MATH3511/6111: Scientific Computing

01. Introduction

Lindon Roberts

Semester 1, 2022

Office: Hanna Neumann Building #145, Room 4.87
Email: lindon.roberts@anu.edu.au
Based on lecture notes written by S. Roberts, L. Stals, Q. Jin, M. Hegland, K. Duru.

Australian National University

Acknowledgements

These lectures are based on previous versions of MATH3511/6111 developed by

- Stephen Roberts
- Linda Stals
- Qinian Jin
- Markus Hegland
- Kenneth Duru

Convenors

This course has two convenors:

- Lindon Roberts (Hanna Neumann Building 4.87, `lindon.roberts@anu.edu.au`)
- Linda Stals (Hanna Neumann Building 2.69, `linda.stals@anu.edu.au`)

I will be teaching weeks 1–6, Linda will take weeks 7–12.

Modern Science

How is science (or engineering, finance, etc.) “done”?

1. Measurements and data
   - Laboratory experiments (e.g. physics, chemistry)
   - Observations (e.g. environmental science, astronomy)
   - Data collection (e.g. finance, machine learning)
2. Models and theory
   - Assumptions about important quantities and their relationships
   - Develop from first principles
   - Validated by data (ideally)
3. Computation
   - Model simulation (i.e. prediction)
   - Fit models to data
   - Optimisation (what inputs to a process give me the result I want?)

Modern Science

Mathematics is the standard language for models and theory in quantitative disciplines.

Scientific computing is the glue between theory and experiments/reality.

Where does scientific computing sit in mathematics?

Theorem (Intermediate Value Theorem)

If $f : \mathbb{R} \to \mathbb{R}$ is continuous and $f(a) < 0 < f(b)$ for some $a < b$, then there exists $c \in (a, b)$ such that $f(c) = 0$.

Theorem (Weierstrass Theorem)

If $f : \mathbb{R}^n \to \mathbb{R}$ is continuous and $A \subset \mathbb{R}^n$ is closed and bounded, then there exists $x \in A$ such that $f(x) \leq f(y)$ for all $y \in A$.

Theorem (Picard-Lindelöf Theorem)

If $f(y, t) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is uniformly Lipschitz continuous in y and continuous in t, then there exists a unique solution $y(t)$ to the ODE $\frac{dy}{dt} = f(y, t)$ with initial condition $y(t_0) = y_0$ in some interval containing t_0.

Scientific Computing in Mathematics

Analysis might tell us a solution to a problem exists, but not what it is or how to find it!

Scientific computing gives us procedures to find (approximate) solutions, with mathematical guarantees of success.

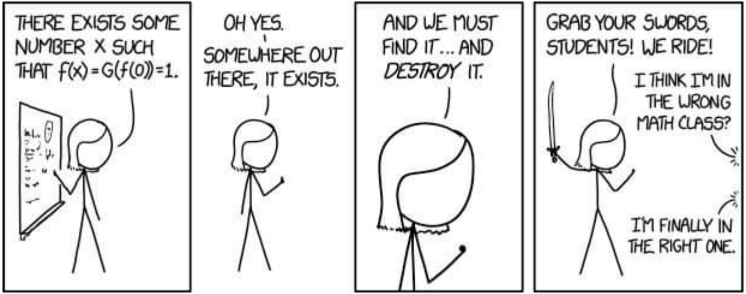

Image: `https://xkcd.com/1856/`

Applications

Scientific computing is important for every quantitative discipline.

Many other applications: weather forecasting, earthquake modelling, tsunami prediction, medical imaging, option pricing, aircraft design, machine learning, ...

The National Computational Infrastructure (Australia's primary supercomputing facility, located at ANU) has been recently used for:

- Genomic analysis for indigenous healthcare
- Simulating melting of ice sheets
- Optimising the design of quantum computers
- Simulating the formation of stars

NCI's Gadi supercomputer (2019) is the fastest computer in Australia and 53rd fastest globally as of November 2021.

Source: `https://nci.org.au/research/research-highlights`. All these examples are from 2021.

My research

My research is in algorithms for optimisation. Here, the underlying mathematical problem is

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}).$$

I have applied optimisation techniques to problems in areas like:

- Machine learning
- Climate modelling
- Image analysis and MRIs
- Forestry

MATH3514 (semester 2, odd-numbered years) covers algorithms for optimisation problems.

Linda's research

Linda's research is in the area of large scale computations, concentrating on finite element methods, adaptive refinement and multigrid methods. To get the best performance out of an algorithm it is necessary to incorporate techniques that are at the forefront of both the mathematical and computer sciences.

Topics that Linda has worked on include

- Ground water flow (application)
- Plasma ion implantation (application)
- Radiation transport equations (application)
- Parallel multigrid methods (numerical technique)
- Fault tolerant iterative solvers (numerical technique)
- Finite element based thin plate splines (numerical technique)

MATH3512 (semester 2, even-numbered years) covers algorithms for the solution of large scale matrix problems.

Course Aims

The aims of the course are:

- Understand a range of important scientific computing techniques
  - Describe key algorithms
  - Understand the mathematical analysis of these algorithms
- Be able to work with floating-point numbers and understand their limitations
- Understand the concepts of consistency, stability and convergence
- Know how to use a common high-level scientific programming language
- Be able to apply numerical algorithms to real-world problems

Course Content

- Solving nonlinear equations
- Function approximation and interpolation
- Methods for calculating derivatives
- Floating-point arithmetic
- Methods for calculating integrals [roughly end of week 6]
- Conditioning and stability
- Solution of linear systems
- Discrete Fourier Transform
- Solution of differential equations

Textbook: no assigned textbook (lectures are self-contained), but see Wattle for some good suggestions.

Prerequisites: MATH2305 or MATH1116 (i.e. good knowledge of calculus, linear algebra, possibly differential equations).

Course Content

"29 major developments in scientific computing" (L. N. Trefethen, Oxford)

Newton's method	Simplex algorithm	Spectral methods
Least-squares fitting	Conjugate gradients and Lanczos	MATLAB
Gaussian elimination	Fortran	Multigrid methods
Gauss quadrature	Stiff ODE solvers	IEEE arithmetic
Adams formulae	Finite Elements	Nonsymmetric Krylov iterations
Runge-Kutta formulae	Orthogonal Linear Algebra	Interior point methods
Finite differences	QR algorithm	Fast multipole methods
Floating-point arithmetic	Fast Fourier Transform	Wavelets
Splines	Quasi-Newton iterations	Automatic differentiation
Monte Carlo methods	Preconditioning	

`https://people.maths.ox.ac.uk/trefethen/inventorstalk.pdf`

Course Content

"29 major developments in scientific computing" (L. N. Trefethen, Oxford)

Newton's method	Simplex algorithm	Spectral methods
Least-squares fitting	**Conjugate gradients** and Lanczos	~~MATLAB~~
Gaussian elimination	~~Fortran~~	Multigrid methods
Gauss quadrature	Stiff ODE solvers	**IEEE arithmetic**
Adams formulae	Finite Elements	Nonsymmetric Krylov iterations
Runge-Kutta formulae	Orthogonal Linear Algebra	Interior point methods
Finite differences	QR algorithm	Fast multipole methods
Floating-point arithmetic	**Fast Fourier Transform**	Wavelets
Splines	Quasi-Newton iterations	Automatic differentiation
Monte Carlo methods	Preconditioning	

We will cover **10.5**+4 topics in MATH3511 (6.5 more in MATH3512 and MATH3514)

Software

Being able to use and write software for scientific computing is an essential part of this course.

We will using Python, a high-level programming language popular for scientific computing.

- I recommend using Anaconda (`https://www.anaconda.com/products/individual`)
  - Contains Python plus many standard packages for scientific computing.
  - Free, so you can install it on your own computer, and available on all ANU computers.
  - Make sure you install Python 3, not Python 2 (no longer maintained).
- Write code in Spyder or using Jupyter notebooks (both included in Anaconda), or PyCharm for package development.

Python is probably the most-used open source programming language for scientific computing and data science. Other common choices are MATLAB (commercial), R (statistics) or Julia (new). Knowing one language makes it *much* easier to learn others.

Format

The contact hours for the course are

- 3x1hr lectures per week (Mon 12-1, Thu 12-1, Fri 4-5)
- Weekly 2hr computer labs (starting in week 3) — work on lab books (assessed!) and theory questions

The course will run as dual delivery: lectures are in-person but streamed & recorded on Echo360, labs will be in-person or over Zoom. Per ANU guidelines, we expect you to attend in-person whenever possible.

Communication

- Check Wattle for all announcements and resources
- Please post general questions about admin or course content in the Wattle discussion forum
- Office hours:
  - Lindon: Thu 1-2 after the lecture (my office + Zoom), weeks 1–6
  - Linda: TBC (check Wattle), weeks 7–12
- Send Linda/me an email for confidential matters (`lindon.roberts@anu.edu.au`, `linda.stals@anu.edu.au`)
- Course reps for anonymous feedback or general class issues you don't want to raise yourself (contact details on Wattle)

Course Reps

We need ~ 2 course representatives — if you are interested can you please email me (`lindon.roberts@anu.edu.au`) this week.

Assessment

Assessment

- Assignments 30%: 5 assignments, 6% each (due weeks 4, 6, 8, 10, 12)
- Lab books 30%: book 0 not graded, books 1–5 worth 6% each (due weeks 3, 5, 7, 9, 11, study week)
- Final exam 40%
- Hurdle requirements: submit lab book 0 (a reasonable attempt), and $\geq 40\%$ in the exam

Late submissions allowed for up to 1 week only (5% per day or extension). If you can't submit for good reasons (e.g. medical), we will adjust your grade accordingly. Assignments/labs can be worked on in groups, but all write-ups must be original and your own, including all code.

Honours Pathway Option (MATH3511 only): a more advanced/theoretical option for interested students (with alternative assignments and final exam). Prerequisites: STAT2001 or $\geq 60\%$ in MATH2305 or MATH1116. Email me if you want to take the HPO.