Stochastic Finite Element Methods (FEMs) for PDE Models with Uncertain Inputs

Catherine Powell

University of Manchester
c.powell@manchester.ac.uk

February 1, 2018
Many **physical processes** are modelled by PDE(s):

- fluid mechanics
- continuum mechanics
- thermodynamics
- etc...

**If** we know the **geometry, boundary conditions, coefficients, initial conditions etc** then we can compute often accurate approximations using finite element methods (**FEMs**).
Uncertainty Quantification (UQ)

Many **physical processes** are modelled by PDE(s):

- fluid mechanics
- continuum mechanics
- thermodynamics
- etc...

If we know the **geometry, boundary conditions, coefficients, initial conditions etc** then we can compute often accurate approximations using finite element methods (**FEMs**).

In many applications, we don’t know all the inputs. **One possibility** is to represent uncertain inputs as **random variables**.

- We have to solve **stochastic/parameter-dependent** PDEs.
- We need tailored numerical methods (**computer models**)
Types of Uncertainty

▷ **Model Uncertainty**
Uncertainty in: form of the model, scales of the physical process, missing physics etc.

▷ **Parameter/Input Uncertainty**
Uncertainty in: coefficients, material parameters, boundary conditions, initial conditions, geometry etc.

▷ **Numerical Uncertainty**
Uncertainty (error) stemming from choice of discretisation, numerical approximation etc.
Forward UQ

Given a **probabilistic description** of the uncertain inputs

- mean, covariance
- distribution
- ...

compute an **approximation** to a quantity of interest (QoI) related to the solution $u$, such as

- $\mathbb{E}[u]$, $\mathbb{E}[f(u)]$, $\text{Var}(u)$
- $\mathbb{P}(f(u) > \text{tol})$
- ...

and an **estimate of the error/uncertainty** in that approximation.
Find \( u(x) \) (the velocity) and \( p(x) \) (the pressure) such that,

\[
- \nu \nabla^2 u + u \cdot \nabla u + \nabla p = f \quad \text{in } D, \\
\nabla \cdot u = 0 \quad \text{in } D, \\
uu u = g \quad \text{on } \partial D_D, \\
nu \frac{\partial u}{\partial n} - np = 0 \quad \text{on } \partial D_N.
\]

If the viscosity is uncertain, we can model it as a single random variable. For example, a Uniform random variable with

\[
E[\nu] = \mu, \quad \text{Var}[\nu] = \sigma^2.
\]
Example: Flow over a step

$\mu = 1/50$ and $\sigma = 1/500$. Streamlines of the mean flow (top) and mean pressure (bottom) computed with a stochastic (Galerkin) FEM.
**Example: Flow over a step**

**Variance** of the magnitude of velocity (top) and **variance** of the pressure (bottom) computed with a **stochastic (Galerkin)** FEM.
A simple model for steady-state fluid flow in a porous medium is:

\[-a \nabla p = u \quad \text{in} \ D,\]
\[\nabla \cdot u = f \quad \text{in} \ D,\]
\[p = g \quad \text{on} \ \partial D_D,\]
\[u \cdot n = 0 \quad \text{on} \ \partial D_N = \partial D \setminus \partial D_D.\]

Solution variables: \(p = p(x)\) (hydraulic head), \(u = u(x)\) (velocity field).

- \(a(x)\): permeability coefficient
- \(f(x)\): source or sink terms
- \(g(x)\): boundary data
- \(D \subset \mathbb{R}^d\): spatial/computational domain
Measurements $a(x_i)$ are known at 41 spatial locations $x_i$. 
Let $a(x)$ be a random variable at each point in space.

Given a \textbf{mean} $\mu(x)$ and \textbf{covariance} $C(x, x')$ for $a(x)$, we want to make statistical predictions about the pressure $p$ and velocity $u$. 
**Realisations** of the path travelled by a particle released into the flow at a fixed location (left). Monte Carlo estimate of the expected flow field (right).
A popular **black-box** strategy for performing forward UQ in PDE models is

- choose a statistical model for the uncertain input(s)
- use **random field generators** to draw samples of the input(s)
- use **FEM codes** to solve the PDEs for each sample
- average results to obtain estimates of the mean, variance etc.

Commonly cited benefits include:

- allows **reuse of existing FEM codes**
- ‘trivially’ parallelisable
- no complicated mathematics!
What’s wrong with Monte Carlo FEM?

The error in approximating the **expected** solution is:

\[
\text{Error} = \left\| \mathbb{E}[u] - \frac{1}{M} \sum_{i=1}^{M} u_h(x, y_i) \right\|
\]
What’s wrong with Monte Carlo FEM?

The error in approximating the expected solution is:

$$\text{Error} = \left\| \mathbb{E}[u] - \frac{1}{M} \sum_{i=1}^{M} u_h(x, y_i) \right\|$$

and breaks up into FEM discretisation and MC statistical errors:

$$\text{Error} = \left\| \mathbb{E}[u - u_h] + \left( \mathbb{E}[u_h] - \frac{1}{M} \sum_{i=1}^{M} u_h(x, y_i) \right) \right\|$$

$$\leq \underbrace{\left\| \mathbb{E}[u - u_h] \right\|}_{\text{FEM error}} + \underbrace{\left\| \left( \mathbb{E}[u_h] - \frac{1}{M} \sum_{i=1}^{M} u_h(x, y_i) \right) \right\|}_{\text{MC sampling error}}$$

$$\approx O(h^\alpha) + O(M^{-1/2}).$$
To obtain $\text{Error} \leq \text{TOL}$,

we need to choose

$$h = O(TOL^{1/\alpha}), \quad M = O(TOL^{-2})$$
To obtain

\[ \text{Error} \leq TOL, \]

we need to choose

\[ h = O(TOL^{1/\alpha}), \quad M = O(TOL^{-2}) \]

Suppose an optimal solver for the FEM system is available. The cost to achieve \( \text{Error} \leq TOL \) is:

\[ \text{Cost} = M \times O(h^{-d}) = O(TOL^{-2-d/\alpha}) \]

Cost of FEM solve
To obtain 

$$\text{Error} \leq \text{TOL},$$

we need to choose

$$h = O(TOL^{1/\alpha}), \quad M = O(TOL^{-2})$$

Suppose an optimal solver for the FEM system is available. The cost to achieve \(\text{Error} \leq \text{TOL}\) is:

$$\text{Cost} = M \times O(h^{-d}) = O(TOL^{-2-d/\alpha})$$

Cost of FEM solve

So, even for a toy problem: \(-\nabla \cdot a \nabla u = f\) in 2d with linear FEM:

$$\text{Cost} = O(TOL^{-4}).$$
There are many alternative schemes that can exploit existing FEM codes.

▶ **Sampling methods**
- Multifidelity & Multilevel Monte Carlo methods.
- Stochastic collocation FEMs (SC-FEMs)
- Reduced basis FEMs (RB-FEMs)

▶ **Not sampling methods**
- Stochastic Galerkin FEMs (SG-FEMs)

**SC-FEMs** and **SG-FEMs** are based on *polynomial* approximation.

**RB-FEMs** can be combined with any sampling method.
We compute an approximation of the form

\[ u(x, y) \approx \sum_{j=1}^{N} u_j(x) \psi_j(y) \]

where \( \{\psi_1, \ldots, \psi_N\} \) are polynomials in the parameters \( y \) (Polynomial Chaos) by solving one linear system,

\[ Au = f \]

of dimension \( NN_h \times NN_h \).

- \( N_h \) is the size of the FEM problem
- \( N \) depends on no. of parameters and chosen polynomial degree
Example: Flow over a step
Estimating the vorticity

Components of the **vorticity** along the bottom channel wall, $\mathbb{E}[Re] \approx 296$. 

![Graphs of PC components](image-url)
Here, just one parameter (viscosity), \( \nu = \mu + \sigma y \). Quantity of interest:

\[
Q(\nu(y)) := \int_3^7 \omega(x, y) ds
\]

The standard deviation of \( Q \) increases as we increase \( \sigma \) and we need more polynomials to accurately compute the QoI.

<table>
<thead>
<tr>
<th>( \sigma = \mu/10 )</th>
<th>( N = 2 )</th>
<th>( N = 3 )</th>
<th>( N = 4 )</th>
<th>( N = 5 )</th>
<th>( N = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma = 2\mu/10 )</td>
<td>1.2639</td>
<td>1.2150</td>
<td>1.2139</td>
<td>1.2138</td>
<td>*</td>
</tr>
<tr>
<td>( \sigma = 3\mu/10 )</td>
<td>1.7752</td>
<td>1.6197</td>
<td>1.6180</td>
<td>1.6167</td>
<td>1.6162</td>
</tr>
<tr>
<td>( \sigma = 4\mu/10 )</td>
<td>2.1379</td>
<td>1.8133</td>
<td>1.8320</td>
<td>1.8287</td>
<td>1.8291</td>
</tr>
</tbody>
</table>
If using a sampling method, for each $\mathbf{y}$ of interest, if we use standard FEM (high fidelity methods),

$$ u(\mathbf{x}, \mathbf{y}) \approx u_h(\mathbf{x}, \mathbf{y}) $$

we have to solve a **sparse** linear system of $N_h$ equations

$$ A(\mathbf{y}) \begin{bmatrix} u(\mathbf{y}) \end{bmatrix} = \mathbf{f}. $$

$$ N_h \times N_h $$
If using a sampling method, for each $y$ of interest, if we use standard FEM (high fidelity methods),

$$u(x, y) \approx u_h(x, y)$$

we have to solve a \textbf{sparse} linear system of $N_h$ equations

$$A(y) \ u(y) = f.$$

If our UQ method (\textit{e.g.} Monte Carlo) uses $M$ samples, and we have an optimal \textbf{iterative} solver for each system, then the cost is

$$M \times O(N_h)$$

Live with this! Try to reduce this!
Instead, for each \( y \) of interest we now seek to approximate:

\[
 u_h(x, y) \approx u_R(x, y)
\]

where \( u_R(x, y) \) belongs to a **lower-dimensional space**.
Basic Idea: Reduced Problem

Instead, for each \( y \) of interest we now seek to approximate:

\[
u_h(x, y) \approx u_R(x, y)\]

where \( u_R(x, y) \) belongs to a lower-dimensional space.

- Run the expensive FEM code \( N_R \) times with \textit{specially chosen} sets of input parameters \( y_i \).
- Collect the solution vectors (\textit{snapshots}) into a matrix

\[
V = [u(y_1), u(y_2), \cdots, u(y_{N_R})].
\]

- Make the columns orthonormal to produce a new matrix \( Q \)

\[
V \rightarrow Q = [q_1, q_2, \cdots, q_{N_R}].
\]

which has size \( N_h \times N_R \) (tall and skinny).
For each new $y$ of interest, approximate $u(y) \approx Qu_R(y)$, where $u_R(y)$ is found by solving the reduced system

$$
\begin{pmatrix}
Q^\top A(y) Q
\end{pmatrix}
\begin{pmatrix}
N_R 	imes N_R
\end{pmatrix}

u_R(y) = Q^\top f
$$

with $u_R(y) \in \mathbb{R}^{N_R}$.

The reduced problem can be used as a surrogate.
Test problem 1: Groundwater Flow

High-Fidelity Mixed FEM: $N_h = 217,464$. 
Average time in seconds to assemble and solve the high fidelity FEM systems ($T$) and reduced basis FEM systems ($t$):

<table>
<thead>
<tr>
<th>$\epsilon_1$</th>
<th>$N_R$</th>
<th>$T$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-3</td>
<td>36</td>
<td>2.13e0</td>
<td>5.57e-4</td>
</tr>
<tr>
<td>1e-4</td>
<td>59</td>
<td>2.14e0</td>
<td>1.06e-3</td>
</tr>
<tr>
<td>1e-5</td>
<td>84</td>
<td>2.14e0</td>
<td>1.75e-3</td>
</tr>
<tr>
<td>1e-6</td>
<td>108</td>
<td>2.08e0</td>
<td>5.03e-3</td>
</tr>
</tbody>
</table>
**Total time** to compute mean and variance of FEM solution.

<table>
<thead>
<tr>
<th>$\epsilon = 10^{-5}$</th>
<th>RB-FEM</th>
<th>Standard FEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>$N_R$ offline</td>
<td>online</td>
</tr>
<tr>
<td>351</td>
<td>64</td>
<td>510.9</td>
</tr>
<tr>
<td>1471</td>
<td>84</td>
<td>795.0</td>
</tr>
<tr>
<td>5503</td>
<td>96</td>
<td>1,002.1</td>
</tr>
</tbody>
</table>

**Note:** In this problem, we have only 5 parameters and a nice structure. When offline times are taken into account, 10 times faster than standard FEM.
Now we have 12 parameters, and a more complicated PDE structure.

<table>
<thead>
<tr>
<th>$\epsilon = 10^{-5}$</th>
<th>RB-FEM</th>
<th>Standard FEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>$N_R$</td>
<td>offline</td>
</tr>
<tr>
<td>2,649</td>
<td>52</td>
<td>1548.9</td>
</tr>
<tr>
<td>17,265</td>
<td>53</td>
<td>9,327.1</td>
</tr>
<tr>
<td>93,489</td>
<td>53</td>
<td><strong>53,685.8</strong></td>
</tr>
</tbody>
</table>

When offline times are taken into account, only twice as fast.
Applied mathematicians are mainly concerned with the accuracy of outputs of computer models with respect to the true solution of the continuous physical model.

Standard Monte Carlo is infeasible for computer models consisting of FEM discretizations for PDE models.

Stochastic Galerkin FEM (polynomial chaos type methods) may be appropriate if there are a modest no. of input parameters and the true solution is well behaved enough.

Building surrogate FEM models using reduced basis ideas might offer some savings if time can be invested in the offline stage.
Standard Reduced Basis Methods


Offline vs Online

RB methods decompose the computational work into **two stages**.

- **Offline**: The reduced basis matrix $Q$ is constructed, and any matrices & vectors that are too expensive to compute online.

  **Warning**: Offline costs may be **very** expensive!

- **Online**: All the reduced systems are solved,

  $$(Q^T A(y_i) Q) u_R(y_i) = f, \quad i = 1, \ldots, N,$$

  and quantities of interest computed.

**Key Philosophy**: Costs should be **independent** of $N_h$. 
Standard Greedy Method

Training set $\Theta_{\text{train}} \subset \Gamma$, error tolerance $\epsilon$, error estimator $\Delta_R(y)$. Initialise $Q = [u(y_0)]$ with some $y_0 \in \Gamma$.

\[\text{while } \max_{y \in \Theta_{\text{train}}} \Delta_R(y) > \epsilon \text{ do}\]

\[\begin{align*}
\text{Find } y_k &= \arg\max_{y \in \Theta_{\text{train}}} \Delta_R(y) \\
\text{Solve high-fidelity problem for } u(y_k) \\
\text{Set } Q &= [Q, u(y_k)] \text{ & Orthonormalise}
\end{align*}\]

end

If the error estimator satisfies

$$\|u_h(y) - u_R(y)\|_{V_h} \leq \Delta_R(y)$$

then we say the method is certified.