

# Advanced Methods in Stochastic Collocation for Polynomial Chaos in RAVEN

Paul W. Talbot

University of New Mexico

Dissertation Proposal, January 22nd 2016

Funded by a Laboratory Directed Research effort at Idaho  
National Laboratory



# Outline

## Discussion Points

- 1 Background
  - Terminology
  - Raven
  - Intrusiveness
  - Sampling Strategies
- 2 Methods
  - UQ Methods
  - Physics Models
- 3 Proposal
  - HDMR
  - Adaptive HDMR
  - Mammoth

# Outline

- 1 Background
  - Terminology
  - Raven
  - Intrusiveness
  - Sampling Strategies

2 Methods

3 Proposal

# Background

## Terminology

- Model - Mathematical representation
  - Simulation/Code - Algorithms to Solve Model
  - Output - Solution to Model
- 
- Input Space - Space Spanning Possible Input Values
  - Response - Map of Input Space to Output Space

# Background

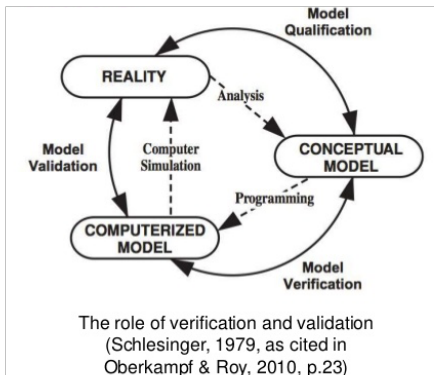
## UQ

### Understanding Simulations

- Verification - Is the code consistent?
  - Spatial, Temporal Convergence
- Validation - Does it match experiment?
  - Fitting, Extrapolation
- Uncertainty Quantification - Behavior of Response
  - Response Surface Characterization

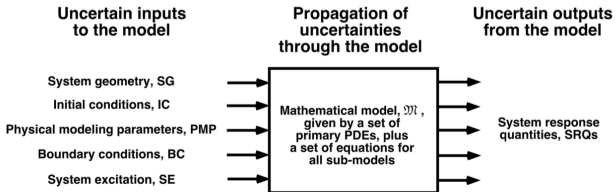
# Background

## Verification and Validation



# Background

## Uncertainty Quantification



Oberkampf and Roy, 2010

# Background

## RAVEN

- (R)isk (A)nalysis (V)irtual (EN)vironment
- Verification and Validation
- Uncertainty Quantification
- Response Surfaces, Risk
- Reduced-Order Models
- Operates on Black-box Models

Goal: Add New UQ Methods to RAVEN



# Background

## Intrusiveness

Two approaches to UQ:

- Intrusive

- Interacts with Simulation Algorithms
- Examples: Stochastic Galerkin, Adjoint-Based

- Non-Intrusive

- Agnostic of Code
- Example: Monte Carlo, Stochastic Collocation

For RAVEN, need non-intrusive algorithms

# Sampling Strategies

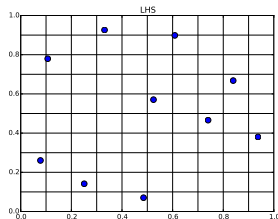
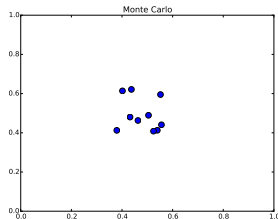
## Non-Intrusive Sampling Strategies

### Monte Carlo

- Generate independent, identically-distributed samples
- Derive mean, variance, response surface

### Latin Hypercube

- Distributed Monte Carlo



# Sampling Strategies

## Limitations

Monte Carlo common choice for UQ

- Pro: Convergence agnostic of dimensionality
- Con: Convergence can take many, many samples

# Outline

1 Background

2 Methods

- UQ Methods
- Physics Models

3 Proposal

# Definitions

- $N \in \mathbb{N}^+$  - Cardinality of Input Space
- Probability Space  $(\Omega, \Theta, \rho)$ ,  $Y : \Theta \rightarrow \mathbb{R}^N$
- $Y = (y_1, \dots, y_N)$  - Uncertain Input Parameter Vector
- Subspaces  $(\Omega_n, \Theta_n, \rho_n)$
- $Y \in \mathbb{R}^N$ ,  $y_n \in \mathbb{R} \forall n \leq N, n \in \mathbb{N}^+$
- Members of  $Y$  are uncorrelated
- $\rho(Y) = \prod_{n=1}^N \rho_n(y_n)$  - Joint Probability Density Function
- $u(Y)$  - QoI as a function of Input Parameters  $Y$

# Generalized Polynomial Chaos

## gPC

$$u(Y) = \sum_{k \in \Lambda} c_k \Phi_k(Y),$$

- $\Lambda$  - a finite set of polynomial indices,
- $k = (k_1, \dots, k_N)$  indexes  $\Lambda$ ,  $k_n \in \mathbb{N}^0$ ,
- $c_k \in \mathbb{R}$  - scalar coefficients,
- $\phi_{k_n}(y_n)$  - Gauss polys order  $k_n$  orthonormal w.r.t  $\rho_n(y_n)$ ,
- $\Phi_k(Y) = \prod_{n=1}^N \phi_{k_n}(y_n)$ ,
- $\Phi_k(Y)$  orthonormal w.r.t  $\rho(Y)$  over  $\Omega$ .

# Generalized Polynomial Chaos

## Truncated

In practice, truncate to some gPC level  $L \in \mathbb{N}^+$ ,

$$u(Y) \approx G_L[u](Y) \equiv \sum_{k \in \Lambda(L)} c_k \Phi_k(Y).$$

Error is of order equal to the polynomial terms removed by truncation.

# Generalized Polynomial Chaos

## Constructing $\Lambda(L)$

$\Lambda(L)$  is list of multidimensional polynomial indices  $k$

For example,  $N = 3$ :

$$k = (3, 1, 2) \rightarrow \Phi_k(Y) = \phi_3(y_1)\phi_1(y_2)\phi_2(y_3).$$



# Generalized Polynomial Chaos

## Constructing $\Lambda(L)$

Naïve choice for  $\Lambda(L)$  is tensor product of all polynomials of order less than or equal to  $L$ ,

$$\Lambda_{\text{TP}}(L) = \left\{ k = (k_1, \dots, k_N) : \max_{1 \leq n \leq N} k_n \leq L \right\}.$$

Size of Index Set:  $|\Lambda_{\text{TP}}(L)| = (L + 1)^N$ .

(3,0)	(3,1)	(3,2)	(3,3)
(2,0)	(2,1)	(2,2)	(2,3)
(1,0)	(1,1)	(1,2)	(1,3)
(0,0)	(0,1)	(0,2)	(0,3)

Table: Tensor Product Index Set,  $N = 2, L = 3$

# Generalized Polynomial Chaos

## Constructing $\Lambda(L)$

Common choice for  $\Lambda(L)$ : Total Degree of all polynomials of order less than or equal to  $L$ ,

$$\Lambda_{\text{TD}}(L) = \left\{ \bar{p} = (p_1, \dots, p_N) : \sum_{n=1}^N p_n \leq L \right\}.$$

Size of Index Set:  $|\Lambda_{\text{TD}}(L)| = \binom{L+N}{N}$ .

(3,0)

(2,0)    (2,1)

(1,0)    (1,1)    (1,2)

(0,0)    (0,1)    (0,2)    (0,3)

**Table:** Total Degree Index Set,  $N = 2, L = 3$

# Generalized Polynomial Chaos

## Constructing $\Lambda(L)$

Hyperbolic Cross: Limited by product of all polynomial orders,

$$\Lambda_{\text{HC}}(L) = \left\{ \bar{p} = (p_1, \dots, p_N) : \prod_{n=1}^N p_n + 1 \leq L + 1 \right\}.$$

Index Set Bound:  $|\Lambda_{\text{HC}}(L)| \leq (L + 1)(1 + \log(L + 1))^{N-1}.$

(3,0)

(2,0)

(1,0) (1,1)

(0,0) (0,1) (0,2) (0,3)

Table: Hyperbolic Cross Index Set,  $N = 2, L = 3$

# Generalized Polynomial Chaos

## Constructing $\Lambda(L)$

Anisotropic Rules using weights  $\alpha = (\alpha_1, \dots, \alpha_N)$

$$\tilde{\Lambda}_{\text{TD}}(L) = \left\{ \bar{p} = (p_1, \dots, p_N) : \sum_{n=1}^N \alpha_n p_n \leq |\alpha|_1 L \right\},$$

$$\tilde{\Lambda}_{\text{HC}}(L) = \left\{ \bar{p} = (p_1, \dots, p_N) : \prod_{n=1}^N (p_n + 1)^{\alpha_n} \leq (L + 1)^{|\alpha|_1} \right\}.$$

$$|\alpha|_1 \equiv \frac{\sum_{n=1}^N \alpha_n}{N}.$$

# Generalized Polynomial Chaos

## Sparse Grid Quadrature

Necessary to find coefficients  $c_k$ ,

$$G[u](Y) = \sum_{k \in \Lambda(L)} c_k \Phi_k(Y).$$

Using orthonormality of  $\Phi_k(Y)$ ,

$$c_k = \langle u(Y) \Phi_k(Y) \rangle = \int_{\Omega} \rho(Y) u(Y) \Phi_k(Y) dY$$

# Generalized Polynomial Chaos

## Sparse Grid Quadrature

The quadrature and polynomials to use depends on the distribution  $\rho_n(y_n)$ ,

PDF Kind	Polynomials / Quadrature
Uniform	Legendre
Normal	Hermite
Gamma	Laguerre
Beta	Jacobi

# Generalized Polynomial Chaos

## Order of Quadrature

Determine order of quadrature

$$\int_{\Omega_x} f(x) \rho(x) dx = \sum_{\ell=1}^{\infty} w_{\ell} f(x_{\ell}).$$

If  $f(x)$  is a polynomial of order  $2m - 1$ ,

$$\int_{\Omega_x} f(x) \rho(x) dx = \sum_{\ell=1}^m w_{\ell} f(x_{\ell}).$$

Let

$$q^m[f(x)] \equiv \sum_{\ell=1}^m w_{\ell} f(x_{\ell}).$$

In our case (let  $N=1$ ),

$$c_k = \int u(Y) \Phi(Y) \rho(Y) dY = q^m[u(Y) \Phi(Y)].$$

# Generalized Polynomial Chaos

## Order of Quadrature

$$c_k = \int_{\Omega} u(Y) \Phi(Y) \rho(Y) dY = q^m [u(Y) \Phi_k(Y)].$$

To determine  $m$ , replace  $u(Y) = G[u](Y)$ :

$$\begin{aligned} c_k &= q^m \left[ \Phi_k(Y) \sum_{k' \in \Lambda(L)} c_{k'} \Phi_{k'}(Y) \right], \\ &= q^m \left[ c_k \Phi_k(Y)^2 \right], \\ &= q^m \left[ \mathcal{O}(k^2) \right], \end{aligned}$$

$$2m + 1 = 2k \quad \rightarrow \quad m = k + \frac{1}{2} \approx m = k + 1.$$



# Generalized Polynomial Chaos

## Sparse Grid Quadrature

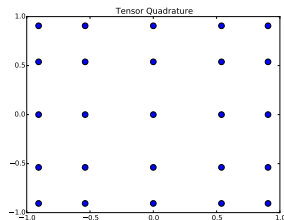
### Multidimensional Quadrature

- Each point needs  $N$  quadrature sets
- Ex.  $N = 2$ ,  $k=(4,4)$ , need (5,5) quadratures
- Needed for each  $k$  in  $\Lambda$

### Tensor quadrature

$$Q^M \equiv q_1^{m_1} \otimes q_2^{m_2} \otimes \cdots \otimes q_N^{m_N},$$

$$= \bigotimes_{n=1}^N q_n^{m_n}.$$



# Generalized Polynomial Chaos

## Sparse Grid Quadrature

Smolyak sparse grids  $S[u](Y)$ :

$$\text{Let } \Delta_k^m[f(x)] \equiv (q_k^m - q_{k-1}^m)[f(x)].$$

$$S_{\Lambda, N}^M[u(Y)\Phi_k(Y)] = \sum_{k \in \Lambda(L)} \left( \Delta_{k_1}^{m_1} \otimes \cdots \otimes \Delta_{k_N}^{m_N} \right) [u(Y)\Phi_k(Y)].$$

# Generalized Polynomial Chaos

## Sparse Grid Quadrature

Equivalently,

$$S_{\Lambda,N}^M[u(Y)\Phi_k(Y)] = \sum_{k \in \Lambda(L)} s(k) \bigotimes_{n=1}^N q_n^{m_n}[u(Y)\Phi_k(Y)],$$

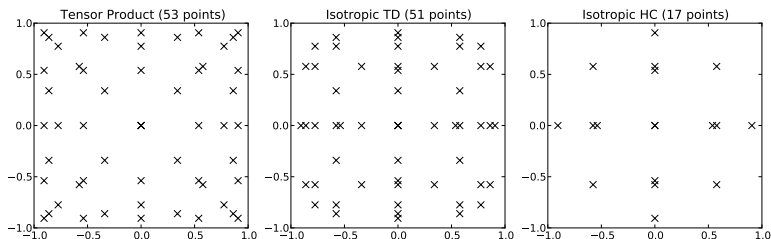
$$s(k) = \sum_{\substack{j=\{0,1\}^N, \\ k+j \in \Lambda}} (-1)^{|j|_1}, \quad |j|_1 \equiv \sum_{p \in j} p.$$

$$\begin{aligned} c_k &= \langle u(Y)\Phi_k(Y) \rangle, \\ &\approx S_{\Lambda,N}^M[u(Y)\Phi_k(Y)]. \end{aligned}$$

# Generalized Polynomial Chaos

## Sparse Grid Quadrature

Comparison,  $N = 2$ ,  $L = 4$



# Generalized Polynomial Chaos

## Adaptive Index Sets

Special property of gPC:

$$G[u](Y) = \sum_{k \in \Lambda} c_k \Phi_k(Y),$$

$$\text{mean} = \langle G[u](Y) \rangle = c_{\{0\}^N},$$

$$\begin{aligned} \text{var} &= \langle G[u](Y)^2 \rangle - c_{\{0\}^N}^2, \\ &= \sum_{k \in \Lambda} c_k^2 - c_{\{0\}^N}^2. \end{aligned}$$

# Generalized Polynomial Chaos

## Adaptive Index Sets

Adaptive Index Set construction using ANOVA

$$\text{Impact Parameter } \eta_k = \frac{c_k^2}{\text{var} \{G[u](Y)\}}, \quad 0 \leq \eta_k \leq 1$$

$$\text{Est. Impact Parameter } \tilde{\eta}_k = \prod_{n=1, k_n > 0}^N \eta_{(k_1, \dots, k_{n-1}, \dots, k_N)}$$

Example,

$$\tilde{\eta}_{(2,1,3)} = \eta_{(1,1,3)} \cdot \eta_{(2,0,3)} \cdot \eta_{(2,1,2)}$$

# Generalized Polynomial Chaos

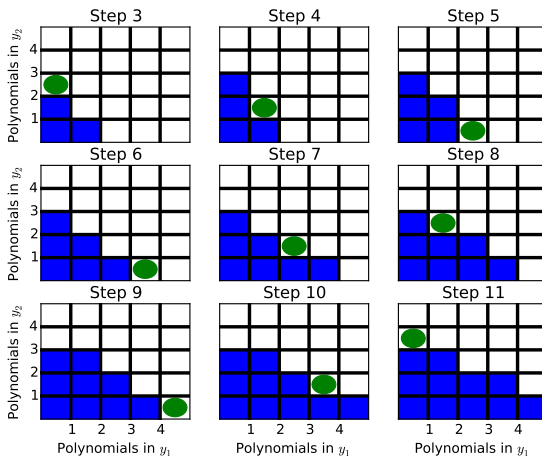
## Adaptive Index Sets

### Adaptive Index Set Algorithm

- Calculate mean (zeroth-order) polynomial expansion
- While not converged:
  - Collect list of indices whose predecessors are in  $\Lambda$
  - Using existing impacts, predict impact of each potential index
  - If total of impacts is less than tolerance, convergence is reached
  - Otherwise, add highest-impact index and construct new expansion

# Generalized Polynomial Chaos

## Adaptive Index Sets





# Physical Models

Efficiency of gPC compared to Monte Carlo depends on

- Regularity of response surface
- Dimensionality of input space

Demonstrate using three models

- Tensor Polynomial
- Attenuation
- Neutron Diffusion

# Physical Models

## Tensor Polynomial

From Ayres and Eaton, 2015

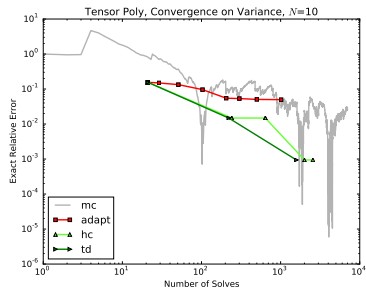
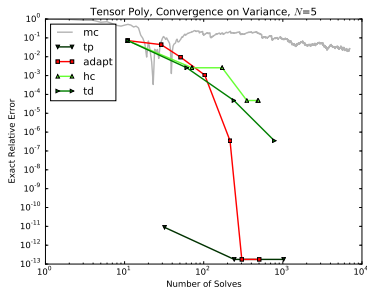
$$u(Y) = \prod_{n=1}^N (y_n + 1), \quad y_n \sim \mathcal{U}[-1, 1]$$

Features:

- Analytic mean, variance
- Regularity
- Exact in finite polynomial space

# Tensor Polynomials

## Results



# Physical Models

## Attenuation

Exit strength of beam incident on discrete absorbing material

$$u(Y) = \prod_{n=1}^N e^{-y_n/N}, \quad y_n \sim \mathcal{U}[0, 1]$$

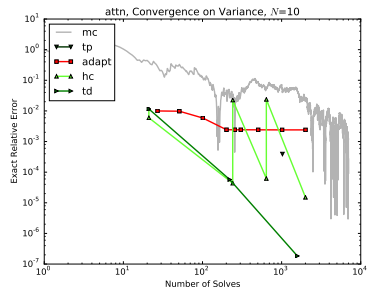
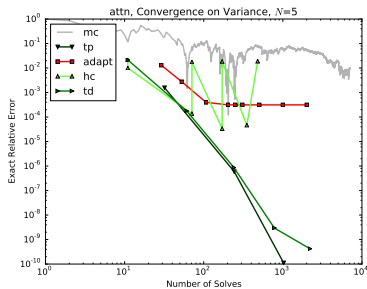
Features:

- Analytic mean, variance
- Regularity
- Inexact in finite polynomial space

Taylor expansion is tensor high-order polynomials

# Attenuation

## Results



# Physical Models

## Neutron Diffusion

Quarter-core 2-group steady state reactor benchmark, 165x165 cm

$$-\nabla \cdot (D_1(\bar{x}) \nabla \phi_1(\bar{x})) + \left( \Sigma_a^{(1)}(\bar{x}) + \Sigma_s^{(1 \rightarrow 2)}(\bar{x}) \right) \phi_1(\bar{x}) = \frac{1}{k(\phi)} \sum_{g'=1}^2 \nu_{g'} \Sigma_f^{(g')}(\bar{x}) \phi_{g'}(\bar{x}),$$

$$-\nabla \cdot (D_2(\bar{x}) \nabla \phi_2(\bar{x})) + \Sigma_a^{(2)}(\bar{x}) \phi_2(\bar{x}) = \Sigma_s^{(1 \rightarrow 2)}(\bar{x}) \phi_1(\bar{x})$$

QoI:  $k(\phi)$

# Neutron Diffusion

## Geometry

5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5
3	3	3	3	3	3	3	5	5	5	5
3	3	3	3	3	3	3	4	5	5	5
2	1	1	1	1	2	2	3	3	5	5
2	1	1	1	1	2	2	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
1	1	1	1	1	1	1	3	3	5	5
2	1	1	1	1	2	2	3	3	5	5

Vacuum, Reflective boundaries

# Neutron Diffusion

## BCs

### Boundary Conditions

$$\left. \frac{\phi_g}{4} - \frac{D_g}{2} \frac{\partial \phi_g}{\partial x_1} \right|_{\partial \Omega_{\text{top}}} = 0, \quad g = 1, 2, \quad (1)$$

$$\left. \frac{\phi_g}{4} - \frac{D_g}{2} \frac{\partial \phi_g}{\partial x_2} \right|_{\partial \Omega_{\text{right}}} = 0, \quad g = 1, 2, \quad (2)$$

$$\left. -D_g \frac{\partial \phi_g}{\partial x_1} \right|_{\partial \Omega_{\text{bottom}}} = 0, \quad g = 1, 2, \quad (3)$$

$$\left. -D_g \frac{\partial \phi_g}{\partial x_2} \right|_{\partial \Omega_{\text{left}}} = 0, \quad g = 1, 2. \quad (4)$$



# Neutron Diffusion

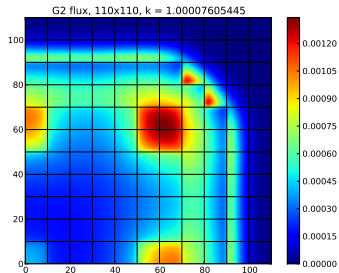
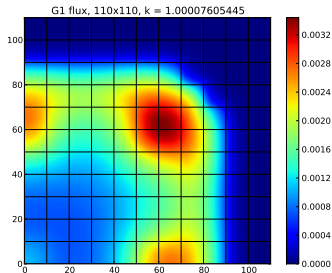
## Input Space

Region	Group	$D_g$	$\Sigma_{a,g}$	$\nu\Sigma_{f,g}$	$\Sigma_s^{1,2}$
1	1	1.255	8.252e-3	4.602e-3	2.533e-2
	2	2.11e-1	1.003e-1	1.091e-1	
2	1	1.268	7.181e-3	4.609e-3	2.767e-2
	2	1.902e-1	7.047e-2	8.675e-2	
3	1	1.259	8.002e-3	4.663e-3	2.617e-2
	2	2.091e-1	8.344e-2	1.021e-1	
4	1	1.259	8.002e-3	4.663e-3	2.617e-2
	2	2.091e-1	7.3324e-2	1.021e-1	
5	1	1.257	6.034e-4	0	4.754e-2
	2	1.592e-1	1.911e-2	0	

# Neutron Diffusion

## Reference Solution

$$k=1.00007605445$$



# Neutron Diffusion

## Uncertainty

5% Uncertainty in Parameters:

$N=3, N=5$

■  $\nu\Sigma_{2,f}^{(1)}$

■  $\nu\Sigma_{2,f}^{(4)}$

■  $D_2^{(5)}$

■  $\Sigma_{2,c}^{(1)}$

■  $\Sigma_{2,c}^{(4)}$

$N=10$

■  $\Sigma_{2,c}^{(1)}, \nu\Sigma_{2,f}^{(1)}$

■  $\Sigma_{2,c}^{(2)}, \nu\Sigma_{2,f}^{(2)}$

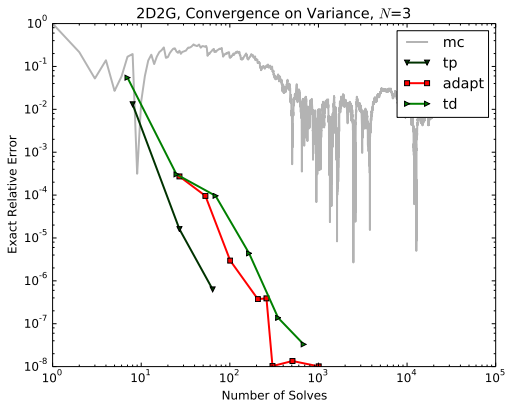
■  $\Sigma_{2,c}^{(3)}, \nu\Sigma_{2,f}^{(3)}$

■  $\Sigma_{2,c}^{(4)}, \nu\Sigma_{2,f}^{(4)}, \nu\Sigma_{1 \rightarrow 2}^{(4)}$

■  $D_2^{(5)}$

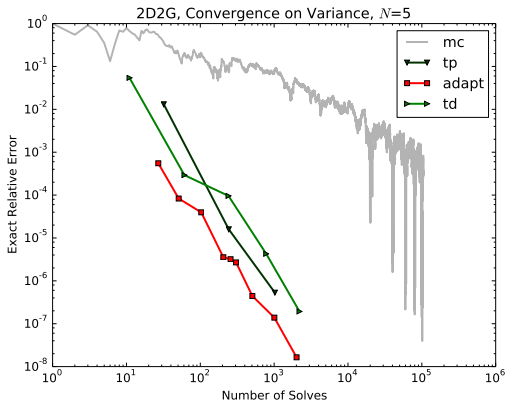
# Neutron Diffusion

## Results

 $N=3$ 

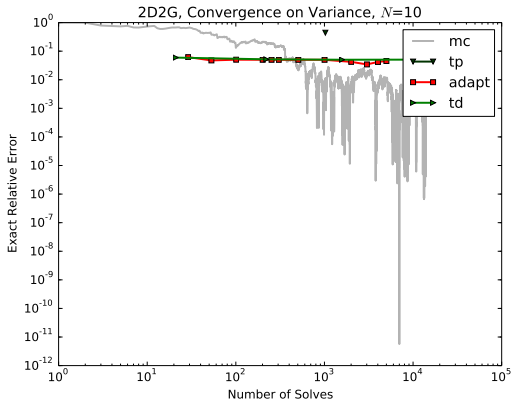
# Neutron Diffusion

## Results

 $N=5$ 

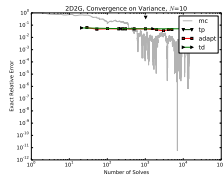
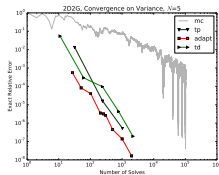
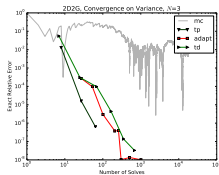
# Neutron Diffusion

## Results

 $N=10$ 

# Neutron Diffusion

## Results



# Results

## Summary

### Preliminary Conclusions

- SC for gPC suffers greatly from curse of dimensionality
- Adaptive SC can help
- For 5 or less dimensions, performs well for regular responses



# Outline

1 Background

2 Methods

3 Proposal

- HDMR
- Adaptive HDMR
- Mammoth

# Proposal

## HDMR

ANOVA method to decompose input space

$$\tilde{H}[u](Y) = u_0 + \sum_{i=1}^N u_i + \sum_{i_1=1}^N \sum_{i_2=1}^{i_1-1} u_{ij} + \cdots + c_{i_1, \dots, i_N},$$

$$u_0 \equiv \int dy_1 \cdots \int dy_N u(Y),$$

$$u_i \equiv \int dy_1 \cdots \int dy_{i-1} \int dy_{i+1} \cdots \int dy_N u(Y) - u_0$$

# Proposal

## HDMR

Sobol' sensitivity coefficients

$$\mathcal{S}_i = \frac{\text{var} \{u_i\}}{\text{var} \left\{ \tilde{H}[u](Y) \right\}}$$

$$\text{var} \{u(Y)\} = \sum_{i=1}^N \mathcal{S}_i + \sum_{i_1=1}^N \sum_{i_2=1}^{i_1-1} \mathcal{S}_{ij} + \cdots + \mathcal{S}_{i_1, \dots, i_N}$$

# Proposal

## HDMR

Problem: Costly integrals (even low-level expansion)

Solution: Cut-HDMR, reference point  $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_N)$

Let  $u(y_i) \equiv u(\hat{y}_1, \dots, \hat{y}_{i-1}, y_i, \hat{y}_{i+1}, \dots, \hat{y}_N)$

$$H[u](Y) = h_0 + \sum_{i=1}^N h_i + \sum_{i_1=1}^N \sum_{i_2=1}^{i_1-1} h_{ij} + \dots + h_{i_1, \dots, i_N},$$

$$h_0 \equiv u(\hat{Y}),$$

$$h_i \equiv u(y_i) - u_0,$$

$$h_{ij} \equiv u(y_i, y_j) - u_i - u_j - u_0$$

# Proposal

## HDMR

Problem with Cut-HDMR: Terms not orthogonal

$$S_i \neq \frac{\text{var} \{h_i\}}{\text{var} \{H[u](Y)\}} \equiv \xi_{ij}$$

$$\text{var} \{u(Y)\} \neq \sum_{i=1}^N \xi_i + \sum_{i_1=1}^N \sum_{i_2=1}^{i_1-1} \xi_{ij} + \cdots + \xi_{i_1, \dots, i_N}$$

However, no integrals!

# Proposal

## HDMR

Next step: HDMR terms are SCgPC

$$H[u](Y) = h_0 + \sum_{i=1}^N h_i + \sum_{i_1=1}^N \sum_{i_2=1}^{i_1-1} h_{ij} + \cdots + h_{i_1, \dots, i_N},$$

$$h_0 \equiv u(\hat{Y}),$$

$$h_i \equiv G[u](y_i) - u_0,$$

$$h_{ij} \equiv G[u](y_i, y_j) - u_i - u_j - u_0$$

Benefit: most terms low dimension, regular

# Proposal

## Adaptive HDMR

Better than HDMR: Adaptive HDMR with Adaptive SCgPC

Choose terms based on impact parameters  $\xi, \eta$

- Evaluate the reference (all mean) case.
- Construct all HDMR first-order SCgPC models.
- While not converged:
  - Using existing subset impacts, predict importance of future subsets
  - Consider impact of adding polynomials to existing subset gPCs
  - Choose: expand existing subsets or add new subsets
  - If contribution of new HDMR expectation less than tolerance, convergence

# Proposal

## Adaptive HDMR

Example,  $N = 3$

$H[u](Y)$	$=$	$h_0 +$	$h_1 +$	$h_2 +$	$h_3 +$	$h_{12} +$	$h_{13} +$	$h_{23} +$	$h_{123}$
		0	1	1	1				
SCgPC Order:		0	2	1	1				
		0	2	1	1	1			
		0	2	2	1	1			
		0	2	2	2	1			
		0	2	2	2	1	1		



# Proposal

## Mammoth

### Mammoth: MOOSE framework MultiApp



- RattleSnake: Neutron Transport code
  - Provides power distributions
  - Tens of thousands of inputs
- Bison: Fuel Performance code
  - Analyzes fuel: stress, temperature, displacement
  - Dozens of inputs

Goal: Demonstrate Adaptive HDMR with Adaptive SCgPC

# Proposal

## Special Thanks

Dept. Nuclear Engineering, UNM

- Dr. Anil Prinja, Adviser and Chair
- Aaron Olsen, Peer

RAVEN team, Idaho National Laboratory

- Dr. Cristian Rabiti, Manager
- Andrea Alfonsi, Mentor
- Daniel Maljovec, Peer

Research performed under LDRD grant through Idaho National  
Laboratory