# Solving An Elliptic Problem in a Random Domain

Paul Constantine

December 9, 2006

## 1 Introduction

The problem of solving a differential equation in a domain with rough geometries has many applications in engineering. One way to model the roughness in geometry is with a random parameterization. In this project, I employed one particular representation of a random geometry – a Karhunen-Loeve type expansion that represents a random field as a finite linear combination of smooth oscillatory functions – and used a technique developed by Xiu and Tartakovsky [1] to solve the resulting random domain problem. This technique involves creating the appropriate random mapping to transform the deterministic problem on a random domain into a stochastic boundary value problem (SBVP) on a deterministic domain. By applying the mapping, the Jacobian terms become stochastic, variable coefficients of a transformed SBVP. The resulting SBVP can be solved using a variety of techniques including a Monte Carlo simulation or a stochastic Galerkin method. These techniques typically compute the first and second moments of the solution at each grid point. With the second moments available, one can determine where the solution is most sensitive to the randomness in the original random domain. In other words, the larger the standard deviation of the solution at a particular grid point, the more sensitive it is to the roughness in geometry. With this information, we have successfully quantified the uncertainty in the solution introduced by the roughness in the domain. In what follows, I present a particular application of this technique by reproducing one of the numerical examples from [1].

## 2 Problem Statement

Consider the following random domain problem. Let $D(\omega) = \{(x_1, x_2) \mid -1 \leq x_1 \leq 1,\ s(x_1, \omega) \leq x_2 \leq 1\}$ where $s(x_1, \omega)$ has a mean equal to negative one, $\langle s(x_1, \omega)\rangle = -1$, and two point covariance function

$$C_s(x_1, z_1) \equiv \mathrm{E}[s(x_1, \omega)s(z_1, \omega)] = e^{-|x_1 - z_1|}.$$

We see a stochastic solution $u(x_1, x_2, \omega)$ such that for almost every $\omega$

$$\Delta u = 0, \qquad \text{in } D(\omega)$$

with boundary conditions

$$\begin{array}{ll} u(1, x_2) = 0 & u(-1, x_2) = 0 \\ u(x_1, 1) = 1 & u(x_1, s(x_1, \omega)) = 0. \end{array}$$

Note that one can increase the variation in the $D(\omega)$ by adjusting the two point covariance function appropriately.

To make this problem computationally feasible, we use a Reynolds-type decomposition to represent $s(x_1, \omega)$ as $\langle s(x_1)\rangle + s'(\omega)$ where $s'(\omega)$ is the zero-mean fluctuation of the process. In particular, we employ the Karhunen-Loeve type expansion [4] to decompose

$$s(x_1, \omega) = \underbrace{\langle s(x_1)\rangle}_{=\,-1} + \sigma \sum_{k=1}^{K} \sqrt{\lambda_k}\psi_k(x_1)Y_k(\omega)$$

where $\{Y_k(\omega)\}$ are mutually independent random variables uniformly distributed over the interval $[-1,1]$ and $\{\lambda_k, \psi_k\}$ are the eigenvalues and eigenfunctions of the integral equations

$$\int_{-1}^{1} C_s(x_1, z_1)\psi_k(z_1)\, dz_1 = \lambda_k \psi_k(x_1), \qquad k = 1, \dots, K.$$

Here, $\sigma \in (0,1)$ is a parameter that controls the amount of randomness in the domain. The solutions to the integral equations have been worked out in [2]; if $\omega_n$ and $\omega_n^*$ solve

$$1 - \omega_n \tan \omega_n = 0 \qquad \omega_n^* + \tan \omega_n^* = 0,$$

which we can compute explicitly with a root-finder routine, then

$$\psi_n(x_1) = \frac{\cos \omega_n x_1}{\sqrt{1 + \frac{\sin 2\omega_n}{2\omega_n}}} \qquad \psi_n^*(x_1) = \frac{\sin \omega_n^* x_1}{\sqrt{1 - \frac{\sin 2\omega_n^*}{2\omega_n^*}}}$$

for $n$ even and odd, respectively, and

$$\lambda_n = \frac{2}{\omega_n^2 + 1} \qquad \lambda_n^* = \frac{2}{(\omega_n^*)^2 + 1}$$

Equipped with a representation for the randomness in the domain, we can forge ahead in our quest to quantify the uncertainty in the solution introduced by the random domain.

# 3 A Random Mapping

A change of variables is often used to transform a problem with a complicated geometry into a problem with simple geometry that is easily solved. Creating the mapping involves solving Laplace's equation for each coordinate of the mapping. We apply the same technique to find the random mapping that will transform our problem from a deterministic problem on a random domain to a stochastic problem on a deterministic domain. Thus given the deterministic domain

$$E = \{(\xi_1, \xi_2) \mid -1 \le \xi_1 \le 1,\ -1 \le \xi_2 \le 1\}$$

we seek to construct a mapping $\mathbf{x}(\xi_1, \xi_2, \omega) = (x_1(\xi_1, \xi_2, \omega), x_2(\xi_1, \xi_2, \omega))$ such that $\mathbf{x}(E, \omega) = D(\omega)$. To find $\mathbf{x}$, we solve the following Laplace equations.

$$\frac{\partial^2 x_1}{\partial \xi_1^2} + \frac{\partial^2 x_1}{\partial \xi_2^2} = 0, \quad \frac{\partial^2 x_2}{\partial \xi_1^2} + \frac{\partial^2 x_2}{\partial \xi_2^2} = 0, \quad \text{in } E$$

subject to

$$\begin{array}{llll} x_1(1, \xi_2) = 1 & x_1(-1, \xi_2) = -1 & x_2(1, \xi_2) = \xi_2 & x_2(-1, \xi_2) = \xi_2 \\ x_1(\xi_1, 1) = \xi_1 & x_1(\xi_1, -1) = \xi_1 & x_2(\xi_1, 1) = 1 & x_2(\xi_1, -1) = s(\xi_1, \omega) \end{array}$$

The solution to first of these equations is readily seen to be $x_1(\xi_1, \xi_2) = \xi_1$ since there is no random input in this system. Due to the our representation of $s(x_1, \omega)$, we thus have a representation of $x_2$ as

$$x_2(\xi_1, \xi_2, \omega) = \sum_{k=0}^{K} \hat{x}_{2,k} Y_k(\omega), \quad \text{where } Y_0 = 1.$$

Substituting this expansion into the appropriate Laplace equation and collecting terms, we get $K+1$ equations for the expansion coefficients.

$$\frac{\partial^2 \hat{x}_{2,0}}{\partial \xi_1^2} + \frac{\partial^2 \hat{x}_{2,0}}{\partial \xi_2^2} = 0$$

subject to

$$\begin{array}{ll} \hat{x}_{2,0}(1,\xi_2) = \xi_2 & \hat{x}_{2,0}(-1,\xi_2) = \xi_2 \\ \hat{x}_{2,0}(\xi_1,1) = 1 & \hat{x}_{2,0}(\xi_1,-1) = -1. \end{array}$$

And for $k = 1, \ldots, K$ solve

$$\frac{\partial^2 \hat{x}_{2,k}}{\partial \xi_1^2} + \frac{\partial^2 \hat{x}_{2,k}}{\partial \xi_2^2} = 0$$

subject to

$$\begin{array}{ll} \hat{x}_{2,k}(1,\xi_2) = 0 & \hat{x}_{2,k}(-1,\xi_2) = 0 \\ \hat{x}_{2,k}(\xi_1,1) = 0 & \hat{x}_{2,k}(\xi_1,-1) = \sigma\sqrt{\lambda_k}\psi_k(\xi_1) \end{array}$$

Again by inspection $\hat{x}_{2,0}(\xi_1,\xi_2) = \xi_2$. The remaining $K$ equations can be solved numerically to get the coefficients of the expansion of the mapping at each grid point in the computational domain. To do this we need to solve the 2D Laplace equation.

# 4 2D Laplace Equation

Let $\mathbf{D}$ be a Chebyshev differentiation matrix in 1D defined by

$$\mathbf{D}_{jk} = \begin{cases} \frac{c_j(-1)^{j+k}}{c_k(x_j-x_k)} & j \neq k \\ \frac{-x_j}{2(1-x_j^2)} & j = k,\ j \neq 0, N \\ \frac{2N^2+1}{6} & j = k = 0 \\ -\frac{2N^2+1}{6} & j = k = N \end{cases}$$

where $x_j = \cos(\pi j/N)$, and $c_i = 2$ if $i \in \{1, N\}$ and $c_i = 1$ otherwise. We discretize the $(\xi_1, \xi_2)$ grid as a tensor product grid of two vectors of Chebyshev collocation points in the interval $[-1, 1]$, i.e.

$$(\xi_{1,j}, \xi_{2,k}) = (\cos(\pi j/N),\ \cos(\pi k/N)), \qquad \text{for } j, k = 0, \ldots, N.$$

If we let $\mathbf{I}$ be an $N+1 \times N+1$ identity matrix, then the matrix representing the 2D Laplace operator is

$$\mathbf{L} = (\mathbf{D} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{D})$$

We modify this operator to enforce the boudary conditions by replacing the rows corresponding to the boundary points with a row vector of zeros with a 1 in the diagonal position – call this modified operator $\tilde{\mathbf{L}}$. We replace the same entries in the right hand side of the system $\mathbf{b}^{(k)}$ with the appropriate boundary conditions. Thus we enforce the equations

$$\begin{array}{llll} \mathbf{u}_{:,0} & = u(\xi_{1,N}, \xi_{2,:}) & = 0 & = \mathbf{b}^{(k)}_{:,0} \\ \mathbf{u}_{:,N} & = u(\xi_{1,0}, \xi_{2,:}) & = 0 & = \mathbf{b}^{(k)}_{:,N} \\ \mathbf{u}_{0,:} & = u(\xi_{1,:}, \xi_{2,0}) & = 0 & = \mathbf{b}^{(k)}_{0,:} \\ \mathbf{u}_{N,:} & = u(\xi_{1,:}, \xi_{2,N}) & = \sigma\sqrt{\lambda_k}\psi_k(\xi_{1,:}) & = \mathbf{b}^{(k)}_{N,:} \end{array}$$

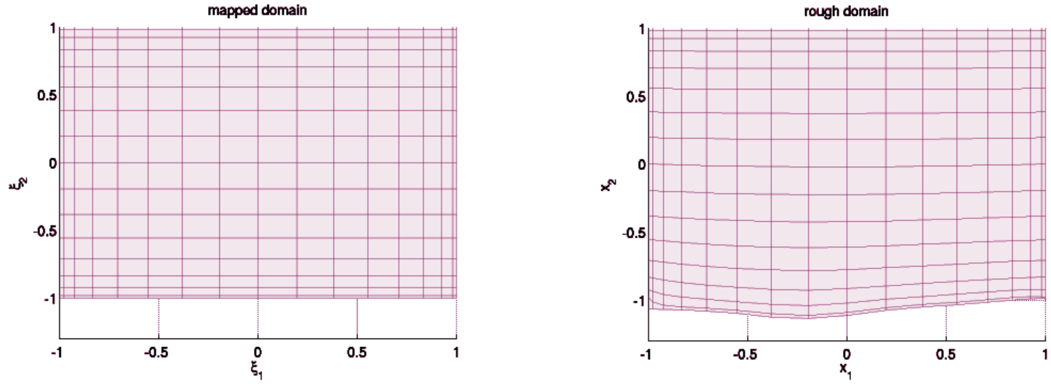
The colon (:) notation indicates taking the set of indices in that particular coordinate. Then we can solve for $\hat{x}_{2,k}$ at the grid points – which we denote with the vector $\hat{\mathbf{x}}_{2,k}$ – by solving the systems

$$\tilde{\mathbf{L}}\hat{\mathbf{x}}_{2,k} = \mathbf{b}^{(k)}, \quad k = 1, \ldots, K$$

Once we have the $\hat{\mathbf{x}}_{2,k}$ we have the mapping $x_2(\xi_1, \xi_2)$ defined at the grid points.

$$\mathbf{x}_2 = \xi_2 + \sum_{i=1}^{K} \hat{\mathbf{x}}_{2,k} Y_k.$$

Below is a plot of one realization of the rough domain and its mapped counterpart.




# 5 The Jacobian Terms - Part I

Note that

$$x_1(\xi_1, \xi_2) = \xi_1 \quad \Rightarrow \quad \frac{\partial x_1}{\partial \xi_1} = 1$$

and

$$x_1(\xi_1, \xi_2) = \xi_1 \quad \Rightarrow \quad \frac{\partial x_1}{\partial \xi_2} = 0.$$

This will simplify the transformation in the next section. The inverse Jacobian of the mapping is given by

$$\begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{\frac{\partial x_2}{\partial \xi_1}}{\frac{\partial x_2}{\partial \xi_2}} & \frac{1}{\frac{\partial x_2}{\partial \xi_2}} \end{bmatrix}$$

so that

$$\frac{\partial \xi_1}{\partial x_1} = 1 \qquad \frac{\partial \xi_1}{\partial x_2} = 0.$$

These facts will greatly reduce the algebra when applying the random mapping to the equation.

# 6 The Transformed Equation

We must transform the original equation in $D(\omega)$ to the corresponding equation in $E$. To do this, we simply apply the chain rule and the product rule. For each operator, we have

$$\frac{\partial u}{\partial x_1} = \frac{\partial u}{\partial \xi_1} \underbrace{\frac{\partial \xi_1}{\partial x_1}}_{=1} + \frac{\partial u}{\partial \xi_2} \frac{\partial \xi_2}{\partial x_1} = \frac{\partial u}{\partial \xi_1} + \frac{\partial u}{\partial \xi_2} \frac{\partial \xi_2}{\partial x_1}$$

And

$$\begin{aligned} \frac{\partial}{\partial x_1}\left(\frac{\partial u}{\partial x_1}\right) &= \frac{\partial}{\partial x_1}\left(\frac{\partial u}{\partial \xi_1}\right) + \frac{\partial}{\partial x_1}\left(\frac{\partial u}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_1}\right) \\ &= \frac{\partial^2 u}{\partial \xi_1^2} \underbrace{\frac{\partial \xi_1}{\partial x_1}}_{=1} + \frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}\frac{\partial \xi_2}{\partial x_1} + \frac{\partial^2 \xi_2}{\partial x_1^2}\frac{\partial u}{\partial \xi_2} + \frac{\partial \xi_2}{\partial x_1}\left[\frac{\partial^2 u}{\partial \xi_1 \xi_2} \underbrace{\frac{\partial \xi_1}{\partial x_1}}_{=1} + \frac{\partial^2 u}{\partial \xi_2^2}\frac{\partial \xi_2}{\partial x_1}\right] \\ &= \frac{\partial^2 u}{\partial \xi_1^2} + 2\frac{\partial \xi_2}{\partial x_1}\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2} + \left(\frac{\partial \xi_2}{\partial x_1}\right)^2 \frac{\partial^2 u}{\partial \xi_2^2} + \frac{\partial^2 \xi_2}{\partial x_1^2}\frac{\partial u}{\partial \xi_2} \end{aligned}$$

And

$$\frac{\partial u}{\partial x_1} = \frac{\partial u}{\partial \xi_1}\underbrace{\frac{\partial \xi_1}{\partial x_2}}_{=0}+\frac{\partial u}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2} = \frac{\partial u}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2}$$

And, finally

$$\begin{aligned}
\frac{\partial}{\partial x_2}\left(\frac{\partial u}{\partial x_2}\right) &= \frac{\partial}{\partial x_2}\left(\frac{\partial u}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2}\right) \\
&= \frac{\partial^2 \xi_2}{\partial x_2^2}\frac{\partial u}{\partial \xi_2}+\frac{\partial \xi_2}{\partial x_2}\left[\frac{\partial}{\partial \xi_2}\left(\frac{\partial u}{\partial \xi_2}\right)\underbrace{\frac{\partial \xi_1}{\partial x_2}}_{=0}+\frac{\partial}{\partial \xi_2}\left(\frac{\partial u}{\partial \xi_2}\right)\frac{\partial \xi_2}{\partial x_2}\right] \\
&= \frac{\partial^2 \xi_2}{\partial x_2^2}\frac{\partial u}{\partial \xi_2}+\left(\frac{\partial \xi_2}{\partial x_2}\right)^2\frac{\partial^2 u}{\partial \xi_2^2}
\end{aligned}$$

After some grouping and simplification, the transformed equation becomes

$$\frac{\partial^2 u}{\partial \xi_1^2}+2\frac{\partial \xi_2}{\partial x_1}\frac{\partial u}{\partial \xi_1 \partial \xi_2}+\left(\frac{\partial^2 \xi_2}{\partial x_1^2}+\frac{\partial^2 \xi_2}{\partial x_2^2}\right)\frac{\partial u}{\partial \xi_2}+\left(\left(\frac{\partial \xi_2}{\partial x_1}\right)^2+\left(\frac{\partial \xi_2}{\partial x_2}\right)^2\right)\frac{\partial^2 u}{\partial \xi_2^2}=0 \qquad (\xi_1,\xi_2)\in E$$

subject to the boundary conditions

$$\begin{array}{ll} u(1,\xi_2)=0 & u(-1,\xi_2)=0 \\ u(\xi_1,1)=1 & u(\xi_1,-1)=0 \end{array}.$$

Note that the Jacobian terms are stochastic processes that are equivalent to random, spatially varying coefficients. The problem with these coefficients is that we do not know the derivatives of the inverse mapping. We only have the mapping defined on gridpoints. Therefore we need to compute the derivatives of the mapping at the grid points and express the derivatives of the inverse mapping in terms of the derivatives of the mapping. This gets messy – hang on!

# 7 The Jacobian Terms - Part II

We know from above that

$$\frac{\partial \xi_2}{\partial x_1}=-\frac{\frac{\partial x_2}{\partial \xi_1}}{\frac{\partial x_2}{\partial \xi_2}} \qquad \frac{\partial \xi_2}{\partial x_2}=\frac{1}{\frac{\partial x_2}{\partial \xi_2}}$$

Next we apply the chain rule, product rule, and quotient rule to get the second derivatives of the mappings.

$$\begin{aligned}
\frac{\partial}{\partial x_2}\left(\frac{\partial \xi_2}{\partial x_2}\right) &= \frac{1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[0-\frac{\partial}{\partial x_2}\left(\frac{\partial x_2}{\partial \xi_2}\right)\right] \\
&= \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[\frac{\partial}{\partial \xi_1}\left(\frac{\partial x_2}{\partial \xi_2}\right)\underbrace{\frac{\partial \xi_1}{\partial x_2}}_{=0}+\frac{\partial}{\partial \xi_2}\left(\frac{\partial x_2}{\partial \xi_2}\right)\frac{\partial \xi_2}{\partial x_2}\right] \\
&= \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[\frac{\partial}{\partial \xi_2}\left(\frac{\partial x_2}{\partial \xi_2}\right)\frac{1}{\frac{\partial x_2}{\partial \xi_2}}\right] \\
&= \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^3}\frac{\partial^2 x_2}{\partial \xi_2^2}
\end{aligned}$$

And

$$\begin{aligned}
\frac{\partial}{\partial x_1}\left(\frac{\partial \xi_2}{\partial x_1}\right) &= \frac{\partial}{\partial x_1}\left(\frac{-\frac{\partial x_2}{\partial \xi_1}}{\frac{\partial x_2}{\partial \xi_2}}\right) \\
&= \frac{1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[\frac{\partial x_2}{\partial \xi_2}\left[\frac{\partial}{\partial x_1}\left(-\frac{\partial x_1}{\partial \xi_1}\right)\right] - \left(-\frac{\partial x_2}{\partial \xi_1}\right)\left[\frac{\partial}{\partial x_1}\left(\frac{\partial x_2}{\partial \xi_2}\right)\right]\right] \\
&= \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[\frac{\partial x_2}{\partial \xi_2}\left[\frac{\partial^2 x_2}{\partial \xi_1^2}\underbrace{\frac{\partial \xi_1}{\partial x_1}}_{=1} + \frac{\partial x_2}{\partial \xi_1 \partial \xi_2}\frac{\partial \xi_2}{\partial x_1}\right] - \frac{\partial x_2}{\partial \xi_1}\left[\frac{\partial x_2}{\partial \xi_1 \partial \xi_2}\underbrace{\frac{\partial \xi_1}{\partial x_1}}_{=1} + \frac{\partial^2 x_2}{\partial \xi_2^2}\frac{\partial \xi_2}{\partial x_1}\right]\right] \\
&= \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\left[\frac{\partial x_2}{\partial \xi_2}\frac{\partial^2 x_2}{\partial \xi_1^2} + \frac{\partial x_2}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_1 \partial \xi_2}\left(\frac{-\frac{\partial x_2}{\partial \xi_1}}{\frac{\partial x_2}{\partial \xi_2}}\right) - \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_1 \partial \xi_2} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial^2 x_2}{\partial \xi_2^2}\left(\frac{-\frac{\partial x_2}{\partial \xi_1}}{\frac{\partial x_2}{\partial \xi_2}}\right)\right] \\
&= \frac{-1}{\frac{\partial x_2}{\partial \xi_2}}\frac{\partial^2 x_2}{\partial \xi_1^2} + \frac{2}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^2}\frac{\partial x_2}{\partial \xi_1 \partial \xi_2}\frac{\partial x_2}{\partial \xi_1} - \frac{-1}{\left(\frac{\partial x_2}{\partial \xi_2}\right)^3}\frac{\partial^2 x_2}{\partial \xi_2^2}\left(\frac{\partial x_2}{\partial \xi_1}\right)^2
\end{aligned}$$

Thus we have everything in terms of derivatives of $x_2(\xi_1, \xi_2)$. And by computing the derivatives of $x_2(\xi_1, \xi_2)$ at the grid points using a Chebyshev pseudospectral method, we can get the values of the derivatives of the inverse mapping at the grid points.

# 8 SBVP with Monte Carlo

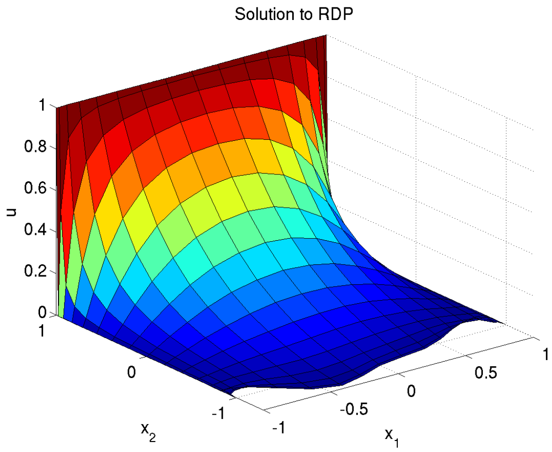
One way to compute the statistics of the solution of the SBVP is a Monte Carlo approach. Recall that

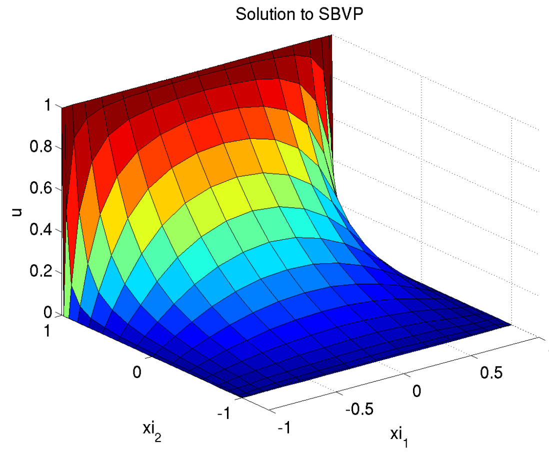
$$x_2 = \xi_2 + \sum_{k=1}^{K} \hat{x}_{2,k} Y_k. \tag{1}$$

The Monte Carlo approach involves the following steps.

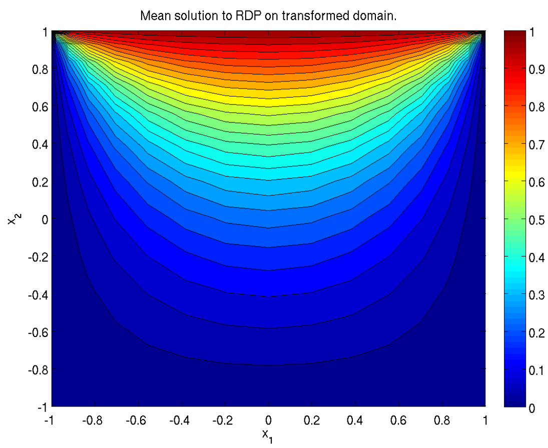
1. Generate $M$ independent samples of $\{Y_k\}$.
2. For each $j = 1, \ldots, M$, solve the Laplace equation for the mapping $\mathbf{x}(\xi_1, \xi_2, \omega_j)$ and construct a realization of the random domain.
3. For each $j = 1, \ldots, M$, compute the Jacobian terms of the mapping and solve the stochastic problem via some standard Laplace solver.
4. Compute the quantities of interest such as the mean and variance of $u$ from each of the $M$ realizations of the solution.

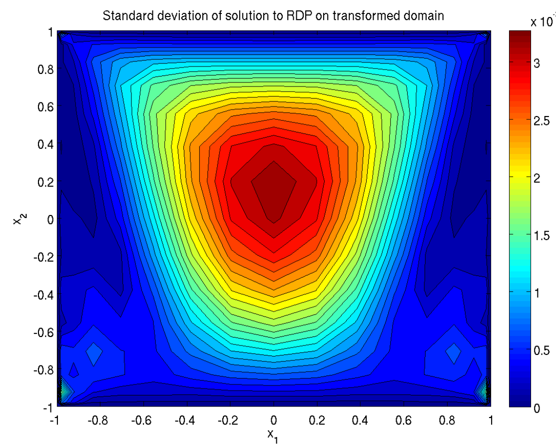
This procedure can be quite expensive and convergence may require large $M$ depending on the complexity of the problem. To give a feel for the type of computation involved, here is a plot of the solution for one realization of the random domain and its mapped counterpart on the deterministic domain.




Here are the moments of the solution computed with 10,000 samples.




# 9 SBVP with a stochastic Galerkin scheme

Another method for computing the statistics of the solution to this problem is a stochastic Galerkin technique. This approach involves representing the random response $u$ as a truncated spectral expansion in a basis of orthogonal polynomials that take a set of i.i.d. random variables as arguments, i.e.

$$u(\xi_1, \xi_2, \omega) \approx \sum_{l=0}^{L} u_l(\xi_1, \xi_2) \Psi_l(\omega).$$

This is often referred to as the generalized polynomial chaos expansion [3]. Since the randomness in our system arises from random variables with a uniform distribution, we can achieve exponential convergence of $u_l$ by choosing $\{\Psi_l\}$ to be the Legendre polynomials that take a set of i.i.d. uniform random variables as arguments. After substituting this representation into the stochastic differential equation, we can take a Galerkin projection to produce a coupled system of differential equations to solve for $\{u_l\}$. (For more details on this interesting method, see [2] and [3].) However, to perform the projection, we need to express the random inputs – in this case the variable coefficients of the transformed equations – as chaos expansions as well. Note that 1 is the PCE of the mapping $x_2$ with PCE coefficients greater than $K$ equal to 0. Therefore we can extend this sum out to $M$ terms corresponding to the order of polynomials that we choose as the basis for the PCE.

$$x_2 = \xi_2 + \sum_{k=1}^{M} \hat{x}_{2,k}\Psi_k.$$

where

$$\hat{x}_{2,k} = 0 \quad \text{for } k > K$$

Then

$$\frac{\partial x_2}{\partial \xi_1} = \sum_{k=1}^{M} \frac{\partial \hat{x}_{2,k}}{\partial \xi_1}\Psi_k \qquad \frac{\partial x_2}{\partial \xi_2} = 1 + \sum_{k=1}^{M} \frac{\partial \hat{x}_{2,k}}{\partial \xi_2}\Psi_k$$

$$\frac{\partial^2 x_2}{\partial \xi_1^2} = \sum_{k=1}^{M} \frac{\partial^2 \hat{x}_{2,k}}{\partial \xi_1^2}\Psi_k \qquad \frac{\partial^2 x_2}{\partial \xi_2^2} = \sum_{k=1}^{M} \frac{\partial^2 \hat{x}_{2,k}}{\partial \xi_2^2}\Psi_k \qquad \frac{\partial x_2}{\partial \xi_1 \partial \xi_2} = \sum_{k=1}^{M} \frac{\partial \hat{x}_{2,k}}{\partial \xi_1 \partial \xi_2}\Psi_k.$$

To compute the derivatives of the $x_2(\xi_1, \xi_2)$ mapping, we apply the appropriate Chebyshev differentiation matrices to the discrete representation of the mapping. So if $\mathbf{D}$ is an $N+1 \times N+!$ differentiation matrix and $\mathbf{I}$ is an $N+1 \times N+1$ identity matrix, then

$$\frac{\partial \hat{\mathbf{x}}_{2,k}}{\partial \xi_1} = (\mathbf{D} \otimes \mathbf{I})\hat{\mathbf{x}}_{2,k} \qquad \frac{\partial \hat{\mathbf{x}}_{2,k}}{\partial \xi_2} = (\mathbf{I} \otimes \mathbf{D})\hat{\mathbf{x}}_{2,k}$$

$$\frac{\partial^2 \hat{\mathbf{x}}_{2,k}}{\partial \xi_1^2} = (\mathbf{D}^2 \otimes \mathbf{I})\hat{\mathbf{x}}_{2,k} \qquad \frac{\partial^2 \hat{\mathbf{x}}_{2,k}}{\partial \xi_2^2} = (\mathbf{I} \otimes \mathbf{D}^2)\hat{\mathbf{x}}_{2,k} \qquad \frac{\partial \hat{\mathbf{x}}_{2,k}}{\partial \xi_1 \partial \xi_2} = (\mathbf{D} \otimes \mathbf{D})\hat{\mathbf{x}}_{2,k}$$

But what we actually need in order to perform the Galerkin projection on the transformed equation is the polynomial chaos expansion of the inverse mapping quantities. First let's consider the inverse mapping $\frac{\partial \xi_2}{\partial x_2} = 1/\frac{\partial x_2}{\partial \xi_2}$. For each $(i,j)$ point in the collocation grid, we can compute the PCE coefficients $\frac{\partial \hat{\xi}_{2,k}^{(i,j)}}{\partial x_2}$, $k = 0 \ldots M$ by solving the system

$$c_r = \sum_{p=0}^{M} \sum_{q=0}^{M} \frac{\langle \Psi_p \Psi_q \Psi_r \rangle}{\langle \Psi_r^2 \rangle} \frac{\partial \hat{\mathbf{x}}_{2,p}^{(i,j)}}{\partial \xi_2} \frac{\partial \hat{\boldsymbol{\xi}}_{2,q}^{(i,j)}}{\partial x_2} \qquad r = 0 \ldots M$$

where $c_r = 1$ when $r = 0$ and $c_r = 0$ otherwise. More generally, we can define product and quotient operations to get the PCE coefficients of products and quotients of the quantities defined on the grids. The product operation for a grid-valued quantity $\sum_{k=0}^{M} \mathbf{f}_k \Psi_k$ with another grid-valued quantity $\sum_{k=0}^{M} \mathbf{g}_k \Psi_k$ is

$$(\mathbf{fg})_r^{(i,j)} = \sum_{p=0}^{M} \sum_{q=0}^{M} \frac{\langle \Psi_p \Psi_q \Psi_r \rangle}{\langle \Psi_r^2 \rangle} \mathbf{f}_p^{(i,j)} \mathbf{g}_q^{(i,j)} \qquad r = 0 \ldots M$$

The quotient operation is defined as

$$\mathbf{f}_r^{(i,j)} = \sum_{p=0}^{M} \sum_{q=0}^{M} \frac{\langle \Psi_p \Psi_q \Psi_r \rangle}{\langle \Psi_r^2 \rangle} \mathbf{g}_q^{(i,j)} \left(\frac{\mathbf{f}}{\mathbf{g}}\right)_p^{(i,j)} \qquad r = 0 \ldots M$$

where the goal is to solve the resulting linear system for $\left(\frac{\mathbf{f}}{\mathbf{g}}\right)_p^{(i,j)}$. With these operations defined, we can compute the PCE coefficients of the inverse mappings at each grid point. So for example, the quantity

$$\frac{\partial \boldsymbol{\xi}_2}{\partial x_1} = -\frac{\frac{\partial \mathbf{x}_2}{\partial \xi_1}}{\frac{\partial \mathbf{x}_2}{\partial \xi_2}}$$

uses one quotient operation. And the quantity

$$\frac{\partial \boldsymbol{\xi}_2}{\partial x_2} = \frac{\mathbf{1}}{\frac{\partial \mathbf{x}_2}{\partial \xi_2}}$$

uses one quotient operation against a grid of ones to get its PCE coefficients. Squaring each of these quantities uses one product operation each. The PCE coefficients of the second derivatives of the inverse mapping are naturally more complex, but are reduced to combinations of the product and quotient operations. Thus

$$\frac{\partial^2 \boldsymbol{\xi}_2}{\partial x_2^2} = \frac{-1}{\left(\frac{\partial \mathbf{x}_2}{\partial \xi_2}\right)^3} \frac{\partial^2 \mathbf{x}_2}{\partial \xi_2^2}$$

uses two products to get the cubed term, followed by a quotient to get the inverse, followed by a product with the second derivative. The PCE of the final mapping quantity $\frac{\partial^2 \boldsymbol{\xi}_2}{\partial x_1^2}$ is computed with similar operations. Finally we have PCE coefficients for the mapping terms in the transformed equations and we can perform the projection.

# 10 The Projection

Number the terms in the transformed equation as follows.

$$\underbrace{\frac{\partial^2 u}{\partial \xi_1^2}}_{(1)} + \underbrace{2\frac{\partial \xi_2}{\partial x_1}\frac{\partial u}{\partial \xi_1 \partial \xi_2}}_{(2)} + \underbrace{\left(\frac{\partial^2 \xi_2}{\partial x_1^2} + \frac{\partial^2 \xi_2}{\partial x_2^2}\right)\frac{\partial u}{\partial \xi_2}}_{(3)} + \underbrace{\left(\left(\frac{\partial \xi_2}{\partial x_1}\right)^2 + \left(\frac{\partial \xi_2}{\partial x_2}\right)^2\right)\frac{\partial^2 u}{\partial \xi_2^2}}_{(4)} = 0$$

Define the $M+1 \times M+1$ diagonal matrix

$$\Sigma = \mathrm{diag}(\langle \Psi_i^2 \rangle)$$

And define the $M+1 \times M+1 \times M+1$ three dimensional tensor $\Gamma$

$$\Gamma(i,j,k) = \langle \Psi_i \Psi_j \Psi_k \rangle$$

By $\Gamma(:,:,k)$, we mean the $M+1 \times M+1$ matrix defined as above. This matrix is very sparse for the Legendre polynomials, but for generality's sake, we consider working with the full matrix.

For notation's sake, define the PCE of the coefficients of the transformed equation – computed above – as

$$\frac{\partial \xi_2}{\partial x_1} = \sum_{k=0}^{M} \mathbf{T}_k^{(2)} \Psi_k$$

$$\left(\frac{\partial^2 \xi_2}{\partial x_1^2} + \frac{\partial^2 \xi_2}{\partial x_2^2}\right) = \sum_{k=0}^{M} \mathbf{T}_k^{(3)} \Psi_k$$

$$\left(\left(\frac{\partial \xi_2}{\partial x_1}\right)^2 + \left(\frac{\partial \xi_2}{\partial x_2}\right)^2\right) = \sum_{k=0}^{M} \mathbf{T}_k^{(4)} \Psi_k$$

where $\mathbf{T}_k^{(2)}$, $\mathbf{T}_k^{(3)}$, and $\mathbf{T}_k^{(4)}$ are defined on the grid points. To create the matrix operators, we reshape each of these into an $(N+1)^2 \times (N+1)^2$ diagonal matrix.

$$\tilde{\mathbf{T}}_k^{(i)} = \mathrm{diag}(\mathrm{reshape}(\mathbf{T}_k^{(i)}, (N+1)^2, 1))$$

Finally, let $\tilde{\mathbf{I}}$ be a $M+1 \times M+1$ identity matrix.

Then after the spatial discretization with the Chebyshev collocation method and the stochastic Galerkin projection, the following terms become

$$(1) \qquad \Sigma \otimes (\mathbf{D}^2 \otimes \mathbf{I}) \equiv \mathbf{B}_1$$

$$(2) \qquad \left( \sum_{k=0}^{M} \Gamma(:,:,k) \otimes \tilde{\mathbf{T}}_k^{(2)} \right) \left( \tilde{\mathbf{I}} \otimes (\mathbf{D} \otimes \mathbf{D}) \right) \equiv \mathbf{B}_2$$

$$(3) \qquad \left( \sum_{k=0}^{M} \Gamma(:,:,k) \otimes \tilde{\mathbf{T}}_k^{(3)} \right) \left( \tilde{\mathbf{I}} \otimes (\mathbf{I} \otimes \mathbf{D}) \right) \equiv \mathbf{B}_3$$

$$(4) \qquad \left( \sum_{k=0}^{M} \Gamma(:,:,k) \otimes \tilde{\mathbf{T}}_k^{(4)} \right) \left( \tilde{\mathbf{I}} \otimes (\mathbf{I} \otimes \mathbf{D}^2) \right) \equiv \mathbf{B}_3$$

Finally, the matrix operator becomes

$$\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3 + \mathbf{B}_4$$

We need to enforce the boundary conditions for the PCE coefficients of $u$ corresponding to the boundary points – the leading coefficients should equal the prescribed boundary value and the remaining coefficients should be zero. To do this, we use the same approach as above for enforcing the boundaries of the mapping equations, i.e. replace the rows of $\mathbf{B}$ corresponding to the PCE coefficients of $u$ at the boundary points by a row of zeros with a 1 in the diagonal position. Call this modified matrix $\tilde{\mathbf{B}}$. Then we modify the corresponding entry in the right hand side to be the value prescribed at the boundary. In this case, the points along one boundary are equal to 1 and the rest are 0. Call this right hand side $\mathbf{b}$.

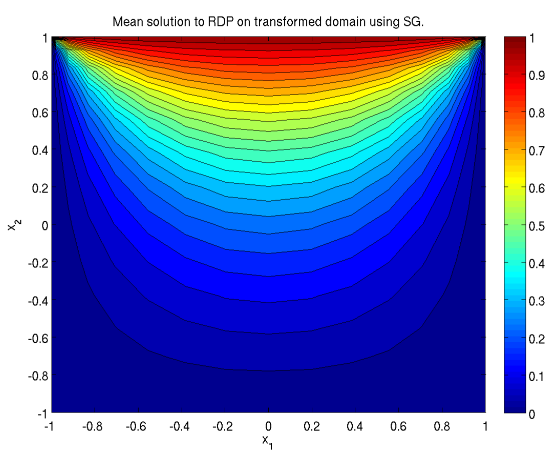
Now we can solve

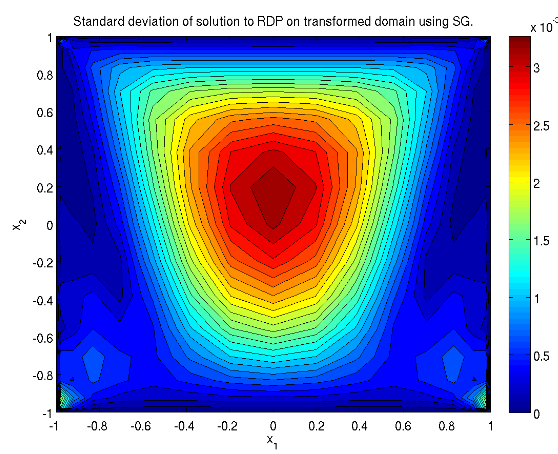
$$\tilde{\mathbf{B}}\mathbf{u} = \mathbf{b}$$

where $\mathbf{u}$ is a vector of length $(M+1)(N+1)^2$ and

$$\mathbf{u} = [\mathbf{u}^{(0)} \ldots \mathbf{u}^{(M)}]^T$$

are the PCE coefficients of $u$ at the grid points. Here are the moments of the solution computed with 6 random variables and polynomials of order 2.




Visually, these results compare very well with the results from the Monte Carlo method.

# 11 Conclusions, Remarks, and Future Work

In summary, this method employs a particular, meaningful representation of a random domain to model rough, uncertain geometries. It constructs a mapping from the random domain to the deterministic domain and solves the mapped problem by generally accepted methods for stochastic boundary value problems. It then computes statistics of the solution in order to quantify the uncertainty in the solution introduced by the randomness in the domain.

It is worth noting that the error in this method is not well known. Some of the assumptions in the randomness of the problem - such as independence of $\{Y_k\}$ - are not completely justified, and the effect on the solution is not well characterized. The other primary issue with this method is the well-posedness of the random domain problem. With sufficient variation, some problems may not be well-posed and regularity conditions may break down.

Nonetheless, I find this method very appealing and intend to pursue its applications further.

# 12 References


1. D. Xiu and D. Tartakovsky, *Numerical methods for differential equations in random domain*, SIAM J. Sci. Comput.,Vol. 28, No. 3, pp. 11671185, 2006.

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

3. D. Xiu and G. E. Karniadakis, *The Wiener-Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619644.

4. M. Loeve, *Probability Theory*, 4th ed., Springer-Verlag, New York, Berlin, 1977.