Steady-state diffusion equation

$$-\nabla \cdot (c(x)\nabla u(x)) = f(x), \ x \in D$$
Background

- **Steady-state diffusion equation**
  \[-\nabla \cdot (c(x)\nabla u(x)) = f(x), \; x \in D\]

- **Differential coefficient and source term subject to uncertainty**
  (heat conductivity, material porosity)
  \[-\nabla \cdot (c(x, \omega)\nabla u(x, \omega)) = f(x, \omega), \; (x, \omega) \in D \times \Omega\]
Background

- **Steady-state diffusion equation**
  \[-\nabla \cdot (c(x)\nabla u(x)) = f(x), \; x \in D\]

- **differential coefficient and source term subject to uncertainty** (heat conductivity, material porosity)
  \[-\nabla \cdot (c(x, \omega)\nabla u(x, \omega)) = f(x, \omega), \; (x, \omega) \in D \times \Omega\]

- **Stochastic partial differential equations (SPDEs)**
Stochastic partial differential equations (SPDEs)

\[-\nabla \cdot (c(x, \omega) \nabla u(x, \omega)) = f(x, \omega), \quad (x, \omega) \in D \times \Omega\]

- Monte Carlo Method (MCM)
  - large numbers of sampling
  - deterministic subproblem
Background

Stochastic partial differential equations (SPDEs)

\[-\nabla \cdot (c(x, \omega) \nabla u(x, \omega)) = f(x, \omega), (x, \omega) \in D \times \Omega\]

- Monte Carlo Method (MCM)
  - large numbers of sampling
  - deterministic subproblem
- Stochastic Finite Element Method (SFEM)
  - discretization of sample space
  - solving large linear system
Stochastic partial differential equations (SPDEs)

\[-\nabla \cdot (c(x,\omega)\nabla u(x,\omega)) = f(x,\omega), (x,\omega) \in D \times \Omega\]

- Monte Carlo Method (MCM)
  - large numbers of sampling
  - deterministic subproblem
- Stochastic Finite Element Method (SFEM)
  - discretization of sample space
  - solving large linear system

Goal: Solving stochastic PDEs efficiently

- SFEM formulation
- Multigrid
Boundary value problem

The stochastic steady-state diffusion equation

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

where

- stochastic coefficient \( c(x, \omega) : D \times \Omega \rightarrow \mathbb{R} \)
- probability space \( (\Omega, \mathcal{F}, P) \)
- random field \( u(x, \omega) : D \times \Omega \rightarrow \mathbb{R} \)
Boundary value problem

The stochastic steady-state diffusion equation

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

where

- stochastic coefficient \( c(x, \omega) : D \times \Omega \rightarrow \mathbb{R} \)
- probability space \((\Omega, \mathcal{F}, P)\)
- random field \( u(x, \omega) : D \times \Omega \rightarrow \mathbb{R} \)

We seek a weak solution \( u(x, \omega) \in H = H^1_0(D) \otimes L^2(\Omega) \) satisfying

\[
\int_{\Omega} \int_D c(x, \omega) \nabla u(x, \omega) \cdot \nabla v(x, \omega) dxdP = \int_{\Omega} \int_D f(x)v(x, \omega) dxdP
\]

for \( \forall v \in H \).
If \( c(x, \omega) \) has continuous covariance function \( r(x, y) \), then

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

The eigenpair \( (\lambda_k, c_k(x)) \) can be computed by

\[
\int_D \frac{r(x, y)}{\nu} c_k(x) dx = \lambda_k c_k(y).
\]
Karhunen-Loève expansion

If \( c(x, \omega) \) has continuous covariance function \( r(x, y) \), then

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

The eigenpair \( (\lambda_k, c_k(x)) \) can be computed by

\[
\int_D \frac{r(x, y)}{\nu} c_k(x) dx = \lambda_k c_k(y).
\]

Introducing KL expansion,

\[
\int_\Gamma p(\xi) \int_D c(x, \xi) \nabla u(x, \xi) \cdot \nabla v(x, \xi) dx d\xi = \int_\Gamma p(\xi) \int_D f(x) v(x, \xi) dx d\xi
\]

where \( p(\xi) \) is the joint density function, \( \Gamma \) is the joint image of \( \{\xi_k\}_{k=1}^{m} \).
Need a finite-dimensional subspace:

\[ S = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\} \subset H^1_0(D) \]

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

\[ V^h = S \otimes T = \text{span}\{\phi(x)\psi(\xi), \phi \in S, \psi \in T\} \]
SFEM formulation

Need a finite-dimensional subspace:

\[ S = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\} \subset H^1_0(D) \]
\[ T = \text{span}\{\psi_1(\xi), \ldots, \psi_M(\xi)\} \subset L^2(\Gamma) \]
\[ V^h = S \otimes T = \text{span}\{\phi(x)\psi(\xi), \phi \in S, \psi \in T\} \]

- \( \phi(x) \) - piecewise linear/bilinear basis function
SFEM formulation

Need a finite-dimensional subspace:

\[ S = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\} \subset H^1_0(D) \]

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

\[ \mathcal{V}^h = S \otimes T = \text{span}\{\phi(x)\psi(\xi), \phi \in S, \psi \in T\} \]

- \( \phi(x) \) - piecewise linear/bilinear basis function
- \( \psi(\xi) \) - \( m \)-dimensional “polynomial chaos” of order \( p \)
  - orthogonality relationship

\[ \int \psi_i(\xi)\psi_j(\xi)p(\xi)d\xi = \delta_{ij} \]

- dimension of subspace \( T \)

\[ M = \frac{(m + p)!}{m!p!} \]
Find $u_{hp} \in V^h$, satisfying

$$
\int_{\Gamma} p(\xi) \int_{D} c(x, \xi) \nabla u_{hp}(x, \xi) \cdot \nabla v(x, \xi) dx d\xi = \int_{\Gamma} p(\xi) \int_{D} f(x) v(x, \xi) dx d\xi
$$

for $\forall v \in V^h$.

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

$$
v(x, \xi) = \phi_i(x) \psi_r(\xi), \ i = 1 : N, \ r = 1 : M$$

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

Find \( u \in \mathbb{R}^{MN} \), such that

\[
Au = f,
\]

where

\[
A = \begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1M} \\
A_{21} & A_{22} & \cdots & A_{2M} \\
\vdots & \vdots & \ddots & \vdots \\
A_{M1} & A_{M2} & \cdots & A_{MM}
\end{pmatrix},
\]

\[
f = \begin{pmatrix}
f_1 \\
f_2 \\
\vdots \\
f_M
\end{pmatrix},
\]

and

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

\[
[f_r]_i = \int_\Gamma p(\xi) \int_D f(x) \phi_i(x) \psi_r(\xi) dx d\xi
\]
Matrix formulation

The matrix block

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

\[
K_0(i, j) = \int_{D} c_0(x) \nabla \phi_i(x) \nabla \phi_j(x) dx,
\]

\[
K_k(i, j) = \int_{D} \sqrt{\lambda_k} c_k(x) \nabla \phi_i(x) \nabla \phi_j(x) dx.
\]
Matrix formulation

The matrix block

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

\[ K_0(i,j) = \int_D c_0(x)\nabla \phi_i(x)\nabla \phi_j(x)dx, \]

\[ K_k(i,j) = \int_D \sqrt{\lambda_k} c_k(x)\nabla \phi_i(x)\nabla \phi_j(x)dx. \]

In tensor product notation,

\[ A = G_0 \otimes K_0 + \sum_{k=1}^m G_k \otimes K_k. \]
Multigrid

Solving linear system $Au = f$, where

- $A$ is symmetric and positive definite
- the choice of basis functions ensures that $A$ is sparse
- $\text{size}(A) = MN \times MN$

\[ N \sim \frac{1}{h^2}, \quad M = \frac{(m + p)!}{m!p!} \]

(for $h = 2^{-7}$, $m = p = 4$, $MN \sim 1,000,000$)
Multigrid

Solving linear system $Au = f$, where

- $A$ is symmetric and positive definite
- the choice of basis functions ensures that $A$ is sparse
- size($A$) = $MN \times MN$

$$N \sim \frac{1}{h^2}, \quad M = \frac{(m + p)!}{m!p!}$$

(for $h = 2^{-7}, m = p = 4, MN \sim 1,000,000$)

Multigrid method has been successfully used in solving large sparse systems that arise from deterministic problems.
Two-grid correction scheme

- **fine grid space**
  \[ V^h = S^h \otimes T, \quad A u = f \]

- **coarse grid space**
  \[ V^{2h} = S^{2h} \otimes T, \quad \bar{A} \bar{u} = \bar{f} \]

- **fine grid correction space**
  \[ V^h = V^{2h} + B^h \]
Two-grid correction scheme

- fine grid space
  \[ V^h = S^h \otimes T, \quad Au = f \]
- coarse grid space
  \[ V^{2h} = S^{2h} \otimes T, \quad \bar{A}\bar{u} = \bar{f} \]
- fine grid correction space
  \[ V^h = V^{2h} + B^h \]

- prolongation operator
  \[ l_{2h}^h : V^{2h} \to V^h \]
- restriction operator
  \[ l_{2h}^h : V^h \to V^{2h} \]
Grid transfer operators

- If $\mathbf{v}^{2h}$ is the coefficient vector of $\nu_{2h}$ in $V^{2h}$, then the coefficient vector of $\nu_{2h}$ in $V^h$ is $\mathcal{P}\mathbf{v}^{2h}$. Prolongation matrix:

  $$\mathcal{P} = I \otimes P.$$  

- Restriction matrix is defined as

  $$\mathcal{R} = I \otimes R = I \otimes P^T.$$
Grid transfer operators

- 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 $P \mathbf{v}^{2h}$. Prolongation matrix:
  
  $$P = I \otimes P.$$ 

- Restriction matrix is defined as
  
  $$R = I \otimes R = I \otimes P^T.$$ 

- Relations for matrix $A$ and right-hand side $f$
  
  $$\tilde{A} = RAP, \quad \tilde{f} = Rf$$
Two-grid correction scheme

- Smoother - reduce the fine grid component of the error

\[ u - u^{(0)} = e^{(0)} = Pe^{(0)}_{V^{2h}} + e^{(0)}_{B^{h}} \]

\[ e^{(k)} = (I - Q^{-1}A)^{k}e^{(0)} = Pe^{(k)}_{V^{2h}} + e^{(k)}_{B^{h}} \]
Two-grid correction scheme

- Smoother - reduce the fine grid component of the error

\[ u - u^{(0)} = e^{(0)} = Pe_{V^{2h}} + e^{(0)}_{B^h} \]

\[ e^{(k)} = (I - Q^{-1}A)^k e^{(0)} = Pe^{(k)}_{V^{2h}} + e^{(k)}_{B^h} \]

- Stationary iteration

\[ A = Q - Z, Au = f \implies Qu = Zu + f \]

\[ u^{k+1} = Q^{-1}Zu^{(k)} + Q^{-1}f \]
\[ = Q^{-1}(Q - A)u^{(k)} + Q^{-1}f \]
\[ = (I - Q^{-1}A)u^{(k)} + Q^{-1}f \]
Two-grid correction scheme

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

- For multigrid, apply the above scheme recursively
Implementation

- Platform: Macbook Air, 1.6 GHz Intel Core i5, 4 GB 1600 MHz DDR3
- Language: MATLAB R2015a
- IFISS & S-IFISS for producing Galerkin system
  (Incompressible Flow & Iterative Solver Software)
- Multigrid routine
  - grid transfer operators
  - smoother
  - iterative method
Model problem: \( D = (-1, 1)^2, f = 1 \). Covariance function

\[
r(x, y) = \nu \exp\left(-\frac{1}{b} |x_1 - y_1| - \frac{1}{b} |x_2 - y_2| \right)
\]
Model problem: $D = (-1, 1)^2$, $f = 1$. Covariance function

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

KL expansion

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

Normal distribution

$$\Omega = \mathbb{R}^m, \ p(\xi) = \frac{1}{(2\pi\nu)^{m/2}} e^{-\frac{\xi^2}{\nu^2}}, \text{Hermite polynomials}$$

Uniform distribution

$$\Omega = (-1, 1)^m, \ p(\xi) = \frac{1}{2^m}, \text{Legendre polynomials}$$
Validation

- Comparison with Monte Carlo
- Multigrid analysis
  - textbook convergence rate, independent of $h$
  - independent of $m, p$
Galerkin solution

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

\[ U = \begin{pmatrix}
    u_{11} & u_{12} & \cdots & u_{1M} \\
    u_{21} & u_{22} & \cdots & u_{2M} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{N1} & u_{N2} & \cdots & u_{NM}
\end{pmatrix} \]
Extension

- **Galerkin solution**

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

\[ U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NM} \end{pmatrix} \]

- **Matrix form**

\[ Au = f, \quad A = G_0 \otimes K_0 + \sum_{k=1}^{m} G_k \otimes K_k \]

\[ \Rightarrow K_0 UG_0 + \sum_{k=1}^{m} K_k UG_k = F \]
Seek low-rank approximation

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

which significantly reduce the computational cost.
Seek low-rank approximation

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

which significantly reduce the computational cost.

- Compute \( U_k \) with iterative methods
- Richardson, conjugate gradient (CG), Biconjugate gradient stabilized method (BiCGstab)
- Multigrid
Schedule

- 10/15 Generate Galerkin system from IFISS/S-IFISS
  Write the multigrid routine
- 11/15 Validation with multigrid analysis and Monte Carlo
- 12/15 Mid-year presentation
- 02/16 Implement multigrid for low-rank approximate solutions
- 03/16 Implement BiCGstab for low-rank approximate solutions (if time permitting)
- 04/16 Collect computational results.
- 05/16 Final presentation
Deliverables

- Multigrid routine for Stochastic Galerkin system
- Multigrid routine for Low-rank approximation
- Documented code
- Reports and presentations
References


The End

- Thank you!
- Questions?