University of New Mexico

Doctoral Thesis

Advanced Methods in Stochastic Collocation for Polynomial Chaos in RAVEN

Author:
Paul W. Talbot

Supervisor:
Dr. Anil K. Prinja

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering

in the

Department of Nuclear Engineering

August 2015

Abstract

As the complexity of experiments in fields such as nuclear engineering continues to increase, so too does the demand for robust computational methods to simulate these experiments in virtual space. In many of these simulations, exact input design parameters as well as intrinsic properties of the experiment are often sources of input uncertainty. Often, small perturbations in these uncertain parameters have significant impact on the outcome of an experiment. For instance, when considering nuclear fuel performance experiments, small changes in the thermal conductivity of the fuel can greatly affect the maximum stress on the surrounding cladding. In recent years quantifying the impact of the input uncertainties in such an experimental system has grown as the complexities in these systems increase. For some problems, the input parametric space and corresponding quantity of interest output space is sufficiently explored with a few low-cost computational calculations. For others, however, the computational model is costly and it takes a great many random samples to obtain a good understanding of the output space. This research explores the possibilities of advanced methods in stochastic collocation for generalized polynomial chaos (SCgPC) as an alternative to traditional uncertainty quantification techniques such as Monte Carlo (MC) and Latin Hypercube sampling (LHS) methods. In this proposal we explore the behavior of traditional isotropic tensor product (TP) SCgPC, then expand to consider truncated polynomial spaces using total degree (TD) and hyperbolic cross (HC) construction strategies. Next, we consider applying anisotropy to the polynomial space construction, and analyze methods whereby the level of anisotropy can be approximated. This leads to introducing the Sobol decomposition method, or high-dimensional model representation (HDMR) method, both as a reduced-order model and as a method of obtaining sensitivity indices for informing anisotropic SCgPC. We analyze these methods on nontrivial neutron diffusion transport problems. Finally, we propose implementing adaptive algorithms for building both the polynomial space for SCgPC and the constituent subset space of HDMR in order to approach ideal efficiency in modeling high-dimension problems. Further, we propose implementing these methods in the uncertainty qunatification framework RAVEN and applying them to nuclear fuels performance code BISON.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fuels performance codes are numerical simulations intended to characterize the performance of a set of materials in a particular geometry under a certain environment, over time. The environmental considerations might include temperature, neutron flux, external pressure, and similar factors. In many cases, the performance is quantified by considering the maximum stress undergone by cladding around the fuel as it expands and makes contact. By varying the construction materials and geometry of the fuel, its cladding, and the gap between them, fuel can be designed for optimal performance without experiencing a rupture or similar break.

There are a plethora of parameters that go into simulating fuel performance. The fuel itself is made up of many constituent materials with a variety of densities and structures, as well as behavior under irradiation. The contents of the fuel-cladding gap determine how effectively heat can conduct out of the fuel and to the cladding, then out to a moderator, and the thickness of this gap determines the amount of fuel expansion allowed before contact is made and outward pressure begins increasing. The material and geometry of the cladding determine limits on stress and efficiency of heat transfer. Any of the material properties in the fuel, gap, or cladding, along with the environmental conditions, can be a source of uncertainty in determining the maximum stress applied to the cladding.

There are two categories into which sources of uncertainty fall: aleatoric, or the statistical uncertainty inherent in a system; and epistemic, or the systematic uncertainty due to imprecision in measurement or existence of measurable unknowns. While there are aleatoric uncertainties in fuel performance (such as the neutronics of irradiated fuel), in this work we consider mostly epistemic uncertainties surrounding the material properties and geometries of the problems. For an example case, we can consider the overall reactor power, fuel mesoscale grain growth, and fuel thermal expansion coefficient as uncertain input parameters, with maximum Von Mises stress in the axial center of a fuel rod as a quantity of interest in the output space.

In this work, we consider several methodologies for quantifying the uncertainty in fuel performance calculations. In order to demonstrate clearly the function of these methods, we demonstrate them first on several simpler problems, such as polynomial evaluations or projectile motion. The first method we consider is traditional, analog Monte Carlo (MC) analysis, wherein random sampling of the input space generates a view of the output space. MC is used as a benchmark methodology; if other methods converge on quantities of interest more quickly and accurately than MC, we consider them "better" for our purposes.

The second method we consider is isotropic, tensor-product (TP) stochastic collocation for generalized polynomial chaos (SCgPC)[1][2][3][4], whereby deterministic collocation points are used to develop a polynomial reduced-order model of the output quantities of interest as a function of the inputs. The other methods we consider expand on this method. First, we introduce non-tensor-product methods for determining polynomial bases, using the total degree (TD) and hyperbolic cross (HC) polynomial set construction methods[5]. These bases will then be used to construct Smolyak-like sparse grids for collocation[6]. Second, we consider anisotropic sparse grids, allowing additional collocation points for preferential input parameters. We also consider methods for determining weights that determine the level of preference to give parameters, and explore the effects of a variety of anisotropic choices.

The third method we consider is high-dimension model representation (HDMR), which correlates with Sobol decomposition [7]. This method is useful both for developing sensitivities of the quantity of interest to subsets of the input space, as well as constructing a reduced-order model itself. We demonstrate the strength of HDMR as a method to inform anisotropic sensitivity weights for SCgPC.

Additionally, we propose continued work on developing adaptive algorithms for both SCgPC and HDMR[8]. In adaptive SCgPC, the polynomial basis is constructed level-by-level based on the highest-impact subset polynomials. In adaptive HDMR, the constituent subset input spaces are developed similarly, based on the highest-impact input subset. The crowning achievement we propose is combining HDMR and SCgPC to develop both the subset input space as well as the overall reduced-order model adaptively in an attempt to construct a competitively-efficient method for uncertainty quantification.

Finally, we propose all these methods be developed within Idaho National Laboratory's `RAVEN`[9] uncertainty quantification framework. `RAVEN` is a Python-written framework that non-intrusively provides tools for analysts to quantify the uncertainty in their simulations with minimal impact. `RAVEN` has already been shown to work seamlessly with `MOOSE`-based fuel performance code `BISON`[10][11], on which we propose to demonstrate the various methods described above.

The remainder of this work will proceed as follows:

- Chapter 2: We mathematically describe the problems solved by the simulations we will be running, including polynomial evaluations, attenuation, projectile, diffusion, and fuel performance. We discuss their potential applications and approach using random sampling.
- Chapter 3: We describe several methods for uncertainty quantification, including Monte Carlo, Latin Hypercube sampling, generalized Polynomial Chaos, and high-dimension model representation. We also discuss methods to accelerate the convergence of SCgPC and HDMR models and.
- Chapter 4: We discuss proposed work extending both SCgPC and HDMR to be constructed adaptively. We also discuss the predicted shortfalls in the adaptive algorithms and some potential methods to address them.

Chapter 2

Simulation Physics

2.1 Simulations Used

To demonstrate the efficacy of the various uncertainty quantification methods we use in this work, we employ several independent simulations (hereafter referred to as “codes”) of increasing complexity. These codes range from simple polynomial evaluations to analytic solutions of simple physics, to nonlinear multistage iterative solvers representing complex codes. We describe each briefly here.

2.2 Polynomial Evaluations

In order to benchmark the simplest cases, we make use of simple polynomial expressions of the form

$$u(Y) = \prod_{i=1}^{N} (y_i^b + a), \tag{2.1}$$

where $u(Y)$ is the quantity of interest, $Y = [y_1, y_2, \ldots, y_N]$ is the vector of uncertain inputs, and a and b are arbitrary scalar values. The input variables Y can be distributed arbitrarily. These polynomials are demonstrations where SCgPC evaluates exactly at some finite expansion level.

2.3 Attenuation

To demonstrate the convergence rates of various methods, we make use of an adjusted attenuation problem that is equivalent to the penetration of point particles in a purely-absorbing medium. The uncertain input variables are each the length of a segment of material, and each segment has an identical cross section equal to 1. The quantity of interest is the percentage of particles exiting the material. We normalize the length of the segments to preserve significant values for the percentage exiting. The solution to this problem is

$$u(Y) = \prod_{i=1}^{N} e^{-Y_i/L}, \tag{2.2}$$

$$u(Y) = \sum_{i=1}^{N} \exp\left[\frac{\sum_{i=1}^{N} -Y_i}{L}\right], \tag{2.3}$$

where $L = |Y|_1$ is the total segment length. The input variables Y can be distributed arbitrarily.

The benefit of this model is a simple analytic solution, which makes analytic benchmarks possible. In addition, because of the exponential form, SCgPC will converge on a solution, but not replicate it exactly as in the polynomial case. This allows accurate comparison between MC, SCgPC methods, and HDMR methods.

2.4 Projectile

For a nonlinear case without an analytic solution, we consider the path traveled by a projectile near the surface of the Earth, considering varying gravitational pull with height as well as drag on the projectile from stagnant air. The equations governing travel in both vertical position x and horizontal position y are given by

$$y(t) = \frac{v_T}{g}(v\sin\theta + v_T)\left(1 - \exp\left[\frac{-gt}{v_T}\right]\right) - v_T t, \tag{2.4}$$

$$x(t) = \frac{v v_T \cos\theta}{g}\left(1 - \exp\left[\frac{-gt}{v_T}\right]\right), \tag{2.5}$$

where t is time, g is acceleration due to gravity, v is scalar velocity, θ the angle between the velocity vector and the horizontal ground, $v_T = \frac{mg}{D}$ is terminal velocity, $D = \frac{\rho C A}{2}$ is the acceleration due to drag, C is the drag coefficient, and $A = \pi r^2$ is the surface area of the projectile in the direction of travel. The projectile is assumed to present an identical surface area in both x and y directions. The quantity of interest is the range, or the total distance in x traveled by the ball before reaching a height of $y = 0$. The uncertain input variables are distributed uniformly as described in Table 2.1.

This simulation has the benefit of an analytic solution when $C = 0$ and eight distinct input parameters of varying importance. This is especially useful in considering anisotropic treatment of the input space.

Variable	Name	Mean	Range ($\pm$)	Units
y_i	Initial Height	1	1	m
v	Initial Velocity	35.5	2.5	m/s
θ	Initial Angle	45	10	degrees
g	Accel. Gravity	9.79888	0.0349	m/s/s
m	Projectile Mass	0.145	0.0725	kg
r	Projectile Radius	0.0336	0.00336	m
C	Drag Coefficient	0.5	0.5	
ρ	Air Density	1.2	0.1	kg/m^3

TABLE 2.1: Projectile Problem Distributions

2.5 Neutron Diffusion

For a nonlinear system with complicated physics, we consider a two-group, two-dimensional neutron diffusion criticality calculation. We make use of the diffusion approximation for neutron transport, which provides us with a coupled set of elliptic PDEs to solve:

$$-\nabla\cdot(D_1(\bar{x})\nabla\phi_1(\bar{x}))+\left(\Sigma_a^{(1)}(\bar{x})+\Sigma_s^{(1\to2)}(\bar{x})\right)\phi_1(\bar{x})=\frac{1}{k(\phi)}\sum_{g'=1}^{2}\nu_{g'}\Sigma_f^{(g')}(\bar{x})\phi_{g'}(\bar{x}), \quad (2.6)$$

$$-\nabla\cdot(D_2(\bar{x})\nabla\phi_2(\bar{x}))+\Sigma_a^{(2)}(\bar{x})\phi_2(\bar{x})=\Sigma_s^{(1\to2)}(\bar{x})\phi_1(\bar{x}), \quad (2.7)$$

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:

$$\left.\frac{\phi_g}{4}-\frac{D_g}{2}\frac{\partial\phi_g}{\partial x_1}\right|_{\partial\Omega_{\text{top}}}=0,\;\; g=1,2, \quad (2.8)$$

$$\left.\frac{\phi_g}{4}-\frac{D_g}{2}\frac{\partial\phi_g}{\partial x_2}\right|_{\partial\Omega_{\text{right}}}=0,\;\; g=1,2, \quad (2.9)$$

$$\left.-D_g\frac{\partial\phi_g}{\partial x_1}\right|_{\partial\Omega_{\text{bottom}}}=0,\;\; g=1,2, \quad (2.10)$$

$$\left.-D_g\frac{\partial\phi_g}{\partial x_2}\right|_{\partial\Omega_{\text{left}}}=0,\;\; g=1,2. \quad (2.11)$$

The criticality eigenvalue and quantity of interest $k(\phi)$ is given by

$$k(\phi) = \sum_{g=1}^{2} \iint_D \frac{\nu\Sigma_f^{(g)}\phi_g(\bar{x})}{\left(-\nabla \cdot D_g \nabla + \Sigma_r^{(g)}\right)\phi_g(\bar{x})}\, d\bar{x}. \tag{2.12}$$

We address solving ϕ_1, ϕ_2, and k nonlinearly and simultaneously. The material properties are shown in Table 2.2, and the domain $\Omega = [0, 200 \text{ cm}]^2$. The reference flux solutions are plotted in Fig. 2.2, and for the reference problem k=1.00007605445.

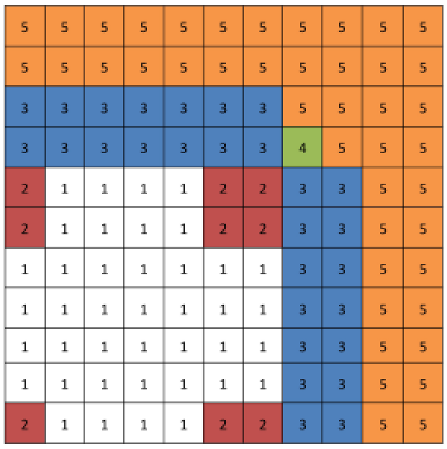
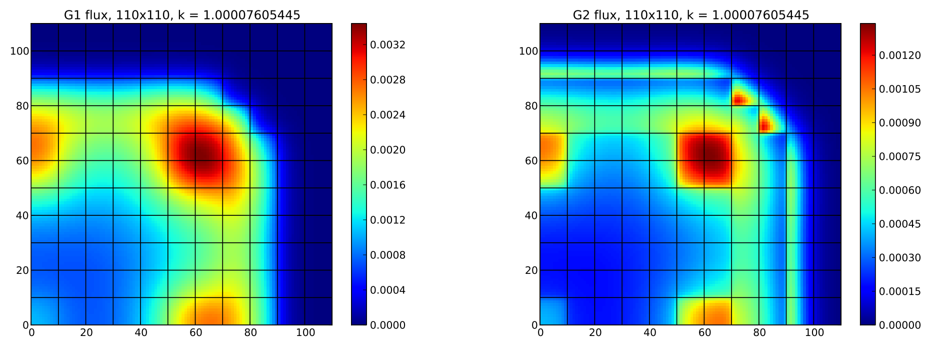


FIGURE 2.1: Core Geometry

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 2.2: Reference Material Properties for Benchmark Core



(A) ϕ, Group 1 (B) ϕ, Group 2

Figure 2.2: Reference Flux Profiles

2.6 Fuel Rod Performance

While the previous codes were designed and written by the author, `BISON` is a professional use code maintained by Idaho National Laboratory. The fuel performance simulation used in this work is described in [12], and involves an axial-symmetric LWR fuel rodlet, composed of ten uranium dioxide pellets, zirconium-4 cladding, gap, and upper plenum. The problem is a power ramp-up transient as delineated in Table 2.3. The three uncertain input parameters are distributed in a truncated normal and are shown in Table 2.4. Note that scaling parameters are used for both grain radius and reactor power, such that a scaling factor of one is the nominal unperturbed case.

Time (s)	Linear Power (W/m)
0	0
1e4	3.5e4
1.5e8	3.5e4

Table 2.3: Linear Power Time Evolution

Variable	Min. Value	Max. Value
Grain Radius Scale Factor	0.4	1.5
Thermal Expansion Coefficient	9e-6	1.1e-5
Linear Power Scaling Factor	0.95	1.05

Table 2.4: Fuel Performance Input Distributions

The quantity of interest are the stresses on the axial center of the cladding. The uncertain parameters used in this test case are the mesoscale grain radius of the fuel, the simulated power of the reactor surrounding the fuel rod, and the thermal expansion coefficient of the fuel.

Chapter 3

Uncertainty Quantification Techniques

3.1 Uncertainty Quantification Methods

We consider a few methods for uncertainty quantification (UQ). One method to classify UQ methods is by their interaction level with a simulation code. Non-intrusive methods treat the code as a black box, perturbing the inputs and collecting the outputs without modifying the code. These methods are ideal for generic application frameworks, where the simulation code may be unknown or precompiled. Examples of non-intrusive methods include analog Monte Carlo (MC), Latin Hypercube sampling (LHS), and deterministic collocation methods. Alternatively, intrusive methods require access to the solution algorithm itself. Sometimes this can provide more efficient solutions. In particular, adjoint methods require the solution operator to be reversible, and provide very efficient methods to determine sensitivities and analyze output-input dependence. While many intrusive methods have benefits, they lack the flexibility and universal applicability of non-intrusive methods, and so we neglect them in this work.

3.2 Correlation

An assumption we make going forward is that the uncertain input parameters are independent or at least uncorrelated. If this is not the case, methods such as Karhunen-Loeve expansion can be used to develop an orthogonal input space to replace the original. Similarly, principle component analysis can be used to attempt to find the fundamental uncorrelated space that maps into the correlated variable space.

3.3 Monte Carlo

In analog Monte Carlo uncertainty quantification, a single point in the input space is selected randomly, and an output collected, until an appropriately accurate view of the output space is acquired. While few samples result in a poor understanding of the quantity of interest, sufficiently large samples converge on a correct solution. The convergence rate of Monte Carlo is consistent, as

$$\epsilon = \frac{c}{\sqrt{\eta}}, \tag{3.1}$$

where c is a constant and η is the number of samples taken. While this convergence rate is slow, it is possibly one of the most reliable methods available. This makes MC a good choice for benchmarking.

One of the downsides of MC (and LHS) when compared with other methods is that they do not generate a reduced-order model as part of the evaluation; however, interpolation methods can be used to generate additional samples.

3.4 Latin Hypercube Sampling

Latin hypercube sampling[13] is another stochastic method that specializes in multi-dimensional distributions. In this method, the input domain is divided into M^N sub-domains that contain an equal probability volume, with M divisions along each axis. Sampling is then performed in these volumes, assuring no two samples share the same "row" for a given input dimension; that is, once a sample is taken within the i-th sub-division of an input range (axis), another sample may not be taken that is within that subdivision for that variable. Thus, subdividing each axis into M sections results in M total possible samples.

LHS is useful as a sampling strategy because it assures samples are distributed throughout the entirety of the input space, which is less guaranteed using MC sampling. There is some cost, however, associated with subdividing and tracking the grid on the input space.

3.5 Generalized Polynomial Chaos

In general, polynomial chaos expansion (PCE) methods seek to represent the simulation code as a combination of polynomials of varying degree and combination in each dimension of the input space. Originally Wiener proposed expanding in Hermite polynomials for Gaussian-normal distributed variables [14]. Askey and Wilson generalized this method for a range of Gaussian-based distributions with corresponding polynomials, including Legendre polynomials for uniform distributions, Laguerre polynomials for Gamma distributions, and Jacobi polynomials for Beta distributions[15].

In each of these cases, a probability-weighted integral over the distribution can be cast in a way that the corresponding polynomials are orthogonal over the same weight and interval. These chaos Wiener-Askey polynomials were used by Xiu and Karniadakis to develop the generalized polynomial chaos expansion method (gPC), including a transformation for applying the same method to arbitrary distributions (as long as they have a known inverse CDF)[4]. Two significant methodologies have grown from gPC application. The first makes use of Lagrange polynomials to expand the original function or simulation code, as they can be made orthogonal over the same domain as the distributions[16]; the other uses the Wiener-Askey polynomials[4]. We consider the latter in this work.

We consider a simulation code that produces a quantity of interesy u as a function $u(Y)$ whose arguments are the uncertain, distributed input parameters $Y = [Y_1, \ldots, Y_n, \ldots, Y_N]$. A particular realization ω of Y_n is expressed by $Y_n(\omega)$, and a single realization of the entire input space results in a solution to the function as $u(Y(\omega))$. We acknowledge obtaining a realization of $u(Y)$ may take considerable computation time and effort, and may be solved nonlinearly and without analytic solution. There may be other input parameters that contribute to the solution of $u(Y)$; we neglect these, as our interest is chiefly in the uncertainty space. In addition, it is possible that the quantity of interest $u(Y)$ is an integrated quantity or some norm of a value that is temporally or spatially distributed; in any case, we restrict $u(Y(\omega))$ to a single scalar output.

We expand $u(Y)$ in orthonormal multidimensional polynomials $\Phi_k(Y)$, where k is a multi-index tracking the polynomial in each axis of the polynomial Hilbert space, and $\Phi_k(Y)$ is constructed as

$$\Phi_k(Y) = \prod_{n=1}^{N} \phi_{k_n}(Y_n), \tag{3.2}$$

where $\phi_{k_n}(Y_n)$ is a single-dimension Wiener-Askey orthonormal polynomial of order k_n and $k = [k_1, \ldots, k_n, \ldots, k_N]$. The gPC for $u(Y)$ using this notation is

$$u(Y) \approx \sum_{k \in \Lambda(L)} u_k \Phi_k(Y), \tag{3.3}$$

where u_k is a weighting polynomial coefficient. The polynomials used in the expansion are determined by the set of multi-indices $\Lambda(L)$, where L is a truncation order for isotropic methods. In the limit that Λ contains all possible combinations of polynomials of any order, Eq. 3.2 is exact. Practically, however, Λ is truncated to some finite set of combinations, discussed in section 3.5.1.

Using the orthonormal properties of the Wiener-Askey polynomials,

$$\int_\Omega \Phi_k(Y)\Phi_{\hat{k}}(Y)\rho(Y)dY = \delta_{k\hat{k}}, \tag{3.4}$$

where $\rho(Y)$ is the combined PDF of Y, Ω is the multidimensional domain of Y, and δ_{nm} is the Dirac delta, we can isolate an expression of the polynomial expansion coefficients. We multiply both sides of Eq. 3.2 by $\Phi_{\hat{k}}(Y)$, integrate both sides over the probability-weighted input domain, and sum over all $\hat{k}$ to obtain the coefficients, sometimes referred to as polynomial expansion moments,

$$u_k = \frac{\langle u(Y)\Phi_k(Y)\rangle}{\langle \Phi_k(Y)^2\rangle}, \tag{3.5}$$

$$= \langle u(Y)\Phi_k(Y)\rangle, \tag{3.6}$$

where we use the angled bracket notation to denote the probability-weighted inner product,

$$\langle f(Y)\rangle \equiv \int_\Omega f(Y)\rho(Y)dY. \tag{3.7}$$

When $u(Y)$ has an analytic form, these coefficients can be solved by integration; however, in general other methods must be applied to numerically perform the integral. While tools such as Monte Carlo integration can be used to evaluate the integral, we can harness the properties of Gaussian quadratures because of the probability weights and domain. This stochastic collocation method is discussed in section **??**.

3.5.1 Polynomial Index Set Construction

There are many ways by which a polynomial index set can be constructed. Here we present three: tensor product, total degree, and hyperbolic cross.

In the nominal tensor product case, $\Lambda(L)$ contains all possible combinations of polynomial indices up to truncation order L in each dimension, as

$$\Lambda_{\text{TP}}(L) = \Big\{\bar{p} = [p_1, \cdots, p_N] : \max_{1\leq n\leq N} p_n \leq L\Big\}. \tag{3.8}$$

The cardinality of this index set is $|\Lambda_{\text{TP}}(L)| = (L+1)^N$. For example, for a two-dimensional input space (N=2) and truncation limit $L = 3$, the index set $\Lambda_{\text{TP}}(3)$ is given in Table 3.1, where the notation $(1,2)$ signifies the product of a polynomial that is first order in Y_1 and second order in Y_2.

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

It is evident there is some inefficiencies in this index set. First, it suffers dramatically from the *curse of dimensionality*; that is, the number of polynomials required grows exponentially with increasing dimension. Second, the total order of polynomials is not considered. Assuming the contribution of each higher-order polynomial is smaller than lower-order polynomials as per gPC, the (3,3) term is contributing sixth-order corrections that are likely smaller than the error introduced by ignoring fourth-order corrections (4,0) and (0,4). This leads to the development of the *total degree* (TD) and *hyperbolic cross* (HC) index set construction strategies[5].

In TD, only multidimensional polynomials whose *total* order is less than L are permitted, as

$$\Lambda_{\text{TD}}(L) = \Big\{\bar{p} = [p_1, \cdots, p_N] : \sum_{n=1}^{N} p_n \leq L\Big\}. \tag{3.9}$$

The cardinality of this index set is $|\Lambda_{\text{TD}}(L)| = \binom{L+N}{N}$, which grows with increasing dimension much more slowly than TP. For the same $N = 2, L = 3$ case above, the TD index set is given in Table 3.3.

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

Table 3.2: Total Degree Index Set, $N = 2, L = 3$

In HC, the *product* of polynomial orders is used to restrict allowed polynomials in the index set. This tends to polarize the expansion, emphasizing higher-order polynomials in each dimension but lower-order polynomials in combinations of dimensions, as

$$\Lambda_{\text{HC}}(L) = \Big\{\bar{p} = [p_1, \ldots, p_N] : \prod_{n=1}^{N} p_n + 1 \leq L + 1\Big\}. \tag{3.10}$$

The cardinality of this index set is bounded by $|\Lambda_{\text{HC}}(L)| \leq (L+1)(1+\log(L+1))^{N-1}$. It grows even more slowly than TD with increasing dimension, as shown in Table **??** for $N = 2, L = 3$.

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

Table 3.3: Hyperbolic Cross Index Set, $N = 2, L = 3$

It has been shown that the effectiveness of TD and HC as index set choices depends strongly on the regularity of the output space[5]. TD tends to be most effective for infinitely-continuous response surfaces, while HC is more effective for surfaces with limited smoothness or discontinuities.

3.5.2 Anisotropy

While using TD or HC to construct the polynomial index set combats the curse of dimensionality present in TP, it is not eliminated and continues to be a problem for problems of large dimension. Another method that can be applied to mitigate dimensionality problem is anisotropy, or the unequal treatment of multiple dimensions. In this strategy, weighting factors $\alpha = [\alpha_1, \dots, \alpha_n, \dots, \alpha_N]$ are applied in each dimension to allow additional polynomials in some dimensions and less in others. This change adjusts the TD and HC construction rules as follows, where $|\alpha|_1$ is the one-norm of α.

$$\tilde{\Lambda}_{\text{TD}}(L) = \Big\{\bar{p} = [p_1, \cdots, p_N] : \sum_{n=1}^{N} \alpha_n p_n \leq |\alpha|_1 L\Big\}, \tag{3.11}$$

$$\tilde{\Lambda}_{\text{HC}}(L) = \Big\{\bar{p} = [p_1, \cdots, p_N] : \prod_{n=1}^{N} (p_n + 1)^{\alpha_n} \leq (L+1)^{|\alpha|_1}\Big\}. \tag{3.12}$$

As it is desirable to obtain the isotropic case from a reduction of the anisotropic cases, we define the one-norm for the weights as

$$|\alpha|_1 = \frac{\sum_{n=1}^{N} \alpha_n}{N}. \tag{3.13}$$

Considering the same case above ($N = 2, L = 3$), we apply weights $\alpha_1 = 5, \alpha_2 = 3$, and the resulting index sets are Tables 3.4 (TD) and 3.5 (HC).

(2,0)				
(1,0)	(1,1)	(1,2)		
(0,0)	(0,1)	(0,2)	(0,3)	(0,4)

Table 3.4: Anisotropic Total Degree Index Set, $N = 2, L = 3$

There are many methods by which anisotropy weights can be assigned. Often, if a problem is well-known to an analyst, it may be enough to use heuristics to assign importance

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

TABLE 3.5: Anisotropic Hyperbolic Cross Index Set, $N = 2, L = 3$

arbitrarily. Otherwise, a smaller uncertainty quantification solve can be used to roughly determine sensitivity coefficients (such as Pearson coefficients), and the inverse of those can then be applied as anisotropy weights. Sobol coefficients obtained from first- or second-order HDMR, discussed later, could also serve as a basis for these weights. As will be shown, a good choice of anisotropy weight can greatly speed up convergence; however, a poor choice can slow convergence considerably, as computational resources are used to resolve low-importance dimensions.

3.6 Stochastic Collocation

Stochastic collocation is the process of using collocated points to approximate integrals of stochastic space numerically. In particular we consider using Gaussian quadratures (Legendre, Hermite, Laguerre, and Jacobi) corresponding to the polynomial expansion polynomials for numerical integration. Quadrature integration takes the form

$$\int_a^b f(x)\rho(x) = \sum_{\ell=1}^{\infty} w_\ell f(x_\ell), \tag{3.14}$$

$$\approx \sum_{\ell=1}^{L} w_\ell f(x_\ell), \tag{3.15}$$

where w_ℓ, x_ℓ are corresponding points and weights belonging to the quadrature set, truncated at order $\hat{L}$. At this point, this $\hat{L}$ should not be confused with the polynomial expansion truncation order L. We can simplify this expression using the operator notation

$$q^{(\hat{L})}[f(x)] \equiv \sum_{\ell=1}^{\hat{L}} w_\ell f(x_\ell). \tag{3.16}$$

A nominal multidimensional quadrature is the tensor product of individual quadrature weights and points, and can be written

$$Q^{(\mathbf{L})} = q_1^{(\hat{L}_1)} \otimes q_2^{(\hat{L}_2)} \otimes \cdots, \tag{3.17}$$

$$= \bigotimes_{n=1}^{N} q_n^{(\hat{L}_n)}. \tag{3.18}$$

It is worth noting each quadrature may have distinct points and weights; they need not be constructed using the same quadrature rule. In general, one-dimensional Gaussian quadrature excels in exactly integrating polynomials of order $2p-1$ using p points and weights; equivalently, it requires $(p+1)/2$ points to integrate an order p polynomial. ¡TODO¿ A summary of the Gaussian quadratures and corresponding probability distribution weight functions are described in an appendix ¡/TODO¿. For convenience we repeat here the coefficient integral we desire to evaluate, Eq. 3.5.

$$u_k = \langle u(Y)\Phi_k(Y)\rangle. \tag{3.19}$$

We can approximate this integral with the appropriate Gaussian quadrature as

$$u_k \approx Q^{(\hat{\mathbf{L}})}[u(Y)\Phi_k(Y)], \tag{3.20}$$

where we use bold vector notation to note the order of each individual quadrature, $\hat{\mathbf{L}} = [\hat{L}_1, \ldots, \hat{L}_n, \ldots, \hat{L}_N]$. For clarity, we remove the bold notation and assume a one-dimensional problem, which extrapolates as expected into the multidimensional case.

$$u_k \approx q^{(\hat{L})}[u(Y)\Phi_k(Y)], \tag{3.21}$$

$$= \sum_{\ell=1}^{\hat{L}} w_\ell u(Y_\ell)\Phi_k(Y_\ell). \tag{3.22}$$

In order to determine the quadrature order $\hat{L}$ needed to accurately integrate this expression, we consider the gPC formulation for $u(Y)$ in Eq. 3.2 and replace it in the sum,

$$u_k \approx \sum_{\ell=1}^{\hat{L}} w_\ell \Phi_k(Y_\ell) \sum_{k\in\Lambda(L)} u_{\hat{k}}\Phi_{\hat{k}}(Y_\ell). \tag{3.23}$$

Using orthogonal properties of the polynomials, this reduces as $\hat{L} \to \infty$ to

$$u_k \approx \sum_{\ell=1}^{\hat{L}} w_\ell u_k \Phi_k(Y_\ell)^2. \tag{3.24}$$

Thus, the integral, to the same approximation that the gPC expansion itself is approximated, the quadrature is approximating an integral of order $2k$, and the quadrature order itself should be order

$$p = \frac{2k+1}{2} = k + \frac{1}{2} < k+1, \tag{3.25}$$

so we can conservatively use $k+1$ as the quadrature order. In the case of the largest polynomials with order $k = L$, the quadrature size $\hat{L}$ is the same as $L+1$. It is worth noting that if $u(Y)$ is effectively of much higher-order polynomial than L, this equality for quadrature order does not hold true; however, it also means that gPC of order L will be a poor approximation.

While a tensor product of highest-necessary quadrature orders could serve as a suitable multidimensional quadrature set, we can make use of Smolyak-like sparse quadratures to reduce the number of function evaluations necessary for the TD and HC polynomial index set construction strategies.

3.6.1 Smolyak Sparse Grids

Smolyak sparse grids[6] are an attempt to discover the smallest necessary quadrature set to integrate a multidimensional integral with varying orders of predetermined quadrature sets. In our case, the polynomial index sets determine the quadrature orders each one needs in each dimension to be integrated accurately. For example, the polynomial index set point (2,1,3) requires three points in Y_1, two in Y_2, and four in Y_3,or

$$Q^{(2,1,3)} = q_1^{(3)} \otimes q_2^{(2)} \otimes q_3^{(4)}. \tag{3.26}$$

The full tensor grid of all collocation points would be the tensor product of all quadrature for all points, or

$$Q^{(\Lambda(L))} = \bigotimes_{k\in\Lambda} Q^{(k)}. \tag{3.27}$$

Smolyak sparse grids consolidate this tensor form by adding together the points from tensor products of subset quadrature sets. Returning momentarily to a one-dimensional problem, we introduce the notation

$$\Delta_k^{(\hat{L})}[f(x)] \equiv \left(q_k^{(\hat{L})} - q_{k-1}^{(\hat{L})}\right)[f(x)], \tag{3.28}$$

$$q_0^{(\hat{L})}[f(x)] = 0. \tag{3.29}$$

A Smolyak sparse grid then be defined and applied to the desired integral in Eq. 3.5,

$$S_{\Lambda,N}^{(\hat{\mathbf{L}})}[u(Y)\Phi_k(Y)] = \sum_{k\in\Lambda(L)} (\Delta_{k_1}^{(\hat{L}_1)} \otimes \cdots \otimes \Delta_{k_N}^{(\hat{L}_N)}[u(Y)\Phi_k(Y)]. \tag{3.30}$$

Equivalently, and in a more algorithm-friendly approach,

$$S_{\Lambda,N}^{(\hat{\mathbf{L}})}[u(Y)\Phi_k(Y)] = \sum_{k\in\Lambda(L)} c(k) \bigotimes_{n=1}^{N} q_n^{(\hat{L}_n)}[u(Y)\Phi_k(Y)] \tag{3.31}$$

where

$$c(k) = \sum_{\substack{j=\{0,1\}^N, \\ k+j\in\Lambda}} (-1)^{|j|_1}, \tag{3.32}$$

using the traditional 1-norm for $|j|_1$. The values for u_k can then be calculated as

$$u_k = \langle u(Y)\Phi_k(Y)\rangle, \tag{3.33}$$

$$\approx S^{(\hat{\mathbf{L}})}_{\Lambda,N}[u(Y)\Phi_k(Y)], \tag{3.34}$$

and gPC for $u(Y)$ is complete.

3.7 High-Dimension Model Representation (HDMR)

While using SCgPC is one method for creating a reduced-order model for a simulation code $u(Y)$, another useful model reduction is HDMR[7], or Sobol decomposition. In particular, we consider Cut-HDMR in this work. In this methodology, the representation of $u(Y)$ is built up from the contributions of subsets of the input space. We consider a reference point realization of Y (nominally the mean of each distribution) $Y^{(0)} = [Y_1^{(0)}, \ldots, Y_N^{(0)}]$ and introduce the notation

$$\hat{Y}_n \equiv [Y_1, \ldots, Y_{n-1}, Y_{n+1}, \ldots, Y_N], \tag{3.35}$$

and

$$u(Y_n)\Big|_{\hat{Y}_n} \equiv u(\hat{Y}_n^{(0)}, Y_n). \tag{3.36}$$

In words, this notation describes holding all the input variables except Y_n constant and only allowing Y_n to vary. Similarly for higher dimensions,

$$u(Y_m, Y_n)\Big|_{\hat{Y}_{m,n}} \equiv u(\hat{Y}_{m,n}^{(0)}, Y_m, Y_n), \tag{3.37}$$

$$= u(Y_1^{(0)}, \ldots, Y_{m-1}^{(0)}, Y_m, Y_{m+1}^{(0)}, \ldots, Y_{n-1}^{(0)}, Y_n, Y_{n+1}^{(0)}, \ldots, Y_N^{(0)}). \tag{3.38}$$

The HDMR reduced-order model is then constructed as

$$\begin{aligned} u(Y) = u_0 &+ \sum_{n=1}^{N} u_{Y_n} \\ &+ \sum_{n_1=1}^{N}\sum_{n_2=1}^{n_1-1} u_{Y_{n_1},Y_{n_2}} \\ &+ \sum_{n_1=1}^{N}\sum_{n_2=1}^{n_1-1}\sum_{n_3=1}^{n_2-1} u_{Y_{n_1},Y_{n_2},Y_{n_3}} \\ &\cdots \end{aligned} \tag{3.39}$$

$$+ u_{Y_{n_1},\cdots,Y_{n_N}}, \tag{3.40}$$

where

$$u_0 = u(Y^{(0)}), \tag{3.41}$$

$$u_{Y_n} = u(Y_n)\Big|_{Y_n} - u_0, \tag{3.42}$$

$$u_{Y_m,Y_n} = u(Y_m, Y_n)\Big|_{Y_m,Y_n} - u_{Y_m} - u_{Y_n} - u_0, \tag{3.43}$$

$$\cdots \tag{3.44}$$

If not truncated, this wastes significant computational effort; however, truncating at second- or third-order interactions can often capture much of the physics with less computation effort than a full solve. It is important to note that the HDMR is not a value itself, but a reduced-order model. To speed up computation using this model, each component term in the HDMR expansion can be replaced in turn by a reduced-order model. In particular, because SCgPC is most effective in calculations involving a small input space, gPC reduced-order models are excellent for representing components of the HDMR expansion.

In addition, Sobol sensitivity indices s can be obtained by taking the ratio of any one expansion term against the remainder of the terms. For first-order sensitivities,

$$s_n = \frac{\text{var}(u_{Y_n})}{\text{var}(u(Y))}, \tag{3.45}$$

and for second-order,

$$s_{m,n} = \frac{\text{var}(u_{Y_m,Y_n})}{\text{var}(u(Y))}, \tag{3.46}$$

etc. These sensitivities can be used to inform adaptive SCgPC calculations on the full model.

Chapter 4

Proposed Work

4.1 Proposal

There are several opportunities to continue this preliminary work. First, it is possible [17] to adaptively construct polynomial index sets instead of using deterministic index sets. Similarly, HDMR can be expanded with adaptive algorithms to build up the representation in consecutive subsets[8]. These adaptive algorithms have been shown to greatly affect the curse of dimensionality, to the point where HDMR based on SCgPC is competitive with Monte Carlo sampling for input spaces with dimensionality on the order of hundreds.

In concert with this, while Gaussian quadrature is effective at integrating polynomials, it has been shown [18] that other quadratures with convenient properties can compete effectively with them. In particular, when performing adaptive sampling, nested quadratures offer a significant reduction in necessary computational cost for convergence, since quadrature points are re-used in successive layers. We propose implementing and considering some nested quadrature strategies such as Clenshaw-Curtis and Gauss-Patterson and analyzing their impact on convergence when used in SCgPC.

Similarly, it would be of interest to contrast the use of Lagrange polynomials in the gPC expansion with versus the Wiener-Askey polynomials used thus far. We expect to see little change in convergence rates except in cases where the simulation model is easily-replicated by either set of polynomial families.

Lastly, we propose implementing all the methods described in this work in the `RAVEN` uncertainty quantification framework, and perform thorough analysis on a selected `BISON` case. Similar analysis has been performed recently [19] but without the benefit of some of the adaptive and combined strategies established and proposed in this work. Such implementation would demonstrate significant viability for the methods and make them more readily available to uncertainty quantification analysts.

Bibliography

[1] 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.

[2] Voker Barthelmann, Erich Novak, and Klaus Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12, 2000.

[3] Hans-Joachim Bungartz and Michael Griebel. Sparse Grids. *Acta Numerica*, 13, 2004.

[4] D. Xiu and G. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[5] Erich Novak and Klaus Ritter. The curse of dimension and a universal method for numerical integration. In Günther Nürnberger, JochenW. Schmidt, and Guido Walz, editors, *Multivariate Approximation and Splines*, volume 125 of *ISNM International Series of Numerical Mathematics*, pages 177–187. Birkhäuser Basel, 1997.

[6] Sergey A Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Dokl. Akad. Nauk SSSR*, volume 4, page 123, 1963.

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

[8] D. Ayres and M. D. Eaton. Uncertainty quantification in nuclear criticality modelling using a high dimensional model representation. *Annals of Nuclear Energy*, 80:379–402, May 2015.

[9] C. Rabiti, A. Alfonsi, D. Mandelli, J. Cogliati, and R. Kinoshita. Raven, a new software for dynamic risk analysis. In *PSAM 12 Probabilistic Safety Assessment and Management*, Honolulu, Hawaii, June 2014.

[10] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandié. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768 – 1778, 2009.

[11] Chris Newman, Glen Hansen, and Derek Gaston. Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in nuclear fuel rods. *Journal of Nuclear Materials*, 392(1):6 – 15, 2009.

[12] C. Rabiti, J. Cogliati, G. Pastore, R. J. Gardner, and A. Alfonsi. Fuel reliability analysis using bison and raven. In *PSA 2015 Probabilistic Safety Assessment and Analysis*, Sun Valley, Idaho, April 2015.

[13] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, 1979.

[14] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60:897–936, 1938.

[15] R. Askey and J. Wilson. Some basic hypergeometric orthogonal polynomials that generalize jacobi polynomials. *Memoirs of the American Mathematical Society*, 54: 1–55, 1985.

[16] 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.

[17] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71, 2003.

[18] Lloyd N. Trefethen. Is guass quadrature better than clenshaw-curtis? *SIAM Review*, 50(1):67–87, 2008.

[19] A. Boulore, C. Struzik, and F. Gaudier. Uncertainty and sensitivity analysis of the nuclear fuel thermal behavior. *Nuclear Engineering and Design*, 253:200–210, 2012.