Applied Mathematics Methods Stream

Irene Kyza

Discipline of Mathematics
School of Science and Engineering
University of Dundee

September 2016

Overview: what are methods (for)?

- Most mathematical models coming e.g. from real-life population dynamics, solar physics, fluids etc. can't be solved exactly and explicitly.
- We need to find approximate solutions in a systematic manner:
  - asymptotic approximations based on small parameters;
  - numerical approximations obtained by computer.

  These approaches are often complementary.

- We need to know:
  - what sort of problems we can tackle with these methods;
  - how to adapt them to different problems;
  - what is an appropriate method for each problem;
  - what the approximate solutions tell us.

Delivery and broad topics

- Asymptotic methods for ODEs [5 lectures]
  - Dr David Pritchard (Strathclyde) and Dr Antonia Wilmot– Smith (St Andrews).
- Transforms and integral solutions [3 lectures]
  - Dr Dumitru Trucu (Dundee).
- Further applications of asymptotics [2 lectures]
  - Dr David Pritchard (Strathclyde).

- Numerical methods for stochastic DEs [2 lectures]
  - Prof. Des Higham (Strathclyde).
- Numerical methods for ODEs [2 lectures]
  - Prof. Dugald Duncan (Heriot–Watt).
- Numerical methods for PDEs [4 lectures]
  - Prof. Ping Lin (Dundee).
- Numerical linear algebra [2 lectures]
  - Dr Phil Knight (Strathclyde).

Prerequisites and delivery

We assume you're familiar with standard undergraduate material:

- single- and multi-variable calculus;
- Taylor's theorem;
- differential equations;
- linear algebra;
- complex variables.

Lectures: 13:00–15:00 on Wednesdays. These are in "flipped" format, with exercises to complete beforehand — including before the first lecture!

Assessment

Assessment will be based on two pieces of submitted work per module.

- Each assignment will cover four to six lectures.
- Assignments will include "paper and pencil" and computer work.
- You'll have at least two weeks to tackle each assignment.

Provisional timetable:

- Module 1:
  - Assignment 1 (lectures 1–5): to be submitted by 18 November 2016.
  - Assignment 2 (lectures 6–10): to be submitted by 9 January 2017.
- Module 2:
  - Assignment 1 (lectures 1–4): to be submitted by 3 March 2017.
  - Assignment 2 (lectures 5–10): to be submitted by 7 April 2017.

Asymptotic methods

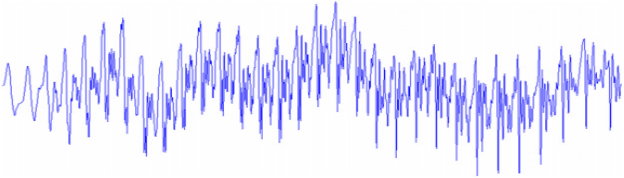
Problems frequently contain a "small parameter" $\epsilon \ll 1$.

- Can be effectively a *large parameter* $\epsilon = \frac{1}{N} \ll 1$.
- Can change the character of the problem

$$i\partial_t u + \epsilon \Delta u + \frac{1}{\epsilon} V u = 0.$$

Some examples:

- In PDEs (e.g. Navier–Stokes, Schrödinger), small terms may give rise to boundary layers or caustics.



- In forced oscillators, small forcings may lead to resonance.

These lectures will provide a toolkit for tackling such problems.

Transforms and integrals

Classical transforms (Fourier, Laplace, Hankel. . .) reduce PDEs to ODEs. The solutions are naturally written as contour integrals,

$$w(z) = \int_C F(z,t)\mathrm{d}t.$$

(Many special functions are also naturally written this way.)

- This offers different ways to study these DEs or functions.
- In particular, we can evaluate integrals asymptotically.
  - E.g. how does
    $$f(x) = \int_C e^{xh(t)}\phi(t)\,\mathrm{d}t$$
    behave as $x \to \infty$?
  - Tools: Watson's lemma; the method of steepest descent.
  - Applications include the Airy function, Bessel functions...

More advanced topics

These two lectures will cover some useful but tangential topics...

- Dimensional analysis and self-similar solutions.
  A set of classical techniques for reducing the complexity of problems and finding informative exact solutions.
- Resummation and improved convergence.
  Some tricks for extracting extra information from the first few terms of a series.

Numerical methods for ODEs

There are lots of ways to solve ODEs numerically.
Unfortunately, the obvious ones aren't always the best...

- The Becker–Döring equation is an innocuous-looking system of ODEs with time scale $O(10^{20})$.
  - I know about Euler's method for ODEs, so use it and let the computer do the leg-work.
  - Euler stability limits the time step to $< 0.2 \implies$ need $O(10^{21})$ steps.
  - CPU time per step ≈ 0.0001 seconds on a powerful workstation.
  - Total computer time is 10^9 years $\gg$ duration of PhD funding.
- A friend suggests Runge–Kutta methods are "much better".
  - It takes a week to modify the code.
  - Now the computer time is only 0.5×10^9 years.
- Ask an expert (not necessarily your supervisor).
  - A good implicit package takes a few minutes.

Numerical methods for stochastic DEs

Stochastic differential equations model systems that evolve subject to some random effects, e.g.:

- Brownian motion of suspended colloidal particles;
- population dynamics and epidemiology;
- behaviour of markets and other financial systems.

This is a growing area of modelling...

The mathematics is not as frightening as it sounds!

- We'll recap probability, random variables etc.
- We take a straightforward approach: "standard calculus plus ε".

Numerical methods for PDEs

Many models in applied mathematics are based on PDEs.

- Many pieces of software claim to be able to solve PDEs.
- Rather too often, their output is pretty but wrong.
- Informed scepticism is required!

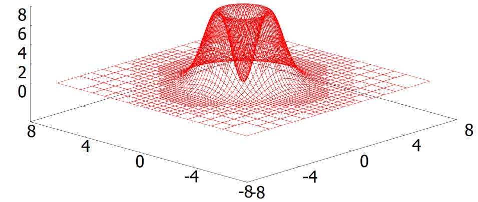
These lectures will look at some of the basic questions:

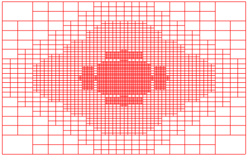
- How can we discretise PDEs to give systems of algebraic equations?
- How can we solve the resulting systems efficiently?
- How do we prove these methods have desirable properties, like
  - consistency (representing the desired PDE)?
  - stability (not being hypersensitive to data)?
  - convergence (error decreasing as resolution increases)?

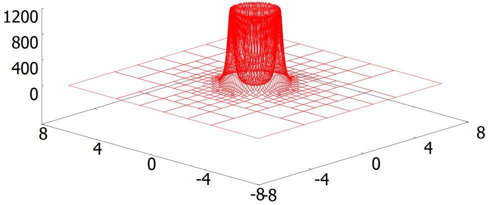
Finite Elements $\rightsquigarrow$ More Advance Techniques!

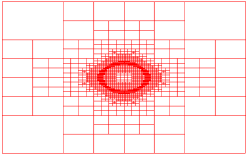
Semilinear parabolic equations: Regional blowup

- $\partial_t u - \Delta u = u^2$ in $\Omega \times (0, T]$, $\Omega = (-8, 8) \times (-8, 8)$
- $u_0(x, y) = 10(x^2 + y^2)e^{-0.5(x^2+y^2)}$ $\rightsquigarrow$ Blowup set: circle centred on the origin









Numerical linear algebra

- If $A \in \mathbb{R}^{n \times n}$ and $\det A \neq 0$, then the solution of $A\underline{x} = \underline{b}$ is

$$\underline{x} = A^{-1}\underline{b}.$$

- This is true, but it does not mean that one must or should work out the inverse of A to solve the linear system.
- Alternatives are Gaussian elimination, LU factorisation, iterative methods...
- Why not use A^{-1}, e.g. for PDE discretisations?
  - It is much more expensive to compute, and
  - it uses much more memory ($\sim n$ times more)

  than alternatives.
- For most problems, it is almost never sensible to work out the matrix inverse.

Summary

This stream provides a toolkit that we can use to obtain approximate solutions to tough problems.

- Asymptotic methods:
  - applicable in extreme cases (small / large parameters).
  - Topics incl. multiple scales; matched asymptotics; contour integrals.
- Numerical methods:
  - applicable most of the rest of the time.
  - Topics incl. ODEs; SDEs; PDEs; linear algebra.

If you have any questions, contact me (ikyza@dundee.ac.uk) or the lecturer for that section of the stream.