Fréchet Derivatives of Matrix Functions and Applications

Samuel Relton

samuel.relton@maths.man.ac.uk @sdrelton

samrelton.com blog.samrelton.com

Joint work with Nicholas J. Higham

higham@maths.man.ac.uk @nhigham

www.maths.man.ac.uk/~higham nickhigham.wordpress.com

University of Manchester, UK

September 4, 2014

Manchester Numerical Analysis

Outline

- Matrix Functions, their Derivatives, and the Condition Number
- Elementwise Sensitivity
- Physics: Nuclear Activation Sensitivity Problem
- Differential Equations: Predicting Algebraic Error in the FEM

Matrix Functions

We are interested in functions $f : \mathbb{C}^{n\times n} \mapsto \mathbb{C}^{n\times n}$ e.g.

Matrix Exponential $$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Matrix Cosine $$\cos(A) = \sum_{k=0}^{\infty} \frac{(-1)^k A^{2k}}{(2k)!}$$

M\crNA
Manchester Numerical Analysis

Matrix Functions

We are interested in functions $f : \mathbb{C}^{n\times n} \mapsto \mathbb{C}^{n\times n}$ e.g.

$$\text{Matrix Exponential} \quad e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

$$\text{Matrix Cosine} \quad \cos(A) = \sum_{k=0}^{\infty} \frac{(-1)^k A^{2k}}{(2k)!}$$

- Define $f(A)$ by Taylor series when f is analytic
- If $A = XDX^{-1}$ then $f(A) = Xf(D)X^{-1}$
- Differential equations: $\frac{du}{dt} = Au(t), \quad u = e^{tA}u(0)$
- Use $\cos(A)$ and $\sin(A)$ for second order ODEs

M\crNA Manchester Numerical Analysis

Fréchet Derivatives

Let $f : \mathbb{C}^{n \times n} \mapsto \mathbb{C}^{n \times n}$ be a matrix function.

Definition (Fréchet derivative)

The Fréchet derivative of f at A is the unique linear function $L_f(A, \cdot) : \mathbb{C}^{n \times n} \mapsto \mathbb{C}^{n \times n}$ such that for all E

$$f(A + E) - f(A) - L_f(A, E) = o(\|E\|).$$

Manchester Numerical Analysis

Fréchet Derivatives

Let $f : \mathbb{C}^{n\times n} \mapsto \mathbb{C}^{n\times n}$ be a matrix function.

Definition (Fréchet derivative)

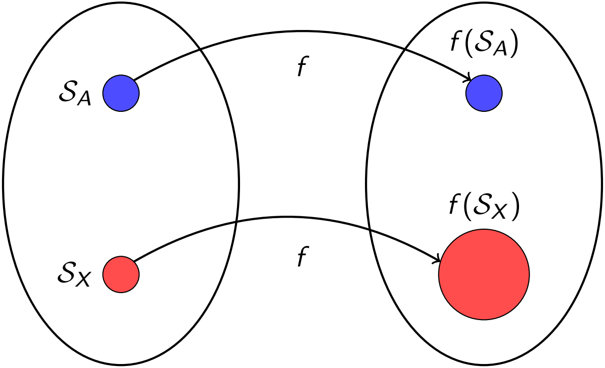
The Fréchet derivative of f at A is the unique linear function $L_f(A, \cdot) : \mathbb{C}^{n\times n} \mapsto \mathbb{C}^{n\times n}$ such that for all E

$$f(A + E) - f(A) - L_f(A, E) = o(\|E\|).$$

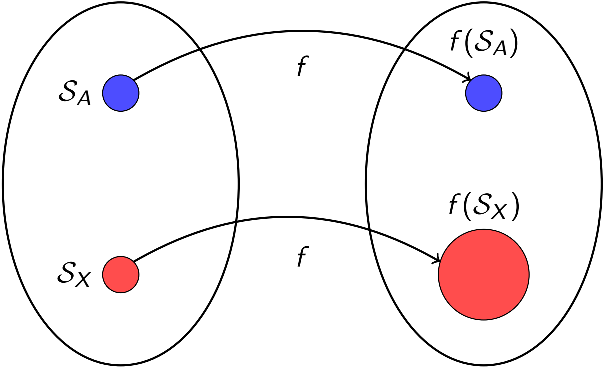
- Applications include manifold optimization, Markov models, bladder cancer, image processing, and network analysis
- Higher order derivatives recently analyzed (Higham & R., 2014)

Manchester Numerical Analysis

Sensitivity of Matrix Functions



Sensitivity of Matrix Functions



The function f is well conditioned at A and
ill conditioned at X

The Norm-wise Condition Number

The two condition numbers for a matrix function are:

$$\text{cond}_{\text{abs}}(f, A) = \max_{\|E\|=1} \|L_f(A, E)\|,$$

$$\text{cond}_{\text{rel}}(f, A) = \max_{\|E\|=1} \|L_f(A, E)\| \frac{\|A\|}{\|f(A)\|}.$$

Manchester Numerical Analysis

Elementwise Sensitivity

If we change just one element A_{ij}, how is $f(A)$ affected?

Let $E_{ij} = [\delta_{ij}]$, then the difference between $f(A)$ and $f(A + \epsilon E_{ij})$ is

$$\|f(A) - f(A + \epsilon E_{ij})\| \approx \epsilon \|L_f(A, E_{ij})\|.$$

- $\|L_f(A, E_{ij})\|$ gives the sensitivity in (i, j) component
- Sometimes we want the t most sensitive elements for $t = 5\colon 20$

M\crNA
Manchester Numerical Analysis

A simple algorithm

To compute the most sensitive t entries of A:

```
for i = 1: n
    for j = 1: n
        if A_ij ≠ 0
            Compute and store ||L_f(A, E_ij)||
        end if
    end for
end for
Take the largest t values of ||L_f(A, E_ij)||
```

Cost: Up to $O(n^5)$ flops since computing $L_f(A, E)$ costs $O(n^3)$ flops

Manchester Numerical Analysis

A simple algorithm

To compute the most sensitive t entries of A:

for $i = 1\colon n$
  for $j = 1\colon n$
    if $A_{ij} \neq 0$
      Compute and store $\|L_f(A, E_{ij})\|$
    end if
  end for
end for
Take the largest t values of $\|L_f(A, E_{ij})\|$

Cost: Up to $O(n^5)$ flops since computing $L_f(A, E)$ costs $O(n^3)$ flops

- Trivially parallel but still very expensive when A is large
- Speed this up using block norm estimation (work in progress)

Manchester Numerical Analysis

The Nuclear Activation Sensitivity Problem

- Chemical reactions: $u'(t) = Au(t)$
- $u(t) = e^{At}u(0)$ tells us the concentration of each element at time t
- $q^T u(t)$ is the dosage at time t
- A_{ij} represents the reaction between elements i and j (so ignore $A_{ij} = 0$)
- A_{ij} is subject to measurement error
  What happens to $q^T u(t)$ when it changes?

Implications for safety in radiation exposure models etc.

M\cr NA
Manchester Numerical Analysis

Nuclear Activation Solution - 1

If A_{ij} is perturbed, this introduces a relative error in $q^T u(t)$ of

$$\frac{|q^T(e^{tA+\epsilon E_{ij}} - e^{tA})u(0)|}{|q^T e^{tA} u(0)|} \approx \epsilon \frac{|q^T L_{e^x}(tA, E_{ij})u(0)|}{|q^T e^{tA} u(0)|}$$

Manchester Numerical Analysis

Nuclear Activation Solution - 1

If A_{ij} is perturbed, this introduces a relative error in $q^T u(t)$ of

$$\frac{|q^T(e^{tA+\epsilon E_{ij}} - e^{tA})u(0)|}{|q^T e^{tA} u(0)|} \approx \epsilon \frac{|q^T L_{e^x}(tA, E_{ij})u(0)|}{|q^T e^{tA} u(0)|}$$

We note that:

- The denominator is the same for all perturbations
- This requires computing a derivative in all directions $A_{ij} \neq 0$
- Can we improve upon this?

M\crNA Manchester Numerical Analysis

Nuclear activation solution - 2

Using $\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X)$ we see the sensitivity in direction E_{ij} is

$$|q^T L_{e^x}(tA, E_{ij})u(0)| = |(u(0)^T \otimes q^T) K_{e^x}(tA)\, \mathrm{vec}(E_{ij})|.$$

Therefore the sensitivity in ALL n^2 directions is

$$|[(u(0)^T \otimes q^T) K_{e^x}(tA)]^T| = |\mathrm{vec}(L_{e^x}(tA, \mathrm{unvec}(u(0) \otimes q)^T)^T|.$$

- Only 1 derivative needed for all sensitivities
- Found 2 bugs in existing commercial software!
- Extend for time dependent coefficients $A = A(t)$

M\crNA Manchester Numerical Analysis

Predicting Algebraic Error in an ODE

Let's solve the model ODE

$$-u'' = f(x), \quad x \in (0,1), \quad u(0) = u(1) = 0$$

with the finite element method using piecewise linear basis functions ϕ_i.

- Exact solution $u(x) = e^{-5(x-0.5)^2} - e^{5/4}$ determines $f(x)$
- Generate a grid of $n = 19$ equally spaced points x_i
- Generate system $Ax = b$ where $A_{ij} = \int_0^1 \phi_i \phi_j$ and $b_i = f(x_i)$.
  $A = \text{diag}(-1, 2, -1)$ in this case
- Solve with CG iteration

MicrNA
Manchester Numerical Analysis

Algebraic and discretization errors

- Let V_h be our finite element space (dimension 19)
- Let $u_h \in V_h$ be the best solution possible from V_h
- Let u_{est}^k be our numerical solution corresponding to k iterations of CG

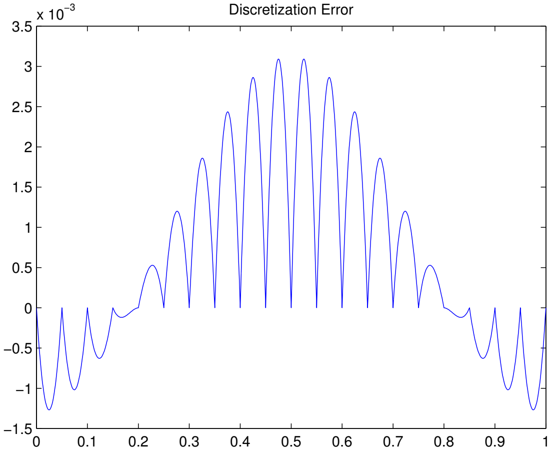
Algebraic and discretization errors

- Let V_h be our finite element space (dimension 19)
- Let $u_h \in V_h$ be the best solution possible from V_h
- Let u_{est}^k be our numerical solution corresponding to k iterations of CG

- The discretization error is $u - u_h$
- The algebraic error is $u_h - u_{est}^k$
- The total error is $u - u_{est}^k =$ alg. err. + disc. err.
- Sometimes alg err dominates the total err, how do we detect this?

Manchester Numerical Analysis

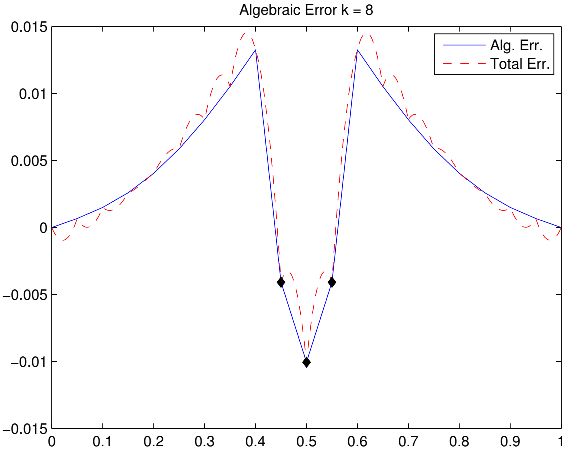
Discretization error



$u - u_h$

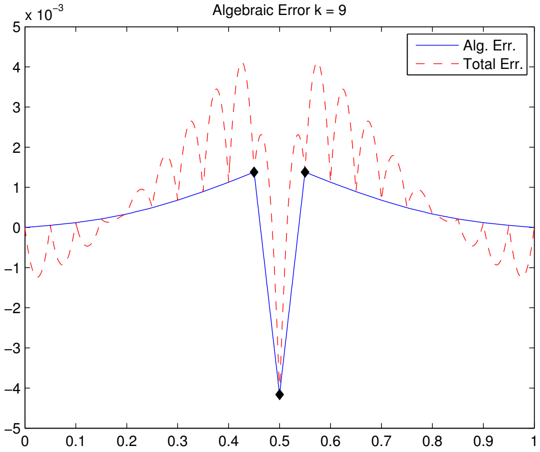
Manchester Numerical Analysis

Algebraic Error - 8 CG iterations



Nodes 9–11 highlighted

Algebraic Error - 9 CG iterations



Nodes 9–11 highlighted

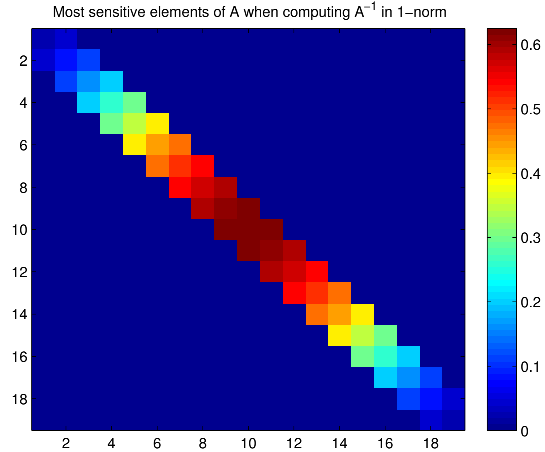
Manchester Numerical Analysis

Elementwise sensitivity analysis

- Taking $f(A) = A^{-1}$ we can calculate the sensitivity of each element
- $L_f(A, E) = -A^{-1}EA^{-1}$ so easily computed
- Ignore $A_{ij} = 0$ since the two basis elements don't overlap
- Results plotted on the following heat map

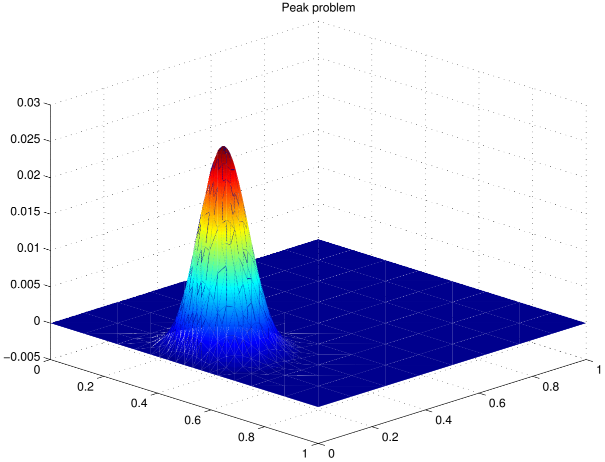
Manchester Numerical Analysis

Elementwise sensitivity analysis

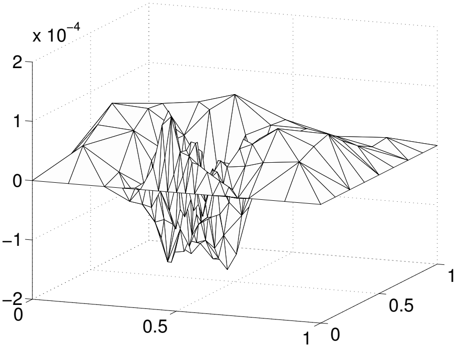


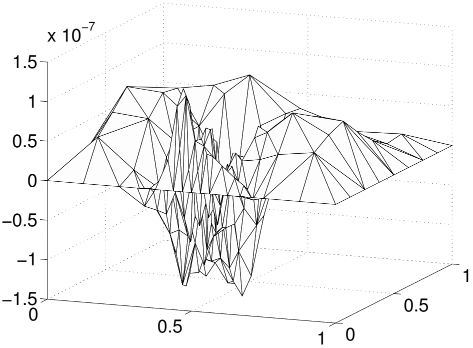
Row/Cols 9–11 in the middle

2D Peak Problem



Algebraic Error Estimation

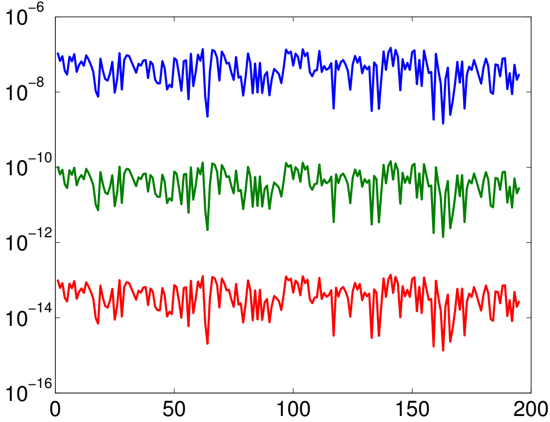




Left: True algebraic error using 7 CG iterations.
Right: Error in estimated algebraic error using 1st Fréchet derivative.

Manchester Numerical Analysis

Higher Order Derivatives to Estimate Alg. Err.



Componentwise error using kth order derivatives, $k = 1, 3, 5$.

Possible extensions

- Can this be used to modify the discretization mesh to obtain better accuracy? (See Papez, Liesen, and Strakos 2014)
- Currently too expensive: can we estimate the sensitivities?
- Can this be extended to $f(A) = e^A$ (exponential integrators)?

Manchester Numerical Analysis

Conclusions

- Explained elementwise sensitivity of matrix functions
- New applications in nuclear physics and FEM analysis
- Former is basically solved, latter needs to be cheaper

Future work:

- Estimate sensitivities more efficiently (block norm estimation)
- Further comparison of nuclear physics solution to commercial alternative
- Further analysis of ODE problem

M\crNA Manchester Numerical Analysis

Higher Order Fréchet Derivatives

Higher order derivatives can be defined recursively:

$$L_f^{(k)}(A + E_{k+1}, E_1, \dots, E_k) - L_f^{(k)}(A, E_1, \dots, E_k) = L_f^{(k+1)}(A, E_1, \dots, E_k, E_{k+1}) + o(\|E_{k+1}\|)$$

Also have a simple method to compute them. For example:

$$f\left(\begin{bmatrix} A & E_1 & E_2 & 0 \\ 0 & A & 0 & E_2 \\ 0 & 0 & A & E_1 \\ 0 & & 0 & A \end{bmatrix}\right) = \begin{bmatrix} f(A) & L_f(A, E_1) & L_f(A, E_2) & L_f^{(2)}(A, E_1, E_2) \\ 0 & f(A) & 0 & L_f(A, E_2) \\ 0 & 0 & f(A) & L_f(A, E_1) \\ 0 & 0 & 0 & f(A) \end{bmatrix}$$

More info in Higham & Relton, SIMAX 35(4), 2014.

M\crNA
Manchester Numerical Analysis