ASSESSMENT OF COLLOCATION AND GALERKIN APPROACHES TO LINEAR DIFFUSION EQUATIONS WITH RANDOM DATA

Howard C. Elman,[1,] Christopher W. Miller,[2] Eric T. Phipps,[3] & Raymond S. Tuminaro[4]*

[1]*Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*

[2]*Department of Applied Mathematics and Scientific Computation, University of Maryland, College Park, MD 20742, USA*

[3]*Sandia National Laboratories, PO Box 5800, MS 1318, Albuquerque, NM 87185, USA*

[4]*Sandia National Laboratories, PO Box 969, MS 9159, Livermore, CA 94551*

We compare the performance of two methods, the stochastic Galerkin method and the stochastic collocation method, for solving partial differential equations (PDEs) with random data. The stochastic Galerkin method requires the solution of a single linear system that is several orders larger than linear systems associated with deterministic PDEs. The stochastic collocation method requires many solves of deterministic PDEs, which allows the use of existing software. However, the total number of degrees of freedom in the stochastic collocation method can be considerably larger than the number of degrees of freedom in the stochastic Galerkin system. We implement both methods using the Trilinos software package and we assess their cost and performance. The implementations in Trilinos are known to be efficient, which allows for a realistic assessment of the computational complexity of the methods. We also develop a cost model for both methods which allows us to examine asymptotic behavior.

1. PROBLEM STATEMENT

We investigate the linear elliptic diffusion equation with zero Dirichlet boundary conditions, where diffusivity is given by a random field. If D is an open subset of $\mathbb{R}^n$ and (Ω, Σ, P) is a complete probability space, then this can be written as

$$-\nabla \cdot [a(\mathbf{x}, \omega)\nabla u(\mathbf{x}, \omega)] = f(\mathbf{x}, \omega) \qquad (\mathbf{x}, \omega) \in D \times \Omega \qquad (1)$$

$$u(x, \omega) = 0 \qquad (\mathbf{x}, \omega) \in \partial D \times \Omega.$$

The random input field is often given as a truncated Karhunen–Loève (KL) expansion [1] or by a polynomial chaos (PC) expansion [2]. The truncated KL expansion is given by

---

*Correspond to Howard C. Elman, E-mail: elman@cs.umd.edu

$$a(\mathbf{x}, \omega) \approx \hat{a}_M(\mathbf{x}, \boldsymbol{\xi}(\omega)) = a_0(\mathbf{x}) + \sum_{k=1}^{M} \sqrt{\lambda_k} \xi_k(\omega) a_k(\mathbf{x}), \tag{2}$$

where (λ_i, a_i) are solutions to the integral equation

$$\int_D C(\mathbf{x}_1, \mathbf{x}_2) a_i(\mathbf{x}_2) d\mathbf{x}_2 = \lambda_i a_i(\mathbf{x}_1), \tag{3}$$

and C is the covariance kernel of the random field. That is, (λ_i, a_i) are eigenvalues and eigenfunctions of the covariance operator $\mathcal{C}$ defined by

$$[\mathcal{C}(\alpha)](x_1) = \int_D C(x_1, x_2) \alpha(x_2) \, dx_2. \tag{4}$$

The random variables are uncorrelated, mean zero, and are given by

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}} \int_D [a(\mathbf{x}, \omega) - a_0(\mathbf{x})] a_k(\mathbf{x}) \, d\mathbf{x}. \tag{5}$$

We make the further modeling assumption that the random variables $\{\xi_k\}$ are independent and admit a joint probability density of the form $\rho(\boldsymbol{\xi}) = \prod_{k=1}^{M} \rho_k(\xi_k)$. The covariance kernel is positive semidefinite and its eigenvalues can be ordered so that $\lambda_1 \geq \lambda_2 \geq ... \geq 0$. To ensure the existence of a unique solution to (1) it is necessary to assume that the diffusion is uniformly bounded away from zero; we assume that there exist constants $a_{\min}$ and $a_{\max}$ such that

$$0 < a_{\min} \leq \hat{a}_M(\mathbf{x}, \boldsymbol{\xi}) \leq a_{\max} < \infty, \tag{6}$$

almost everywhere P-almost surely, $\hat{a}_M(\cdot, \boldsymbol{\xi}) \in L_2(D)$ P-almost surely, and $\hat{f}_M \in L_2(\Omega) \otimes L_2(D)$.

The goal of this paper is to model the computational costs and compare the performance of the stochastic Galerkin method [3–7] and the sparse grid collocation method [8–10] for computing the solution of (1) (cf. [11] for related work). Section 2 outlines the stochastic Galerkin method. Section 3 outlines the sparse grid collocation method. Section 4 presents our model of the computational costs of the two methods. Section 5 explores the performance of the methods applied to several numerical examples using the *Trilinos* software package [12]. Finally, in Section 6 we draw some conclusions.

2. STOCHASTIC GALERKIN METHOD

Define $\Gamma = \times_{k=1}^{M} \Gamma_k = \times_{k=1}^{M} Im(\xi_k)$ and let

$$\langle u, v \rangle = \int_{\Gamma} u(\boldsymbol{\xi}) v(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) \, d\boldsymbol{\xi} = \int_{\Omega} u[\boldsymbol{\xi}(\omega)] v[\boldsymbol{\xi}(\omega)] \, dP \tag{7}$$

be the inner product over the space $L_2(\Gamma) = \{v(\boldsymbol{\xi}) : ||v||_{L_2(\Gamma)}^2 = \langle v^2 \rangle < \infty\}$. We can define a variational form of (1) in the stochastic domain by the following: For all $\mathbf{x} \in D$, find $u(\mathbf{x}, \boldsymbol{\xi}) \in L_2(\Gamma)$ such that

$$-\langle \nabla \cdot (a \nabla u), v \rangle = \langle f, v \rangle \tag{8}$$

for all $v \in L_2(\Gamma)$. This leads to a set of coupled second-order linear partial differential equations (PDEs) in the spatial dimension. It is common in the literature to combine (8) with a variational formulation of the spatial component of the problem, which after discretization of both the spatial and stochastic components, leads to the stochastic finite element method. A variant of this approach, which we use, is to discretize in space by finite differences. Details are as follows.

Define S_p to be the space of multivariate polynomials in ξ, of total degree at most p. This space has dimension $N_\xi = [(M+p)!]/(M!p!)$. Let $\{\Psi_k\}_{k=0}^{N_\xi-1}$ be a basis for S_p orthonormal with respect to the inner product (7). Substituting KL-expansions for $a(\mathbf{x}, \omega)$ and $f(\mathbf{x}, \omega)$ and restricting (8) to $v \in S_p$ gives

$$-\int_\Gamma \nabla \cdot \left[\hat{a}_M(\mathbf{x}, \xi) \left(\sum_{i=0}^{N_\xi-1} \nabla u_i(\mathbf{x})\Psi_i\right)\right] \Psi_j \, d\xi = \int_\Gamma \hat{f}_M \Psi_j \, d\xi \quad \forall j = 0 : N_\xi - 1. \tag{9}$$

This is a set of coupled second-order differential equations for the unknown functions $u_i(\mathbf{x})$ defined on D, which can then be discretized using finite differences. This gives rise to a global linear system of the form

$$A\vec{u} = \vec{f}. \tag{10}$$

In practice the random variables appearing in the KL expansion of $a(\mathbf{x}, \omega)$ and $f(\mathbf{x}, \omega)$ would be different since the diffusivity and loading terms would typically have different correlation structures. In this case one would expand a, f, and u as

$$a(\mathbf{x}, \omega) \approx \hat{a}_M(\mathbf{x}, \xi) = a_0(\mathbf{x}) + \sum_{k=1}^{M} \sqrt{\lambda_k} \xi_k a_k(\mathbf{x}) \tag{11}$$

$$f(\mathbf{x}, \omega) \approx \hat{f}_M(\mathbf{x}, \tilde{\xi}) = f_0(\mathbf{x}) + \sum_{k=1}^{M} \sqrt{\tilde{\lambda}_k} \tilde{\xi}_k f_k(\mathbf{x}) \tag{12}$$

$$u(\mathbf{x}, \omega) \approx u(\mathbf{x}, \xi_1, ..., \xi_M, \tilde{\xi}_1, ..., \tilde{\xi}_k), \tag{13}$$

where $\tilde{\lambda}_k$ and $\tilde{\xi}_k$ are the eigenvalues and random variables appearing in the KL expansion of f. For the sake of simplicity we choose to ignore this issue and proceed as if the random variables appearing in the KL expansion of f and a are the same.

With orderings of $\vec{u}$ and $\vec{f}$ (equivalently, the columns and rows of A, respectively) corresponding to a blocking by spatial degrees of freedom, $\left(\vec{u}^T = \left[u_1^T, u_2^T, \cdots, u_{N_\xi}^T\right]\right)$, the coefficient matrix and right-hand side have the tensor product structure

$$A = \sum_{k=0}^{M} G_k \otimes A_k, \quad \vec{f} = \sum_{k=0}^{M} \vec{g_k} \otimes \vec{f_k}. \tag{14}$$

The matrices $\{G_k\}$ and the vectors $\{g_k\}$ depend only on the stochastic basis,

$$G_0(i,j) = \langle \Psi_i \Psi_j \rangle, \; g_0(i) = \langle \Psi_i \rangle = \delta_{i0}, \tag{15}$$

$$G_k(i,j) = \langle \xi_k \Psi_i \Psi_j \rangle, \; g_k(i) = \langle \xi_k \Psi_i \rangle, \; (k > 0).$$

The matrices $\{A_k\}$ correspond to a standard five-point operator for $-\nabla \cdot (a_k \nabla u)$, and $\{f_k\}$ are the associated right-hand side vectors. In the two-dimensional examples we explore below, we use a uniform mesh of width h. The discrete difference operators are formed by using the following five-point stencil

$$\begin{bmatrix} & a_k\left(x, y+\dfrac{h}{2}\right) & \\ a_k\left(x-\dfrac{h}{2}, y\right) & a_k(x,y) & a_k\left(x+\dfrac{h}{2}, y\right) \\ & a_k\left(x, y-\dfrac{h}{2}\right) & \end{bmatrix}. \tag{16}$$

The matrix A_k is symmetric for all k and A is positive-definite by (6). Since the random variables appearing in (5) are mean-zero, it also follows from (6) that A_0 is positive-definite.

The matrix A is of order $N_x N_\xi$, where N_x is the number of degrees of freedom used in the spatial discretization. It is also sparse in the block sense due to the orthogonality of the stochastic basis functions. Specifically, since the random variables $\{\xi_k\}$ are assumed to be independent, we can construct the stochastic basis functions $\{\Psi_i\}$ by taking tensor products of univariate polynomials satisfying the orthogonality condition

$$\langle \psi_i(\xi_k), \psi_j(\xi_k) \rangle_k = \int_{\Gamma_k} \psi_i(\xi_k)\psi_j(\xi_k)\rho_k(\xi_k)\,d\xi_k = \delta_{ij}. \tag{17}$$

This basis is referred to as the generalized polynomial chaos of order p. The use of this basis for representing random fields is discussed extensively in [4] and [7]. The univariate polynomials appearing in the tensor product can be expressed via the familiar three-term recurrence

$$\psi_{i+1}(\xi_k) = (\xi_k - \alpha_i)\psi_i(\xi_k) - \beta_i\psi_{i-1}(\xi_k), \tag{18}$$

where $\psi_0 = 1$, $\psi_{-1} = 0$. It follows that

$$G_0(i,j) = \langle \Psi_i, \Psi_j \rangle = \prod_{k=1}^{M} \langle \psi_{i_k}(\xi_k), \psi_{j_k}(\xi_k) \rangle_k = \prod_{k=1}^{M} \delta_{i_k j_k} = \delta_{ij}, \tag{19}$$

and for $k > 0$ the entries in G_k are

$$G_k(i,j) = \langle \xi_k \Psi_i, \Psi_j \rangle = \langle \xi_k \psi_{i_k}, \psi_{j_k} \rangle_k \prod_{l=1, l\neq k}^{M} \langle \psi_{i_l}, \psi_{j_l} \rangle_l$$

$$= (\langle \psi_{i_k+1}, \psi_{j_k} \rangle_k + \alpha_{i_k}\langle \psi_{i_k}, \psi_{j_k} \rangle_k + \beta_{i_k}\langle \psi_{i_k-1}, \psi_j \rangle_k) \prod_{l=1, l\neq k}^{M} \langle \psi_{i_l}, \psi_{j_l} \rangle_l. \tag{20}$$

Thus G_0 is diagonal and G_k has at most three entries per row for $k > 0$. Furthermore, if the density functions ρ_k are symmetric with respect to the origin, i.e., $\rho_k(\xi_k) = \rho_k(-\xi_k)$, then the coefficients α_i in the three-term recurrence are all zero and G_k then has at most two non-zeros per row.

The stochastic Galerkin method requires the solution to the large linear system (10). Once the solution to (10) is obtained, statistical quantities such as moments or a probability distribution associated with the solution process can be obtained cheaply [4]. Although the Galerkin linear system is large, there are techniques available by which this task can be performed efficiently. We elect to directly solve the large symmetric and positive-definite Galerkin system using the conjugate gradient (CG) method. CG only requires the evaluation of matrix–vector products so that it is unnecessary to store the assembled matrix A. The matrix–vector products can be performed implicitly following a procedure described in [13]. Each matrix A_k is assembled and the matrix–vector product is expressed as $(Au)_j = \sum_{i=0}^{N_\xi-1} \sum_{k=0}^{M} \langle \xi_k \Psi_i \Psi_j \rangle (A_k u_i)$. The terms $A_k u_i$ are precomputed and then scaled as needed. This approach is efficient since most of the terms $\langle \xi_k \Psi_i \Psi_j \rangle$ are zero. The cost of performing the matrix–vector product in this manner is essentially determined by the computation of $A_k u_i$ for $0 \le k \le M$ and $0 \le i \le N_\xi - 1$, which entails $(M+1)N_\xi$ sparse matrix–vector products by matrices $\{A_k\}$ of order N_x. The implicit matrix–vector product also only requires the assembly of $M+1$ order-N_x stiffness matrices and the assembly of the components $\langle \xi_k \Psi_i \Psi_j \rangle$ of $\{G_k\}$. Alternatively, one could assemble the entire Galerkin matrix and perform the block matrix–vector product in the obvious way. This is, of course, less efficient in terms of memory usage since it requires the assembly and storage of many matrices of the form $\langle \xi_k \Psi_i \Psi_j \rangle (A_k u_i)$. It is also shown in [13] that performing the matrix–vector products in this way is less efficient in terms of memory bandwidth.

To obtain fast convergence, we will also use a preconditioner. In particular, it has been shown in [14] that an effective choice is an approximation to $A_0^{-1} \otimes G_0^{-1}$, where A_0 is the mean stiffness matrix. Since the stochastic basis functions are orthonormal, G_0 is the identity matrix. The preconditioner then entails the approximate action of N_ξ uncoupled copies of A_0^{-1}. For this we use a single iteration of an algebraic multigrid solver provided by [15].

3. SPARSE GRID COLLOCATION

An alternative to the Galerkin scheme is the collocation method, which samples the input operator at a predetermined set of points $\Theta = \left\{ \boldsymbol{\xi}^{(1)}, ..., \boldsymbol{\xi}^{(n)} \right\}$ and constructs a high-order polynomial approximation to the solution function using discrete solutions to the deterministic PDEs

$$-\nabla \cdot \left[\hat{a}_M \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right) \nabla u \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right) \right] = \hat{f}_M \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right), \tag{21}$$

where the diffusion coefficients are evaluated at the sample points. Once the polynomial approximation to u is constructed, statistical information can be obtained at low cost [10], as for the stochastic Galerkin method.

For simplicity of presentation, we first discuss a collocation method using the full tensor product of one-dimensional point sets. Let $\{\psi_i\}$ be the set of polynomials orthogonal with respect to the measure ρ_k. Let $\theta_i = \{\xi : \psi_i(\xi) = 0\} := \left\{ \xi_{i,k}^{(j)} \right\}_{j=1}^{i+1}$ for $i = 1, 2, ...,$ and $j = 1, 2, ..., i$. These are the abscissas for an (i)-point Gauss quadrature rule with respect to the measure ρ_k. A one-dimensional (i)-point interpolation operator is given by

$$U^i(u)(\xi) = \sum_{j=1}^{i} u \left(\xi_i^{(j)} \right) l_i^{(j)}(\xi), \; l_i^{(j)}(\xi) = \prod_{n=1, n \neq j}^{i} \frac{\xi - \xi_i^{(n)}}{\xi_i^{(j)} - \xi_i^{(n)}}. \tag{22}$$

These can be used to construct an approximation to the M-dimensional random function $u(\mathbf{x}, \boldsymbol{\xi})$ by defining a tensor interpolation operator

$$U^{i_1} \otimes \cdots \otimes U^{i_M}(u)(\boldsymbol{\xi}) = \sum_{j_1=1}^{i_1} \cdots \sum_{j_M=1}^{i_M} u \left(\xi_{i_1}^{(j_1)}, \cdots, \xi_{i_M}^{(j_M)} \right) \left(l_{i_1}^{(j_1)} \otimes \cdots l_{i_M}^{(j_M)} \right). \tag{23}$$

The evaluation of this operator requires the solution of a collection of deterministic PDEs (21), one for each sample point in $\Theta_{\text{tensor}} = \times_{j=1}^{M} \theta_{i_j}$.

This method suffers from the so-called curse of dimensionality, since the number of sample points $|\Theta_{\text{tensor}}| = \prod_{j=1}^{M} |\theta_{i_j}| = \prod_{j=1}^{M} (i_j)$ grows exponentially with the dimension of the problem. This makes tensor-product collocation inappropriate for problems where the stochastic dimension is moderate or large. This cost can be significantly reduced using sparse grid methods [10].

Sparse grid collocation methods are based on the Smolyak approximation formula. The Smolyak operator $\mathcal{A}(p, M)$ is a linear combination of the product formulas in (23). Let $Y(p, M) = \{ \boldsymbol{i} \in \mathbb{N}^M : p + 1 \leq |\mathbf{i}|_1 \leq p + M \}$. Then the Smolyak formula is given by

$$\mathcal{A}(p, M)(u) = \sum_{\mathbf{i} \in Y(p,M)} (-1)^{p+M-|\mathbf{i}|_1} \binom{M-1}{p+M-|\mathbf{i}|_1} (U^{i_1} \otimes \cdots \otimes U^{i_M}). \tag{24}$$

The evaluation of the Smolyak formula requires the solution of deterministic PDEs (21) for $\boldsymbol{\xi}^{(l)}$ in the set of points

$$\Theta_{p,M} = \bigcup_{\mathbf{i} \in Y(p,M)} (\theta_{i_1} \times \cdots \times \theta_{i_M}). \tag{25}$$

For moderate or large values of M, $|\Theta_{p,M}| \ll |\Theta_{\text{tensor}}|$.

If Gaussian abscissas are used in the definition of θ_i and if u is an M-variate polynomial of total degree p in $\boldsymbol{\xi}$, then $u = \mathcal{A}(p, M)u$ [11]; that is, the Smolyak interpolant exactly reproduces such polynomials.[1] We refer to the

---

[1] An alternative choice of sparse grid points is to use the Clenshaw–Curtis abscissas with $|\theta_1| = 1$ and $|\theta_i| = 2^{i-1} + 1$ for $i > 1$, which produces nested sparse grids [9, 10, 16]. The choice used here, non-nested Gaussian abscissas with a linear growth rate, $|\theta_i| = i$, produces grid sets of cardinalities comparable to those for the nested Clenshaw–Curtis grids, i.e., $|\Theta_{p,M}^{\text{Gaussian}}| \approx |\Theta_{p,M}^{\text{Clenshaw-Curtis}}|$.

parameter p in $\mathcal{A}(p, M)$ as the sparse grid level. It is shown in [9] that sampling the differential operator on the sparse grid $\Theta_{p,M}$ will produce $\mathcal{A}(p, M)(u) = u_p$, where u_p is an approximate solution to (1) of similar accuracy to the solution obtained using an order p stochastic Galerkin scheme. The sparse grid will have on the order of 2^p more points than there are stochastic degrees of freedom in the Galerkin scheme, $|\Theta| \approx 2^p N_\xi$ for $M \gg 1$ [10].

For a fully nonintrusive collocation method, the diffusion coefficients of (21) would be sampled at the points in the sparse grid, and for each sample the deterministic stiffness matrix would be constructed for the PDE,

$$-\nabla \cdot \left[\hat{a}_M \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right) \nabla u \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right) \right] = \hat{f}_M \left(\mathbf{x}, \boldsymbol{\xi}^{(l)} \right). \tag{26}$$

This repeated assembly can be very expensive. We elect in our implementations to take advantage of the fact that the stiffness matrix at a given value of the random variable is a scaled sum of the stiffness matrices appearing in (14). For a given value of $\boldsymbol{\xi}$, the deterministic stiffness matrix can be expressed as

$$A(\boldsymbol{\xi}) = A_0 + \sum_{k=1}^{M} \xi_i A_k. \tag{27}$$

In our implementation we assemble the matrices $\{A_k\}$ first and then compute the scaled sum (27) at each collocation point. This is somewhat intrusive in that this method may not be compatible with with existing deterministic solvers; however, it greatly reduces the amount of time required to perform assembly in the collocation method.

One could construct a separate multigrid preconditioner for each of the deterministic systems. This can become very expensive, as the cost of constructing an algebraic multigrid (AMG) preconditioner can often be of the same order as the iterative solution. This repeated cost can be eliminated if one simply builds an algebraic preconditioner for the mean problem A_0^{-1} and applies this preconditioner to all of the deterministic systems. If the variance of the operator is small, then the mean-based AMG preconditioner is nearly as effective as doing AMG on each subproblem and saves time in setup costs. Other techniques for developing preconditioners balancing performance with the cost of repeated construction are considered in [16].

4. MODELING COMPUTATIONAL COSTS

From an implementation perspective, collocation is quite advantageous in that it requires only a modest modification to existing deterministic PDE applications. Collocation samples the stochastic domain at a discrete set of points and requires the solution of uncoupled deterministic problems. This can be accomplished by repeatedly invoking a deterministic application with different input parameters determined by the collocation point-sampling method. A Galerkin method, on the other hand, is much more intrusive as it requires the solution of a system of equations with a large coefficient matrix which has been discretized in both spatial and stochastic dimensions. To better understand the relationship between these two methods, we develop a model for the computational costs.

We begin by stating in more detail some of the computational differences between the two methods. The Galerkin method requires the computation of the matrices $G_0 = \langle \Psi_i \Psi_j \rangle$ and $G_k = \langle \xi_k \Psi_i \Psi_j \rangle$ associated with the stochastic basis functions, the assembly of the right-hand side vector and the spatial stiffness matrices $\{A_k\}$, and finally, the solution to the large coupled system of equations. Collocation requires the construction of a sparse grid and the derivation of an associated sparse grid quadrature rule, and the assembly/solution of a series of deterministic subproblems. Further, as observed above, the number of sample points needed for collocation tends to be much larger than the dimension of the Galerkin system required to achieve comparable accuracy.

In this study we examine only methods which are isotropic in the stochastic dimension, allocating an equal number of degrees of freedom to each stochastic direction. Anisotropic versions of both the sparse grid collocation method and the stochastic Galerkin could be implemented by weighting the maximum degree of the approximation space in each direction. This has been explored in the case of sparse grid collocation [17]. We expect a cost comparison for an anisotropic stochastic Galerkin method and the anisotropic sparse grid collocation method to be comparable to that of their isotropic counterparts. Additional modifications to the stochastic collocation for adaptively dealing with very high dimensional problems are considered in [18, 19]. We do not consider these methods here.

For a fixed M, p, let Z_G be the number of preconditioned conjugate gradient (PCG) iterations required to solve the Galerkin system, let $N_\xi \alpha$ be the cost of applying the mean-based preconditioner during a single iteration of the stochastic Galerkin method, and let $N_\xi \gamma$ be the cost of a single matrix–vector product for (10), where α and γ are constants. Note in particular that α is constant because of the optimality of the multigrid computation. Then the total cost of the Galerkin method can be modeled by

$$\text{Galerkin cost} = N_\xi Z_G (\alpha + \gamma). \tag{28}$$

The parameter γ can be thought of as the number of order-N_x matrix–vector products required per block row in the stochastic Galerkin matrix. When implementing the implicit matrix–vector product, γ is equal to $M + 1$.

We can model the costs of the collocation method with the mean-based multigrid preconditioner by

$$\text{Collocation cost} = Z_C 2^p N_\xi (\alpha + 1), \tag{29}$$

where p is the Smolyak grid level, N_ξ is the number of degrees of freedom needed by an order p Galerkin system, Z_C is the average number of PCG iterations needed to solve a single deterministic system, and $\alpha + 1$ is the cost of the preconditioning operation and a single order-N_x matrix–vector product. The factor of 2^p derives from the relation between the number of degrees of freedom for the stochastic Galerkin and sparse grid collocation methods for large M.

In our application, we fix the multigrid parameters as follows: One V-cycle is performed at each iteration and within each V-cycle one symmetric Gauss–Seidel iteration is used for both presmoothing and postsmoothing. The coarsest grid is assumed coarse enough so that a direct solver can be used without affecting the cost per iteration; in our implementations we use a 1×1 grid. These parameters were chosen to optimize the run time of a single deterministic solve. The cost to apply a single multigrid iteration is roughly equivalent to 5–6 matrix products (two matrix–vector products for fine-level presmoothing, another two for fine-level postsmoothing, and one matrix–vector product for a fine-level residual calculation). Thus, α can be assumed to be 5 or 6 after accounting for computational overhead.

The relative costs of the two methods depend on the parameter values. In particular,

$$\frac{\text{Collocation cost}}{\text{Galerkin cost}} = \left(\frac{Z_C}{Z_G} \right) 2^p \frac{(\alpha + \gamma)}{(\alpha + 1)}. \tag{30}$$

If, for example, the ratio of iteration counts (Z_G / Z_C) is close to 1 and the preconditioning costs dominate the matrix vector costs (i.e., $\alpha \gg \gamma$), then we can expect the stochastic Galerkin method to outperform the sparse grid collocation method because of the factor 2^p. Alternatively, if γ is comparable compared to α, the preconditioning cost, then collocation is more attractive. The cost of the two methods is identical when (29) and (28) are equal. After canceling terms this gives $2^p \alpha \approx (Z_{SG} / Z_C)(\alpha + \gamma)$. Table 1 gives values of N_ξ and $|\Theta|$ for various values of M and p. One can observe that $2^p N_\xi \approx |\Theta|$ is a slight overestimation, but it improves as M grows larger. For reference, the number of points used by a full tensor product grid is also shown.

In the remainder of this paper, we explore the model and assess the validity of assumptions. In particular, we compare the accuracy of a level-p Smolyak grid with a degree-p polynomial approximation in the Galerkin approach. We also investigate the cost of matrix–vector products and the convergence behavior of mean-based preconditioning.

5. EXPERIMENTAL RESULTS AND MODEL VALIDATION

In this section we present the results of numerical experiments with both discretization methods, with the aims of comparing their accuracy and solution costs and validating the model developed in the previous section. First, we investigate a problem with a known solution to verify that both methods are converging to the correct solution and to examine the convergence of the PCG iteration. Second, we examine two problems where the diffusion coefficient is defined using a known covariance function, and we measure the computational effort required by each method.

TABLE 1: Degrees of freedom for various methods.

$M = 2$	Level p sparse grid (Gaussian) $\|\Theta\|$	Galerkin N_ξ	Non-zero blocks per row in Galerkin matrix	Tensor grid
$p = 1$	5	3	2.33	4
$p = 2$	13	6	3.00	9
$p = 3$	29	10	3.40	16
$p = 4$	53	15	3.67	25
$M = 10$				
$p = 1$	21	11	2.82	1024
$p = 2$	221	66	4.33	59,049
$p = 3$	1581	286	5.62	1,048,576
$p = 4$	8761	1001	6.71	9,765,625
$M = 20$				
$p = 1$	41	21	2.90	1.04×10^6
$p = 2$	841	231	4.64	3.49×10^9
$p = 3$	11,561	1771	6.22	1.10×10^{12}

5.1 Behavior of the Preconditioned Conjugate Gradient Algorithm

For well-posed Poisson problems, PCG with a multigrid preconditioner converges rapidly. Since collocation entails the solution of multiple deterministic systems, we expect multigrid to behave well. For Galerkin systems, the performance of mean-based preconditioning is more complicated. To understand this we investigate the problem

$$-\nabla \cdot [a(\mathbf{x}, \xi) u(\mathbf{x}, \xi)] = f(\mathbf{x}, \xi) \tag{31}$$

in the domain $[-0.5, 0.5]^2$ with zero Dirichlet boundary conditions, where the diffusion coefficient given as a one-term KL expansion,

$$a(\mathbf{x}, \xi) = 1 + \sigma \frac{1}{\pi^2} \xi \cos\left[\frac{\pi}{2}(x^2 + y^2)\right]. \tag{32}$$

We choose the function

$$u = \exp(-|\xi|^2) 16(x^2 - 0.25)(y^2 - 0.25) \tag{33}$$

as the exact solution, and the forcing term f is defined by applying (31) to u.

The diffusion coefficient must remain positive for the problem to remain well-posed. This is the case provided

$$\left| \sigma \frac{1}{\pi^2} \xi \cos\left(\frac{\pi}{2} r^2\right) \right| < 1, \tag{34}$$

which holds when $|\xi| < (\pi^2)/(\sigma)$. As a consequence of this, well-posedness cannot be guaranteed when ξ is unbounded. There are various ways this can be addressed. We assume here that the random variable in (32) has a *truncated Gaussian density*,

$$\rho(\xi) = \frac{1}{\int_{-c}^{c} \exp(-\frac{\xi^2}{2})\, d\xi} \exp\left(-\frac{\xi^2}{2}\right) \mathbf{1}_{[-c,c]}, \tag{35}$$

which corresponds to taking the diffusion coefficient from a screened sample where the screening value c is chosen to enforce the conditions (1.7) for ellipticity and boundedness. The cutoff parameter c is chosen to be equal to 2.575. For

this cutoff the area under a standard normal distribution between $\pm c$ is equal to 0.99. For this value of c, $|\xi| < 2.575$ and the problem is guaranteed to remain well posed provided that $\sigma < (\pi^2)/[\max(|\xi|)] = 3.8329$.

Polynomials orthogonal to a truncated Gaussian measure are referred to as Rys polynomials [20]. As the parameter c is increased, the measure approaches the standard Gaussian measure and the Rys polynomials are observed to approach the behavior of the Hermite polynomials. For our implementation of collocation, the sparse grids are based on the zeros of the Rys polynomials for the measure determined by (35). This leads to an efficient multidimensional quadrature rule using the Gaussian weights and abscissas.

The recurrence coefficients for orthogonal polynomials can be expressed explicitly as

$$\alpha_i = \frac{\int_\Gamma \xi \psi_i(\xi)^2 \rho(\xi)\, d\xi}{\int_\Gamma \psi_i(\xi)^2 \rho(\xi)\, d\xi}, \quad \beta_i = \frac{\int_\Gamma \psi_i(\xi)^2 \rho(\xi)\, d\xi}{\int_\Gamma \psi_{i-1}(\xi)^2 \rho(\xi)\, d\xi}. \tag{36}$$

In the case of Hermite polynomials there exist closed forms for the recurrence coefficients. No such closed form is known in general for the Rys polynomials so a numerical method must be employed. The generation of orthogonal polynomials by numerical methods is discussed extensively in [20] and the use of generalized polynomial chaos bases in the stochastic Galerkin method is discussed in [7]. We compute the coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ via the discretized Steltjies procedure [21] where integrals in (36) are approximated by quadrature.

Testing for both the sparse grid collocation method and the stochastic Galerkin method was performed using the truncated Gaussian PDF and Rys polynomials for several values of σ. The linear solver in all cases was stopped when $(||r_k||_2)/(||b||_2) < 10^{-12}$, where $r_k = b - Ax_k$ is the linear residual and A and b are the coefficient matrix and right-hand side, respectively. We constructed the sparse grids using the *Dakota* software package [22].

Table 2 reports $||\langle e_p \rangle||_{l_\infty}$, the discrete l_∞-norm of the mean error $\langle e_p \rangle$ evaluated on the grid points. For problems in one random variable, the stochastic collocation and stochastic Galerkin methods produce identical results. Table 3 shows the average number of iterations required by each deterministic subproblem as a function of grid level and σ.

TABLE 2: Mean error in the discrete l_∞ norm for the stochastic collocation and stochastic Galerkin methods.

Level/p	σ				
	1	2	3	4	5
1	0.1856	0.1971	0.2175	0.2466	0.2807
2	0.0737	0.0811	0.0932	0.1095	0.1207
3	0.0245	0.0279	0.0331	0.0389	0.1195
4	0.0070	0.0082	0.0099	0.0121	DNC
5	0.0017	0.0021	0.0026	0.0029	DNC
6	3.7199e-4	4.6301e-4	5.7900e-4	6.7702e-4	DNC
7	7.2002e-5	9.1970e-5	1.1605e-4	4.1598e-4	DNC

TABLE 3: Iterations for the stochastic collocation (left) and stochastic Galerkin methods (right).

Level	σ					p	σ				
	1	2	3	4	5		1	2	3	4	5
1	10	10	10.5	11	11	1	13	15	16	18	21
2	10	10.33	10.67	11.33	12.67	2	13	17	22	28	38
3	10	10.5	11	12.25	22	3	14	19	26	39	140
4	10	10.6	11.2	13	DNC	4	14	20	29	53	DNC
5	10.17	10.5	11.33	13.83	DNC	5	14	21	31	69	DNC
6	10.14	10.43	11.43	15	DNC	6	15	21	33	94	DNC
7	10.13	10.63	11.38	16.75	DNC	7	15	21	34	136	DNC

Problems to the right of the double line do not satisfy (34), and some of the associated systems will be indefinite for a high enough grid level, as some of the collocation points will be placed in the region of ill-posedness. If the solver failed to converge for any of the individual subproblems, the method is reported as having failed using "DNC".

Table 3 shows the PCG iteration counts for both methods. Again, problems to the right of the double line are ill-posed, and the Galerkin linear system as well as a subset of the individual collocation systems are guaranteed to become indefinite as the degree of polynomial approximation p (for stochastic Galerkin) or sparse grid level (for collocation) increases [14]. Table 3 shows that the iteration counts are fairly well behaved when mean-based preconditioning is used. In general, iterations grow as the degree of polynomial approximation increases.

It is well known that bounds on convergence of the conjugate gradient method are determined by the condition number of the matrix. It is shown in [14] that if the diffusion coefficient is given by a stationary field, as in (32), then the eigenvalues of the preconditioned stochastic Galerkin system lie in the interval $[1 - \tau, 1 + \tau]$, where

$$\tau = C_{p+1}^{\max} \frac{\sigma}{\mu} \left(\sum_{k=1}^{M} \sqrt{\lambda_k} ||a_k(\mathbf{x})||_{L_\infty} \right), \tag{37}$$

and $C_{p+1}^{\max}$ is the magnitude of the largest zero of the degree $p + 1$ orthogonal polynomial. Therefore the condition number is bounded by $\kappa(A) \leq [1 + \tau]/[1 - \tau]$. It is possible to bound the eigenvalues of a single system arising in collocation in a similar manner using the relation (27). The eigenvalues of the system arising from sampling (27) at ξ, lie in the bounded interval $[1 - \tilde{\tau}(\xi), 1 + \tilde{\tau}(\xi)]$ where

$$\tilde{\tau}(\xi) = \frac{\sigma}{\mu} \left(\sum_{k=1}^{M} \sqrt{\lambda_k} ||a_l(\mathbf{x})||_{L_\infty} |\xi_k| \right). \tag{38}$$

Likewise, the condition number for a given preconditioned collocation system can be bounded by $\kappa[A(\xi)] \leq (1 + \tilde{\tau})/(1 - \tilde{\tau})$. For both methods, as σ increases relative to μ the associated systems may become ill-conditioned and will eventually become indefinite. Likewise, as p or the sparse grid level increases, $C_{p+1}^{\max}$ and $\max_{\Theta_{p,M}} |\xi|$ increase and the problems may again become indefinite. However, if Γ is bounded then both $C_{p+1}^{\max}$ and $\max_{\Theta_{p,M}} |\xi|$ are bounded for all choices of p and the sparse grid level and the systems are guaranteed to remain positive definite, provided σ is not too large.

The effect of these bounds can be seen in the above examples since as σ increases the iteration counts for both methods increase until finally for large choices of σ and large p or grid level the PCG iteration fails to converge. However, for smaller values of σ the PCG iteration converges in a reasonable number of iterations for all tested values of p and grid level.

5.2 Computational Cost Comparison

In this section we compare the performance of the two methods using both the model developed above and the implementations in *Trilinos*. For our numerical examples, we consider a problem where only the covariance of the diffusion field is given. We consider two problems of the form

$$-\nabla \cdot \left\{ \left[\mu + \sigma \sum_{k=1}^{M} \sqrt{\lambda_k} \xi_k f_k(x) \right] \nabla u \right\} = 1, \tag{39}$$

where values of M between 3 and 15 are explored and $\{\lambda_k, f_k\}$ are the eigenpairs associated with the covariance kernel

$$C(\mathbf{x}_1, \mathbf{x}_2) = \exp(-|x_1 - x_2| - |y_1 - y_2|). \tag{40}$$

The KL expansion of this kernel is investigated extensively in [4]. For the first problem, the random variables $\{\xi_k\}$ are chosen to be identically independently distributed uniform random variables on $[-1, 1]$. For the second problem, the

random variables $\{\xi_k\}$ are chosen to be identically independently distributed truncated Gaussian random variables as in the previous section. For the first, problem $\mu = 0.2$ and $\sigma = 0.1$. For the second problem, $\mu = 1$ and $\sigma = 0.25$. These parameters were chosen to ensure that the problem remains well posed. Table 4 shows approximate values for τ for both of the above problems. In the second case, where truncated Gaussian random variables are used, $1 - \tau$ becomes close to zero as the stochastic dimension of the problem increases. Thus this problem could be said to be nearly ill-posed. In terms of computational effort this should favor the sparse grid collocation method, since, as was seen in the previous section, iteration counts for the stochastic Galerkin method increased faster than those for the collocation method as the problem approaches ill-posedness. The spatial domain is discretized by a uniform mesh with discretization parameter $h = 1/32$. Note that the mean-based preconditioning eliminates the dependence on h of the conditioning of the problem [14], so we consider just a single value of the spatial mesh parameter.

Approximate solutions are used to measure the error since there is no analytic expression for the exact solution to either of the above problems. To measure the error for the Galerkin method the exact solution is approximated by a high order ($p = 10$) Galerkin scheme. For the collocation method we take the solution from a level-10 sparse grid approximation as an approximation to the exact solution. These two differed by an amount on the order of the machine precision. The error in the stochastic space is then estimated by computing the mean and variance of the approximate solutions and comparing it to the mean and variance of the order-10 (level-10) approximations. The linear solves for both methods stop when $(\|r_k\|_2)/(\|b\|_2) < 10^{-12}$. In measuring the time, setup costs are ignored. The times reported are nondimensionalized by the time required to perform a single deterministic matrix vector product and compared with the model developed above.

Figure 1 explores the accuracy obtained for the two discretizations for $M = 4$; the behavior was the same for $M = 3$ and $M = 5$. In particular, it can be seen that for both sample problems, the same value of p (corresponding to the polynomial space for the Galerkin method and the sparse grid level for the collocation method) in the two methods

TABLE 4: Approximate values of τ for model problems.

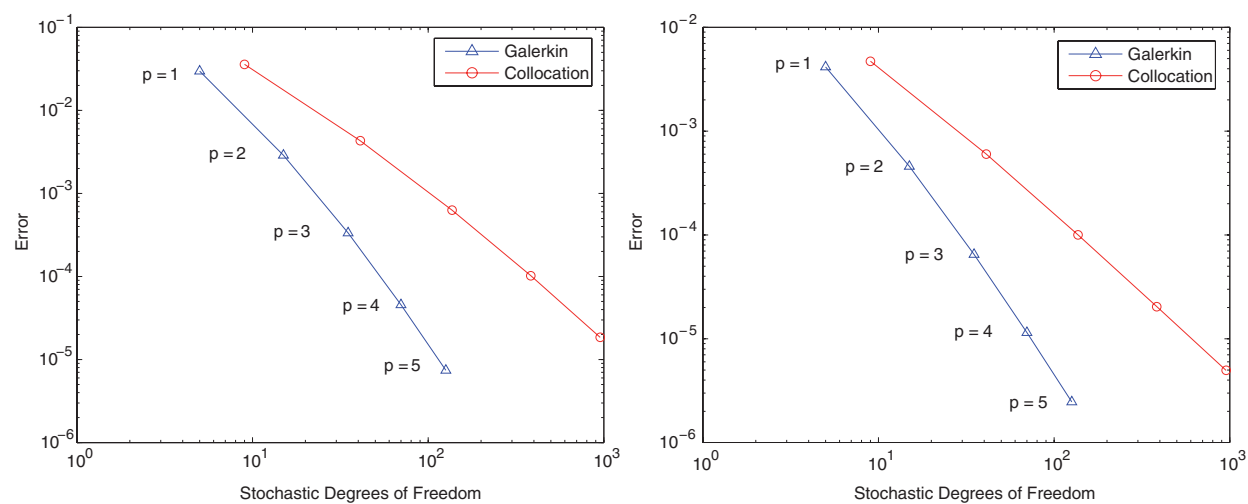| M	Uniform random variables $\Gamma_i = [-1, 1]$, $\sigma = 0.1$, $\mu = 0.2$	Truncated Gaussian random variables $\Gamma_i = [-2.576, 2.576]$, $\sigma = 0.25$, $\mu = 1$
3	0.533	0.686
4	0.549	0.708
5	0.566	0.729

FIG. 1: Errors vs stochastic DOF for $M = 4$. Uniform random variables (left), and truncated Gaussian random variables (right).

produce solutions of comparable accuracy. Thus, the Galerkin method gives higher accuracy per stochastic degree of freedom. Since the unknowns in the Galerkin scheme are coupled, the cost per degree of freedom will be higher. In terms of computational effort, the question is whether or not the additional accuracy per degree of freedom will be worth the additional cost.

Figures 2 and 3 compare the costs incurred by the two methods, measured in CPU time, for obtaining solutions of comparable accuracy. The timings reflect time spent to execute the methods on an Intel Core 2 Duo machine running at 3.66 GHz with 6 Gb of RAM. In the figures these timings are nondimensionalized by dividing by the cost of a single sparse matrix–vector product with the (five-diagonal) nonzero structure of $\{A_k\}$. This cost is measured by dividing the total time used by the collocation method for matrix–vector products by the total number of CG iterations performed in the collocation method. This allows the times to be compared to the cost model (28) and (29), which in turn helps ensure that the implementations are of comparable efficiency. The model is somewhat less accurate for the collocation method, because for these relatively low-dimensional models the approximation $|\Theta_{p,M}| = 2^p N_\xi$ is an overestimate. For the values of M used for these results ($M = 3, 4$, and 5), it can be seen that the Galerkin method requires less CPU time than the collocation method to compute solutions of comparable accuracy, and that the gap widens as the
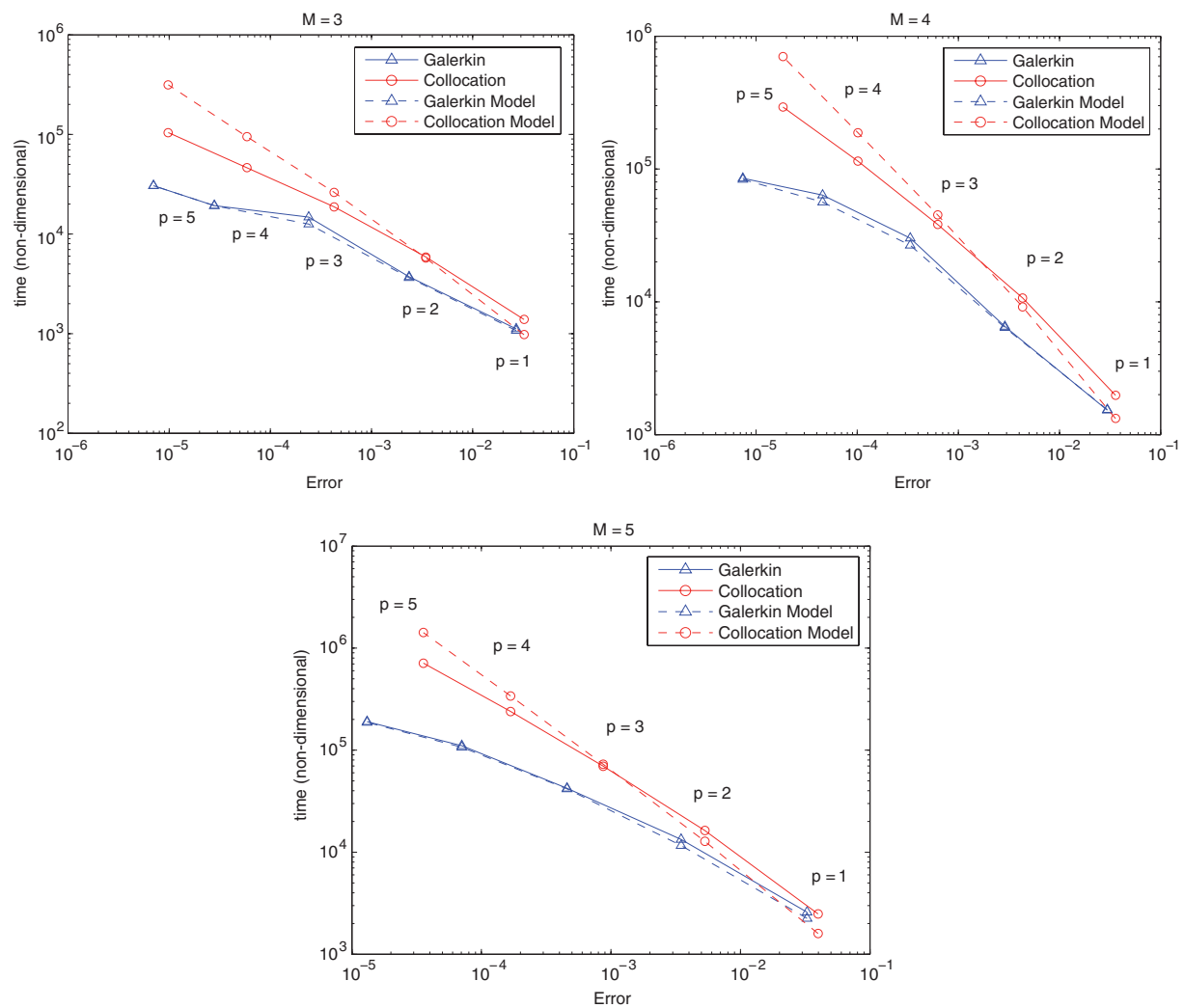

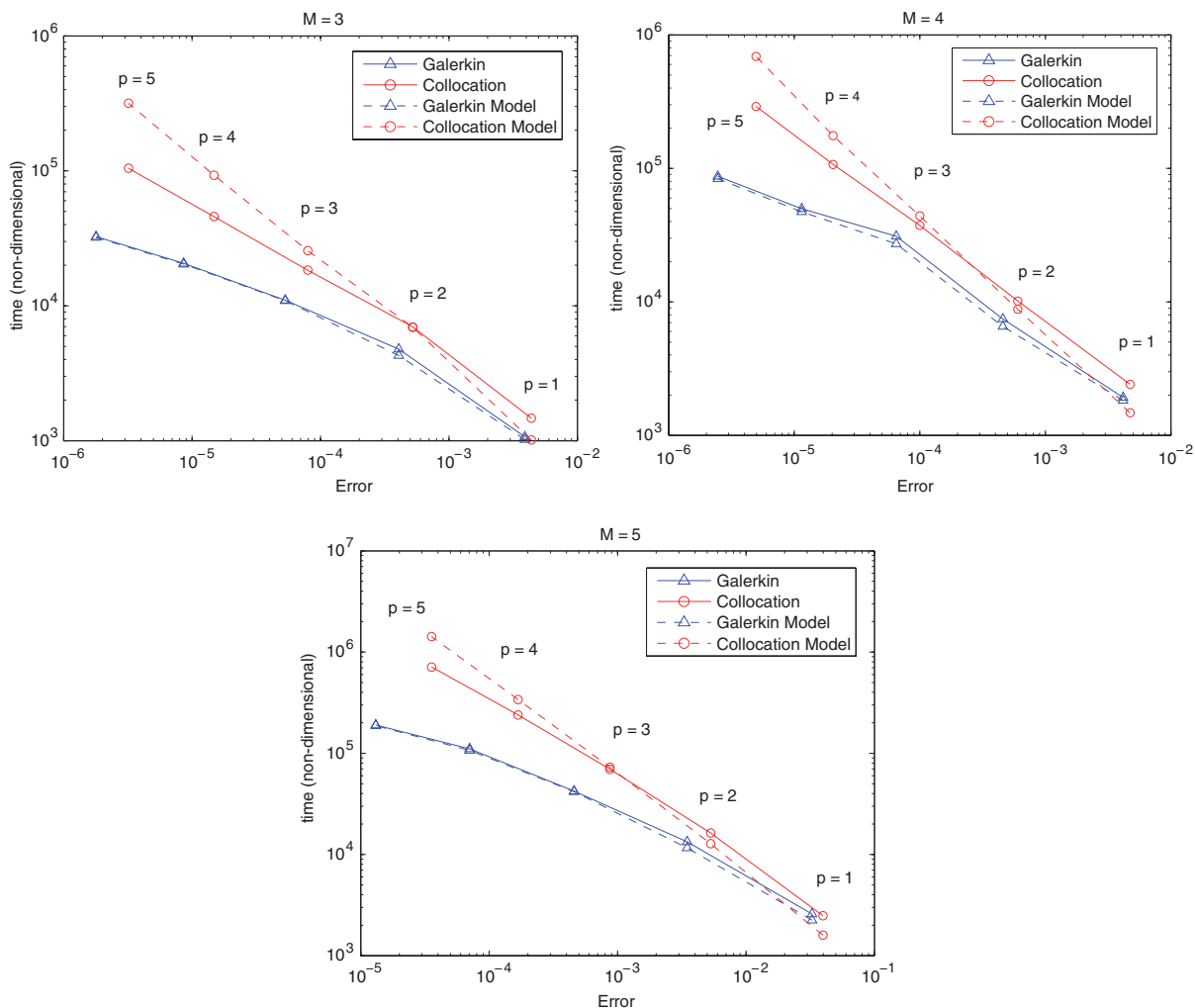**FIG. 2:** Solution time vs error for $M = 3, 4, 5$. Uniform random variables.

FIG. 3: Solution time vs error for $M = 3, 4, 5$. Truncated Gaussian random variables.

dimension of the space of random variables increases. Also, it is seen in Figs. 2 and 3 that the performance of each method is largely independent of the density functions used in defining the random variables ξ_k.

Table 5 expands on these results for larger values of M, based on our expectation that the same value of p (again, corresponding to the polynomial space for the Galerkin method or the level for the collocation method) yields solutions of comparable accuracy. The trends are comparable for all M and show that as the size of the approximation space increases, the overhead for collocation associated with the increased number of degrees of freedom becomes more significant.

6. CONCLUSION

In this study we have examined the costs of solving the linear systems of equations arising when either the stochastic Galerkin method or the stochastic collocation method is used to discretize the diffusion equation in which the diffusion coefficient is a random field modeled by (2). The results indicate that when mean-based preconditioners are coupled with the conjugate gradient method to solve the systems that arise, the stochastic Galerkin method is quite competitive with collocation. Indeed, the costs of the Galerkin method are typically lower than for collocation, and this differential

TABLE 5: Solution (preconditioning) time in seconds for second model problem.

	Stochastic Galerkin			Sparse grid collocation		
	$M = 5$	$M = 10$	$M = 15$	$M = 5$	$M = 10$	$M = 15$
Level/$p = 1$	0.058139	0.147306	0.320443	0.068934	0.163258	0.285779
	(0.026912)	(0.051521)	(0.085775)	(0.036288)	(0.078107)	(0.123893)
2	0.269301	1.20465	3.80461	0.532407	2.13126	5.07825
	(0.119066)	(0.0385744)	(1.04111)	(0.275829)	(0.98289)	(2.1247)
3	1.20353	13.1382	51.448	2.41468	16.9871	57.9837
	(0.372013)	(2.57246)	(7.40171)	(1.20969)	(7.54744)	(23.1414)
4	3.50061	53.786	168.112	8.31068	102.595	493.042
	(1.1846)	(10.1633)	(41.325)	(4.14521)	(44.0484)	(193.199)
5	6.510255	117.729		24.5645	515.751	
	(2.89493)	(36.2012)		(12.0362)	(221.546)	

becomes more pronounced as the number of terms in the truncated KL expansion increases. We have also developed a cost model for both methods that closely mirrors the complexity of the algorithms.

REFERENCES

1. Loève, M., *Probability Theory*, Springer-Verlag, New York, 1978.

2. Weiner, N., The homogeneous chaos, *Am. J. Math.*, 60(4):897–936, 1938.

3. Babuška, I., Tempone, R., and Zouraris, G., Galerkin finite element approximations of stochastic elliptic partial differential equations, *SIAM J. Numer. Anal.*, 42:800–825, 2004.

4. Ghanem, R. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

5. Sarkar, A. and Ghanem, R., Mid-frequency structural dynamics with parameter uncertainty, *Comput. Methods Appl. Mech. Eng.*, 191:5499–5513, 2002.

6. Xiu, D. and Shen, J., Efficient stochastic Galerkin methods for random diffusion equations, *J. Comput. Phys.*, 228:266–281, 2009.

7. Xiu, D. and Karniadakis, G., Modeling uncertainty of elliptic partial differential equations via generalized polynomial chaos, *Comput. Methods Appl. Mech. Eng.*, 191:4928–4948, 2002.

8. Babuška, I., Nobile, F., and Tempone, R., A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM J. Numer. Anal.*, 45:1005–1034, 2007.

9. Nobile, F., Tempone, R., and Webster, C. G., A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

10. Xiu, D. and Hesthaven, J., High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.

11. Bäck, J., Nobile, F., Tamellini, L., and Tempone, R., Stochastic Galerkin and Collocation Methods for PDEs with Random Coefficients: A Numerical Comparison, Tech. Rep. 09-33, Institute for Computational Engineering and Sciences, Univer-

sity of Texas at Austin, 2009. To appear in *Proceedings of ICOSAHOM'09*, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2009.

12. Heroux, M., Bartlett, R., Hoekstra, V. H. R., Hu, J., Kolda, T., Lehoucq, R., Long, K., Pawlowski, R., Phipps, E., Salinger, A., Thornquist, H., Tuminaro, R., Willenbring, J., and Williams, A., An Overview of Trilinos, Tech. Rep. SAND2003-2927, Sandia National Laboratories, 2003.

13. Ghanem, R. and Pellissetti, M., Iterative solution of systems of linear equations arising in the context of stochastic finite elements, *Adv. Eng. Software*, 31(8):607–616, 2000.

14. Powell, C. and Elman, H., Block-diagonal preconditioning for spectral stochastic finite element systems, *IMA J. Numer. Anal.*, 29:350–375, 2009.

15. Gee, M., Siefert, C., Hu, J., Tuminaro, R., and Sala, M., ML 5.0 Smoothed Aggregation User's Guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.

16. Gordon, A. and Powell, C., Solving Stochastic Collocation Systems with an Algebraic Multigrid, MIMS EPrint 2010.19, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, February 2010.

17. Nobile, F., Tempone, R., and Webster, C. G., An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

18. Ma, X. and Zabaras, N., An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, *J. Comput. Phys.*, 228:3084–3113, 2009.

19. Ma, X. and Zabras, N., High-dimensional stochastic model representation technique for the solution of stochastic PDEs, *J. Comput. Phys.*, 229(10):3884–3915, 2010.

20. Gautschi, W., *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, Oxford, 2004.

21. Sagar, R. and Smith, V., On the calculation of Rys polynomials and quadratures, *Int. J. Quant. Chem.*, 43:827–836, 1992.

22. Eldred, M., Giunta, A., van Bloemen Waanders, B., Wojtkiewicz, S., Hart, W., and Alleva, M., Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, Version 4.0 user's manual, Tech. Rep. SAND2006-6337, Sandia National Laboratories, October 2006.