Strain and stress analysis of uncertain engineering systems

By D. Ghosh, C. Farhat AND P. Avery

1. Motivation and objectives

Structural properties of any material show variability among different samples of the material resulting from natural variability in microstructure and from the manufacturing process. The manufacturing process also causes variability in the geometric properties of the components made from these materials. These variabilities induce uncertainty in the predicted response of a physical system. Additional factors such as uncertainty in external loading, and in some cases insufficient details about the underlying physics such as behavior of the joints contribute to magnify the uncertainty in this predicted response. These uncertainties can be modeled and their effects can be analyzed using a probability theory based framework.

Among the response quantities, the strains and stresses are often of the greatest interest in an engineering problem. However, the current literature on probabilistic engineering mechanics focuses on the issue of finding the displacement field; the issue of finding the strains and stresses is not sufficiently addressed. This paper addresses the computational issues related to strain and stress computation. Here the problem of uncertainty analysis is posed in a stochastic finite elements framework.

Among the probabilistic methods of uncertainty analysis, Stochastic Finite Element Methods (SFEM) (Ghanem & Spanos 2003) have gained considerable attention recently. The major advantages of these methods are their ability to handle stochastic processes, encapsulated representation of random quantities, and lower computational cost.

Let $(\Omega, \mathcal{F}, P)$ denote a probability space, where Ω is the set of the outcomes θ of physical experiments, $\mathcal{F}$ is a σ-algebra in Ω, and P is a probability measure on $\mathcal{F}$. Let $\mathcal{X}$ denote the physical domain of the system. Consider

$$\mathcal{L}(u) = f \ , \tag{1.1}$$

where $\mathcal{L} = \mathcal{L}(\theta)$ and $f = f(\theta)$. Randomness in the parameters of the underlying physical system induces randomness in $\mathcal{L}$ and f. Some of these random parameters can be modeled as random variables $\{\eta_i(\theta)\}_{i=1}^{i=r}$ and some as random processes $\kappa(\boldsymbol{x}, \theta)$, where $\boldsymbol{x} \in \mathcal{X}$. For example, a spring stiffness can be modeled as a random variable, whereas the thickness of a plate can be modeled as a random field. The processes $\kappa(\boldsymbol{x}, \theta)$ can be discretized using a random basis set $\{\eta_i(\theta)\}_{i=r+1}^{i=s}$ in $L_2(\Omega, \mathcal{F}, P)$, where the coefficients of the random variables $\{\eta_i(\theta)\}_{i=r+1}^{i=s}$ turn out to be functions of the parameter $\boldsymbol{x}$. For example, if the covariance function $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$ of the process $\kappa(\boldsymbol{x}, \theta)$ is known then the process can be discretized using the Karhunen-Loève expansion (Ghanem & Spanos 2003) as

$$\kappa(\boldsymbol{x}, \theta) = \sum_{i=0}^{\infty} \sqrt{\lambda_i} \phi^{(i)}(\boldsymbol{x}) \eta_i(\theta) \ \ , \tag{1.2}$$

where λ_i are the eigenvalues of the covariance kernel $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$, arranged in descending order, $\phi^{(i)}$ are the corresponding eigenvectors, and $\eta_i(\theta)$ are zero-mean and orthonormal random variables. For computational convenience this series is truncated after the first few terms. The set of all random variables $\{\eta_i(\theta)\}_{i=1}^{i=s}$ completely characterizes the uncertainty in the underlying system. These random variables are characterized by their joint probability measure, if this measure is not Gaussian, these variables can be transformed into a nonlinear functional of an independent Gaussian vector $\{\xi_i(\theta)\}_{i=1}^{i=m}$ (Ghanem & Doostan 2006, Das *et al.* 2006); the integer m is often referred as stochastic dimension of the problem (Ghanem & Spanos 2003, Debusschere *et al.* 05). This new set of independent standard random variables will be denoted by an m-dimensional vector $\boldsymbol{\xi}$. Thus, $\mathcal{L}(\theta)$ and $f(\theta)$ can now also be denoted by $\mathcal{L}(\boldsymbol{\xi})$ and $f(\boldsymbol{\xi})$. The solution u is also a function of $\boldsymbol{\xi}$, yielding the notation $u(\boldsymbol{\xi})$, or more specifically $u(\boldsymbol{x}, \boldsymbol{\xi})$. The formulation presented in this paper is valid for $\boldsymbol{\xi}$ being non-Gaussian as well, and thus it is equally applicable to a variety of expansions (Xiu & Karniadakis 2003, LeMaitre *et al.* 2004).

Once $\mathcal{L}(\boldsymbol{\xi})$ and $f(\boldsymbol{\xi})$ are constructed, the solution $u(\boldsymbol{x}, \boldsymbol{\xi})$ is next represented in polynomial chaos expansion (PCE) (Ghanem & Spanos 2003), where a square-integrable random process is expressed as

$$u(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{i=0}^{\infty} u^{(i)}(\boldsymbol{x}) \psi_i(\boldsymbol{\xi}) \quad , \tag{1.3}$$

where $\psi_i(\boldsymbol{\xi})$ are the Hermite polynomials, and $u^{(i)}(\boldsymbol{x})$ are deterministic coefficients called as *chaos coefficients*. For computational purposes, the series is truncated after a finite number of terms, yielding

$$u(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{i=0}^{P-1} u^{(i)}(\boldsymbol{x}) \psi_i(\boldsymbol{\xi}) \ . \tag{1.4}$$

The index P is determined by the stochastic dimension and the highest order of polynomial chaos to be retained in the expansion. For example, in a second-order expansion in two stochastic dimension $P = 6$ and the polynomials $\psi_i(\xi_1, \xi_2)$ are (Ghanem & Spanos 2003)

$$\begin{aligned} &\psi_0(\xi_1, \xi_2) = 1 \ , && \psi_1(\xi_1, \xi_2) = \xi_1 \ , && \psi_2(\xi_1, \xi_2) = \xi_2 \ , \\ &\psi_3(\xi_1, \xi_2) = \xi_1^2 - 1 \ , && \psi_4(\xi_1, \xi_2) = \xi_1 \xi_2 \ , && \psi_5(\xi_1, \xi_2) = \xi_2^2 - 1 \ . \end{aligned}$$

The chaos coefficients $u^{(i)}(\boldsymbol{x})$ can be computed by minimizing either the error in the solution or the residual in the equation (Ghanem & Spanos 2003). In both cases, a Galerkin approach is used to find the optimal solution. The second method, when applied to a linear statics problem of the form

$$K(\boldsymbol{\xi}) u(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) \ , \quad K(\boldsymbol{\xi}) \in \mathbb{R}^{n \times n}, \quad u(\boldsymbol{\xi}), f(\boldsymbol{\xi}) \in \mathbb{R}^n \ , \tag{1.5}$$

where

$$K(\boldsymbol{\xi}) = \sum_{i=0}^{L-1} K^{(i)} \psi_i(\boldsymbol{\xi}) \ , \qquad f(\boldsymbol{\xi}) = \sum_{i=0}^{M-1} f^{(i)} \psi_i(\boldsymbol{\xi}) \ , \qquad u(\boldsymbol{\xi}) = \sum_{i=0}^{P-1} u^{(i)} \psi_i(\boldsymbol{\xi}) \ ,$$
$$P > L, M, \quad K^{(i)} \in \mathbb{R}^{n \times n}, \quad f^{(i)}, u^{(i)} \in \mathbb{R}^n \ , \tag{1.6}$$

yields a system of linear deterministic equations of the form

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \ , \quad \boldsymbol{K} \in \mathbb{R}^{nP \times nP}, \quad \boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^{nP} \ , \tag{1.7}$$

where

$$\boldsymbol{K} = \begin{bmatrix} \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_0 \psi_0 \rangle & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_1 \psi_0 \rangle & \dots & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_{P-1} \psi_0 \rangle \\ \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_0 \psi_1 \rangle & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_1 \psi_1 \rangle & \dots & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_{P-1} \psi_1 \rangle \\ \dots & \dots & \dots & \dots \\ \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_0 \psi_{P-1} \rangle & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_1 \psi_{P-1} \rangle & \dots & \sum_{i=0}^{L-1} K^{(i)} \langle \psi_i \psi_{P-1} \psi_{P-1} \rangle \end{bmatrix} , \tag{1.8}$$

$$\boldsymbol{u} = \left\{ \begin{array}{c} u^{(0)} \\ u^{(1)} \\ \cdot \\ u^{(P-1)} \end{array} \right\} , \qquad \boldsymbol{f} = \left\{ \begin{array}{c} \langle \psi_0^2 \rangle f^{(0)} \\ \langle \psi_1^2 \rangle f^{(1)} \\ \cdot \\ \langle \psi_{M-1}^2 \rangle f^{(M-1)} \\ \cdot \\ 0 \end{array} \right\} , \tag{1.9}$$

and $\langle \cdot \rangle$ denotes the mathematical expectation operator:

$$\langle g(\boldsymbol{\xi}) \rangle = \int_{\mathbb{R}^m} g(\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \ , \tag{1.10}$$

where $g(\boldsymbol{\xi})$ is any function of the m-dimensional random vector $\boldsymbol{\xi}$ and $p(\boldsymbol{\xi})$ is the joint probability density function (PDF) of $\boldsymbol{\xi}$. To solve (1.7) efficiently, the sparsity of matrix $\boldsymbol{K}$ can be exploited and iterative solvers can be employed, as described in Section 4. Once the coefficients $u^{(i)}$ are estimated, any statistical moment and the PDF of $u(\boldsymbol{x}, \boldsymbol{\xi})$ can be computed using (1.4). For example, the mean (the first moment) and standard deviation (square root of the second moment about the mean) are

$$\langle u(\boldsymbol{x}, \boldsymbol{\xi}) \rangle = u^{(0)}(\boldsymbol{x}) \ , \tag{1.11}$$

and

$$stdev\,(u(\boldsymbol{x}, \boldsymbol{\xi})) = \sqrt{\sum_{i=1}^{P-1} {u^{(i)}}^2(\boldsymbol{x}) \, \langle \psi_i^2(\boldsymbol{\xi}) \rangle} \ , \tag{1.12}$$

respectively.

However, in a real engineering problem the displacement field is of very little interest; the strains and stresses are considered to be more important. It will be shown here that often computation of the statistical moments of strains and stresses is not as simple as that of the displacement field. This is because an orthogonal expansion such as PCE is not easily obtainable for such strains and stresses (explained in details in the following section). Von Mises stress by any numerical method using $50,000$ grid points takes about

45 minutes, which is significant. There may be additional quantities of interest such as strains, which will require additional computational time. Moreover, as previously mentioned, these numerical techniques will bring additional overhead of computational and programming complexity.

These issues are not addressed at length in the current literature, the goal of this paper is to discuss these issues and to find a set of feasible solutions. Cases where numerical integration must be used are identified, and a few integration techniques are considered. These techniques are further studied for their computational cost and complexity in implementation.

The stress tensor components σ_{ij} can be derived from the strain tensor components ε_{ij} using the constitutive relationship. For the isotropic materials these stress components are

$$\sigma_{ii} = \frac{E}{(1-2\nu)(1+\nu)}\{(1-\nu)\varepsilon_{ii} + \nu\varepsilon_{jj} + \nu\varepsilon_{kk}\} \qquad i \neq j \neq k \ , \tag{1.13}$$

and

$$\sigma_{ij} = \frac{E}{2(1+\nu)}\varepsilon_{ij} \qquad i \neq j \ , \tag{1.14}$$

where E denotes the Young's modulus and ν denotes the Poisson's ratio. The Von Mises stress can be computed as

$$\sigma_{VM} = \sqrt{I_1^2 - 3I_2} \ , \tag{1.15}$$

where

$$I_1 = \sigma_{ii} + \sigma_{jj} + \sigma_{kk} \quad \text{and} \quad I_2 = \sigma_{ii}\sigma_{jj} + \sigma_{jj}\sigma_{kk} + \sigma_{kk}\sigma_{ii} - \tau_{ij}^2 - \tau_{jk}^2 - \tau_{ki}^2 \ . \tag{1.16}$$

Evaluation of the mean and standard deviation of these stress and strain quantities will be considered next. In a nondeterministic case the displacement field in three spatial directions can be represented in a form similar to (1.4) as

$$u_i(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{p=0}^{P-1} u_i^{(p)}(\boldsymbol{x})\psi_p(\boldsymbol{\xi}) \qquad i,j = 1,2,3 \ . \tag{1.17}$$

The strain tensor components are derived from the displacement field as

$$\varepsilon_{ii} = \frac{\partial u_i}{\partial x_i} \quad \text{and} \quad \varepsilon_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \quad \text{for} \quad i \neq j \ ; \qquad i,j = 1,2,3 \ . \tag{1.18}$$

For an uncertain system, using (1.17) and (1.18), the representation of the strain components become

$$\varepsilon_{ij}(\boldsymbol{x},\boldsymbol{\xi}) = \sum_{p=0}^{P-1} \varepsilon_{ij}^{(p)}(\boldsymbol{x})\psi_p(\boldsymbol{\xi}) \ . \tag{1.19}$$

For brevity, the arguments $\boldsymbol{x}$ and $\boldsymbol{\xi}$ are dropped hereafter. The principal strains ε_{pr_i} ($i = 1,2,3$) are eigenvalues of the matrix $\boldsymbol{E}$, thus they are highly nonlinear functions of the strain components ε_{ij}. Thus mean and standard deviations of the principal strains cannot be easily computed using orthogonality among the polynomials; a numerical technique is needed for this. The same is true for the Von Mises strains.

A numerical integration technique is therefore is needed for computing the statistical moments of the terms that cannot be evaluated using the orthogonality property of chaos polynomials. A few techniques are considered and compared here for their computational efficiency and algorithmic complexity. These are (i) Monte Carlo simulation, (ii) Latin hypercube sampling, (iii) Gaussian quadrature on the standard tensor product grid, and (iv) Smolyak cubature. These methods are presented below.

2. Numerical integration

The integral of the following form will be considered here:

$$I = \int_{\mathcal{D}} g(s)p(s)ds \ , \tag{2.1}$$

where $\mathcal{D} \in \mathbb{R}^k$ is the domain of integration, $g(s)$ is the integrand, and $p(s)$ is the weighing function. For example, in a probabilistic context, if $p(s)$ is a joint probability density function of a random vector s then the above integral yields the statistical moment of the function $g(s)$. In our case $k = m$ and $s = \boldsymbol{\xi}$. A numerical integration rule to evaluate this integral typically looks like

$$\hat{I} = \sum_{i=1}^{N} w_i g(s_i) \ , \tag{2.2}$$

where s_i are the points where the function $g(s)$ is evaluated according to the integration scheme and w_i are the corresponding weights. In this paper the terms *quadrature points* and *nodal points* will be used synonymously. In the probabilistic methods of integration these points are often referred as realizations of the random variable s and N is called sample size. Evaluation of multi-dimensional integrals deserves special attention to manage the increasing computational cost with dimension. Numerical integration can be categorized into two different classes, probabilistic methods (Evans & Swartz 2000, Liu 2001) and deterministic methods (Abramowitz & Stegun 1984, Stroud & Secresr 1966). Here two probabilistic methods, Monte Carlo simulation and Latin hypercube sampling, and two deterministic methods, Gaussian quadrature with standard tensor product and Smolyak cubature are considered.

2.1. *Monte Carlo (MC) simulation*

In this method the quadrature points s_i are generated using a random number generator such that their probability density function becomes $p(s)$. The weights w_i are $1/N$ (Evans & Swartz 2000, Liu 2001). The procedure is explained with a specific example: computing the statistical moments of the Von Mises stress. To do this, at first realizations of random numbers $\boldsymbol{\xi}$ are generated. Each of these realizations corresponds to a realization of the random system model. Then for each realization, the following steps are followed. First, using (1.19), realization of the strain tensor components is computed, simultaneously realization of the material properties such as E, ν is computed, which yields the corresponding realization of the constitutive matrix. From the strain tensor components and constitutive matrix, realization of the stress tensor components is computed. Then the corresponding realization of Von Mises stress is computed using (1.15) and (1.16). Finally, over numerous realizations the statistical moments of the Von Mises stress are computed. The main advantage of this method is its straightforward implementation.

However, this method needs a large number of realizations to obtain a good estimate of the statistical moments; thus it is computation intensive.

2.2. *Latin hypercube sampling*

Latin hypercube sampling (LHS) (Mckay *et al.* 1979) is a method of selecting the integration points to achieve faster convergence than the standard Monte Carlo method. This is a variance reduction technique (pp 391 of Davis & Rabinowitz 1984, pp 183 of Evans & Swartz 2000). As a result, for any fixed error level LHS requires a smaller sample size than MC to estimate the integral. The computational procedure of the integral evaluation is exactly the same as the Monte Carlo simulation, only the sampling points (that is, the realizations of random numbers $\boldsymbol{\xi}$) are different. Usage of LHS in SFEM applications has also been addressed in (Choi et al. 2004, Olsson & Sandberg 2002, Ghiocel & Ghanem 2002),

2.3. *Gaussian quadrature on standard tensor product grid*

This is a deterministic quadrature rule (Abramowitz & Stegun 1984, Stroud & Secresr 1966) that has also been used in SFEM (Field 2002). For an integration of form (2.1) this method can evaluate the integral exactly if $g(s)$ is a polynomial. In our case, the $p(s)$ is the distribution of a Gaussian random vector. Gaussian quadrature with Gaussian distribution as weighing function is also known as Gauss-Hermite quadrature. In one dimension, an N-point quadrature rule can exactly evaluate the integral of a polynomial of order up to $(2N-1)$. For a given quadrature rule in one dimension, its extension to a multidimensional case is as follows.

Let an m_i point numerical integration rule for a function g of one variable s^i be

$$U^i(g) = \sum_{j=1}^{m_i} a^i_j g(s^i_j) \ , \tag{2.3}$$

where s^i_j are quadrature points, and a^i_j are weights. This integration rule can then be extended to a d-dimensional integration using the Kronecker product as

$$\hat{I} = (U^{i_1} \otimes \cdots \otimes U^{i_d})(g) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} (a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_d}_{j_d}) g(s^{i_1}_{j_1} \cdots s^{i_d}_{j_d}) \ , \tag{2.4}$$

where $m_{i_1}, \ldots, m_{i_d}$ are the number of quadrature points used for defining integration rule in variables $s^{i_1}, \ldots, s^{i_d}$, respectively. Thus, computation of the above integral needs $(m_{i_1} \cdots m_{i_d})$ function evaluations. For example, if an m-point integration rule is used in each dimension in a d-dimensional space, then evaluation of a d-dimensional integral in this space needs m^d function evaluations. The computational burden therefore increases significantly with dimension d. This is often referred as the *curse of dimensionality.*

2.4. *Smolyak cubature*

In order to avoid the so-called *curse of dimensionality*, another approach of numerical integration, Smolyak cubature (Smolyak 1963), is widely used. Here, instead of using the full tensor product grid a recursive contribution of lower-order tensor products is used to estimate the integral, stated as

$$\hat{I}_{q,d} = \sum_{q-d+1 \le |\boldsymbol{i}| \le q} (-1)^{q-|\boldsymbol{i}|} \binom{d-1}{q-|\boldsymbol{i}|} (U^{i_1} \otimes \cdots \otimes U^{i_d})(g) \ , \tag{2.5}$$

$$\boldsymbol{i} = \{i_1, \ldots, i_d\}, \quad |\boldsymbol{i}| = i_1 + \ldots + i_d \ .$$

Here $\hat{I}_{q,d}$ is the cubature rule, $q \geq d$ is a parameter to be selected, $\boldsymbol{i}$ is a d-dimensional index set, U^{i_k} are 1-dimensional quadrature rules as in (2.3), and $m_{i_j} \neq m_{i_k}$ for $i_j \neq i_k$. Instead of considering $U^{i_k}(g)$ as the integration rule in any particular variable s^{i_k}, it can be viewed as a quadrature rule for any dimension. In fact, a closer examination of (2.5) reveals that the one-dimensional quadrature rules are permuted in all dimensions. These 1-dimensional quadrature rules can be selected as Gaussian quadrature rules, with $m_{i_1} = 1, m_{i_2} = 2, \ldots$. Computational cost saving in Smolyak cubature compared to the standard tensor product integration rule increases as d, the dimension of the problem grows. It is proved in (Heiss & Winschel 2006) that if every univariate quadrature rule U^i as in (2.3) in the sequence $\{U^i, \quad i = i_1, i_2, \ldots\}$ can integrate any univariate polynomial of order up to $(2m_i - 1)$ exactly, then the Smolyak quadrature (2.5) yields the exact integral of a d-variate polynomial of total order up to $(2M - 1)$, where

$$M = \max_{i_1 \leq i_k \leq i_d} m_{i_k} \ . \tag{2.6}$$

3. Numerical integration of a polynomial integration: dimension reduction

When the integrand admits a polynomial form and the order of the polynomial is less than the dimension or the number of independent variables involved, then the integration cost can be reduced significantly by using a reduced dimensional integration, described here. Let us define the term *effective dimension* of a function to refer to the maximum number of independent random variables ξ_i present in the function; for example, the *effective dimension* of $\xi_5^4\xi_8$ is 2, since it involves two independent random variables: ξ_5 and ξ_8. Let a random quantity be expressed as an expansion of polynomials of d-dimensional orthonormal random variables, and the highest order of the polynomial in the expansion be p. If $p < d$ then the highest *effective dimension* of any term in the expansion is p, and not d. Obviously a p-dimensional integration rule is computationally more efficient than a d-dimensional rule. If a d-dimensional integration rule is used to evaluate such function, it only contributes to increase the computational cost, without affecting the result. Thus in this situation a p-dimensional integration rule is sufficient and computationally more efficient. In this paper such reduced dimensional integration will be referred as Integration in Reduced Dimension (IRD). For general cases where IRD is not used, and instead a full dimensional integration is carried out which will be referred to as Integration in Full Dimension (IFD).

4. Numerical study

Two numerical studies are conducted and presented here. In the first study numerical integration of a polynomial expansion is considered. Three deterministic integration techniques are compared regarding the computational cost to integrate a fourth order polynomial expansion. These techniques are: (i) Gaussian quadrature with standard tensor product grid in full dimension (IFD), (ii) Gaussian quadrature with standard tensor product grid in reduced dimension (IRD), and (iii) Smolyak cubature. In the second numerical study a plate model with random material properties is considered and the first two statistical moments of a few strain and stress quantities of this model are computed using some of the methods described previously. Computational issues such as speed and complexity in implementation are discussed.

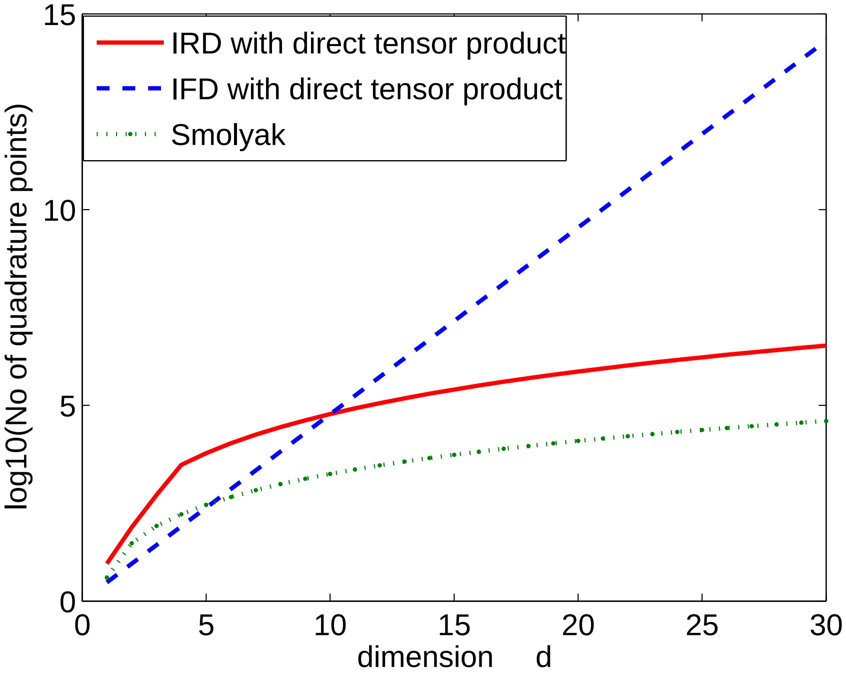

FIGURE 1. Number of quadrature points needed in different deterministic integration schemes: fourth degree polynomial expansion

4.1. *Example 1*

Throughout this paper the Gauss-Hermite quadrature is used as the basic univariate integration rule for the deterministic integration techniques, that is, as $U^i(g)$ in (2.3) and as $U^{i_1}(g)$ in (2.5). For a fourth order polynomial expansion, the number of quadrature points needed by three exact (and deterministic) methods, IFD with Kronecker product, IRD with Kronecker product, and Smolyak cubature are plotted in Fig. 1. From this plot it is noted that IRD with Kronecker product yields a significant improvement over IFD with Kronecker product, especially as the dimension increases. The Smolyak cubature is found to be the most computational efficient among the three methods. It is further noted that for lower dimensional integration, the number of quadrature points needed by Smolyak cubature exceeds the number of points needed by IFD with direct tensor product . It is experienced that in terms of complexity in implementation, the sequence is reverse. That is Smolyak is the most complex and IFD with Kronecker product is the simplest. The complexity of Smolyak algorithm arises primarily from the need of some recursive functions to generate the grid points.

4.2. *Example 2*

In the second study a square plate model is used to compare numerical integration schemes to compute the statistical moments of stresses and strains. The dimensions of the plate are $2.3m \times 2.3m \times 5mm$. The plate is built using 20 metal strips joined side by side along the edges, and the dimensions of each strip are $2.3m \times 0.115m \times 5mm$, as shown in Fig. 2. Young's modulus of the strips are assumed to be random, modeled as

$$E_i = \bar{E}_i + stdev_{E_i}\xi_i \qquad i = 1, \ldots, 20 \ , \tag{4.1}$$

thus $L = 20$ in (1.6). Here $\bar{E}_i$ and $stdev_{E_i}$ are the mean and standard deviation of the Young's modulus of the material in strip i, and ξ_i are independent standard normal random variables. The plate is fixed along its four edges. A finite element model with 400 square elements is used. Out of the twenty different materials, ten are chosen to be of mean Young's modulus $2.0MPa$ and ten of mean $2.1MPa$; standard deviations for all the materials are 20% of the respective mean values. Density of the plate material is assumed to be $7860Kg/m^3$, the same as steel. In addition to the self-weight, the plate is loaded by three concentrated forces in its center, $400KN$, $300KN$, and $-2KN$ in directions x, y, and z, respectively. Displacement is represented using second-order chaos expansion as in (1.4) with $P = 231$. The chaos coefficients are computed by solving (1.7). To solve the system of deterministic equations a preconditioned conjugate gradient method is used with Block-Jacobi preconditioner. The matrix-vector (mat-vec) product computation is optimized by performing them in the block level (Pellissetti & Ghanem 2000). After estimating the chaos coefficients $u^{(i)}$ of the displacement field, standard deviation of the Von Mises stress are computed by Smolyak cubature and plotted in Fig. 3. Techniques for computing the chaos polynomials at a given grid point can be found in (Ghanem & Spanos 2003, Debusschere *et al.* 05). Next using MC simulation, LHS, and Smolyak cubature $\bar{\sigma}_{vm}$ and $stdev(\sigma_{vm})$ are computed. Values of these quantities at an arbitrary chosen node (node 45) are plotted in Figs. 4 - 5 against the number of quadrature points used. In this paper the random numbers are generated using Matlab. Usually the computer-generated random numbers do not satisfy the orthonormal properties for a finite sample size. Thus the generated random numbers are orthonormalized using a transformation (Ang & Tang 1984), and the new numbers are used for MC- and LHS-related computation. From (1.15), it is clear that the σ^2_{VM} is a sixth order polynomial expansion. Following the discussion in Section 2.4, the cubature rule needs $M = 4$ for the exact evaluation of $< \sigma^2_{VM} >$, where M is defined in (2.6). For $d = 20$, it is found that the numbers of quadrature points in Smolyak cubature are 861, 12341, and 135751 for $M = 2$, 3, and 4, respectively.

The resullts of our study led to the following observations. Although it is previously shown that for Smolyak cubature $M = 4$ or 135751 quadrature points are needed to exactly evaluate $< \sigma^2_{VM} >$, it is observed from the plots that $M = 3$ or 12341 quadrature points suffice. Although not plotted here, this trend was observed for most other FE nodes as well. There were only a few nodes for which changing M from 3 to 4 changed the estimate of the integral to some extent. Thus, for most of the nodes, contribution from the higher-order terms in the expansion of $< \sigma^2_{VM} >$ is very small. It is also observed that in general all the three methods MC, LHS and Smolyak cubature provided a fair level of approximation of the statistical moments.

Transformation of the machine-generated random numbers helped improving the convergence of MC and LHS. No significant difference is noticed between the performance of MC and LHS. One possible reason is that the integrands are not monotonic functions of the arguments, thus variance reduction in LHS is not guaranteed (Mckay *et al.* 1979).

5. Conclusions

Stochastic dimension of the system, that is, the number of basic random variables involved in uncertainty modeling, is an important factor in selecting an integration al-

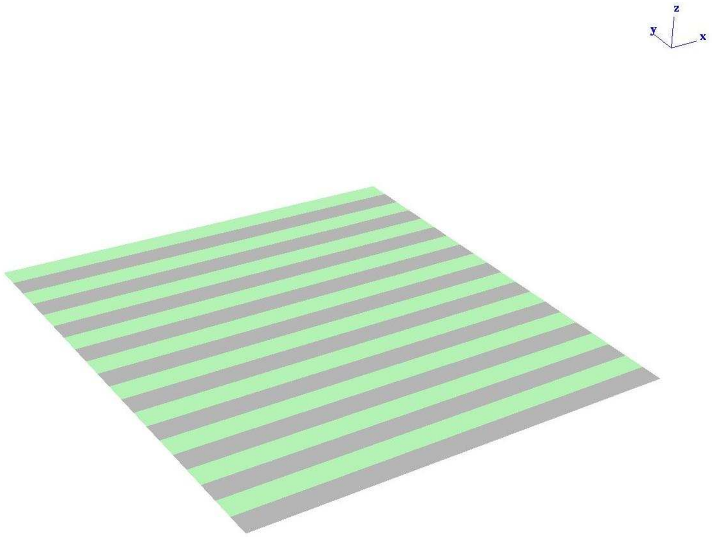

Figure 2. The plate model

gorithm. If the dimension is very low (say less than four or five) then computational cost is not very high for any of the methods. In that case even Gaussian quadrature with full tensor product rule can be used to achieve good accuracy as well as to enjoy lower algorithmic complexity. As the dimension increases, the computational cost of this method becomes prohibitive, and more advanced techniques such as Monte Carlo sampling, Latin hypercube sampling, or Smolyak cubature should be used. It is observed from the numerical experiments that all of these three methods work satisfactorily. In the simulation-based methods, the error is random, thus there is a probability of the error being higher than tolerance. This probability decreases as more number of realizations or quadrature points are included.

Thus according to the increasing computational cost, the deterministic methods can be arranged as Smolyak cubature, Integration in Reduced Dimension, Integration in Full Dimension (IFD) in Kronecker tensor product grid. According to the increasing complexity in implementation, the sequence becomes Integration in Full Dimension with Kronecker tensor product, Integration in Reduced Dimension, Smolyak cubature. Implementation of Monte Carlo and Latin hypercube sampling are as easy as Integration in Full Dimension with Kronecker tensor product grid, excluding the additional complexity to generate the random numbers needed for these methods. The choice of a particular integration scheme depends upon complexity level that can be afforded, availability of subroutine, and level of accuracy needed.

6. Future plans

According to the current literature SFEM has been applied to a variety of problems. However, most of these applications are on fairly small-scale problems. Thus, with the vision of analyzing and quantifying uncertainty of real-life problems currently we are working on applying domain decomposition techniques to solve SFEM-related problems in large-scale systems.

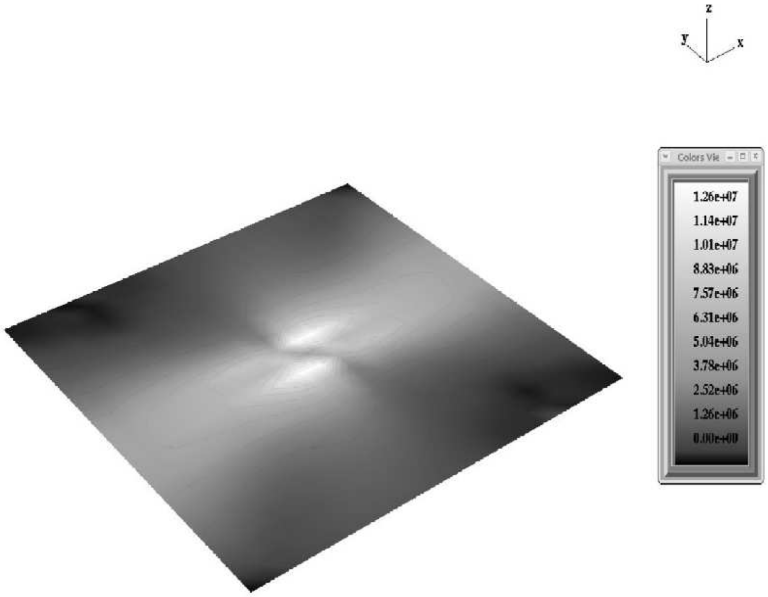

FIGURE 3. Standard deviation of Von Mises stress ($stdev(\sigma_{vm})$) computed by IFD, used Smolyak cubature with $M = 4$

Acknowledgment

Financial support from the Advanced Simulation and Computing Program of the Department of Energy is gratefully acknowledged.

REFERENCES

CHOI, S. K. & GRANDHI, R. V. & CANFIELD, R. A. & PETTIT, C. L. 2004 Polynomial chaos expansion with Latin hypercube sampling for estimating response variability. *AIAA Journal* **42(6)**, 1191-1198.

DAS S. & GHANEM R. & SPALL J. 2006, Asymptotic sampling distribution for polynomial chaos representation of data: A maximum entropy and Fisher information approach. *45th IEEE Conference on Decision and Control, San Diego, CA* December 13-15.

DEBUSSCHERE JB & NAJM HN & PBAY PP & KNIO OM & GHANEM RG & LE MAÎTRE OP 2005 Numerical challenges in the use of polynomial chaos representations for stochastic processes *SIAM Journal on Scientific Computing archive* **26(2)**, 698–719.

FIELD R. V. JR. 2002 Numerical methods to estimate the coefficients of the polynomial chaos expansion *15th ASCE Engineering Mechanics Conference, Columbia University, New York, NY* June 2-5

GHANEM R. & DOOSTAN A. 2006 On the construction and analysis of stochastic predictive models: Characterization and propagation of the errors associated with limited data. *Journal of Computational Physics* **217(1)**, 63-81.

GHIOCEL D. M. & GHANEM R. 2002 Stochastic finite-element snalysis of seismic soil-structure interaction *Journal of Engineering Mechanics* **128(1)**, 66-77.

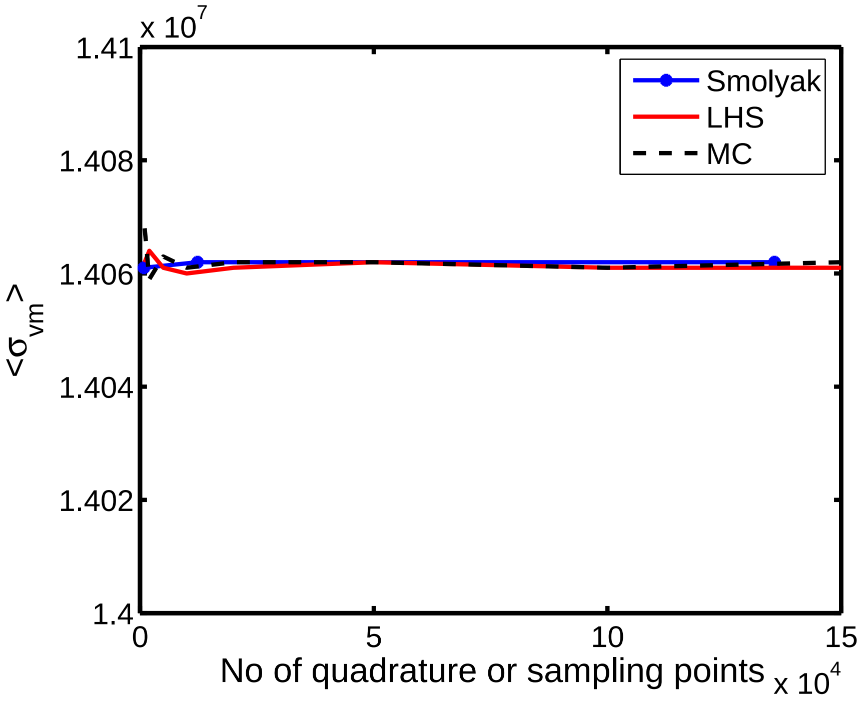

FIGURE 4. Mean Von Mises stress $\bar{\sigma}_{vm}$ (or $\langle\sigma_{vm}\rangle$) at node 45: convergence of different methods

HEISS F. & WINSCHEL V. 2006 Estimation with numerical integration on sparse grids *Discussion Papers in Economics* **15**

LE MAITRE O. P. & NAJM H. & GHANEM R. & KNIO O. 2004 Multi-resolution analysis of wiener-type uncertainty propagation schemes. *J. Comp. Ph.* **197(2)**, 502-531.

MATLAB User's Guide, Version 7 *The MathWorks, Inc. MA*

MCKAY M. D. & BECKMAN R. J. & CONOVER W. J. 1979 A comparison of three methods for selecting input variables in the analysis of output from a computer code. *Technometrics* **21(2)**, 239-245.

OLSSON A. M. J. & SANDBERG G. E. 2002 Latin hypercube sampling for stochastic finite element analysis *Jl. Engng. Mech.* **128(1)**, 121-125.

PELLISSETTI M. & GHANEM R. 2000 Iterative solution of systems of linear equations arising in the context of stochastic finite elements *Advances in Engng. Software* **31**, 607-616.

SMOLJAK S. A. 1963 Quadrature and interpolation formulas for tensor products of certain classes of functions *Soviet Math. Dokl.* **4**, 240-243.

ABRAMOWITZ M. & STEGUN I. 1984 Handbook of mathematical functions with formulas, graphs, and mathematical tables *John Wiley & Sons Inc.*

ANG A. H.-S. & TANG W. H. 1984 Probability Concepts in Engineering Planning and Design, Vol II Decision, Risk and Reliability *John Wiley & Sons*

GHANEM R. & SPANOS P. D. 2003 Stochastic Finite Elements: A Spectral Approach *Revised Edition, Dover Publications*

DAVIS P.J. & RABINOWITZ P. 1984 Methods of numerical integration *Academic Press*

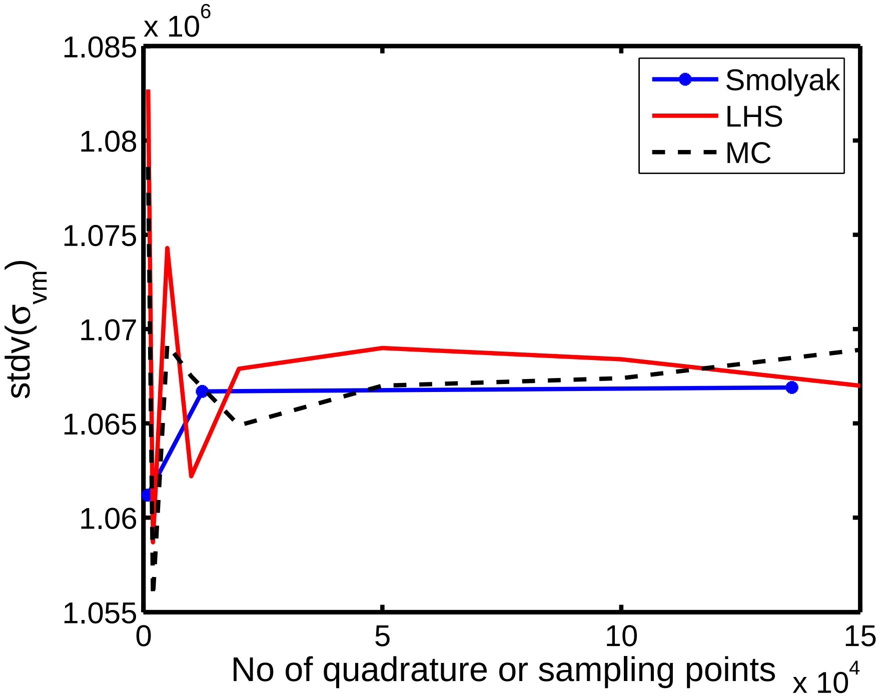

Figure 5. Standard deviation of Von Mises stress $stdev(\sigma_{vm})$ at node 45: convergence of different methods

Evans M. & Swartz T. Approximating integrals via Monte Carlo & deterministic methods *Oxford University Press*

Liu J. S. 2001 Monte Carlo strategies in scientific computing *Springer-Verlag*

Stroud A. H. & Secresr D. 1966 Gaussian quadrature formulas *Prentice-Hall Inc*

Xiu D. & Karniadakis G. 2003 Modeling uncertainty in flow simulations via generalized polynomial chaos. *J. Comp. Phys.* **187(1)**, 137-167.