Forward UQ: How and Why to be Intrusive

Catherine E. Powell

University of Manchester

catherine.powell@manchester.ac.uk

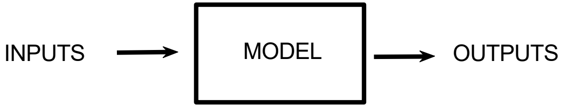
January 13, 2024

M\crNA

Part I. What is Forward UQ?

UQ is a (very!) **broad** phrase used to describe methods for taking account of **uncertainties** when **mathematical** & **computer models** are used to describe real-world phenomena and make predications.

Traditional **applied** mathematical modelling:



- ▷ Choose a model (ODEs, **PDEs**, NNs etc).
- ▷ Choose **inputs** for the model.
- ▷ Find/**approximate** the solution (outputs).

Forward UQ: Given **probability distribution for inputs**, approximate quantities of interest related to model solution.

Major 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 Error:** Uncertainty stemming from choice of discretization, numerical approximation etc.

PDEs + Uncertain Inputs

Example: Poroelasticity Model

▷**Applications:** geophysical flows, fluid flow in central nervous system.

$$-\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{f},$$
$$-\nabla \cdot \mathbf{u} - \lambda^{-1}(p_T - \alpha p_F) = 0$$
$$\lambda^{-1}(\alpha p_T - \alpha^2 p_F) - s_0 p_F + \nabla \cdot (\kappa \nabla p_F) = g,$$

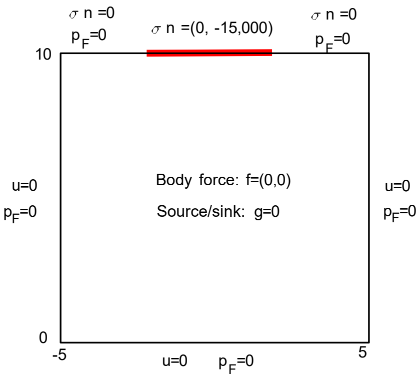
where the stress tensor is $\boldsymbol{\sigma} := 2\mu\boldsymbol{\epsilon}(\mathbf{u}) - p_T \mathbf{I}$ and

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

- **Solution fields**: $\mathbf{u}$ p_F, p_T (displacement, fluid pressure, total pressure)
- **Multiple physical parameters:** $\nu, \alpha, s_0, E, \kappa$

References: Lee, Mardal & Winther (2017), Oyarzúa & Ruiz–Baier (2016).

Simple Example: Footing Problem



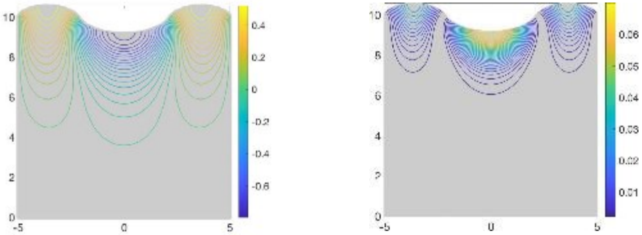
▷ **Uncertain Young's modulus & hydraulic conductivity:**

$$E = e_0 + e_1 y_1, \qquad \kappa = \kappa_0 + \kappa_1 y_2, \qquad y_1, y_2 \sim U(-1, 1)$$

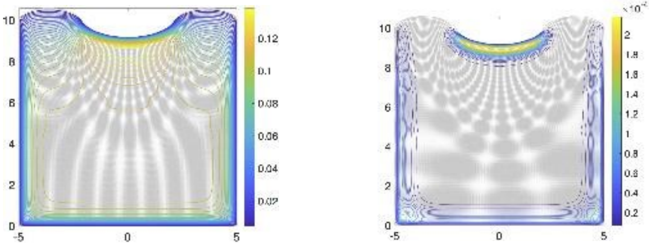
where $e_0 = 3 \times 10^4$, $e_1 = 0.5 \times e_0$, and $\kappa_0 = 10^{-4}$, $\kappa_1 = 0.5 \times \kappa_0$.

Mean (left) & Variance (right), $\nu = 0.4995$, $\tilde{s}_0 = 30$

Vertical displacement



Fluid Pressure



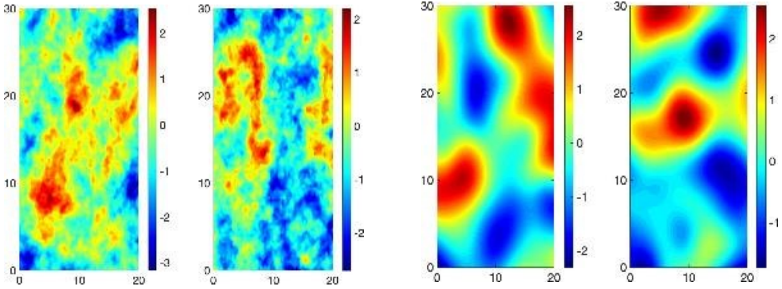
Reference: Khan & Powell, **SISC** (2021).

$$-\kappa \nabla p = \boldsymbol{u}, \qquad \nabla \cdot \boldsymbol{u} = f$$

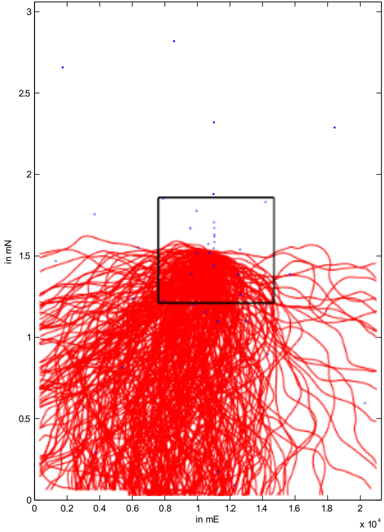
- **Solution fields:** **u**, p (velocity, fluid pressure)
- **Inputs:** $\kappa = \kappa(\mathbf{x})$ (a spatial field).

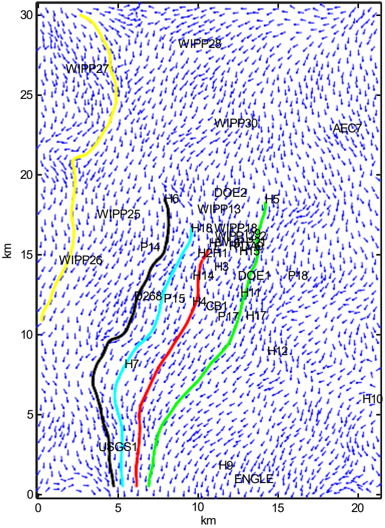
Uncertain **spatial** functions represented as **random fields** with prescribed **mean** $\mu(\boldsymbol{x})$ and **covariance functions:**

$$C(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^{2} \exp\left(-\frac{|x_i - x_i'|}{\ell_i}\right), \qquad C(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{\ell^2}\right).$$



Samples of the estimated path of a particle released into the flow (left). Estimated **mean** velocity field (right).





Part II: Non-intrusive Methods for Forward UQ

Aim: Propagate **uncertainty** from model inputs to outputs

$$\mathcal{L}_{\mathbf{x}}(\mathbf{y})u(\mathbf{x},\mathbf{y}) = f(\mathbf{x},\mathbf{y}), \qquad \mathbf{x} \in D \subset \mathbb{R}^{2,3}$$

- ☐ Represent uncertain inputs as **functions of random parameters $y \in \Gamma$**.
- ☐ Approximate QoIs related to solution (e.g., $\mathbb{E}[u]$, $\mathrm{Var}[u]$, $\mathbb{P}(\phi(u) > c)$).

- ☐ **Non-intrusive/sampling methods:**
  - ▷ **Monte Carlo**
  - ▷ Stochastic Collocation
  - ▷ Polynomial Chaos
  - ▷ Gaussian Process Emulation
  - ▷ Reduced Basis
  - ▷ ...

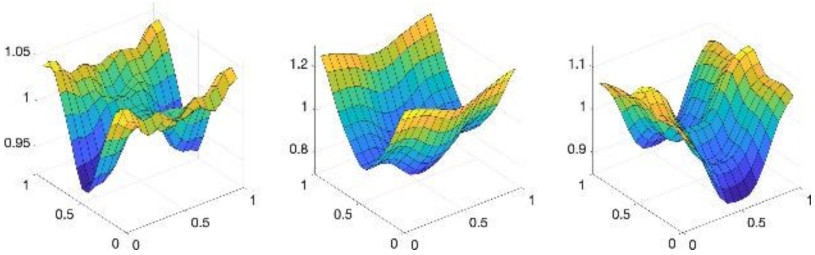
Common Advantage: Easy to wrap around existing (e.g., FEM) codes.

Aside: Karhunen-Loève (KL) expansion

Second-order random fields $\kappa(\boldsymbol{x}, \omega) \in L^2(\Omega, L^2(D))$ can be decomposed as

$$\kappa(\boldsymbol{x}, \boldsymbol{y}(\omega)) = \underbrace{\mu(\boldsymbol{x})}_{\text{mean}} + \underbrace{\sum_{m=1}^{\infty} \sqrt{\lambda_m}\phi_m(\boldsymbol{x}) y_m(\omega)}_{\text{random part}}$$

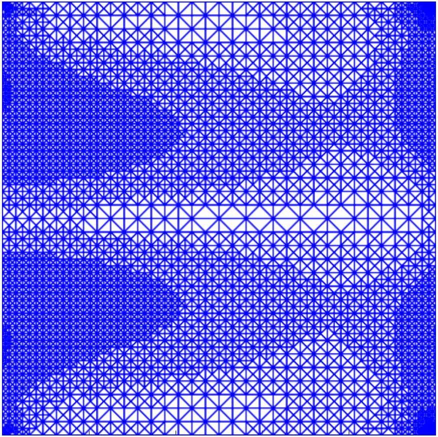
- $(\lambda_m, \phi_m(\boldsymbol{x}))$ are **eigenvalues** & **eigenfunctions** of an integral operator associated with the **covariance function** $C(\boldsymbol{x}, \boldsymbol{x}')$.
- $y_1, y_2, \ldots$ are **uncorrelated** with mean zero and unit variance.

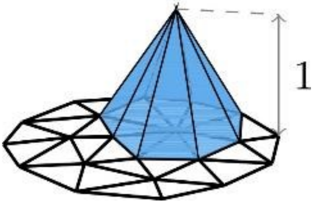


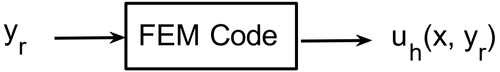
Reminder: Finite Element Methods (FEMs)

FEM codes **approximate** PDE solutions using basis defined on a **mesh**.

$$u(\mathbf{x}) \approx u_h(\mathbf{x}) := \sum_{i=1}^{n_h} u_i \phi_i(\mathbf{x}), \quad (h = \text{mesh size parameter}).$$







1. **Randomly** generate independent samples $\boldsymbol{y}_r$ of the input(s)
2. **Use (FEM) code** to approximate PDE solution $u(\boldsymbol{x}, \boldsymbol{y}_r) \approx u_h(\boldsymbol{x}, \boldsymbol{y}_r)$

$$A_r \mathbf{u}_r = \mathbf{f}_r$$

3. Take **sample average:**

$$\mathbb{E}[u] \approx \frac{1}{N} \sum_{r=1}^{N} u_h(\boldsymbol{x}, \boldsymbol{y}_r)$$

Computational benefits:

- ✓ Allows reuse of existing (FEM) codes
- ✓ ‘Trivially’ paralellisable
- ✓ **No restrictions on type of random inputs/PDE structure**

What's wrong with standard Monte Carlo?

The error in approximating $Q = \mathbb{E}(u)$ has two components:

$$RMSE \leq O(h^a) + O(N^{-1/2}).$$

- ▷ To **guarantee** that $RMSE \leq TOL$, we need to choose

$$h = O(TOL^{1/a}), \qquad N = O(TOL^{-2}).$$

- ▷ Even if an **optimal solver** is used, the **cost** to achieve $RMSE \leq TOL$ is:

$$N \times \underbrace{O(n_h)}_{\text{1 solve}} = N \times O(h^{-d}) = O(TOL^{-2-d/a}).$$

☐ **Advantages:**

- ✓ Do not suffer from the **curse of dimensionality**
- ✓ Modern **multilevel MC methods** may reduce cost significantly

☐ **Disadvantages/Issues:**

- × Only gives samples/approximations of moments.
- × Can only handle fixed no. of random input parameters.
- × A posteriori error estimation?

Part III: Intrusive Methods

- Refers to **Galerkin approximation** of a weak form of PDE.
- Approximation usually takes the form:

$$u(\boldsymbol{x}, \mathbf{y}) \approx \sum_{\boldsymbol{\alpha}\in J_P} u_{\boldsymbol{\alpha}}(\boldsymbol{x}) \underbrace{\psi_{\boldsymbol{\alpha}}(\mathbf{y})}_{\text{polynomials}} .$$

Assume: Parameters y_m are associated with **independent** random variables.

Key Benefits:

- ✓ Provides a **surrogate**
- ✓ Can handle (countably) **infinitely** many random inputs
- ✓ Computable **error estimators** η available for which can prove

$$C_1\,\eta \leq \mathit{Error} \leq C_2\,\eta$$

- ✓ (For some problems) can **break** curse of dimensionality.

Early Work: Ghanem & Spanos (1991), Deb, Babuška & Oden (2001), Babuška, Tempone, Zouraris (2004), Frauenfelder, Schwab & Todor (2005), Matthies & Keese (2005), . . .

Example: Scalar Elliptic PDEs

1. **Stochastic/Parametric PDE:**

$$-\nabla \cdot (\kappa(\boldsymbol{x}, \mathbf{y}) \nabla u(\boldsymbol{x}, \mathbf{y})) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in D \subset \mathbb{R}^{2,3}, \quad \boldsymbol{y} \in \Gamma$$

with

$$0 < \kappa_{\min} \leq \kappa(\boldsymbol{x}, \mathbf{y}) \leq \kappa_{\max} < \infty \quad \text{a.e. in } D \times \Gamma.$$

2. **Weak Problem:** Find $u \in V := L^2_\pi(\Gamma, H^1(D))$ satisfying

$$\int_\Gamma \int_D \kappa \nabla u \cdot \nabla v \, d\mathbf{x} \quad d\pi(\mathbf{y}) = \int_\Gamma \int_D f \, v \, d\mathbf{x} \quad d\pi(\mathbf{y}) \qquad \forall v \in V,$$

where π is a **probability measure**.

Example: Scalar Elliptic PDEs

1. **Stochastic/Parametric PDE:**

$$-\nabla \cdot (\kappa(\boldsymbol{x}, \mathbf{y}) \nabla u(\boldsymbol{x}, \mathbf{y})) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in D \subset \mathbb{R}^{2,3}, \quad \boldsymbol{y} \in \Gamma$$

with

$$0 < \kappa_{min} \leq \kappa(\boldsymbol{x}, \mathbf{y}) \leq \kappa_{max} < \infty \quad \text{a.e. in } D \times \Gamma.$$

2. **Weak Problem:** Find $u \in V$ satisfying:

$$A(u, v) = \ell(v), \qquad \forall v \in V$$

where $A(\cdot, \cdot)$ is an **inner product** that induces an **energy norm** $\|\cdot\|_A$.

Example: Scalar Elliptic PDEs

1. **Stochastic/Parametric PDE:**

$$-\nabla \cdot (\kappa(\boldsymbol{x}, \mathbf{y}) \nabla u(\boldsymbol{x}, \mathbf{y})) = f(\boldsymbol{x}), \qquad \boldsymbol{x} \in D \subset \mathbb{R}^{2,3}, \quad \boldsymbol{y} \in \Gamma$$

with

$$0 < \kappa_{min} \leq \kappa(\boldsymbol{x}, \mathbf{y}) \leq \kappa_{max} < \infty \quad \text{a.e. in } D \times \Gamma.$$

2. **Galerkin Approximation:** Find $u_X \in X \subset V$ satisfying:

$$A(u_X, v) = \ell(v), \qquad \forall v \in X$$

where $A(\cdot, \cdot)$ is an **inner product** that induces an **energy norm** $\|\cdot\|_A$.

3. **Solve Linear System:** $A\mathbf{u} = \mathbf{f}$.

4. **Error Equations:** The error $e := u - u_X \in V$ satisfies:

$$A(e, v) = \underbrace{\ell(v) - A(u_X, v)}_{\text{residual } R(v)} \qquad \forall v \in V.$$

☐ **Key Question:** How to choose X so that $\|e\|_A \leq TOL$?

The Simple Way: Tensor Product Spaces

$$X = H \otimes P$$

- $H = \text{span}\{\phi_i(\boldsymbol{x}), i = 1, \ldots, n_h\}$ is a **finite element** space on D.
- $P = \text{span}\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}), \boldsymbol{\alpha} \in J_P\} \subset L^2_\pi(\Gamma)$ is a set of **global polynomials**.

Example: Let $\boldsymbol{\alpha} = (1, 0, 2, 0, 10, 0, 0, \ldots)$. Set $\psi_0(y_m) = 1$ and define

$$\psi_{\boldsymbol{\alpha}}(\boldsymbol{y}) = \prod_{m=1}^{\infty} \psi_{\alpha_m}(y_m) = \psi_1(y_1)\psi_2(y_3)\psi_{10}(y_5).$$

Choose univariate polynomials $\psi_0, \psi_1, \ldots$ so that

$$\mathbb{E}_\pi\left[\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\psi_{\boldsymbol{\beta}}(\boldsymbol{y})\right] = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}$$

Straight-forward for **product measures**.

Reference: Xiu & Karniadakis, *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM J. Sci. Comput., 24(2), 2002.

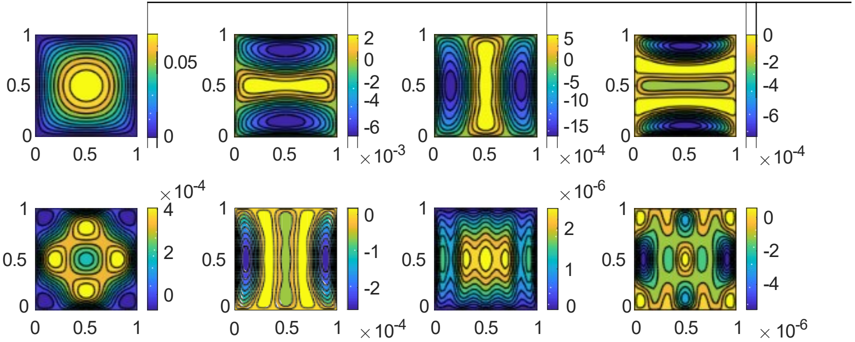
Useful as a Surrogate

Solving $A\mathbf{u} = \mathbf{f}$ gives the coefficients $u_{i,\boldsymbol{\alpha}}$ in the approximation

$$u_X(\mathbf{x},\mathbf{y}) = \sum_{\boldsymbol{\alpha}\in J_P} \left(\sum_{i=1}^{n_H} u_{i,\boldsymbol{\alpha}}\phi_i(\mathbf{x}) \right) \psi_{\boldsymbol{\alpha}}(\mathbf{y}) = \sum_{\boldsymbol{\alpha}\in J_P} \underbrace{u_{\boldsymbol{\alpha}}(\mathbf{x})}_{\in H} \psi_{\boldsymbol{\alpha}}(\mathbf{y}).$$

Can be evaluated cheaply for any $\mathbf{y}^*$ of interest. Useful for: Forward UQ, **Inverse UQ**, Design/Optimisation, etc.

Test Problem: 8 spatial modes $u_{\boldsymbol{\alpha}}(\mathbf{x})$



Linear System (Stochastically Linear)

If $X = H \otimes \mathrm{P}$ and random inputs are **stochastically linear**, then

$$A = G_0 \otimes K_0 + \sum_{m=1}^{M} G_m \otimes K_m$$

and matrices G_m and K_m are **sparse.**

Problem: No. of equations is $n_H n_{\mathrm{P}} = \dim(H) \times \dim(\mathrm{P})$.

Example: If P is polynomials of **total degree** $\leq k$ in M variables,

$$n_{\mathrm{P}} = \frac{(M+k)!}{M!k!}.$$

$M = 10$			$M = 20$		
$k = 2$	$k = 2$	$k = 4$	$k = 2$	$k = 3$	$k = 4$
66	286	1,001	231	1,771	10,626

Preconditioned CG: Toy Problem

$$-\nabla \cdot (\kappa(\mathbf{x}, \mathbf{y}) \nabla u(\mathbf{x}, \mathbf{y})) = f(\mathbf{x}), \qquad \kappa(\boldsymbol{x}, \boldsymbol{y}) = \mu + \sigma \sum_{m=1}^{\infty} \sqrt{\lambda_m} \varphi_m(\boldsymbol{x}) y_m,$$

on $D = [-1, 1]^2$ where $y_m \sim U(-\sqrt{3}, \sqrt{3})$, and (λ_m, φ_m) are eigenpairs associated with **separable exponential** covariance.

FEM discretisation has $\boxed{n_H = 65,025}$.

M	k	n_P	$iter$	$time(s)$
	2	231	6	7.4e1
20	3	1,771	6	5.6e2
	4	**10,629***	**Out of Memory**	-

***691 million equations.**

$$\|u - u_X\|_A \leq ?$$

- ✓ If solution is **analytic** in $\boldsymbol{y}$, error associated with choice of P decays **exponentially** w.r.t polynomial degree[1].
- ✓ When working with simple tensor product spaces

$$X = H \otimes \mathrm{P}$$

convergence rate deteriorates as the no. of parameters $M \to \infty$.

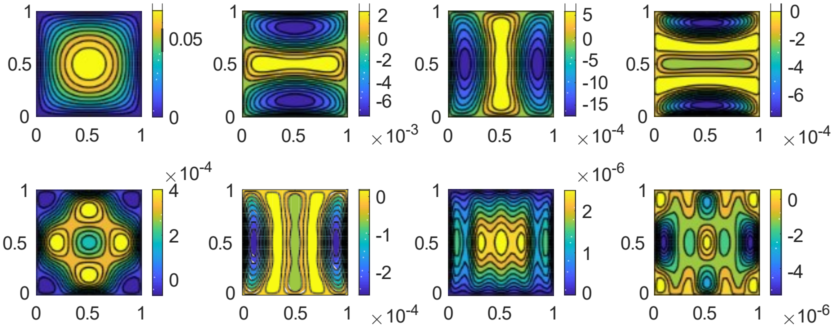
To **break curse of dimensionality**, use 'multilevel' spaces of the form

$$X := \bigoplus_{\boldsymbol{\alpha} \in J_P} H_1^{\boldsymbol{\alpha}} \otimes P^{\boldsymbol{\alpha}}, \qquad P^{\boldsymbol{\alpha}} := \mathrm{span}\{\psi_{\boldsymbol{\alpha}}(\boldsymbol{y})\}$$

[1] **Cohen, DeVore, Schwab**. *Analytic regularity and polynomial approx. of parametric and stochastic elliptic PDE's*, Anal. Appl., 9(1), 2011.

$$u_X(\boldsymbol{x},\mathbf{y}) = \sum_{\boldsymbol{\alpha}\in J_P} \left(\sum_{i=1}^{n_\alpha} u_{i,\alpha}\phi_i^{\alpha}(\boldsymbol{x}) \right) \psi_{\boldsymbol{\alpha}}(\mathbf{y}) = \sum_{\boldsymbol{\alpha}\in J_P} \underbrace{u_{\boldsymbol{\alpha}}(\boldsymbol{x})}_{\in H_1^{\alpha}} \psi_{\boldsymbol{\alpha}}(\mathbf{y}).$$

Test Problem: 8 spatial modes $u_{\alpha}(\boldsymbol{x})$



The Smart Way: Adaptive Multilevel Spaces

$$\boxed{X := \bigoplus_{\boldsymbol{\alpha}\in J_P} H_1^{\boldsymbol{\alpha}} \otimes P^{\boldsymbol{\alpha}}}.$$

- Start with a **low-dimensional** space X and compute $u_0 \in X$.
- Estimate the (energy) error using only **a posterior information**

$$\eta \approx \|u - \hat{u}_0\|_A.$$

- Decide how best to **enrich X** if $\eta > TOL$.
- Compute a sequence of approximations $\hat{u}_0, \ldots, \hat{u}_L$ until

$$\boxed{\eta \leq TOL}.$$

One possibility is **'Hierarchical Error Estimation'**[2].

[2] Bank & Weiser, Bank & Smith, Ainsworth & Oden

Hierarchical Error Estimation - in a nutshell

For $u_X \in X \subset V$, we know $e := u - u_X \in V$ satisfies:

$$\boxed{A(e, v) = R(v) \qquad \forall v \in V}.$$

---

1. Consider $e_W \in W \supset X$ such that:

$$A(e_W, v) = R(v), \qquad \forall v \in W.$$

2. Choose

$$W = X \oplus \underbrace{Y}_{\text{'detail'}}, \qquad X \cap Y = \{0\}$$

and define **error estimate** $\boxed{\eta := \|e_Y\|_A}$ where

$$e_Y \in Y: \quad A(e_Y, v) = R(v), \qquad \forall v \in Y.$$

☐ Solve a new problem (linear system) to obtain $e_Y \in Y$.

Key Challenges: Error Estimation & Adaptivity

- **Accuracy:** How to choose Y so that $\eta = \|e_Y\|_A$ satisfies

$$\boxed{C_1\,\eta \le \|e\|_A \le C_2\,\eta}, \qquad \text{with } C_1 \approx 1 \approx C_2?$$

- **Localisation:** How to choose Y so that problem to be solved for e_Y decouples into 'local' subproblems whose solutions give accurate estimates of **distinct** contributions to current approximation error?

- **Convergence:** How best to use error components to inform enrichment of X to drive an adaptive algorithm for which

$$\eta_k \approx \|u - \hat{u}_{X_k}\|_A$$

decays at the 'optimal' rate as $k \to \infty$?

Example: Synthetic KL Expansion

$D = [0,1]^2$, $\kappa(\mathbf{x},\mathbf{y}) = 1 + \sum_{m=1}^{\infty} \kappa_m(\mathbf{x})\, y_m$ with $y_m \sim U(-1,1)$ and

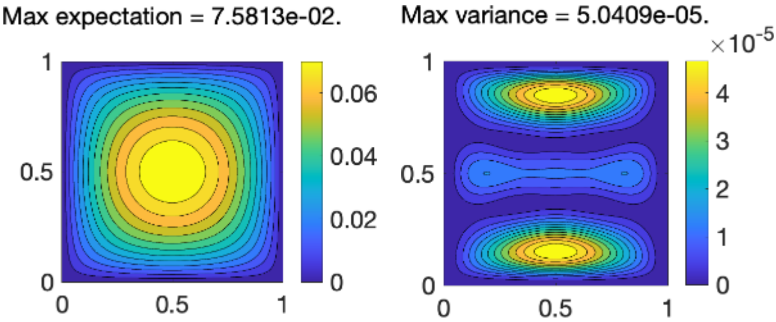
$$\kappa_m(\mathbf{x}) := 0.547 m^{-2} \cos(2\pi\beta_m^1 x_1)\cos(2\pi\beta_m^2 x_2)$$

$$X := \bigoplus_{\boldsymbol{\alpha}\in J_P}^{M} H_1^{\boldsymbol{\alpha}} \otimes P^{\boldsymbol{\alpha}}$$

- **INITIALIZE:** $J_P = \{\mathbf{0}, (1,0,\ldots,0)\}$ and $H_1^{\boldsymbol{\alpha}} = \mathrm{Q}_1(h)$ (bilinear FEM) on coarse uniform mesh.
- **CHOOSE DETAIL SPACE**
- **ERROR ESTIMATION: If** $\eta \leq TOL$, then **STOP.** Otherwise,
  - Improve $H_1^{\boldsymbol{\alpha}}$ (e.g. refine the mesh) for one or more $\boldsymbol{\alpha} \in J_P$, **OR**
  - Add new multi-indices $\boldsymbol{\beta}$ to J_P
- Choose TOL = `1.5e-3`.
- **Target convergence rate:** $\boxed{N_{\text{dof}}^{-1/2}}$.

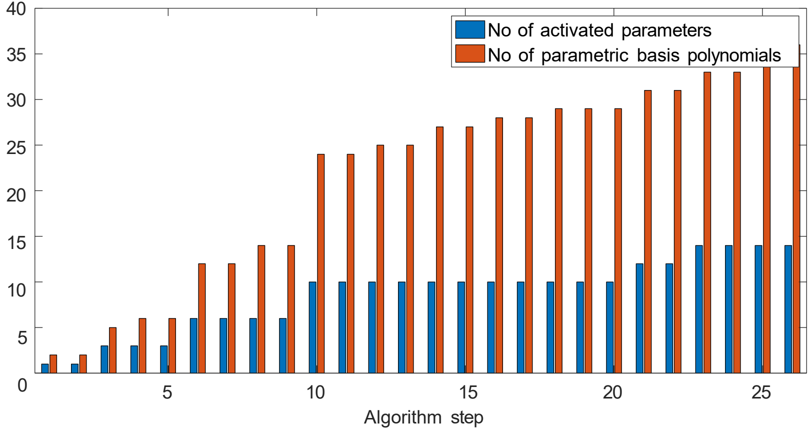
Eigel, Gittelson, Schwab, Zander. Adaptive stochastic Galerkin FEM. **CMAME** (2014).

Example: Final Mean & Variance




```
Time = 20.90 seconds.
Total iterations = 26.
No. parametric polynomials = 36
No. activated variables = 14.
Total DOF = 104,452.
```

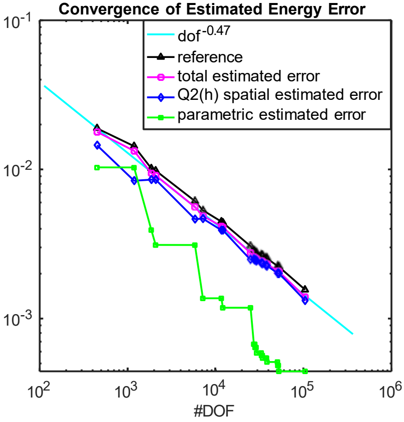
Example: Final Approximation Space

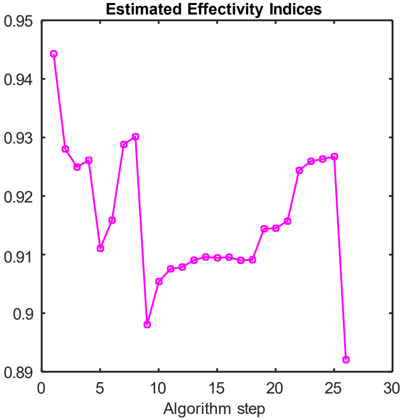


At the final step $X := \bigoplus_{\boldsymbol{\alpha} \in J_P} H_1^{\boldsymbol{\alpha}} \otimes P^{\boldsymbol{\alpha}}$

- J_P contains 36 multi-indices, ($M = 14$ **activated parameters**)
- $H_1^{\boldsymbol{\alpha}} = \mathrm{Q}_1(h)$ with $h = 2^{-8}, 2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}$ **(1,1,3,6,25 terms)**

Example: Convergence & Accuracy





Summary: Adaptive Intrusive Methods

- **MATLAB code:**

  https://github.com/ceapowell/ML-SGFEM

- Adaptive stochastic Galerkin methods can be used to build **surrogates** with automated and **rigorous error control.**
- For 'nice enough' problems, **multilevel** SG methods can achieve **convergence rates** associated with the chosen spatial discretization for the analogous **deterministic** problem.
- Accurate a posteriori **error estimation** is key for driving adaptive algorithms and designing X in a smart way.
- Not just for scalar elliptic PDEs . . .

SIAM UK & Ireland Section

- Next annual meeting will be in **Manchester** on **Friday 22nd March**.
- Thanks to **IMA** for supplementary funding.
- Election for new **Treasurer/Secretary** soon.

- **National Student Chapter Conference** will take place in **Cardiff**.