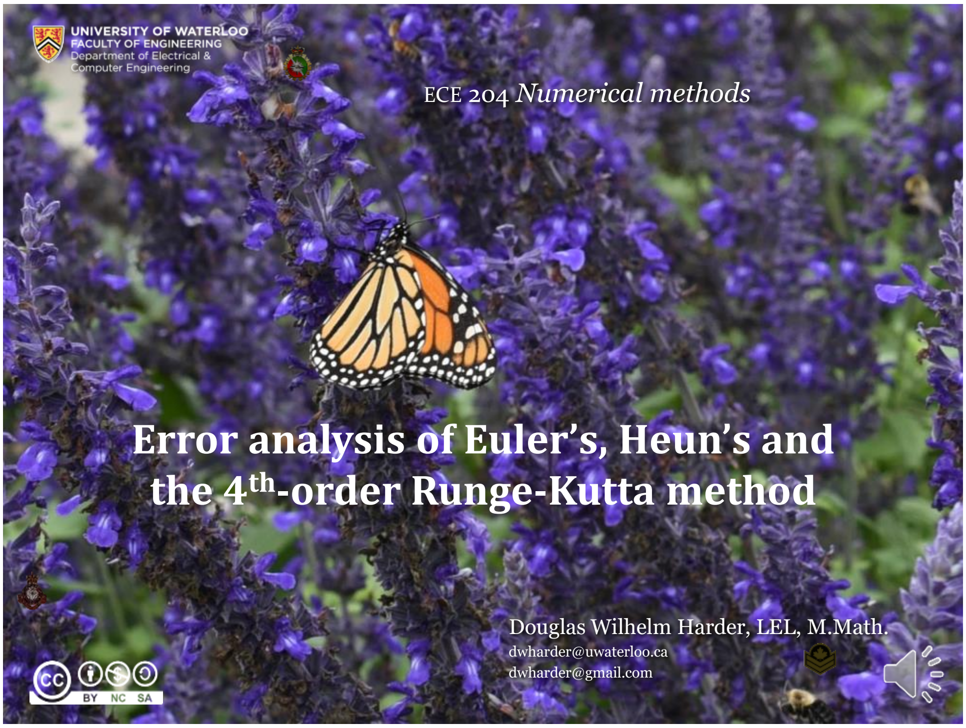


1

Introduction

- In this topic, we will
  - Calculate $y^{(2)}(t)$ given $y^{(1)}(t) = f(t, y(t))$
  - Describe the Lipschitz constant
  - Consider the worst-case scenario for solving an initial-value problem (IVP)
  - Comment on defining error for IVPs
  - Look at how we can estimate the error of Euler's, Heun's and the 4th-order Runge-Kutta method
  - Discuss issues with this approach

2

Initial-value problem solvers

- Our error analysis showed that for each step of these techniques, the error was $O(h^N)$ for a single step,
  but for multiple steps, it was $O(h^{N-1})$
  - This was based on a similar analysis we saw for integration
- Problem:
  With integration, we could assume each value was exact
- With IVP solvers,
  all but once, we estimate y_{k+1} using a previous estimate

3

3

Euler's method

- Fortunately, the error is never-the-less proportional to h^{N-1},
  but, the coefficient may be slightly larger than expected
- We will do a slightly deeper dive on Euler's method
  - First: the error depends on $y^{(2)}(\tau)$, but can we estimate this?
  - Recall that

$$y^{(1)}(t) = f(t, y(t))$$

  - Thus

$$\begin{aligned} y^{(2)}(t) &= \frac{\mathrm{d}}{\mathrm{d}y} f(t, y(t)) \\ &= \frac{\partial}{\partial t} f(t, y(t)) + \frac{\partial}{\partial y} f(t, y(t)) \frac{\mathrm{d}}{\mathrm{d}y} y(t) \\ &= \frac{\partial}{\partial t} f(t, y(t)) + \frac{\partial}{\partial y} f(t, y(t))\, f(t, y(t)) \end{aligned}$$

4

4

Euler's method

- Let's look at our two examples
  - Given $y^{(1)}(t) = -y(t)$

$$y^{(2)}(t) = 0 + (-1)(-y(t))$$
$$= y(t)$$

  - Given $y^{(1)}(t) = -0.2y(t) - \sin(t) - 0.1$

$$y^{(2)}(t) = -\cos(t) + (-0.2)(-0.2y(t) - \sin(t) - 0.1)$$
$$= -\cos(t) + 0.04y(t) + 0.2\sin(t) + 0.02$$

5

5

Lipschitz constant

- A function *f* on a domain $[t_0, t_f] \times [y_L, y_U]$ is said to satisfy a *Lipschitz condition* if

$$\left| f(t, y_2) - f(t, y_1) \right| \leq L \left| y_2 - y_1 \right|$$

  - For example, given $f(t, y) = -y$

$$\left| (-y_2) - (-y_1) \right| = 1 \cdot \left| y_2 - y_1 \right|$$

  - For example, given $f(t, y) = -0.2y - \sin(t) - 0.1$

$$\left| (-0.2y_2 - \sin(t) - 0.1) - (-0.2y_1 - \sin(t) - 0.1) \right|$$
$$= \left| 0.2(y_1 - y_2) \right|$$
$$= 0.2 \left| y_2 - y_1 \right|$$

6

6

Euler's method

- You can therefore show the error may be larger than expected:

$$\left|y(t_f) - y_N\right| \le h\frac{\left|y^{(2)}(\tau)\right|}{2L}\left(e^{L(t_f - t_0)} - 1\right)$$

$$= h\frac{\left|y^{(2)}(\tau)\right|}{2}\left((t_f - t_0) + \frac{1}{2}L(t_f - t_0)^2 + \frac{1}{6}L^2(t_f - t_0)^3 + \cdots\right)$$

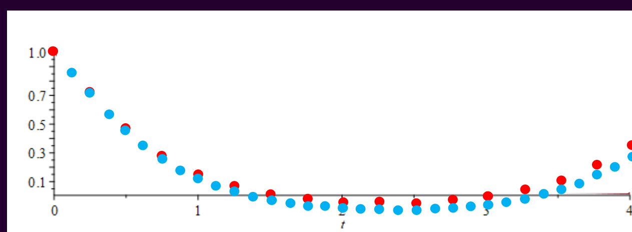
$$= h\frac{\left|y^{(2)}(\tau)\right|}{2}(t_f - t_0) + h\frac{y^{(2)}(\tau)}{2}\left(\frac{1}{2}L(t_f - t_0)^2 + \frac{1}{6}L^2(t_f - t_0)^3 + \cdots\right)$$

A comment on error

- With previous algorithms,
  we required successive approximations to be within ε_{abs}
  - Previous problems have included:
    - Convergence of the root finding algorithms
- Approximating a solution to an IVP is an open-ended problem
  - We may not know at which point we wish to stop
  - Thus, we will use the maximum absolute error per unit time
  - Given an IVP and we want to approximate a solution at t_f,
    we will expect the error to be less than $\varepsilon_{\text{abs}}(t_f - t_0)$

Is our approximation good enough?

- Question: With Euler's method, how large should h be?
  - Consider the following strategy:
    - Choose an n and find y_n which approximates $y(t_f)$
    - Repeat this process but now use $2n$ intervals
      - Let these approximations be denoted z_k so that z_{2n} also approximates $y(t_f)$



Is our approximation good enough?

- Thus, we have two approximations of $y(t_f)$:

$$y(t_f) \approx y_n + Ch$$
$$y(t_f) \approx z_{2n} + C\frac{h}{2}$$
$$0 \approx (y_n + Ch) - \left(z_{2n} + C\frac{h}{2}\right)$$

  - Thus, we have

$$z_{2n} - y_n \approx C\frac{h}{2}$$

  - Therefore, the error of z_{2n} is approximately

$$y(t_f) - z_{2n} \approx z_{2n} - y_n$$

Is our approximation good enough?

- Thus, by subtracting the worse approximation from the better approximation, we have an approximation of the error of z_{2n}

$$y(t_f) - z_{2n} \approx z_{2n} - y_n$$

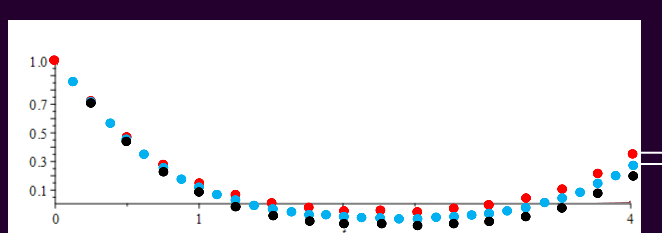
  - Not only that, but we can do one better:

$$y(t_f) \approx 2z_{2n} - y_n$$

  - Thus, having calculated y_n and z_{2n}, we can find an even better approximation of $y(t_f)$
  - This weighted average is an $O(h^2)$ approximation!

Is our approximation good enough?

- Consequently, given our two approximations
  - The difference between the last two is an approximation of the error of the last z_{2n}
  - Also, for each y_k we have that $y(t_k)$, $2z_{2k} - y_k$ is a better approximation of $y(t_k)$



Is our approximation good enough?

- Thus, suppose we want our approximation to $y(t_f)$ to be no larger than $\varepsilon_{\text{abs}}(t_f - t_0)$
  - Thus, we will do the following:
    - Choose an n and calculate the approximations $y_1, \ldots, y_n$
    - With $2n$ intervals, calculate the approximations $z_1, \ldots, z_{2n}$
    - Recall that $|y_n - z_{2n}|$ is only an estimate of the error
    - To be safe, check if $2|y_n - z_{2n}| < \varepsilon_{\text{abs}}(t_f - t_0)$
      - If this is true, we are finished and approximate $y(t_k)$ with the n values $y(t_k) \approx 2z_{2k} - y_k$
      - Otherwise, repeat this process again but now with $4n$ intervals
- Because these techniques require us to approximate solutions with n, then $2n$ and possibly $4n$, $8n$, etc. intervals, we will call these approaches *iterative* solvers

Is our approximation good enough?

- We can do this with Heun and RK4

Heun

$$y(t_f) \approx y_n + Ch^2$$
$$y(t_f) \approx z_{2n} + C\frac{h^2}{4}$$
$$0 \approx (y_n + Ch^2) - \left(z_{2n} + C\frac{h^2}{4}\right)$$
$$z_{2n} - y_n \approx C\frac{3}{4}h^2$$
$$\frac{z_{2n} - y_n}{3} \approx C\frac{h^2}{4}$$
$$y(t_f) - z_{2n} \approx \frac{z_{2n} - y_n}{3}$$
$$y(t_f) \approx \frac{4z_{2n} - y_n}{3}$$

RK4

$$y(t_f) \approx y_n + Ch^4$$
$$y(t_f) \approx z_{2n} + C\frac{h^4}{16}$$
$$0 \approx (y_n + Ch^2) - \left(z_{2n} + C\frac{h^4}{16}\right)$$
$$z_{2n} - y_n \approx C\frac{15}{16}h^4$$
$$\frac{z_{2n} - y_n}{15} \approx C\frac{h^4}{16}$$
$$y(t_f) - z_{2n} \approx \frac{z_{2n} - y_n}{15}$$
$$y(t_f) \approx \frac{16z_{2n} - y_n}{15}$$

Issues

- This approach, however, does have its drawbacks:
  - It's expensive: we are making a minimum of $3n$ approximations
  - Even if the approximation at $y(t_f)$ is sufficiently accurate, this does not mean that each point is sufficiently accurate…
  - Also, suppose the solution is smooth at some points, and variable at others
    - Where the solution is smooth, we can get away with a larger step size
    - Where the solution varies, we probably require a smaller step size
  - This approach, however, requires us to use that smaller step size everywhere

Adaptive techniques

- We will, instead, look at adaptive techniques that allow us to dynamically vary the step size
  - Question: How can we estimate the error if we don't know what the solution actually is?

Summary

- Following this topic, you now
  - Understand how to get $y^{(2)}(t)$ given $y^{(1)}(t) = f(t, y(t))$
  - Are aware that the error may be more significant
    - You are not required to master this in this course
    - You must simply be aware of the issue,
      many of you will never come across this issue
  - Understand we will define the maximum absolute error in terms of an acceptable error ε_{abs} per unit time
    - Because $|y_n - z_{2n}|$ only approximates the error, we'll double it
  - Know how to iteratively estimate the error with Euler's, Heun's and the 4th-order Runge-Kutta method
  - Know how to get an even better approximation using two less-optimal approximations
  - Understand that there are issues with this approach

17

17

References

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

[2] https://en.wikipedia.org/wiki/Extrapolation

18

18

Acknowledgments

None so far.

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.

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.

21