MATH50003
Numerical Analysis

https://github.com/Imperial-MATH50003/MATH50003NumericalAnalysis

Office Hour: ~~[illegible]~~, Huxley 6M40

TBC

Dr Sheehan Olver

What is Numerical Analysis?

Algorithms for continuous problems

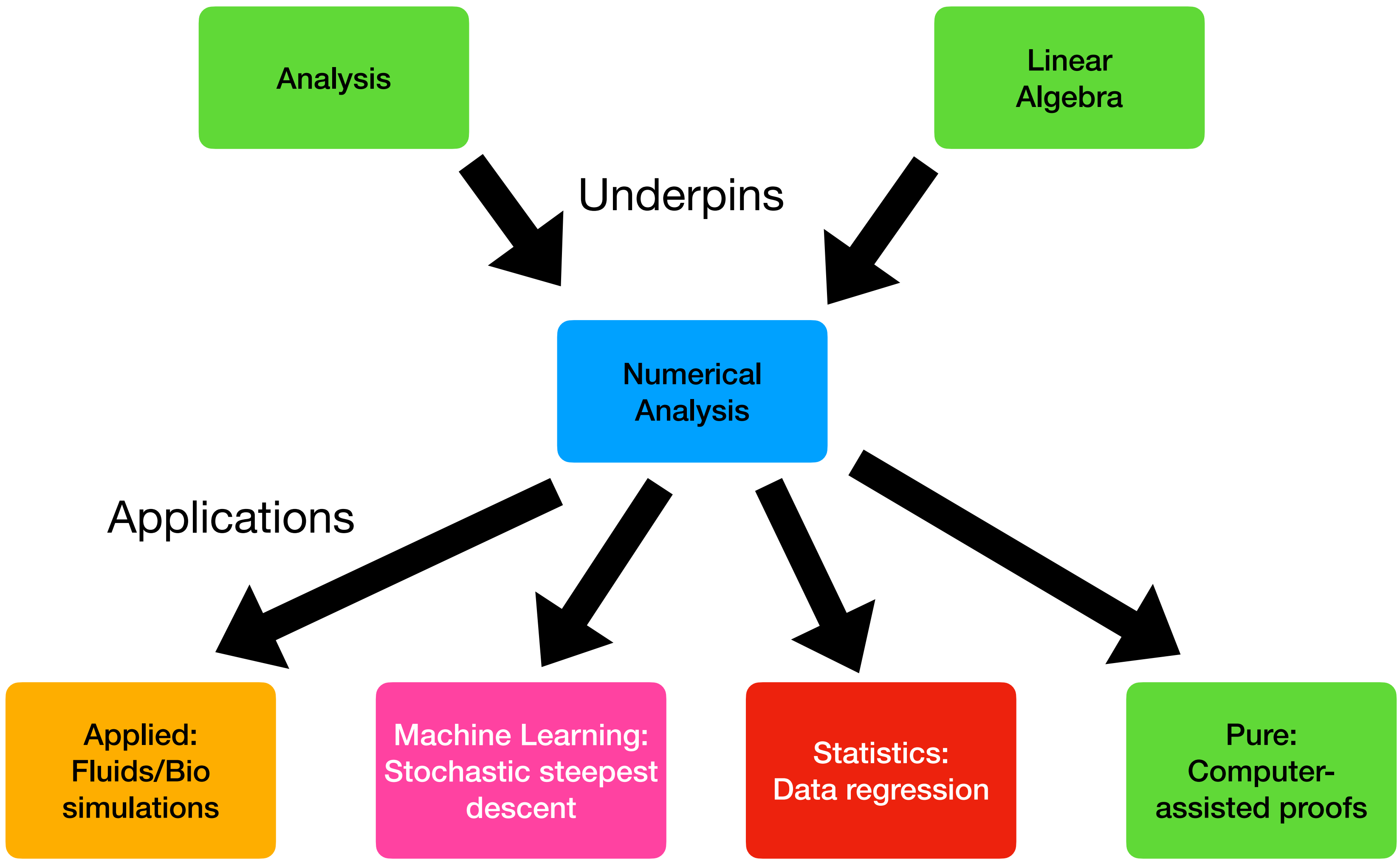
Implementation in software

Numerics

Analysis of convergence and stability

error → 0

reliable


Analysis
Linear Algebra
Underpins
Numerical Analysis
Applications
Applied: Fluids/Bio simulations
Machine Learning: Stochastic steepest descent
Statistics: Data regression
Pure: Computer-assisted proofs

Who am I?

[Dr Sheehan Olver]

- PhD in Cambridge followed by
  Junior Research Fellow at St. John’s College, Oxford
- Imperial since 2017
- Researcher in numerical analysis / scientific computing studying:
  - Computational complex analysis
  - Random matrix theory
  - Partial/fractional differential equations
- Won the Adam’s Prize in 2012 for developing numerical methods for
  Riemann–Hilbert problems

Course content

computer-based assessment

I. Calculus on a Computer
- Integration, differentiation, root finding

II. Representing Numbers
- Floating point numbers, bounding errors, interval arithmetic

III. Numerical Linear Algebra
- Data regression, differential equations, least squares

IV. Approximation Theory
- Fourier series, orthogonal polynomials, Gaussian quadrature

Assessment

Computer-based

- Labs
- Practice Computer-based Exam
- Computer-based Exam

Pen-and-paper

- Problem sheets
- Final Exam

Submit labs/problem sheets to venkata.melanathuru19@imperial.ac.uk
for informal marking by GTAs

Julia is a programming language designed by MIT for Scientific Computing, Numerical Analysis and Machine Learning

Compiled: generates efficient high performance code and allows us to see what the computer is actually doing

Easy to add custom types to understand mathematical concepts

Course website

https://github.com/Imperial-MATH50003/MATH50003NumericalAnalysis

Part I

Calculus on a Computer

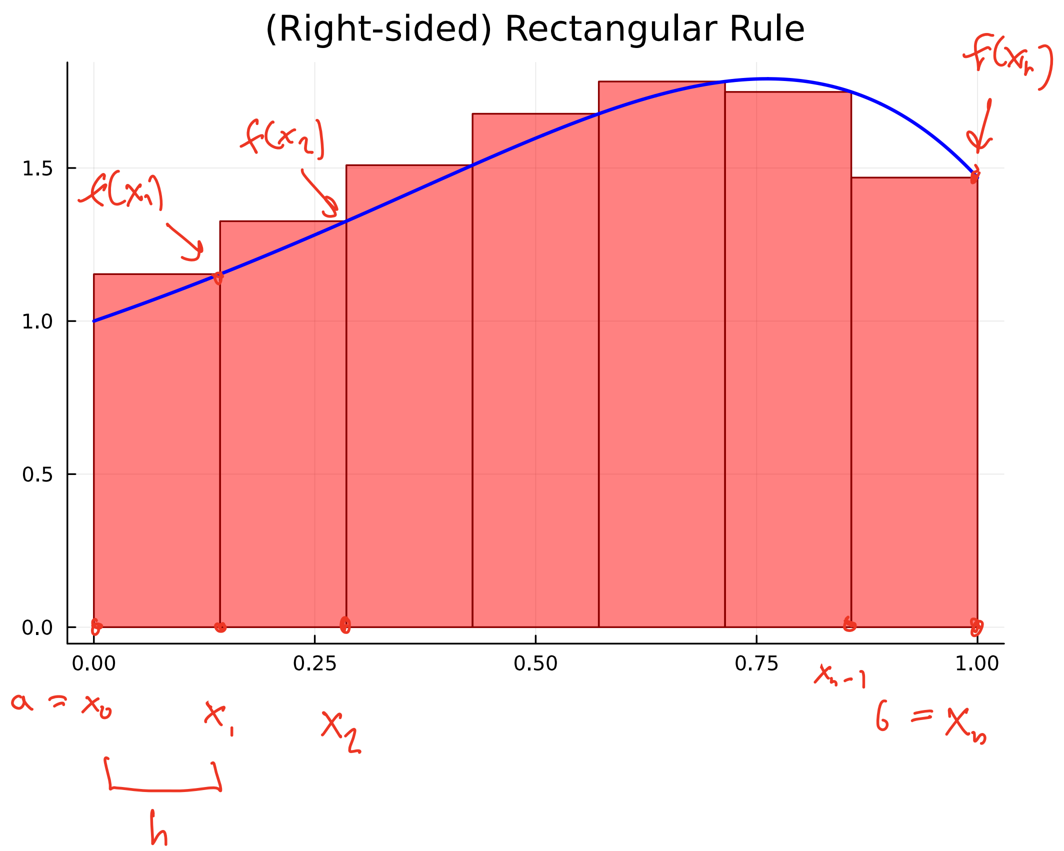
1. Rectangular rules for integration
2. Divided differences for differentiation
3. Dual numbers for differentiation

$$\int_a^b f(x)\mathrm{d}x = \lim_{n\to\infty} h \sum_{j=1}^{n} f(x_j)$$

where

$$\mathrm{h} = \frac{\mathrm{b-a}}{\mathrm{n}}$$

$$\mathrm{x_j = a + jh}$$



Idea: make n big so that

$$\int_a^b f(x)\,dx \approx h \sum_{j=1}^{n} f(x_j).$$

How accurate n?

Lemma 1 ((Right-sided) Rectangular Rule error on one panel). *Assuming f is differentiable we have*

$$\int_a^b f(x)\mathrm{d}x = \underbrace{(b-a)f(b)}_{\text{approx.}} + \underbrace{\delta}_{\text{error}}$$

where $|\delta| \le M(b-a)^2$ *for* $M = \sup_{a\le x\le b}|f'(x)|$.

Proof

$$\int_a^b f(x)\,dx = \int_a^b (x-a)' f(x)\,dx$$

$$= \Big[(x-a)f(x)\Big]_a^b - \int_a^b (x-a)f'(x)\,dx$$

$$= (b-a)f(b) + \underbrace{\left(-\int_a^b (x-a)f'(x)\,dx\right)}_{\delta}$$

$-$

where

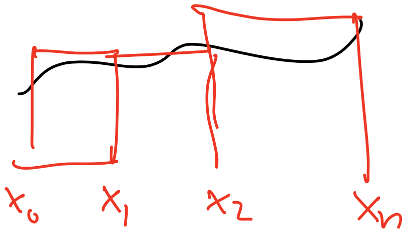
$$|\delta| \leq \int_a^b |x-a| \, |f'(x)| \, dx$$

$$\leq \underbrace{(b-a)}_{\text{width}} \sup_{a \leq x \leq b} |x-a| \, |f'(x)|$$

$$\leq (b-a) \underbrace{\sup |x-a|}_{=(b-a)} \underbrace{\sup |f'(x)|}_{M}$$

$$= (b-a)^2 M$$

$\boxed{M}$



Theorem 1 (Rectangular Rule error). *Assuming f is differentiable we have*

$$\int_a^b f(x)\mathrm{d}x = h\sum_{j=1}^{n} f(x_j) + \delta$$

where $|\delta| \le M(b-a)h$ *for* $M = \sup_{a \le x \le b} |f'(x)|$, $h = (b-a)/n$ *and* $x_j = a + jh$.

Proof

$$\int_a^b f(x)\,dx = \left(\int_{x_0}^{x_1} + \int_{x_1}^{x_2} + \cdots + \int_{x_{n-1}}^{x_n} \right) f(x)\,dx$$

$$= \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} f(x)\,dx$$

$$= \sum_{j=1}^{n} \left[(x_j - x_{j-1}) f(x_j) + \delta_j \right]$$

Lemma 1

where $|\delta_j| \le (x_j - x_{j-1})^2 \sup_{x_{j-1} \le x \le x_j} |f'(x)|$

$$\le h^2 M$$

$$= h \sum_{j=1}^{n} f(x_j) + \underbrace{\sum_{j=1}^{n} \delta_j}_{\delta}$$

total error

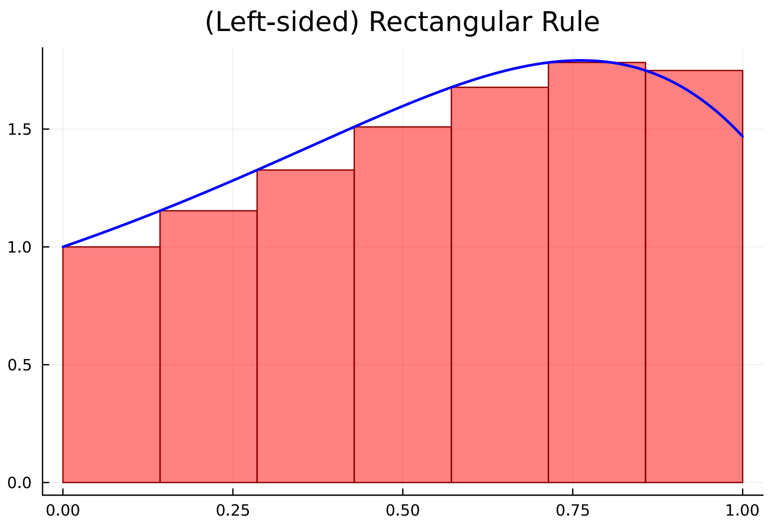
where

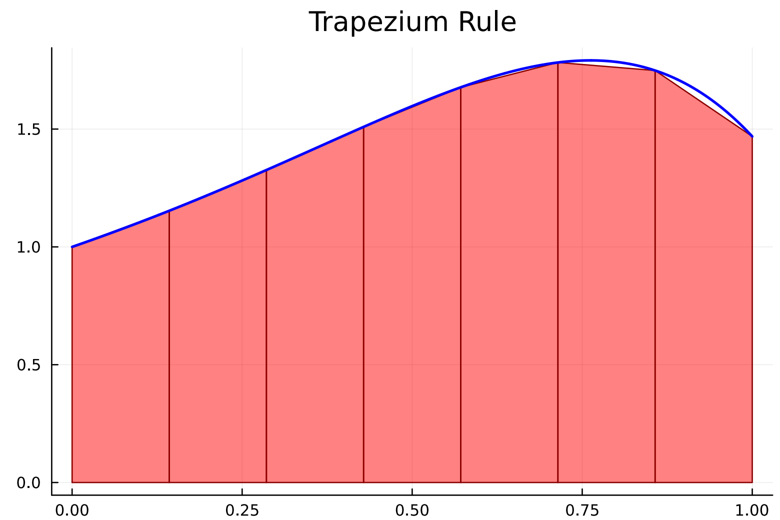
$$|\delta| \le \sum_{j=1}^{n} |\delta_j| \le \sum_{j=1}^{n} M h^2 = M n \underbrace{h^2}_{\frac{(b-a)^2}{n^2}}$$

$$= \underbrace{M(b-a)h}_{\to 0}$$

$h \to 0$

Other Approximations





$$h \sum_{j=0}^{n-1} f(x_j)$$

$$h\left[\frac{f(x_0)}{2} + \sum_{j=1}^{n-1} f(x_j) + \frac{f(x_n)}{2}\right]$$

How to do it in practice?

Three setup steps

Warning: don't install from package manager

1. Download julia

2. Download course content on Git from
https://github.com/Imperial-MATH50003/MATH50003NumericalAnalysis

3. Open Lab 1 in Jupyter