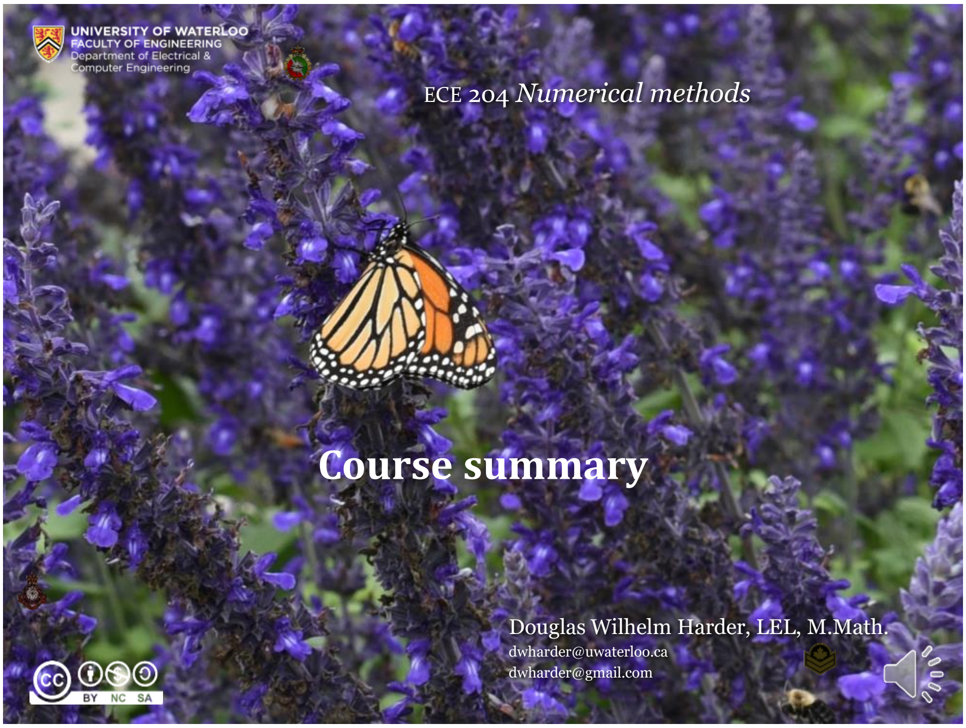

1

Introduction

- In this topic, we will
  - Review this course in numerical analysis
  - Describe the material covered
  - Highlight the interest aspects involved

2

2

Course summary

- In this course, we:
  - Observed that there were issues with representing real numbers by finite-precision floating-point numbers
  - Introduced a number of tools that would be used for approximating solutions to numerical problems
  - We then looked at four different categories of numerical problems:
    - Evaluating the value of an expression
    - Approximating solutions to algebraic equations
    - Approximating solutions to analytic equations
    - Unconstrained optimization

3

3

1. Course introduction

- We started by:
  - Defining absolute error, relative error and percent relative error
  - Looked at the decimal and binary representation of real numbers
  - Defined rounding a value to a given number of digits or bits
  - Defined the measure of significant digits
  - Defining accuracy and precision
    - Accuracy: yielding a result close to the correct value
    - Precision: yielding results with a consistent number of significant digits

4

4

2. Representation of real numbers

- Next we considered the representation of real numbers on the computer
  - We considered fixed-point representations of integers and real numbers
  - We considered the benefits of a floating-point representation that parallels scientific notation
  - We looked at a six-decimal-digit floating-point representation followed by the binary double-precision floating-point representation
    - We considered arithmetic operations and comparisons
  - We looked at loss of precision, subtractive cancellation and the loss of associativity: $x + (y + z)$ may not equal $(x + y) + z$
  - Finally, we looked at infinity, not-a-number (NaN) and denormalized numbers

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  1. Averages, weighted averages and convex combinations
     - Sampling n uniformly random samples from $[a, b]$, a weighted average of the minimum and maximum values gave a better estimate of a and b than any other value
     - We saw different weighted averages of points on a function gave a better estimation of an integral than others
       - Later we saw that this is because the weights $\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$ were a consequence of finding interpolating quadratic polynomials

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  2. Iteration and the fixed-point theorem
     - We saw an iterative algorithm that appeared to converge to the square root of two
       – Later we saw this was a special case of Newton's method applied to $x^2 - 2 = 0$:

$$x - \frac{x^2 - 2}{2x} = \frac{x}{2} + \frac{1}{x}$$

     - We described the fixed-point theorem:
       – Given an x_0 and a function $f(x)$, if we define the iterative procedure that $x_{k+1} \leftarrow f(x_k)$, then if this converges, it converges to a solution of the equation $x = f(x)$

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  3. Linear algebra
     - We described Gaussian elimination with partial pivoting
       – An algorithm that allows us to solve a system of linear equations that mitigates the finite precision of floating-point representations
     - We looked at an iterative technique to approximate a system of linear equations
     - We saw some systems of linear equations will result in poor numerical approximations no matter what algorithms we use
       – These were described using the *condition number* and large condition numbers implied ill conditioning of the system

Course summary

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  4. Interpolation
     - We saw linear, quadratic and then polynomial interpolation
     - We noted that these were found by solving a system of linear equations
       - We defined the Vandermonde matrix
       - We saw that it was easy to set up try to interpolate points that resulted in poor interpolating polynomials if we did not use partial pivoting
     - We found that shifting towards the origin and scaling resulted in a Vandermonde matrix with a smaller condition number
     - We saw that equally spaced points tended to have Vandermonde matrices with smaller condition numbers

9

9

Course summary

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  5. Taylor series
     - We revisited Taylor series from first year, and introduced an alternative notation:

$$f(x+h) = f(x) + f^{(1)}(x)h + \frac{1}{2} f^{(1)}(x)h^2 + \cdots + \frac{1}{n!} f^{(n)}(x)h^n + \frac{1}{(n+1)!} f^{(n+1)}(\xi)h^{n+1}$$

     - We introduced the idea of using big-Oh notation to represent the error
       - The above n^{th}-order Taylor series approximation has an error that is $O(h^{n+1})$

10

10

Course summary

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  6. Bracketing
     - We described the idea of bracketing a solution
     - We looked at how a binary search brackets what we are looking for when we are searching for a value in an array
     - We introduced the concept of an interpolation search
     - Later, we saw
       – The bisection method, which parallels a binary search
       – The bracketed secant method, which parallels the interpolation search

11

11

Course summary

3. Tools for developing numerical algorithms

- Next, we looked at seven tools that we would use either to develop or analyze numerical algorithms, or describe patterns that would appear repeatedly in our algorithms
  7. Intermediate-value theorem
     - Given a continuous function f defined on $[a, b]$, and given a value of y that falls between $f(a)$ and $f(b)$, then there is an $a < x < b$ such that $y = f(x)$
     - Consequently, given n x-values on $[a, b]$, then given a convex combination of f evaluated at these n values must have an $a \le \xi \le b$ such that $f(\xi)$ equals this convex combination

12

12

4. Sources of error

- Following this, we did a deep dive into looking at and categorizing sources of error in modelling and numerical algorithms
  - Random error: errors resulting from the inability to make perfectly precise and accurate measurements
    - These can be described through statistical distributions
      - We saw normally, uniformly and Poisson distributed errors
  - Systematic errors: errors that we can measure and compensate for in our models
- We then looked at sources of error:
  - Floating-point truncation error due to finite precision
  - Model, environmental, drift, production and calibration errors
- We described how averaging repeated readings will minimize random errors

13

13

Categories of numerical problems

- We then looked at four categories of numerical problems:
  1. Approximating the values of expressions
  2. Approximating the solutions to algebraic equations and systems of algebraic equations
     - This linear equations and systems of linear equations together with non-linear equations and systems of non-linear equations which are reduced to root-finding problems
  3. Approximating the solutions to analytic equations and systems of analytic equations
     - This includes differential, integral, and integro-differential equations, the latter two of which are reduced to differential equations
  4. Approximating solutions to unconstrained optimization problems

14

14

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - We started by seeing that Horner's rule is the most efficient and accurate algorithm for evaluating a polynomial at a point
  - The run time is O(n) where n is the degree of the polynomial
    - This requires n additions and n multiplications

15

15

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Next, we considered polynomials interpolating equally-spaced points as a basis of evaluation
    - We approximated the value of a function at a point between the given equally-spaced points
    - Estimating the derivative at a point
    - Estimating an integral between two points
  - A reminder: we had to solve the system with the Vandermonde matrix $V\mathbf{a} = \mathbf{y}$

16

16

Course summary

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Next, we considered polynomials interpolating equally-spaced points as a basis of evaluation
    - To evaluate an interpolating polynomial between two or four points, we shifted and scaled the *x*-values to the points
      $-0.5, 0.5$ or $-1.5, -0.5, 0.5, 1.5$,
      respectively, and evaluated the polynomial at $-0.5 < \delta < 0.5$
    - To evaluate an interpolating polynomial between the most recent of a sequence of points, shifting and scaling the *x*-values to the points
      $-1.5, -0.5, 0.5$ or $-2.5, -1.5, -0.5, 0.5$,
      respectively, but still evaluating the polynomial at $-0.5 < \delta < 0.5$

17

17

Course summary

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Next, we considered polynomials interpolating equally-spaced points as a basis of evaluation
    - We found we could approximate the derivative:

$$f^{(1)}(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

$$y^{(1)}(t) = \frac{y(t) - y(t-h)}{h} + O(h)$$

$$y^{(1)}(t) = \frac{3y(t) - 4y(t-h) + y(t-2h)}{2h} + O(h^2)$$

Centered and backward divided-difference formulas

    - We could also approximate the second derivative:

$$f^{(2)}(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

$$y^{(2)}(t) = \frac{y(t) - 2y(t-h) + y(t-2h)}{h^2} + O(h)$$

$$y^{(2)}(t) = \frac{2y(t) - 5y(t-h) + 4y(t-2h) - y(t-3h)}{h^2} + O(h^2)$$

18

18

Course summary

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Next, we considered polynomials interpolating equally-spaced points as a basis of evaluation
    - We found we can also approximate integration across an interval

$$\int_a^{a+h} f(x)\,\mathrm{d}x = \frac{f(a)+f(a+h)}{2}h + O(h^3)$$

$$\int_a^{a+h} f(x)\,\mathrm{d}x = \frac{f(a)+4f(a+\tfrac{1}{2}h)+f(a+h)}{6}h + O(h^5)$$

$$\int_a^{a+h} f(x)\,\mathrm{d}x = \frac{-f(a-h)+13f(a)+13f(a+h)-f(a+h)}{24}h + O(h^5)$$

    - We could also use previous information

$$\int_{b-h}^{b} f(t)\,\mathrm{d}t = \frac{-y(b-2h)+8y(b-h)+5y(b)}{12}h + O(h^4)$$

$$\int_{b-h}^{b} f(t)\,\mathrm{d}t = \frac{y(b-3h)-5y(b-2h)+19y(b-h)+9y(b)}{24}h + O(h^5)$$

    - Composite applications reduced *h*,
      but also reduced the error term by one

19

19

Course summary

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Finally, we considered least-squares best-fitting polynomials on equally-spaced points as a basis of evaluation
    - We approximated the value of a function at a point
    - Estimating the derivative at a point
    - Estimating an integral between two points
  - A reminder: We had to create a Vandermonde matrix, but only with columns corresponding to terms we were interested in
    - We then solved $V^{\mathrm{T}}V\mathbf{a} = V^{\mathrm{T}}\mathbf{y}$

20

20

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Finally, we considered least-squares best-fitting polynomials on equally-spaced points as a basis of evaluation
    - To evaluate the best-fitting polynomial between $n + 1$ points, we shifted and scaled the x-values to the points
      $$-n, -n + 1, \ldots, -3, -2, -1, 0$$
      respectively, and evaluated the polynomial at $-1 < \delta < 1$
    - We looked at linear and quadratic best-fitting polynomials
    - We found we could update the polynomials in O(1) time

21

21

Course summary

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Finally, we considered least-squares best-fitting polynomials on equally-spaced points as a basis of evaluation
    - We found we could approximate the derivative:
      - Around the most recent point $t_k + \delta h$, the approximation of the derivative was either

$$\frac{a_1}{h} \text{ or } \frac{2a_2\delta + a_1}{h}$$

22

22

5. Approximating the values of expressions

- Our first class of problems was evaluating an expression to a value
  - Finally, we considered least-squares best-fitting polynomials on equally-spaced points as a basis of evaluation
    - We also found we could approximate the integral:
      - Either over the most recent interval $[t_{k-1}, t_k]$, the approximation of the integral was either

$$\left(a_0 - \frac{a_1}{2}\right)h \quad \text{or} \quad \left(a_0 - \frac{a_1}{2} + \frac{a_2}{3}\right)h$$

      - Or estimating over the next interval $[t_k, t_{k+1}]$, the approximation of the integral was either

$$\left(a_0 + \frac{a_1}{2}\right)h \quad \text{or} \quad \left(a_0 + \frac{a_1}{2} + \frac{a_2}{3}\right)h$$

6. Approximating solutions to algebraic equations and systems of algebraic equations

- Our second class of problems was approximating solutions to algebraic equations and systems of algebraic equations
  - We started by seeing that the quadratic formula is not necessarily the best formula to find the roots of a quadratic polynomial
  - We transformed a non-linear algebraic equation to a root-finding problem in one variable
  - We saw more advanced techniques for iterative algorithms for approximating solutions to systems of linear equations
  - We then generalized Newton's method for approximating a solution to a system of non-linear equations

6. Approximating solutions to algebraic equations and systems of algebraic equations

- Our second class of problems was approximating solutions to algebraic equations and systems of algebraic equations
  - For the quadratic formula, we saw there were two possible formulations

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

25

25

Course summary

6. Approximating solutions to algebraic equations and systems of algebraic equations

- Our second class of problems was approximating solutions to algebraic equations and systems of algebraic equations

  - After having transformed a non-linear algebraic equation to a root-finding problem in one variable, we looked at:
    - Newton's method
    - The bisection method
    - The bracketed secant method
    - The secant method
    - Muller's method
    - Inverse quadratic interpolation
    - The Brent-Dekker method

26

26

Course summary

6. Approximating solutions to algebraic equations and systems of algebraic equations

- Our second class of problems was approximating solutions to algebraic equations and systems of algebraic equations
  - The two more advanced techniques for iterative algorithms for approximating solutions to systems of linear equations were
    - Gauss-Seidel method
    - The method of successive over-relaxation

27

27

Course summary

6. Approximating solutions to algebraic equations and systems of algebraic equations

- Our second class of problems was approximating solutions to algebraic equations and systems of algebraic equations
  - We then generalized Newton's method for approximating a solution to a system of non-linear equations
    - We converted the non-linear root-finding problem into a linear root-finding problem, which we can solve with linear algebra

$$\mathbf{J}(\mathbf{f})(\mathbf{u}_k)\Delta\mathbf{u}_k = -\mathbf{f}(\mathbf{u}_k)$$

    - Consequently, we let

$$\mathbf{u}_{k+1} \leftarrow \mathbf{u}_k + \Delta\mathbf{u}_k$$

28

28

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  - We reduced analytic equations to differential equations
  - We started by approximating solutions to initial-value problems (IVPs)
  - We continued by approximating solutions to boundary-value problems (BVPs)
  - We then proceeded to approximate solutions to problems in partial differential equations (PDEs)

29

29

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  - We started with approximations to 1st-order IVPs using solvers similar to that of composite integration:
    - Euler's method
    - Heun's method
    - The 4th-order Runge-Kutta method
  - Again, one step was order 2, 3, and 5, respectively, but multiple steps reduced the order by one
  - We discussed how we could use cubic splines to approximate solutions between the points of our approximation

30

30

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  – We proceeded to find adaptive techniques for approximating solutions to 1^{st}-order IVPs
    - Our Euler-Heun method
    - The Dormand-Prince method
  – These techniques picked an appropriate step size at each iteration to ensure our desired accuracy

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  – We then proceeded to find solvers for systems of 1^{st}-order IVPs:
    - We saw that all previous techniques trivially extend to approximating such systems by simply using vectors instead of scalars
  – We then proceeded to find solvers for higher-order IVPs:
    - We saw we could convert an n^{th}-order IVP into a system of n 1^{st}-order initial-value problems
  – We concluded that we could convert a system of higher-order IVPs into a system of 1^{st}-order IVPs

Course summary

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  - We continued by approximating solutions to BVPs:
    - We started with the shooting method
      - This transformed our BVPs into a system of two IVPs which we then iterated
    - We continued with using our divided-difference approximations of the derivative and second derivative to create a system of linear equations
      - We called this a finite-difference method
  - We also introduced Neumann, and specifically insulated, boundary conditions

33

33

Course summary

7. Approximating solutions to analytic equations and systems of analytic equations

- Our third class of problems was approximating solutions to analytic equations and systems of analytic equations
  - We concluded by approximating solutions to PDEs:
    - Starting in one dimension, we approximation solutions to
      - The heat equation
      - The wave equation
    - We then went to two and three dimensions, discussing solutions to Laplace's equation
      - This resulted in a system of linear equations similar to that of BVPS
    - We then discussed approximating solutions in two and three dimensions to
      - The heat equation
      - The wave equation

34

34

8. Optimization

- Our fourth-and-last class of problems was approximating solutions optimization problems
  - We restricted ourselves to unconstrained optimization
    - We reduced the problem of finding maxima to finding minima, so we focused on finding local minima to functions
  - We started with approximating minima to real-valued functions of a real variable
  - We then continued by approximating minima to real-value functions of a vector variable

Course summary

8. Optimization

- Our fourth-and-last class of problems was approximating solutions optimization problems
  - To find the minima of a real-valued function of a real variable, we looked at the following algorithms:
    - Step-by-step optimization
    - Newton's method for finding extrema
    - The golden-ratio search
    - Successive parabolic interpolation
    - Brent's method

8. Optimization

- Our fourth-and-last class of problems was approximating solutions optimization problems
  - To find the minima of a real-valued function of a vector variable, we looked at the following algorithms:
    - The Hooke-Jeeves method
    - Neweton's method for finding extrema in n dimensions
    - Gradient descent
      - This converted an n-dimensional problem into a 1-dimensional problem

37

37

Summary

- Following this course, you now
  - Are aware that there are issues when we represent real numbers as finite-precision floating-point numbers
  - Understand there are tools we can use to mitigate the issues that arise when using floating-point numbers
  - Have seen the application of these tools to approximating solutions to the following classes of problems that have numerical solutions:
    - Evaluating an expression
    - Algebraic equations and systems thereof
    - Analytic equations and systems thereof
    - Optimization problems

38

38

References

[1] https://en.wikipedia.org/wiki/Numerical_analysis

39

39

Acknowledgments

None so far.

40

40

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.
Mathematical equations are prepared in MathType by Design Science, Inc.
Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

41

41

Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

42

42