and level of interpolation, q

Solve the deterministic problem defined by each set of collocated points

**POSTPROCESSING**
Compute moments and other statistics with simple operations of the deterministic data at the collocated points

Use any validated deterministic solution procedure.
Completely non intrusive

Std deviation of temperature: Natural convection
Given the stochastic dimension $M$

1) Choose an appropriate depth of interpolation, $q$

2) Construct the set of sparse points corresponding to this depth

3) If required, transform these points from $[0,1]^M$ to appropriate values.

4) Use simple transforms to move from finite support to infinite support. Transforms can be based on the cdf of the required distribution.

$\text{erf}(x)$
Once the function values are available at all the sparse points, one can construct the stochastic solution.

One is usually interested in the mean, standard deviation and the probability density function of the dependant variables at various points in the physical domain

1) Mean and higher moments:

   These moments are computed very efficiently as a linear combination of the hierarchical surpluses at the sparse grid points with a set of a priori computed weights as detailed before.

2) Probability density function (PDF)

   The function is approximated using the sparse grid method. Following this a large number of function evaluations are performed. A normalized histogram of the function distribution then gives the PDF
PDF:

If \( u \) is the stochastic solution that has been approximated using the collocation method, the PDF of \( u \) can be computed as follows:

1) Construct the stochastic solution using the sparse grid collocation method

2) For \( k \) random values from the domain of \( u \) (i.e. \( Y^{(i)}, i = 1, \ldots, k \)), compute the corresponding function values

3) Get the smoothed histogram of these values. You can use the Matlab function \( ksdensity.m \)
Other choices of sampling points

- An important feature of the sparse grid we discuss before is its nested nature. This feature allows us to write the Smolyak algorithm in a hierarchical fashion and use the so-called hierarchical surpluses as the error indicator.

- However, the support of these two grids in one dimension is [0,1], which means it corresponds to the uniform distribution. Therefore, if we want to treat other probabilistic distributions, we need to transform the uniform distribution. A possible way is to use the so-called inverse CDF transformation method. But, inverse CDF is only available for some distributions, such as Gaussian distribution.

- Therefore, we need other sampling points which can handle arbitrary probability measures. A possible choice is the Gauss quadrature points with Lagrange polynomial interpolation.

- It is noted that equal-distant points such as Newton-Cotes cannot be used as the sampling points for Lagrange polynomial interpolation due to the well-known Runge’s phenomenon.
There is a rich mathematical theory that relates ‘good’ interpolating functions to the best interpolating function.

Every interpolation function can be related to the best approximation polynomial through its Lebesgue constant $^1$

$$
\| f - \tilde{f} \|_\infty \leq \| f - p_n^*(f) \| (1 + \Lambda_n(X))
$$

The Lebesgue constant $\Lambda_n(X) = \max_{x \in [a,b]} (\sum_{i=0}^{n} | L_i(x) |)$

*Note that the Lebesgue constant depends only on the node distribution and not on the function.*

The distribution of Gauss quadrature points is shown to minimize the Lebesgue constants $^1$.

---

The correspondence between the Gauss quadrature points and the probability distribution is the same as that of gPC.

<table>
<thead>
<tr>
<th>Distribution of $\xi$</th>
<th>Askey-polynomials ${I_n}$</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>Hermite</td>
<td>$(-\infty, \infty)$</td>
</tr>
<tr>
<td>Gamma</td>
<td>Laguerre</td>
<td>$(0, \infty)$</td>
</tr>
<tr>
<td>Beta</td>
<td>Jacobi</td>
<td>$[a, b]$</td>
</tr>
<tr>
<td>Uniform</td>
<td>Legendre</td>
<td>$[a, b]$</td>
</tr>
</tbody>
</table>

In this case, the integration of the basis function with respect to the probability distribution is equal to the corresponding Gauss integration weights, which is also known *apriori*.

$$
\int_{\Gamma} a_j^i(Y) \rho(Y) dY = W_j^i
$$

where $W_j^i$ is the tensor product of the integration weights from each dimension.
**Gauss quadrature points: issues**

- The Gauss quadrature points are non equal-distant points. Thus, the interpolation basis function is the Lagrange polynomial basis.

- The Gauss quadrature points is a non-nested grid. The advantage of hierarchical interpolation is not very obvious. Therefore, in this case, we recommend to use the following version of Smolyak algorithm:

\[
A_{q,N}(f) = \sum_{q-N+1 \leq |i| \leq q} (-1)^{q-|i|} \cdot \binom{N-1}{q-|i|} \cdot (\mathcal{U}^i_1 \otimes \ldots \otimes \mathcal{U}^i_N)
\]

- It is noted here the coefficients of this formula are no longer the hierarchical surpluses. Instead, it is the function value at the collocation points.

- Only dimensional adaptive sparse grid collocation algorithm is applicable hear if we want to use Gauss quadrature for adaptive construction. Remember, the local adaptive algorithm is only applicable for equal-distant collocation points.
Interpolation or integration?

- In this lecture, we focused on sparse grid interpolation method. However, sparse integration follows exactly in the same way. In general, sparse grid integration (quadrature) is easier than interpolation since you do not need to construct the interpolant of the solution. Instead, it is just an arithmetic sum of the solution at the quadrature points multiplying the corresponding integration weights.

- In our opinion, we recommend focusing on the interpolation. It not only provides us the moments of the solution but also gives us an approximation of the solution. This approximation can be used to extract the PDF or as a surrogate stochastic model for other applications such as in Bayesian inference approach.

- If integration is used directly, there is no way to extract the PDF.
NUMERICAL EXAMPLES
The governing equations are:

\[
\begin{align*}
\frac{dy_1}{dt} &= y_1y_3, \\
\frac{dy_2}{dt} &= -y_2y_3, \\
\frac{dy_3}{dt} &= -y_1^2 + y_2^2
\end{align*}
\]

(X. Wan, G. Karniadakis, 2005)

The discontinuity then occurs at the planes \( y_1 = 0 \) and \( y_2 = 0 \).

We first study the stochastic response subject to the following random 1D initial input:

\[
y_1(0; \omega) = 1, \quad y_2(0; \omega) = 0.1Y(0; \omega), \quad y_3(0; \omega) = 0
\]

where \( Y \sim U(-1,1) \). Since the random initial data \( y_2(0; \omega) \) can cross the plane \( y_2 = 0 \) we know that gPC will fail for this case.
One-dimensional random input: variance

- gPC fails after $t = 6\text{s}$ while the ASGC method converges even with a large threshold. The solid line is the result from MC-SOBOL sequence with $10^6$ iterations.
From the adaptive sparse grid, it is seen that most of the refinement after level 8 occurs around the discontinuity point \( Y = 0.0 \), which is consistent with previous discussion.

The refinement stops at level 16, which corresponds to 425 number of points, while the conventional sparse grid requires 65537 points.
One-dimensional random input

Comparison of maximum error of the variance of $y_1$, $y_2$ and $y_3$ at $t = 30$ for the 1D K-O problem for the same number of sample points.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Level</th>
<th>#Points</th>
<th>Error</th>
<th>Error</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>11</td>
<td>117</td>
<td>$5.13 \times 10^{-3}$</td>
<td>$4.62 \times 10^{-2}$</td>
<td>$4.27 \times 10^{-2}$</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>16</td>
<td>425</td>
<td>$2.34 \times 10^{-4}$</td>
<td>$1.40 \times 10^{-2}$</td>
<td>$1.28 \times 10^{-3}$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>17</td>
<td>1381</td>
<td>$2.08 \times 10^{-5}$</td>
<td>$7.71 \times 10^{-3}$</td>
<td>$5.00 \times 10^{-4}$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>26</td>
<td>5349</td>
<td>$2.59 \times 10^{-6}$</td>
<td>$2.85 \times 10^{-3}$</td>
<td>$9.54 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

The ‘exact’ solution is taken as the results given by MC-SOBOL.

The error level is defined as

$$\max_{i=1,2,3} \left| \text{Var}(y_i) - \text{Var}(y_{i,MC}) \right|_{t=30}$$
One-dimensional random input: long term behavior

![Graphs showing mean and variance of y1, y2, y3 over time for different methods and ε values.](image-url)
These realizations are reconstructed using hierarchical surpluses. It is seen that at earlier time, the discontinuity has not yet been developed, which explains the reason the gPC is accurate at earlier times.
Next, we study the K-O problem with 2D input:

\[ y_1(0; \omega) = 1, \quad y_2(0; \omega) = 0.1Y_1(0; \omega), \quad y_3(0; \omega) = Y_2(0; \omega) \]

Now, instead of a point, the discontinuity region becomes a line.
It can be seen that even though the gPC fails at a larger time, the adaptive collocation method converges to the reference solution given by MC-SOBOL with $10^6$ iterations.
**Two-dimensional random input**

Comparison of maximum error of the variance of $y_1$, $y_2$ and $y_3$ at $t = 10$ for the 2D K-O problem for the same number of sample points.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>#Points</th>
<th>Error</th>
<th>Error</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>903</td>
<td>$3.95 \times 10^{-2}$</td>
<td>$4.62 \times 10^{-2}$</td>
<td>$1.30 \times 10^{-3}$</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>6457</td>
<td>$1.50 \times 10^{-4}$</td>
<td>$1.21 \times 10^{-2}$</td>
<td>$9.43 \times 10^{-4}$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>23099</td>
<td>$1.83 \times 10^{-5}$</td>
<td>$3.34 \times 10^{-3}$</td>
<td>$5.25 \times 10^{-5}$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>79213</td>
<td>$5.43 \times 10^{-6}$</td>
<td>$1.26 \times 10^{-3}$</td>
<td>$6.46 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

The error level is defined as

$$\max_{i=1,2,3} \left| \operatorname{Var}(y_i) - \operatorname{Var}(y_i,MC) \right|_{t=10}$$

Comparison of computational costs for the 2D K-O problem.

<table>
<thead>
<tr>
<th>Error level</th>
<th>ASGC</th>
<th>$h$-adaptive ME-gPC</th>
<th>$h$-adaptive ME-PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time(s)</td>
<td>#Points</td>
<td>Time(s)</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>5.82</td>
<td>1575</td>
<td>20.89</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>17.13</td>
<td>2251</td>
<td>34.69</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>27.01</td>
<td>6457</td>
<td>118.29</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>144.48</td>
<td>23099</td>
<td>383.57</td>
</tr>
</tbody>
</table>

The number of points of the conventional grid with interpolation 20 is 12582913.
Two-dimensional random input: Long term behavior

Error in the mean (left) and the variance (right) of $y_1$ with different thresholds for the 2D K-O problem in $[0, 50]$. 
Next, we study the K-O problem with 3D input:

This problem is much more difficult than any of the other problems examined previously. This is due to the strong discontinuity (the discontinuity region now consists of the planes $Y_1 = 0$ and $Y_2 = 0$) and the moderately-high dimension.

It can be verified from comparison with the result obtained by MC-SOBOL that unlike the previous results, here $2 \times 10^6$ iterations are needed to correctly resolve the discontinuity.

Due to the symmetry of $y_1$ and $y_2$ and the corresponding random input, the variances of $y_1$ and $y_2$ are the same.

Finally, in order to show the accuracy of the current implementation of the h-adaptive ME-gPC and ME-PCM, we provide the results for this problem.
Three-dimensional random input: ASGC

Evolution of the variance of $y_1 = y_2$ (left) and $y_3$ (right) for the 3D K-O problem using ASGC
Three-dimensional random input: ME-method

Evolution of the variance of $y_1 = y_2$ (left) and $y_3$ (right) for the 3D K-O problem using ME-method with $\theta_1 = 10^{-4}$ and $\theta_2 = 10^{-3}$
Three-dimensional random input

Due to the strong discontinuity in this problem, it took much longer time for all the methods to arrive at the same accuracy.

### Computational results for 3D K-O problem using ASGC.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>#Points</th>
<th>Error</th>
<th>Time</th>
<th>Error</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>46953</td>
<td>$1.95 \times 10^{-3}$</td>
<td>0.09 hour</td>
<td>$2.72 \times 10^{-3}$</td>
<td>$6.23 \times 10^{-5}$</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>210177</td>
<td>$2.95 \times 10^{-4}$</td>
<td>0.85 hours</td>
<td>$8.99 \times 10^{-4}$</td>
<td>$6.61 \times 10^{-5}$</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>498025</td>
<td>$1.58 \times 10^{-4}$</td>
<td>3 hours</td>
<td>$5.47 \times 10^{-4}$</td>
<td>$2.74 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

### Computational results for 3D K-O problem using $h$-adaptive ME-gPC.

<table>
<thead>
<tr>
<th>order $p$</th>
<th>$h$-adaptive ME-gPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Elements</td>
</tr>
<tr>
<td>3</td>
<td>5584</td>
</tr>
<tr>
<td>4</td>
<td>3256</td>
</tr>
<tr>
<td>5</td>
<td>2336</td>
</tr>
<tr>
<td>6</td>
<td>1624</td>
</tr>
</tbody>
</table>

### Computational results for 3D K-O problem using $h$-adaptive ME-PCM.

<table>
<thead>
<tr>
<th>Level $k$</th>
<th>$h$-adaptive ME-PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Points</td>
</tr>
<tr>
<td>6</td>
<td>590364</td>
</tr>
<tr>
<td>7</td>
<td>14305746</td>
</tr>
<tr>
<td>8</td>
<td>15788608</td>
</tr>
</tbody>
</table>
Here, we adopt the model problem

\[-\nabla \cdot (a_N(\omega, \cdot) \nabla u(\omega, \cdot)) = f_N(\omega, \cdot) \quad \text{in } D \times \Gamma\]

\[u(\omega, \cdot) = 0 \quad \text{on } \partial D \times \Gamma\]

with the physical domain \( D = \{(x, y) \in [0,1]^2\} \). To avoid introducing errors of any significance from the domain discretization, we take a deterministic smooth load \( f_N(\omega, x, y) = \cos(x) \sin(y) \) with homogeneous boundary conditions.

The deterministic problem is solved using the finite element method with 900 bilinear quadrilateral elements. Furthermore, in order to eliminate the errors associated with a numerical K-L expansion solver and to keep the random diffusivity strictly positive, we construct the random diffusion coefficient \( a_N(\omega, x) \) with 1D spatial dependence as

\[
\log\left(a_N(\omega, x) - 0.5\right) = 1 + \sum_{n=1}^{N} \left(\frac{\sqrt{\pi L}}{2}\right)^{1/2} e^{-\frac{(n-1)^2 \pi^2 L^2}{8}} \cos\left(2\pi x(n-1)\right) Y_n(\omega)
\]

where \( Y_n(\omega), n = 1, \ldots, N \) are independent uniformly distributed random variables in the interval \([-\sqrt{3}, \sqrt{3}]\).
Stochastic elliptic problem

In the earlier expansion, 

\[ \xi_n := \left(\sqrt{\pi L}\right)^{1/2} \exp \left(-\left(\frac{n}{2}\right)^2 \frac{\pi L}{8}\right), \quad \text{if } n > 1 \]

and 

\[ \phi_n(x) := \begin{cases} 
\sin \left(\frac{n}{2} \frac{\pi x}{L_p}\right), & \text{if } n \text{ even} \\
\cos \left(\frac{n}{2} \frac{\pi x}{L_p}\right), & \text{if } n \text{ odd} 
\end{cases} \]

The parameter \( L_p \) can be taken as \( L_p = \max \{1, 2L_c\} \) and the parameter \( L \) is \( L = L_c / L_p \)
Stochastic elliptic problem

- This expansion is similar to a K-L expansion of a 1D random field with stationary covariance

$$\log\left(a_N(\omega, x) - 0.5\right)(x_1, x_2) = \exp\left(-\frac{(x_1 - x_2)^2}{L^2}\right)$$

- Small values of the correlation $L$ correspond to a slow decay, i.e. each stochastic dimension weighs almost equally. On the other hand, large values of $L$ results in fast decay rates, i.e., the first several stochastic dimensions corresponds to large eigenvalues weigh relatively more.

- By using this expansion, it is assumed that we are given an analytic stochastic input. Thus, there is no truncation error. This is different from the discretization of a random filed using the K-L expansion, where for different correlation lengths we keep different terms accordingly.

- In this example, we fix $N$ and change $L$ to adjust the importance of each stochastic dimension. In this way, we investigate the effects of $L$ on the ability of the ASGC method to detect the important dimensions.
Stochastic elliptic problem: $N = 11$
Stochastic elliptic problem: N = 11

- We estimate the $L_2(D)$ approximation error for the mean and variance. Specifically, to estimate the computation error in the $q$-th level, we fix $N$ and compare the results at two consecutive levels, e.g. the error for the mean is

$$E\left[ A_{q,N}(u_N) - A_{q+1,N}(u_N) \right]$$

- The previous figures shows results for different correlation lengths at $N=11$.

- For small $L$, the convergence rates for the CSGC and ASGC are nearly the same. On the other hand, for large $L$, the ASGC method requires much less number of collocation points than the CSGC for the same accuracy.

- This is because more points are placed only along the important dimensions which are associated with large eigenvalues.

- Next, we study some higher-dimensional cases. Due to the rapid increase in the number of collocation points, we focus on moderate high correlation lengths so that the ASGC is effective.
Stochastic elliptic problem: higher-dimensions

$L_c = 0.6$

Comparison:
CSGC: 68074001

#points: 20271
In order to further verify our results, we compare the mean and the variance when \( N = 75 \) using the AGSC method with \( \varepsilon = 10^{-8} \) with the ‘exact’ solution obtained by MC-SOBOL simulation with \( 10^6 \) iterations.

\[
\frac{|E\left(A_{q,N} (u_N)\right) - E\left(u_{MC}\right)|}{E\left(u_{MC}\right)}
\]

Computational time: ASGC : 0.5 hour      MC: 2 hour
Correlation length affects the convergence of ASGC. If every dimension is equally important, the ASGC is not effective. It is even worse than standard MC method for a high dimensional problem as a result of the weak dependence on the dimensionality in the logarithmic term of the error bound.
Filled with air ($Pr = 0.7$). $Ra = 2500$, which is larger than the critical Rayleigh number, so that convection can be initialized by varying the hot wall temperature. Thus, the hot wall temperature is assumed to be a random variable.

Prior stochastic simulation, several deterministic computations were performed in order to find our the range where the critical point lies in.
As we know, below the critical temperature, the flow vanishes, which corresponds to a decaying curve. On the other hand, the growing curve represents the existence of heat convection. Therefore, the critical temperature lies in the range \([0.5,0.55]\).
Rayleigh-Bénard instability: deterministic formulation

- Steady state Nusselt number which denotes the rate of heat transfer:

\[ Nu = \frac{1}{\theta_h - \theta_c} \int_0^1 \frac{\partial \theta}{\partial y} \bigg|_{y=0} \, dx \]

Steady-state Nusselt number for different hot wall temperatures

<table>
<thead>
<tr>
<th>( \theta_h )</th>
<th>0.30</th>
<th>0.40</th>
<th>0.50</th>
<th>0.55</th>
<th>0.60</th>
<th>0.70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nu</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.01496</td>
<td>1.07357</td>
<td>1.17744</td>
</tr>
</tbody>
</table>

\[ Nu = 1 \rightarrow \text{Conduction} \]
\[ Nu > 1 \rightarrow \text{Convection} \]

- This again verifies the critical hot wall temperature lies in the range [0.5,0.55]. However, the exact critical value is not known to us. So, now we try to capture this unstable equilibrium using the ASGC method.
We assume the following stochastic boundary condition for the hot wall temperature:

$$\theta_h = 0.4 + 0.3Y$$

where $Y$ is a uniform random variable in the interval $[0,1]$. Following the discussion before, both conductive and convective modes occur in this range.
Rayleigh-Bénard instability: stochastic formulation

Solution of the variables versus hot-wall temperature at point (0.1,0.5)

Nonlinear: convection

Linear: conduction

0.541
Rayleigh-Bénard instability: stochastic formulation

- Prediction of the temperature when $\theta_h = 0.436984$ using ASGC (left) and the solution of the deterministic problem using the same $\theta_h$ (right).

- Fluid flow vanishes, and the contour distribution of the temperature is characterized by parallel horizontal lines which is a typical distribution for heat conduction.
Rayleigh-Bénard instability: stochastic formulation

- Prediction of the temperature when $\theta_h = 0.66789$ using ASGC (top) and the solution of the deterministic problem using the same $\theta_h$ (bottom).

Max = 4.33161
Max = 4.31858
Max = 0.667891
Max = 4.33384
Max = 4.3208
Max = 0.667891
Rayleigh-Bénard instability: Mean

Max = 1.73709

ASGC (top)

Max = 1.73123

MC-SOBOL (bottom)

Max = 1.7315

Max = 1.73128

Max = 0.5

Max = 0.549918

Max = 0.55
Rayleigh-Bénard instability: Variance

ASGC (top)                   MC-SOBOL (bottom)

Max = 3.30617
Max = 3.28535
Max = 0.0129218
Max = 3.29984
Max = 3.27903
Max = 0.0128783
Max = 3.30617
Max = 3.28535
Max = 0.0129218
Max = 3.29984
Max = 3.27903
Max = 0.0128783

0.012
0.011
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001

0.012
0.011
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
Motivation of HDMR

- Conventional and adaptive collocation methods are not suitable for high-dimensional problems due to their weakly dependence on the dimensionality (logarithmic) in the error estimate.

- Although ASGC can alleviate this problem to some extent, its performance depends on the regularity of the problem and the method is only effective when some random dimensions are more important than others.

- In e.g. random heterogeneous media we often deal with a very small correlation length and this results in a rather high-dimensional stochastic space with nearly the same weights along each dimension. In this case, all the previously developed stochastic methods are obviously not applicable.

- These modeling issues for high-dimensional stochastic problems motivate the use of the High Dimensional Model Representation (HDMR) technique.
The finite-dimensional noise assumption

The solution of the problem of interest is described by a set of random variables, i.e.

\[ u(x, \omega) = u(x, Y_1(\omega), \ldots, Y_N(\omega)) \]

The SPDE problem is stated as: Find the stochastic function \( u : \Gamma \times D \to \mathbb{R} \) such that

\[
L(x, Y; u) = f(x, Y), \quad \forall x \in D \times \Gamma \\
B(x, Y; u) = g(x, Y), \quad \forall x \in \partial D \times \Gamma
\]

In this work, we assume that \( \{Y_i(\omega)\}_{i=1}^N \) are independent random variables with probability density function \( \rho_i \). Let \( \Gamma_i \) be the image of \( Y_i \). Then

\[
\rho(Y) = \prod_{i=1}^N \rho_i(Y_i), \quad \forall Y \in \Gamma
\]

is the joint probability density of \( Y = (Y_1, \ldots, Y_N) \) with support

\[
\Gamma \equiv \prod_{i=1}^N \Gamma_i \in \mathbb{R}^N
\]
Let \( f(Y) \) be a real-value multivariate stochastic function: \( \mathbb{R}^N \rightarrow \mathbb{R} \), which depends on a \( N \)-dimensional random vector \( Y = (Y_1, Y_2, \ldots Y_N) \in [0,1]^N \). A HDMR of \( f(Y) \) can be described by

\[
f(Y) = f_0 + \sum_{s=1}^{N} \sum_{i_1 < \cdots < i_s} f_{i_1 \cdots i_s}(Y_{i_1}, \ldots, Y_{i_s})
\]

where the interior sum is over all sets of \( S \) integers \( i_1, \ldots, i_s \), that satisfy \( 1 \leq i_1 < \cdots < i_s \leq N \). This relation means that

\[
f(Y) = f_0 + \sum_{i=1}^{N} f_i(Y_i) + \sum_{i_1 < i_2} f_{i_1 i_2}(Y_{i_1}, Y_{i_2}) + \sum_{i_1 < i_2 < i_3} f_{i_1 i_2 i_3}(Y_{i_1}, Y_{i_2}, Y_{i_3}) + \cdots + \sum_{i_1 < \cdots < i_s} f_{i_1 \cdots i_s}(Y_{i_1}, \ldots, Y_{i_s}) + \cdots + f_{12\ldots N}(Y_1, \ldots, Y_N)
\]

It can be viewed as a finite hierarchical correlated function expansion in terms of the input random variables with increasing dimensions.

For most physical systems, the first- and second-order expansion terms are expected to have most of the impact upon the output.

This equation is often written in a more compact notation:

\[ f(Y) = \sum_{u \subseteq D} f_u(Y_u) \]

for a given set \( u \subseteq D \) where \( D := \{1, \ldots, N\} \) denotes the set of coordinate indices and \( f_\emptyset(Y_\emptyset) = f_0 \). Here, \( Y_u \) denotes the \(|u|\) - dimensional vector containing those components of \( Y \) whose indices belong to the set \( u \), where \(|u|\) is the cardinality of the corresponding set \( u \), i.e. \( Y_u = (Y_i)_{i \in u} \).

For example, if \( u = \{1, 3, 5\} \), then \(|u| = 3\) and \( f_u(Y_u) \) implies \( f_{135}(Y_1, Y_3, Y_5) \).

The component functions \( f_u(Y_u) \) can be derived by minimizing the error functional

\[
\int_{\Gamma} \left[ f(Y) - f_0 - \sum_{i=1}^{N} f_i(Y_i) - \cdots - \sum_{i_1 < \cdots < i_s} f_{i_1 \cdots i_s}(Y_{i_1}, \ldots, Y_{i_s}) \right]^2 d\mu(Y) \quad 0 \leq s \leq N
\]

subject to the orthogonal constraint

\[
\int_{\Gamma^N} f_u(Y_u) f_v(Y_v) d\mu(Y) = 0, \text{ for } u \neq v
\]
The measure $d\mu$ determines the particular form of the error functional and of the component functions.

By the variational principle, the component functions $f_u(Y_u)$ can be explicitly given as

$$f_u(Y_u) := \sum_{v \subseteq u} (-1)^{|u|-|v|} P_v f(Y_v)$$

where the measure $\mu$ induces the projection operator $P_u : \Gamma^N \rightarrow \Gamma^{|u|}$

$$P_u f(Y_u) := \int_{\Gamma^{N-|u|}} f(Y) d\mu_{D\setminus u}(Y)$$

where

$$d\mu_{D\setminus u}(Y) := \prod_{i \notin u} d\mu_i(Y_i)$$

There are two different forms of HDMR induced by different measure: ANOVA-HDMR and CUT-HDMR.
ANOVA-HDMR versus CUT-HDMR

**HDMR**

**ANOVA-HDMR**

Lebesgue measure

\[ d\mu(Y) = d(Y) = \prod_{i=1}^{N} Y_i \]

Dimensional integration

\[ P_u f(Y_u) := \int_{\Gamma^{N-|u|}} f(Y) dY_{\mathcal{D}\setminus u} \]

Computational expensive -- requires a N-dimensional integral for the constant terms

**CUT-HDMR**

Dirac measure

\[ d\mu(Y) = \prod_{i=1}^{N} \delta(Y_i - \bar{Y}_i) dY_i \]

at a reference point \( \bar{Y} = (\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_N) \)

Dimensional function

\[ P_u f(Y_u) := f(Y) \mid_{Y = \bar{Y} \setminus Y_u} \]

Computational efficient -- requires function evaluations at sample points
Within the framework of CUT-HDMR, we can write

$$f(Y) = \sum_{u \subseteq D} f_u(Y_u) = \sum_{u \subseteq D} \sum_{v \subseteq u} (-1)^{|u|-|v|} f(Y_v)_{Y=Y_v \backslash Y_v}$$

where the notation $Y=Y_v \backslash Y_v$ means that the components of $Y$ other than those indices that belong to the set $u$ equal to those of the reference point.

If the HDMR is a converged expansion, the choice of this point does not affect the approximation. In this work, the mean of the random input vector is chosen as the reference point.

Therefore, the $N$-dimensional stochastic problem is transformed to several lower-order $|v|$-dimensional problems $f(Y_v)_{Y=Y_v \backslash Y_v}$ which can easily solved by ASGC:

$$f(Y) = \sum_{u \subseteq D} \sum_{v \subseteq u} (-1)^{|u|-|v|} \sum_{||i|| \leq N+q} \sum_{j \in B_i} w^{ij}_v(x) \cdot a^i_j(Y_v)$$

where $w^{ij}_v(x)$ are the hierarchical surpluses for different sub-problems indexed by $v$ and $a^i_j(Y_v)$ is only a function of the coordinates belonging to $v$. 
Let us denote
\[ J_u = \sum_{v \subseteq u} (-1)^{|u|-|v|} \sum_{||i|| \leq N+q} \sum_{j \in B_i} w^{ij}(x) \cdot I^i_j \]

as the mean of the component function \( f_u \). Then the mean of the HDMR expansion is simply
\[ \mathbb{E}[f(Y)] = \sum_{u \subseteq D} J_u. \]

The basic conjecture underlying HDMR is that the component functions arising in typical physical systems will not likely exhibit high-order cooperativity among the input variables such that the 1st- and 2nd-order expansion terms are expected to have most of the impact upon the output and the contribution of higher-order terms would be insignificant.

In other words, instead of solving the \( N \)-dimensional problem directly using ASGC, which is impractical for extremely high dimensional problems, we only need to solve several **one-** or **two-** dimensional problems, which can be solved efficiently via ASGC.
Effective dimension of a stochastic function

Let $\hat{f} := \sum_{u \subseteq D} |J_u|$ be the sum of all contributions to the mean value. Here, $|$ denotes the absolute value.

Then, for the proportion $\alpha \in (0, 1]$, the truncation dimension is defined as the smallest integer $N_t$, such that

$$\sum_{u \subseteq \{1, \ldots, N_t\}} |J_u| \geq \alpha \hat{f},$$

whereas, the superposition dimension is defined as the smallest integer $N_s$, such that

$$\sum_{|u| \leq N_s} |J_u| \geq \alpha \hat{f}$$

The superposition dimension is also called the order of the HDMR expansion.

With the definition of effective dimensions, we can thus truncate the expansion and take only a subset $S$ of all indices $u \subseteq D$. Here we assume that the set $S$ satisfies the following admissibility condition:

$$u \in S \text{ and } v \subset u \Rightarrow v \in S$$

This is to guarantee that all the terms can be calculated via the recursive expression for computing the component functions.
Effective dimension of a stochastic function

- In practice, we always truncate the expansion by taking only a subset of all indices \( u \subseteq D \). We can define an interpolation formula \( A_{S,f} \) for the \( S \) approximation of \( f \) as

\[
A_{S,f} := \sum_{u \in S} A(f_u)
\]

- It is common to refer to the terms \( \{ f_u : |u| = l \} \) collectively as the \( l \)-“order- terms”. Then the expansion order is the maximum of \( l \). The number of collocation points in this expansion is defined as the sum of the number of points for each sub-problem, i.e. \( M = \sum_{v \in S} M_v \)

- However, the number of order- \( l \) component functions is \( \sum_{i=1}^{l} \frac{N!}{i!(N-i)!} \), which increases quickly with the number of dimensions. Therefore, we developed an adaptive version of HDMR.
We fix $\alpha \in (0, 1]$ and assume that $N_s$ and $N_t$, the corresponding superposition and truncation dimensions, are known. With the definition of the index set $S_{N_t,N_s} := \{u \subseteq \{1, \ldots, N_t\}, |u| \leq N_s\}$, we have the following theorem:

**Theorem 1.** Let $S = S_{N_t,N_s}$, and let $\mathcal{A}$ be the ASGC interpolant with the same error threshold $\varepsilon$ for all the sub-problems. Then:

$$|f - \mathcal{A}_S f| \leq c(N_s, N_t)\varepsilon + \varepsilon_t,$$

for all $f \in F_N$. Here, the constant $c(N_t, N_s)$ depends on the effective dimensions, but does not depend on the nominal dimension $N$. $\varepsilon_t$ is the truncation error of according to the definition of effective dimensions.

Therefore, it is expected that the expansion converges to the true value with decreasing error threshold $\varepsilon$ and increasing number of component functions.
For extremely high dimensional problems, even a 2nd order expansion is impractical due to the increase of the number of component functions. Therefore, we would like to develop an adaptive version of HDMR for automatically and simultaneously detecting the truncation and superposition dimensions.

We assume each component function $f_u$ is associated with a weight $\eta_u \geq 0$ which describes the contribution of the term $f_u$ to the HDMR.

First, we try to find the important dimensions. To this end, we always construct the 0th - and 1st-order HDMR expansion. We define a weight:

$$\eta_i = \frac{\| J\{i\} \|_{L_2}}{\| f_0(\overline{Y}) \|_{L_2}} \quad J\{i\} = \int f_i(Y_i) dY_i$$

Then we define the important dimensions as those whose weights are larger than a predefined error threshold $\theta_1$. Only higher-order terms which consist of only these important dimensions are considered.

Here, the $L_2$ norm is defined in the spatial domain.
Adaptive HDMR

- For example, if the important dimensions are 1, 3 and 5, then only the higher-order terms \{13\}, \{15\}, \{35\} and \{135\} are considered.

- However not all the possible terms are computed. For higher-order term, a weight is also defined as

\[ \eta_u = \frac{\| J_u \|_{L_2}}{\| \sum_{v \in S, |v| \leq |u|-1} J_v \|_{L_2}} \]

We also define the important terms in a similar way. We put all the important dimensions and higher-order terms in to a set \( \mathcal{T} \). When adaptively constructing HDMR for each new order, we only calculate the term \( f_u \) whose indices satisfy the admissibility relation

\[ u \in \mathcal{D} \text{ and } v \subset u \Rightarrow v \in \mathcal{T} \]
Continued with the example, now if we want to construct the 2nd-order expansion, only \{13\}, \{15\} and \{35\} are calculated.

Then we compute the weights for each term. Assume \{13\} is the important term, the important index set \( T = \{0, 1, 3, 5, 13\} \).

Now, we go to 3rd order expansion. The only possible term is \{135\}. Its subsets \{35\} and \{15\} do not belong to the important index set \( T \), i.e. \{1, 3, 5\} does not satisfy the admissibility condition. Therefore, the construction stops.

In other words, among all possible indices, we only find the terms which can be computed using the previous known important component functions and have significant contributions to the overall expansion.
Let us denote the order of expansion as $p$. Furthermore, we also define a relative error $\rho$ of the integral value between two consecutive expansion orders $p$ and $p - 1$ as

$$\rho = \frac{\left\| \sum_{|u| \leq p} J_u - \sum_{|u| \leq p-1} J_u \right\|_{L_2}}{\left\| \sum_{|u| \leq p-1} J_u \right\|_{L_2}}$$

If $\rho$ is smaller than another predefined error threshold $\theta_2$, the HDMR is regarded as converged and the construction stops.

In this way, the construction will automatically stop and then the obtained HDMR expansion can be used as a stochastic surrogate model (response surface) for the solution. Any statistics can be easily computed through this expansion.
Algorithm 2 Adaptive construction of the index set $S$

Initialize: Let $S = \{\emptyset\}$, $R = \{\emptyset\}$ and $T = \{\emptyset\}$. Set $p = 1$.

Construct the zeroth and first-order component functions:

- Solve each sub-problem using the ASGC method with error threshold $\varepsilon$ and add all the indices to $S$.
- Compute the weights of each first-order term. Add those dimensions which satisfy $\eta \geq \theta_1$ to set $T$.

repeat

- $p \leftarrow p + 1$. Construct the set $R$ whose indices satisfy the admissibility relation for $|u| = p$.
- If $R \neq \{\emptyset\}$, for each index $u \in R$, solve the corresponding sub-problem using ASGC with error threshold $\varepsilon$ and add all the indices to $S$.
- Compute the weight of component functions. Add those indices which satisfy $\eta \geq \theta_1$ to set $T$ and clear set $R$.
- Compute the relative error $\rho$.

until $R = \{\emptyset\}$ or $\rho < \theta_2$;
Numerical example: Flow through random media

Basic equation for pressure and velocity in a domain

\[ \nabla \cdot \mathbf{u} = f \quad \text{in} \quad D \]
\[ \mathbf{u} = -k \nabla p \quad \text{in} \quad D \]

where \( f(x) \) denotes the deterministic source/sink term. Homogeneous boundary condition is applied. Mixed finite element method is used to solve the deterministic problem at the collocation points.

➢ To impose the non-negativity of the permeability, we will treat the permeability as a log random field obtained from the K-L expansion

\[ Y(\omega) = \log(K(\omega)) = \sum_{i=1}^{N} \sqrt{\lambda_i} \phi_i(x) Y_i \]

where \( Y \) is a zero mean Gaussian random field with covariance function

\[ \text{Cov}(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp \left( -\frac{|x_1 - y_1|}{L} - \frac{|x_2 - y_2|}{L} \right) \]

where \( L \) is the correlation length and \( \sigma \) is the standard deviation.
Numerical example: K-L Expansion

- The eigenvalues and their corresponding eigenfunctions can be determined analytically. The $Y_i$ are assumed as i.i.d. uniform random variables on $[-1,1]$.

- According to the decay rate of eigenvalues, the number of stochastic dimensions is $N = 33,108$ and $500$, respectively for $L = 1.0, 0.5$ and $0.25$.

- Monte Carlo simulations are conducted for the purpose of comparison. For each case, the reference solution is taken from $10^6$ samples and all errors are defined as normalized $L_2$ errors. In all cases, $\theta_2 = 10^{-4}$.
Standard deviation for different correlation lengths

\[ \sigma^2 = 1.0, \epsilon = 10^{-6} \]

\[ N = 500 \]
error = 1.39 \times 10^{-3}

\[ N = 108 \]
error = 1.19 \times 10^{-3}

Number of component functions is 2271 while for the full 2nd-order expansion it is 125251. The advantage of using adaptive HDMR is obvious.

Standard deviation of the \( v \) velocity-component along the cross section \( y = 0.5 \) for different correlation lengths
PDF at (0, 0.5) for different correlation lengths

PDF of the \( v \) velocity-component at point (0, 0.5) for different correlation lengths

\( N = 33 \)

\( N = 500 \)

\( N = 108 \)

\( \sigma^2 = 1.0, \epsilon = 10^{-6} \)
Convergence of the normalized errors

\[ \sigma^2 = 1.0, \theta_1 = 10^{-4} \]

\( N = 33 \quad N = 500 \quad N = 108 \)

Algebraic convergence rate better than MC

Nearly the same for three cases, it does not depend on the smoothness of the random space

Convergence of the normalized errors of the standard deviation of the \( v \) velocity-component for different correlation lengths
Standard deviations for different $\sigma^2$ with $N = 500$

$e = 8.08 \times 10^{-4}$

$\sigma^2 = 0.01$

$\sigma^2 = 0.25$

$e = 7.37 \times 10^{-4}$

Standard deviation of the $v$ velocity-component along the cross section $y = 0.5$ for different $\sigma^2$. 

$\sigma^2 = 2.0$
PDF at $(0, 0.5)$ for different $\sigma^2$ with $N = 500$

For low input variability, even 1st-order expansion is accurate.

For high input variability, the 1st-order expansion deviates from MC. More component terms are needed to improve accuracy.

For moderate input variability, 1st order does not deviate significantly from MC. However, a few 2nd-order terms are still needed.

$L = 0.25, N = 500$
Convergence of the normalized errors with $N = 500$

Direct solution of the 500 dimensional problem using ASGC is impractical due to the huge computational cost.

Convergence rate deteriorates with increasing input variability. However, it is still better than that of MC.
Effect of choices of the reference point

From our test studies, the mean vector always gives satisfactory results with much less computational cost.

It is also interesting to note that when \( \bar{Y} = (0.6, \ldots, 0.6) \), in order to achieve an error of order \( O(10^{-3}) \), the number of component functions is 880 while the number is 6018 when using conventional HDMR.

<table>
<thead>
<tr>
<th>( \bar{Y} )</th>
<th># Terms</th>
<th># Points</th>
<th>( N_i )</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.5, \ldots, 0.5)</td>
<td>287</td>
<td>27968</td>
<td>23</td>
<td>2</td>
</tr>
<tr>
<td>(0.6, \ldots, 0.6)</td>
<td>880</td>
<td>170992</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>(0.1, \ldots, 0.1)</td>
<td>5526</td>
<td>1854460</td>
<td>33</td>
<td>4</td>
</tr>
</tbody>
</table>

\( N_i \) is the number of important dimension

\( p \) is the highest expansion order
Natural Convection: $L = 0.1$ and $\sigma = 1.0$

$N = 10, \ Y_i \sim [-1, 1]$
Natural Convection: $L = 0.1$ and $\sigma = 1.0$

- We choose $\epsilon = 10^{-5}$ and $\theta_1 = 10^{-4}$. **The construction stopped after order 2.** However, not all terms are needed. **Only 17 component functions are in the final HDMR.**
- Only **1085 collocation points** are needed to achieve an error $1.78 \times 10^{-3}$. 

---

**Graph Details:**

- The graph shows the normalized error of std against $M$ for $L = 0.1, \sigma = 1.0$.
- The error is approximately $O(M^{-1.61})$.
- The HDMR with $\theta_1 = 10^{-4}$ is also shown on the graph.
Seamlessly couple stochastic analysis with multiscale analysis.

Multiscale framework (large deformation/thermal evolution) + Adaptive stochastic collocation framework

Provides roadmap to efficiently link any validated multiscale framework

Coupled with a data-driven input model strategy to analyze realistic stochastic multiscale problems.
Use sparse grid collocation to obtain the stochastic characteristic of macroscale properties.

Mean of A at the end of deformation process

Variance of A at the end of deformation process

<table>
<thead>
<tr>
<th>$\bar{E}$ (MPa)</th>
<th>Var($E$) (MPa)²</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.41e05</td>
<td>4.42e08</td>
</tr>
<tr>
<td>1.41e05</td>
<td>4.39e08</td>
</tr>
</tbody>
</table>

Adaptive Sparse grid (level 8)

MC 10,000 runs

FCC copper

B. Kouchmeshky and N. Zabaras, Computational Materials Science 47 (2009) 342--352
Some input parameters, boundary conditions, initial conditions, or operational sequences have a significant impact on the stochastic solution.

Improved input representation (i.e. lesser uncertainty) significantly reduces the uncertainty in the final solution.

The stochastic solution is sensitive to which parameters or input terms? Need a robust, seamlessly interfacing methodology to estimate this.

- Which input parameters to consider as random?
- Quantify the effect of uncertainty on each parameter on system response
- Decide the length scale of experimentation and the needed fidelity of each measurement.
Define a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$. We are interested to find stochastic functions $u : \Omega \times D \to \mathbb{R}$, $p : \Omega \times D \to \mathbb{R}$, and $S : \Omega \times [0,T] \times D \to \mathbb{R}$ and such that for $P$-almost everywhere (a.e.) $\omega \in \Omega$, the following holds:

\[
\nabla \cdot u(x, \omega) = 0, \quad u(x, \omega) = -K(x, \omega)\nabla p(x, \omega) \quad \forall x \in D
\]

\[
\frac{\partial S(x, t, \omega)}{\partial t} + u(x, t, \omega) \cdot \nabla S(x, t, \omega) = 0, \quad \forall x \in D, t \in [0, T]
\]

with the boundary condition

\[
p = \bar{p} \quad \text{on} \quad \partial D_p, \quad u \cdot n = 0 \quad \text{on} \quad \partial D_u.
\]

Length scale of the system,

Length scale of permeability variation,

**Multiscale paradigm**: Construct a strategy that solves for the coarse-scale features while accounting for the fine-scale features.

But exact permeability unknown. Some statistics or limited data

Permeability is a realization from corresponding probability space $K(\omega, .) \in \Omega$. 
For each collocation point, \( m = 1, \ldots, M \), we have:

\[
\nabla \cdot \mathbf{u}(\mathbf{x}, Y^m) = 0
\]

\[
\mathbf{u}(\mathbf{x}, Y^m) = -k(\mathbf{x}, Y^m) \nabla p(\mathbf{x}, Y^m)
\]

\[
\frac{\partial S(\mathbf{x}, Y^m)}{\partial t} + \mathbf{u}(\mathbf{x}, Y^m) \cdot \nabla S(\mathbf{x}, Y^m) = 0
\]

Local subgrid problem:

\[
\nabla \tilde{u}_{ik}(x) = 0, \quad \tilde{u}_{ik}(x) = -K \nabla \tilde{p}_{ik}(x), \quad \forall x \in E
\]

\[
\tilde{u}_{ik} \cdot \mathbf{n} = Q_{ik} \cdot \frac{T_{\nu_a}}{\sum_{\nu_b \subseteq \Lambda} T_{\nu_b} |\nu_b|}, \quad \text{on } \Lambda \subset \partial E
\]

Modified bilinear form:

\[
A_h(K^{-1} \mathbf{u}_c, \mathbf{v}_c) := \sum_{E \in T_c} \sum_{k=1}^4 \frac{\tau_k}{|E|} \int_E U_k(x) \cdot K^{-1} V_k(x) \, dx
\]

Coarse scale weak form:

\[
A_h(K^{-1} \mathbf{u}_c, \mathbf{v}_c) - (p_c, \nabla \cdot \mathbf{v}_c) = - \langle p_0, \mathbf{v}_c \cdot \mathbf{n} \rangle, \quad \forall \mathbf{v}_c \in V_c
\]

\[
(l_c, \nabla \cdot \mathbf{u}_c) = 0, \quad \forall l_c \in W_c
\]

\[
\nabla \cdot \mathbf{u}_h = 0, \quad \mathbf{u}_h = -K \nabla p_h, \quad \forall x \in E,
\]

\[
\mathbf{u}_h \cdot \mathbf{n}|_\Lambda = q^c \cdot \frac{T_{\nu_a}}{\sum_{\nu_b \subseteq \Lambda} T_{\nu_b} |\nu_b|}, \quad \text{on } \Lambda \subset \partial E
\]

Reconstruct fine scale velocity:
Solution Methodology

Generate the permeability sample given the collocation point, set coarse discretization

Compute the stiffness matrix for each coarse element

Compute the stochastic coarse-scale fluxes

Solve the subgrid problems for each basis function at quadrature points

POSTPROCESSING: Compute the statistics of the solution

Solve stochastic multiscale problem with HDMR

Return function value at collocation point

Solve the transport problem

Reconstruct the fine-scale velocity

Solve the subgrid problems with coarse-scale flux
**Example**

Fine-scale grid: $80 \times 80$

Coarse-scale scale grid: $20 \times 20$

$$Y(\omega) = \log (K(\omega)) = \sum_{i=1}^{N} \sqrt{\lambda_i} \phi_i(\mathbf{x}) Y_i$$

$$\text{Cov}(\mathbf{x}, \mathbf{y}) = \sigma^2 \exp \left( -\frac{|x_1 - y_1|}{L} - \frac{|x_2 - y_2|}{L} \right)$$

where $Y$ is a zero mean random field with covariance function $L$ is the correlation length and $\sigma$ is the standard deviation.

$L = 0.1, \quad \sigma = 1.0 \Rightarrow N = 500$

- We truncate the stochastic dimension after 500 terms and we assume the random variables to be uniform in $[-1,1]$. 
It is interesting to note that the mean is nearly the same as the homogeneous solution with the mean permeability. This is called “heterogeneity-induced dispersion”, where the heterogeneity smoothes the water saturation problem in the ensemble sense, although individual realizations show heterogeneity.
The results indicate that higher water saturation variance are concentrated near displacement fronts, which are areas of steep saturation gradients.

The number of component functions is 1281 and the total number of collocation points is 36087, while there are $10^5$ MC samples.
MC method with fine scale solver

HDMR method with multiscale solver

- The saturation front has the largest variation occurred.
Curse of Dimensionality: Conclusions

- Curse of Dimensionality for stochastic PDEs
  - Higher-order hierarchical basis functions (not easy to implement)
  - Relevance kernel machines – a Bayesian approach (very promising … more on this soon from our group …)
  - Need to take advantage of sparsity

- These techniques make sense if driven by realistic (data-driven) stochastic input models (future lecture on the OA Forum):
  - Manifold learning methods, Kernel PCA
  - …..

- How about model reduction of the underlying SPDEs (e.g. stochastic POD!)