RAVEN Workshop

Advanced UQ With Collocation Methods

Nuclear Engineering Methods Development Department
Idaho National Laboratory

Idaho National Laboratory

Table of Contents

Overview

Overview: Session Goal

Use advanced methods to accelerate UQ

- For low dimensionality, far fewer runs
- For smooth respones, provides accurate ROMs
- ROMs contain analytic mean, variance, sensitivities

Motivations

Motivations: Monte Carlo

Gold standard in UQ is Monte Carlo, Latin Hypercube

- Consistently convergent (central limit theorem)
- Easy to develop
- Error diminishes slowly
- Requires $1/\epsilon^2$ samples to achieve error ϵ

Motivations: Alternatives to Monte Carlo

Some structured samplers can improve greatly on Monte Carlo
Example: Stochastic Collocation for generalized Polynomial Chaos

- Expand model in orthogonal polynomials
- Use integration to determine expansion coefficients
- Quadrature methods perform polynomial integrations

Input dimensions need to be orthogonalized before running

Methods

Methods: generalized Polynomial Chaos expansion

$$f(x) \approx G(x) = \sum_{k \in \Lambda} c_k \Phi_k(x), \tag{1}$$

where

- $f(x)$ is the original model,
- x are the uncertain inputs in $f(x)$,
- $G(x)$ is the generalized Polynomial Chaos expansion,
- k is a multi-index to a polynomial, e.g. (3,1,2),
- Λ is a pre-determined set of polynomial indices,
- c_k are expansion coefficients,
- $\Phi_k(x)$ are multidimensional orthogonal polynomials

Methods: gPC Coefficients

$$f(x) \approx G(x) = \sum_{k \in \Lambda} c_k \Phi_k(x), \tag{2}$$

$\Phi_k(x)$ are orthogonal,

$$c_k = \int_\Omega \rho(x) f(x) \Phi_k(x) dx \tag{3}$$

where $\rho(x)$ is the joint probability distribution.
Numerically, use quadratures for c_k

$$c_k \approx \sum_{\ell=1}^{L} w_\ell f(x_\ell) \Phi_k(x_\ell)) \tag{4}$$

where w_ℓ are the weights and x_ℓ are the points

Methods: Large Dimensionality

Tensor SCgPC only efficient if $N < 4$
Fight curse of dimensionality using Sparse Grids

- Hyperbolic Cross
- Total Degree
- Smolyak Quadrature

Methods: SCgPC in RAVEN

ROM

```
<ROM name="rom" subType="GaussPolynomialRom">
  <Target>ans</Target>
  <Features>y1,y2</Features>
  <IndexSet>TensorProduct</IndexSet>
  <PolynomialOrder>2</PolynomialOrder>
</ROM>
```

Sampler

```
<SparseGridCollocation name="sc">
  <variable name="y1">
    <distribution>uni</distribution>
  </variable>
  <variable name="y2">
    <distribution>uni</distribution>
  </variable>
  <ROM class="Models" type="ROM">rom</ROM>
  <Restart class="DataObjects" type="PointSet">collset</Restart>
</SparseGridCollocation>
```

Methods: SCgPC in RAVEN

Steps

```
<MultiRun name="sample" sleepTime="1e-4">
  <Input class="DataObjects" type="PointSet">dummyIN</Input>
  <Sampler class="Samplers" type="SparseGridCollocation">sc</Sampler>
  <Model class="Models" type="ExternalModel">poly</Model>
  <Output class="DataObjects" type="PointSet">collset</Output>
</MultiRun>
```

```
<RomTrainer name="train">
  <Input class="DataObjects" type="PointSet">collset</Input>
  <Output class="Models" type="ROM">rom</Output>
</RomTrainer>
```

```
<IOStep name="stats">
  <Input class="Models" type="ROM">rom</Input>
  <Output class="OutStreams" type="Print">stats_tp2</Output>
</IOStep>
```

Methods: Interpolation Node

What if I want to use a different quadrature?
What if some dimensions require higher-order polynomials than others?
Specify through an interpolation node

```
<ROM name="rom" subType="GaussPolynomialRom">
  <Target>ans</Target>
  <Features>y1,y2</Features>
  <IndexSet>TotalDegree</IndexSet>
  <PolynomialOrder>3</PolynomialOrder>
  <Interpolation quad="Legendre" weight="0.7">y1</Interpolation>
  <Interpolation quad="Legendre" weight="0.5">y2</Interpolation>
</ROM>
```

Methods: Adaptive

What if I don't know what dimensions are higher order?
Use the Adaptive SCgPC sampler!

```
<AdaptiveSparseGrid name="sc">
  <variable name="y1">
    <distribution>uni</distribution>
  </variable>
  <variable name="y2">
    <distribution>uni</distribution>
  </variable>
  <ROM class="Models" type="ROM">rom</ROM>
  <Restart class="DataObjects" type="PointSet">collset</Restart>
  <TargetEvaluation class="DataObjects" type="PointSet">collset</TargetEvaluati
  <Convergence maxRuns="100" target="variance">1e-15</Convergence>
  <convergenceStudy>
    <runStatePoints>4,8,16,32,64</runStatePoints>
    <baseFilename>stats_adapt</baseFilename>
  </convergenceStudy>
</AdaptiveSparseGrid>
```

Methods: Sobol Decomposition

Methods: Sobol Decomposition

Another kind of expansion

$$
\begin{aligned}
f(x,y,z) = & f_0 \\
& + f_1(x) + f_2(y) + f_3(z) \\
& + f_{1,2}(x,y) + f_{1,3}(x,z) + f_{2,3}(y,z) \\
& + f_{1,2,3}(x,y,z),
\end{aligned}
$$

where $f_1(x) = \int \int f(x,y,z)\; dy\; dz$, etc.
Benefits:

- Many problems dominated by low-order interacations
- Provides easy access to Sobol sensitivities
- Each sub-term can be modelled as SCgPC ROM
  - Most are dimension 2 or less!
- Can also be constructed adaptively
  - Highest efficiency: Adaptive Sobol with Adaptive SCgPC

Methods: Sobol in RAVEN

Sampler

```
<AdaptiveSobol name="sc">
  <variable name="y1">
    <distribution>uni</distribution>
  </variable>
  <variable name="y2">
    <distribution>uni</distribution>
  </variable>
  <ROM class="Models" type="ROM">rom</ROM>
  <Restart class="DataObjects" type="PointSet">collset</Restart>
  <TargetEvaluation class="DataObjects" type="PointSet">collset</TargetEvaluati
  <convergenceStudy>
    <runStatePoints>4,8,16,32,64</runStatePoints>
    <baseFilename>stats_adsob</baseFilename>
  </convergenceStudy>
  <estimateMethod>product</estimateMethod>
  <Convergence>
    <relTolerance>1e-30</relTolerance>
    <maxRuns>100</maxRuns>
    <maxSobolOrder>2</maxSobolOrder>
    <logFile>adsob.log</logFile>
  </Convergence>
</AdaptiveSobol>
```

Methods: Sobol in RAVEN

ROM

```
<ROM name="rom" subType="HDMRRom">
  <Target>ans</Target>
  <Features>y1,y2</Features>
  <IndexSet>TotalDegree</IndexSet>
  <PolynomialOrder>1</PolynomialOrder>
  <SobolOrder>5</SobolOrder>
</ROM>
```

Demonstration: Attenuation Problem

Demonstration: the Model

$$f(x) = \prod_{n=1}^{N} \exp(-x_n/N), \tag{5}$$

$$x_n \sim \mathcal{U}(0,1). \tag{6}$$

Taylor expansion suggests many combinations of high-order terms

$$\exp(-x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots \tag{7}$$

Expected:

- Tensor Product, Total Degree, Adaptive perform well
- Hyperbolic Cross should struggle

Demonstration: Results

Two-dimension case

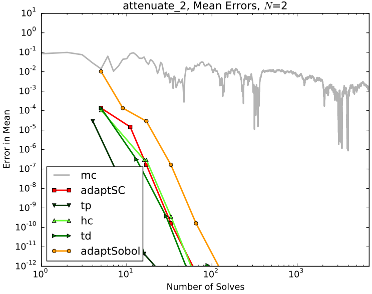

Figure: Mean, $N = 2$

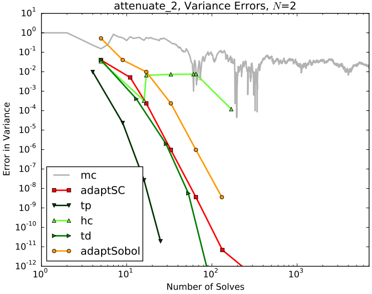

Figure: Variance, $N = 2$

Demonstration: Results

Four-dimension case

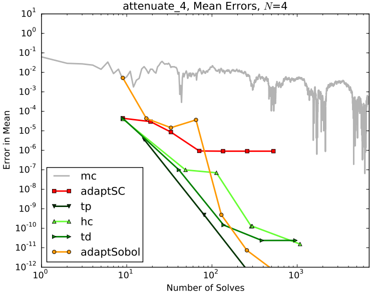

Figure: Mean, $N = 4$

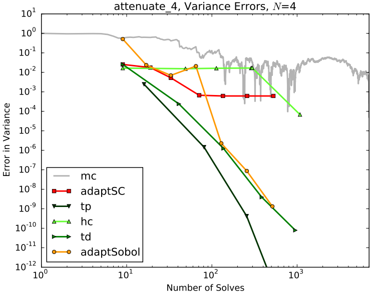

Figure: Variance, $N = 4$

Demonstration: Results

Six-dimension case

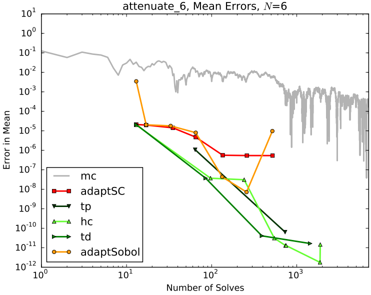

Figure: Mean, $N = 6$

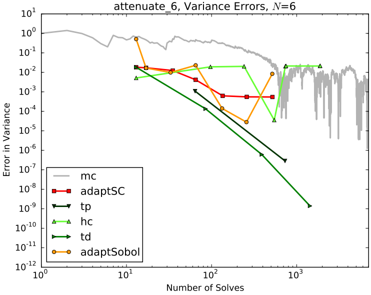

Figure: Variance, $N = 6$