

Solving the steady-state diffusion equation with
uncertainty
Final Paper

Virginia Forstall
vhfors@gmail.com

Advisor: Howard Elman
elman@cs.umd.edu
Department of Computer Science

May 14, 2012

Abstract

The goal of this project is to explore computational methods designed to solve the steady-state diffusion equation with a random coefficient. Although such equations can be solved using Monte-Carlo methods, the lengthy computation time can be constraining. Using a Karhunen-Loève expansion allows the random coefficient to be approximated with a finite sum of random variables. Using a linear expansion for a lognormal random field produces a sparse stochastic Galerkin system which provides a potentially faster way to compute moments of solution than the Monte-Carlo method.

Contents

1	Introduction	3
2	Algorithm	4
2.1	Karhunen-Loève expansion	4
2.1.1	Gaussian random field	4
2.1.2	Lognormal random field	6
2.2	Solution method	7
2.2.1	Deterministic diffusion equation	7
2.2.2	Monte-Carlo method	8
2.2.3	Stochastic Galerkin method	8
3	Implementation	10
3.1	Spatial discretization	11
3.1.1	Discretization of the eigenvalue problem	11
3.2	Deterministic diffusion equation	13
3.3	Probability density function	13
3.4	Monte-Carlo method	14
3.5	Stochastic Galerkin method	14
4	Results	16
4.1	Solution of the eigenvalue problem	16
4.2	Deterministic diffusion equation	17
4.3	Monte-Carlo method	17
4.4	Probability distribution	19
4.5	Stochastic Galerkin Method	21
5	Conclusion	25
6	References	26

1 Introduction

The goal of this project is to explore computational algorithms for solving the steady-state diffusion equation when the diffusion coefficient is a random field. Formally,

$$-\nabla \cdot (k(x, \omega) \nabla u(x, \omega)) = f(x) \quad \text{on } D \times \Omega \quad (1)$$

$$u(x, \omega) = g(x) \quad \text{on } \partial D_d \quad (2)$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial D_n. \quad (3)$$

Assume the spatial domain $D \subset \mathbb{R}^2$ is bounded. The source term $f(x)$ and Dirichlet boundary condition $g(x)$ are deterministic. The diffusion coefficient, $k(x, \omega)$, is a random field. In the case of this project the random field is assumed to be lognormal. This means $k(x, \omega) = \exp(a(x, \omega))$ where $a(x, \omega)$ is a Gaussian random field. This construction ensures $k(x, \omega)$ is positive for all x and thus guarantees existence and uniqueness of the solution, $u(x, \omega)$ [2].

One application of this equation is modeling groundwater flow through a porous medium [17]. To model groundwater flow, the permeability of the medium is needed at every location in the spatial domain. However, knowing this at every point is infeasible. Instead, the permeability can be treated as a random variable at each point in the domain. Since the permeability at one point is related to the points around it, the random variables will be correlated. Thus the model of a random field is reasonable.

The solution, $u(x, \omega)$, is a function of the sample space, so it is necessary to characterize what is meant by solving this equation. The focus of this project is calculating the moments of the solution. Another meaningful result would be the cumulative distribution function of the solution.

Previous work has been done where $\log(k(x, \omega)) = a(x, \omega)$ was written as an infinite series expansion of random variables [13]. This expansion, known as the Karhunen-Loève expansion, can be truncated to provide an approximation of this field. This expansion, however, is a nonlinear function of random variables which leads to non-sparse matrices in the stochastic Galerkin method. The Karhunen-Loève expansion of $k(x, \omega)$ is a linear function of random variables.

In section 2 the algorithm is outlined. The Karhunen-Loève expansion, the Monte-Carlo method, and stochastic Galerkin method are discussed. The discretization and implementation of the methods is discussed in Section 3. Section 4 discusses the validation of the implementation and presents the results.

2 Algorithm

The algorithm consists of two key parts. The first is to numerically approximate the random field using a Karhunen-Loève expansion. The second part of the algorithm is a method for approximating the solution of the partial differential equation. The Monte-Carlo method and stochastic Galerkin methods are presented.

2.1 Karhunen-Loève expansion

The mean of the random field, $a(x, \omega)$, defined as

$$a_0(x) = \int_{\Omega} a(x, \omega) dP(\omega) \quad (4)$$

and its an associated covariance function is

$$C(x_1, x_2) = \int_{\Omega} (a(x_1, \omega) - a_0(x_1))(a(x_2, \omega) - a_0(x_2)) dP(\omega) . \quad (5)$$

A random field can be written as an infinite series of random variables known as the Karhunen-Loève expansion. The expansion,

$$a(x, \vec{\xi}) = a_0(x) + \sum_{s=1}^{\infty} \sqrt{\lambda_s} a_s(x) \xi_s \quad (6)$$

is written in terms of the eigenvalues, λ_s , and eigenfunctions, $a_s(x)$, of the covariance operator associated with this field. The random variables, ξ_s , are uncorrelated with $E[\xi_s] = 0$, $\text{Var}[\xi_s] = 1$ [15].

The eigenpairs with respect to the covariance operator satisfy,

$$(\mathcal{C}a)(x_1) = \int_D C(x_1, x_2) a(x_2) dx_2 = \lambda a(x_1) , \quad (7)$$

where $C(x_1, x_2)$ is the covariance function defined in equation 5 [6].

2.1.1 Gaussian random field

A Gaussian random field, $a(x, \omega)$, is defined as a random field such that any $a(x_1, \omega)$, $a(x_2, \omega)$, ..., $a(x_n, \omega)$ follow a multivariate normal distribution [7]. Thus, the Gaussian random field can be characterized completely by its

mean and covariance function. In this paper we consider the exponential covariance function. In one spatial dimension this covariance function is

$$C_g(x_1, x_2) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|}{b}\right) \quad (8)$$

where σ^2 is the variance of the random field and b is the correlation length. Note that large values of b indicate that the random variables at points close in space are highly correlated. For this covariance function, the analytic expressions for the eigenvalues and eigenfunctions can be found by solving equation 7 analytically as shown in Reference [6]. The eigenvalues for a one-dimensional domain, $D = [-l, l]$, are

$$\lambda_n = \sigma^2 \frac{2b}{\omega_n^2 + b^2} \quad (9)$$

$$\lambda_n^* = \sigma^2 \frac{2b}{\omega_n^{*2} + b^2} \quad (10)$$

where ω_n and ω_n^* solve the following:

$$b - \omega \tan(\omega l) = 0 \quad (11)$$

$$\omega^* + b \tan(\omega^* l) = 0. \quad (12)$$

The eigenfunctions are

$$f_n(x) = \frac{\cos(\omega_n x)}{\sqrt{l + \frac{\sin(2\omega_n l)}{2\omega_n}}} \quad (13)$$

$$f_n^*(x) = \frac{\sin(\omega_n^* x)}{\sqrt{l - \frac{\sin(2\omega_n^* l)}{2\omega_n^*}}}. \quad (14)$$

Solving equations 11 and 12 with Newton's method gives the numerical values for the eigenvalues of the random field.

The Gaussian field outlined above is one-dimensional. The two-dimensional covariance function is

$$C_g((x_1, y_1), (x_2, y_2)) = \sigma^2 \exp\left(\frac{-|x_1 - x_2|}{b_x} + \frac{-|y_1 - y_2|}{b_y}\right) \quad (15)$$

where b_x and b_y are the correlation lengths in the x and y directions. This function is separable into an x -component and y -component. Thus the eigenvalues for the two-dimensional problem can be determined using the one-dimensional eigenvalues and eigenfunctions.

A nice feature of the Gaussian random field is that in the KL expansion

$$a(x, \vec{\xi}) = a_0(x) + \sum_{n=1}^{\infty} \sqrt{\lambda_n} a_n(x) \xi_n \quad (16)$$

has random variables that are independent Gaussian random variables ($\xi_n \sim N(0, 1)$).

Thus one way to approximate the lognormal field, $k(x, \omega)$, is

$$k(x, \xi) = \exp[a_0(x) + \sum_{i=1}^{m_g} \sqrt{\lambda_i} a_i(x) \xi_i] \quad (17)$$

where m_g is chosen to be the number of terms such that

$$\frac{\sum_{i=1}^{m_g} \lambda_i}{\sum_{i=1}^{\infty} \lambda_i} \geq 0.95 . \quad (18)$$

2.1.2 Lognormal random field

The mean, variance, and covariance function for a lognormal random variable, Y , are

$$E[Y] = e^{\sigma^2/2} \quad (19)$$

$$\text{Var}[Y] = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1) \quad (20)$$

$$C_l((x_1, y_1), (x_2, y_2)) = e^{2\mu + \sigma^2} (e^{C_g((x_1, y_1), (x_2, y_2))} - 1) , \quad (21)$$

where $\ln(Y)$ is a normal random variable with mean μ , standard deviation σ and covariance function, $C_g(x_1, x_2)$ [10].

Thus, another way to represent the lognormal random field is to take the KL expansion directly,

$$\widehat{k}(x, \eta) = k_0(x) + \sum_{i=1}^{\infty} \sqrt{\mu_i} k_i(x) \eta_i . \quad (22)$$

Here the eigenvalues μ_i and eigenfunctions $k_i(x)$ are the eigenpairs of the covariance operator formed using the covariance function defined in equation 21.

An approximation of the field can be found by keeping the largest m_l eigenvalues satisfying equation 18. The computation of the eigenpairs is discussed in Section 3.1.1. The advantage of this approach is that this expansion is linear. However, the distribution of η_i are unknown in this case, a point that is addressed further in Section 3.3.

2.2 Solution method

Two methods are considered for solving the steady-state diffusion equation: the Monte Carlo method and the stochastic Galerkin method. First, we present the solution method for the deterministic diffusion equation.

2.2.1 Deterministic diffusion equation

Consider the diffusion equation with a deterministic coefficient:

$$-\nabla \cdot (k(x)\nabla u(x)) = f(x) \text{ on } D \quad (23)$$

$$u(x) = g(x) \text{ on } \partial D_d \quad (24)$$

$$\frac{\partial u}{\partial n} = 0 \text{ on } \partial D_n. \quad (25)$$

Since $D \subset \mathbb{R}^2$, $x = (x_1, x_2)$. Define

$$L^2(D) = \{v(x) : \int_D v(x)^2 dx < \infty\}. \quad (26)$$

$$H^1(D) = \{v(x) : |v, \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2} \in L^2(D)\}. \quad (27)$$

Define the following subspaces of $H^1(D)$:

$$H_{E_0}^1(D) = \{u \in H^1(D) : u = 0 \text{ on } \partial D_d\} \quad (28)$$

$$H_E^1(D) = \{u \in H^1(D) : u = g(x) \text{ on } \partial D_d\}. \quad (29)$$

The weak formulation of equation 23 is to find $u \in H_E^1(D)$ such that

$$\int_D k(x)\nabla u(x) \cdot \nabla v(x) dx = \int_D f(x)v(x) dx \quad (30)$$

is satisfied for all $v(x) \in H_{E_0}^1(D)$.

Assume that $\phi_j(x)$ are the basis functions associated with the discretization of D with interval size h by h . The finite element solution is

$$u_h(x) = \sum_{j=1}^n u_j \phi_j(x) + \sum_{j=n+1}^{n+n_d} u_j \phi_j(x), \quad (31)$$

where n is the number of elements on the interior and n_d . To find the coefficients u_j , solve $Au = b$ where

$$A_{ij} = \int_D k(x)\nabla \phi_j(x) \cdot \nabla \phi_i(x) dx \quad (32)$$

$$b_i = \int_D \phi_i(x) f(x) dx - \sum_{j=n+1}^{n+n_d} u_j \int_D k(x) \nabla \phi_j(x) \cdot \phi_i(x) dx \quad (33)$$

2.2.2 Monte-Carlo method

We now return to the stochastic problem introduced in section 1. The Monte-Carlo method, based on sampling, allows us to compute statistical quantities such as the sample mean and sample variance of the solution. However, the convergence of the sample mean and sample variance is slow. To implement for this model, samples of the random field $k^{(i)}(x)$ are needed. For each of these samples, the deterministic diffusion equation is solved. Denote the solution $u_h^{(i)}(x)$ for $i = 1, \dots, q$ samples, and the sample mean and variance of the solution can be computed:

$$\mathbf{E}_q[u_h(x)] = \frac{1}{q} \sum_{i=1}^q u_h^{(i)}(x). \quad (34)$$

$$\text{Var}_q[u_h(x)] = \frac{1}{q-1} \sum_{i=1}^q (u_h^{(i)}(x)^2 - \mathbf{E}_q[u_h(x)]^2) \quad (35)$$

The error in the mean is $\mathbf{E}[u] - \mathbf{E}_q[u_h] = \mathbf{E}[u] - \mathbf{E}[u_h] + \mathbf{E}[u_h] - \mathbf{E}_q[u_h]$. Thus the error can be written as the sum of the error in the spatial discretization $\mathbf{E}[u] - \mathbf{E}[u_h]$ and the stochastic error $\mathbf{E}[u_h] - \mathbf{E}_q[u_h]$. The first difference will approach 0 as the size of the spatial discretization h approaches 0. The second difference approaches 0 as $q \rightarrow \infty$, because of the convergence of the Monte-Carlo method [7].

2.2.3 Stochastic Galerkin method

An alternative to the Monte-Carlo solution is to formulate a problem similar to the deterministic problem from which statistical properties of the solution can be estimated. The Karhuen-Loève expansion approximates the random coefficient $k(x, \omega)$ with $\hat{k}^{(m)}(x, \eta)$ from equation 22. Denote the probability density function of $\eta = [\eta_1, \dots, \eta_m]$ as $\hat{\rho}(\eta)$. This approach approximates Ω by Γ where Γ is defined as the support of $\hat{\rho}(\eta)$. See Section 3.3 for more details on $\hat{\rho}(\eta)$.

The stochastic weak formulation of the problem is to find $u(x, \eta) \in H_E^1(D) \times L^2(\Gamma)$ such that

$$\int_{\Gamma} \int_D \hat{k}(x, \eta) \nabla u \cdot \nabla v \hat{\rho}(\eta) dx d\eta = \int_{\Gamma} \int_D f v \hat{\rho}(\eta) dx d\eta \quad (36)$$

$\forall v(x, \eta) \in H_{E_0}^1(D) \times L^2(\Gamma)$. The computation is done by discretizing in space, as described in Section 2.2.1.

The stochastic space, Γ , is discretized using the chaos polynomials denoted $\psi_{\mathbf{j}}(\eta)$. These are products of polynomials in each of the variables η_i .

$$\psi_{\mathbf{j}}(\eta) = \psi_{j_1}(\eta_1)\psi_{j_2}(\eta_2)\dots\psi_{j_m}(\eta_m) . \quad (37)$$

Polynomials are chosen to satisfy the orthogonality condition

$$\mathbb{E}[\psi_{\mathbf{i}}(\eta)\psi_{\mathbf{j}}(\eta)] = \delta_{\mathbf{i}\mathbf{j}} . \quad (38)$$

Just as the interval size h determines the accuracy of the spatial approximation, the accuracy of the chaos polynomial basis is determined by the upper bound (N) on the degree of polynomials,

$$\deg(\psi_{\mathbf{j}}) = \deg(\psi_{j_1}) + \dots + \deg(\psi_{j_m}) \leq N \quad \forall \mathbf{j} . \quad (39)$$

The number of polynomials which satisfy this condition is

$$n_\eta = \binom{N+m}{m} . \quad (40)$$

For simplicity the polynomials are reindexed so that $\psi_l(\eta)$ for $l = 1, \dots, n_\eta$.

Having a basis in each of the spaces, the solution can be written as a combination of products of the basis functions,

$$u_h(x, \eta) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_\eta} u_{ij} \phi_i(x) \psi_j(\eta) . \quad (41)$$

Since the test functions are in $H_{E_0}^1(D) \times L^2(\Gamma)$, it is enough to require that the governing equation holds for all $v(x, \eta) = \phi_k(x)\psi_l(\eta)$.

Therefore, the discrete problem is to find the coefficients u_{ij} which satisfy

$$\begin{aligned} \sum_{i=1}^{n_x} \sum_{j=1}^{n_\eta} \int_\Gamma \int_D u_{ij} \widehat{k}(x, \eta) \nabla \phi_i(x) \cdot \nabla \phi_k(x) \psi_j(\eta) \psi_l(\eta) \widehat{\rho}(\eta) \, dx \, d\eta \\ = \int_\Gamma \int_D f(x) \phi_k(x) \psi_l(\eta) \widehat{\rho}(\eta) \, dx \, d\eta \end{aligned} \quad (42)$$

for each $k = 1, \dots, n_x$ and $l = 1, \dots, n_\eta$ where n_x is the number of points in the spatial discretization and n_η is defined above.

For ease of notation let us define $\mu_0 = 1$ and $\eta_0 = 1$, so the KL expansion can be written

$$\widehat{k}(x, \eta) = \sum_{s=0}^m \sqrt{\mu_s} k_s(x) \eta_s . \quad (43)$$

The solution vector u can be found by solving $\widehat{A}u = b$ where

$$\widehat{A} = \sum_{p=0}^m G_p \otimes A_p \quad (44)$$

$$[A_p]_{ik} = \int_D \sqrt{\mu_p} k_p(x) \nabla \phi_i(x) \cdot \nabla \phi_k(x) dx \quad (45)$$

$$[G_p]_{jl} = \int_{\Gamma} \eta_p \psi_j(\eta) \psi_l(\eta) \widehat{\rho}(\eta) d\eta \quad (46)$$

$$b = \int_D f \phi_k(x) dx \int_{\Gamma} \psi_l \widehat{\rho}(\eta) d\eta . \quad (47)$$

Note that each A_p is $n_x \times n_x$ and each G_p is $n_\eta \times n_\eta$. The moments of the solution can then be computed using the u_{ij} . The mean and the variance of the solution are

$$\mathbb{E}[u_h(x, \eta)] = \sum_{i=1}^{n_x} u_{i1} \phi_i(x) \quad (48)$$

$$\text{Var}[u_h(x, \eta)] = \mathbb{E}[(u_h(x, \eta))^2] - (\mathbb{E}[u_h(x, \eta)])^2 \quad (49)$$

If the KL expansion of the Gaussian random field, $k(x, \xi)$, defined in equation 17, the coefficient is

$$k(x, \xi) = \sum_{j=1}^{n_\xi} \alpha_j \psi_j(\xi) \quad (50)$$

where α_j are defined to match equation 17. This approach would require the sum in equation 44 to be from $p = 1, \dots, n_\eta$. Thus, using a linear expansion produces many fewer terms in the sum and ultimately more sparse Galerkin system.

3 Implementation

All algorithms were implemented using Matlab and run on a Dell desktop computer with 1.9 GB RAM and Intel Core 2 Duo E4600 processor. The code incorporates functions from E. Ullman as part of the SFemLib, the Matlab package IFISS [12], and OPQ [5].

3.1 Spatial discretization

3.1.1 Discretization of the eigenvalue problem

When the spatial domain D is discretized, eigenpairs of the operator can be approximated using the covariance matrix. Let C be the covariance matrix such that

$$C_{ij} = C(x_i, x_j) , \quad (51)$$

where $C(x_i, x_j)$ is the covariance function defined in equation 5.

Following the discussion of [6], the relationship between the eigenvalues and eigenfunctions of the operator in equation 7 and the eigenvalues and eigenvectors of the covariance matrix in equation 51 is derived.

The eigenfunctions, $a_n(x)$, can be written as a linear combination of the spatial basis functions

$$a_n(x) = \sum_j s_j^n \phi_j(x) . \quad (52)$$

Requiring that error terms are normal to the space, gives rise to a discrete approximation to the eigenvalue problem in equation 7. The Galerkin system is

$$GV = \Lambda BV \quad (53)$$

where

$$G_{ij} = \int_D \int_D C(x_1, x_2) \phi_i(x_2) dx_2 \phi_j(x_1) dx_1 \quad (54)$$

$$V_{ij} = v_i^j \quad (55)$$

$$\Lambda_{ij} = \delta_{ij} \lambda_i \quad (56)$$

$$B_{ij} = \int_D \phi_j(x_1) \phi_i(x_1) dx_1 . \quad (57)$$

B is known as the mass matrix. The eigenvalues (the diagonal entries of Λ) and the eigenvectors (the columns of V) provide the information needed to determine the KL expansion. The eigenfunctions should be normalized so that $\int_D a_i(x) a_j(x) dx = \delta_{ij}$ and is chosen so that A , the matrix with columns $a_i(x)$, satisfies $A^T B A = I$.

Consider a one-dimensional domain with uniform intervals. Let $D = [a, b]$ and h be the width of the uniform intervals. The usual hat functions serve as a basis. If we approximate the integrals on the discrete domain using the Riemann midpoint rule,

$$G_{ij} = h^2 C(x_i, x_j) = h^2 C_{ij} \quad (58)$$

$$B_{ij} = h\delta_{ij} . \quad (59)$$

The eigenvalue problem becomes

$$hCV = \Lambda V . \quad (60)$$

The eigenvector approximating the eigenfunction is $a_s(x_j) = \frac{V_{js}}{\sqrt{h}}$ where $v_s(x)$ are the columns of V .

Extending this problem to a two-dimensional domain with uniform intervals of size h_x and h_y . This spatial discretization yields

$$G_{ij} = h_x^2 h_y^2 C_{ij} \quad (61)$$

$$B_{ij} = h_x h_y \delta_{ij} , \quad (62)$$

and the eigenvalue problem becomes

$$h_x h_y CV = \Lambda V . \quad (63)$$

The eigenfunctions will be $a_s(x_j) = \frac{V_{js}}{\sqrt{h_x h_y}}$.

There are two methods we consider for determining the covariance matrix C . The first is used when an analytic expression for the covariance function is known; the other method takes samples of the random field to generate the sample covariance matrix.

Method 1: Evaluate the covariance function

When the covariance function is known, the matrix in equation 61 is formed by evaluating the covariance function at each pair of points. Although this is inexpensive to compute, the covariance function is not always known. So a method using samples of the random field would be useful.

Method 2: Form the sample covariance matrix

If the covariance function $C(x_1, x_2)$ is unknown for a random field but the random field $a(x, \xi)$ can be sampled, the eigenvalues and eigenvectors can be found by constructing the sample covariance matrix. This is defined as

$$\hat{C}_{ij} = \frac{1}{n} \sum_{k=1}^n (a(x_i, \omega_k) - \hat{a}_i)(a(x_j, \omega_k) - \hat{a}_j) \quad (64)$$

$$\hat{a}_i = \frac{1}{n} \sum_{k=1}^n a(x_i, \omega_k) \quad (65)$$

where n is the number of samples and $\hat{a}(x)$ is the sample mean. This matrix \hat{C} is known as the sample covariance matrix.

To save computation time, it is not necessary to compute \hat{C} . Define a matrix

$$F_{ik} = a(x_i, \omega_k) - \hat{a}_i . \quad (66)$$

Then the sample covariance matrix can be written as

$$\hat{C} = \frac{1}{n} F F^T . \quad (67)$$

Therefore, the eigenvalues of \hat{C} can be computed by taking the singular value decomposition, $F = U \Sigma V^T$. The eigenvalues of \hat{C} are $\frac{1}{n} \Sigma^2$ and the eigenvectors are the columns of U . The SVD approach is preferred when there are a large number of samples n taken for two reasons. First, the matrix multiplication is not needed which will save computation time. Second, using the entire covariance matrix can introduce imaginary components to the eigenvalues (with magnitude on the order of machine error). Using the singular value decomposition ensures that the computed eigenvalues are numerically real.

3.2 Deterministic diffusion equation

The Monte-Carlo method requires the solution of the deterministic diffusion problem. This is found using the Matlab package Incompressible Flow & Iterative Solver Software (IFISS) [12]. The domain is $D = [0, 1] \times [0, 1]$ which is discretized with bilinear elements on quadrilaterals of size h by h .

3.3 Probability density function

Consider the approximation of a lognormal field as an exponential of a Gaussian random field in equation 17 and the alternate approximation in equation 22. Recall that in the KL expansion of the Gaussian random field, $\{\xi_i\}$ are independent $N(0, 1)$ variables. Thus their joint pdf, $\rho(\xi)$, is known. However, as discussed in Section 2.2.3, using a linear expansion with the stochastic Galerkin method would produce more sparse matrices. The KL expansion of the lognormal field, equation 22, has random variables η_i which are unknown. However, an approximation to the joint density function $\hat{\rho}(\eta)$ can be derived from the relationship between the two expansions and a change of variables.

Let $m = \max(m_g, m_l)$. Define matrices $A = [a_1|a_2|\dots|a_m]$ and $K = [k_1|k_2|\dots|k_m]$ where the columns are the nodal coefficients of the discrete

eigenfunctions. Let $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ and $M = \text{diag}(\mu_1, \mu_2, \dots, \mu_m)$ be diagonal matrices with the eigenvalues in the Gaussian and lognormal expansions, respectively.

Let the vector $\xi = [\xi_1, \xi_2, \dots, \xi_m]^T$ denote the standard normal random variables in the Gaussian random field and let $\eta = [\eta_1, \eta_2, \dots, \eta_m]^T$ denote the unknown random variables in the lognormal expansion. Equating the two expansion yields

$$\Lambda A^T B(k(x, \xi) - a_0(x)) = \Lambda A^T B(\widehat{k}(x, \eta) - a_0(x)). \quad (68)$$

Thus, the vector ξ can be written as a function of η ,

$$\xi = g(\eta) = \Lambda A^T B(\ln(k_0 + KM\eta) - a_0). \quad (69)$$

Therefore, the joint probability of the density is

$$\widehat{\rho}(\eta) = \rho(g(\eta))|J(\eta)| \quad (70)$$

where $|J(\eta)|$ is the absolute value of the determinant of the Jacobian of g , which can be computed since $g(\eta)$ is differentiable. The support of this distribution is all η such that $k_0 + KM\eta > 0$.

3.4 Monte-Carlo method

Use $k(x, \xi)$ as defined in equation 17. The random variables $\{\xi_s\}$ sampled using m_g $N(0, 1)$ samples to produce a sample of the coefficient $k_i(x)$ for $i = 1, \dots, q$. For each of these samples, the deterministic diffusion equation is solved. The software package IFISS is used to find the solution $u_h^{(i)}(x)$ [12].

3.5 Stochastic Galerkin method

The spatial discretization uses the same bilinear quadrilateral elements as in the deterministic problem ($\phi_i(x)$). These are used to compute the matrices defined in equation 45.

Constructing the Galerkin matrix requires the introduction of an assumption. We will assume that the random variables, η_1, \dots, η_m , are independent and the joint density function can be written $\widehat{\rho}(\eta) = \rho_1(\eta_1)\rho_2(\eta_2)\dots\rho_m(\eta_m)$. With this assumption the stochastic Galerkin matrices in equation 46 consist of p separable integrals. Therefore the one dimensional orthogonal polynomials can be constructed. Treat the i th marginal density function as a weight function, the recurrence coefficient for the orthogonal polynomials in that variable $\psi_j(\eta_i)$ can be computed using the Stieljtes procedure [4].

Below is an outline of the procedure which is implemented to construct the polynomials for each of the m random variables.

For the Stieljtes procedure, the $p + 1$ degree polynomial is constructed using the following:

$$\psi_{p+1}(\eta_i) = (\eta_i - \alpha_p)\psi_p(\eta_i) - \beta_p\psi_{p-1}(\eta_i) \quad (71)$$

for $p = 0, 1, \dots$, where $\psi_{-1}(\eta_i) = 0$ and $\psi_0(\eta_i) = 1$. The recurrence coefficients are

$$\alpha_p = \frac{\int \eta_i \psi_p \psi_p \rho_i(\eta_i) d\eta_i}{\int \psi_p \psi_p \rho_i(\eta_i) d\eta_i} \quad (72)$$

for $p = 0, 1, 2, \dots$ and

$$\beta_p = \frac{\int \psi_p \psi_p \rho_i(\eta_i) d\eta_i}{\int \psi_{p-1} \psi_{p-1} \rho_i(\eta_i) d\eta_i} \quad (73)$$

for $p = 1, 2, \dots$

If the moments of the $\rho_i(\eta_i)$ can be computed easily, the modified Chebyshev procedure can be used to determine these coefficients. However, the number of moments needed for each component of η makes this procedure computationally untractable. Using the multi-component discretization procedure outlined in Gautschi [4] discretizes the measure (marginal density functions) on the finite support of $\rho_i(\eta_i)$, $[a, b]$. The number of points in the discretization is chosen to satisfy some tolerance. It is an iterative method, which takes some number of points R and uses Steiljtes procedure to find the recurrence coefficients and build the polynomials as follows

$$\alpha_{p,R} = \frac{\sum_{t=1}^R \eta_t w_t \psi_{p,R}(\eta_t) \psi_{p,R}(\eta_t) \rho(\eta_t)}{\sum_{t=1}^R w_t \psi_{p,R}(\eta_t) \psi_{p,R}(\eta_t) \rho(\eta_t)} \quad (74)$$

$$\beta_{p,R} = \frac{\sum_{t=1}^R w_t \psi_{p,R}(\eta_t) \psi_{p,R}(\eta_t) \rho(\eta_t)}{\sum_{t=1}^R w_t \psi_{p-1,R}(\eta_t) \psi_{p-1,R}(\eta_t) \rho(\eta_t)} \quad (75)$$

The weights and nodes are found using a Fejer quadrature where the nodes are related to the roots of the chebyshev polynomials.

$$\eta_t = \frac{1}{2}(b - a) \cos\left(\frac{2t - 1}{2R}\right) + \frac{1}{2}(a + b) \quad (76)$$

$$w_t = \frac{1}{R} \left(1 - 2 \sum_{n=1}^{\lfloor R/2 \rfloor} \frac{\cos(2n(\frac{2t-1}{2R}))}{4n^2 - 1} \right) \quad (77)$$

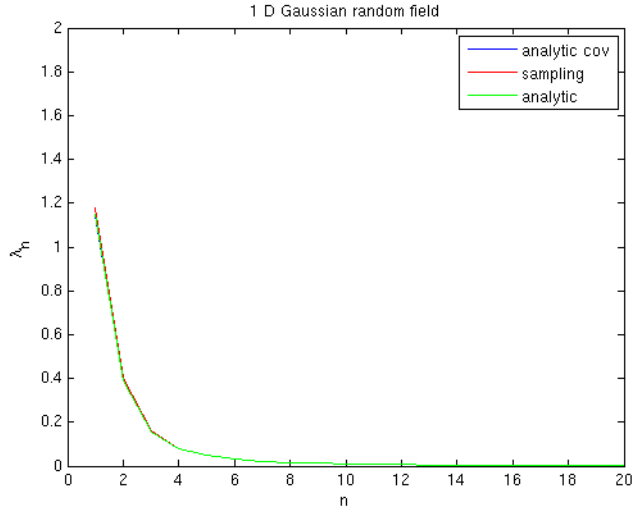


Figure 1: Eigenvalues of Gaussian random field with parameters $b = 1$, computed using analytic expression and the two covariance matrices. The sample covariance matrix contains $q = 10000$ samples.

for $t = 1, \dots, R$.

The stopping condition given R_s is defined by

$$\beta_p^{(s)} - \beta_p^{(s-1)} \leq \epsilon \beta_p^{(s)} \quad (78)$$

where ϵ is the tolerance for convergence. If this condition is not satisfied R is increased to a given R_{cap} .

This procedure was implemented for Matlab by Gautschi [5] where the interval is broken up into component intervals to decrease the time required for convergence.

This procedure was called to construct the polynomials for each of the component marginal density functions. Then the Galerkin system is constructed by multiplying each of the component integrals.

4 Results

4.1 Solution of the eigenvalue problem

To verify the eigenpairs for the Gaussian random field, the eigenvalues computed using the covariance function and the sampling covariance matrix

were compared to the analytic expressions. Figure 1 shows the agreement of these eigenvalues.

Since analytic expressions for the eigenpairs of lognormal field are not known, the eigenvalues and eigenvectors were computed using the two methods discussed in Section 3. The sampling covariance matrix uses samples $k(x_i, \xi_j)$ defined in equation 17. A comparison of these two methods indicates convergence for the sampling method is obtained. (See Figure 4.1).

The procedure for determining the eigenvalues from the sample covariance matrix has been verified. At the beginning of the project, this was the method that would be used to find the eigenvalues of the lognormal random field $k(x, \eta)$. However, through the course of this project, the analytic expression for the covariance function from [10] was discovered. Using the analytic construction of the covariance matrix is faster and more accurate, so it is used in the remainder of the project. The validation of the sampling method illustrates the way in which this model could be used in an application where the covariance function is not known.

4.2 Deterministic diffusion equation

To obtain an idea of the error in the solution due to the spatial discretization, consider the finite element solution to the diffusion equation with a deterministic coefficient as computed by the solver IFISS on the domain $D = [0, 1] \times [0, 1]$. For $u(\vec{x}) = 0$ on the boundary, $f(\vec{x}) = 1$, $k(\vec{x}) = 1$, $h = 0.0625$, $n_d = 64$, $n = 225$, IFISS produces the solution shown in Figure 3. For the given source, coefficient, and boundary conditions there is a known analytic series solution [1]:

$$u(x_1, x_2) = \frac{16}{\pi^4} \sum_{l=0}^{\infty} \sum_{k=0}^{\infty} \frac{\sin((2k+1)\pi x_1) \sin((2l+1)\pi x_2)}{(2k+1)(2l+1)((2k+1)^2 + (2l+1)^2)}. \quad (79)$$

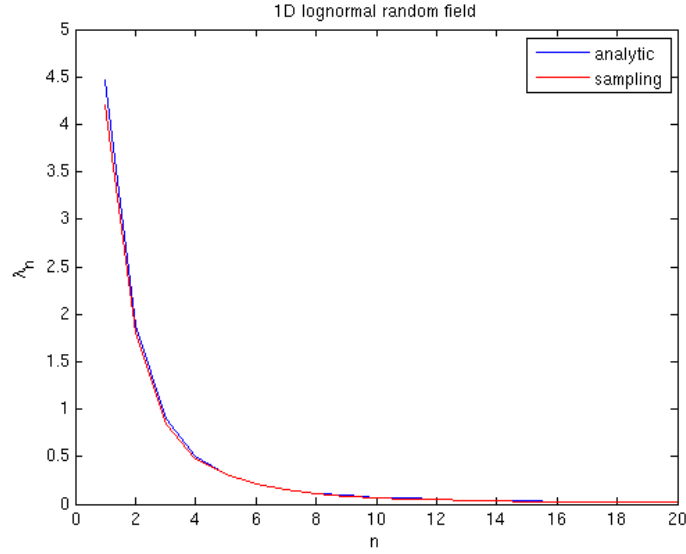
Thus the error due to the spatial discretization is

$$\frac{\|u_h(\vec{x}) - u(\vec{x})\|_2}{\|u(\vec{x})\|_2} = 3.31 \times 10^{-3} \quad (80)$$

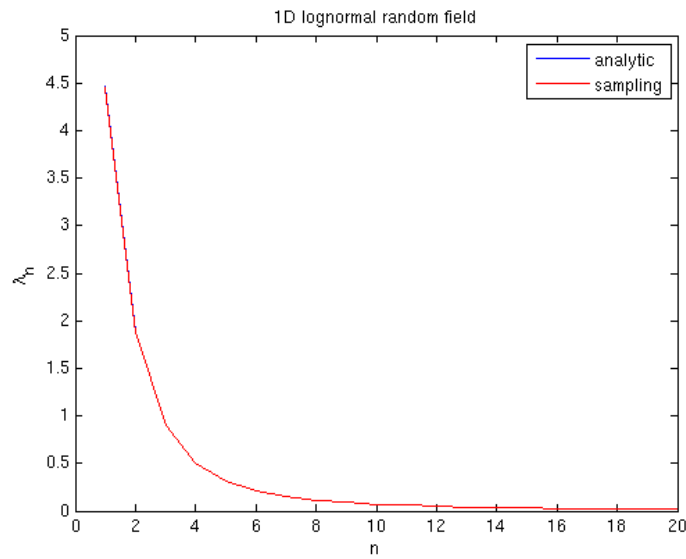
The same spatial discretization with this value of h will be used for the stochastic problem.

4.3 Monte-Carlo method

The Monte-Carlo method was validated by checking that the sample mean of the solution converges to the deterministic solution with coefficient $E[k]$,



(a) $n=1000$



(b) $n=10000$

Figure 2: Eigenvalues of $k(x, \xi)$, a lognormal random field. The eigenvalues obtained using the sample covariance matrix, converge to the analytic covariance matrix results as the number of samples is increased. Figures use correlation length $b = 1$.

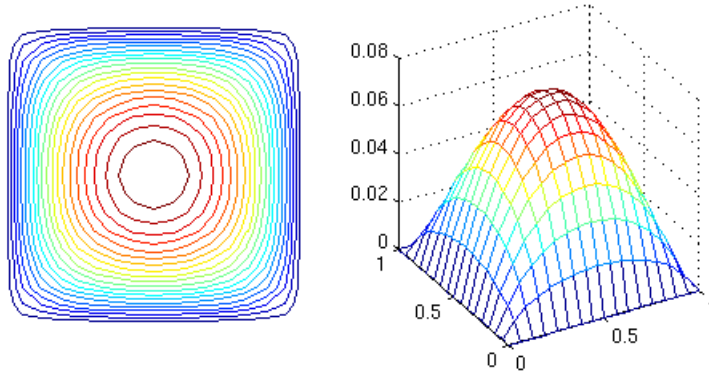


Figure 3: Deterministic solution, $k(\vec{x}) = 1$

when the variance is small. Running the code with variance is 0 and $E[k] = 1$ produces an error $||E[u_h] - E_q[u_h]||$ on the order of machine epsilon. Once the variance is large enough, the mean solution is affected. Figure 4 contains two examples of the Monte-Carlo solution with a nonzero variance. Not surprisingly, fewer samples q are required for convergence for problems with a smaller variance.

4.4 Probability distribution

Several steps were taken to verify the probability distribution discussed in Section 3.3. For computational ease the distribution was checked using one dimension of space. A high correlation length was chosen, $b = 10$. With this correlation, enough of the variance is captured in the first eigenvalue. 97.7% of variance of the Gaussian random field is captured with the two term expansion and 95.9% for the first eigenvalue of the lognormal expansion. Thus the density function $\hat{\rho}(\eta_1)$ will be a good approximation of the joint density. Figure 5 shows this density. Note the sharp cutoff just below zero. This occurs where the $k_0 + KM\eta$ is zero and thus extra resolution is needed in this area. First, Matlab's quadrature routine was used to check that this function integrates to 1. The mean was approximately 0 and the variance was approximately 1, agreeing with the characteristics of η_1 in the KL expansion.

Using the probability density function $\hat{\rho}(\eta_1, \eta_2)$ for the same correlation length more variance can be incorporated in the expansion. Shown in Figure 6 this probability density also integrates to 1.

Figure 4: Monte Carlo solution: $E[k(x, \xi)] = 1$, $f(x) = 1$, $m = 5$, $g(x) = 0$.

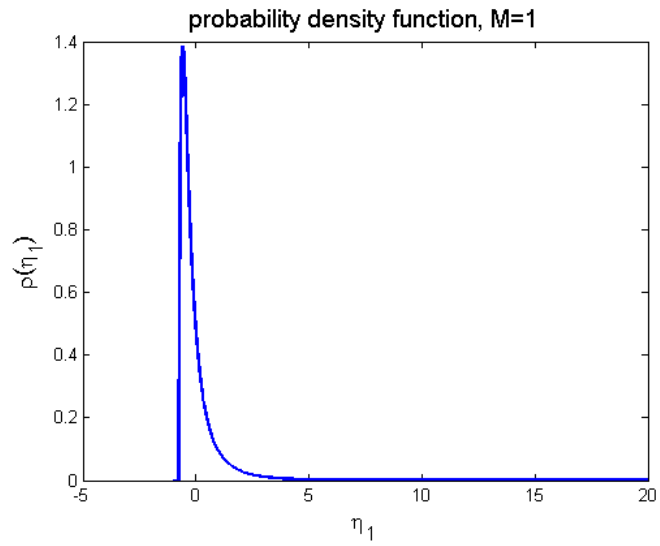
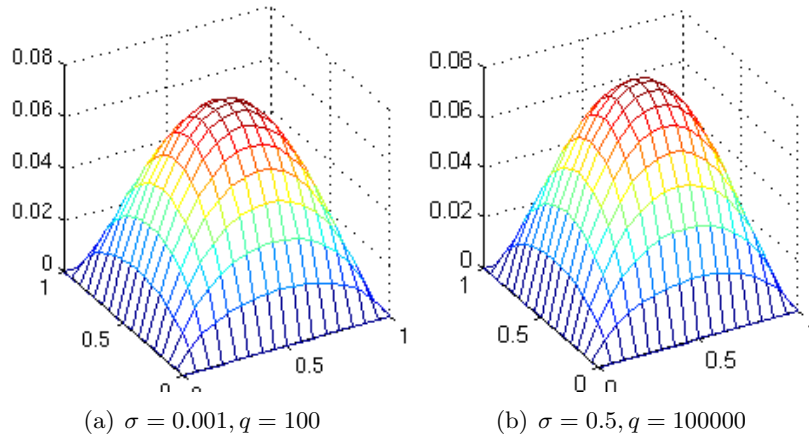


Figure 5: Probability density function for $m = 1$, $b = 10$.

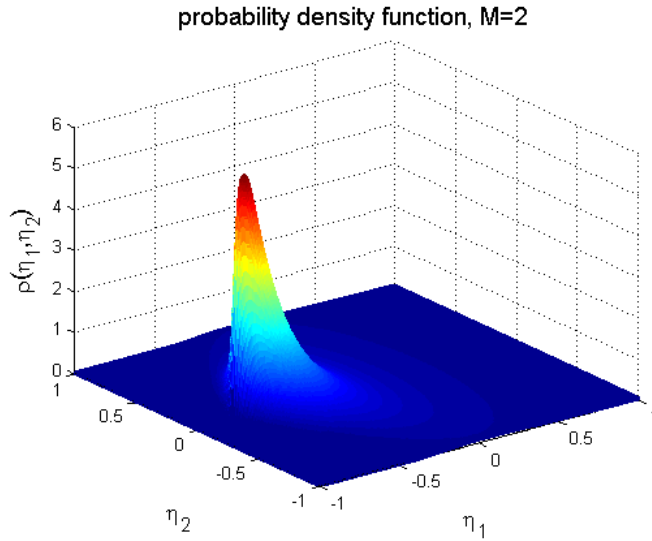


Figure 6: Probability density function for $m = 2$, $b = 10$.

The probability density function was also validated by comparing the sample means of $k(x, \xi)$ and $\hat{k}(x, \eta)$. The sample mean of $k(x, \xi)$ is simple to compute, because it depends on independent normal random variables. To generate samples of $\hat{k}(x, \eta)$, samples of this joint density function are needed. To do this, accept-reject sampling is used. Uniform samples are taken over the support of the density function. This is the candidate sample. The joint density function is then evaluated at this point. One other uniform sample on $(0,1)$ is generated. If the result of the joint density function evaluation is below this uniform sample, the sample is accepted. Otherwise, it is rejected. Repeating this procedure many times, samples of the lognormal density function are obtained and the coefficient $\hat{k}(x, \eta)$ can be computed. Generating the equivalent number of Gaussian samples, the sample means can be computed. The sample means are shown to converge to the same value in Figure 7. Note this sampling technique is computationally expensive, and a large number of samples ($q \sim 10^6$) are required for convergence.

4.5 Stochastic Galerkin Method

Having verified the Monte-Carlo method the results from that method can be used to verify the results of the Stochastic Galerkin solver. All Monte-

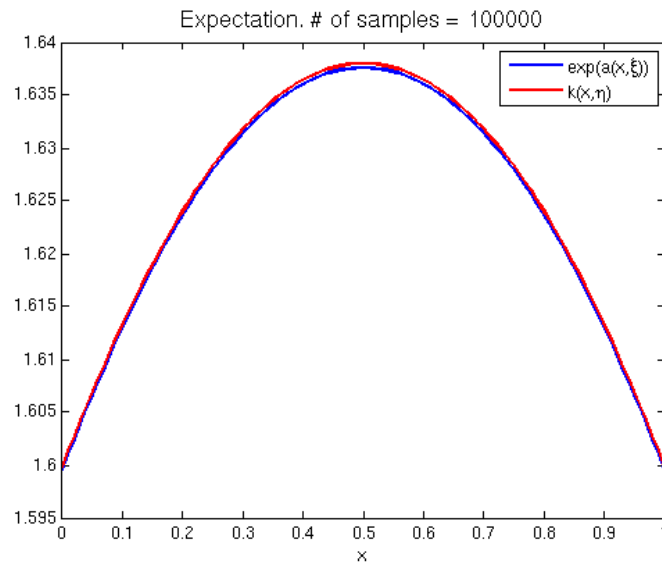
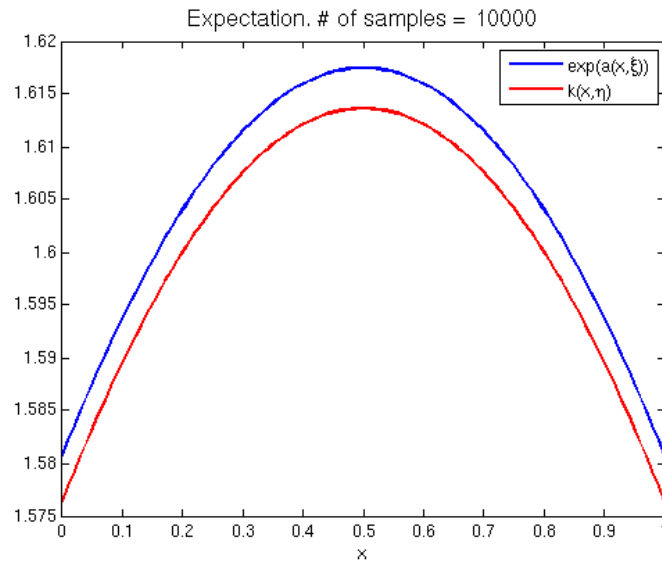
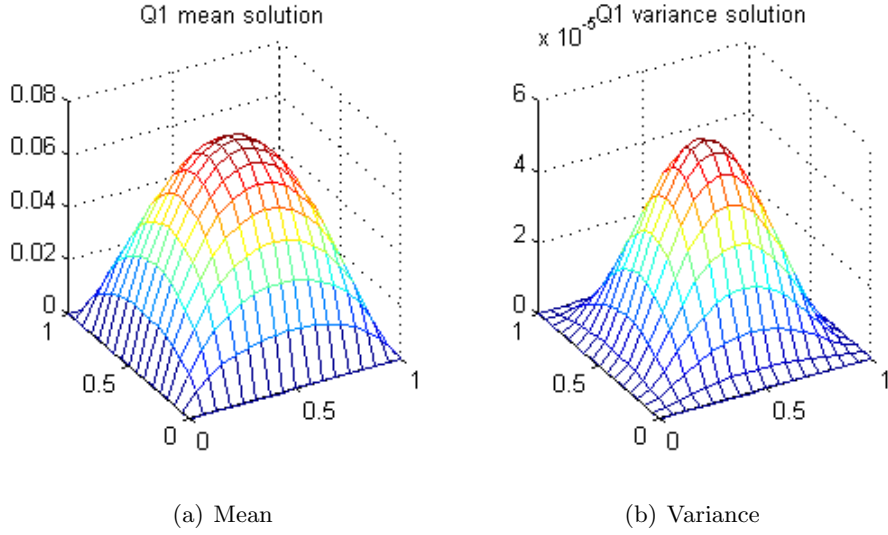


Figure 7: Compare $E[\hat{k}(x, \eta)]$ and $E[k(x, \xi)]$ using Monte-Carlo method. Lognormal samples found using accept/reject technique.

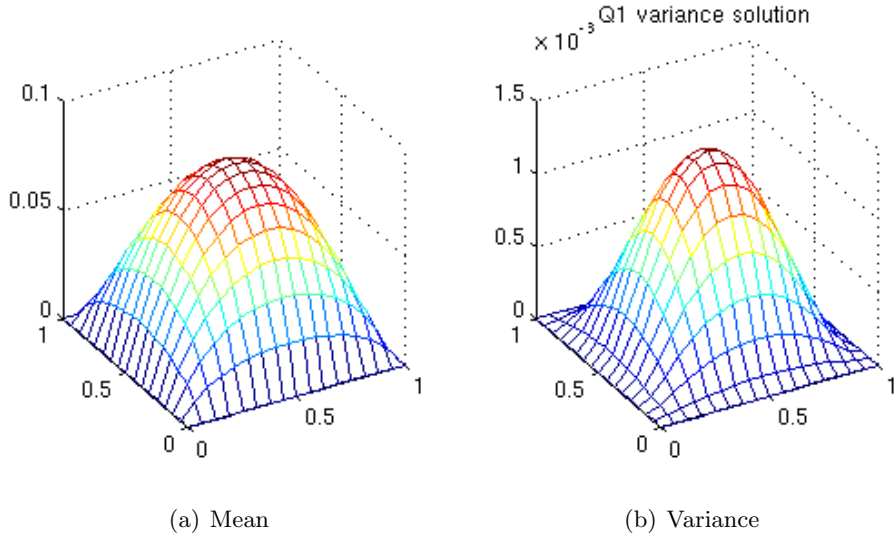
Figure 8: Stochastic Galerkin solution: $E[k(x, \eta)] = 1$, $f(x) = 1$, $m = 1$, $\sigma^2 = 0.1$, $b_x = b_y = 10$.



	$\ MC\ $	$\ SG\ $	$\ MC - SG\ $	$\frac{\ MC - SG\ }{\ MC\ }$
$E[u]$	6.65×10^{-1}	6.65×10^{-1}	3.07×10^{-4}	4.61×10^{-4}
σ	6.49×10^{-2}	6.44×10^{-2}	5.84×10^{-4}	9.00×10^{-3}

Table 1: Results, $m = 1$, $\sigma = 0.1$, $b_x = b_y = 10$

Figure 9: Stochastic Galerkin solution: $E[k(x, \eta)] = 1$, $f(x) = 1$, $m = 2$, $\sigma = 0.5$, $b_x = b_y = 10$.



Carlo solutions presented in this section use $m = 5$ terms in the expansion of the Gaussian random field. Recall that in order to separate the integrals defining the Galerkin matrices in equation 46, an assumption was made that the joint density function $\hat{\rho}(\eta)$ can be written as a product of the marginal density functions. In the case of $m = 1$ no approximation is needed. To get as much of the variance captured by the first eigenvalue, a high correlation length $b_x = b_y = 10$ and a small standard deviation $\sigma = 0.1$ were used. This enables the KL expansion to incorporate 93.22% of the variance with only two terms (the mean and one random variable). The stochastic Galerkin method to compute the mean and the standard deviation of the solution. The comparison to Monte-Carlo is contained in Table 1.

The Monte-Carlo solution ($q = 10,000$) takes 5.5 minutes. The stochastic Galerkin solution requires 3.3 minutes. This standard deviation for this problem is quite small meaning the solution is not far off from the deterministic solution. To handle a higher standard deviation, more terms in the KL expansion are needed.

When the expansion is extended to include another random variable, the assumption of independence is introduced. For a standard deviation,

	$\ MC\ $	$\ SG\ $	$\ MC - SG\ $	$\frac{\ MC - SG\ }{\ MC\ }$
$E[u]$	7.43×10^{-1}	7.41×10^{-1}	2.94×10^{-3}	3.95×10^{-3}
σ	3.81×10^{-1}	3.31×10^{-1}	5.05×10^{-2}	1.32×10^{-1}

Table 2: Results, $m = 2$, $\sigma = 0.5$, $b_x = b_y = 10$

$\sigma = 0.5$ and $b_x = b_y = 10$, with two variables 94.67% of the variance is captured. Table 2 contains these results.

We see from these results, that the approximation has caused an increase in the error, especially in the standard deviation of the solution. This solution method behaves properly for standard deviations up to $\sigma = 0.5$. A standard deviation higher than this introduces a singularity in the solution, likely from the break down of the approximation or by the small value of m . However, the $\sigma = 0.5$ solution is far enough from the deterministic solution that we can be sure the stochastic Galerkin solution is in fact an approximation to the stochastic problem. The simulations outlined above use polynomials of maximum degree $N = 3$. Increasing this degree can help the expand the range of standard deviations.

As for computation time, Monte-Carlo results ($q = 100,000$) for $m = 2$ is about 3.5 hours, while the stochastic Galerkin method took approximately 0.5 hours. The majority of the time (79.28%) spent on the stochastic Galerkin method is in computing the two marginal density functions needed. Using a more creative quadrature than Matlab's quad routine could improve this computation time.

5 Conclusion

The stochastic Galerkin method has been verified and the solution method in the case of $m = 1$ and $m = 2$ is much faster than the Monte-Carlo method. Because the stochastic Galerkin method scales as the number of parameters m increases, the $m = 3$ case will take longer than the Monte-Carlo solution. One potential improvement to the stochastic Galerkin code is in the computation of the marginal density functions. Since this is the most expensive piece of the code, using a less accurate quadrature than Matlab's quad routine could provide a significant speedup. It is also possible to solve the system for several different spatial and stochastic discretization by saving these marginal density functions ahead of time. Using a linear expansion produces a sparse stochastic Galerkin matrix so solving the matrix system is quick.

Having this linear expansion and its joint density function illustrates that the problem can be solved using the direct expansion of the lognormal field. In fact, the problem with computing marginal densities could potentially be avoided by using the Stochastic Collocation Method. This method requires $\hat{\rho}(\eta)$ but not the orthogonal polynomials. Thus, the collocation method is an approach with the potential to avoid some of the difficulties of the stochastic Galerkin method, such as the computational constraint of the marginal densities and the unfounded assumption of independence of the random variables.

The deliverables include the code for computing the moments via Monte-Carlo and stochastic Galerkin methods. The other deliverables, the comparison of m values and computational times is included in Section 4.5.

6 References

References

- [1] N. Asmar, Partial Differential Equations with Fourier Series and Boundary Value Problems, Pearson Prentice Hall, New Jersey, 2005.
- [2] I. Babuška, F. Nobile, and R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal Numerical Analysis, 45, (2007), 1005-1034.
- [3] H. C. Elman, D. J. Silvester, and A. J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, Oxford University Press, New York, 2005.
- [4] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, New York, 2004.
- [5] W. Gautschi, OPQ: A Matlab Suite of Programs for Generating Orthogonal Polynomials and Related Quadrature Rules, <http://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html>.
- [6] R. Ghanem and P. Spanos, Stochastic Finite Elements: A Spectral Approach, Dover Publications, Mineola, New York, 2003.
- [7] A. Gordon, Solving stochastic elliptic partial differential equations via stochastic sampling methods , M.S Thesis, University of Manchester, 2008.

- [8] C. Moler, Numerical Computing with Matlab, Chapter 10: Eigenvalues and Singular Values, 2004, <http://www.mathworks.com/moler/chapters.html>.
- [9] C.E. Powell and H.C. Elman, Block-diagonal preconditioning for spectral stochastic finite-element systems, IMA Journal of Numerical Analysis, 29, (2009), 350-375.
- [10] J. Rendu, Normal and lognormal estimation, Mathematical Geology, 11, 4, (1979), 407-422.
- [11] C. Schwab and R. Todor, Karhunen-Loève approximation of random fields by generalized fast multipole methods, Journal of Computational Physics, 217, (2006), 100-122.
- [12] D. J. Silvester, H. C. Elman, and A. Ramage, Incompressible Flow & Iterative Solver Software, <http://www.cs.umd.edu/~elman/ifiss/index.html>.
- [13] E. Ullmann, H. C. Elman, and O. G. Ernst, Efficient iterative solvers for stochastic Galerkin discretization of log-transformed random diffusion problems, SIAM Journal on Scientific Computing, 34, (2012), A659-A682.
- [14] X. Wan and G. Karniadakis, Solving elliptic problems with non-Gaussian spatially-dependent random coefficients, Computational Methods in Applied Mechanical Engineering, 198, (2009), 1985-1995.
- [15] D. Xiu, Numerical Methods for Stochastic Computations, Princeton University Press, New Jersey, 2010.
- [16] D. Xiu and J. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM Journal on Scientific Computing, 27, (2005), 1118-1139.
- [17] D. Zhang, Stochastic Methods for Flow in Porous Media. Coping with Uncertainties, Academic Press, San Diego, CA, 2002.