MATH50003 Numerical Analysis

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

Office Hour: Mondays 4pm, Huxley 6M40

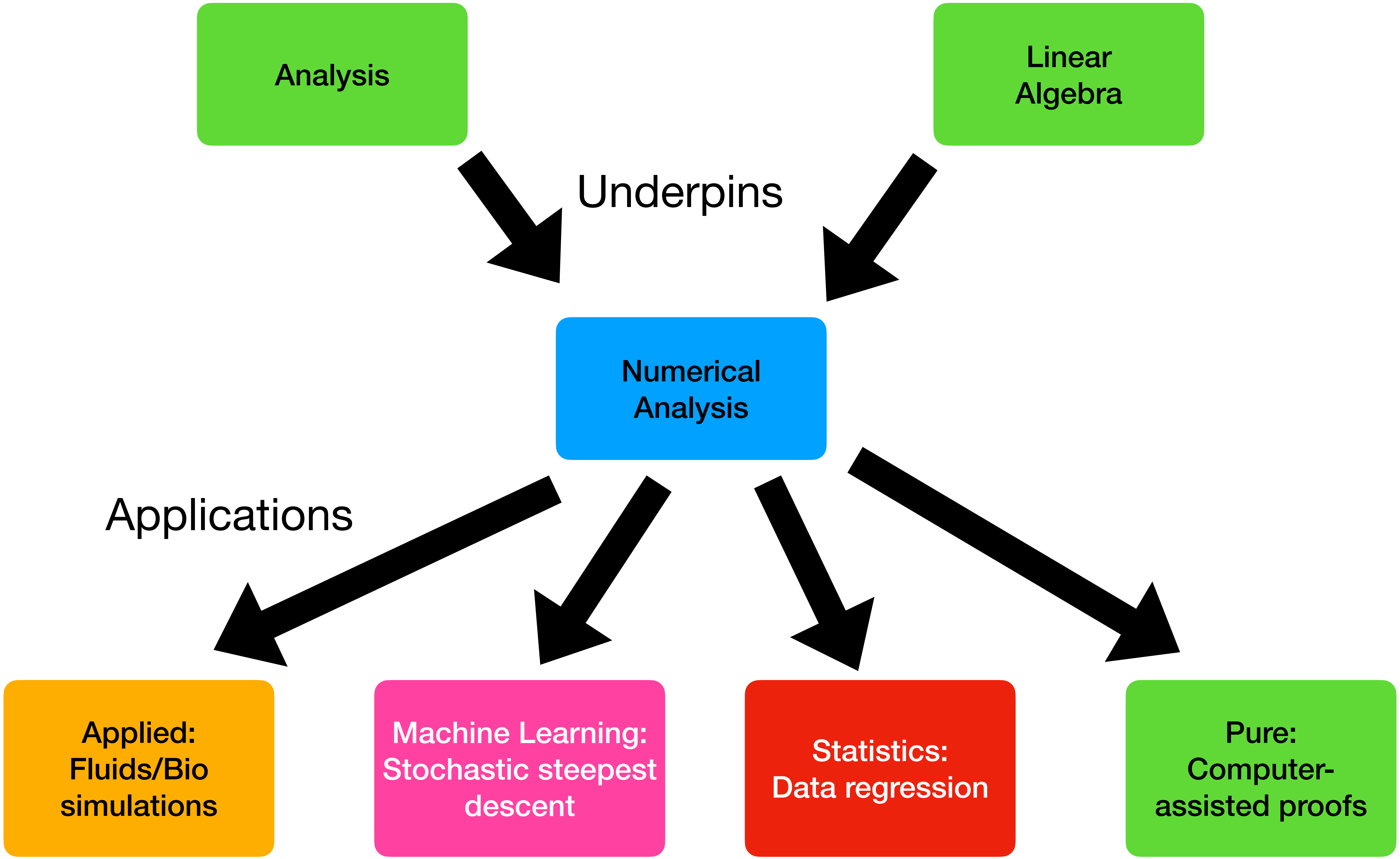
Dr Sheehan Olver

What is Numerical Analysis?

Algorithms for continuous problems

Implementation in software

Analysis of convergence and stability



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

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

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

III. Numerical Linear Algebra
   - Structured matrices, LU & QR factorisations, least squares

IV. Linear Algebra Applications
   - Data regression, differential equations

V. Numerical Fourier series
   - Fourier expansions and transforms, differential equations with periodic conditions

VI. Orthogonal Polynomials
   - Classical 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 maria.ricciuti18@imperial.ac.uk
for informal marking by GTAs

julia

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

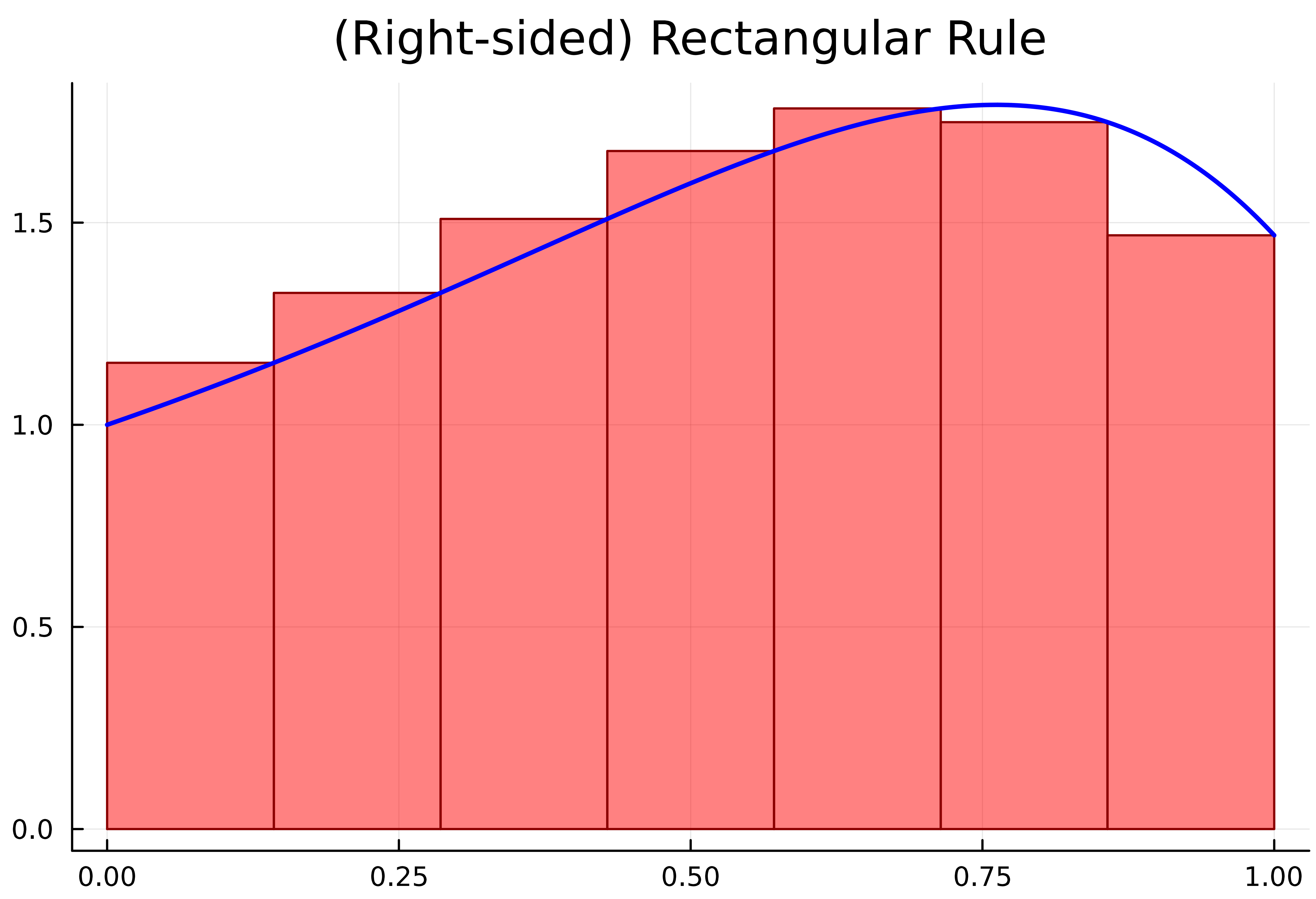
1. Rectangular rules for integration
2. Divided differences for differentiation
3. Dual numbers for differentiation
4. Newton’s method for root finding

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

where

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

$$x_j = a + jh$$



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

$$\int_a^b f(x)\mathrm{d}x = (b-a)f(b) + \delta$$

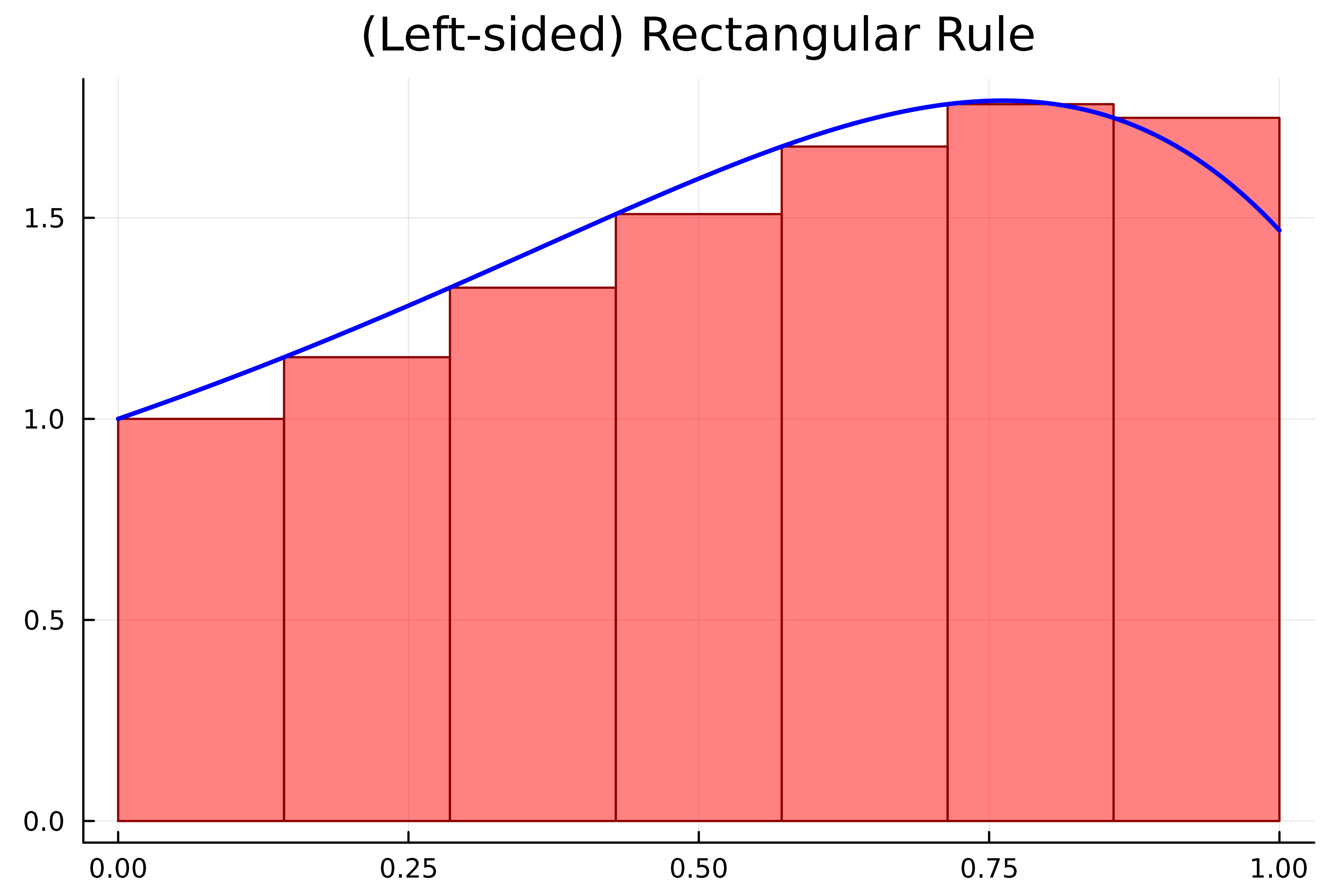
where $|\delta| \leq M(b-a)^2$ *for* $M = \sup_{a \leq x \leq b} |f'(x)|$.

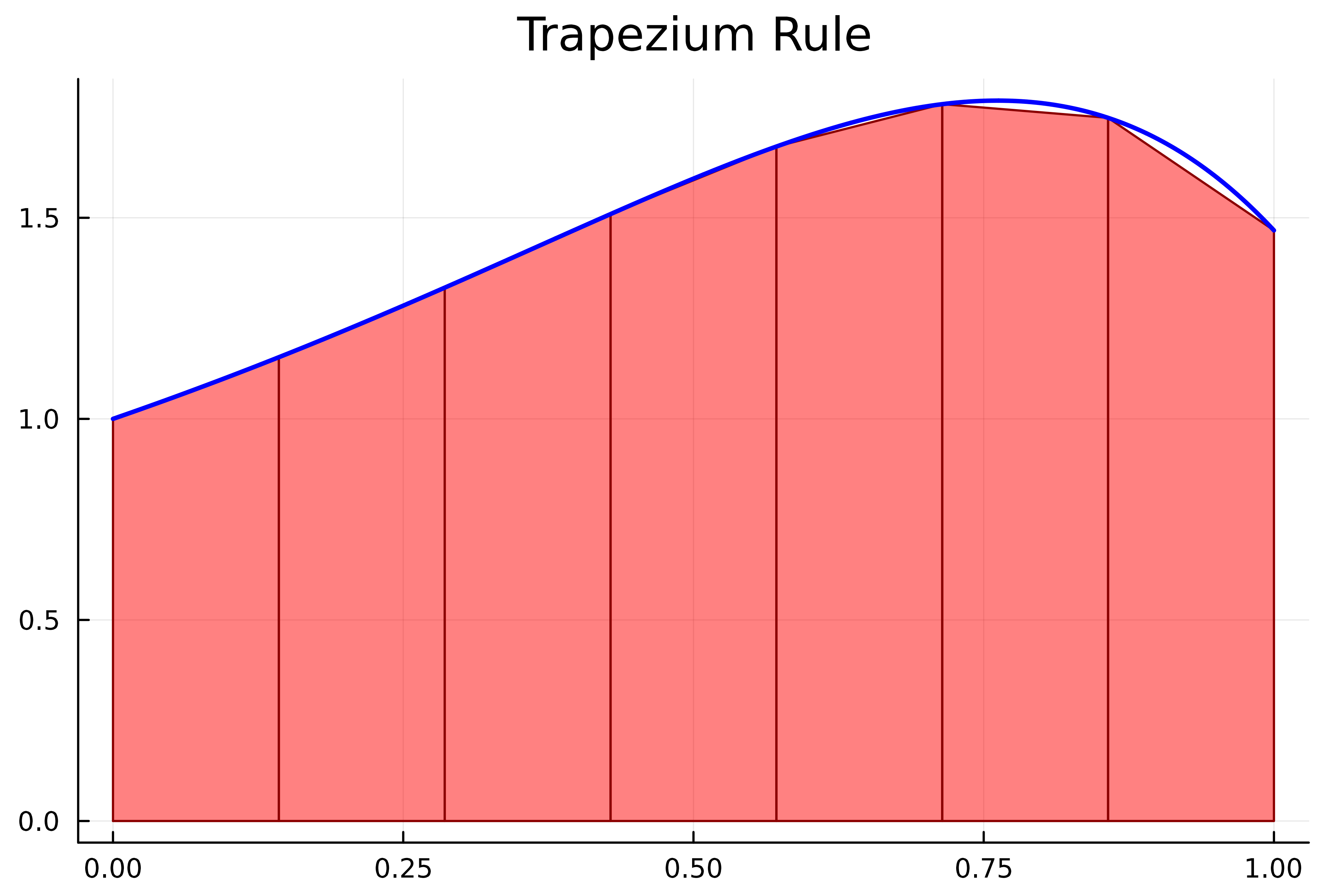
Theorem 1 (Rectangular Rule error). *Assuming f is differentiable we have*

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

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

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

1. Download julia

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

3. Open Lab 1 in Jupyter