High Density Model Reduction Uncertainty Quantification for Multigroup Diffusion Neutronics

Paul Talbot*and Anil K. Prinja

Department of Chemical and Nuclear Engineering
University of New Mexico

November 4, 2014

*talbotp@unm.edu

1 Introduction

There have been many recent advances in uncertainty quantification in numerical models for computational physics [1]. In this document, we present ongoing work to demonstrate the efficiency of high-density model reduction (HDMR) techniques as well as sparse-grid stochastic collocation on sparse grids (SC) [6] in comparison with traditional analog Monte Carlo (MC) and Latin Hypercubefor (LHC) uncertainty quantification methods.

2 Model

The uncertainty quantification methods we describe are agnostic of the deterministic problem whose uncertainty is quantified. To demonstrate the effectiveness of these methods, we apply uncertainty quantification to a two-dimensional nonlinear dual-energy group neutron diffusion criticality benchmark problem [4]. This problem simulates an orthogonal quarter-core reactor by imposing reflective conditions on the left and bottom boundaries, with vacuum no-traction conditions on the top and right. the problem makes use of five materials in 121 regions. Fig. 1 illustrates the core.

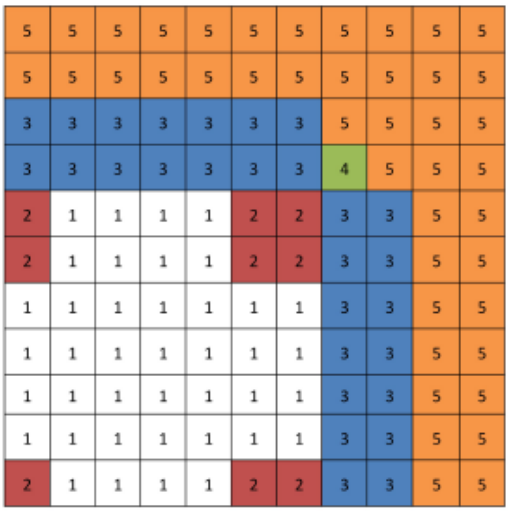


Figure 1: Core Geometry

2.1 Mathematical Model

The PDE describing steady-state neutron transport in the core is the two-group neutron diffusion criticality approximation,

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

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

where we use the following parametric coefficients: the absorption cross section $\Sigma_{g,a}=\Sigma_{g,c}+\Sigma_{g,f}$; the capture and fission cross sections $\Sigma_{g,c}$ and $\Sigma_{g,f}$; the diffusion coefficient D_g which depends on the scattering cross section of the medium; and the fission multiplication factor ν_g, the ratio of new neutrons per fission-producing neutron. The solution to this PDE is the neutron scalar flux

$\phi_g(\bar{x})$. We apply no-traction conditions on the vacuum boundaries and zero-derivative current on the reflecting boundaries for both energy groups:

$$\frac{\phi_g}{4} - \frac{D_g}{2}\left.\frac{\partial \phi_g}{\partial x_1}\right|_{\partial\Omega_{\text{top}}} = 0, \quad g = 1, 2, \tag{3}$$

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

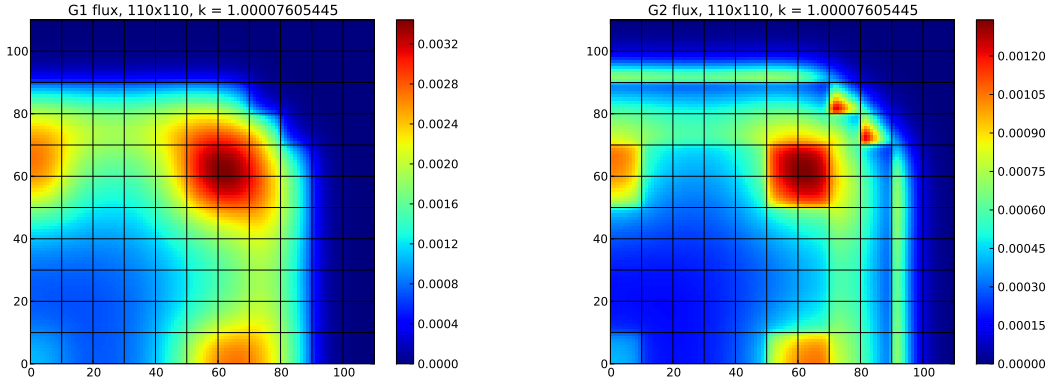
$$-D_g \left.\frac{\partial \phi_g}{\partial x_1}\right|_{\partial\Omega_{\text{bottom}}} = 0, \quad g = 1, 2, \tag{5}$$

$$-D_g \left.\frac{\partial \phi_g}{\partial x_2}\right|_{\partial\Omega_{\text{left}}} = 0, \quad g = 1, 2. \tag{6}$$

The material properties are shown in Table 1, and the domain $\Omega = [0, 200 \text{ cm}]^2$. The reference flux solutions are plotted in Fig. 2, and for the reference problem k=1.00007605445.

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	

Table 1: Reference Material Properties for Benchmark Core



(a) ϕ, Group 1 (b) ϕ, Group 2

Figure 2: Reference Flux Profiles

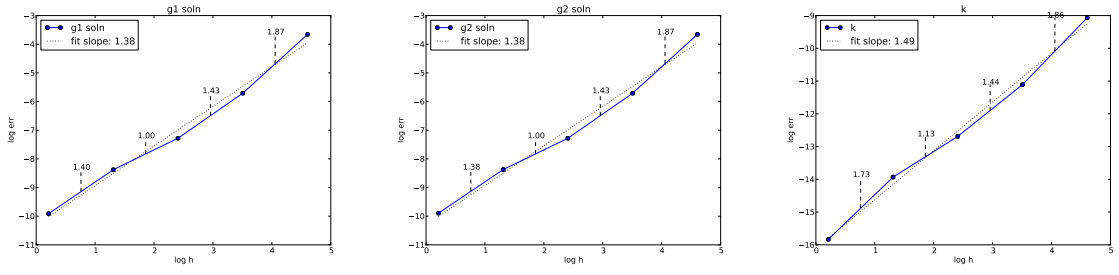
2.2 Deterministic Solver

The criticality eigenvalue and quantity of interest $k(\phi)$ is converged on iteratively by

$$k^{n+1} = k^n \sum_{g=1}^{2} \frac{\iint\limits_{\Omega} \nu\Sigma_f^{(g)} \phi_g^{n+1}(\bar{x})\ d\bar{x}}{\iint\limits_{\Omega} \nu\Sigma_f^{(g)} \phi_g^{n}(\bar{x})\ d\bar{x}}. \tag{7}$$

until a convergence tolerance is achieved. We solve ϕ_1, ϕ_2, and k nonlinearly and simultaneously using a globally- and locally-conservative finite volume approximation in space. We solve implicitly with a Jacobian-Free Newton Krylov method, using a GMRES package from nonlinear solver package `trilinos` [2].

To verify the deterministic solver, we compare flux profiles and k to the original benchmark [4] as well as demonstrate the convergence of k with increasing mesh grid refinement. The deterministic solver is in agreement with the benchmark, and convergence for k and both group fluxes are demonstrated in Fig. 3. Both fluxes as well as the k-eigenvalue converge between first and second order. The finite volume method is second order, but the overall convergence order is reduced by first-order boundary condition treatments. In all cases, reducing the size of $h = \Delta x = \Delta y$ decreases the error in the solution. The reference solution is a level of refinement finer than the finest point shown. The flux error is calculated by considering the maximum error among several pointwise flux values, while k error is the global value for each refinement.



(a) ϕ, Group 1 (b) ϕ, Group 2 (c) k

Figure 3: Deterministic Solver Convergence

3 Uncertainty Quantification

Throughout the remainder of this document, we will represent the solution to the k-eigenvalue problem generically as $u(Y)$, where u corresponds to the value of k and Y is the vector of uncertain input parameters.

There are 33 total input parameters in Table 1 that can be treated as uncertain. For simplicity, any uncertainty introduced by these parameters is taken to be uniformly distributed within 10% of the reference value. We consider three cases for uncertainty quantification, distinguished chiefly by the number of input parameters with uncertainty (N): material 1 uncertainty, where all the material properties for material 1 are treated as uncertain (N=7); low-energy group uncertainy, where all material properties for group 2 are treated as uncertain (N=13); and full uncertainty, where all

material properties are treated as uncertain (N=33).

The test for a method's effectiveness will be considering the level of error reached as a function of the number of calls to the deterministic solver required. While there is some difference in overhead cost between methods, this diminishes for sufficiently costly determinstic solutions.
In general, moments of the quantity of interest $u(Y)$ are obtained by

$$\mathbb{E}[u(Y)^r] = \int_\Gamma u(Y)^r P(Y) dY, \tag{8}$$

where r is the desired moment, $Y = [Y_1, Y_2, \ldots, Y_N]$ are the uncertain inputs, $P(Y)$ is the joint probability distribution of Y, and Γ is the uncertainty space spanned by Y.

3.1 Monte Carlo and Latin Hypercube

By way of benchmark, we consider both analog Monte Carlo (MC) and Latin Hypercube (LHC) as uncertainty quantification techniques.

3.1.1 Monte Carlo

Analog Monte Carlo determines the moments of a function by randomly sampling from the uncertainy space repeatedly until an accurate idea of the result is obtained. Monte Carlo benefits from a guaranteed consistent rate of convergence; however, this convergence rate is quite low; it typically converges as 1 over the square root of the number of deterministic solves. MC approximates moments of $u(Y)$ as

$$\mathbb{E}[u(Y)^r] \approx \frac{1}{\eta} \sum_{m=1}^{\eta} u(Y^{(m)})^r, \tag{9}$$

where η is the total number of samples taken, m is the index of a sample, and $Y^{(m)}$ is a point randomly sampled from Γ.

3.1.2 Latin Hypercube

TODO

3.2 Stochastic Collocation

In stochastic collocation, we approximate $u(Y)$ as the sum of u evaluated at η collocated points multiplied by multidimensional Lagrangian polynomials. We make use of the quadrature index k, not to be confused with the criticality factor $k(Y)$ (represented by $u(Y)$).

$$u(Y) \approx u_{h,\eta,\Lambda(L)}(Y) = \sum_{k=0}^{\eta} u(Y^{(k)}) \mathcal{L}_k(Y), \tag{10}$$

$$\mathcal{L}_k(Y) = \prod_{n=1}^{N} \mathcal{L}_{k_n}(Y_n), \tag{11}$$

$$\mathcal{L}_{k_n}(Y_n) = \prod_{j=1}^{i} \frac{Y_n - Y_n^{(i)}}{Y_n^{(k_n)} - Y_n^{(i)}}, \tag{12}$$

$$\mathbb{E}[u(Y)] \approx \mathbb{E}[u_h(Y)] = \sum_{k=1}^{\eta} w_k \; u_h\Big(Y^{(k)}\Big), \tag{13}$$

where $u_h(Y)$ is the spatially-discretized PDE solution, and $Y^{(k)} = [Y^{(k_1)}, \cdots, Y^{(k_N)}]$ are realizations of Y similar to Y_m in Monte Carlo but chosen at quadrature points $Y^{(k)}$ with corresponding weights w_k. For this study, uniformly-distributed uncertain parameters suggest Gauss-Legendre quadrature to obtain collocation points and weights. The necessary quadrature points are obtained based on polynomial expansion orders from an index set $\Lambda(L)$.

3.2.1 Index Set $\Lambda(L)$

The index set $\Lambda(L)$ provides the basis for most of the polynomial representation, including determining the quadrature set to evaluate the sum in Eq. 10. L is the polynomial degree of the stochastic collocation expansion. For example, for a single uncertain parameter ($N = 1$) and a fourth-order polynomial approximation ($L = 4$), Λ includes all polynomial orders from 0 to 4, ($\Lambda = [0, 1, 2, 3, 4]$). Each index point $p \in \Lambda$ corresponds to a polynomial expansion moment of order i.

In the multivariate case, there are several methods to determine what index set to use. In the most naive case, $\Lambda(L)$ is a tensor product of polynomial expansion orders,

$$\Lambda_{\text{TP}}(L) = \Big\{\bar{p} = [p_1, ..., p_N] : \max_{1 \le n \le N} p_n \le L\Big\}, \quad |\Lambda_{\text{TP}}(L)| = (L+1)^2. \tag{14}$$

For example, for $N = 2$ and $L = 4$, the index set $\Lambda_{\text{TP}}(L)$ includes all combinations of the expansion orders $[0, 1, 2, 3, 4] \otimes [0, 1, 2, 3, 4]$ for a total of 25 polynomial expansion indices. The collection of expansion points in this example index set is shown in Fig. 4a.

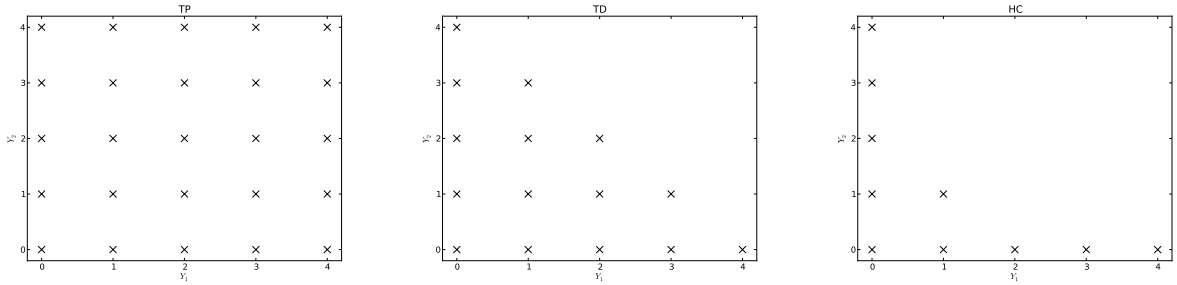
Other index sets with less cardinality can be employed to reduce the number of collocation points. Two in particular are the "total degree" set (see Fig. 4b), which is ideal for quantities that are analytic in stochastic space,

$$\Lambda_{\text{TD}}(L) = \Big\{\bar{p} = [p_1, ..., p_N] : \sum_{n=1}^{N} p_n \le L\Big\}, \quad |\Lambda_{\text{TD}}(L)| = \binom{L+N}{N}, \tag{15}$$

and the "hyberbolic cross" index set (see Fig. 4c), for quantities that have limited smoothness in stochastic space,

$$\Lambda_{\text{HC}}(L) = \Big\{\bar{p} = [p_1, ..., p_N] : \prod_{n=1}^{N} p_n + 1 \le L + 1\Big\}, \quad |\Lambda_{\text{HC}}(L)| \le (L+1)(1 + \log(L+1))^{N-1}. \tag{16}$$

Figure 4 shows each of these index sets for $N = 2, L = 4$. The savings of total degree and hyperbolic cross help fight the curse of dimensionality present in tensor product.



(a) Tensor Product (b) Total Degree (c) Hyperbolic Cross

Figure 4: Index Set Examples: $N = 2, L = 4$

3.2.2 Sparse Grid Quadrature

The collocation points used in the Lagrange polynomial expansion are obtained based on the index set chosen. To uniquely specify an isotropic Smolyak-like sparse grid, it is necessary to determine the number of uncertain variables N, the desired expansion order L, the index set $\Lambda(L)$, the quadrature rule $p(i)$, and the quadrature points-and-weights generating function for one-dimensional quadratures. This provides the isotropic sparse grid approximation

$$u(Y) \approx \mathcal{S}_{N,\Lambda(L)}[k](Y) = \sum_{\boldsymbol{i}\in\Lambda(L)} c(\boldsymbol{i}) \bigotimes_{n=1}^{N} \mathcal{U}_{n,p(i_n)}[u](Y), \tag{17}$$

$$c(\boldsymbol{i}) = \sum_{\substack{\boldsymbol{j}=\{0,1\}^N, \\ \boldsymbol{i}+\boldsymbol{j}\in\Lambda(L)}} (-1)^{|\boldsymbol{j}|_1}, \tag{18}$$

$$\bigotimes_{n=1}^{N} \mathcal{U}_{n,p(i_n)}[u](Y) \equiv \sum_{k_1=0}^{p(i_1)} \cdots \sum_{k_N=0}^{p(i_N)} u_h\Big(Y^{(k_1)}, \cdots, Y^{(k_N)}\Big) \prod_{n=1}^{N} \mathcal{L}_{k_n}(Y_n), \tag{19}$$

$$= \sum_{k}^{p(\vec{i})} u_h\Big(Y^{(k)}\Big) \mathcal{L}_k(Y), \tag{20}$$

where $p(i)$ is the "quadrature rule" used to obtain the number of quadrature points for a given expansion order. While this can be chosen arbitrarily, it is common to select $p(i) = i$ for Gauss quadrature and $p(i) = 2^i$ for Clenshaw-Curtis quadrature. The sparse grid quadrature is obtained through the tensor product of smaller quadratures, which has less cardinality than the full tensor quadrature. The savings in collocation points are demonstrated in Figure 5 for two identically distributed uniform variables, each on [-1,1]. The reduction in collocation points grows with the number of input parameters N and the expansion order L. A visual example of sparse grids for two variables uniformly distributed between -1 and 1 using Gauss-Legendre quadrature is shown in Fig. 5.

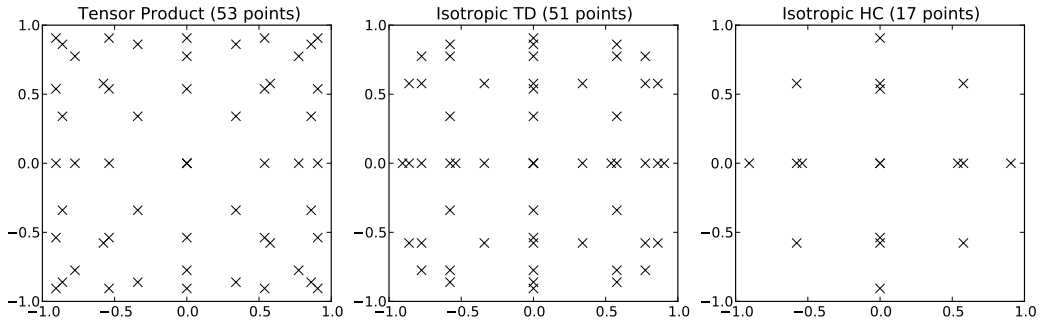


Figure 5: Sparse Grids, $N = 2, L = 4, p(i) = i$, Legendre points

For comparison, we show the number of index points for two input spaces of dimensionality N and several expansion levels L for all three index sets, as well as the number of collocation points for total degree and hyperbolic cross rules in Table 2. We note that accuracy cannot be directly drawn from polynomial expansions; for this problem the same number of collocation points results in a similar magnitude error in both total degree and hyperbolic cross. This implies, for this problem, that a much lower-order polynomial expansion constructed using the total degree rule is comparable in error to a larger-order polynomial expansion constructed using the hyperbolic cross rule.

N	L	TP	TD		HC	
		$\lvert\Lambda(L)\rvert$	$\lvert\Lambda(L)\rvert$	η	$\lvert\Lambda(L)\rvert$	η
3	4	125	35	165	16	31
	8	729	165	2,097	44	153
	16	4,913	969	41,857	113	513
5	2	293	21	61	11	11
	4	3,125	126	781	31	71
	8	59,049	1,287	28,553	111	481

Table 2: Index Set and Collocation Size Comparison

While the hyberbolic cross collocation points are significantly more sparse than the total degree collocation points, we note that the increased number of polynomial expansion moments in total degree make it much more accurate for the same total polynomial expansion level L. The results of this work demonstrate that for this problem, the same number of collocation points using either total degree or hyperbolic cross results in a similar magnitude of error.

3.2.3 Anisotropic Sparse Grid

One disadvantage of sparse grids as defined above is that it treats all dimensions of uncertainty space equally. In many cases, however, the sensitivity of the stochastic solution to one dimension is less than another. For example, we heuristically expect the sensitivity of the k-eigenvalue to the reflecting material (material 5) diffusion coefficient to be much less than the sensitivity to the fission cross section in the main fissile material (material 1).

We can leverage the varying sensitivity by introducing importance parameters $\vec{\alpha} = [\alpha_1, \cdots, \alpha_N]$ that

parametrize the sensitivity of the stochastic solution to each dimension. Using the one-norm

$$|\vec{\alpha}|_1 \equiv \frac{1}{N}\sum_{n=1}^{N}\alpha_n, \tag{21}$$

these weight parameters adjust the index set rules $\Lambda(L)$ for total degree and hyperbolic cross as

$$\tilde{\Lambda}_{\mathrm{TD}}(L) = \Big\{\bar{p} = [p_1, ..., p_N] : \sum_{n=1}^{N}\alpha_n p_n \leq |\vec{\alpha}|_1 L\Big\}, \tag{22}$$

$$\tilde{\Lambda}_{\mathrm{HC}}(L) = \Big\{\bar{p} = [p_1, ..., p_N] : \prod_{n=1}^{N}(p_n+1)^{\alpha_n} \leq (L+1)^{|\vec{\alpha}|_1}\Big\}. \tag{23}$$

In this formulation, greater values of α_n result in less quadrature points for uncertain input Y_n. Smaller values of α_n are assigned to more sensitive dimensions to prioritize collocation points.

3.2.4 SC Results

In order to compare magnitude of error, we use a "benchmark" solution generated with a high number of collocation points using isotropic sparse grid stochastic collocation as a reference solution. Monte Carlo and stochastic collocation solutions for varying numbers of PDE solves are computed, and relative error is plotted as a function of number of solves. The error is in the moments r of the quantity of interest $k(Y) = u(Y)$, and is given by

$$\epsilon_h^{(r)} = \frac{|\mathbb{E}[u^{(r)}] - \mathbb{E}[u_{\mathrm{ref}}^{(r)}]|}{\mathbb{E}[u_{\mathrm{ref}}^{(r)}]}, \tag{24}$$

$$\mathbb{E}[u^{(r)}] \approx \mathbb{E}[S_{N,\Lambda_{\mathrm{TD}}(L)}[u](Y)^{(r)}] = \sum_{k=1}^{\eta} w_k \; u^{(r)}\Big(Y^{(k)}\Big), \tag{25}$$

where the weights w_k and points $Y^{(k)}$ come from the multivariate quadrature used in stochastic collocation.

3.3 HDMR

Despite the gains in efficiency from using anisotropic sparse grids, stochastic collocation still suffers from the curse of dimensionality, becoming computationally unwieldy for more than a dozen uncertain inputs. We can further improve the efficiency of deterministic uncertainty quantification by making use of high-density model reduction (HDMR) techniques [5]. These methods have been applied recently to neutronics problems for significant gain in efficiency [3].

Cut-HDMR makes use of representative "slices" or cuts of the uncertainty space. In particular, it approximates a function as a sum of increasing interactions,

$$u(Y) = H[u](Y) = u_0 + \sum_{i}^{N} u_i + \sum_{i}^{N}\sum_{j}^{i} u_{i,j} + ..., \tag{26}$$

$$u_0 = u(\bar{Y}), \tag{27}$$

$$u_i \equiv u(Y_i, \bar{Y}) - u_0, \tag{28}$$

$$u_{i,j} \equiv u(Y_i, Y_j, \bar{Y}) - u_i - u_j - u_0, \tag{29}$$

and so on. Here N is the total number of uncertain inputs, and $\bar{Y}$ indicates holding any variables not explicitly listed at a reference value (in this case, the mean value). That is,

$$u(Y_i, \bar{Y}) \equiv u(\bar{Y}_1, \ldots, Y_i, \ldots, \bar{Y}_N). \tag{30}$$

This Cut-HDMR representation allows us to approximate by truncating at a particular interaction level H. For instance, an $H2$ approximation would only include the terms

$$u(Y) \approx H_2[u](Y) = u_0 + \sum_i^N u_i + \sum_i^N \sum_j^{i-1} u_{i,j}. \tag{31}$$

A variety of methods can be used to represent $u(Y)$. In our case, we make use of stochastic collocation on sparse grids, which is most efficient for a moderate number of uncertain inputs. For example,

$$u(Y) \approx H_2[u](Y) \approx S_0 + \sum_i^N S_i + \sum_i^N \sum_j^{i-1} S_{i,j}, \tag{32}$$

$$S_0 \equiv S[u](\bar{Y}), \tag{33}$$

$$S_i \equiv S[u](Y_i, \bar{Y}) - S_0, \tag{34}$$

$$S_{i,j} \equiv S[u](Y_i, Y_j, \bar{Y}) - S_i - S_j - S_0. \tag{35}$$

4 Acknowledgments

TODO This research was supported by research grants from Idaho National Laboratory.

References

[1] Ivo Babuska, Fabio Nobile, and Raul Tempone. A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis*, 45, 2007.

[2] Michael Heroux et. al. An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[3] Zhengzheng Hu, Ralph C. Smith, Jeffrey Willert, and C. T. Kelley. High Dimensional Model Representations for the Neutron Transport Equation. *NS&E*, 177, 2014.

[4] Argonne National Laboratory. Argonne code center: Benchmark problem book. *ANL-7416 M&C Division of ANS*, 1968.

[5] G. Li, C. Rosenthal, and H. Rabitz. High Dimensional Model Representations. *J. Phys. Chem. A*, 105, 2001.

[6] Nobile, Tempone, and Webster. A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis*, 46, 2008.