PhD Research in Numerical Analysis

Catherine E. Powell

University of Manchester

c.powell@manchester.ac.uk

September 11, 2020

Introduction

NA is concerned with the **development** and **analysis** of algorithms for **approximating** solutions to problems, such as

- Integration: $\int_D f(\mathbf{x})\,d\mathbf{x}$,
- Interpolation / polynomial approximation: $p(\mathbf{x}) \approx f(\mathbf{x})$
- ODEs: $\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t))$
- PDEs: $-\nabla^2 u(\mathbf{x}) = f(\mathbf{x})$
- Linear systems: $A\mathbf{v} = \mathbf{b}$
- Optimisation: $\min_{\mathbf{x}\in S} f(\mathbf{x})$
- ...

Introduction

NA is concerned with the **development** and **analysis** of algorithms for **approximating** solutions to problems, such as

- Integration: $\int_D f(\mathbf{x})\, d\mathbf{x}$,
- Interpolation / polynomial approximation: $p(\mathbf{x}) \approx f(\mathbf{x})$
- ODEs: $\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}(t))$
- PDEs: $-\nabla^2 u(\mathbf{x}) = f(\mathbf{x})$
- Linear systems: $A\mathbf{v} = \mathbf{b}$
- Optimisation: $\min_{\mathbf{x} \in S} f(\mathbf{x})$
- ...

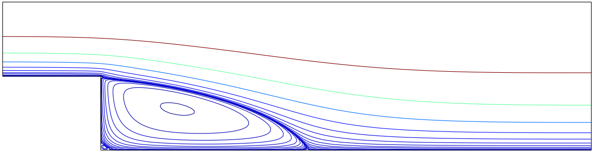
Modern research is driven by: **complex real-world applications**, evolving computer architectures and **increasing computing power**.

PDE Example: Navier–Stokes Equations

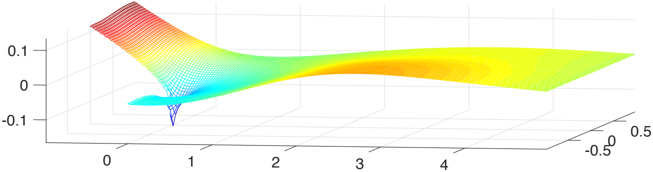
$$-\nu\nabla^2\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p \quad = \quad \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0.$$

We can approximate **u** (the velocity) and p (the fluid pressure) using e.g. **finite element methods** (FEMs).

Streamlines: selected



Pressure field



Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

- Prove whether a solution u **exists** and is **unique**.

Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

- Prove whether a solution u **exists** and is **unique**.
- Prove a **regularity** result for u. (How smooth is u?)

Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

- Prove whether a solution u **exists** and is **unique**.
- Prove a **regularity** result for u. (How smooth is u?)
- Design a **computationally efficient** method for computing an approximation u_n which is **stable** and **convergent**

$$\|u - u_n\| \to 0, \qquad \text{as} \quad n \to \infty,$$

(and respects the underlying physics in the case of PDEs).

Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

- Prove whether a solution u **exists** and is **unique**.
- Prove a **regularity** result for u. (How smooth is u?)
- Design a **computationally efficient** method for computing an approximation u_n which is **stable** and **convergent**

$$\|u - u_n\| \to 0, \qquad \text{as} \quad n \to \infty,$$

  (and respects the underlying physics in the case of PDEs).
- Prove '**a priori**' bounds for $e = \|u - u_n\|$ and **rates** of convergence.

Numerical Analysis Approach

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in D.$$

- Prove whether a solution u **exists** and is **unique**.
- Prove a **regularity** result for u. (How smooth is u?)
- Design a **computationally efficient** method for computing an approximation u_n which is **stable** and **convergent**

$$\|u - u_n\| \to 0, \qquad \text{as} \quad n \to \infty,$$

  (and respects the underlying physics in the case of PDEs).
- Prove '**a priori**' bounds for $e = \|u - u_n\|$ and **rates** of convergence.
- Design a method to estimate e '**a posteriori**'.

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- ▷ Pick a tolerance, TOL.

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- Pick a tolerance, TOL.
- Compute a **cheap** approximation u_1 (for a small value of n)

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- Pick a tolerance, TOL.
- Compute a **cheap** approximation u_1 (for a small value of n)
- Compute an estimate of the error

$$\eta \approx \|u - u_1\|.$$

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- Pick a tolerance, TOL.
- Compute a **cheap** approximation u_1 (for a small value of n)
- Compute an estimate of the error

$$\eta \approx \|u - u_1\|.$$

- If $\eta \leq TOL$, then STOP.

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- Pick a tolerance, TOL.
- Compute a **cheap** approximation u_1 (for a small value of n)
- Compute an estimate of the error

$$\eta \approx \|u - u_1\|.$$

- If $\eta \leq TOL$, then STOP. Otherwise, compute a **more accurate** approximation u_2 (with a higher value of n), ...

Balancing Work & Accuracy

Modern numerical algorithms are usually **adaptive**.

- Pick a tolerance, TOL.
- Compute a **cheap** approximation u_1 (for a small value of n)
- Compute an estimate of the error

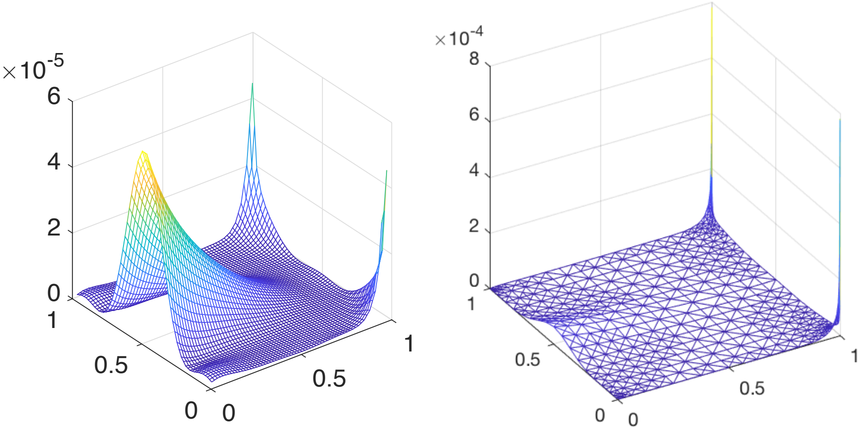
$$\eta \approx \|u - u_1\|.$$

- If $\eta \leq TOL$, then STOP. Otherwise, compute a **more accurate** approximation u_2 (with a higher value of n), ...

- Compute a **sequence** of approximations $u_1, u_2, \ldots$ until

$$\eta \leq TOL.$$

Non-adaptive vs. Adaptive FEM Approximation



Active Areas of NA Research in UK*

*Disclaimer: This list is by no means complete ...

- **Numerical Solution of PDEs:** approximation theory, application-specific methods for problems in engineering, biology etc, wave scattering, development & analysis of FEMs, spectral methods, physics-preserving methods, adaptivity, ...

Active Areas of NA Research in UK*

*Disclaimer: This list is by no means complete ...

- ▷ **Numerical Solution of PDEs:** approximation theory, application-specific methods for problems in engineering, biology etc, wave scattering, development & analysis of FEMs, spectral methods, physics-preserving methods, adaptivity, . . .
- ▷ **Numerical Linear Algebra:** iterative methods for solving linear systems, preconditioning, matrix equations, eigenvalue problems, mixed precision algorithms, . . .

Active Areas of NA Research in UK*

*Disclaimer: This list is by no means complete ...

- **Numerical Solution of PDEs:** approximation theory, application-specific methods for problems in engineering, biology etc, wave scattering, development & analysis of FEMs, spectral methods, physics-preserving methods, adaptivity, . . .
- **Numerical Linear Algebra:** iterative methods for solving linear systems, preconditioning, matrix equations, eigenvalue problems, mixed precision algorithms, . . .
- **Uncertainty Quantification (UQ):** stochastic ODEs & PDEs, Bayesian inverse problems, multilevel and adaptive methods, . . .

Active Areas of NA Research in UK*

*Disclaimer: This list is by no means complete ...

- **Numerical Solution of PDEs:** approximation theory, application-specific methods for problems in engineering, biology etc, wave scattering, development & analysis of FEMs, spectral methods, physics-preserving methods, adaptivity, ...
- **Numerical Linear Algebra:** iterative methods for solving linear systems, preconditioning, matrix equations, eigenvalue problems, mixed precision algorithms, ...
- **Uncertainty Quantification (UQ):** stochastic ODEs & PDEs, Bayesian inverse problems, multilevel and adaptive methods, ...
- **Optimisation:** Fundamental theory, inverse problems, imaging applications and machine learning, compressed sensing, ...

The Challenge: Derive **computationally efficient** methods to solve a class of elliptic PDEs with uncertain inputs, that converges at an **optimal rate**.

Project 1: Methodological (PDEs)

The Challenge: Derive **computationally efficient** methods to solve a class of elliptic PDEs with uncertain inputs, that converges at an **optimal rate**.

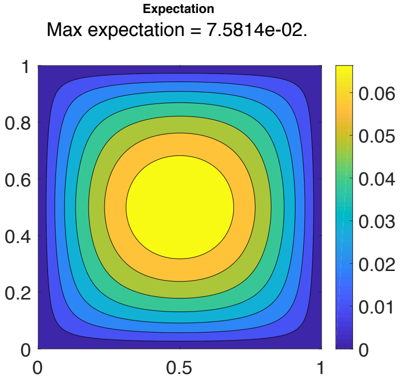
$$-\nabla \cdot \left(\underbrace{a(\mathbf{x}, \mathbf{y})}_{\text{coefficient}} \nabla u(\mathbf{x}, \mathbf{y}) \right) = f(\mathbf{x})$$

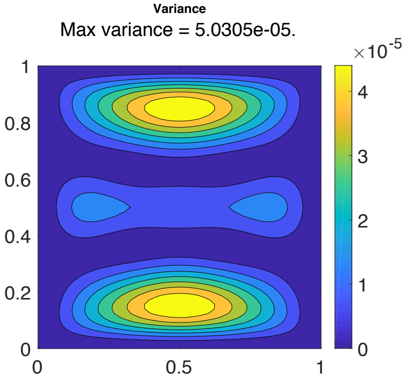
where the coefficient is a **parameter-dependent** function

$$a(\mathbf{x}, \mathbf{y}) = a_0(\mathbf{x}) + \sum_{m=1}^{\infty} a_m(\mathbf{x}) \underbrace{y_m .}_{\text{parameter}}$$

Numerical Results: Test Problem

Estimated **mean** (left) and **variance** (right) of $u(\mathbf{x}, \mathbf{y})$

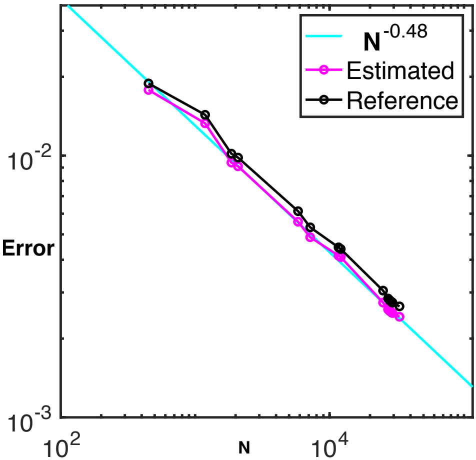




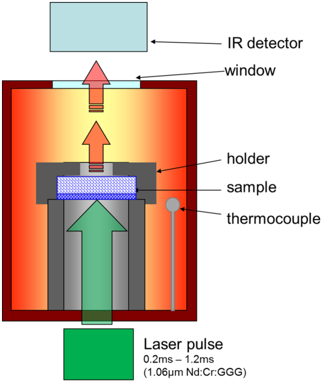
Estimated Error & Convergence Rate

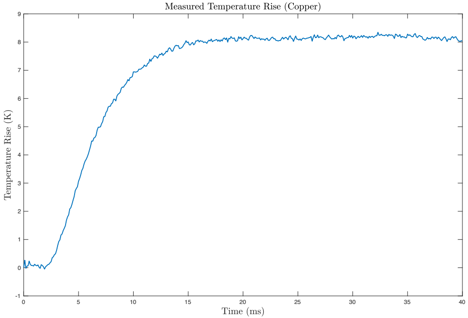
N:= dimension of approximation space.

Optimal convergence rate: $N^{-1/2}$ (for this test problem).



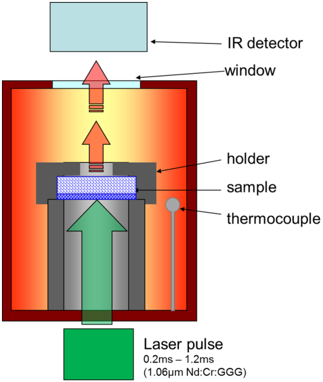
The Challenge: Approximate thermal conductivity λ of materials using data from laser flash experiments & estimate associated uncertainty.

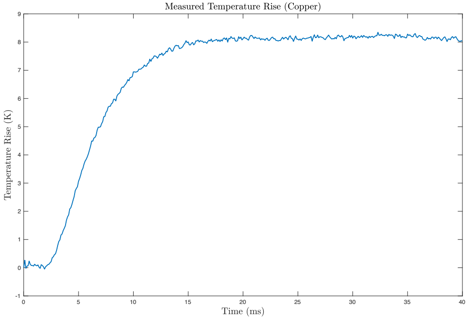




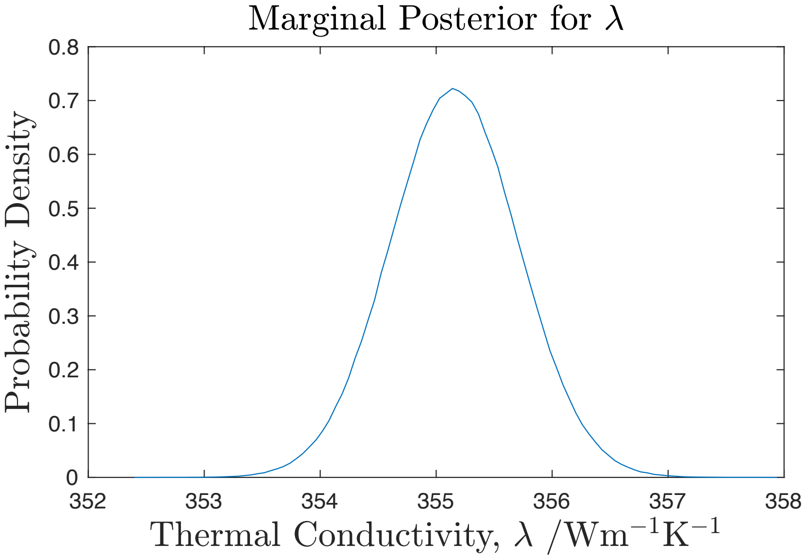
Project 2: Application-Focused (iCASE PhD with NPL)

The Challenge: Approximate thermal conductivity λ of materials using data from laser flash experiments & estimate associated uncertainty.

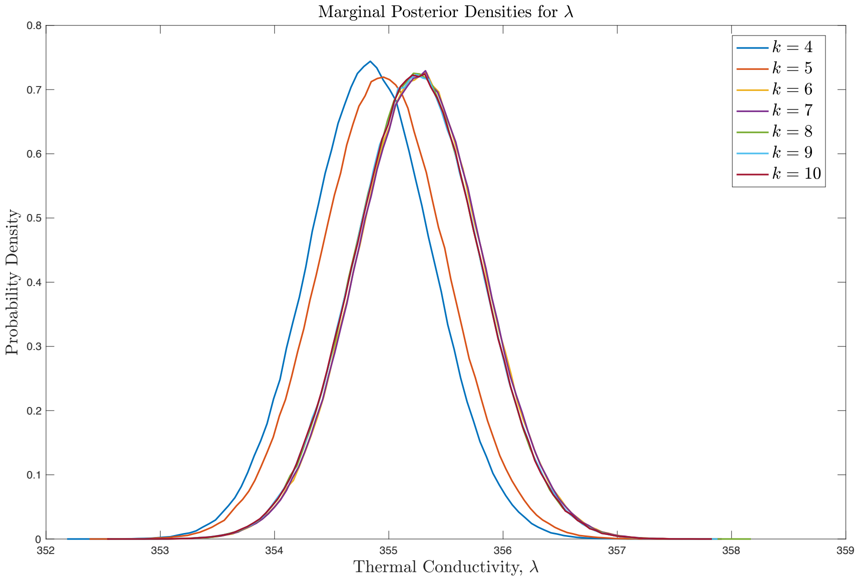




Approach: Formulate as a **Bayesian inverse problem** and approximate the conditional pdf of λ (given the data).


Marginal Posterior for λ
Probability Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
352
353
354
355
356
357
358
Thermal Conductivity, λ /$\mathrm{Wm^{-1}K^{-1}}$

Convergence (as polynomial degree $k \to \infty$)



Project 3: Methodological (Linear Algebra)

Challenge: Design adaptive iterative methods for solving **massive** linear systems $A\mathbf{v} = \mathbf{b}$ where

$$A = \underbrace{G_1}_{m \times m} \otimes \underbrace{K_1}_{n \times n} + G_1 \otimes K_2 + \ldots + G_M \otimes K_M,$$

and $\otimes :=$ the matrix 'Kronecker product'.

Project 3: Methodological (Linear Algebra)

Challenge: Design adaptive iterative methods for solving **massive** linear systems $A\mathbf{v} = \mathbf{b}$ where

$$A = \underbrace{G_1}_{m\times m} \otimes \underbrace{K_1}_{n\times n} + G_1 \otimes K_2 + \ldots + G_M \otimes K_M,$$

and $\otimes :=$ the matrix 'Kronecker product'.

Equivalently, approximate the **matrix** V satisfying the matrix equation

$$K_1 V G_1^\top + K_2 V G_2^\top + \ldots + K_M V G_M^\top = B$$

where $\mathbf{v} = \text{vec}(V)$ and $\mathbf{b} = \text{vec}(B)$.

Challenge: Design adaptive iterative methods for solving **massive** linear systems $A\mathbf{v} = \mathbf{b}$ where

$$A = \underbrace{G_1}_{m\times m} \otimes \underbrace{K_1}_{n\times n} + G_1 \otimes K_2 + \ldots + G_M \otimes K_M,$$

and $\otimes :=$ the matrix 'Kronecker product'.

Equivalently, approximate the **matrix** V satisfying the matrix equation

$$K_1 V G_1^\top + K_2 V G_2^\top + \ldots + K_M V G_M^\top = B$$

where $\mathbf{v} =$vec(V) and $\mathbf{b} =$vec(B).

- **Applications:** Dynamical systems, high-dimensional & stochastic PDEs.

Things to think about before applying

- Try and **narrow down the area** (in NA) you want to work in
- Do you want to prove theorems, do computations, or both?
- Do you want to do **core methodological** research, or something **application-driven**?
- University webpages can be hard to navigate - look at **research** pages.
- **Look beyond CDTs** - not every institution has one.
- Attend (virtual?) **Open Days**.