RAVEN Workshop

Multi-Step Input Reduction

Nuclear Engineering Methods Development Department
Idaho National Laboratory

Discussion Points

Overview

Overview: Uncertainty Quantification

Benefits:

- Quantity of Interest Variance
- Failure Probabilities
- Limit Surface Construction
- Design of Experiment

Overview: Session Goal

Reduce high-dimension input spaces

- Use PCA to eliminate correlated inputs
- Use sensitivity to eliminate low-impact inputs
- Accurate UQ on a reduced input space

Overview: Assumptions

We assume:

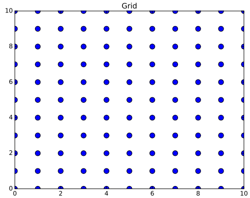
- Simulation codes are expensive to run
- All inputs are initially perturbable
- Saving time and money is good

Motivations

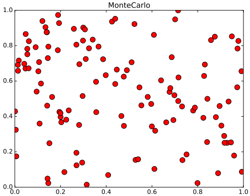
Motivations: Sampling Strategies

Two classes of forward sampling strategies in RAVEN:

- Structured (Orthogonal Grids, Limit Surface, Latin Hypercube)



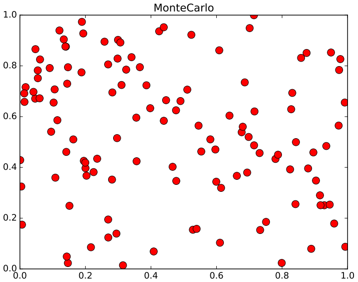
- Unstructured (Monte Carlo)



Motivations: Unstructured Sampling

Traditional Monte Carlo sampling

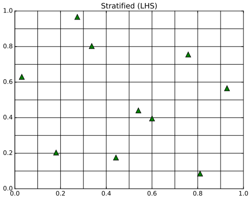
- Agnostic of Dimensionality
- Consistent, but slow convergence

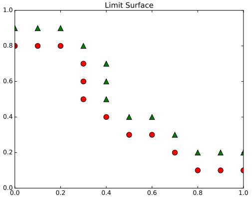


Motivations: Structured Sampling

RAVEN has several structured solvers

- Orthogonal Grids
- Sparse Grid
- Stratified
- Limit Surface Search





All of these suffer from the Curse of Dimensionality

Motivations: Proposed Solution

Remove unnecessary input dimensions

- Correlated inputs have redundancy
- Low-impact inputs aren't useful to perturb

Perform UQ on reduced space

Methods

Methods: Procedure Overview

- (optional) Benchmark original problem
- Perform PCA on original input space
- Truncate to essential latent variables
- Perform global sensitivity analysis
- Truncate latent variables to exclude non-essential

Methods: Principal Component Analysis

Methods: Principal Component Analysis

Used to orthogonalize input space

Represent input space as sum of "latent" variables $\boldsymbol{L}$

$$\boldsymbol{M} = \boldsymbol{Q} \cdot \boldsymbol{L}, \tag{1}$$

where

- $\boldsymbol{M}$ is the vector of original variables, $|M| \times 1$,
- $\boldsymbol{Q}$ is the PCA transformation matrix, $|M| \times |L|$,
- $\boldsymbol{L}$ is the vector of latent variables, $|L| \times 1$

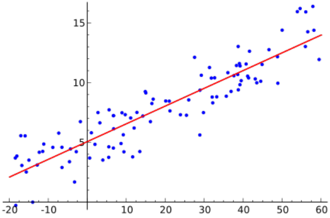
All $\boldsymbol{L}$ are distributed as standard normal distributions

Methods: Principal Component Analysis

$$\boldsymbol{M} = \boldsymbol{Q} \cdot \boldsymbol{L}, \tag{2}$$

Perform eigenvalue decomposition of covariance matrix and rank eigenvalues

Truncate latent variables at desirable eigenvalue



Methods: PCA in RAVEN

Add transformation to `Sampler`:

```
<MonteCarlo name='mcsamp'>
  <samplerInit>
    <limit>100</limit>
  </samplerInit>
  <variablesTransformation distribution="mvn">
    <manifestVariables>x1,x2,x3,x4,x5,x6</manifestVariables>
    <latentVariables>y1,y2,y3,y4,y5,y6</latentVariables>
    <method>pca</method>
  </variablesTransformation>
  <variable name='y1'>
    <distribution dim='1'>mvn</distribution>
  </variable>
  <variable name='y2'>
    <distribution dim='2'>mvn</distribution>
  </variable>
  <variable name='y3'>
    <distribution dim='3'>mvn</distribution>
  </variable>
  <variable name='y4'>
    <distribution dim='4'>mvn</distribution>
  </variable>
  <variable name='y5'>
    <distribution dim='5'>mvn</distribution>
  </variable>
  <variable name='y6'>
    <distribution dim='6'>mvn</distribution>
  </variable>
</MonteCarlo>
```

Methods: PCA in RAVEN

Use a PostProcessor to extract the rankings

PostProcessor

```
<PostProcessor name="stats" subType="ImportanceRank">
  <what>pcaindex</what>
  <features>y1,y2,y3,y4,y5,y6</features>
  <targets>ans</targets>
  <dimensions>1,2,3,4,5,6</dimensions>
  <mvnDistribution>mvn</mvnDistribution>
</PostProcessor>
```

Step

```
<PostProcess name="stats">
  <Input class='DataObjects' type='PointSet'>solns</Input>
  <Model class="Models" type="PostProcessor">stats</Model>
  <Output class="Files" type="">statsfile</Output>
</PostProcess>
```

Use results from file TODO to pick PCA truncation

Methods: PCA in RAVEN

Once reduction level is decided, change sampler truncation and distribution truncation

```
<Sobol name='sobolsamp'>
  <variablesTransformation distribution="mvn">
    <manifestVariables>x1,x2,x3,x4,x5,x6</manifestVariables>
    <latentVariables>y1,y2,y3,y4</latentVariables>
    <method>pca</method>
  </variablesTransformation>
  <variable name='y1'>
    <distribution dim='1'>mvn</distribution>
  </variable>
  <variable name='y2'>
    <distribution dim='2'>mvn</distribution>
  </variable>
  <variable name='y3'>
    <distribution dim='3'>mvn</distribution>
  </variable>
  <variable name='y4'>
    <distribution dim='4'>mvn</distribution>
  </variable>
  <ROM class='Models' type='ROM'>rom</ROM>
</Sobol>
```

Methods: PCA in RAVEN

Once reduction level is decided, change sampler truncation and distribution truncation

```
<MultivariateNormal method="pca" name="mvn">
  <mu>0.2 0.3 0.4 0.5 0.6 0.7</mu>
  <covariance>
    0.02 0.80 0.90 0.95 0.96 0.97
    0.80 0.03 0.02 0.00 0.00 0.00
    0.90 0.02 0.04 0.00 0.00 0.00
    0.95 0.00 0.00 0.05 0.90 0.93
    0.96 0.00 0.00 0.90 0.06 0.02
    0.97 0.00 0.00 0.93 0.02 0.07
  </covariance>
  <transformation>
    <rank>4</rank>
  </transformation>
</MultivariateNormal>
```

Methods: Global Sensitivity Analysis

Now that we've reduced the input, we can do global sensitivity analysis

- Pearson Correlation Coefficients
- Spearman Rank Coefficients
- Sobol Sensitivity Coefficients

Pearson and Spearman can be sampled using forward samplers

Linear Sobol expansion often more efficient (1 run per latent variable)

Methods: Global Sensitivity Analysis

Pearson Coefficients

- Global correlation between dimensions
- Can be input-input, input-output, or output-output
- Can be calculated using most RAVEN Samplers

Methods: Global Sensitivity Analysis

More on Sobol sampler and HDMRRom in Collocation workshop!

Sobol sampler

```
<Sobol name='sobolsamp'>
  <variablesTransformation distribution="mvn">
    <manifestVariables>x1,x2,x3,x4,x5,x6</manifestVariables>
    <latentVariables>y1,y2,y3,y4</latentVariables>
    <method>pca</method>
  </variablesTransformation>
  <variable name='y1'>
    <distribution dim='1'>mvn</distribution>
  </variable>
  <variable name='y2'>
    <distribution dim='2'>mvn</distribution>
  </variable>
  <variable name='y3'>
    <distribution dim='3'>mvn</distribution>
  </variable>
  <variable name='y4'>
    <distribution dim='4'>mvn</distribution>
  </variable>
  <ROM class='Models' type='ROM'>rom</ROM>
</Sobol>
```

Methods: Global Sensitivity Analysis

HDMRRom

```xml
<ROM name="rom" subType="HDMRRom">
  <SobolOrder>1</SobolOrder>
  <Target>ans</Target>
  <Features>y1,y2,y3,y4</Features>
  <IndexSet>TensorProduct</IndexSet>
  <PolynomialOrder>2</PolynomialOrder>
</ROM>
```

ROM Output

```xml
<Print name="sobol_stats">
  <type>xml</type>
  <source>rom</source>
  <what>all</what>
</Print>
```

Methods: Global Sensitivity Analysis

Steps

```
<Steps>
  <MultiRun name="sample" sleepTime="1e-5">
    <Input class="DataObjects" type="Point">inputPlaceHolder</Input>
    <Model class="Models" type="ExternalModel">atten</Model>
    <Sampler class="Samplers" type="MonteCarlo">sobolsamp</Sampler>
    <Output class="DataObjects" type="PointSet">solns</Output>
  </MultiRun>
  <RomTrainer name="train">
    <Input class="DataObjects" type="PointSet">solns</Input>
    <Output class="Models" type="ROM">rom</Output>
  </RomTrainer>
  <IOStep name="dump">
    <Input class="DataObjects" type="PointSet">solns</Input>
    <Output class="OutStreams" type="Print">solns_dump_sobol</Output>
  </IOStep>
  <IOStep name="stats">
    <Input class="Models" type="ROM">rom</Input>
    <Output class="OutStreams" type="Print">sobol_stats</Output>
  </IOStep>
</Steps>
```

Methods: Global Sensitivity Analysis

Output from ROM


```xml
<ReducedOrderModel>
  <ans>
    <mean>0.781756851724</mean>
    <variance>0.275604991951</variance>
    <numRuns>9</numRuns>
    <indices>
      <tot_variance>0.275604991951</tot_variance>
      <variables>y1
        <partial_variance>0.264042832714</partial_variance>
        <Sobol_index>0.958048077595</Sobol_index>
      </variables>
      <variables>y2
        <partial_variance>0.00984470571011</partial_variance>
        <Sobol_index>0.035720346139</Sobol_index>
      </variables>
      <variables>y4
        <partial_variance>0.00157893548534</partial_variance>
        <Sobol_index>0.00572897999474</Sobol_index>
      </variables>
      <variables>y3
        <partial_variance>0.000138518041296</partial_variance>
        <Sobol_index>0.000502596271263</Sobol_index>
      </variables>
    </indices>
  </ans>
</ReducedOrderModel>
```

Methods: Twice-Reduced Analysis

Change dimensions in Sampler to prioritize

```
<MonteCarlo name='mcsamp'>
  <samplerInit>
    <limit>100</limit>
  </samplerInit>
  <variablesTransformation distribution="mvn">
    <manifestVariables>x1,x2,x3,x4,x5,x6</manifestVariables>
    <latentVariables>y1,y2,y4</latentVariables>
    <method>pca</method>
  </variablesTransformation>
  <variable name='y1'>
    <distribution dim='1'>mvn</distribution>
  </variable>
  <variable name='y2'>
    <distribution dim='2'>mvn</distribution>
  </variable>
  <variable name='y4'>
    <distribution dim='4'>mvn</distribution>
  </variable>
</MonteCarlo>
```

Methods: Twice-Reduced Analysis

Now what?

- Make a ROM
- Perform surrogate statistics sampling
- Compute limit surface on surrogate

Full Example

Example case: Cross Section Model

- 308 Correlated Cross Sections (SCALE)
- Simulation model is polynomial combinations
- PCA Reduction: 308 to 50
- Sensitivity Reduction: 50 to 9

See included examples:

- Original Monte Carlo benchmark: `run_mc_orig.xml`
- PCA reduction: `run_mc_pca.xml`
- PCA output: `first/TODO`
- Sensitivity reduction: `run_mc2_[#].xml`
- Sensitivity output: `first/sobol_dump.xml`
- ROM with twice-reduced space: `sc_td[#].xml`
- Twice-reduced output: `td_[#]_dump.xml`

Full Example: Mean Results

Original, PCA (50 terms), PCA (no sens.), PCA and Sensitivity

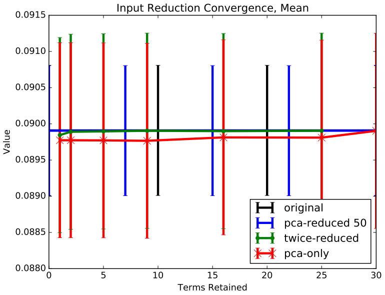


Figure: Mean Values by Terms Kept

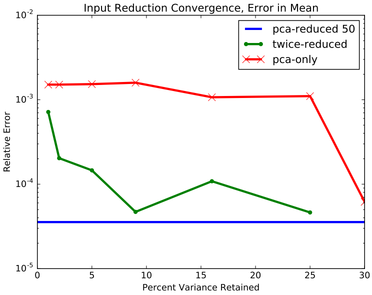


Figure: Mean Errors by Terms Kept

Full Example: Variance Results

Original, PCA (50 terms), PCA (no sens.), PCA and Sensitivity

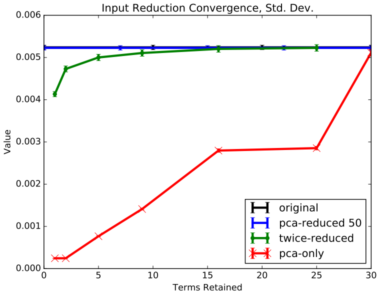


Figure: Variance Values by Terms Kept

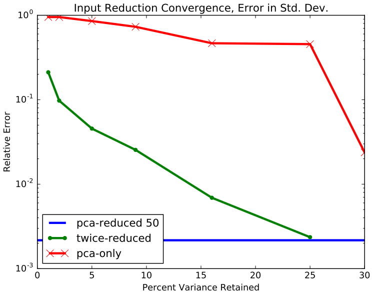


Figure: Variance Errors by Terms Kept

Full Example: Number of Runs to Mean

Monte Carlo (10k samples), Monte Carlo, Sparse Grid, Adaptive Sobol

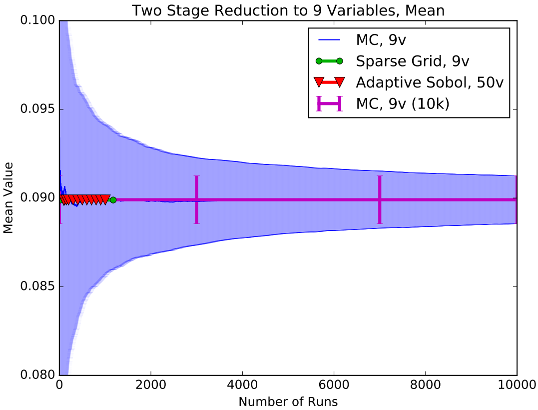


Figure: Runs to Convergence, Mean, Twice-Reduced

Full Example: Number of Runs to Mean

Monte Carlo (10k samples), Monte Carlo, Sparse Grid, Adaptive Sobol

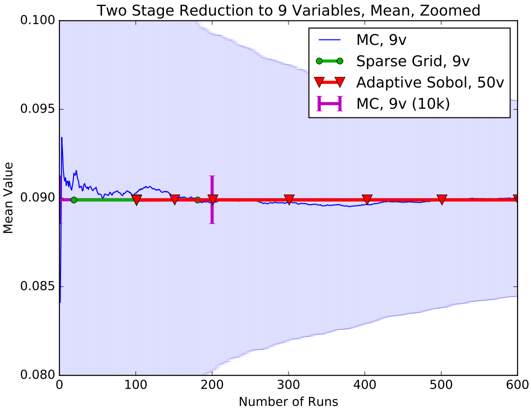


Figure: Runs to Convergence, Mean, Twice-Reduced

Full Example: Number of Runs to Std. Dev.

Monte Carlo (10k samples), Monte Carlo, Sparse Grid, Adaptive Sobol

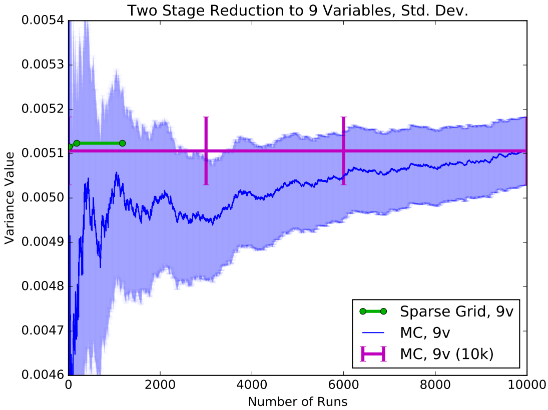


Figure: Runs to Convergence, Std. Dev., Twice-Reduced

Full Example: Number of Runs to Std. Dev.

Monte Carlo (10k samples), Monte Carlo, Sparse Grid, Adaptive Sobol

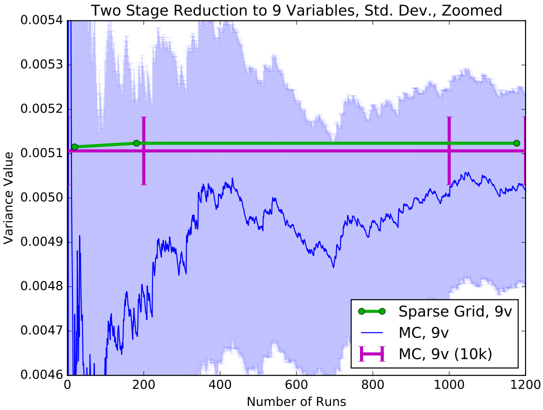


Figure: Runs to Convergence, Std. Dev., Twice-Reduced

Full Example: Convergence to Original Model

Monte Carlo (10k samples), Monte Carlo, Sparse Grid, Adaptive Sobol

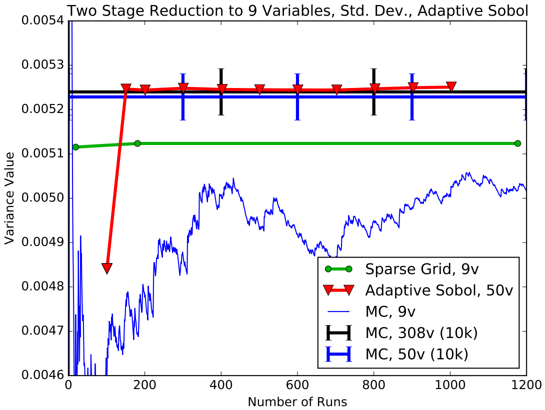


Figure: Convergence to Original Model

End of Session