Solving the Stochastic Steady-State Diffusion Problem Using Multigrid

Tengfei Su
tengfesu@math.umd.edu
Applied Mathematics and Scientific Computing Program

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

October 15, 2015

Abstract

In this project we use multigrid method to solve the stochastic steady-state diffusion problem. We follow the stochastic finite element formulation for the discretization of the problem, and apply a multigrid algorithm to solve the linear system. Implementing the multigrid method to generate low-rank approximate solutions is studied as the second part of the project.

Contents

1 Background **1**

2 Stochastic finite element method **2**
2.1 Weak form 2
2.2 Karhunen-Loève expansion 3
2.3 Finite-element form 4
2.4 Polynomial chaos 4
2.5 Matrix form 5

3 Multigrid **6**
3.1 Grid transfer operators 6
3.2 Smoothing 7
3.3 Two-gird correction scheme 8

4 Numerical experiments **8**
4.1 Implementation 8
4.2 Model problem 8
4.3 Validation 9

5 Low-rank approximate solutions **9**

6 Schedule **10**

7 Deliverables **11**

8 Bibliography **11**

1 Background

Many physical problems can be modeled by partial differential equations (PDEs). A simple example is the steady-state diffusion equation

$$-\nabla \cdot (c(x)\nabla u(x)) = f(x),\ x \in D. \tag{1.1}$$

When given such an equation, we are actually assuming that we know the explicit forms of the diffusion coefficient $c(x)$ and the source term $f(x)$. There terms are related to the material properties such as heat conductivity and material porosity, and they may be subject to uncertainty. In this case, it's fitting to write them as random fields:

$$-\nabla \cdot (c(x,\omega)\nabla u(x,\omega)) = f(x,\omega),\ (x,\omega) \in D \times \Omega, \tag{1.2}$$

where $c(x,\omega)$ and $f(x,\omega)$ are functions not only in the spatial space D but also in the sample space Ω. The above equation is called a stochastic partial

differential equation (SPDE); it's more difficult to deal with than (1.1) due to the newly introduced stochastic part.

One of the popular ways to solve SPDEs is the Monte Carlo method (MCM). The basic idea is to run a large number of sampling, and for each realization, to solve a deterministic subproblem. Then the results are collected to compute the statistical properties of the solution. The number of realization can be large, especially when we have a problem with large variance.

Alternatively, we can use the stochastic finite element method (SFEM) pioneered by Ghanem & Spanos (2003). The spatial domain is discretized similar to a traditional finite element method, and the sample space is discretized as well. From SFEM we will have only one single linear system to solve, but of course, the size of the system will be much larger than that in the subproblem of MCM. One of the most important issues about SFEM is how to solve the linear system efficiently to reduce computational cost.

The goal of this project is to solve the stochastic steady-state diffusion equation efficiently. We are going to follow the SFEM formulation for handling the stochastic part, and use a multigrid method to solve the resulting linear system. In the next sections, we will talk about the two parts with more details.

2 Stochastic finite element method

Consider the stochastic steady-state diffusion equation

$$\begin{cases} -\nabla \cdot (c(x,\omega)\nabla u(x,\omega)) = f(x) & \text{in } D \times \Omega \\ u(x,\omega) = 0 & \text{on } \partial D \times \Omega \end{cases} \tag{2.1}$$

where the stochastic coefficient $c(x,\omega) : D \times \Omega \to \mathbb{R}$. The sample space belongs to a probability space $(\Omega, \mathscr{F}, P)$ with probability measure P. Note that we are considering the case where the source term f is deterministic. The solution of equation (2.1) will be a random field $u(x,\omega) : D \times \Omega \to \mathbb{R}$.

2.1 Weak form

The weak form of (2.1) is given as follows: find $u(x,\omega) \in H = H_0^1(D) \otimes L^2(\Omega)$ satisfying

$$\int_\Omega \int_D c(x,\omega)\nabla u(x,\omega)\nabla v(x,\omega)\mathrm{d}x\mathrm{d}P = \int_\Omega \int_D f(x)v(x,\omega)\mathrm{d}x\mathrm{d}P \tag{2.2}$$

for $\forall v \in H$. Note that we have a double integral with respect to Lebesgue measure and probability measure. The Hilbert space H is defined as

$$H = \{v(x,\omega) : D \times \Omega \to \mathbb{R} | \|v\|_H < \infty, v|_{\partial D \times \Omega} = 0\}$$

with energy norm

$$\|v(x,\omega)\|_H^2 = \int_\Omega \int_D c(x,\omega)|\nabla v(x,\omega)|^2 \mathrm{d}x\mathrm{d}P.$$

Introducing the truncated Karhunen-Loève expansion (Powell & Elman, 2009), which is discussed in Section 2.2, we are able to represent $c(x,\omega)$ by a finite collection of random variables $\{\xi_k\}_{k=1}^m : \Omega \to \mathbb{R}$. Then correspondingly the solution will also have finite stochastic dimension. The weak form (2.2) can be rewritten as: find $u(x,\omega) \in V = H_0^1(D) \otimes L^2(\Gamma)$ such that

$$\int_\Gamma \rho(\xi) \int_D c(x,\xi)\nabla u(x,\xi)\nabla v(x,\xi)\mathrm{d}x\mathrm{d}\xi = \int_\Gamma \rho(\xi) \int_D f(x)v(x,\xi)\mathrm{d}x\mathrm{d}\xi \quad (2.3)$$

for $\forall v \in V$. Here $\rho(\xi)$ is the joint density function, and Γ is the joint image of $\{\xi_k\}_{k=1}^m$. The Hilbert space V is defined as

$$V = \{v(x,\xi) : D \times \Gamma \to \mathbb{R} | \|v\|_V < \infty, v|_{\partial D \times \Gamma} = 0\}$$

with energy norm

$$\|v(x,\xi)\|_V^2 = \int_\Gamma \rho(\xi) \int_D c(x,\xi)|\nabla v(x,\xi)|^2 \mathrm{d}x\mathrm{d}\xi.$$

2.2 Karhunen-Loève expansion

Given a random field $c(x,\omega)$ with mean function $c_0(x)$, constant variance ν, and continuous covariance function $r(x,y)$, it admits an orthogonal decomposition, or Karhunen-Loève (KL) expansion

$$c(x,\omega) = c_0(x) + \sum_{k=1}^{\infty} \sqrt{\lambda_k} c_k(x)\xi_k(\omega) \quad (2.4)$$

where $\{\xi_k\}$ is a sequence of uncorrelated random variables with mean zero and variance ν that we assume to be identically distributed. $(\lambda_k, c_k(x))$ can be computed from the eigenvalue equation

$$\int_D \frac{r(x,y)}{\nu} c_k(x)\mathrm{d}x = \lambda_k c_k(y). \quad (2.5)$$

In the previous section we employed the truncated KL expansion

$$c(x,\omega) \approx c_0(x) + \sum_{k=1}^{m} \sqrt{\lambda_k} c_k(x)\xi_k(\omega) \quad (2.6)$$

to obtain an approximation of the stochastic coefficient $c(x,\omega)$ and represented it by m identically distributed random variables.

2.3 Finite-element form

To obtain the finite-element form for (2.3), we need to find a finite-dimensional subspace of V. In SFEM, the spatial space $H_0^1(D)$ and stochastic space $L^2(\Gamma)$ are discretized separately. For the spatial domain, we use the piecewise linear/bilinear basis functions (depending on whether triangles or quadrilaterals are used for the triangulation). The subspace is given as

$$S = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\} \subset H_0^1(D).$$

Here N is the number of interior nodes of the mesh. The subspace of $L^2(\Gamma)$ can be written in a similar manner:

$$T = \text{span}\{\psi_1(\xi), \ldots, \psi_M(\xi)\} \subset L^2(\Gamma)$$

where the basis functions $\{\psi_r(\xi)\}_{r=1}^M$ are the m-dimensional "polynomial chaos" of total order p (Xiu & Karniadakis, 2002). Details are given in Section 2.4.

Now we have a subspace of V defined as

$$V^h = S \otimes T = \text{span}\{\phi(x)\psi(\xi), \phi \in S, \psi \in T\}. \tag{2.7}$$

This gives us the SFEM formulation: find $u_{hp} \in V^h$, satisfying

$$\int_\Gamma \rho(\xi) \int_D c(x,\xi)\nabla u_{hp}(x,\xi)\nabla v(x,\xi)\mathrm{d}x\mathrm{d}\xi = \int_\Gamma \rho(\xi) \int_D f(x)v(x,\xi)\mathrm{d}x\mathrm{d}\xi \tag{2.8}$$

for $\forall v \in V^h$. Here the discrete solution is a linear combination of basis functions

$$u_{hp}(x,\xi) = \sum_{j=1}^{N}\sum_{s=1}^{M} u_{jl}\phi_j(x)\psi_s(\xi), \tag{2.9}$$

test function can be taken as

$$v(x,\xi) = \phi_i(x)\psi_r(\xi),\ i = 1, \ldots, N, r = 1, \ldots, M, \tag{2.10}$$

and the stochastic coefficient is represented by the KL expansion (2.6)

$$c(x,\omega) = c_0(x) + \sum_{k=1}^{m} \sqrt{\lambda_k} c_k(x)\xi_k(\omega).$$

2.4 Polynomial chaos

In general, a second order (with finite variance) random process admits a polynomial expansion

$$X(\omega) = \sum_{j=0}^{\infty} a_j \psi_j(\xi(\omega)) \tag{2.11}$$

where ξ a multi-dimensional random variable, and polynomials $\{\psi_j\}$ form a complete orthogonal basis in the L^2 space, satisfying

$$\int \psi_i(\xi)\psi_j(\xi)\rho(\xi)\mathrm{d}\xi = \delta_{ij}\int \psi_i(\xi)^2\rho(\xi)\mathrm{d}\xi. \tag{2.12}$$

To construct the subspace T, we restrict the total order of polynomials in (2.11) to p. Take ξ as an m-dimensional random variable, $\{\psi_r(\xi)\}_{r=1}^M$ as normalized order-p polynomials from the expansion, such that

$$\int \psi_i(\xi)\psi_j(\xi)\rho(\xi)\mathrm{d}\xi = \delta_{ij}. \tag{2.13}$$

The dimension of T is

$$M = \frac{(m+p)!}{m!p!}. \tag{2.14}$$

Note that for different distributions of ξ we have different forms of polynomials so that they satisfy the orthogonality relationship with respect to the density function. If ξ is Gaussian, then ψ will be Hermite polynomials; if ξ is uniformly distributed, ψ will take the form of Legendre polynomials.

2.5 Matrix form

Substituting (2.6), (2.9), and (2.10) into (2.8), we finally get the linear system to solve: find $\mathbf{u} \in \mathbb{R}^{MN}$, such that

$$A\mathbf{u} = \mathbf{f}, \tag{2.15}$$

where

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MM} \end{pmatrix}, \mathbf{f} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_M \end{pmatrix},$$

and

$$\mathbf{u} = [u_{11}, \ldots, u_{N1}, \ldots, u_{1M}, \ldots, u_{NM}]^{\mathrm{T}},$$

$$[\mathbf{f}_r]_i = \int_\Gamma \rho(\xi)\int_D f(x)\phi_i(x)\psi_r(\xi)\mathrm{d}x\mathrm{d}\xi.$$

The matrix blocks are given by

$$A_{rs} = K_0\int_\Gamma \psi_r(\xi)\psi_s(\xi)\rho(\xi)\mathrm{d}\xi + \sum_{k=1}^m K_k\int_\Gamma \xi_k\psi_r(\xi)\psi_s(\xi)\rho(\xi)\mathrm{d}\xi,$$

$$K_0(i,j) = \int_D c_0(x)\nabla\phi_i(x)\nabla\phi_j(x)\mathrm{d}x,$$

$$K_k(i,j) = \int_D \sqrt{\lambda_k} c_k(x) \nabla\phi_i(x) \nabla\phi_j(x) \mathrm{d}x,\ k = 1, \ldots, M.$$

Using tensor product notation (Powell & Elman, 2009), we have

$$A = G_0 \otimes K_0 + \sum_{k=1}^{m} G_k \otimes K_k, \tag{2.16}$$

where the stochastic matrices G_k are defined as

$$G_0(r,s) = \int_\Gamma \psi_r(\xi)\psi_s(\xi)\rho(\xi)\mathrm{d}\xi,$$

$$G_k(r,s) = \int_\Gamma \xi_k\psi_r(\xi)\psi_s(\xi)\rho(\xi)\mathrm{d}\xi,\ k = 1, \ldots, M.$$

The matrix A in (2.11) is symmetric and positive definite if the problem is well-posed. Also, the choice of basis functions ensures that A is sparse. In general, A is a large matrix of size $MN \times MN$. For instance, in 2-D, if $h = 2^{-7}, m = p = 4$, then $MN \sim 1,000,000$. In the next section, we talk about multigrid method for solving the linear system.

3 Multigrid

Multigrid method has been successfully used in solving large sparse systems that arise from deterministic problems. Here we use a two-grid correction scheme (Elman & Furnival, 2007) to show how it's applied to the stochastic problem. Multigrid can be given by applying the two-grid procedure recursively.

3.1 Grid transfer operators

The first ingredient of the scheme is the two grid spaces and grid transfer operators. The fine grid space is defined as

$$V^h = S^h \otimes T^p, \tag{3.1}$$

and the coarse grid space is

$$V^{2h} = S^{2h} \otimes T^p, \tag{3.2}$$

where h is the mesh size. Note that the mesh parameter varies from h to $2h$, but the polynomial order p, which corresponds to the subspace T, is held constant. The linear system is represented as $\bar{A}\bar{\mathbf{u}} = \bar{\mathbf{f}}$ on the coarse grid.

Now we can define the prolongation operator that maps a function in V^{2h} to V^h. Any basis function $\phi_j^{2h} \in S^{2h}$ can be written as

$$\phi_j^{2h} = \sum_{i=1}^{N_h} p_{ij}\phi_i^h,\ j = 1, \ldots, N_{2h}. \tag{3.3}$$

Define matrix P such that $P_{ij} = p_{ij}$. Now for $v_{2h} \in V^{2h}$,

$$\begin{aligned} v_{2h} &= \sum_{j=1}^{N_{2h}} \sum_{k=1}^{M} v_{jk}^{2h} \phi_j^{2h} \psi_k = \sum_{j=1}^{N_{2h}} \sum_{k=1}^{M} v_{jk}^{2h} \sum_{i=1}^{N_h} p_{ij} \phi_i^h \psi_k \\ &= \sum_{i=1}^{N_h} \sum_{k=1}^{M} \left(\sum_{j=1}^{N_{2h}} p_{ij} v_{jk}^{2h} \right) \phi_i^h \psi_k = \sum_{i=1}^{N_h} \sum_{k=1}^{M} [P \mathbf{v}_k^{2h}]_i \phi_i^h \psi_k. \end{aligned} \tag{3.4}$$

As $v_{2h} \in V^h$, we also have

$$v_{2h} = \sum_{i=1}^{N_h} \sum_{k=1}^{M} v_{ik}^h \phi_i^h \psi_k. \tag{3.5}$$

By comparing (3.4) and (3.5), we see

$$[P \mathbf{v}_k^{2h}]_i = v_{ik}^h \Rightarrow P \mathbf{v}_k^{2h} = \mathbf{v}_k^h.$$

Therefore, if $\mathbf{v}^{2h}$ is the coefficient vector of v_{2h} in V^{2h}, then the coefficient vector of v_{2h} in V^h is $\mathscr{P}\mathbf{v}^{2h}$, with the prolongation operator defined as

$$\mathscr{P} = I \otimes P. \tag{3.6}$$

The restriction operator maps a function in V^h to V^{2h}. It's defined as

$$\mathscr{R} = I \otimes R = I \otimes P^{\mathrm{T}}. \tag{3.7}$$

With these two operators we have following relationships

$$\bar{A} = \mathscr{R} A \mathscr{P}, \ \bar{\mathbf{f}} = \mathscr{R} \mathbf{f}. \tag{3.8}$$

3.2 Smoothing

Given the two grid spaces, we can write $V^h = \mathscr{P} V^{2h} + B^h$. B^h is called fine grid correction space, and it's the null space of the restriction operator. This in fact is an orthogonal decomposition of the space V^h (Briggs et al, 2000). For an initial guess $\mathbf{u}^{(0)}$, the error $\mathbf{e}^{(0)} = \mathbf{u} - \mathbf{u}^{(0)}$ can be written as

$$\mathbf{e}^{(0)} = \mathscr{P} \mathbf{e}_{V^{2h}}^{(0)} + \mathbf{e}_{B^h}^{(0)}. \tag{3.9}$$

The second ingredient of multigrid, called smoother, can greatly reduce the fine grid component of the error when applied to both sides of (3.9). This means that after the smoothing step, we can restrict our system to the coarse grid, and solve a smaller system without losing much information.

The smoother is given by a stationary iteration (Elman et al, 2014). If we have matrix splitting $A = Q - Z$, then $A\mathbf{u} = \mathbf{f}$, $Q\mathbf{u} = Z\mathbf{u} + \mathbf{f}$. This gives the iteration

$$\begin{aligned}\mathbf{u}^{(k+1)} &= Q^{-1}Z\mathbf{u}^{(k)} + Q^{-1}\mathbf{f} \\ &= Q^{-1}(Q - A)\mathbf{u}^{(k)} + Q^{-1}\mathbf{f} \\ &= (I - Q^{-1}A)\mathbf{u}^{(k)} + Q^{-1}\mathbf{f} \\ &= \mathbf{u}^{(k)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(k)}). \end{aligned} \tag{3.10}$$

The matrix $I - Q^{-1}A$ is the smoother. Note that in practice we are applying it to the residual $\mathbf{f} - A\mathbf{u}$.

3.3 Two-gird correction scheme

With the discussion above, now we have the two-gird correction scheme (with k steps of smoothing for each iteration):

Algorithm
Choose initial guess $\mathbf{u}^{(0)}$
for $i = 0$ until convergence
    for k steps
        $\mathbf{u}^{(i)} \leftarrow \mathbf{u}^{(i)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(i)})$ (smoothing)
    end
    $\bar{\mathbf{r}} = \mathscr{R}(\mathbf{f} - A\mathbf{u}^{(i)})$ (restrict residual)
    solve $\bar{A}\bar{\mathbf{e}} = \bar{\mathbf{r}}$
    $\mathbf{u}^{(i+1)} \leftarrow \mathbf{u}^{(i)} + \mathscr{P}\bar{\mathbf{e}}$ (prolong and update)
end

4 Numerical experiments

4.1 Implementation

All work will be done in MATLAB R2015a on a personal laptop (Macbook Air, 1.6 GHz Intel Core i5, 4 GB 1600 MHz DDR3). We will use the Incompressible Flow & Iterative Solver Software (IFISS) and the stochastic extension (S-IFISS) (Silvester et al) to produce the Galerkin system. We will write the multigrid algorithm for solving the stochastic problem. This will be the main part of the work.

4.2 Model problem

We consider the model problem (Elman & Furnival, 2007) with spatial domain $D = (-1, 1)^2$ and deterministic source term $f = 1$. The covariance

function of $c(x, \omega)$ is given by

$$r(x, y) = \nu \mathrm{e}^{-\frac{1}{b_1}|x_1 - y_1| - \frac{1}{b_2}|x_2 - y_2|}. \tag{4.1}$$

The constants b_1, b_2 are called the correction length and are related to the decay of $\{\lambda_k\}$. In the KL expansion

$$c(x, \omega) = c_0(x) + \sum_{k=1}^{m} \sqrt{\lambda_k} c_k(x) \xi_k(\omega),$$

$c_k(x)$ and λ_k are the eigenfunctions and eigenvalues of $r(x, y)$. We will consider two cases for the distribution of $\xi_k(\omega)$:

- Normal distribution with $c_0(x) = 1$ and $\nu = 0.01$. Basis functions of T will be m-dimensional Hermite polynomials;
- Uniform distribution on $(-1, 1)$ with $c_0(x) = 10$ and $\nu = 1/3$. Basis functions of T will be m-dimensional Legendre polynomials.

4.3 Validation

For validation of the code, we consider the following two aspects:

- Comparison with MCM. We will use the Monte Carlo method to solve the same model problem, and compute the solution properties such as mean and variance. These properties can also be obtained from the SFEM solutions. We will look at the mean and variance of the SFEM solutions, and they should approach what we get from MCM for the same mesh size h.
- Multigrid analysis. There is a "textbook" convergence rate for multigrid: it is independent of the mesh parameter h. For the stochastic version, it's also independent of the dimension m and polynomial order p. We can check that our multigrid algorithm has this property by varying the value of h, m, and p.

5 Low-rank approximate solutions

This part follows the idea of combining iterative methods for solving linear systems with low-rank approximation in Kressner and Tobler's paper (2011). Recall that from the SFEM we have a linear system

$$A\mathbf{u} = \mathbf{f}, \; A = G_0 \otimes K_0 + \sum_{k=1}^{m} G_k \otimes K_k.$$

This can be written into a matrix form

$$K_0 U G_0 + \sum_{k=1}^{m} K_k U G_k = F, \tag{5.1}$$

where

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NM} \end{pmatrix}, F = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1M} \\ f_{21} & f_{22} & \cdots & f_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N1} & f_{N2} & \cdots & f_{NM} \end{pmatrix}.$$

The solution is given as a matrix instead of a long vector. The computational cost can be further reduced if we can write U as

$$U \approx U_k = V_k W_k^{\mathrm{T}},\ V_k \in \mathbb{R}^{N\times k}, W_k \in \mathbb{R}^{M\times k}, k \ll N, M \tag{5.2}$$

and use iterative methods to solve the matrix version of the system. Three iterative methods have been implemented in Kressner and Toblers paper (2011): preconditioned Richardson, preconditioned conjugate gradient (CG), and preconditioned biconjugate gradient stablized method (BiCGstab). As an extension, we will apply the multigrid method to generate low-rank approximate solutions.

6 Schedule

Here is the timeline of the project:

- 10/15 Generate Galerkin system from IFISS/S-IFISS
- 10/22 Write the multigrid routine and implement for model problem
- 11/19 Validation with multigrid analysis and Monte Carlo
- 11/26 Prepare for mid-year presentation
- 01/25 Validation (if not finished yet)
- 02/08 Implement multigrid for low-rank approximate solutions
- 03/07 Implement BiCGstab for low-rank approximate solutions (if time permits)
- 04/04 Collect computational results
- 04/25 Prepare for final presentation

Some milestones:

1. By the end of the first semester, will at least finish writing the multigrid algorithm for solving the Galerkin system, and implementing it to solve the model problem. In a better case, will also finish the validation part, i.e. comparison with Monte Carlo method.

2. By the end of the second semester, will have the multigrid method for low-rank approximate solutions. If time permits, will implement BiCGstab method for comparison of efficiency.

7 Deliverables

Deliverables of the project will include:

1. Documented code. We will have a multigrid routine for

   - stochastic Galerkin systems,
   - low-rank approximate solutions.

2. Reports and presentations.

8 Bibliography

Briggs, W. L., Henson, V. E., & MaCormick, S. F. (2000) *A Multigrid Tutorial. Second Edition.* Philadelphia: SIAM.

Elman, H. & Furnival D. (2007) Solving the stochastic steady-state diffusion problem using multigrid. *IMA Journal of Numerical Analysis*, **27**, 675–688.

Elman, H., Silvester, D. & Wathen, A. (2014) *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics.* Oxford: Oxford University Press.

Ghanem, R. G. & Spanos, P. D. (2003) *Stochastic Finite Elements: A Spectral Approach.* New York: Dover Publications.

Kressner D. & Tobler C. (2011) Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal of Matrix Analysis and Applications*, **32.4**, 1288–1316.

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

Silvester, D., Elman, H. & Ramage, A. Incompressible Flow and Iterative Solver Software, http://www.maths.manchester.ac.uk/∼djs/ifiss.

Xiu, D. & Karniadakis G. M. (2003) Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, **187**, 137–167.