Uncertainty Quantification in Nuclear Criticality Modelling Using a High Dimensional Model Representation

D. Ayres, M.D.Eaton

Nuclear Engineering Group, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, Prince Consort Road, London SW7 2AZ, UK

Abstract

An adaptive high dimensional model representation (HDMR) is used to decompose the response parameter k_{eff} into a superposition of lower dimensional subspaces which are in-turn projected on to a polynomial basis. These projections are evaluated using an adaptive quadrature scheme which is used to infer the polynomial orders of the basis. The combination of adaptive HDMR and adaptive quadrature techniques results in a sparse polynomial expansion which has been optimised to represent the variance of the response with the minimum number of polynomials. The combined application of these techniques is illustrated using UOX and MOX pin cell problems with evaluated nuclear covariance data. We show that this approach to calculating the variance in k_{eff} is an order of magnitude more efficient when compared to Latin hypercube sampling with the same number of samples for problems involving up to 988 random dimensions.

Keywords: Polynomial Chaos, High Dimensional Model Representation, Covariance data Nuclear Criticality

1. Introduction

In recent years there has been significant interest in developing computationally accurate, yet efficient, uncertainty quantification methods that can model both small and large parametric uncertainties. The polynomial chaos method has attracted significant interest as it can model both small and large parametric errors whilst also being computationally efficient, accurate and able to be applied to a very wide range of physical problems. This technique has been used for a variety of applications including reactor physics [1, 2, 3] and neutron transport [4, 5, 6, 7]. This paper aims to determine the computational accuracy and efficiency of polynomial chaos methods when applied to uncertainty quantification in nuclear criticality problems. More specifically, this paper focusses on the determination of the uncertainty of the effective multiplication factor (k_{eff}) due to uncertainties in the microscopic neutron cross-sections.

Email address: d.ayres10@imperial.ac.uk (D. Ayres)

The technique known as polynomial chaos was originally proposed by Wiener [8]. In this approach, a random process is expanded using Hermite polynomials in terms of Gaussian random variables. This work was later generalised by Xiu and Karniadakis [9] who associated an optimal polynomial basis from the Askey scheme to the probability weight of the underlying random process. The statistics of the random process are calculated directly from the coefficients in the polynomial chaos expansion (PCE). The PCE coefficients can be determined using one of two ways. The first is the intrusive approach. Here the random process is replaced explicitly in the governing equations with a PCE. A Bubnov-Galerkin projection is then performed yielding a set of coupled equations for the coefficients, see [10] for a complete description. The second is the non-intrusive approach which is performed either by projection [11] or regression [12]. The non-intrusive approach is the method we adopt here since it requires no modification to existing modelling codes. In the non-intrusive approach the PCE is computed by strategically sampling the space of all the uncertain inputs in a similar manner to Monte Carlo approaches but at prescribed collocation points in stochastic space.

The number of polynomials, N_p, in a PCE depends upon the number of random dimensions, M, and the order of the polynomial expansion, p, as follows:

$$N_p + 1 = \frac{(M+p)!}{M!p!} \tag{1}$$

As we can see from Eqn.(1), the number of terms increases rapidly with M and p. There are some techniques which *a-priori* reduce the number of terms N_p. One such method, known as a low rank set [13], limits the number of multivariate polynomials in the expansion. i.e. only bi-variate or tri-variate polynomials would be admissible. This approach does not converge to the true model response as $p \to \infty$. For example, considering only bi-variate terms would exclude all tri-variate and higher interactions. Another method computes a so called hyperbolic set [13]. This approach restricts the polynomial order p to below a hyperbola which reduces the growth rate of N_p. As $p \to \infty$ the hyperbolic set converges to the true response. The success of both of these PCE construction methods relies upon prior knowledge of the random process. If the structure of the process is unknown, then these constructions may be inefficient.

As we have mentioned, it is the non-intrusive approach that we adopt in this work. More specifically, we will be using the projection based approach. Here the random process is projected onto each polynomial basis in order to calculate the expansion coefficients. This projection involves an M dimensional integral which is most commonly performed using a quadrature scheme. The immediate approach to building a multidimensional quadrature is to use a tensor product of one dimensional quadrature rules. However, this results in the computational effort increasing exponentially with M. This exponential dependence was alleviated to some extent by using a sparse grid construction method [14]. However, this approach has no knowledge of the structure of the function being integrated and may ultimately be inefficient and/or inaccurate. The sparse grid construction was later generalised by Gerstner and Griebel [15, 16] and combined with an adaptive strategy which iteratively builds a set of integration

points in multidimensions. This generalised construction optimally integrates a function and can be used to infer its structure. As such, it has been used by Gilli [17] to build a sparse polynomial chaos expansion. In this approach the structure of the function inferred by the adaptive quadrature scheme has been used to calculate the polynomial basis required for each dimension and between combinations of dimensions.

In the work by Gilli [17], the number of dimensions of each sub-grid in the sparse grid construction was increased iteratively until an error threshold was met. In this work we use an alternative but analogous procedure known as high dimensional model representation (HDMR) [18, 19, 20, 21, 22, 23] to describe the multivariate interactions in the integrand. The HDMR method is used to capture high dimensional relationships between input and output model parameters using a hierarchical expansion of increasing dimension. If the cooperative effect of many input parameters upon the output is weak then the HDMR provides a very efficient representation of the system response allowing the integrand to be expressed using only low order component functions.

For some engineering applications [22] it is stated that component functions of order 3 or greater are negligible but this is obviously problem dependent. To tailor the truncation of the HDMR expansion to specific problems, adaptive strategies have been used [24]. The adaptive cut-HDMR expansion has also been used for problems with a discontinuous response function [25] where an adaptive sparse-grid quadrature method was used to compute the HDMR component functions. The adaptive quadrature method used a local hierarchical basis combined with the Smolyak algorithm. The statistical moments were calculated directly from quadrature hence the solution was not projected onto a polynomial chaos basis.

In summary, the objectives of this work are to use an adaptive HDMR technique to identify all of the important dimensions and interactions that contribute to the uncertainty in k_{eff}. The component functions in the HDMR expansion will be evaluated using an adaptive quadrature rule. This will in turn be used to indicate the order of polynomial basis required in the construction of a PCE. The aim of combining these three techniques is to build a sparse PCE in M dimensions that accurately represents the uncertainty in k_{eff} whilst minimising the number of model evaluations in the non-intrusive approach.

2. Description of the Problem

In this work we are concerned with determining the uncertainty in the calculation of k_{eff} due to the presence of uncertainties in the input nuclear data. The calculation of k_{eff} involves the solution of an eigenvalue problem; the eigenvector in this case describes the distribution of neutrons in space, angle and energy. As such, appropriate discretisations of the space, angle and energy variables must be performed for the problem to be amenable to numerical solution. The energy variable is discretised using the multi-group technique [26] and upon application of this, the eigenvalue problem

for k_{eff} is written as [27]

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\psi_g(\boldsymbol{r},\boldsymbol{\Omega})+\Sigma_{tg}(\boldsymbol{r})\psi_g(\boldsymbol{r},\boldsymbol{\Omega})&\\ =\int_{4\pi}d\boldsymbol{\Omega}'\sum_{h=1}^{N_G}\Sigma_{s;h\to g}(\boldsymbol{r},\boldsymbol{\Omega}'\to\boldsymbol{\Omega})\psi_h(\boldsymbol{r},\boldsymbol{\Omega}')&\\ +\frac{\chi_g}{4\pi k_{\text{eff}}}\int_{4\pi}d\boldsymbol{\Omega}'\sum_{h=1}^{N_G}\overline{\nu}\Sigma_{f,h}(\boldsymbol{r})\psi_h(\boldsymbol{r},\boldsymbol{\Omega}')\quad g=1,N_G&\end{aligned}\tag{2}$$

where

- $\boldsymbol{\Omega}$ is the direction of neutron travel.
- $\boldsymbol{r}$ the spatial position of the neutron.
- ψ_g is the angular neutron flux in group g.
- $\Sigma_{t,g}$ and $\Sigma_{f,g}$ are the total and fission macroscopic cross sections in group g respectively. $\Sigma_{s,h\to g}$ is the differential macroscopic scattering neutron cross section from group h into group g.
- $\overline{\nu}$ is the average number of neutrons produced per fission and is expressed by

  $$\overline{\nu}=\overline{\nu}_p+\sum_i\overline{\nu}_{i,d}$$

  where $\overline{\nu}_p$ is the average number of prompt neutrons produced per fission and $\overline{\nu}_{i,d}$ is the average number of delayed neutrons from delayed precursor group i produced per fission.
- χ is the steady state fission spectrum and is expressed as

  $$\chi=\left[1-\sum_i\beta_i\right]\chi_p+\sum_i\beta_i\chi_{i,d}$$

  where β_i is the fraction of fission neutrons emanating from delayed precursor group i, χ_p is the prompt neutron fission spectrum and $\chi_{i,d}$ is the delayed fission spectrum from precursor group i.

The deterministic nuclear criticality (k_{eff}) calculations have been performed using a code based upon the second-order even-parity form of the neutron transport equation. This code is called EVENT and employs a geometry conforming finite element discretisation of the spatial domain and rotationally invariant spherical harmonic basis for discretising the angular domain [28].

To model the uncertainties present in the nuclear data, the variation is represented parametrically using a set of independent, identically distributed (IID) random variables $\boldsymbol{\xi}(\theta)=\{\xi_1(\theta),\xi_2(\theta),\ldots,\xi_M(\theta)\}$. Here θ is a random event belonging to the

probability space of $\boldsymbol{\xi}$ and M is the total number of stochastic/random dimensions in the problem. In this work it is assumed that the uncertainties are in the microscopic neutron cross-section data and any uncertainties in the number densities, associated with the macroscopic neutron cross-sections, are negligible.

The uncertain nuclear data parameters considered in this work are now represented as sets of random numbers describing the uncertainty of that parameter in each energy group. For example, the total cross section is the set $\Sigma_t(\boldsymbol{r}, \boldsymbol{\xi}) = \{\Sigma_{t,g}(\boldsymbol{r}, \boldsymbol{\xi}); g = 1, \cdots, N_G\}$. The nuclear data are thus a random process [29]. If all of the random variables in a random process are uncorrelated, then the set $\boldsymbol{\xi}$ of standard variables may be defined in the same probability space as that input field. If, however, the random variables in a random field are correlated then $\boldsymbol{\xi}$ must be defined on a Gaussian probability space. For non-Gaussian correlated random fields this necessitates a transformation to the Gaussian field. These transformations will be discussed later. For the moment we focus on the representation of a Gaussian field in terms of the set of IID standard Gaussian variables $\boldsymbol{\xi}$.

2.1. Modelling Correlated Uncertainties

We assume that all of the random processes in this work are real valued and indexed by E on the bounded domain $\Gamma \subseteq \mathbb{R}^+$. The symbol E denotes the piecewise continuous energy variable and Γ is the domain of allowed energies. For any fixed value $E \in \Gamma$, the function $\Sigma(E; \cdot)$ is a random variable. For a fixed event θ, the function $\Sigma(\cdot, \boldsymbol{\xi}(\theta))$ is called a realization of the random process. All realizations $\Sigma(\cdot, \theta)$ are assumed to be in the Hilbert space $L_2(\Gamma)$ and all random variables are assumed to be in the Hilbert space $L_2(\Theta, \mathcal{B}, P)$. The second assumption implies that all random variables have a finite variance.

The mean value in energy group g of a random process describing the uncertainty in the nuclear data is defined as

$$\begin{aligned} \overline{\Sigma}_g &= \int_\Theta \Sigma(E_g, \boldsymbol{\xi})\, p(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \mathbb{E}[\Sigma(E_g, \boldsymbol{\xi})] \end{aligned} \tag{3}$$

where $p(\boldsymbol{\xi})$ is the joint probability density function for the random variables $\boldsymbol{\xi}$ and the expectation operator $\mathbb{E}[\cdot]$ is defined as

$$\mathbb{E}[\cdot] = \int_\Theta \cdot\, p(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{4}$$

For the case of independent Gaussian random variables with zero mean and unit variance, the PDF is defined as follows

$$p(\boldsymbol{\xi}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\xi_i^2\right\} \tag{5}$$

For any two random processes $\Sigma, \Sigma' \in L_2(\Theta, \mathcal{B}, P) \times L_2(\Gamma)$ the covariance between energy groups g and h is defined as follows:

$$\mathrm{Cov}_{\Sigma\Sigma'}(E_g, E_h) = \mathbb{E}\left[[\Sigma(E_g, \boldsymbol{\xi}) - \overline{\Sigma}_g][\Sigma'(E_h, \boldsymbol{\xi}) - \overline{\Sigma}'_h]\right] \tag{6}$$

We may also define the relative covariance

$$\mathrm{rCov}_{\Sigma\Sigma'}(E_g, E_h) = \frac{\mathrm{Cov}_{\Sigma\Sigma'}(E_g, E_h)}{\overline{\Sigma}_g\, \overline{\Sigma}'_h} \tag{7}$$

and the correlation

$$\rho_{\Sigma\Sigma'}(E_g, E_h) = \frac{\mathrm{Cov}_{\Sigma\Sigma'}(E_g, E_h)}{\sqrt{v_{\Sigma_g}}\sqrt{v_{\Sigma'_h}}} \tag{8}$$

where the variance v_{Σ_g} is defined as

$$v_{\Sigma_g} = \mathbb{E}\left[\Sigma^2(E_g, \boldsymbol{\xi})\right] - \overline{\Sigma}_g^2 \tag{9}$$

If the random processes Σ, Σ' comprise a sum of other random processes in energy, namely

$$\Sigma = \sum_i \alpha_i a_i \quad \Sigma' = \sum_j \beta_j b_j \tag{10}$$

where the sets $\{a_i\}$ and $\{b_j\}$ are random process and α_i and β_j are deterministic coefficients, then the covariance defined in Eqn.(6) is written as follows

$$\mathrm{Cov}_{\Sigma\Sigma'}(E_g, E_h) = \sum_i \sum_j \alpha_i \beta_j \mathrm{Cov}_{a_i b_j}(E_g, E_h) \tag{11}$$

2.1.1. The Karhunen Loéve Expansion

For a random process indexed on a bounded domain with finite second order moments (variance), we may expand as follows [30, 31]

$$\Sigma(E_g, \theta) = \overline{\Sigma}_g + \sum_{i=1}^{M} \sqrt{\lambda_i}\Phi_i(E_g)\xi_i(\theta) \tag{12}$$

here $\overline{\Sigma}_g$ is the mean value of the random process at energy E_g, as defined by Eqn.(3), and M is the number of random variables in the random process. The random process has been split into a deterministic function of energy and a random component which has been expanded using the following parameters: $\Phi_i(E)$ which describes the modes of variation as a function of energy, λ_i which describes the magnitude of each mode and $\boldsymbol{\xi} = \{\xi_1, \cdots, \xi_M\}$ which is a set of independent random variables. The values λ_i and $\Phi_i(E)$ are the eigenvalues and eigenvectors of the covariance matrix and are solutions to the following matrix equation:

$$\mathrm{Cov}_{\Sigma\Sigma}\Phi = \lambda\Phi \tag{13}$$

The expansion in equation (12) is optimal in the mean square sense. This means that when truncating the expansion after a number of terms $M_{KL} \leq M$ the resulting approximation of the random process minimizes the mean square error. This error decreases monotonically with the number of terms retained in the expansion at a rate that depends on the decay of the spectrum of the covariance matrix C. The higher the rate of spectral decay the smaller the number of terms needed in the expansion. Specifically, the number of terms to achieve a specified error threshold depends on the correlation structure of the random process. The more correlated the random process the smaller the number of terms needed to achieve the desired threshold. Conversely, if the random process is poorly correlated, a higher number of terms is needed.

The random variables in equation (12) can be found using the following expression:

$$\xi_i = \frac{1}{\sqrt{\lambda_i}} \int_\Gamma \Sigma(E, \boldsymbol{\xi})\Phi(E)dE \tag{14}$$

where it is straightforward to show that the variables ξ_i have zero mean and are mutually uncorrelated, namely

$$\mathbb{E}[\xi_i] = 0 \qquad \mathbb{E}[\xi_i\xi_j] = \delta_{ij} \tag{15}$$

where δ_{ij} is the Kronecker delta. However, even if the variables are uncorrelated, they are only independent for the case of a Gaussian process.

2.1.2. Transforming to Uncorrelated Standard Gaussian Variables

For the uncertainty quantification methods used in this work we must transform the set of correlated multi-group parameters, which we denote by $\boldsymbol{\Sigma} = \{\Sigma(E_g, \theta); g = 1, N_G\}$, to a set of uncorrelated standard Gaussian variables $\boldsymbol{\xi}$. The transform from non-Gaussian correlated variables to uncorrelated standard Gaussian variables is denoted as follows $\boldsymbol{\xi} = T(\boldsymbol{\Sigma})$ with the reverse transformation defined as $\boldsymbol{\Sigma} = T^{-1}(\boldsymbol{\xi})$. These transforms are generally non-linear and may be calculated with one of the following: Rosenblatt [32], Nataf [33], or Box-Cox [34]. To avoid these complicated transforms, which often require numerical treatment, we assume that the correlated variables $\boldsymbol{\Sigma}$ are either Gaussian or log-normally distributed. In theses cases closed form expressions for the transform T and its inverse can be found.

It is a well known fact that a linear combination of Gaussian random variates is itself a Gaussian variable. In this case the inverse transform T^{-1} will simply consist of a shift and a scaling of the set of standard Gaussian variables $\boldsymbol{\xi}$, namely

$$\Sigma_i = \mu_i + \sum_j L_{ij}\xi_j \tag{16}$$

here L is a coefficient matrix formed from a decomposition of the covariance matrix of the variables $\boldsymbol{\Sigma}$ and $\mu_i = \mathbb{E}[\Sigma_i]$ is the mean value of the Gaussian random variable Σ_i .

Using the Karhunen Loéve expansion from Eqn.(12), the components of the coefficient matrix L_{ij} are given by

$$L_{ij} = \sqrt{\lambda_i}\Phi_i(E_j) \tag{17}$$

If we now assume that Σ_i and ξ_k are log-normally distributed then we may transform back to the space of Gaussian variables by taking logarithms. Using this fact and Eqn.(16) we may now write

$$\ln(\Sigma_i) = \mu_i + \sum_{k=1}^{N} L_{ik}\ln(\xi_k) \tag{18}$$

or alternatively

$$\Sigma_i = \exp\left(\mu_i + \sum_{k=1}^{N} L_{ik}\ln(\xi_k)\right) \tag{19}$$

In Eqns.(18) and (19) μ_i is the mean value of the parameter in the space of Gaussian variables and L is the coefficient matrix in the space of Gaussian variables. Since $\ln(\boldsymbol{\xi})$ are Gaussian distributed, we note that Eqn.(19) is the inverse transform: $\boldsymbol{\Sigma} = T^{-1}(\ln(\boldsymbol{\xi}))$. In order to use Eqn.(19) we must calculate μ and L from the known mean and coefficient matrix in the space of log-normal variables. We denote the covariance matrix in the space of Gaussian variables and log-normal variables as $C^{(N)}$ and $C^{(LN)}$ respectively. The covariance matrix $C^{(N)}$ can be calculated from $C^{(LN)}$ using the following

$$C_{ij}^{(N)} = \ln\left[\frac{C_{ij}^{(LN)}}{\mathrm{E}[\Sigma_i]\mathrm{E}[\Sigma_j]} + 1\right] \tag{20}$$

Once the covariance matrix $C^{(N)}$ has been obtained, the mean values of the Gaussian process may be found with the following:

$$\mu_i = \ln\left[\frac{\mathrm{E}[\Sigma_i]^2}{\sqrt{C_{ii}^{(LN)} + \mathrm{E}[\Sigma_i]}}\right] \tag{21}$$

It is important to stress that the transform in Eqn.(20) is undefined for values of $\frac{C_{ij}^{(LN)}}{\mathrm{E}[\Sigma_i]\mathrm{E}[\Sigma_j]} \leq -1$.

2.1.3. Calculation of statistical moments via Latin Hypercube Sampling (LHS)

Once an appropriate representation of the random input parameters has been constructed, what remains is to calculate the uncertainty in the response parameter; which in this case is k_{eff}. One of the simplest and most robust techniques for performing this analysis is the Monte Carlo sampling method. Here the input parameters are randomly

sampled according to their probability distributions and a value of k_{eff} calculated for each realization of the input parameters. The statistical moments of this ensemble of response values can then be used as an approximation of the mean and variance due to the input uncertainty. The advantage of this approach is that the convergence rate is independent of the number of random dimensions in the problem. The downside, however, is that Monte Carlo sampling will require a large number of samples to achieve a good representation of the statistics of the input parameters. Combining this with a long execution time for each k_{eff} calculation makes this method undesirable.

An improvement to the basic Monte Carlo method is the Latin Hypercube sampling (LHS) method [35]. In this method each dimension is split into d equiprobable intervals forming a vector of possible samples. Each of the M vectors is then randomly shuffled. Pairing all M vectors then forms a set of equiprobable sampling locations distributed across the M dimensional space. Thus the number of samples is equal to the number of partitions d. This technique has been extensively used for uncertainty quantification in many fields for several decades. For a thorough review of applications and the history of the development of LHS see [36].

2.2. Nuclear Data Covariances

The uncertainties in the nuclear data are tabulated in an evaluated nuclear data library in the form of covariance matrices. For these data to be compatible with a reactor physics or criticality code, the file must be processed using a nuclear data processing code such as NJOY [37]. For the case of a deterministic criticality code, which uses the multi-group scheme, NJOY performs the necessary tasks to produce group condensed uncertainties in the form of covariance matrices. The covariance library used in this work was very much a "test" library compiled from various different nuclear data evaluations such as: ENDF/B-VI.8, ENDF/B-VII.0, JEF-2.2, JEFF-3.1, JENDL-3.3 and TENDL-2008. The library was produced with three group schemes: 3, 9 and 26 energy groups.

The covariance matrices available in the library were produced for each available nuclide (MAT number in ENDF format) and for every available reaction for that nuclide (MT number). NJOY also produces covariance matrices for the mutual covariances between reactions. However, no data is present for any potential correlations between nuclides which are therefore assumed to be independent.

The covariances are produced at infinite dilution, which means that the weighting spectrum does not include any effects of resonance absorption. Not considering these effects in the calculation of the covariances means that the uncertainties are overestimated. This is due to the damping effect that the resulting flux depressions from resonance absorption have on the multi-group constants. In an attempt to account for this, we made the assumption that the shielded and un-shielded covariances have the same relative uncertainties. In other words, the relative covariances produced by NJOY (see equation (7)) are used and re-normalised using mean values which have

been corrected for resonance absorption.

The mean values of the multi-group nuclear data were generated using the reactor physics code WIMS [38]. The Method of Characteristics code CACTUS [39], which is a part of the WIMS code suite, was used to condense the 172 group WIMS data library to the 3,9 and 26 groups required for the EVENT calculations. The subgroup method was used to calculate the effect of the resonance absorption for U235 , U238 and Pu239; for all other nuclides equivalence theory was used [40].

As we have mentioned, the covariances are produced per reaction per nuclide; this may not, however, correspond directly with the reaction cross sections used in a criticality code such as EVENT or CACTUS. Thus we must define the relationships between the reactions output by NJOY to those used in the reactor physics code and process the covariance data accordingly. The microscopic absorption and scattering cross sections for a given nuclide are defined in CACTUS and EVENT as

$$\sigma_a = \sigma_f + \sigma_c \tag{22}$$

$$\sigma_s = \sigma_e + \sigma_{in} + 2\sigma_{2n} + 3\sigma_{3n} \tag{23}$$

where we have defined the total capture cross sections σ_c as

$$\sigma_c = \sum_i \sigma_{c_i} - \sigma_{2n} - 2\sigma_{3n} \tag{24}$$

From the definition in (24) we can see that the total capture cross section is equal to the sum over many capture reactions such as (n,p), (n,d), (n,t) etc. For the sake of simplicity, and to minimise the amount of data that needs to be processed, we only consider the following MT numbers:

(n,n) : Elastic scattering σ_e (MT=2)
(n,n') : Inelastic scattering σ_{in} (MT=4)
$(n,2n)$: Production of two neutrons σ_{2n} (MT=16)
$(n,3n)$: Production of three neutrons σ_{3n} (MT=17)
(n,γ) : Radiative capture σ_g (MT=102)
(n,f) : Total fission σ_f (MT=18)
nu-bar : Average total number of neutrons per fission $\overline{\nu}$ (MT=452)

The total capture cross section is now reduced to the following:

$$\sigma_c = \sigma_g - \sigma_{2n} - 2\sigma_{3n} \tag{25}$$

We may now use Eqns.(22), (23) and (25) to produce appropriate covariances for use in calculations. This can be achieved using Eqn.(11), for example:

$$\mathrm{Cov}[\sigma_c, \sigma_f] = \mathrm{Cov}[\sigma_g, \sigma_f] - \mathrm{Cov}[\sigma_{2n}, \sigma_f] - 2\mathrm{Cov}[\sigma_{3n}, \sigma_f] \tag{26}$$

A final covariance matrix for each nuclide, which we denote C, is produced detailing the covariances of each reaction and the cross covariance between reactions. The symmetric matrix C has the following form:

$$C = \begin{pmatrix} \mathrm{Cov}[\sigma_c, \sigma_c] & \mathrm{Cov}[\sigma_c, \sigma_s] & \mathrm{Cov}[\sigma_c, \sigma_f] & \mathrm{Cov}[\sigma_c, \overline{\nu}] \\ \mathrm{Cov}[\sigma_s, \sigma_c] & \mathrm{Cov}[\sigma_s, \sigma_s] & \mathrm{Cov}[\sigma_s, \sigma_f] & \mathrm{Cov}[\sigma_s, \overline{\nu}] \\ \mathrm{Cov}[\sigma_f, \sigma_c] & \mathrm{Cov}[\sigma_f, \sigma_s] & \mathrm{Cov}[\sigma_f, \sigma_f] & \mathrm{Cov}[\sigma_f, \overline{\nu}] \\ \mathrm{Cov}[\overline{\nu}, \sigma_c] & \mathrm{Cov}[\overline{\nu}, \sigma_s] & \mathrm{Cov}[\overline{\nu}, \sigma_f] & \mathrm{Cov}[\overline{\nu}, \overline{\nu}] \end{pmatrix} \tag{27}$$

Examples of matrix C, normalised to give correlations, are given in Fig.(1) for U^{235} for the microscopic unmodified NJOY data. Here we can see that the $\overline{\nu}$ data is completely independent of all the reaction cross sections.

The eigenspectrum of the covariance matrix C defined above has been computed

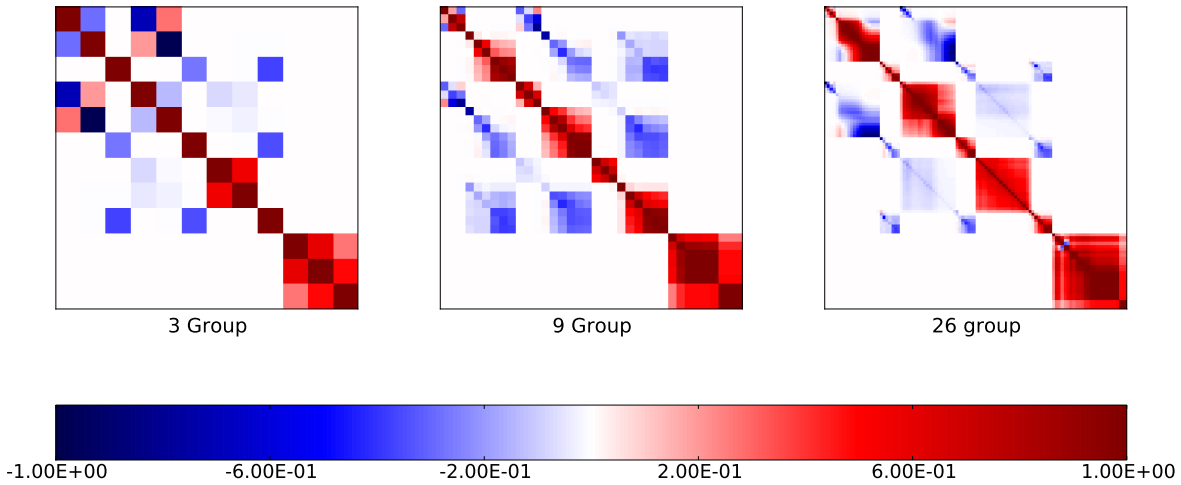


Figure 1: Correlation matrix C from Eqn.(27) of U^{235} for 3, 9 and 26 groups.

for hydrogen, Pu^{239}, Pu^{240}, U^{235} and U^{238}. These eigenspectra have been normalised and are plotted in cumulative form in Fig. (2). From this figure we can see that the eigenspectrum of hydrogen, H, decays very quickly and the majority of the variation may be expressed with the first eigenvalue. This is in contrast to U^{235}, for example, whose eigenspecturm has a much lower rate of decay with a much lower proportion of the variation contained within the first eigenmode.

3. Polynomial Chaos

One approach which has been widely used in the past few decades for uncertainty quantification is the method of polynomial chaos. This technique approximates the response of a model due to random input parameters using a spectral expansion of orthogonal polynomial basis functions. The basis functions are known as polynomial chaoses and are defined in a suitable finite dimensional basis. The spectral expansion of the random model response is known as a polynomial chaos expansion or PCE. The original polynomial chaos method, also termed homogeneous chaos or Wiener

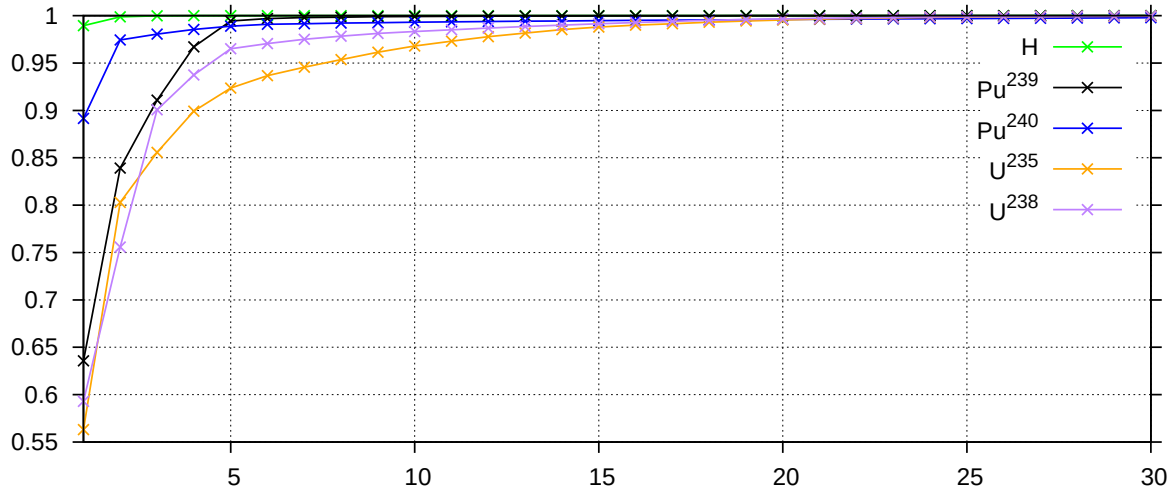


Figure 2: Cumulative graph of the eigenspectra of H, Pu^{239}, Pu^{240}, U^{235} and U^{238} for the covariance matrix defined by Eqn.(27). There are 26 energy groups for each reaction and nu-bar. Only the first 30 eigenvalues have been plotted.

chaos, was first developed by Wiener [8]. Homogeneous chaos is used to expand a second order random process in terms of Hermite polynomials using Gaussian random variables, namely

$$\begin{aligned} \mathrm{X}(\boldsymbol{x},\boldsymbol{\xi}) = \mathrm{x}_0(\boldsymbol{x})\Gamma_0 &+ \sum_{i_1=1}^{\infty} \mathrm{x}_{i_1}(\boldsymbol{x})\Gamma_1(\xi_{i_1}) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} \mathrm{x}_{i_1 i_2}(\boldsymbol{x})\Gamma_2(\xi_{i_1},\xi_{i_2}) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} \mathrm{x}_{i_1 i_2 i_3}(\boldsymbol{x})\Gamma_3(\xi_{i_1},\xi_{i_2},\xi_{i_3}) \\ &+ \cdots \end{aligned} \tag{28}$$

where $\Gamma_n(\xi_{i_1},\cdots,\xi_{i_n})$ denotes the polynomial chaos of order n. For notational convenience we may write

$$\mathrm{X}(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}^M} \mathrm{x}_{\boldsymbol{\alpha}}(\boldsymbol{x})\Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \tag{29}$$

where there is a one-to-one correspondence between the functions $\Gamma_n(\xi_{i_1},\cdots,\xi_{i_n})$ and $\Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$. Here $\mathcal{A}^M$ is the set of all polynomial indexes which is defined as $\mathcal{A}^M \equiv \{\boldsymbol{\alpha}\in\mathbb{N}^M\}$ where M is the number of dimensions. The multi-index $\boldsymbol{\alpha}$ is defined as $\boldsymbol{\alpha} = (\alpha_1,\cdots,\alpha_M)$ where α_i is the index of a univariate polynomial; this results in the following form for the functions $\Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$:

$$\Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \prod_{i=1}^{M} \Phi_{\alpha_i}(\xi_i) \tag{30}$$

which is a product of the appropriate one-dimensional polynomial basis. These basis functions are orthogonal which implies

$$\langle \Phi_{\boldsymbol{\alpha}}, \Phi_{\boldsymbol{\beta}} \rangle = \prod_{i=1}^{M} N_{\alpha_i}^2 \delta_{\alpha_i \beta_j} \tag{31}$$

where $\delta_{\alpha_i \beta_j}$ is the Kronecker delta and $N_{\alpha_i}^2$ is a normalisation constant. The term $\langle \cdot, \cdot \rangle$ denotes the ensemble average which is the inner product in the Hilbert space of variables $\boldsymbol{\xi}$ and defined as

$$\langle \mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi}), \mathrm{Y}(\boldsymbol{x}, \boldsymbol{\xi}) \rangle = \int \mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi}) \mathrm{Y}(\boldsymbol{x}, \boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{32}$$

where $p(\xi)$ is the probability density function of the random variables ξ.

For the purposes of numerical computation, the series given by Eqn.(29) must be truncated. The most commonly used approach is to restrict the maximum total order of the multivariate polynomial basis to $\leq p$. This imposes the following restriction on the index sets

$$\mathcal{A}^{M,p} \equiv \{ \boldsymbol{\alpha} \in \mathbb{N}^M : ||\boldsymbol{\alpha}||_1 \leq p \} \quad ; \quad ||\boldsymbol{\alpha}||_1 \equiv \sum_{i=1}^{M} \alpha_i$$

The response function $\mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi})$ is now approximated by the following expansion:

$$\mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi}) \approx \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{M,p}} \mathrm{x}_{\boldsymbol{\alpha}}(\boldsymbol{x}) \Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \tag{33}$$

For a number of random dimensions M and maximum polynomial order p the total number of coefficients in the expansion given by (33) is as follows

$$N_P + 1 = \frac{(M+p)!}{M!p!} = \binom{M+p}{M} \tag{34}$$

Equation (34) shows that there is an exponential increase in the number of coefficients in the PCE with the number of dimensions M. Using this index set requires many thousands of coefficients even for a modest number of dimensions. This method of constructing the PCE is completely ignorant of the stucture of the stochastic process which it is representing. In other words, polynomials will be present in dimensions which may have little or no variation with the dependant variable ξ, the converse may also be the case for dimensions with a significant amount of variation. This results in a polynomial expansion which may have a large amount of redundancy in some dimensions and an inadequate representation in others.

According to the Cameron-Martin theorem [41], the homogeneous chaos expansion can approximate any functionals in the continuously differentiable L_2 space and

converges in the L_2 sense. This means that the homogeneous chaos expansion converges to any stochastic process with finite second order moments. However, it has been demonstrated that the optimal convergence rate is obtained only for Gaussian processes. For non-Gaussian processes an extension to Wiener's chaos using the Askey family of orthogonal polynomials has been proposed [9]. Here the underlying random variables are chosen according to the weighting function of the polynomial chaos. For example, for a Gaussian probability weight the Hermite polynomials are used as in Wiener's original work and for a uniform probability weight the Legendre polynomials are used. The normalisation coefficients from Eqn.(31) and associated weight functions for these polynomials are as follows:

$$N_i = \frac{1}{2i+1} \qquad p(\xi) = \frac{1}{2} \quad \xi \in [-1:1] \tag{35}$$

$$N_i = i! \qquad p(\xi) = \frac{1}{\sqrt{2\pi}} \exp\left\{\frac{-x^2}{2}\right\} \quad \xi \in [-\infty:\infty] \tag{36}$$

If the polynomial chaos expansion is used with its corresponding stochastic process, optimal (exponential) convergence is observed [9] with respect to the polynomial order p. This extended method is known as Wiener-Askey chaos or generalised polynomial chaos (gPC). In the case where the uncertain input parameters are described using different probability distributions, a mixture of the optimal polynomials may be used to construct a basis [42].

The main benefit of representing the response variables in the semi-analyitcal form of Eqn.(29) is the ability it provides to perform analytic manipulations on the solution. Once the coefficients in the PCE are known, the mean, variance and covariance may be calculated using the following relations:

$$\overline{\mathrm{X}}(\boldsymbol{x}) = \mathrm{x}_0(\boldsymbol{x}) \tag{37}$$

$$\begin{aligned} \mathrm{Var}(\boldsymbol{x}) &= \left(\sum_{\boldsymbol{\alpha} \in \mathcal{A}^{M,p}} \mathrm{x}_{\boldsymbol{\alpha}} \Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \right)^2 - \overline{\mathrm{X}}(\boldsymbol{x})^2 \\ &= \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{M,p}} \mathrm{x}^2_{\boldsymbol{\alpha}}(\boldsymbol{x}) N^2_{\boldsymbol{\alpha}} \end{aligned} \tag{38}$$

$$\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{x'}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}^{M,p}} \mathrm{x}_{\boldsymbol{\alpha}}(\boldsymbol{x}) \mathrm{x}_{\boldsymbol{\alpha}}(\boldsymbol{x'}) N^2_{\boldsymbol{\alpha}} \tag{39}$$

here $\mathcal{A}^{M,p}$ in this case is an arbitrary index set which may be formed using either Eqn.(33) or iteratively using the techniques defined in Section (3.2) and Section (4.2). Expressions for higher order moments such as skew and kurtosis may also be calculated [43].

To obtain a graphical representation of the PDF, the random response X represented by a PCE may be simulated using a Monte Carlo scheme. This provides an ensemble of response quantities $\{\mathrm{X}_1, \cdots, \mathrm{X}_N\}$ which are then used to construct a histogram. In practice a large number of samples are required to obtain an accurate representation, say N=10,000 - 100,000.

3.1. Non-Intrusive Spectral Projection

The Non-Intrusive Spectral Projection (NISP) approach computes the coefficients in a PCE by minimising the mean square approximation error of the model response $\mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi})$ by the polynomial chaos surrogate. The coefficients in the PCE are obtained by using the following expression:

$$\mathrm{x}_{\boldsymbol{\alpha}} = \frac{\mathbb{E}[\mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi})\Phi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})]}{\mathbb{E}[\Phi^2_{\boldsymbol{\alpha}}(\boldsymbol{\xi})]} \quad \forall\ \boldsymbol{\alpha} \in \mathcal{A}^{M,p} \tag{40}$$

The denominator in Eqn.(40) is given by the normalisation constant for the particular polynomial basis. The numerator, however, involves a multidimensional integral of following form

$$\boldsymbol{I}(f) = \int_{\Omega^M} f(\boldsymbol{x}) d\boldsymbol{x} \tag{41}$$

where Ω is the M dimensional domain of integration. The integrals are approximated using a quadrature rule which is a weighted sum of the function $f(\boldsymbol{x})$ evaluated at different points (nodes) in the parameter space. This approximation is written as follows:

$$\begin{aligned} \boldsymbol{I}(f) &\approx \sum_{i=1}^{n} w_i f(\boldsymbol{x}_i) \\ &= U_n f \end{aligned} \tag{42}$$

where U_n denotes a quadrature rule with n points.

We now draw on two important distinctions when deciding between quadrature rules for this work; whether successive quadrature rules are nested and whether the nodes in each rule are constrained. A sequence of quadrature rules is nested if the nodes of U_{n_i} are a subset of the nodes of U_{n_j} for $j > i$. Nested schemes allow the re-use of existing function evaluations which is important for implementing efficient adaptive strategies.

The accuracy of the integration scheme is obviously important and a rule is said to have a degree of polynomial exactness equal to d if it is exact for all polynomials of degree $\leq$ d. The degree of polynomial exactness for constrained rules lies somewhere between $n \leq \deg(U_n) \leq 2n-1$ depending on the order of the interpolating polynomials. Examples of constrained methods are Newton-Cotes type rules where the nodes are chosen to be equally spaced. The advantage of using a constrained rule is that it may be nested by construction. The optimal degree of accuracy, $2n-1$, is obtained with unconstrained, Gaussian-type rules.

Gaussian type rules are available for [0,1] (Gauss-Legendre) and $\mathcal{R}$ (Gauss-Hermite) but are not nested. For Gauss-Legendre rules, an n+1 point extension to an n-point rule is available [44](known as a Konrod extension or Gauss-Patterson rule). This

produces a 2n+1 point rule that is nested by construction. This rule has a degree of accuracy $\deg(U_n) = \frac{1}{2}(3n+1)$ for odd n which is lower than the regular Gauss-Legendre rule but does have the advantage of being nested. An analogous procedure for the Gauss-Hermite rule on $\mathcal{R}$ has been proposed [45] and is known as the Genz-Keister rule. This method is available for rules with n=1,3,9,19 and 35 and has an accuracy of $\deg(U_n) \approx \frac{3}{2}n$.

An alternative parameter used to characterise a quadrature rule is the level. The number of nodes corresponding to a given level, which we denote L, depends on whether a quadrature rule is nested and whether it is open, closed or semi-open. For a closed rule, nodes are allowed on the boundary of the domain, whereas, open rules never have nodes on the boundary. Semi-open rules are a combination of both open and closed. For a given level, in one dimension, the number of nodes, n_L, is calculated using a growth rule:

$$\begin{array}{ll} 2^L + 1 & \text{Closed rules} \\ 2^{L+1} - 1 & \text{Open rules} \end{array} \qquad L \geq 1 \tag{43}$$

For $L = 0 \quad , n_L = 1$.

3.1.1. Smolyak Sparse Grids

The most straight-forward way to extend a univariate quadrature rule to M dimensions is to use a tensor product. For a number of points in the univariate rule n this results in n^M points in the multivariate rule. This exponential growth in the number of points limits the use of this method to low numbers of dimensions. To reduce the number of points in a multivariate quadrature rule without adversely affecting the accuracy, the sparse grid construction was proposed [14]. Sparse grids are built upon tensor products of hierarchical difference sets of one dimensional quadrature rules, U_k, as follows:

$$\Delta_k := U_k - U_{k-1} \quad \text{with} \quad U_0 = 0 \tag{44}$$

for $k \geq 1$. A univariate quadrature rule can then be built from a summation of these difference sets, namely

$$U_L = \sum_{m=1}^{L} \Delta_m \tag{45}$$

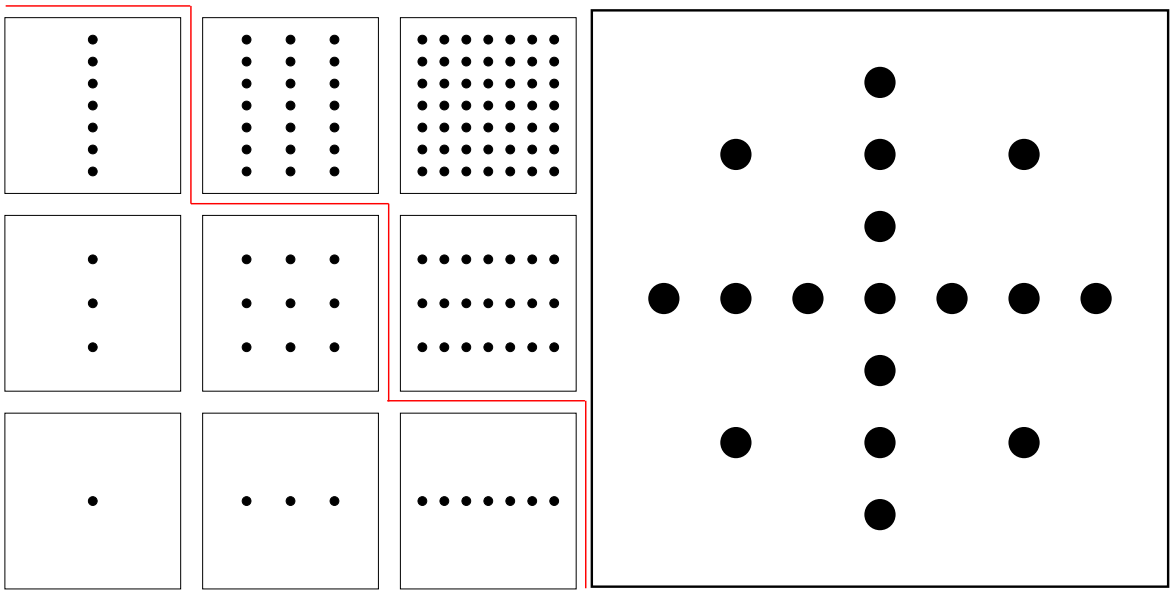
The Smolyak construction $A(L, M)$ for a M dimensional quadrature rule of total level L is

$$A(L, M) = \sum_{|\boldsymbol{k}|_1 \leq M+L-1} \bigotimes_{i=1}^{M} \Delta_{k_i} \tag{46}$$

here $|\boldsymbol{k}|_1 = \sum_{i=1}^{M} k_i$ is the 1-norm. Thus, out of all the possible sets $\boldsymbol{k} \in \mathbb{N}$, only those are chosen whose 1-norm is smaller than a constant. For numerical purposes it is more

convenient to rewrite (46) in terms of the univariate rules U_k. The tensor product of the difference sets $\Delta_{\boldsymbol{k}}$ for an index set $\boldsymbol{k}$ is given by [46]

$$\bigotimes_{i=1}^{M}(U_{k_i} - U_{k_i-1}) = \sum_{\boldsymbol{\alpha}\in\{0,1\}^M} (-1)^{|\boldsymbol{\alpha}|_1} \bigotimes_{i=1}^{M} U_{k_i-\alpha_i} \tag{47}$$



(a) (b)

Figure 3: Example of a Smolyak sparse grid construction with $L = 3$ using a Newton-Cotes formula. Allowed indices are below the red line.

If we let $k_i = j_i + \alpha_i$, the right hand side of Eqn.(46) can be replaced with (47) provided the index set $\boldsymbol{\alpha}$ is subject to the following constraint: $|\alpha| \leq M + L - 1 - |\boldsymbol{j}|$. Equation (46) is now written

$$A(L, M)f = b(M + L - 1 - |\boldsymbol{k}|, M) \left(\bigotimes_{i=1}^{M} U_{k_i} \right) f \tag{48}$$

where

$$\begin{aligned} b(a, b) &= \sum_{\boldsymbol{\alpha}\in\{0,1\},|\boldsymbol{\alpha}|\leq a} (-1)^{|\boldsymbol{\alpha}|} \\ &= (-1)^a \binom{b-1}{a} \end{aligned} \tag{49}$$

Using (49), (48) now has the form

$$A(l, M)f = \sum_{M\leq|\boldsymbol{k}|_1\leq L+M-1} (-1)^{M+L-1-|\boldsymbol{k}|} \binom{M-1}{|\boldsymbol{k}|-L} \left(\bigotimes_{i=1}^{M} U_{k_i} \right) f \tag{50}$$

The construction of a sparse grid using Eqn.(50) is illustrated in Fig.(3). Here we can see that only those indices on the simplex $|\boldsymbol{k}|_1 \leq M + L - 1$ are used in the construction. Since $M = 2$ and $L = 3$, the norm $|\boldsymbol{k}|_1 \leq 2$ thus only the following indices are allowed: $(1,1)$; $(1,2)$; $(1,3)$; $(2,2)$; $(2,1)$; $(3,1)$.

3.1.2. Adaptive Sparse Grids

For highly anisotropic multivariate functions, the ability to use a different order quadrature scheme along different dimensions can improve both accuracy and efficiency. The fundamental idea behind adaptive sparse grids is to place different emphasis upon different dimensions. One approach is to replace the conventional sparse grid index set: $|\boldsymbol{k}|_1 = L + M - 1$ with $a \cdot \boldsymbol{k} \leq L + M - 1$ where a is a weight vector [47]. Here the weights were chosen based on the exponential convergence rate of the integral in each direction with respect to the number of quadrature nodes. This method was shown [47] to perform well (using Clenshaw-Curtis rules) for problems up to 121 dimension when compared with conventional Monte Carlo sampling approaches. However, it may not be possible to estimate the vector a accurately *a-priori* and *a-posteriori* error estimates can be difficult to evaluate for practical problems. An alternative approach proposed in [15, 16] uses a self adaptive algorithm to iteratively build the optimum index set for the required integral. This construction leads to what is known as a generalised sparse grid.

To describe the theory behind the generation of a generalised sparse grid, it is useful to firstly define the admissibility condition on the index sets, viz

$$\boldsymbol{k} - \boldsymbol{e}_j \in \mathcal{I} \quad \text{for} \quad 1 \leq j \leq M, \quad k_j > 1 \tag{51}$$

where $\boldsymbol{e}_j$ is a unit vector in the direction j and $\mathcal{I}$ is the set of all chosen indices $\boldsymbol{k}$. Equation (51) says that, for a new index $\boldsymbol{k}$, the previous order along each direction in $\boldsymbol{k}$ must already exist for the new index to be allowed. As an example we consider a graphical representation of a two dimensional adaptive integration in Figure (4). Here active and current indexes are only placed in positions when their left and bottom faces are connected to a member of the old index set, i.e. a member of the old set is in the backward neighbourhood of the proposed index. Admissible forward neighbours are denoted with an arrow.

With the admissibility condition in (51), the generalised sparse grid is defined as

$$A(\mathcal{I}) = \sum_{\boldsymbol{k} \in \mathcal{I}} \bigotimes_{i=1}^{M} \Delta_{k_i} \tag{52}$$

Equation (52) is not immediately useful in its current form, it is more practical to use Eqn.(47) and express $A(\mathcal{I})$ in terms of the univariate quadratures U_l, namely

$$A(\mathcal{I}) = \sum_{\boldsymbol{k} \in \mathcal{I}} \left(\sum_{\boldsymbol{\alpha} \in \{0,1\}^M} (-1)^{|\boldsymbol{\alpha}|_1} \chi_{\mathcal{I}}(\boldsymbol{k} + \boldsymbol{\alpha}) \right) \bigotimes_{i=1}^{M} U_{k_i} \tag{53}$$

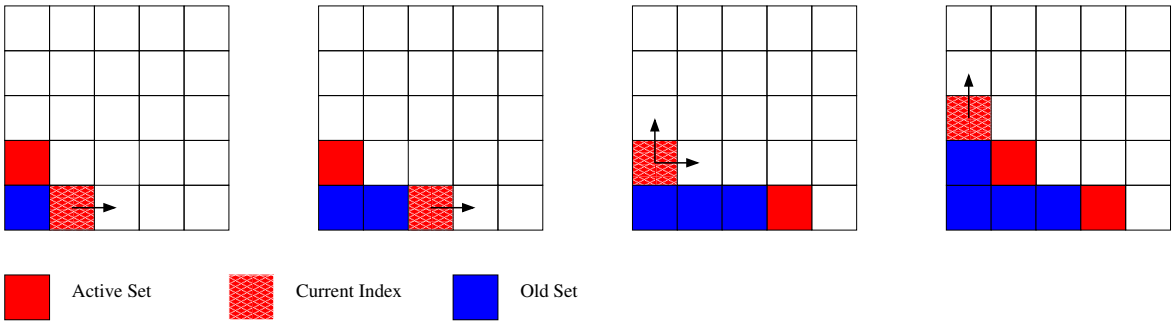


Figure 4: Example of an adaptive sparse grid procedure showing the active and old index sets. The current index is also shown with the admissible forward neighbours.

where the indicator variable $\chi_{\mathcal{I}}$ is defined as

$$\chi_{\mathcal{I}}(\boldsymbol{k}) = \begin{cases} 1 & \text{if } \boldsymbol{k} \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{54}$$

In order for the adaptive algorithm to be successful, there must be some way to predict the importance of an index $\boldsymbol{k}$. In essence, these importance or error measures control the adaptive refinement of the algorithm. If the error measure of an index $\boldsymbol{k}$ is below some threshold then there will be no more refinement in the forward direction of this index. This criterion assumes that the error associated with an index decreases monotonically in the forward direction.

The error indicator for index $\boldsymbol{k}$ is denoted by $g_{\boldsymbol{k}}$ and depends on the differential integral of the function we are integrating. The differential integral is given by

$$\Delta_{\boldsymbol{k}} f = U_{\boldsymbol{k}} f - U_{\boldsymbol{k-1}} f \tag{55}$$

which is the value of the integral including the new index $\boldsymbol{k}$ less the previously calculated indexes. In the majority of uncertainty studies, it is the variance of a function which is the quantity of interest and therefore the adaptive procedure must be tuned to calculate this quantity. As such, we also use the square of the differential integral, namely

$$\Delta_{\boldsymbol{k}} f^2 = U_{\boldsymbol{k}} f^2 - U_{\boldsymbol{k-1}} f^2 \tag{56}$$

The quantity in (56) tells us by how much the square of the integral has changed when the newly calculated index $\boldsymbol{k}$ has been added. The final accuracy of the integral of this absolute measure is of course problem dependent and may result in large relative errors if the value of the integral is small. As such, we have chosen to use the square of the differential integral relative to the square of the integral, viz

$$g_{\boldsymbol{k}} = \frac{\Delta_{\boldsymbol{k}} f^2}{U_{\boldsymbol{k-1}} f^2} \tag{57}$$

In [15] the authors defined an adaptive generalised sparse grid algorithm for computing high dimensional integrals. We reproduce this algorithm in Algorithm(1) for convenience, the only fundamental difference being the use of a different error measure. In words, the algorithm proceeds as follows: Firstly, the active set, $\mathcal{A}$, is initialised with the index corresponding to the nominal value of the function which is assumed to have already been calculated. The old index set, $\mathcal{O}$, is empty, the total error, r, is set to the error of the nominal index, and the integral value, I, is initialised with the contribution from the nominal index. The adaptive algorithm will proceed while the total error is less than a predefined threshold δ. To begin with, within the iterative loop, the index with the largest error, $g_{\boldsymbol{k}}$, is chosen from the active set. This index is then removed from the active set, appended to the old set and its associated error removed from the total error. The next part of the algorithm computes the admissible forward neighbours of the current index. Each allowed index is added to the active set, its contribution to the value of the integral is appended and its associated error, which is calculated using Eqn.(57), added to the total.

Algorithm 1 Adaptive sparse grid

$\boldsymbol{k} = (0, 0, \cdots, 0)$
$\mathcal{O} = \emptyset$
$\mathcal{A} = \{\boldsymbol{k}\}$
$r = g_{\boldsymbol{k}}$
$I = \Delta_{\boldsymbol{k}} f$
while $r < \delta$ **do**
  Choose $\boldsymbol{k} \in \mathcal{A}$ with largest $g_{\boldsymbol{k}}$
  $\mathcal{O} = \mathcal{O} \cup \boldsymbol{k}$
  $\mathcal{A} = \mathcal{A} \setminus \boldsymbol{k}$
  $r = r - g_{\boldsymbol{k}}$
  for l=1 **to** M **do**
    $\boldsymbol{j} = \boldsymbol{i} + \boldsymbol{e}_l$
    if $\boldsymbol{j} - \boldsymbol{e}_q \in \mathcal{O}$ **for all** $q = 1, \cdots, M$ **then**
      $\mathcal{A} = \mathcal{A} \cup \{\boldsymbol{j}\}$
      $I = I + \Delta_{\boldsymbol{k}} f$
      $r = r + g_{\boldsymbol{j}}$
    end if
  end for
end while

3.2. Combination with Polynomial Chaos

When the adaptive generalised sparse grid algorithm has finished successfully, it yields the integral of the function f using a sparse grid index set which is optimal with respect to the error measure $g_{\boldsymbol{k}}$. This set of indexes may be used to infer information about the structure of the integrand in question. i.e. a larger number of quadrature points selected along one direction suggests a higher polynomial behaviour than a di-

rection with fewer quadrature points.

In this section we propose a method of building a PCE directly from the sparse grid quadrature levels. From the level indexes of the sparse grid quadrature rule we infer the orders of the polynomials in the PCE. To do this, we start with the polynomial exactness of the univariate quadrature methods from section (3.1), viz

$$\deg(U_n) = \frac{1}{2}(3n+1) \qquad \text{Gauss-Patterson} \tag{58}$$

$$\deg(U_n) = \frac{3}{2}n \qquad \text{Genz-Keister} \tag{59}$$

where n is the number of points in the quadrature rule. We now insert the expression for the growth rules from Eqn.(43) into (58) and (59) which yields

$$\deg(U_n) = 3\left(2^L - \frac{1}{3}\right) \qquad \text{Gauss-Patterson} \tag{60}$$

$$\deg(U_n) = 3\left(2^L - \frac{1}{2}\right) \qquad \text{Genz-Keister} \tag{61}$$

we now have an expression for the degree of exactness of the quadrature rule in terms of its level L. Since the quadrature rules are being used to perform a projection of a random response on to a polynomial basis, the maximum polynomial order of the integrand will be twice that of the basis we are projecting onto. In other words, the maximum polynomial order of the basis is half the degree of polynomial exactness of the quadrature rule. We now have the following expressions

$$p = 3\left(2^{L-1} - \frac{1}{6}\right) \qquad \text{Gauss-Patterson} \tag{62}$$

$$p = 3\left(2^{L-1} - \frac{1}{4}\right) \qquad \text{Genz-Keister} \tag{63}$$

The growth rate of the polynomial order in Eqns.(62) and (63) is too fast and results in an excessive number of polynomials. Therefore, based upon the above analysis, we use the following expression for the growth rate of the polynomial order

$$p = 2^{L-1} \tag{64}$$

Equation (64) is an underestimate of the total polynomial exactness of the quadrature scheme but we will show that it still provides sufficient accuracy at a reduced number of polynomials.

4. High Dimensional Model Representation (HDMR)

In this section we introduce the basic properties of the HDMR expansion and show how it may be combined with a polynomial chaos expansion to produce a surrogate model of the response surface $\mathrm{X}(\boldsymbol{\xi})$. First, let $\mathrm{X}(\boldsymbol{\xi}) : \mathbb{R}^M \to \mathbb{R}$ be a real valued smooth

multivariate stochastic function where $\boldsymbol{\xi} = \{\xi_1, \cdots, \xi_M\}$ are a set of IID random variables with M being the number of stochastic dimensions. The stochastic function $\mathrm{X}(\boldsymbol{\xi})$ may also be a function of some other deterministic parameter space, i.e. $\mathrm{X}(\boldsymbol{x}, \boldsymbol{\xi})$, but we omit the dependence on $\boldsymbol{x}$ for now to simplify the notation. $\mathrm{X}(\boldsymbol{\xi})$ may also be multivariate, i.e. $\mathrm{X}(\boldsymbol{\xi}) = \{\mathrm{X}_1(\boldsymbol{\xi}), \cdots, \mathrm{X}_I(\boldsymbol{\xi})\}$, but since the focus of this work is on criticality calculations we need only consider a univariate response.

The influence of the input variables on the response can be independent and/or cooperative. Therefore, in some situations, it is more convenient to express the multivariate function in terms of a hierarchical correlated function expansion in terms of the input variables. This is known as the Sobol expansion or the HDMR expansion and is written [18, 19, 20, 21, 22, 23] as:

$$\mathrm{X}(\boldsymbol{\xi}) = \mathrm{x}_0 + \sum_{i=1}^{M} \mathrm{x}_i(\xi_i) + \sum_{1 \le i_1 < i_2 \le M} \mathrm{x}_{i_1 i_2}(\xi_{i_1}, \xi_{i_2}) + \cdots + \mathrm{x}(\xi_{i_1}, \xi_{i_2}, \cdots, \xi_{i_M}) \tag{65}$$

The term $\mathrm{x}_i(\xi_i)$ is a first order term representing the effect of the variable ξ_i acting independently, and in general, non-linearly upon the response $\mathrm{X}(\boldsymbol{\xi})$. The function $\mathrm{x}_{i_1 i_2}(\xi_{i_1}, \xi_{i_2})$ is a second order term that describes the cooperative effects of the variables ξ_{i_1} and ξ_{i_2} on the response. Higher order terms represent the effects of an increasing number of variables with the last function describing the cooperative effect between all M parameters. Usually the higher order terms in Eqn.(65) are negligible [22] so that typically HDMR with only a few low order terms is adequate to describe the output behaviour. Inclusion of only the first and the second order terms results in fast convergence of the expansion. However, the importance of higher order terms is of course problem dependent.

From now the derivation will follow the notation given in [25]. Eqn.(65) is written using a more compact notation as follows

$$\mathrm{X}(\boldsymbol{\xi}) = \sum_{\boldsymbol{u} \subseteq \mathcal{D}} \mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}}) \tag{66}$$

for a given set $\boldsymbol{u} \subseteq \mathcal{D}$, where $\mathcal{D} := \{1, \cdots, M\}$ denotes the set of coordinate indices and $\mathrm{x}_{\emptyset}(\boldsymbol{\xi}_{\emptyset}) = \mathrm{x}_0$. The cardinality of the set $\boldsymbol{u}$ is defined as ν and $\boldsymbol{\xi}_{\boldsymbol{u}}$ denotes the ν dimensional vector containing those components of $\boldsymbol{\xi}$ that belong to the set $\boldsymbol{u}$, i.e. $\boldsymbol{\xi}_{\boldsymbol{u}} = (\xi_i)_{i \in \boldsymbol{u}}$. The cardinality ν is also referred to as the order of the HDMR component function.

Now let us define the following

$$d\mu(\boldsymbol{\xi}) = \prod_{i=1}^{M} d\mu(\xi_i) \tag{67}$$

which is an M dimensional product measure defined on Borel subsets of Ω^M. The measure $d\mu$ determines the particular form of the component functions and induces the projection operator $P_{\boldsymbol{u}} : \Omega^M \to \Omega^\nu$, namely

$$P_{\boldsymbol{u}}\mathrm{X}(\boldsymbol{\xi}_{\boldsymbol{u}}) = \int_{\Omega^{M-\nu}} \mathrm{X}(\boldsymbol{\xi}) p_{\mathcal{D}/\boldsymbol{u}}(\boldsymbol{\xi}) d\mu_{\mathcal{D}/\boldsymbol{u}}(\boldsymbol{\xi}) \tag{68}$$

where $d\mu_{\mathcal{D}/\boldsymbol{u}}(\boldsymbol{\xi}) := \prod_{i\in\mathcal{D}, i\notin\boldsymbol{u}} d\mu_i(\xi_i)$ and $p_{\mathcal{D}/\boldsymbol{u}}(\boldsymbol{\xi}) := \prod_{i\in\mathcal{D}, i\notin\boldsymbol{u}} p_i(\xi_i)$. Here $p(\xi_i)$ is the probability density function of the random variable ξ_i. The terms $\mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})$ in (66) can also be defined recursively by

$$\mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}}) = P_{\boldsymbol{u}}\mathrm{X}(\boldsymbol{\xi}_{\boldsymbol{u}}) - \sum_{\boldsymbol{v}\subset\boldsymbol{u}} \mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}}) \tag{69}$$

At this point we note that there are several forms of HDMR associated with different measures. In [21] the Lebesgue measure $d\mu(\boldsymbol{\xi}) = d(\boldsymbol{\xi}) = \prod_{i=1}^M d\xi_i$ is used. With this choice of measure the decomposition in Eqn.(65) is known as the ANOVA-HDMR where ANOVA stands for the ANAlysis Of VAriance. A significant drawback of ANOVA-HDMR is the need to compute high dimensional integrals, for example, the computation of the zero order term x_0 takes the form

$$\mathrm{x}_0 = \int_{\Omega^M} \mathrm{X}(\xi_1, \cdots, \xi_M) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{70}$$

which is an M dimensional integral. The ANOVA method is usually computed using sampling or the Fourier amplitude sensitivity test [48] but more recently has been computed using the coefficients of the polynomial chaos expansion [49]. To avoid the large number of multidimensional integrals involved in ANOVA-HDMR, a more computationally efficient method known as the cut-HDMR method was developed [21, 22]. In this method the measure is chosen as the Dirac measure located at a reference point $\boldsymbol{c} \in \Omega^M$, i.e. $d\mu(\boldsymbol{\xi}) = \prod_{i=1}^M \delta(\xi_i - c_i) d\xi_i$ which, from (68), induces the following projections

$$P_{\boldsymbol{u}}\mathrm{X}(\boldsymbol{\xi}_{\boldsymbol{u}}) = \mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}} \tag{71}$$

where we use the notation $\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\xi_i} = \mathrm{X}(c_1, \cdots, \xi_i, \cdots, c_M)$. The terms of the cut-HDMR decomposition are thus related to the terms of the classical ANOVA decomposition in the sense that all integrals are replaced by point evaluations at a fixed anchor point $\boldsymbol{c} \in \Omega^M$. As such, this approach is also referred to as anchored-ANOVA. This decomposition represents $\mathrm{X}(\boldsymbol{\xi})$ as a superposition of its values on lines, faces and hyperplanes, etc, which intersect the cut point $\boldsymbol{c}$ and are parallel to the coordinate axes. In the convergence limit, where all the correlated functions in Eqn.(65) are considered, cut-HDMR is invariant to the choice of the reference point $\boldsymbol{c}$. However, in practice, the choice of reference point $\boldsymbol{c}$ is very important for cut-HDMR especially if only low order interactions are considered. In [50] the authors show the optimality of

different reference points (also called anchor points) using numerical simulation. Other authors [19, 51] suggest that the cut-HDMR expansion usually leads to a satisfactory approximation within a desired accuracy if the reference point is chosen as the mean vector.

The component functions in the HDMR expansion are orthogonal with respect to the inner product induced by the measure μ, namely

$$\int \mathrm{X}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})\mathrm{X}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})d\mu(\boldsymbol{\xi}) \quad \text{for} \quad \boldsymbol{u} \neq \boldsymbol{v} \tag{72}$$

Computing the expectation of the cut-HDMR expansion only requires multidimensional integration equal to the order of the component function, namely, the mean value of the response function can be computed using

$$\begin{aligned} \mathbb{E}[\mathrm{X}(\boldsymbol{\xi})] &= \sum_{\boldsymbol{u}\subseteq\mathcal{D}} \mathbb{E}[\mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})] \\ &= \sum_{\boldsymbol{u}\subseteq\mathcal{D}} \overline{\mathrm{x}}_{\boldsymbol{u}} \end{aligned} \tag{73}$$

where

$$\overline{\mathrm{x}}_{\boldsymbol{u}} = \mathbb{E}[\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}] - \sum_{\boldsymbol{v}\subset\boldsymbol{u}} \overline{\mathrm{x}}_{\boldsymbol{v}} \tag{74}$$

In other words, a third order cut-HDMR component will require a three dimensional integration. The variance of the response function $\mathrm{X}(\boldsymbol{\xi})$ may also be represented using a hierarchical expansion, to derive this we start with the familiar expression for the variance:

$$\mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})] = \mathbb{E}[\mathrm{X}^2(\boldsymbol{\xi})] - \mathbb{E}[\mathrm{X}^2(\boldsymbol{\xi})]^2. \tag{75}$$

Inserting (66) into (75) we have

$$\mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})] = \mathbb{E}\left[\left(\sum_{\boldsymbol{u}\subseteq\mathcal{D}} \mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})\right)^2\right] - \mathbb{E}\left[\left(\sum_{\boldsymbol{u}\subseteq\mathcal{D}} \mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})\right)\right]^2 \tag{76}$$

Using the recurrence relation from Eqn. (69) and the Dirac measure for the cut HDMR expansion from (71) we have

$$\begin{aligned} \mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})] = \mathbb{E}&\left[\left(\sum_{\boldsymbol{u}\subseteq\mathcal{D}} \left(\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}} - \sum_{\boldsymbol{v}\subset\boldsymbol{u}} \mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right)\right)^2\right] \\ &- \mathbb{E}\left[\sum_{\boldsymbol{u}\subseteq\mathcal{D}} \left(\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}} - \sum_{\boldsymbol{v}\subset\boldsymbol{u}} \mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right)\right]^2. \end{aligned} \tag{77}$$

Expanding the squares in (77) and using the orthogonality relation from Eqn.(72), we have

$$\mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})] = \sum_{\boldsymbol{u}\subseteq\mathcal{D}} \left(\mathbb{E}\left[\mathrm{X}^2(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}\right] - \mathbb{E}\left[\mathrm{X}^2(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}\right]\right) \tag{78}$$
$$- \sum_{\boldsymbol{u}\subseteq\mathcal{D}} \left(\mathbb{E}\left[\sum_{\boldsymbol{v}\subset\boldsymbol{u}} \mathrm{x}_{\boldsymbol{v}}^2(\boldsymbol{\xi}_{\boldsymbol{v}})\right] - \mathbb{E}\left[\sum_{\boldsymbol{v}\subset\boldsymbol{u}} \mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right]^2\right)$$

which may be written as

$$\mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})] = \sum_{\boldsymbol{u}\subseteq\mathcal{D}} \sigma_{\boldsymbol{u}}^2 \tag{79}$$

where

$$\sigma_{\boldsymbol{u}}^2 = \mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}] - \sum_{\boldsymbol{v}\subset\boldsymbol{u}} \sigma_{\boldsymbol{v}}^2 \tag{80}$$

and

$$\mathrm{Var}[\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}] = \mathbb{E}[\mathrm{X}^2(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}] - \mathbb{E}[\mathrm{X}(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\boldsymbol{c}/\boldsymbol{\xi}_{\boldsymbol{u}}}]^2 \tag{81}$$
$$\sigma_{\boldsymbol{v}} = \mathbb{E}\left[\mathrm{x}_{\boldsymbol{v}}^2(\boldsymbol{\xi}_{\boldsymbol{v}})\right] - \mathbb{E}\left[\mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right]^2. \tag{82}$$

The notion of order when referring to the cut-HDMR expansion should not be confused with the order of a Taylor series expansion. It has been shown [18], however, that a first order cut-HDMR component function of a single input parameter is equal to the sum of all Taylor series terms involving that parameter only. Likewise, in the general case of M input parameters, the cut-HDMR component function is equal to the sum of all Taylor series terms involving all M parameters [19]. Unlike a Taylor expansion of a given order, the cut-HDMR expansion does not limit the non-linearity of the response function. Thus, any truncated cut-HDMR expansion provides a better representation of the response than any truncated Taylor expansion.

4.1. Adaptive HDMR

The total number of terms in the full cut-HDMR expansion is 2^M. To be computationally feasible for large M, this expansion must be truncated. There are two ways to do this: limit the number of variables in each component function and limit the number of component functions for a given HDMR order. The first restriction defines the truncation dimension N_t and the second defines the superposition dimension N_{s_i} for a number of dimensions i. These numbers describe roughly the number of important dimensions and the order of important interactions respectively. The total number of terms can now be calculated by

$$N = 1 + \sum_{i=1}^{N_t} N_{s_i}$$

In practice, the truncation and superposition dimensions are not known *a priori* and in order to calculate them all 2^M terms in the full cut-HDMR expansion would be required. This would be impractical since it would require more computational effort than integrating the response function $\mathrm{X}(\boldsymbol{\xi})$ directly. In this work we would like to minimise the total number of terms in the HDMR expansion and hence minimise the total number of quadrature points needed to compute the integral in Eqns.(74) and (80). Therefore, we would like to look at adaptive strategies that automatically detect the truncation and superposition dimensions. The adapted cut-HDMR expansion is written as follows:

$$\begin{aligned}\mathrm{X}(\boldsymbol{\xi}) = \mathrm{x}_0 + \sum_{i\in\mathcal{F}_1} \mathrm{x}_i(\xi_i) + \sum_{(i_1,i_2)\in\mathcal{F}_2} \mathrm{x}_{i_1 i_2}(\xi_{i_1},\xi_{i_2}) + \sum_{(i_1,i_2,i_3)\in\mathcal{F}_3} \mathrm{x}_{i_1 i_2 i_3}(\xi_{i_1},\xi_{i_2},\xi_{i_3}) \\ + \cdots + \sum_{(i_1,\cdots,i_{N_t})\in\mathcal{F}_{N_t}} \mathrm{x}_{i_1\cdots i_3}(\xi_{i_1},\cdots,\xi_{i_{N_t}})\end{aligned} \tag{83}$$

which can be written in a more compact form

$$\mathrm{X}(\boldsymbol{\xi}) = \sum_{u\in\mathcal{F}} \mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}}) \tag{84}$$

where $\mathcal{F}$ is the union of all the sparse index sets $\mathcal{F}_i$, namely

$$\mathcal{F} = \bigcup_{i=0}^{N_t} \mathcal{F}_i$$

Each of the sparse index sets $\mathcal{F}_i$ are calculated adaptively with the cardinality of each equalling the superposition dimension for that truncation dimension, viz. $|\mathcal{F}_i| = N_{s_i}$. Every component $\mathrm{x}_{\boldsymbol{u}}$ in Eqn.(84) has an associated weight $\eta_{\boldsymbol{u}} \geq 0$ which describes the contribution of $\mathrm{x}_{\boldsymbol{u}}$ to the full expansion. The adaptive strategy consists of computing the superposition dimension N_{s_k} for each set $\mathcal{F}_k$ successively for increasing expansion order until the relative change in expansion approximation, according to a predefined measure, falls below some threshold. To this end, the zeroth and first order expansions must be constructed. This is the minimal amount of computational effort that is required to start the adaptive procedure and it scales linearly with the number of dimensions M. The weight η_i for all of the first order terms are calculated and the sparse set $\mathcal{F}_1$ is formed. There are several alternatives in the literature [24, 25] for calculating the weight η. Since we are mainly concerned with the convergence of the second order statistics, we have chosen the weight based upon the variance contribution of each first order component relative to the variance attributed by all first order components, namely

$$\eta_i = \frac{\mathrm{Var}(\mathrm{x}_i)}{\sum_{j=1}^{M} \mathrm{Var}(\mathrm{x}_j)} \tag{85}$$

Each first order function describes the sole interaction of that dimension upon the output when it is acting independently of all other dimensions. This quantity can be

thought of as a sensitivity measure for the ith dimension. The importance of any given dimension can then be defined when the weight η_i is above some threshold θ. A similar weight is defined for the higher order interactions $\eta_{\boldsymbol{u}}$ as follows:

$$\eta_{\boldsymbol{u}} = \frac{|\mathrm{Var}(\mathrm{x}_{\boldsymbol{u}})|}{\left|\sum_{\boldsymbol{v}\in\mathcal{F}_{|\boldsymbol{v}|},|\boldsymbol{v}|\leq\nu-1} \mathrm{Var}(\mathrm{x}_{\boldsymbol{v}})\right|} \tag{86}$$

The weights η can be used to determine the superposition dimensions $\{N_{s_k}; k = 1, \cdots, N_t\}$. The truncation dimension N_t is determined when $\rho_t < \theta$ where the weight ρ is determined using the following:

$$\rho_p = \frac{\left|\sum_{\nu\leq p} \mathrm{Var}(\mathrm{x}_{\boldsymbol{u}}) - \sum_{\nu\leq p-1} \mathrm{Var}(\mathrm{x}_{\boldsymbol{u}})\right|}{\left|\sum_{\nu\leq p-1} \mathrm{Var}(\mathrm{x}_{\boldsymbol{u}})\right|} \tag{87}$$

The adaptive HDMR algorithm is given in Algorithm (2). The computation proceeds as follows: To begin, the zeroth and first order terms are calculated. The first order sparse set $\mathcal{F}_1$ is a tuple of the index i and associated weight η_i. The indexes of the intermediate-second order set $\mathcal{R}$ (also a tuple) is now formed from all of the allowed index subsets of the Cartesian product $\mathcal{F}_1 \times \mathcal{F}_1$. An associated weight $\tilde{\eta}$ is calculated for each of the index subsets of $\mathcal{R}$ based upon the weights of $\mathcal{F}_1$. For example:

$$(1, 2) \rightarrow \tilde{\eta}_{12} = \eta_1\eta_2$$

The weights $\tilde{\eta}$ provide a prediction of the importance of the second order terms based upon the calculated importance of the first order terms. The set $\mathcal{R}$ is now sorted in ascending order according to the predicted importance $\tilde{\eta}$. Each member of $\mathcal{R}$ is computed until its calculated weight η falls below the threshold θ. The sparse set $\mathcal{F}_2$ is now formed from all those members of $\mathcal{R}$ whose calculated weights η are above the threshold θ. The above process is repeated for all of the higher order interactions until the weight $\rho_n < \theta$. For the case of $M = 5$, the adaptive procedure is illustrated in Fig.(5).

The proposed algorithm assumes that the actual importance η decreases monotonically with the predicted importance $\tilde{\eta}$. Therefore, the accuracy of the adaptive procedure is dependent upon the quality of the predicted importance. The accuracy is also dependent upon the rate of decay of the values of η. If the multivariate terms decay very quickly, Algorithm (2) will be very efficient. If , however, the contribution of these terms is isotropic, then the algorithm may finish before enough of the terms have been computed.

4.2. Combination with Polynomial Chaos

The computation of the expectation of the response surface using HDMR defined in Eqn.(84) consists of a series of integrations along cuts and hyperplanes in the stochastic space. This procedure can easily be extended to produce a surrogate model of the

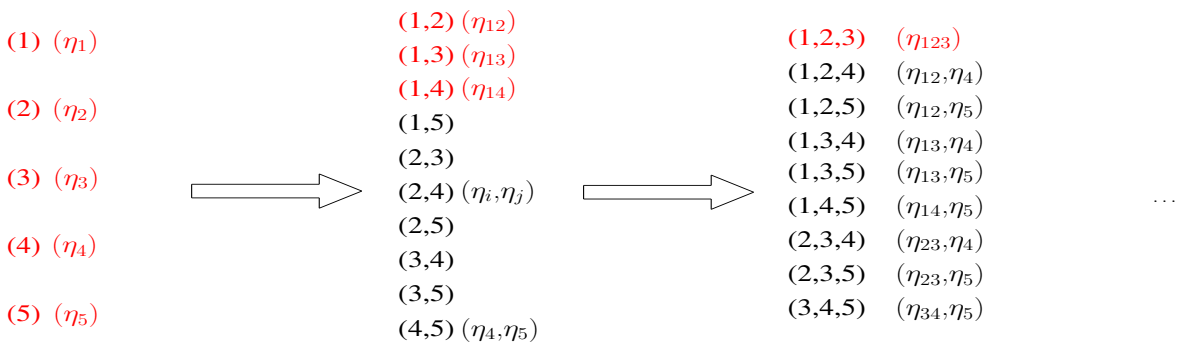


Figure 5: Graphical example of the adaptive HDMR strategy.

Algorithm 2 Modified adaptive HDMR

Calculate x_0, $\mathrm{x}_i \; \forall \, i \in \{1, \cdots, M\}$
$\mathcal{F}_1 = \{(i, \tilde{\eta}_i); i = 1, \cdots, M\}$
$i = 2$
while $\rho > \theta$ **do**
  $\mathcal{R} = \{(\boldsymbol{v}, \eta) \in \mathcal{F}_{i-1} \times \mathcal{F}_1; \boldsymbol{v} \in \mathcal{D}\}$
  Re-order the set $\mathcal{R}$ in ascending order according to $\tilde{\eta}$
  $\mathcal{F}_i = \emptyset$
  while $\boldsymbol{u} \in \mathcal{R}$ **and** $\eta_{\boldsymbol{u}} > \theta$ **do**
    Calculate $\bar{\mathrm{x}}_{\boldsymbol{u}}$, $\sigma^2_{\boldsymbol{u}}$ and weight $\eta_{\boldsymbol{u}}$ associated with set $\boldsymbol{u}$
    $\mathcal{F}_i += (\boldsymbol{u}, \eta_{\boldsymbol{u}})$
  end while
  Calculate ρ
  $i = i + 1$
end while

response function $\mathrm{X}(\boldsymbol{\xi})$ by projecting each of the HDMR component functions $\mathrm{x}_{\boldsymbol{u}}$ onto its own candidate polynomial basis. Combining the adaptive HDMR and PCE approaches in this manner allows a sparse PC representation to be built which optimally represents the response function.

Starting with the expression for the PC coefficients using the non-intrusive approach from Eqn.(40)

$$\alpha_{\boldsymbol{j}} = \frac{< \Phi_{\boldsymbol{j}}(\boldsymbol{\xi}) \mathrm{X}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) >}{< \Phi^2_{\boldsymbol{j}}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) >} = \frac{\mathbb{E}\left[\Phi_{\boldsymbol{j}}(\boldsymbol{\xi}) \mathrm{X}(\boldsymbol{\xi})\right]}{N^2_{\boldsymbol{j}}} \qquad \boldsymbol{j} \in \mathcal{A} \tag{88}$$

where $\mathcal{A}$ is the set of all polynomial chaos indexes. We may now insert the expression for the adapted HDMR expansion from Eqn.(84) into (88) to yield

$$\alpha_{\boldsymbol{j}} = \frac{\mathbb{E}\left[\Phi_{\boldsymbol{j}}(\boldsymbol{\xi}) \sum\limits_{\boldsymbol{u} \in \mathcal{F}} \mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})\right]}{N^2_{\boldsymbol{j}}} \tag{89}$$

Using the recurrence relations in (69) we may replace the integration over the component $\mathrm{x}_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}})$ with one involving the response surface $\mathrm{X}(\boldsymbol{\xi})$

$$\begin{aligned}\alpha_{\boldsymbol{j}} &= \frac{1}{N_{\boldsymbol{j}}^2}\mathbb{E}\left[\Phi_{\boldsymbol{j}}(\boldsymbol{\xi})\sum_{\boldsymbol{u}\in\mathcal{F}}\left(\mathrm{X}(\boldsymbol{\xi}_{\boldsymbol{u}}) - \sum_{\boldsymbol{v}\subset\boldsymbol{u}}\mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right)\right] \\ &= \frac{1}{N_{\boldsymbol{j}}^2}\sum_{\boldsymbol{u}\in\mathcal{F}}\left(\mathbb{E}\left[\Phi_{\boldsymbol{j}}(\boldsymbol{\xi}_{\boldsymbol{u}})\mathrm{X}(\boldsymbol{\xi}_{\boldsymbol{u}})\right] - \sum_{\boldsymbol{v}\subset\boldsymbol{u}}\mathbb{E}\left[\Phi_{\boldsymbol{j}}(\boldsymbol{\xi}_{\boldsymbol{u}})\mathrm{x}_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}})\right]\right) \end{aligned} \tag{90}$$

From Eqn.(90) we can see that each ν dimensional quadrature over the response function requires the subtraction of all ν dimensional projections onto all subsets of $\boldsymbol{u}$. For example, consider the bi-variate polynomial basis spanning the two dimensional surface described by the variables ξ_1 and ξ_2. The surrogate model for this surface is then defined by the coefficients $\alpha_{\boldsymbol{j}}$ where $\boldsymbol{j} = (1,0),(2,0),(1,1),(0,1),(0,2)$. The first coefficient $\alpha_{(1,0)}$ can be calculated as follows:

$$\begin{aligned} N_1^2 N_0^2 \alpha_{(1,0)} = &\int \Phi_1(\xi_1)\mathrm{X}(\xi_1)p(\xi_1)d\xi_1 \\ &+ \left(\int\int \Phi_1(\xi_1)\mathrm{X}(\xi_1,\xi_2)p(\xi_1,\xi_2)d\xi_1 d\xi_2 - \int \Phi_j(\xi_1)\mathrm{X}(\xi_1)p(\xi_1)d\xi_1\right) \\ &+ \text{ higher order projections} \end{aligned}$$

I.e. the coefficient for the one dimensional polynomial spanning dimension 1 has contributions from the univariate term in the HDMR expansion, $\mathrm{x}_1(\xi_1)$, a contribution from the two dimensional term $\mathrm{x}_{12}(\xi_1,\xi_2)$ and all other higher dimensional terms involving dimension 1.

5. Results

In this section we present the results of the application of the above theory to two test cases and a set of criticality problems. The two test cases are performed in order to verify the implementation of the numerical methods and to also show their behaviour in an idealised situation. The criticality calculations are performed on an infinite array of Uranium-Oxide (UOX) and mixed-Oxide (MOX) pin cells. The nuclear data uncertainties used in these calculations are taken from evaluated libraries.

5.1. Sparse polynomial chaos expansion from adaptive quadrature

In this first test, the adaptive sparse grid quadrature scheme is used to determine the polynomials in the PCE of the response. In particular, we have calculated the variance of the response using the PCE and compared it to exact results. The variance of a function with respect to the probability measure $p(\xi)$ is calculated using

$$\mathrm{Var}[f] = \int f(\boldsymbol{\xi})^2 p(\boldsymbol{\xi})d\boldsymbol{\xi} - \left[\int f(\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi}\right]^2 \tag{91}$$

The function that we have chosen is the following:

$$f(\boldsymbol{\xi}) = \prod_{i=1}^{M}(1+\xi_i) \tag{92}$$

which is a tensor product of linear polynomials in each dimension. Analytic expressions for the variance of the function in Eqn.(92) with respect to the Gaussian probability weight, the uniform probability weight and a mixture of the two are given as follows:

$$\begin{aligned}
\mathrm{Var}[f_{\mathrm{uni}}] &= \int \prod_{i=1}^{M}(1+\xi_i)^2 p(\xi_i)d\xi_i - \left[\int \prod_{i=1}^{M}(1+\xi_i)p(\xi_i)d\xi_i\right]^2 \\
&= \left(\frac{4}{3}\right)^M - 1 \qquad (93) \\
\mathrm{Var}[f_{\mathrm{norm}}] &= \int \prod_{i=1}^{M}(1+\eta_i)^2 p(\eta_i)d\eta_i - \left[\int \prod_{i=1}^{M}(1+\eta_i)p(\eta_i)d\eta_i\right]^2 \\
&= 2^M - 1 \qquad (94) \\
\mathrm{Var}[f_{\mathrm{mix}}] &= \int \left(\prod_{i=1}^{m_1}(1+\eta_i)^2 p(\eta_i)\right)\left(\prod_{i=1}^{m_2}(1+\xi_i)^2 p(\xi_i)\right) d\xi_i d\eta_i \\
&\quad - \left[\int \left(\prod_{i=1}^{m_1}(1+\eta_i)p(\eta_i)\right)\left(\prod_{i=1}^{m_2}(1+\xi_i)p(\xi_i)\right) d\xi_i d\eta_i\right]^2 \\
&= 2^{m_1}\left(\frac{4}{3}\right)^{m_2} - 1 \qquad (95)
\end{aligned}$$

where the probability density functions $p(\xi_i)$ and $p(\eta_i)$ are given by Eqns.(35) and (36) respectively.

We now calculate a PCE for Eqn.(92) using the three different probability weights. For the numerical calculations we choose $M = 4$ and $m_1 = m_2 = 2$. The coefficients of the PCE are calculated using the non-intrusive approach defined in Eqn.(40). The orders of the polynomials in the expansion, which make up the set $\mathcal{A}$ in Eqn.(40), are calculated using Eqn.(64) for both the generalised and conventional sparse grid integration schemes. The results of these simulations are presented in Figures (6) and (7); in Figure (6) the relative error is plotted versus the number of integration points and in Figure (7) the relative error is plotted versus the number of polynomials in the PCE. In both Figures (6) and (7), the quadrature level used in the conventional sparse grid, labelled "SG", has been printed near to the corresponding data points. The results shows that the adaptive construction significantly out-performs the conventional approach for this particular function. The reason for this is that the adaptive procedure allows all indexes in the tensor product of one dimensional indexes and quickly captures the high monomial order cross terms in Eqn.(91) viz. $\prod_{i=1}^{M}\xi_i$.

In Figures (6) and (7) the results for uniformly distributed uncertainties are more accurate than the results for Gaussian uncertainties. This is due to the fact that the

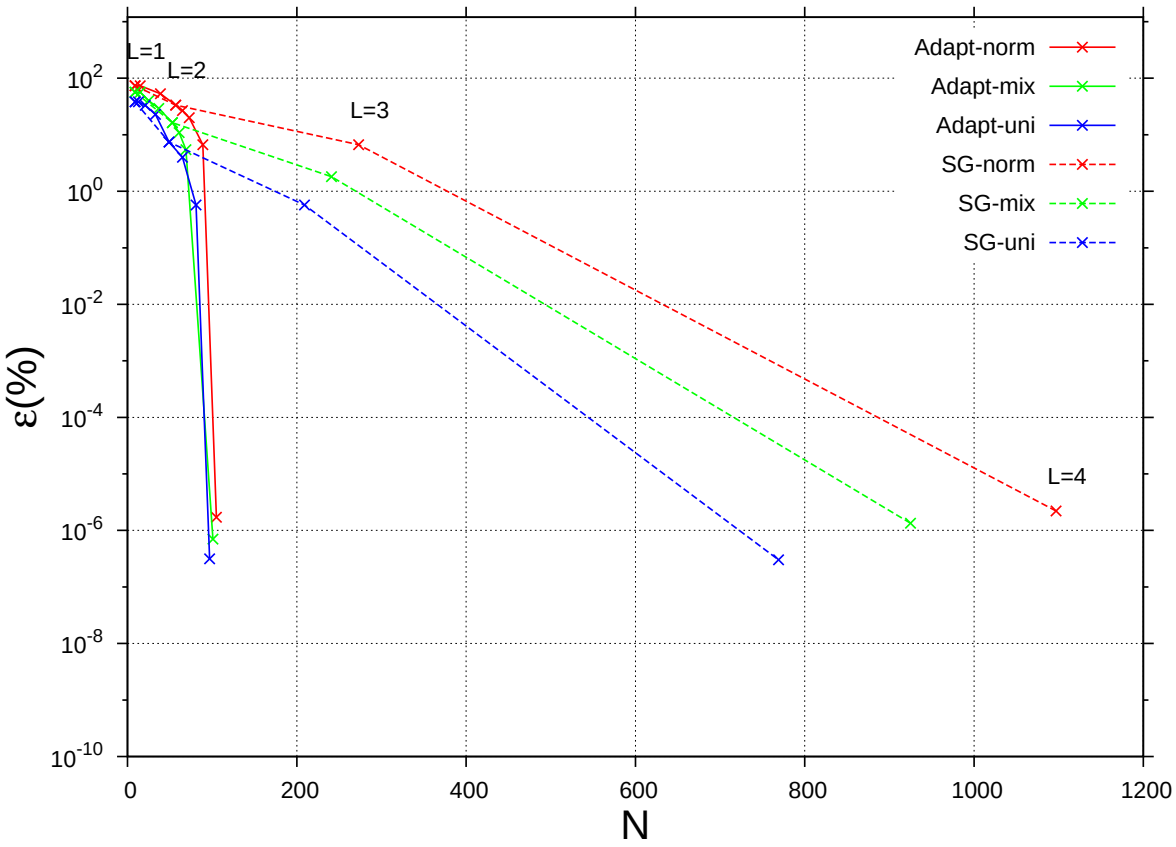


Figure 6: Convergence of the variance calculated using a polynomial chaos expansion built with conventional sparse grid and adaptive sparse grid indexes versus the number of quadrature points N. The value of L corresponds to the conventional sparse grid level.

Gauss-Patterson quadrature rules have a greater accuracy per node than the Genz-Keister rules. The mixture of both lies somewhere in between the two.

5.2. HDMR with Adaptive Quadrature

In this example, the adaptive quadrature method is combined with the adaptive HDMR technique and applied to a simple test function. The following test function was chosen

$$\mathrm{X}(\boldsymbol{\xi}) = \frac{1}{1 + \sigma\sqrt{3}\sum_{k=1}^{M} \alpha_k \xi_k} \quad \xi_k \in [-1, 1] \tag{96}$$

with

$$\alpha_k = \exp\{-|k|/l\} \tag{97}$$

The terms α_k control the contribution of each random dimension through the parameter l. The parameter l is referred to as the structure parameter for the remainder of this example. The larger the value of l the slower the set $\{\alpha\}$ decays and the more dimensions contribute to the non-linear function $\mathrm{X}(\boldsymbol{\xi})$. The parameter σ is the standard

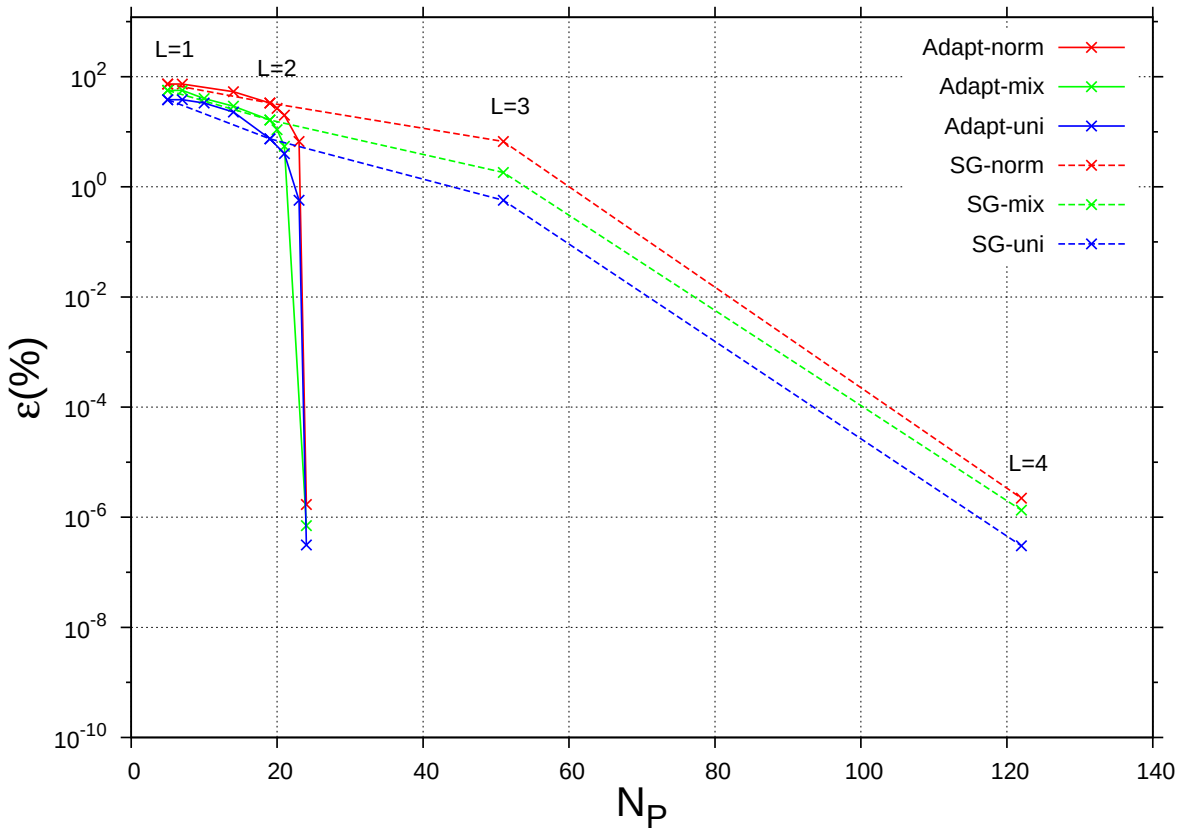


Figure 7: Convergence of the variance calculated using a polynomial chaos expansion built with conventional sparse grid and adaptive sparse grid indexes versus number of polynomials N_P. The value of L corresponds to the conventional sparse grid level.

deviation which controls the magnitude of the variation in the random input. The random variables ξ_k vary uniformly between $[-1, 1]$. The number of dimensions $M = 10$, the standard deviation is $\sigma = 0.1$ and the structure parameter $l = 2.0$. For the following example we have used the relative error which is defined as

$$\epsilon = \frac{||\mathrm{X}(\boldsymbol{\xi}) - \tilde{\mathrm{X}}(\boldsymbol{\xi})||_2}{||\mathrm{X}(\boldsymbol{\xi})||_2}$$

where $\tilde{\mathrm{X}}$ is the approximate solution calculated using the HDMR expansion. The reference solution is calculated using the LHS method with 10^8 samples.

In this investigation, the variance of Eqn.(96) was calculated using Eqn.(79) where the integrals of the components in Eqn.(79) were calculated using adaptive quadrature. The nested Gauss-Patterson rules were used for all of the integrations. In Fig.(8) the relative error in the variance is plotted versus the number of realisations N. In Fig.(9) the relative error in the variance is plotted versus the adaptive HDMR tolerance θ. In both figures the integrals of the HDMR components were performed using a range of adaptive integration tolerances from $\delta = 10^{-3} \rightarrow 10^{-11}$ which correspond to the dashed lines and for non-adaptive quadrature using levels $L = 2 \rightarrow 6$ which correspond to the solid lines. The adaptive HDMR tolerance was decreased from $\theta = 0.5 \rightarrow 10^{-11}$.

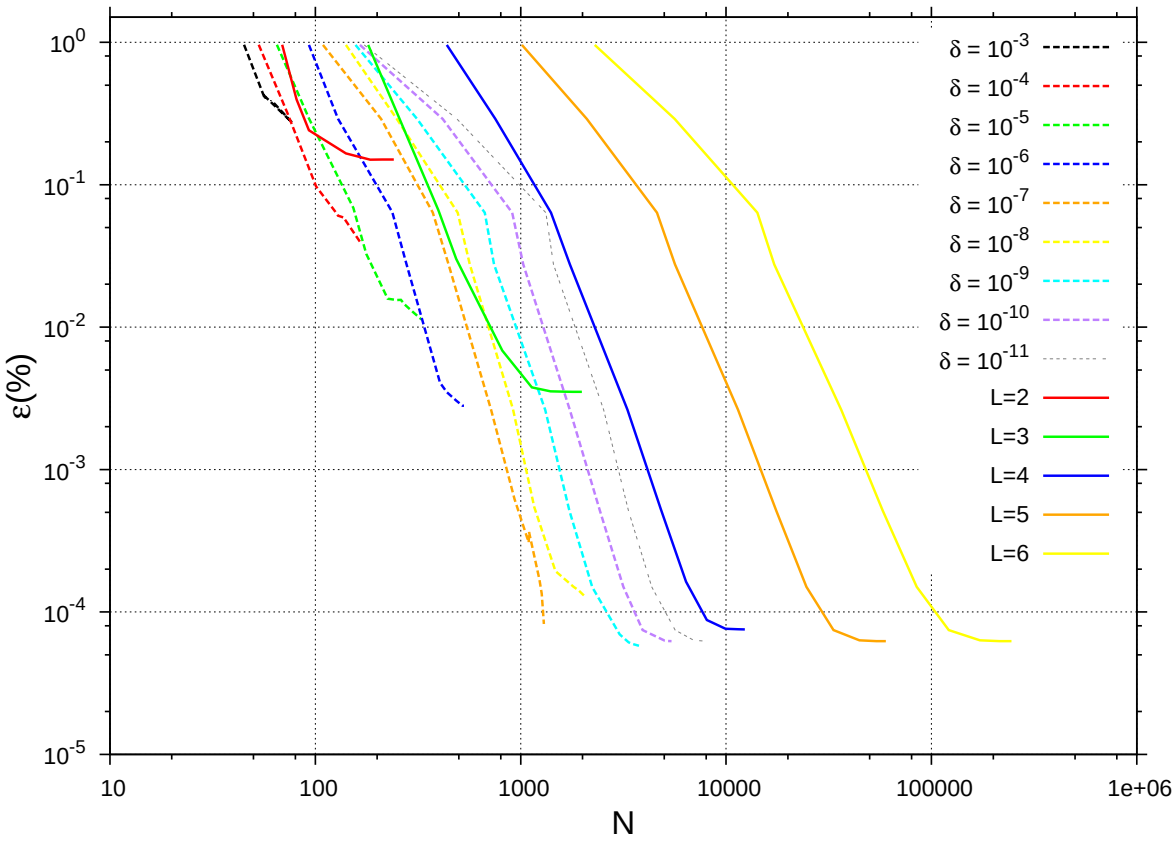


Figure 8: Error versus the number of realisations (N). Values of $\sigma = 0.1$ and $l = 2.0$ have been used whilst integration has been performed with Gauss-Patterson rules. Values of L are used to control the accuracy of the conventional sparse grid and δ controls the accuracy of the adaptive sparse grid.

In Fig.(8) we can clearly see that the use of the level L=5 and L=6 non-adaptive sparse grid quadrature rules are far less efficient than any of the adaptive procedures. From Fig.(9) we can see that the reduction in the relative error versus adaptive HDMR tolerance θ is very similar for the adaptive integration tolerances $\delta = 10^{-7}$ and 10^{-8} and the Level L=4 regular sparse grid. However from Fig.(8), the adaptive quadratures are more efficient than the non-adaptive ones; in fact, the $\delta = 10^{-7}$ results are about an order of magnitude more efficient than the L=4 quadratures. The results in both Figures (8) and (9) show that in order to improve the accuracy of the HDMR solution both the adaptive HDMR tolerance (θ) and the adaptive quadrature tolerance (δ) must be decreased simultaneously. If the correct combination of θ and δ are chosen, HDMR with adaptive quadrature is far more efficient than its non-adaptive counterpart.

5.3. Adaptive HDMR versus Adaptive Quadrature

In this section we compare the use of the combined adaptive HDMR - adaptive quadrature method to the purely adaptive quadrature method or stochastic collocation method (SCM). Both approaches are used to build a sparse polynomial chaos expansion (PCE) of the response function. The test function used for this investigation is defined in Eqn.(96) with the number of dimensions $M = 10$, the standard deviation is $\sigma = 0.1$ and the structure parameter $l = 2.0$. The number of coefficients

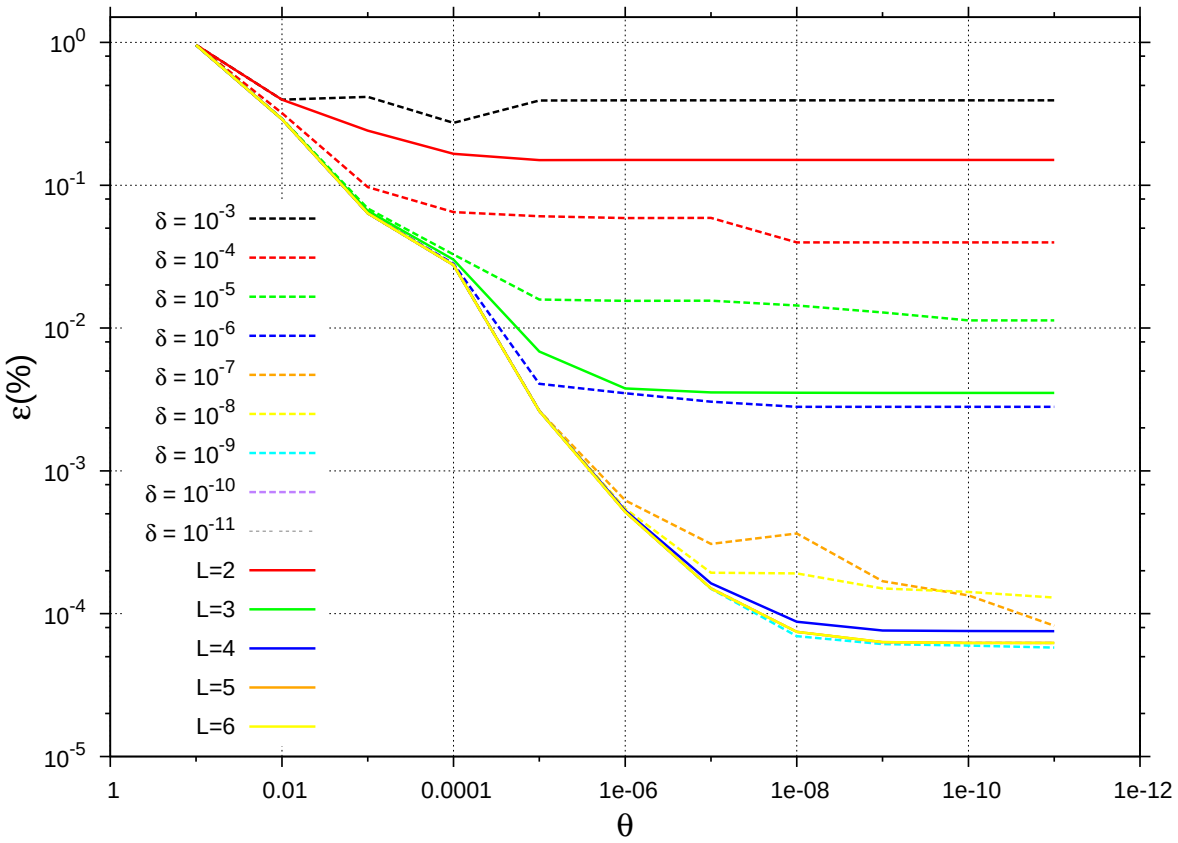


Figure 9: Error versus adaptive HDMR tolerance (θ). Values of $\sigma = 0.1$ and $l = 2.0$ have been used whilst integration has been performed with Gauss-Patterson rules. Values of L are used to control the accuracy of the conventional sparse grid and δ controls the accuracy of the adaptive sparse grid.

in the PCE is increased by decreasing either the adaptive HDMR tolerance or the adaptive quadrature tolerance. The variance in the PCE of the response parameter is calculated using Eqn.(38). The relative error in the variance is then plotted against the number of quadrature points, N, in Figure (10) and against the number of polynomials in the PCE, N_P, in Figure (11); these results are for HDMR with adaptive and non-adaptive quadrature and also for the stochastic collocation method (SCM) with adaptive and non-adaptive quadrature.

The results in both Figure (10) and (11) give an indication of the relative efficiency of each approach. The results show that the SCM with adaptive quadrature is consistently the most efficient although, with the appropriate choice of θ and δ, the HDMR approach with adaptive quadrature is still competitive in terms of accuracy per realisation/polynomial. We also note that HDMR using the conventional sparse grid of level L converges to the non-adaptive SCM with the same level as the adaptive HDMR tolerance $\theta \to 0$. A more decisive factor in the choice of HDMR with adaptive quadrature over the use of the SCM with adaptive quadrature, is the execution time of the algorithms. The relative error in the calculated variance is plotted versus execution time for both HDMR and SCM with adaptive quadrature for 10 and 20 stochastic di-

mensions, M, in Figure (12). Here we can see that increasing the number of dimensions from 10 to 20 results in greater than an order of magnitude increase in the execution time of the adaptive SCM, whereas the adaptive HDMR is comparatively insensitive to this change. However, the implementation of the adaptive quadrature scheme may not be the most efficient due to the high level programming methods that were used. Efficient data structures for the adaptive quadrature algorithm have been proposed in [16] and may go some way in reducing the execution time.

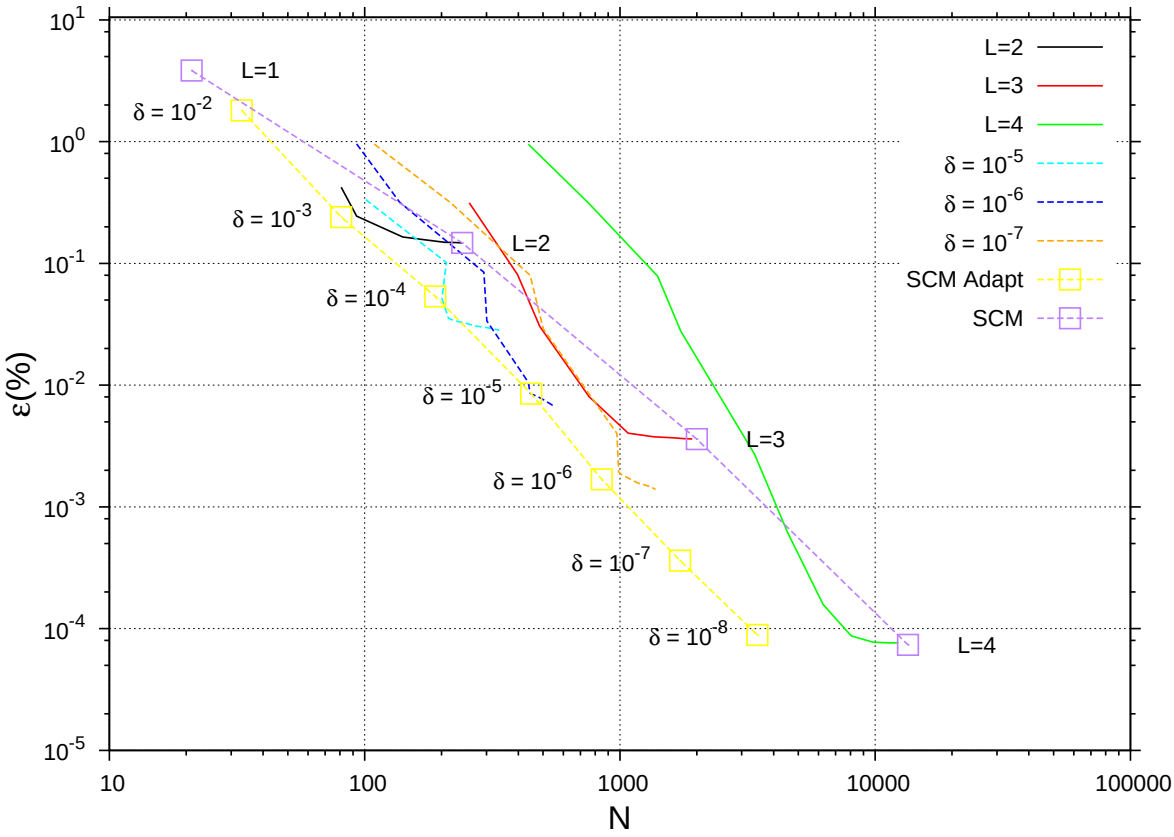


Figure 10: Comparison of conventional and adaptive sparse grids with the HDMR method using conventional and adaptive sparse grids. The relative error in the calculated variance is plotted versus the number of realisations (N).

5.4. Rowlands' Pin Cell Problems

The Light Water Reactor (LWR) Rowlands' pincell benchmarks are a set of pincell problems described in an OECD report by J.L. Rowlands [52] for benchmarking LWR lattice physics codes. There are two pincells described in the report, the first is a simple PWR square Uranium Oxide (UO_2) pincell problem without leakage and the second is a geometrically similar Mixed Oxide (MOX) pincell. They are both three region cells consisting of a central fuel region, a zirconium clad region and water coolant region. The outer radii of the fuel and clad regions are denoted r_1 and r_2 respectively and the size of the square coolant region denoted by (x_d, y_d) which are widths of the x and y dimensions respectively. The geometries of the problems are specified by the following

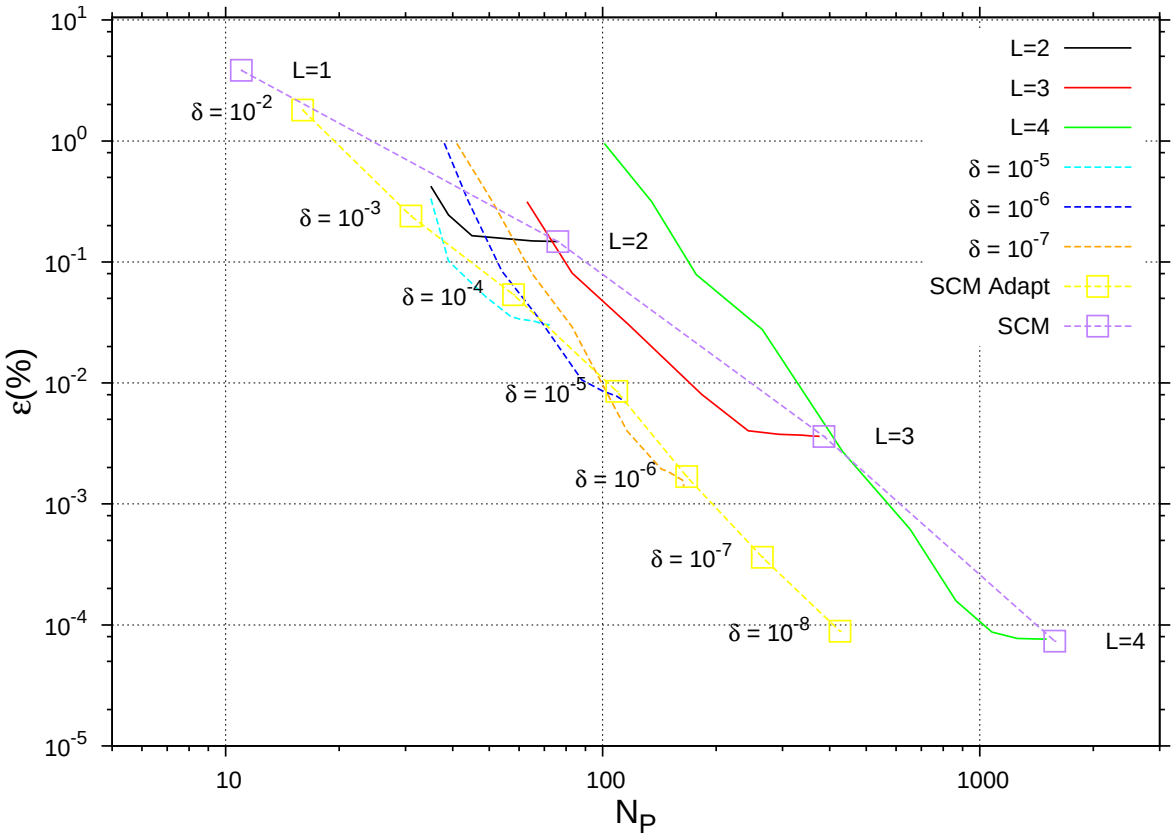


Figure 11: Comparison of conventional and adaptive sparse grids with the HDMR method using conventional and adaptive sparse grids. The relative error in the calculated variance is plotted versus the number of polynomials in the PCE (N_P).

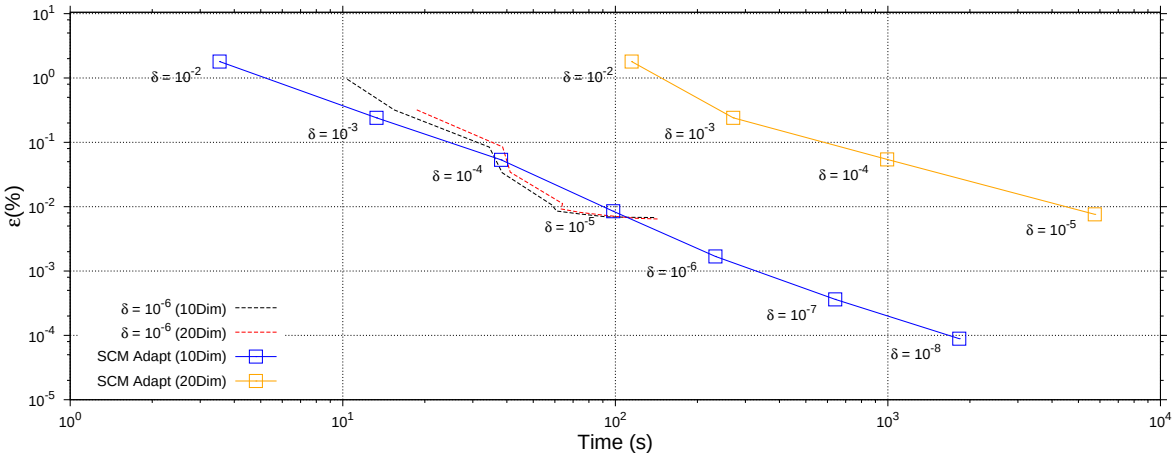


Figure 12: Relative error in the calculated variance versus execution time for HDMR and the SCM using adaptive quadrature.

values:

$$\begin{array}{lllll} \text{UOX:} & r_1 = 0.40\text{cm} & r_2 = 0.450\text{cm} & x_d = y_d = 1.20\text{cm} \\ \text{MOX:} & r_1 = 0.41\text{cm} & r_2 = 0.475\text{cm} & x_d = y_d = 1.26\text{cm} \end{array}$$

The temperature of the each pincell is assumed to be a homogeneous value of 293.16K. A schematic of the problems is shown in Figure (13) and the material compositions are given in Table (1).

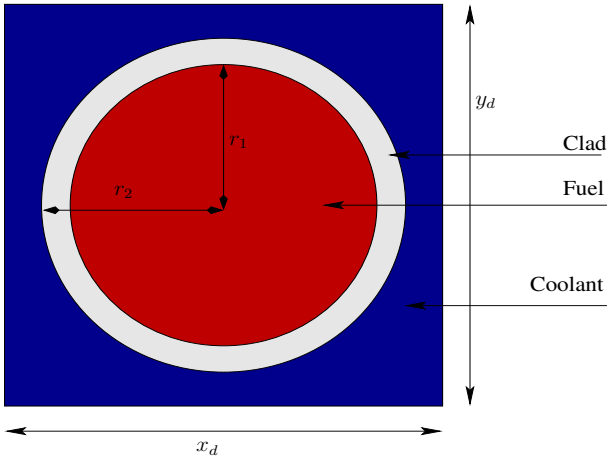


Figure 13: Two-dimensional xy geometry of the LWR Rowlands Pincell problem

Material	Composition		Number density (atoms.barn^{-1}.cm^{-1})	
	UOX	MOX	UOX	MOX
Fuel	U_{235}	U_{235}	7.0803×10^{-4}	5.1050×10^{-5}
	U_{238}	U_{238}	2.2604×10^{-2}	2.0370×10^{-2}
	O_{16}	O_{16}	4.6624×10^{-2}	4.5880×10^{-2}
		Pu_{238}		4.6690×10^{-5}
		Pu_{239}		1.4650×10^{-3}
		Pu_{240}		5.6910×10^{-4}
		Pu_{241}		2.7130×10^{-4}
		Pu_{242}		1.4130×10^{-4}
		Am_{241}		3.0280×10^{-5}
Clad	Zr	Zr	4.3241×10^{-2}	3.8800×10^{-2}
Coolant	H	H	6.6988×10^{-2}	4.7440×10^{-2}
	O_{16}	O_{16}	3.3414×10^{-2}	2.3720×10^{-2}

Table 1: Material compositions for the fuel, clad and coolant regions of the Rowlands' PWR square pincell problem

The clad region of the Rowlands' pin cell problem is composed of zirconium which is formed from 5 naturally occurring isotopes according to Table (2).

As was mentioned in Section (2), all k_{eff} calculations for the uncertainty analysis have been performed using the even parity transport code EVENT [28]. The energy discretisation used 3, 9 and 26 energy groups. The angular variation in the neutron flux was modelled using a P_3 spherical harmonic expansion and the scattering was

Isotope	Abundance (%)
Zr^{90}	51.45
Zr^{91}	11.22
Zr^{92}	17.15
Zr^{94}	17.38
Zr^{96}	2.80

Table 2: Isotopic abundance of Zirconium

assumed to be isotropic. The low angular discretisation and the assumption of isotropic scattering were prescribed in order to minimise the execution time of the simulations. The pin cell is modelled in xy geometry using one eighth symmetry to reduce the

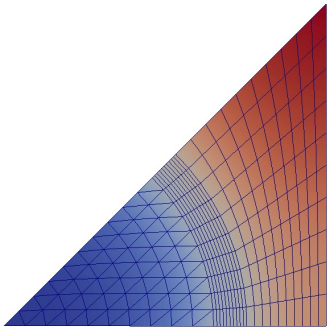


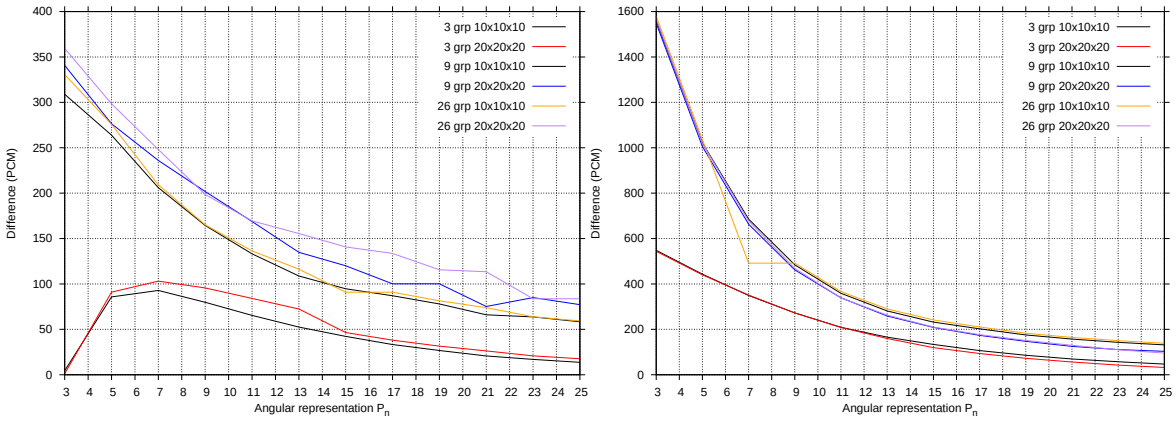
Figure 14: Discretised geometry of the Rowlands' pin cell problems for EVENT calculations.

number of unknowns in the problem. The geometry of the pin cells was discretised using the finite element method using 100 linear triangles in the fuel region, 100 bilinear quadrilaterals in the clad region and 100 bi-linear quadrilaterals in the coolant region as shown in Fig.(14). As a consequence of using linear triangles and quadrilaterals, the mesh is a polygonal approximation to the exact circular geometry of the fuel and clad regions in the pin cell. Therefore, in order to preserve the fissile, clad and coolant areas, the radii need to be modified. The relationship used to modify the radii of the two-dimensional finite element approximation of the sector of the circle is given by:

$$R' = \left(\sqrt{\frac{\theta}{N_\Delta \sin(\frac{\theta}{N_\Delta})}} \right) R \tag{98}$$

The symbol R' is the new radius, R is the old radius, N_Δ is the number of divisions on the arc and θ is the angle of the arc of the circle being meshed in radians.

The convergence of EVENT k_{eff} calculations with respect to the reference CACTUS solution are given in Fig.(15). The CACTUS reference solution used 11 azimuthal angles and 7 polar angles with a track spacing of 0.005. The spatial domain was



(a) UOX Pin Cell (b) MOX Pin Cell

Figure 15: Convergence of EVENT calculations for the UOX and MOX Rowlands' pin cell problems with space and angle.

split 30x30 in the x-y directions. The problem has also been modelled using diagonal symmetry.

From Fig.(15) we can see that both UOX and MOX problems converge to the CACTUS solution as the angular representation of the flux is increased. The justification for using the lowest, P_3 representation was to minimise the execution time making many hundreds of thousands of realisations for Monte Carlo sampling feasible. The P_3 solution with a 10x10x10 mesh, a splitting of 10 along each line of each region in Fig.(14), takes fractions of a second to compute whereas the higher spatial and angular representations take in the order of 5 minutes. As a result of the low fidelity representation a systematic error will be present in each EVENT calculation. We do not expect this systematic error to have any significance on the quantification of the uncertainty on the nuclear data when comparisons are made between Monte Carlo sampling and the polynomial chaos methods.

For all of the following problems, the input random processes (the correlated uncertain nuclear data) are represented using a Karhunen Loéve expansion. The number of terms in the Karhunen Loéve expansion is truncated when the cumulative sum of the eigenvalues in the decomposition = 1. Truncating the expansion in this way represents all of the random information contained in the random process whilst removing any degeneracy.

5.4.1. HDMR versus Latin Hypercube Sampling (LHS)

The purpose of the following numerical experiments was to look at the efficiency of the HDMR method compared to LHS. To this end, a set of test cases was devised using the Rowlands' pin cell problems which had an increasing number of uncertain input parameters. This hierarchy of test cases is detailed in Table(3).

The number of random dimensions in each problem is varied by assuming different nuclides are uncertain. For example; the UOX_{12} problem in Table(3) only considers

Groups	UOX			MOX		
	Name	Covariance Data	M	Name	Covariance Data	M
3	UOX_{12}	U^{235}	12			
	UOX_{42}	U + H + O + Zr	42	MOX_{114}	U + Pu + Am^{241} + H + O + Zr	114
9	UOX_{36}	U^{235}	36			
				MOX_{288}	U + Pu + Am^{241}	288
	UOX_{136}	U + H + O + Zr	136	MOX_{342}	U + Pu + Am^{241} + H + O + Zr	342
26	UOX_{104}	U^{235}	104	MOX_{416}	U^{235} + U^{238} + Pu^{239} + Pu^{240}	416
				MOX_{832}	U + Pu + Am^{241}	832
	UOX_{364}	U + H + O + Zr	364	MOX_{988}	U + Pu + Am^{241} + H + O + Zr	988

Table 3: Hierarchy of Rowlands' pincell problems with differing numbers of uncertain input parameters. U=U^{235} + U^{238}, Pu=Pu^{238} + Pu^{239} + Pu^{240} + Pu^{241} + Pu^{242}

uncertainties on U^{235} and all other nuclides are deterministic. This set of test problems provides a spread in the number of dimensions from 12 - 988. The number of uncertain parameters is given as a subscript in the problem name. All of the input uncertainties are assumed to have a Gaussian distribution which is a standard assumption.

To investigate the efficiency of both the LHS and HDMR approaches, the relative error is plotted against the number of realisations. The number of realisations was determined during run time of the HDMR method; once found, this number of Latin Hypercube samples was performed for three different random number seeds. An error was computed for each of the seeded LHS computations. The average error for each of these calculations was also computed. The relative error requires a reference solution, this was computed using 10^6 Latin Hypercube samples. The results of the reference solutions are given in Table (4) below:

Groups	UOX				MOX			
	Name	Mean	Variance	R(%)	Name	Mean	Variance	R(%)
3	UOX_{12}	1.3876	2.2277E-4	1.0757				
	UOX_{42}	1.3876	2.7559E-4	1.1964	MOX_{114}	1.2157	1.5015E-4	1.0079
9	UOX_{36}	1.3831	8.8523E-5	0.6802				
					MOX_{288}	1.2003	3.6955E-5	0.5064
	UOX_{136}	1.3831	1.1482E-4	0.7747	MOX_{342}	1.2003	3.8520E-5	0.5170
26	UOX_{104}	1.3829	8.7052E-5	0.6746	MOX_{416}	1.2001	3.2026E-5	0.4715
					MOX_{832}	1.2001	3.5750E-5	0.4982
	UOX_{364}	1.3829	1.1300E-4	0.7686	MOX_{988}	1.2001	3.7203E-5	0.5082

Table 4: Mean, variance and relative standard deviation R of k_{eff} for the problems defined in Table (3). These are the reference values calculated using 10^6 Latin hypercube samples.

The HDMR results were computed using an adaptive HDMR tolerance of $\theta = 10^{-5}$ and an adaptive quadrature tolerance of $\delta = 10^{-5}$. The relative error has been computed for the mean and the variance of k_{eff} for each problem and the results are plotted in Figures (16) and (17) respectively. The number of realisations used in each

of the HDMR and LHS results are given in Table (5). The results for the mean show that the HDMR error remains largely constant whilst the LHS error steadily decreases. The error in the HDMR calculation of the mean is consistently lower than the average LHS error for all values of M. The HDMR calculation of the variance is about an order of magnitude more accurate then the average LHS calculation for all of the problems up to 988 dimensions.

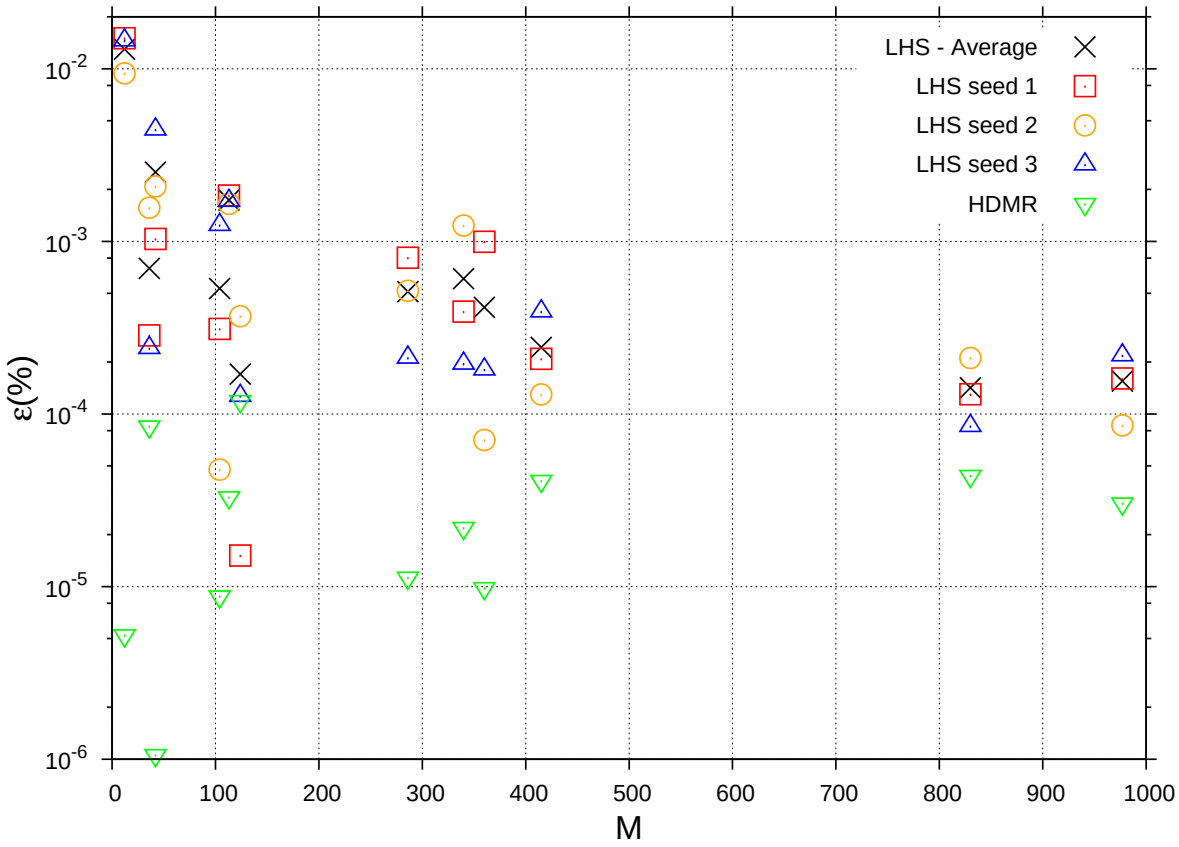


Figure 16: Relative error in the mean calculated using HDMR and LHS versus number of stochastic dimensions M for the problems in Table(3).

Groups	UOX			MOX		
	Name	M	N	Name	M	N
3	UOX_{12}	12	55	MOX_{114}	114	279
	UOX_{42}	42	149			
9	UOX_{36}	36	89	MOX_{288}	288	603
	UOX_{136}	136	297	MOX_{342}	342	711
26	UOX_{104}	104	215	MOX_{416}	416	855
				MOX_{832}	832	1691
	UOX_{364}	364	739	MOX_{988}	988	1985

Table 5: The number of realisations used in the HDMR and LHS calculations presented in Figs. (16) and (17) for the Rowland's pin cell problems.

To assess the effectiveness of the adaptive HDMR procedure, the relative contri-

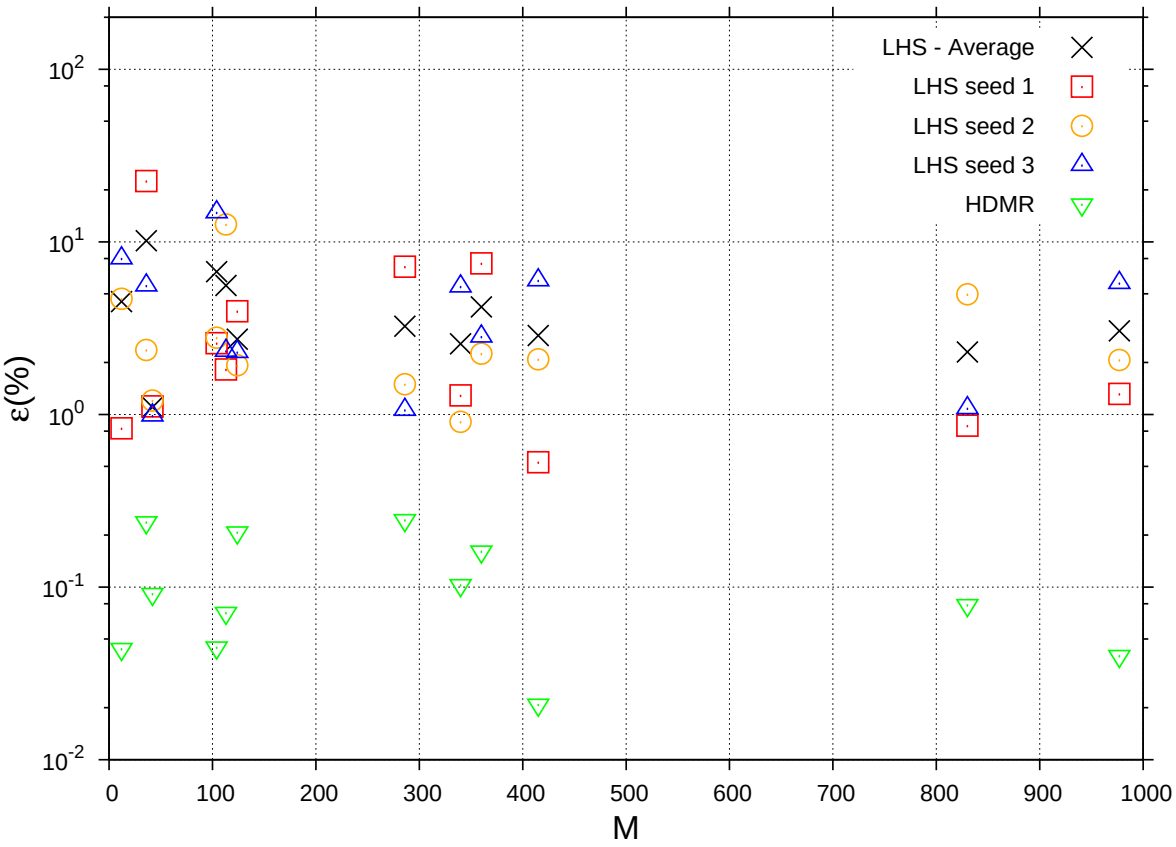


Figure 17: Relative error in the variance calculated using HDMR and LHS versus number of stochastic dimensions M for the problems in Table(3).

bution of the variance from each HDMR order, ν, (not to be confused with nu-bar, the average number of neutrons produced per fission) has also been calculated. This relative variance is defined as follows:

$$\mathrm{Var}_\nu = \frac{\sum_{|\boldsymbol{u}|=\nu} \sigma_{\boldsymbol{u}}}{\mathrm{Var}_T} \tag{99}$$

where Var_T is the total variance represented by the HDMR expansion and $\sigma_{\boldsymbol{u}}$ is the variance of the HDMR component for dimensions $\boldsymbol{u}$. The variance ratio defined by $\mathrm{Var}_{\boldsymbol{\nu}}$ in Eqn.(99) gives the variance attributed by all HDMR components with a number of dimensions equal to ν. The variance ratio for all of the problems in Table (3) are plotted in Fig.(18) along with the relative number of HDMR components, which is denoted $n(\nu = 1, 2, \mathrm{etc})$, and the relative number of realisations which is denoted $N(\nu = 1, 2, \mathrm{etc})$. In other words $n(\nu = 1)$ is the ratio of one dimensional HDMR components to the total number in the expansion, and $N(\nu = 1)$ is the ratio of the number of samples required to calculate the one dimensional terms to the total number of samples. We have also plotted the relative contributions to the polynomial chaos variance for each polynomial order p which we denote Var_p. The set of polynomial chaos indexes $\mathcal{A}$ was computed iteratively using the adaptive HDMR and quadrature

methods. The variance ratio for polynomial order p is defined as

$$\mathrm{Var}_p = \frac{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A};|\boldsymbol{\alpha}|=p} N_{\boldsymbol{\alpha}}^2 \mathrm{x}_{\boldsymbol{\alpha}}^2}{\sum\limits_{\boldsymbol{\alpha}\in\mathcal{A}} N_{\boldsymbol{\alpha}}^2 \mathrm{x}_{\boldsymbol{\alpha}}^2} \tag{100}$$

The relative number of polynomial coefficients which is denoted $N_P(p = 1, 2, \text{etc})$ has also been calculated where $N_p(p = 1)$ is the ratio of linear polynomials to the total number in the polynomial expansion and $N_P(p = 2)$ is the ratio of quadratic polynomials etc. In Fig.(18a) results of variance ratio and number ratio for $\nu = 1$ and $p = 1$ are plotted. Here we can see that the first order ($\nu = 1$) HDMR components contribute at least 60% of the total components and contribute greater than 99.97% of the calculated, total variance. The same is also true for the PCE variance ratio. In Fig.(18b) results are plotted for $\nu = 2$ and $p = 2$. Here we can see that the relative number of components and realisations are in the range of $0.1\% \rightarrow 10\%$ for all problems but the variance ratio of the HDMR components is of the order of $10^{-2}\% \rightarrow 10^{-3}\%$. This means that, in the best case scenario, 0.1% of the work results in a $10^{-2}\%$ contribution to the variance and in the worst case, 10% of the work results in a $10^{-3}\%$ contribution. Even in the best case, the utilisation of the computational effort results in a small contribution to the final result. A similar trend is observed in the results for the PCE variance ratio. In Fig.(18c) results are plotted for $\nu = 3$ and $p = 3$. Again, as with the results for $\nu = 2$, the relative number of components and realisations are in the range of $0.1\% \rightarrow 10\%$ for all problems but in this instance the variance ratio of the HDMR components is as low as $10^{-9}\%$. We note that only four problems computed the third order HDMR components resulting in the discontinuities in the HDMR variance ratio. For al other problems, the algorithm decided three variable terms unimportant. Values of the PCE variance ratio for $p = 3$ do exist for all problems since it is the adaptive quadrature algorithm that determines these values and sufficient non-linear behaviour may exist in any of the HDMR terms. Although no restriction was placed on the maximum order ν, no fourth order ($\nu = 4$) components were calculated by the algorithm for any of the problems.

For all of the problems in Table (3), each component in the HDMR expansion of the k_{eff} response surface was projected onto its own polynomial chaos basis using Eqn.(90). The resulting sparse PCE's were then used to calculate a PDF; one PDF was calculated for each polynomial order p. These PDF's are plotted in Figure (19) alongside the PDF calculated using the reference Latin Hypercube solution. Here we can see that there is no visible difference between PDFs for different values of p. Since the PCE variance ratios for $p = 2$ and $p = 3$ in Fig.(18) are insignificant, very little difference between the PDFs was expected. We also note that the $p = 1$ PDF, i.e. linear polynomials only, provides a sufficient representation of the reference solution.

5.4.2. Log-Normal versus Gaussian Distributed Uncertainties

For all of the pin cell problems studied in section (5.4.1), the uncertainties in the nuclear data were modelled using a Gaussian distribution. This distribution supports



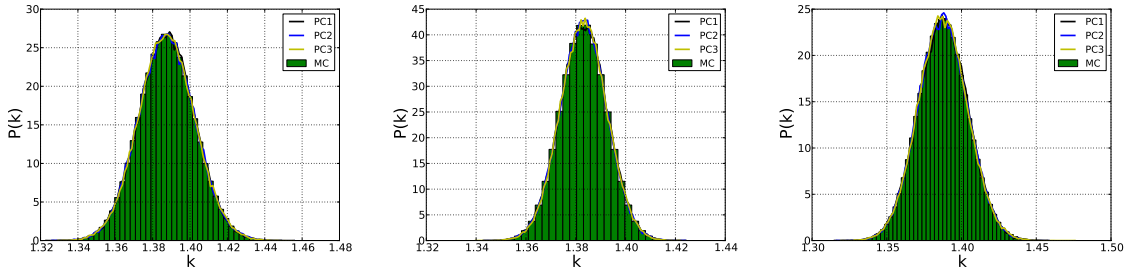
(a) $\nu = 1$ / p=1

(b) $\nu = 2$ / p=2

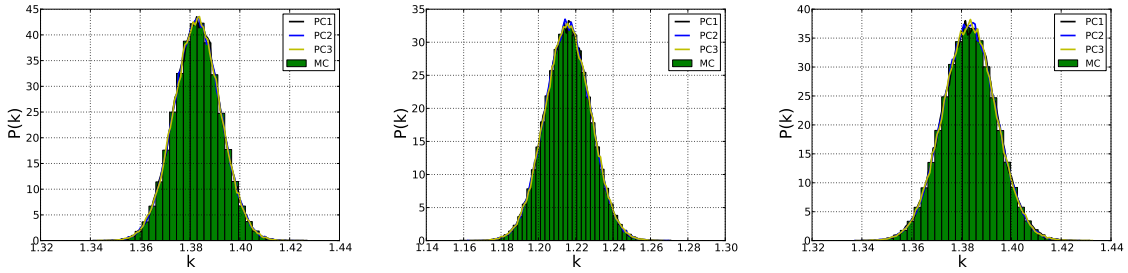
(c) $\nu = 3$ / p=3

Figure 18: Relative contributions of the variance of each HDMR order ν, the variance of each polynomial order p, the relative number of HDMR terms, the relative number of quadrature points and the relative number of polynomials.

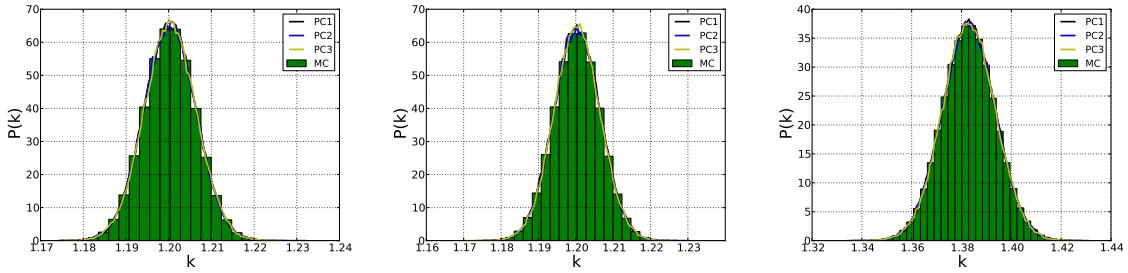
negative numbers and as such, it allows negative values of the nuclear data to be re-



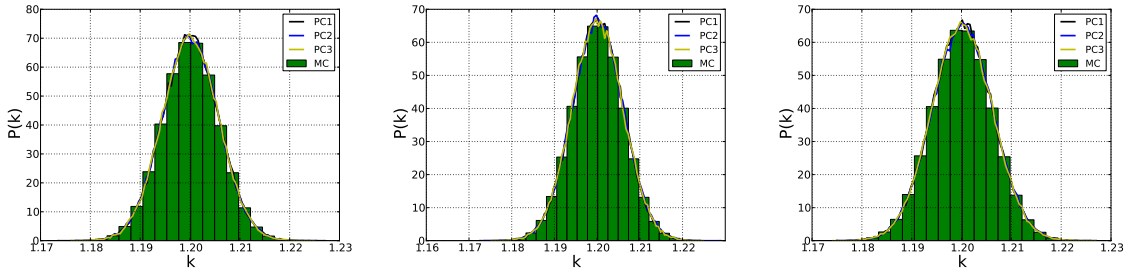
(a) UOX_{12} (b) UOX_{36} (c) UOX_{42}



(d) UOX_{104} (e) MOX_{114} (f) UOX_{136}



(g) MOX_{288} (h) MOX_{342} (i) UOX_{364}



(j) MOX_{416} (k) MOX_{832} (l) MOX_{988}

Figure 19: Probability density functions for the problems in Table (3) calculated using polynomial chaos and LHS. $\theta, \delta = 10^{-5}$.

alised when sampling; this is obviously un-physical. An alternative distribution, which is closely related to the Gaussian distribution, is the Log-Normal distribution. This distribution is sometimes preferable to the Gaussian as it removes any possibility of

generating negative valued samples.

In this section a comparison is made between Log-Normal and Gaussian distributions for one of the Rowlands' pin cell problems, namely UOX_{12}. The covariance data is again sourced from evaluated data and processed in the same way as in the previous section. There is however, an additional processing step which is required before the correlated nuclear data may be sampled according to the Log-Normal distribution. The covariance matrices must be transformed from a Log-Normal to a Gaussian space using Eqn.(20). The resulting Gaussian covariances were then decomposed using the Karhunen Loéve method; correlated samples were generated in the usual way and transformed back to the Log-Normal space using Eqn.(19).

To look at the efficiency of the HDMR method when using Log-Normal and Gaussian uncertainties, the relative error is plotted versus the number of realisations in Fig.(20). The reference solution is calculated using LHS with 10^6 samples. The accuracy and hence number of function evaluations required in the HDMR expansion was varied by decreasing the values of θ from $10^{-1} \rightarrow 10^{-10}$. The adaptive quadrature method was used with a value of $\delta = 10^{-5}$. The results in Fig.(20) show that the HDMR method is more efficient when using Gaussian distributed uncertainties when compared to Log-Normally distributed uncertainties for this problem. The results for the Gaussian uncertainties have fewer points since the number of terms which contribute a proportion of the variance above the threshold θ is small hence the algorithm only calculates a small number of components.

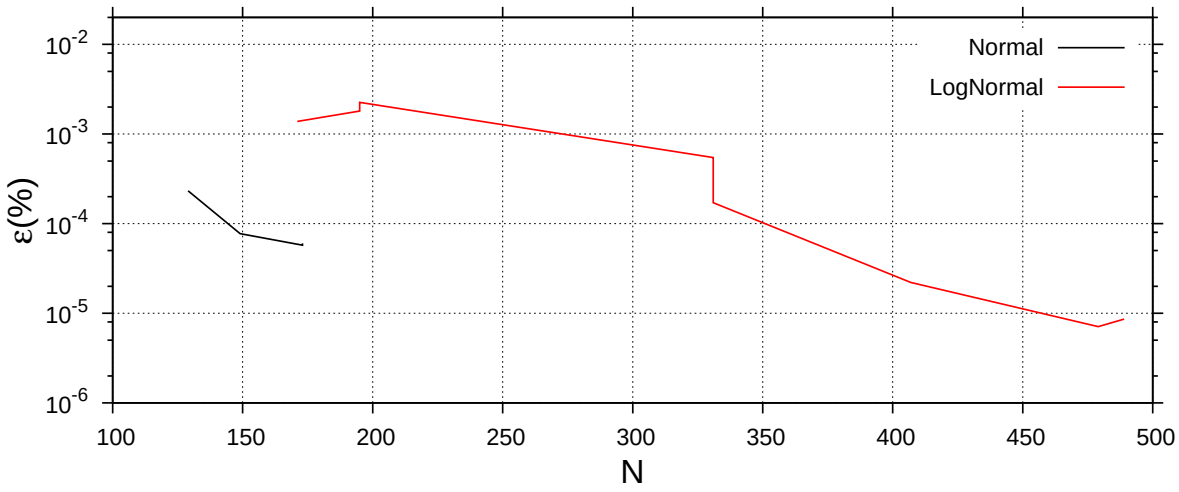


Figure 20: Relative error versus number of realisations for the UOX_{42} Rowlands' pin cell problem using Log-Normal and Gaussian distributed uncertainties.

To explain the differences in computational expense for the two different probability distributions, the variance ratio from Eqn.(99) is plotted for each distribution and the first three HDMR orders ($\nu = 1, 2, 3$) in Fig.(21). In Fig.(21a) we can see that as the adaptive HDMR threshold θ decreases, the variance contribution of the first order

terms for Gaussian uncertainties remains constant at a value of approximately 100%. This is compared to a value of approximately 97% for the LogNormal uncertainties. We also note that the number of components for the Gaussian case does not depreciate substantially with θ, whereas, the number of components for the Log-Normal case drops to around 10%. In Fig.(21b) the variance ratios and the component ratios for the second order HDMR components are plotted. Here we can see that the second order components contribute approximately 3% when using LogNormally distributed uncertainties whereas the contribution when assuming a Gaussian distribution is 10^{-2}%. We also note the increase in the relative number of second order components of the Log-Normal case which corresponds to this decrease in the first order terms. In Fig.(21c) the third order components are plotted. The variance contributions are of the order of 0.1% and 10^{-4} for Log-Normal and Gaussian uncertainties respectively. We also note that the threshold θ for the inclusion of third order Gaussian components is a lot lower than Log-Normal since the values of the components themselves are a lot smaller.

The reason for the increased contribution of the second and third order terms is probably due to the additional processing of the covariance data and the additional non-linearities which are introduced when sampling from a Log-Normal distribution.

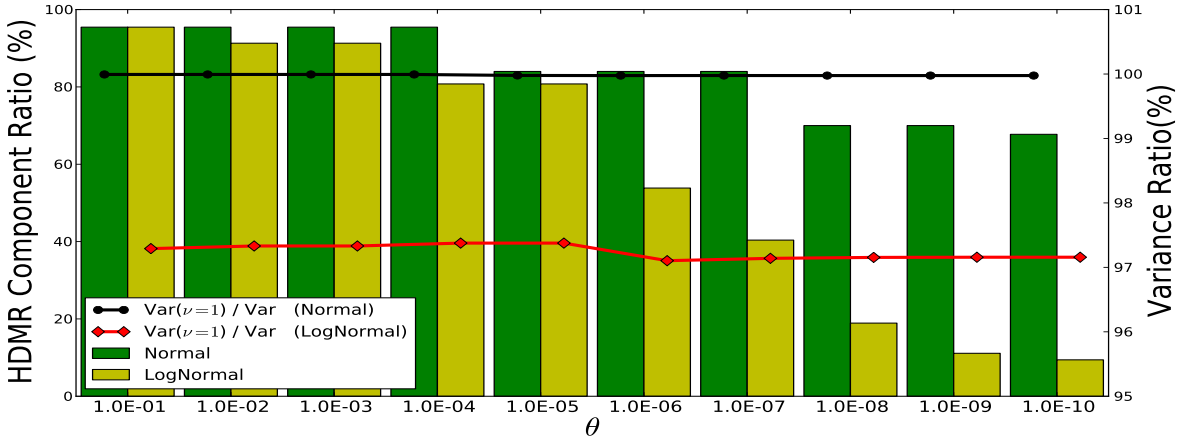
6. Summary and Conclusions

In this work, an adaptive HDMR method has been combined with an adaptive sparse grid quadrature scheme to iteratively compute the coefficients in a polynomial chaos expansion via projection.

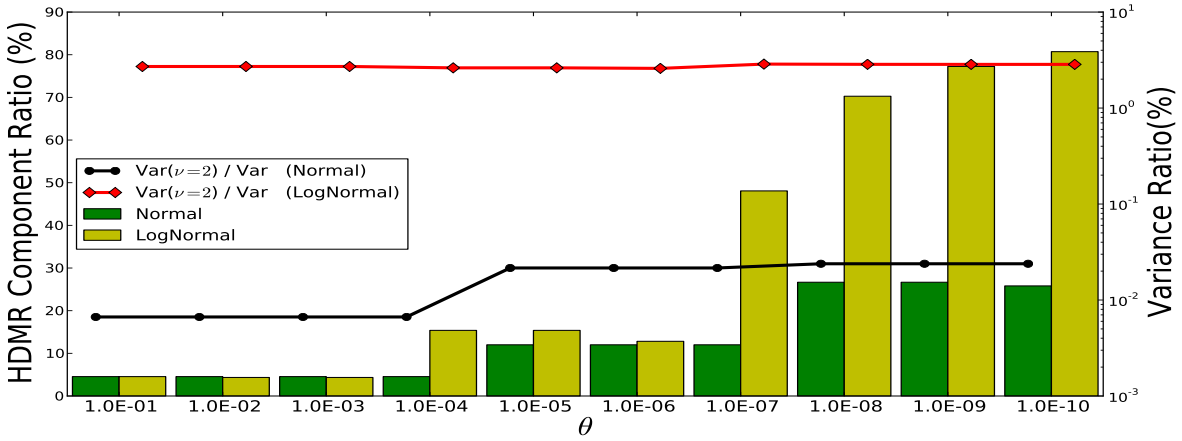
In Section (5.1) the adaptive sparse quadrature method was used to calculate a sparse polynomial expansion for a test response function. The results were for a problem with four random dimensions using Gaussian distributed uncertainties, uniform distributed uncertainties and a mixture of both. In all cases, the construction of a PCE using adaptive quadrature was far more efficient that the conventional sparse grid.

In Section (5.2) the adaptive sparse grid quadrature rules were used to calculate the components in the adaptive HDMR. The variance of a test response function was calculated in this way for a range of adaptive HDMR thresholds θ and also using a range of adaptive quadrature tolerances δ and non-adaptive quadrature levels L. The results showed that HDMR with adaptive quadrature is more efficient than with non-adaptive quadrature and as the tolerance θ is decreased, so must the adaptive quadrature tolerance δ.

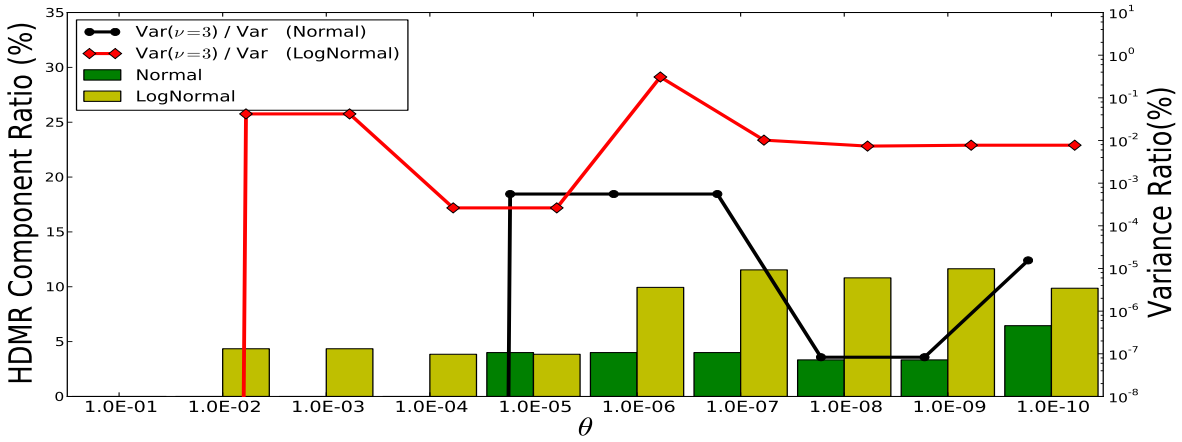
In Section (5.3) the adaptive HDMR method was compared to the adaptive quadrature algorithm by computing the variance of a test response function. The results showed that the adaptive quadrature is slightly more efficient than adaptive HDMR with adaptive quadrature. However, when the number of random variables used in the problem increased from 10 to 20, the adaptive quadrature algorithm took 10 times



(a) $\nu = 1$



(b) $\nu = 2$



(c) $\nu = 3$

Figure 21: HDMR component and variance ratios for the Rowlands' UOX_{12} problem using Gaussian and Log-Normal nuclear data uncertainties.

longer to compute the solution whereas, the HDMR method was comparatively insensitive to this change.

In Section (5.4.1), the adaptive HDMR method was used to compute the variance in k_{eff} due to evaluated nuclear data covariances for 12 Rowlands' pin cell problems. The number of uncertain parameters in these problems ranged from 12 to 988. HDMR solutions were compared to those computed using LHS with an equivalent number of samples. The results showed that the HDMR method was an order of magnitude more accurate than LHS when computing the variance for all problems up to 988 random dimensions. For all of these problems, where the uncertain nuclear data are described by a Gaussian distribution, the vast majority ($\geq 99.7\%$) of the calculated variance is due to the first order (uni-variate) HDMR components. The computation of the second and third order terms are inefficient for the Rowlands' problems when considering the contribution they provide to the final result and the number of calculations they require.

In Section (5.4.2) the variance in k_{eff} is calculated for one of the Rowlands' pin cell problems as in Section (5.4.1) but with the assumption that the uncertainties are described by a Log-Normal distribution. The results showed that the contribution to the variance of the second and third order terms were larger than those calculated for the case of Gaussian uncertainties. The second order terms contributed around 3% of the total variance for Log-Normal uncertainties and therefore should not be ignored.

In further work, we wish to investigate differential parameters such as reaction rates and leakages which may be subject to uncertainties other than nuclear data covariances. The purpose of this further work is to ascertain if response parameters other than k_{eff} exhibit non-linear dependencies that require high order HDMR terms for the accurate determination of response statistics. The inclusion of higher order terms in the HDMR expansion increases the computational cost prompting re-investigation of the efficiency of adaptive HDMR versus Latin Hypercube. We expect, with the inclusion of high order HDMR terms, that the efficiency of the HDMR approach will decrease making LHS more competitive. The exact number of dimensions, where the HDMR is still more efficient than LHS, will depend upon the number of non-linear HDMR terms and will hence be problem dependent.

7. Acknowledgements

Mr D. Ayres would like to acknowledge the support of the EPSRC under their industrial doctorate programme as well as industrial support from the national nuclear laboratory (NNL). Mr D. Ayres would also like to acknowledge the support of Mr R. P. Hiles and Dr. B. D. Turland from AMEC Winfrith for their support and discussion during this work. Dr M.D. Eaton would like to to acknowledge the support of the Royal Academy of Engineering under their fellowship programme. Dr. M.D. Eaton would also like to acknowledge the support of the EPSRC through the funding of the EPSRC Grant entitled "Adaptive hierarchical radiation transport methods to meet future challenges in reactor physics (EPSRC Grant EP/J002011/1)".

[1] Zoltán Perkó, Danny Lathouwers, Jan Leen Kloosterman, and Tim van der Hagen. Large scale applicability of a fully adaptive non-intrusive spectral projection technique: Sensitivity and uncertainty analysis of a transient. *Annals of Nuclear Energy*, 71(0):272 – 292, 2014.

[2] C.M. Cooling, M.M.R. Williams, E.T. Nygaard, and M.D. Eaton. The application of polynomial chaos methods to a point kinetics model of mipr: An aqueous homogeneous reactor. *Nuclear Engineering and Design*, 262(0):126 – 152, 2013.

[3] M.M.R. Williams. Uncertainty quantification in resonance absorption. *Annals of Nuclear Energy*, 47(0):124 – 133, 2012.

[4] Erin D. Fichtl. *Stochastic Methods for Uncertainty Quantification in Radiation Transport*. PhD thesis, University of New Mexico, 2009.

[5] Erin D. Fichtl and Anil K. Prinja. The stochastic collocation method for radiation transport in random media. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(4):646–659, 2011.

[6] D.A.F. Ayres, M.D. Eaton, A.W. Hagues, and M.M.R. Williams. Uncertainty quantification in neutron transport with generalized polynomial chaos using the method of characteristics. *Annals of Nuclear Energy*, 45:14–28, 2012.

[7] M. M. R. Williams and Matthew Eaton. A probabilistic study of the influence of parameter uncertainty on solutions of the neutron transport equation. *Progress in Nuclear Energy*, 52(6):580–588, 2010.

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

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

[10] Roger Ghanem. Ingredients for a general purpose stochastic finite elements implementation. *Computer Methods in Applied Mechanics and Engineering*, 168(1-4):19–34, 1999.

[11] Dongbin Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

[12] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element : a non intrusive approach by regression. *European Journal of Computational Mechanics*, 15:81–92, 2006.

[13] Géraud Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Université BLAISE PASCAL - Clermont II, 2009.

[14] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Doklady Akademii Nauk SSSR*, 4:240–243, 1963.

[15] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

[16] Thomas Gerstner and Michael Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71:2003, 2003.

[17] L Gilli, D Lathouwers, J L Kloosterman, and T H J J Van Der Hagen. Development of an Adaptive Nonintrusive Spectral Technique for Uncertainty Quantification and Application to a Multiphysics Problem. *Nuclear Science and Engineering*, 175(2):172–187, 2013.

[18] S. Rahman and H. Xu. A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics. *Probabilistic Engineering Mechanics*, 19(4):393 – 408, 2004.

[19] H. Xu and S. Rahman. A generalized dimension-reduction method for multi-dimensional integration in stochastic mechanics. *Int. J. Numer. Meth. Engng.*, 61(12):1992–2019, 2004.

[20] H. Xu and S. Rahman. Decomposition methods for structural reliability analysis. *Probabilistic Engineering Mechanics*, 20(3):239 – 250, 2005.

[21] Herschel Rabitz, Ömer F. Aliş, Jeffrey Shorter, and Kyurhee Shim. Efficient input-output model representations. *Computer Physics Communications*, 117:11 – 20, 1999.

[22] Herschel Rabitz and Ömer F. Aliş. General foundations of high dimensional model representations. *Journal of Mathematical Chemistry*, 25:197–233, 1999.

[23] Rajib Chowdhury, B. N. Rao, and A. Meher Prasad. High-dimensional model representation for structural reliability analysis. *Commun. Numer. Meth. Engng.*, 25(4):301–337, 2009.

[24] Xiu Yang, Minseok Choi, Guang Lin, and George Em Karniadakis. Adaptive anova decomposition of stochastic incompressible and compressible flows. *Journal of Computational Physics*, 231(4):1587 – 1614, 2012.

[25] Xiang Ma and Nicholas Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *Journal of Computational Physics*, 229(10):3884 – 3915, 2010.

[26] E. E. Lewis. *Computational Methods of Neutron Transport*. American Nuclear Society, 1993.

[27] Alain Hébert. *Applied Reactor Physics*. Presses Internationales Polytechnique, 2009.

[28] C.R.E. de Oliveira. An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering. *Progress in Nuclear Energy*, 18(12):227 – 236, 1986.

[29] Erik Vanmarcke. *Random Fields: Analysis and Synthesis*. MIT Press, 1983.

[30] M Loeve. *Probability Theory*. Springer-Verlag, 1977.

[31] P. Spanos and R. Ghanem. Stochastic finite element expansion for random media. *Journal of Engineering Mechanics*, 115(5):1035–1053, 1989.

[32] M. Rosenblatt. Remarks on a multivariate transformation. *Annals of Mathematical Statistics*, 23(3):470472, 1952.

[33] A. Der Kiureghian and P. L. Liu. Structural reliability under incomplete information. *J. Eng. Mech*, page 85104, 1986.

[34] G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[35] M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[36] J.C. Helton and F.J. Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69, 2003.

[37] R.E. MacFarlane and A.C. Kahler. Methods for processing ENDF/B-VII with NJOY. *Nuclear Data Sheets*, 111(12):2739 – 2890, 2010.

[38] T.D. Newton and L.J. Hutton. The next generation wims lattice code: Wims9. In *PHYSOR*, 2002.

[39] M J Halsall. Cactus, a characteristics solution to the neutron transport equations in complicated geometries. Technical report, United Kingdom Atomic Energy Establishment, 1980.

[40] D.J. Powney and T.D. Newton. Overview of the wims 9 resonance treatment. Technical Report ANSWERS/WIMS/TR.26, Serco Assurance, 2004.

[41] R. Cameron and W. Martin. The orthogonal development of nonlinear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, 48(2):385–392, 1947.

[42] D. Ayres, S. Park, and M.D. Eaton. Propagation of input model uncertainties with different marginal distributions using a hybrid polynomial chaos expansion. *Annals of Nuclear Energy*, 66(0):1 – 4, 2014.

[43] O.P.Le Maitre and O.M.Knio. *Spectral Methods for Uncertainty Quantification*. Springer, 2010.

[44] T.N.L Patterson. The optimum addition of points to quadrature formulae. *Mathematics of Computation*, 22:847–856, 1968.

[45] Alan Genz and B.D Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. *Journal of Computational and Appied Mathematics*, 71:299–309, 1996.

[46] Grzegorz W. Wasilkowski and Henryk Wozniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11:1–56, 1995.

[47] F. Nobile, R. Tempone, and C. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.

[48] Andrea Saltelli. *Sensitivity Analysis*. Wiley, 2000.

[49] Sudret Bruno. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, July 2008.

[50] Z. Zhang, M. Choi, and G. Karniadakis. Error estimates for the anova method with polynomial chaos interpolation: Tensor product functions. *SIAM Journal on Scientific Computing*, 34(2):A1165–A1186, 2012.

[51] I.M. Sobol'. Theorems and examples on high dimensional model representation. *Reliability Engineering & System Safety*, 79(2):187 – 193, 2003.

[52] J.L. Rowlands. The pincell benchmarks and other calculational benchmarks. Technical Report JEF/DOC-494, OECD/NEA Paris, 1994.