ASSESSMENT OF COLLOCATION AND GALERKIN APPROACHES TO PARTIAL DIFFERENTIAL EQUATIONS WITH RANDOM DATA

CHRISTOPHER W. MILLER[*], RAYMOND S. TUMINARO[†], ERIC T. PHIPPS[‡], AND HOWARD C. ELMAN[§]

Abstract. 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 number of systems that need to be solved for the stochastic collocation method can be several times larger than the number of unknowns in the stochastic Galerkin system.

We implement both of the above methods using the Trilinos software package [7] 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 that 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 a open subset of $\mathbb{R}^n$ and (Ω, Σ, P) is a complete probability space, then this can be written as

$$\begin{aligned} -\nabla \cdot (a(\mathbf{x},\omega)\nabla u(\mathbf{x},\omega)) &= f(\mathbf{x},\omega) \qquad && (\mathbf{x},\omega) \in D \times \Omega \\ u(x,\omega) &= 0 && (\mathbf{x},\omega) \in \partial D \times \Omega. \end{aligned} \tag{1.1}$$

The random input field is often given as a truncated Karhunen-Loève (KL) expansion [8]. The truncated KL-expansion is given by

$$a(\mathbf{x},\omega) \approx a(\mathbf{x},\boldsymbol{\xi}(\omega)) = a_0(\mathbf{x}) + \sum_{k=1}^{M} \sqrt{\lambda_k}\xi_k(\omega)a_k(\mathbf{x}), \tag{1.2}$$

where (λ_i, a_i) are eigenvalues and eigenfunctions of the covariance operator

$$(C^*(\alpha))(x_1) = \int_D C(x_1,x_2)\alpha(x_2)\,dx_2, \tag{1.3}$$

and C is the covariance kernel of the random field. The random variables are uncorrelated, mean zero, and are given by

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}}\int_D a(\mathbf{x},\omega)a_k(\mathbf{x})\,d\mathbf{x}. \tag{1.4}$$

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 semi-definite 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 (1.1) it is necessary

[*] University of Maryland at College Park: Department of Applied Mathematics and Scientific Computation, cmiller@math.umd.edu

[†] Sandia National Laboratories, rstumin@sandia.gov

[‡] Sandia National Laboratories, etphipp@sandia.gov

[§] University of Maryland at College Park: Department of Applied Mathematics and Scientific Computation, elman@cs.umd.edu to assume that the diffusion is uniformly bounded away from zero, we assume that there exist constants a_{min} and a_{max} such that

$$0 < a_{min} \leq a(\mathbf{x}, \boldsymbol{\xi}) \leq a_{max} < \infty, \tag{1.5}$$

almost everywhere, P-almost surely and $f \in L_2(\Omega) \otimes L_2(D)$.

The goal of this paper is to model the computational costs and compare the performance of the stochastic Galerkin method [2], [6] and the sparse grid collocation method [1], [9], [12] for the solution of (1.1). 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. 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 $L_2(\Gamma) = \{v(\boldsymbol{\xi}) : \langle v^2 \rangle < \infty\}$. The norm over $L_2(\Gamma)$ is induced by the inner product

$$\langle u, v \rangle = \int_{\Gamma} u(\boldsymbol{\xi}) v(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) \, d\boldsymbol{\xi} = \int_{\Omega} u(\boldsymbol{\xi}(\omega)) v(\boldsymbol{\xi}(\omega)) \, dP. \tag{2.1}$$

We can define a variational form of (1.1) in the stochastic domain by: for all $\mathbf{x} \in D$, find $u(\mathbf{x}, \boldsymbol{\xi}) \in L_2(\Gamma)$ such that

$$-\langle \nabla \cdot (a \nabla u), v \rangle = \langle f, v \rangle \tag{2.2}$$

for all $v \in L_2(\Gamma)$. This leads to a set of coupled second-order linear partial differential equations in the spatial dimension. The stochastic Galerkin method combines a Galerkin projection in the stochastic dimension with a standard discretization in the spatial dimension. Details are as follows.

Define S_p to be the space of multivariate polynomials in $\boldsymbol{\xi}$ of total degree p. This space has dimension $N_\xi = \frac{(M+p)!}{M!p!}$. Let $\{\Psi_k\}_{k=0}^{N_\xi - 1}$ be a basis for S_p orthogonal with respect to the inner product (2.1). Substituting KL-expansions for $a(\mathbf{x}, \omega)$ and $f(\mathbf{x}, \omega)$ and restricting (2.2) to $v \in S_p$ gives

$$-\langle \nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}) \sum_{i=1}^{N_\xi} \nabla u_i(\mathbf{x}) \Psi_i(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \rangle = \langle f(\mathbf{x}, \boldsymbol{\xi}), \Psi_j \rangle \ \forall j = 0 : N_\xi - 1. \tag{2.3}$$

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 $A\vec{u} = \vec{f}$. The coefficient matrix and right-hand side have the tensor product structure

$$A = \sum_{k=0}^{M} G_k \otimes A_k, \quad \vec{f} = \sum_{k=0}^{M} \vec{g_k} \otimes \vec{f_k}. \tag{2.4}$$

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

$$\begin{aligned} G_0(i,j) &= \langle \Psi_i \Psi_j \rangle, \ g_0(i) = \langle \Psi_i \rangle = c\delta_{i0} \\ G_k(i,j) &= \langle \xi_k \Psi_i \Psi_j \rangle, \ g_k(i) = \langle \xi_k \Psi_i \rangle. \end{aligned} \tag{2.5}$$

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. A_k is symmetric for all k and A_0 is positive definite by (1.5). The assumption (1.5) ensures that A 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. 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

$$\langle \psi_i(\xi_k), \psi_j(\xi_k)\rangle_k = \int_{\Gamma_k} \psi_i(\xi_k)\psi_j(\xi_k)\rho_k(\xi_k)\, d\xi_k = \delta_{ij}. \tag{2.6}$$

These polynomials can be expressed via the familiar three-term recurrence

$$\psi_{i+1}(\xi_k) = (\xi_k - \alpha_i)\psi_i(\xi_k) - \beta_i \psi_{i-1}(\xi_k), \tag{2.7}$$

where $\psi_0 = 1,\ \psi_{-1} = 0$. Therefore, the entries in G_0 are

$$G_0(i,j) = \langle \Psi_i, \Psi_j\rangle = \prod_{k=1}^{M} \langle \psi_{i_k}(\xi_k), \psi_{j_k}(\xi_k)\rangle_k = \prod_{k=1}^{M} \delta_{i_k j_k} = \delta_{ij}, \tag{2.8}$$

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

$$G_k(i,j) = (\langle \psi_{i_k+1}, \psi_{j_k}\rangle_k + \alpha_{i_k}\langle \psi_{i_k}, \psi_{j_k}\rangle_k + \beta_{i_k}\langle \psi_{i_k-1}, \psi_j\rangle_k) \prod_{l=1, l\neq k}^{M} \langle \psi_{i_l}, \psi_{j_l}\rangle_l. \tag{2.9}$$

Thus G_0 is the identity matrix and G_k has at most three entries per row for $k > 0$.

The stochastic Galerkin method requires the solution to the large linear system (2.4). Since the Galerkin matrix is symmetric and positive-definite, the system can be solved by the conjugate gradient (CG) method. CG only requires the evaluation of matrix-vector products, so that it is not necessary to store the assembled matrix A. The matrix-vector products can be performed implicitly following a procedure described in [5]. Each matrix A_k is assembled and the matrix-vector product is expressed as $(Au)_j = \sum_{i=0}^{N_\xi - 1}\sum_{k=0}^{M} \langle \xi_k \Psi_i \Psi_j\rangle (A_k u_i)$. 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 dominated by the computation of $A_k u_i$, which entails $(M+1)N_\xi$ sparse matrix-vector products by matrices $\{A_k\}$ of order N_x. This approach 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 $\{G_k\}$. It is shown in [5] that this approach makes optimal use of storage and memory bandwidth. Once the solution to (2.4) is obtained, statistical quantities such as moments or a probability distribution associated with the solution process can be obtained cheaply [6].

To obtain fast convergence, we will also use a preconditioner. In particular, it has been shown in [10] that an effective choice is an approximation to $A_0^{-1} \otimes G_0^{-1}$, where A_0 is the mean stiffness matrix. Since G_0 is the identity, the preconditioner application entails the approximate action of N_ξ uncoupled copies of A_0^{-1}. For this, we will use a single iteration of an algebraic multigrid solver provided by [4].

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

$\Theta = \{\boldsymbol{\xi}^{(1)}, ..., \boldsymbol{\xi}^{(n)}\}$ and constructs a high-order polynomial approximation to the solution function using the solutions to the deterministic PDEs

$$-\nabla \cdot (a(\mathbf{x}, \boldsymbol{\xi}^{(l)}) \nabla u(\mathbf{x}, \boldsymbol{\xi}^{(l)}) = f(\mathbf{x}, \boldsymbol{\xi}^{(l)}), \tag{3.1}$$

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, as for the stochastic Galerkin method [12].

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 $\theta_i = \{\xi : \psi_{i+1}(\xi) = 0\} = \{\xi_{i,k}^{(j)}\}_{j=1}^{i+1}$ for $i = 0, 1, ...,$ and $j = 1, 2, ..., i + 1$. A one-dimensional, $(i + 1)$-point interpolation operator is given by

$$U^i(u)(\xi) = \sum_{j=1}^{i+1} u(\xi_i^{(j)}) l_i^{(j)}(\xi), \tag{3.2}$$

where $l_i^{(j)}(\xi)$ are the standard Lagrange interpolating polynomials. 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

$$U^{i_1} \otimes \cdots \otimes U^{i_M}(u)(\boldsymbol{\xi}) = \sum_{j_1=1}^{i_1+1} \cdots \sum_{j_M=1}^{i_M+1} u(\xi_{i_1}^{(j_1)}, \cdots, \xi_{i_M}^{(j_M)})(l_{i_1}^{(j_1)} \otimes \cdots l_{i_M}^{(j_M)}). \tag{3.3}$$

The evaluation of this operator requires the solution of a collection of deterministic PDEs (3.1), one for each sample point in $\Theta_{tensor} = \times_{j=1}^{M} \theta_{i_j}$. This method suffers from the so-called curse of dimensionality since the number of sample points $|\Theta_{tensor}| = \prod_{j=1}^{M} |\theta_{i_j}| = \prod_{j=1}^{M} (i_j + 1)$ grows exponentially with the dimension of the problem. This cost can be significantly reduced using sparse grid methods [12].

Sparse grid collocation methods are based on the Smolyak approximation formula. The Smolyak operator $A(p, M)(u)$ is a linear combination of the product formulas in (3.3). Let $Y(p, M) = \{\boldsymbol{i} \in \mathbb{N}^N : p - M + 1 \leq |\mathbf{i}|_1 \leq p\}$. Then the Smolyak formula is given by

$$A(p, M)(u) = \sum_{\mathbf{i} \in Y(p,M)} (-1)^{p+M-|\mathbf{i}|_1} \binom{M-1}{p+M-|\mathbf{i}|_1} (U^{i_1} \otimes \cdots \otimes U^{i_M}). \tag{3.4}$$

The evaluation of the Smolyak formula requires the solution of deterministic PDEs (3.1) for $\boldsymbol{\xi}^{(l)}$ in the set of points

$$\Theta_{p,M} = \bigcup_{\mathbf{i} \in Y(p,M)} (\theta_{i_1} \times \cdots \times \theta_{i_M}). \tag{3.5}$$

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

It is shown in [9] that sampling the differential operator on the sparse grid $\Theta_{p,M}$ will produce $A(p, M)(u) = u_p$ where u_p is an approximate solution to (1.1) 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 $M \gg 1$ [12].

The repeated assembly of deterministic stiffness matrices 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 linear combination of the stiffness matrices $\{A_k\}$ appearing in (2.4). For a given value of $\boldsymbol{\xi}$ the deterministic stiffness matrix can be expressed as

$$A(\boldsymbol{\xi}) = A_0 + \sum_{k=1}^{M} \xi_i A_k. \tag{3.6}$$

In our implementation we assemble the matrices $\{A_k\}$ first and then compute the scaled sum (3.6) at each collocation point. This 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 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.

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. 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.

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 (2.4), where α and γ are constants. The parameter γ can be thought of as the number of order-N_x matrix-vector products required per stochastic degree of freedom in the stochastic Galerkin method. If the matrix-vector product is performed as in [5], then γ is equal to $M + 1$. The total cost of the Galerkin method can be modeled by

$$\textit{Galerkin cost} = N_\xi Z_G(\alpha + \gamma). \tag{4.1}$$

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

$$\textit{collocation cost} = Z_C 2^p N_\xi(\alpha + 1), \tag{4.2}$$

M = 2	Level k Sparse Grid	Galerkin	Non-Zero Blocks	Tensor Grid
	$\|\Theta\|$	N_ξ	per row in SFEM	
k = 1	5	3	2.33	4
k = 2	13	6	3.00	9
k = 3	29	10	3.40	16
M = 10				
k = 1	21	11	2.82	1024
k = 2	221	66	4.33	59049
k = 3	1581	286	5.62	1048576
M = 20				
k = 1	41	21	2.90	1.04×10^6
k = 2	841	231	4.64	3.49×10^9
k = 3	11561	1771	6.22	1.0995×10^{12}.

Table 4.1

Degrees of Freedom for Various Methods

where p is the Smolyak grid level, N_ξ is the number of degrees of freedom needed by an order p Galerkin system, and Z_C is the number of PCG iterations needed to solve a single deterministic system.

The relative costs of the two methods depend on the parameter values. If, for example, the ratio of iteration counts (Z_{SG}/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 significantly 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. Table 4.1 gives values of N_ξ and $|\Theta|$ for various values of M and p. One can observe that the estimate $2^p N_\xi \approx |\Theta|$ is a slight overestimate but improves as M grows larger. For reference, the number of points used by a full tensor product grid as well as the number of non-zero blocks in the Galerkin method are also shown.

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. Next, we examine two problems where the diffusion is defined using a known covariance function and measure the computational effort required by each method.

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

$$-\nabla \cdot (a(\mathbf{x}, \xi) u(\mathbf{x}, \xi)) = f \tag{5.1}$$

in the domain $[-.5, .5]^2$ with zero Dirichlet boundary conditions, where the diffusion coefficient given in a one-term KL expansion,

$$a(\mathbf{x}, \xi) = 1 + \sigma \frac{1}{\pi^2} \xi \cos\left(\frac{\pi}{2}(x^2 + y^2)\right). \tag{5.2}$$

We choose the function

$$u = exp(-|\xi|^2)16(x^2 - .25)(y^2 - .25) \tag{5.3}$$

as the exact solution.

The assumption (1.5) is valid provided

$$\left|\sigma \frac{1}{\pi^2} \xi \cos\left(\frac{\pi}{2} r^2\right)\right| < 1 \Rightarrow |\xi| < \frac{\pi^2}{\sigma}. \tag{5.4}$$

We assume here that the random variable in (5.2) has a truncated Gaussian density

$$\rho(\xi) = \frac{1}{\int_{-c}^{c} exp(-\frac{\xi^2}{2})\, d\xi} exp\left(-\frac{\xi^2}{2}\right) \mathbb{1}_{[-c,c]}. \tag{5.5}$$

The cutoff parameter c is chosen to be equal to 2.575. For this value of c, the problem is guaranteed to remain well posed provided that $\sigma < \frac{\pi^2}{max(|\xi|)} = 3.8329$.

Polynomials orthogonal to a truncated Gaussian measure are referred to as Rys polynomials [11]. For our implementation of collocation, the sparse grids are based on the zeros of the polynomials orthogonal with respect to the measure determined by (5.5). This leads to an efficient multidimensional quadrature rule using the Gaussian weights and abscissas. We compute the recurrence coefficients for the orthogonal polynomials $\{\alpha_i\}$ and $\{\beta_i\}$ via the discretized Steltjies procedure [11].

Testing for both the sparse grid collocation method and the stochastic Galerkin method is performed using the truncated Gaussian PDF and Rys polynomials for several values of σ. The linear solver in all cases is stopped when $\frac{||r_k||_2}{||b||_2} < 10^{-12}$. We construct the sparse grids using the *Dakota* package [3].

Level/p	σ				
	1	2	3	4	5
1	0.1856	0.1971	0.2175	0.2466	0.2807
2	0.0737	0.0811	.0932	0.1095	0.1207
3	0.0245	.0279	.0331	0.0389	0.1195
4	0.0070	.0082	.0099	0.0121	DNC
5	0.0017	0.0021	.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 5.1

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

Table 5.1 shows the discrete l_∞ mean error for both methods. When $M = 1$ both methods return identical results so only one value for the error is reported in the table. Table 5.2 shows the PCG iteration counts for both methods. Problems to the right of the double line are ill-posed and the Galerkin linear system, as well as a

Level	σ				
	1	2	3	4	5
1	10	10	10.5	11	11
2	10	10.33	10.67	11.33	12.67
3	10	10.5	11	12.25	22
4	10	10.6	11.2	13	DNC
5	10.17	10.5	11.33	13.83	DNC
6	10.14	10.43	11.43	15	DNC
7	10.13	10.63	11.38	16.75	DNC

p	σ				
	1	2	3	4	5
1	13	15	16	18	21
2	13	17	22	28	38
3	14	19	26	39	140
4	14	20	29	53	DNC
5	14	21	31	69	DNC
6	15	21	33	94	DNC
7	15	21	34	136	DNC

TABLE 5.2

Iterations for the stochastic collocation (left) and stochastic Galerkin methods (right)

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 [10]. If the solver fails to converge for any of the individual sub-problems, the method is reported as having failed using "DNC". Table 5.2 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 [10] that if the diffusion is given by a stationary field, as in (5.2), then the condition number of the preconditioned stochastic Galerkin system is proportional to the ratio $\frac{\sigma}{\mu}$ and the magnitude of the largest zero of the degree p orthogonal polynomial. This same result extends to the individual deterministic sub-problems arising in collocation. In 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, then the zeros of the orthogonal polynomials move outward and the problems may again become indefinite. However, if Γ is bounded then all of the orthogonal polynomial zeros are bounded and the systems are guaranteed to remain positive definite, provided σ is not too large.

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

$$-\nabla \cdot [(\mu + \sigma \sum_{k=1}^{M} \lambda_k \xi_k a_k(x))\nabla u] = 1, \tag{5.6}$$

where $M = 3, 4,$ or 5, and $\{\lambda_k, a_k\}$ are the eigenpairs associated with the covariance kernel

$$C(\mathbf{x}_1, \mathbf{x}_2) = exp(-|x_1 - x_2| - |y_1 - y_2|). \tag{5.7}$$

The KL-expansion of this kernel is investigated extensively in [6]. 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 $\mu = .2$ and $\sigma = .1$. For

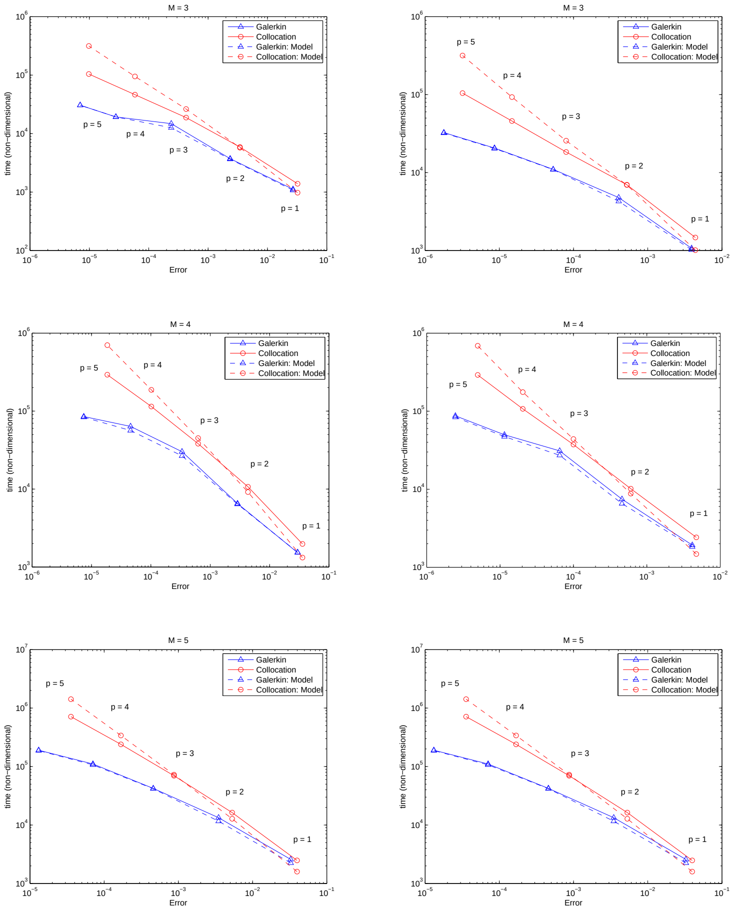

FIG. 5.1. *Solution Time Vs. Error for $M = 3, 4, 5$. Uniform Random Variables (Left) Truncated Gaussian Random Variables (Right)*

the second problem $\mu = 1$ and $\sigma = .25$. These parameters were chosen to ensure that the problem remains well posed but a_{min} is close to zero. In terms of computational effort, this choice 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 became closer to being ill-posed.

Since no analytic expression for the solution to either of these problems is known to exist, to measure the error for both methods the exact solution is approximated by a high order ($p = 10$) approximation. The error in the stochastic space is then estimated by computing the mean of the approximate solutions and comparing it to the mean of the order-10 approximations. The times reported are non-dimensionalized by the time required to perform a single deterministic matrix-vector product and are compared with the model developed above.

Each method produces similar results as a function of the degree of polynomial approximation, as can be seen in the close vertical alignment of the data points in Fig. 5.1. Fig 5.1 shows that the Galerkin method performs significantly better than the collocation method, producing a more accurate solution in less time, and that the gap widens as the dimension of the space of random variables increases. Also, the performance of each method is largely independent of the density functions used in defining the random variables ξ_k. Furthermore, Fig 5.1 shows that the cost model developed above is accurate for the Galerkin method. 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 not valid.

6. Conclusion. We have shown that when mean-based preconditioning is used, the stochastic Galerkin method outperforms the sparse grid collocation method for solving the linear elliptic diffusion when the random diffusion coefficient is modeled by a truncated KL-expansion. This is especially true when the truncated KL-expansion uses a large number of random variables. We have also developed a cost model for both methods that closely mirrors the complexity of the algorithms.

REFERENCES

[1] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1005–1034.

[2] I. Babuška, R. Tempone, and G. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[3] M. Eldred, A. Giunta, B. van Bloemen Waanders, S. Wojtkiewicz, W. Hart, and M. Alleva, *Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 4.0 users manual*, Tech. Rep. SAND2006-6337, Sandia National Laboratories, October 2006.

[4] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.

[5] R. Ghanem and M. Pellissetti, *Iterative solution of systems of linear equations arising in the context of stochastic finite elements*, Advances in Engineering Software, 31 (2000), pp. 607–616.

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

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

[8] M. Loève, *Probability Theory*, Springer-Verlag, New York, 1978.

[9] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2309–2345.

[10] C. Powell and H. Elman, *Block-diagonal preconditioning for spectral stochastic finite element systems*, IMA Journal of Numerical Analysis, 29 (2009), pp. 350–375.

[11] R. Sagar and V. Smith, *On the calculation of Rys polynomials and quadratures*, International Journal of Quantum Chemistry, 43 (1992), pp. 827–836.

[12] D. Xiu and J. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1118–1139.