

Solving the Stochastic Steady-State Diffusion Problem Using Multigrid

Tengfei Su

Applied Mathematics and Scientific Computing Program

Advisor: Howard Elman
Department of Computer Science

May 5, 2016

- 1 Project Review
- 2 Multigrid
 - Two-Grid Method
 - Algorithm
 - Validation
- 3 Low-Rank Approximation
 - Motivation
 - Algorithm
 - Numerical Results
- 4 Conclusions

The stochastic steady-state diffusion equation

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

with stochastic coefficient $c(x, \omega) : D \times \Omega \rightarrow \mathbb{R}$.

- Approaches
 - Monte Carlo method (MCM)
 - Stochastic finite element method (SFEM)
[Ghanem & Spanos, 2003]
- Solver: Multigrid [Elman & Furnival, 2007]

Karhunen-Loève expansion [Loève, 1960]

$$c(x, \omega) \approx c_0(x) + \sum_{i=1}^m \sqrt{\lambda_i} c_i(x) \xi_i(\omega).$$

Weak form

$$\begin{aligned} & \int_{\Gamma} \rho(\xi) \int_D c(x, \xi) \nabla u(x, \xi) \cdot \nabla v(x, \xi) dx d\xi \\ &= \int_{\Gamma} \rho(\xi) \int_D f(x) v(x, \xi) dx d\xi, \end{aligned}$$

for $\forall v(x, \omega) \in V = H_0^1(D) \otimes L^2(\Gamma)$. Here $\rho(\xi)$ is the joint density function, and Γ is the joint image of $\{\xi_i\}_{i=1}^m$.

Finite-dimensional subspace

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

with basis functions

- $\phi(x)$: piecewise linear/bilinear basis functions
- $\psi(\xi)$: m -dimensional orthonormal polynomials, total order not exceeding p [Xiu & Karniadakis, 2003].

$$\int_{\Gamma} \psi_r(\xi) \psi_s(\xi) \rho(\xi) d\xi = \delta_{rs}, \quad M = \frac{(m+p)!}{m! p!}$$

SFEM solution

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

Galerkin System

Find $\mathbf{u} \in \mathbb{R}^{MN}$, such that [Powell & Elman, 2009]

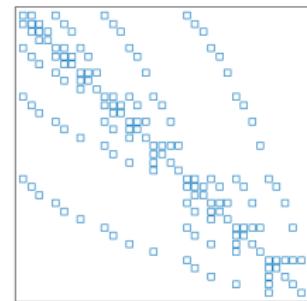
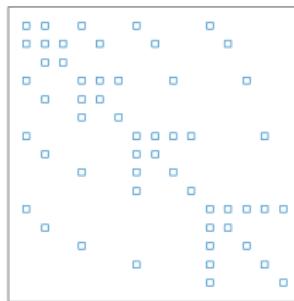
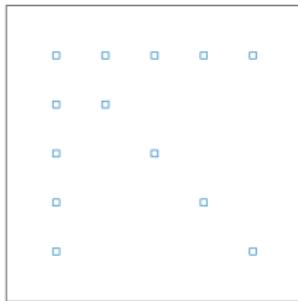
$$A\mathbf{u} = \mathbf{f},$$

where

$$\mathbf{u} = [u_{11}, u_{21}, \dots, u_{N1}, \dots, u_{1M}, u_{2M}, \dots, u_{NM}]^T,$$

$$A = G_0 \otimes K_0 + \sum_{i=1}^m G_i \otimes K_i, \quad \mathbf{f} = g_0 \otimes f_0.$$

Block structure of A ($m = 4, p = 1, 2, 3$)



1 Project Review

2 Multigrid

- Two-Grid Method
- Algorithm
- Validation

3 Low-Rank Approximation

- Motivation
- Algorithm
- Numerical Results

4 Conclusions

Two-Grid Method

- Two-grid correction scheme [Elman & Furnival, 2007]

Choose initial guess $\mathbf{u}^{(0)}$

for $i = 0$ until convergence

 for k steps

$$\mathbf{u}^{(i)} \leftarrow \mathbf{u}^{(i)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(i)})$$

 end

$$\bar{\mathbf{r}} = \mathcal{R}(\mathbf{f} - A\mathbf{u}^{(i)})$$

$$\text{solve } \bar{A}\bar{\mathbf{e}} = \bar{\mathbf{r}}$$

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathcal{P}\bar{\mathbf{e}}$$

 for k steps

$$\mathbf{u}^{(i+1)} \leftarrow \mathbf{u}^{(i+1)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(i+1)})$$

 end

end

- Multigrid: Apply the two-grid method recursively

Two-Grid Method

- Two grid spaces

$$V^{hp} = T^p \otimes S^h, \quad V^{2h,p} = T^p \otimes S^{2h}$$

- Prolongation and restriction operators

$$\mathcal{P} = I \otimes P, \quad \mathcal{R} = I \otimes P^T$$

- Construction of \bar{A}

$$\bar{A} = G_0 \otimes K_0^{2h} + \sum_{i=1}^m G_i \otimes K_i^{2h}$$

- Damped Jacobi smoother

$$Q = \frac{1}{\omega} D, \quad D = \text{diag}(A) = I \otimes \text{diag}(K_0)$$

Algorithm 1: Multigrid for stochastic Galerkin systems

```

initialization:  $i = 0, \mathbf{r}^{(0)} = \mathbf{f}, r_0 = \|\mathbf{f}\|$ 
while  $r > tol * r_0 \& i \leq maxit$  do /* solve residual equation */
     $\mathbf{e}^{(i)} = \text{MgIter}(A, \mathbf{0}, \mathbf{r}^{(i)}, level)$ 
     $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathbf{e}^{(i)}$ 
     $\mathbf{r}^{(i+1)} = \mathbf{f} - A\mathbf{u}^{(i+1)}$ 
     $r = \|\mathbf{r}^{(i+1)}\|, i = i + 1$ 

function  $\mathbf{u}^{(1)} = \text{MgIter}(A, \mathbf{u}^{(0)}, \mathbf{f}, level)$ 
    if  $level == 2$  then /* coarsest grid level */
         $\mathbf{u}^{(1)} = A \backslash \mathbf{f}$ 
    else /* apply one MG iteration */
        for  $k$  steps do
             $\mathbf{u}^{(0)} \leftarrow \mathbf{u}^{(0)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(0)})$ 
             $\bar{\mathbf{r}} = \mathcal{R}(\mathbf{f} - A\mathbf{u}^{(0)})$ 
             $\bar{\mathbf{e}} = \text{MgIter}(\bar{A}, \mathbf{0}, \bar{\mathbf{r}}, level - 1)$ 
             $\mathbf{u}^{(1)} = \mathbf{u}^{(0)} + \mathcal{P}\bar{\mathbf{e}}$ 
            for  $k$  steps do
                 $\mathbf{u}^{(1)} \leftarrow \mathbf{u}^{(1)} + Q^{-1}(\mathbf{f} - A\mathbf{u}^{(1)})$ 
    
```

- $D = (-1, 1)^2$, $f = 1$. Covariance function

$$r(x, y) = \sigma^2 \exp\left(-\frac{1}{b}|x_1 - y_1| - \frac{1}{b}|x_2 - y_2|\right)$$

- KL expansion

$$c(x, \omega) = c_0(x) + \sqrt{3} \sum_{i=1}^m \sqrt{\lambda_i} c_i(x) \xi_i(\omega)$$

- Uniform distribution (assuming independence)

$$\xi_i \sim U(-1, 1), \rho(\xi) = \frac{1}{2^m}$$

- Take $c_0(x) = 1, \sigma = 0.3, b = 2$, multigrid $tol = 10^{-6}$

Validation: Convergence Performance

- Independent of h

$$m = 5, p = 3$$

h	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
n	7	8	8	8	8	8

$$m = 3, p = 5$$

h	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
n	7	8	8	8	8	9

- Independent of m, p ($h = 2^{-3}$)

$(m = 3) \ p$	1	2	3	4	5	6
n	6	6	7	7	8	8

$(p = 3) \ m$	1	2	3	4	5	6
n	6	7	7	7	8	8

- Monte Carlo method

$$\xi_i \sim U(-1, 1), (K_0 + \sum_{i=1}^m \xi_i K_i) \mathbf{u} = f_0$$

- Compute mean and variance

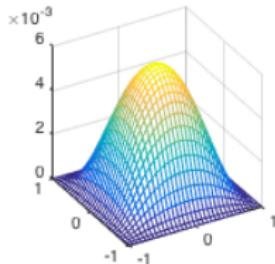
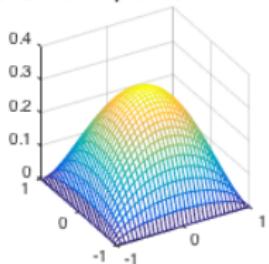
$$\mathbb{E}[u_{MC}] = \frac{1}{q} \sum_{r=1}^q u_{MC}^r$$

$$\text{Var}[u_{MC}] = \frac{1}{q-1} \sum_{r=1}^q (u_{MC}^r - \mathbb{E}[u_{MC}])^2$$

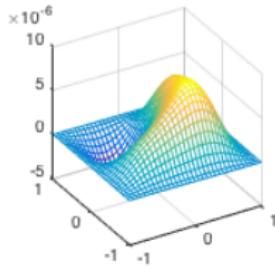
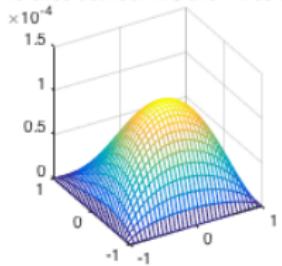
- Monte Carlo: $h = 2^{-4}, m = 3, q = 1,000,000$
Multigrid: $h = 2^{-4}, m = 3, p = 9, tol = 10^{-12}$

Validation: Monte Carlo

MC solution: expectation and variance



Difference between MC and FE solutions



$$\frac{\|\mathbb{E}[u_{FE}] - \mathbb{E}[u_{MC}]\|_2}{\|\mathbb{E}[u_{MC}]\|_2} = 3.35 \times 10^{-4}, \quad \frac{\|\text{Var}[u_{FE}] - \text{Var}[u_{MC}]\|_2}{\|\text{Var}[u_{MC}]\|_2} = 1.17 \times 10^{-3}$$

1 Project Review

2 Multigrid

- Two-Grid Method
- Algorithm
- Validation

3 Low-Rank Approximation

- Motivation
- Algorithm
- Numerical Results

4 Conclusions

Galerkin system

$$A\mathbf{u} = (G_0 \otimes K_0 + \sum_{i=1}^m G_i \otimes K_i)\mathbf{u} = \mathbf{f}.$$

SFEM solution

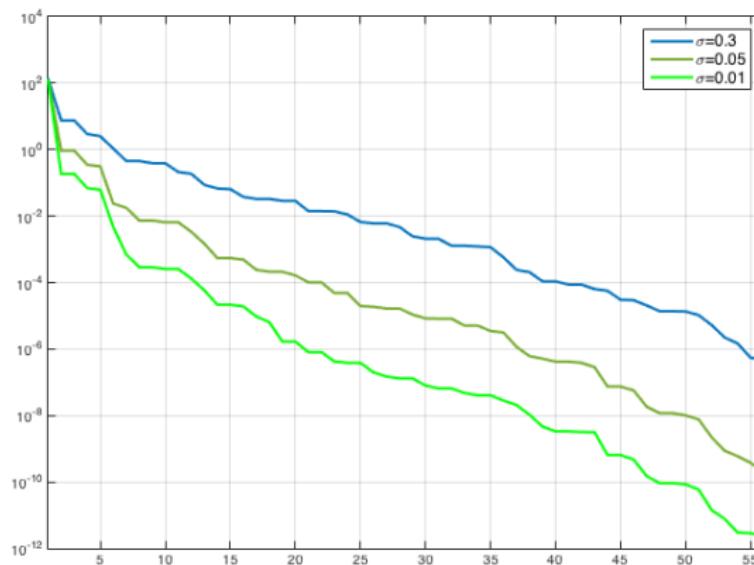
$$\mathbf{u} = [u_{11}, \dots, u_{N1}, \dots, u_{1M}, \dots, u_{NM}]^T$$

Write into matrix form

$$U = \text{mat}(\mathbf{u}) = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NM} \end{pmatrix}$$

$$\mathcal{A}(U) = K_0 U G_0^T + \sum_{i=1}^m K_i U G_i^T = F.$$

Decay of singular values for the solution matrix U
 $(h = 2^{-6}, m = 5, p = 3)$



- U can be well approximated by a low-rank matrix!

Low-rank approximation

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

$$(K_0 V_k)(G_0 W_k)^T + \sum_{i=1}^m (K_i V_k)(G_i W_k)^T = F_l F_r^T.$$

- Memory and computational cost

$$NM \text{ v.s. } k(N + M)$$

- Low-rank matrix iterates are convenient to implement

$$\begin{aligned} \mathcal{A}(VW^T) &= (K_0 V)(G_0 W)^T + \sum_{i=1}^m (K_i V)(G_i W)^T \\ &= [K_0 V, K_1 V, \dots, K_m V][G_0 W, G_1 W, \dots, G_m W]^T \end{aligned}$$

- Matrix rank grows rapidly: $k \rightarrow (m+1)k$. The iterates should be truncated in every iteration.

Algorithm 2: Multigrid with low-rank truncations

initialization: $i = 0, R^{(0)} = F$ in low-rank format, $r_0 = \|F\|$

while $r > tol * r_0 \text{ & } i \leq maxit$ **do** /* solve residual equation */

$E^{(i)} = \text{MgIter}(A, 0, R^{(i)}, level)$
 $U^{(i+1)} = U^{(i)} + E^{(i)}, \quad U^{(i+1)} = \mathcal{T}_{\text{abs}}(U^{(i+1)})$
 $R^{(i+1)} = F - \mathcal{A}(U^{(i+1)}), \quad R^{(i+1)} = \mathcal{T}_{\text{abs}}(R^{(i+1)})$
 $r = \|R^{(i+1)}\|, i = i + 1$

function $U^{(1)} = \text{MgIter}(A, U^{(0)}, F, level)$

if $level == 2$ **then** /* coarsest grid level */

solve $\mathcal{A}(U^{(1)}) = F$ directly

else /* apply one MG iteration */

for k steps **do**

$U^{(0)} \leftarrow U^{(0)} + \mathcal{S}(F - \mathcal{A}(U^{(0)})), \quad U^{(0)} = \mathcal{T}_{\text{rel}}(U^{(0)})$
 $\bar{R} = F - \mathcal{A}(U^{(0)}), \quad \bar{R} = \mathcal{T}_{\text{rel}}(\bar{R})$
 $\bar{R} = \mathcal{R}\bar{R}, \bar{E} = \text{MgIter}(\bar{A}, 0, \bar{R}, level - 1)$
 $U^{(1)} = U^{(0)} + \mathcal{P}\bar{E}$

for k steps **do**

$U^{(1)} \leftarrow U^{(1)} + \mathcal{S}(F - \mathcal{A}(U^{(1)})), \quad U^{(1)} = \mathcal{T}_{\text{rel}}(U^{(1)})$

The truncation operator $\tilde{U} = \mathcal{T}(U)$ [Kressner & Tobler, 2011]

- $U = VW^T, V \in \mathbb{R}^{N \times k}, W \in \mathbb{R}^{M \times k}$
- $\tilde{U} = \tilde{V}\tilde{W}^T, \tilde{V} \in \mathbb{R}^{N \times \tilde{k}}, \tilde{W} \in \mathbb{R}^{M \times \tilde{k}}$

Computation

- QR factorization: $V = Q_V R_V, W = Q_W R_W$ where $R_V, R_W \in \mathbb{R}^{k \times k}$
- SVD: $R_V R_W^T = \hat{V} \text{diag}(\sigma_1, \dots, \sigma_k) \hat{W}^T$
- Truncation

$$\sqrt{\sigma_{k+1}^2 + \dots + \sigma_k^2} \leq \epsilon_{\text{rel}} \sqrt{\sigma_1^2 + \dots + \sigma_k^2}$$

$$\tilde{k} = \max\{k \mid \sigma_k \geq \epsilon_{\text{abs}}\}$$

- $\tilde{V} = Q_V \hat{V}(:, 1 : \tilde{k}), \tilde{W} = Q_W \hat{V}(:, 1 : \tilde{k}) \text{diag}(\sigma_1, \dots, \sigma_{\tilde{k}})$
- Cost: $O((M + N + k)k^2)$

- $D = (-1, 1)^2, f = 1$. Covariance function

$$r(x, y) = \sigma^2 \exp\left(-\frac{1}{b}|x_1 - y_1| - \frac{1}{b}|x_2 - y_2|\right)$$

- KL expansion

$$c(x, \omega) = c_0(x) + \sqrt{3} \sum_{i=1}^m \sqrt{\lambda_i} c_k(x) \xi_i(\omega)$$

- Uniform distribution (assuming independence)

$$\xi_i \sim U(-1, 1), \rho(\xi) = \frac{1}{2^m}$$

- Take $c_0(x) = 1$. m and b are chosen such that

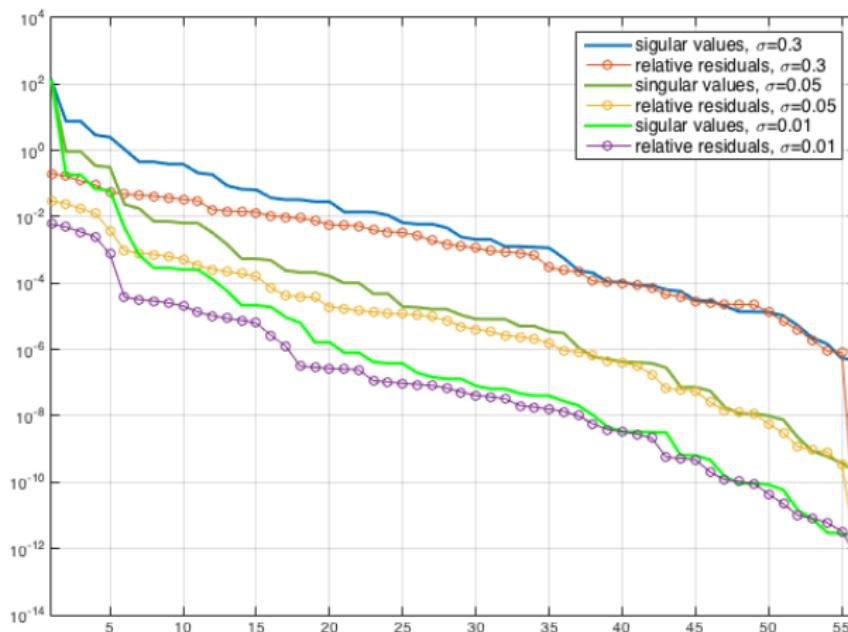
$$\sum_{i=1}^m \lambda_i / \sum_{i=1}^{\infty} \lambda_i \geq 95\%$$

- IFISS (Incompressible Flow and Iterative Solver Software, Silvester, et al) and SIFISS: Generate the Galerkin systems
- MATLAB R2015a
- 1.6 GHz Intel Core i5, 4 GB memory

Numerical Results

Case 1: Standard deviation σ

- Decay of singular values of solution matrix U
- Relative residuals for low-rank approximations



Numerical Results

Case 1: Standard deviation σ ($N = 16129, M = 56$)

	Truncation	10^{-6}	10^{-4}	No truncation	
$\sigma = 0.001$	Rank	6	2		
	Iterations	5	3	5	3
	Elapsed time	1.46	0.27	5.50	3.34
	Rel residual	3.0e-6	6.2e-4	1.2e-6	2.2e-4
$\sigma = 0.01$	Rank	13	6		
	Iterations	5	3	4	3
	Elapsed time	2.58	0.73	4.56	3.39
	Rel residual	1.1e-5	6.0e-4	1.6e-5	2.2e-4
$\sigma = 0.1$	Rank	40	14		
	Iterations	7	4	4	3
	Elapsed time	11.2	2.09	4.58	3.47
	Rel residual	1.7e-5	1.1e-3	1.9e-5	2.4e-4
$\sigma = 0.3$	Rank	55	35		
	Iterations	9	6	6	3
	Elapsed time	29.1	7.90	6.57	3.30
	Rel residual	1.8e-5	1.7e-3	1.4e-5	1.7e-3

Numerical Results

Case 2: Spatial dimension N ($\sigma = 0.01, M = 56$)

	Truncation	10^{-6}	10^{-4}	No truncation	
$nc = 7$ $N = 16129$	Rank	13	6		
	Iterations	5	3	4	3
	Elapsed time	2.58	0.73	4.56	3.39
	Rel residual	1.1e-5	6.0e-4	1.6e-5	2.2e-4
$nc = 8$ $N = 65025$	Rank	16	6		
	Iterations	5	3	4	3
	Elapsed time	13.20	3.14	18.87	14.08
	Rel residual	1.3e-5	5.8e-4	1.8e-5	2.3e-4
$nc = 9$ $N = 261121$	Rank	11	5		
	Iterations	5	3	4	2
	Elapsed time	34.20	12.54	80.89	39.64
	Rel residual	4.6e-5	1.7e-3	1.9e-5	3.3e-3
$nc = 10$ $N = 1046529$	Rank	11	2		
	Iterations	5	2	4	2
	Elapsed time	207.12	21.14	1144.10	572.71
	Rel residual	4.5e-5	6.7e-3	1.9e-5	3.3e-3

Numerical Results

Case 3: Stochastic dimension M ($\sigma = 0.01$, $N = 16129$, $p = 3$)

	Truncation	10^{-6}	10^{-4}	No truncation	
$m = 5$ $M = 56$	Rank	13	6		
	Iterations	5	3	4	3
	Elapsed time	2.58	0.73	4.56	3.39
	Rel residual	1.1e-5	6.0e-4	1.6e-5	2.2e-4
$m = 7$ $M = 84$	Rank	24	8		
	Iterations	6	4	5	3
	Elapsed time	7.56	1.84	14.84	8.77
	Rel residual	5.5e-6	4.6e-4	1.2e-6	2.2e-4
$m = 9$ $M = 220$	Rank	36	10		
	Iterations	7	4	5	3
	Elapsed time	16.93	4.88	31.90	19.04
	Rel residual	4.0e-6	1.7e-4	1.2e-6	2.2e-4
$m = 11$ $M = 364$	Rank	47	12		
	Iterations	7	5	5	4
	Elapsed time	31.86	7.21	61.40	49.86
	Rel residual	2.4e-6	7.5e-5	1.2e-6	1.6e-5

1 Project Review

2 Multigrid

- Two-Grid Method
- Algorithm
- Validation

3 Low-Rank Approximation

- Motivation
- Algorithm
- Numerical Results

4 Conclusions

Summary

- The multigrid solver works well for the stochastic Galerkin systems and can be more efficient than the Monte Carlo method.
- The solution matrix can be better approximated by a low-rank matrix when the standard deviation in the random coefficient is small.
- Low-rank truncations reduce computing time for multigrid, especially when the spatial degree of freedom is large.

Low-rank truncation based on singular values

- Cost: $O((N + M + k)k^2)$
- QR factorization can be expensive
($nc = 7, m = 5, p = 3$)

σ	0.01	0.1	0.3
QR	51.2%	67.7%	73.2%
SVD	3.4%	2.2%	2.2%

Better truncation strategy: take advantage of grid hierarchy?

- ① Ghanem, R. G. & Spanos, P. D. (2003). *Stochastic Finite Elements: A Spectral Approach*. New York: Dover Publications.
- ② Elman, H. & Furnival D. (2007). Solving the stochastic steady-state diffusion problem using multigrid. *IMA Journal of Numerical Analysis* 27, 675–688.
- ③ Loève, M. (1960). *Probability Theory*. New York: Van Nostrand.
- ④ Xiu, D. & Karniadakis G. M. (2003). Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187, 137–167.
- ⑤ Powell, C. & Elman H. (2009). Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis* 29, 350–375.
- ⑥ 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.
- ⑦ Silvester, D., Elman, H. & Ramage, A. Incompressible Flow and Iterative Solver Software, <https://www.cs.umd.edu/users/elman/ifiss3.4.html>

- Thank you!
- Questions?