Ordinary Differential Equations

Ch. 25

Orientation

- ODE's
  - Motivation
  - Mathematical Background
- Runge-Kutta Methods
  - Euler's Method
  - Huen and Midpoint methods

Lesson Objectives

- Be able to classify ODE's and distinguish ODE's from PDE's.
- Be able to reduce nth order ODE's to a system of first order ODE's.
- Understand the visual representations of Euler's method.
- Know the relationship of Euler's Method to the Taylor series expansion and the insight it provides regarding the error of the method
- Understand the difference between local and global truncation errors for Euler's method.

Ordinary Differential Equations: Motivation

- Very Common in Engineering
- Fundamental laws are based on changes in physical properties
  - Q =-k dT/dx   Fourier's Law
  - F= d/dt (mv) Newton's 2nd law
- Many ODEs can be solved analytically, however more complex ones must be attacked numerically

Differential Equations: *Classification*

- Order of a differential Equation.
- Ordinary vs. Partial differential equations.
- Linear/Non-linear

$$y' - y = 0$$

$$mx'' + cx' + kx = F(t)$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = F(t)$$

ODEs – Numerical Solutions

- Concentrate on 1st order ODE's because higher order ODE's can be reduced to a set of 1st order ODEs
- 1st Order ODE → F(x,y,y')=0

$$y' - y + x = 0$$

- 2nd Order ODE → F(x,y,y',y'')=0

$$y'' + 2xy' = e^x \cos(y)$$

Ordinary Differential Equations

- Reducing higher order differential equations to a system of first order equations:

$$mx'' + cx' + kx = 0$$

Define a new variable

$$y = \frac{dx}{dt}$$

Substitute into the original DE

$$my' + cy + kx = 0$$

Ordinary Differential Equations

- Reducing higher order differential equations to a system of first order equations:

$$my' + cy + kx = 0 \quad \longrightarrow \quad y = \frac{dx}{dt}, \quad \frac{dy}{dt} = -\left(\frac{cy + kx}{m}\right) = 0$$

In general, an n^{th} order ODE can be reduced to n 1^{st} order ODEs (with appropriate boundary or initial conditions)

ODEs – Numerical Solutions

- Initial Value Problems: all conditions are specified at the same value of the independent variable ($t=0$ or $x=0$). Provide a unique solution (for an *nth* order differential equation, n conditions are required).
- Boundary Value Problems: conditions are specified at different values of the independent variable, I.e.,

  $y(x=0)=0$ & $y(x=4)=3$

Answer the following

- What is (are) the dependent variable(s)?
- What is (are) the independent variable (s)?
- Is this a ODE or PDE?
- What order is this differential equation?
- Is this linear or nonlinear?

$$\frac{dC}{dt} = \frac{d^2C}{dx^2}$$

Leonhard Euler

Courtesy of Wikipedia Encylopedia

y = f (x)

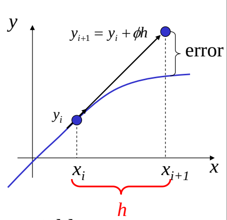
Runge-Kutta Methods – CH 25

Solve ODEs of the form:

$$\frac{dy}{dx} = f(x, y)$$

Can be solved Numerically using:

$$y_{i+1} = y_i + \phi h$$



ϕ = slope estimate

h = step size

y_i = current value of the dependant variable

y_{i+1} estimate of dependant variable over dist. H

Formula can be applied step by step to trace out the solution trajectory.

Euler's Method

The first derivative provides the slope at x_i

$$\frac{dy}{dx} = y' = \phi = f(x, y)$$

Hence, $y_{i+1} = y_i + f(x_i, y_i)h$ → Euler's Method

Note: the slope at the beginning of the interval is taken as the average slope over the entire interval

Euler's Method example

Use Euler's method to numerically integrate $y' = -2x^3 + 12x^2 - 20x + 8.5$ from x=0 to x=4 with a step size of 0.5. The initial condition at x=0 is y=1.

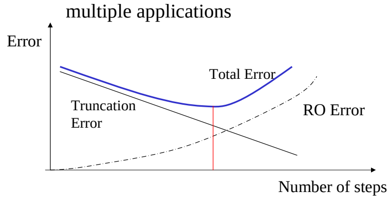
Euler's Method – Error Assessment

2 Sources of Error:

1. Truncation – Taylor Series
2. Round-Off – significant Digits

Truncation Error: 2 parts

1. Local – method application over 1 step
2. Global – accumulated additive error over multiple applications



Euler's Method – Error Assessment

Local Truncation Error:

- First, derive Euler's method from T-S Expansion to represent: $y' = f(x, y)$ with $h = (x_{i+1} - x_i)$

$$y_{i+1} = y_i + y'_i h + \frac{y''_i h^2}{2!} + \ldots + \frac{y''_i h^n}{n!} + R_n$$

Now let $y' = f(x, y)$

$$\underbrace{y_{i+1} = y_i + f(x_i, y_i)h}_{\text{Euler's Method}} + \underbrace{\frac{f'(x_i, y_i)h^2}{2!}}_{\text{Next term}} + \ldots + \frac{f^{n-1}(x_i, y_i)h^n}{n!} + O(h^{n+1})$$

$$\therefore E_a = O(h^2) \rightarrow \text{Local truncation Error}$$

Euler's Method – Error Assesment

Notes:

- This is only the *local truncation* error
- The *global truncation* error is *O(h)*
- If the function is a first order polynomial the method is exact → "1st Order Method"
- The error pattern holds for higher order methods (n^{th} order method) , That is:
  - They yield exact results for n^{th} order polynomial
  - *Local truncation* error is $O(h^{n+1})$
  - *Global truncation* error is $O(h^n)$

Matlab Pseudocode for Euler's Method

'set integration range
'initialize variables
'set step size
'loop to generate x array
'loop to implement Euler's Method
'display results

We have learned

- How to classify differential equations
- How to reduce nth order ODE's to a system of 1st order ODE's.
- The visual representation of Euler's method.
- The relationship between the Taylor series expansion and Euler's Method
- The difference between global and local truncation error in Euler's Method.

Euler's Method – Beyond Error

- *Convergence:* In the absence of Round-off Errors if our numerical solution approaches the exact solution as the step size *h* is reduced, it is said to be convergent
- *Stability*: Depends on the method and the differential equation

Euler's Method – Stability

- A numerical method is unstable if the error grows without bound (e.g. exponential growth) for a problem in which the exact solution is bounded.
- Can depend on the method as well as the differential equation.
- Example:

$$\frac{dy}{dx} = \lambda y$$

$$y = y_0 e^{\lambda x}$$

Euler Method

$$\begin{cases} y_{i+1} = y_i + \lambda y_i h = y_i(1+\lambda h) \\ y_1 = y_0(1+\lambda h) \\ y_2 = y_1(1+\lambda h) = y_0(1+\lambda h)(1+\lambda h) = y_0(1+\lambda h)^2 \\ y_n = y_0(1+\lambda h)^n \end{cases}$$

Euler's method is conditionally stable for:

$$|1+\lambda h| \le 1$$

Euler's Method – Stability

$$|1+\lambda h| \le 1$$

This Implies

$$\lambda < 0$$

$$h \le \frac{2}{|\lambda|}$$

A numerical Method is ***unconditionally stable*** if it is stable for any values of *h* and other parameter is the differential equation

Huen's Method – "Predictor – Corrector" Approach

1. Begin as with Euler:

$$y^o_{i+1} = y_i + f(x_i, y_i)h \longrightarrow \text{Predictor Equation}$$

2. Use to estimate slope at the end of the interval, *h*

$$y'_{i+1} = f(x_{i+1}, y^o_{i+1})$$

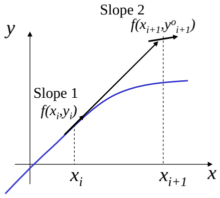
3. Calculate an average slope

$$\overline{y'} = \frac{f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})}{2}$$

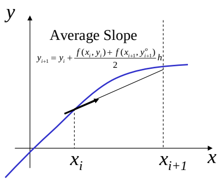
4. Extrapolate linearly from y_i to y_{i+1}

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})}{2} h \longrightarrow \text{Corrector Equation}$$

Huen's Method – "Predictor – Corrector" Approach



Predictor Step



Corrector Step
Use average slope to
Obtain new estimate

Huen's Method – Iteration step

Since y_{i+1} is on both sides of the corrector equation it can be applied iteratively as:

$$\overline{y'} = \frac{f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})}{2}$$

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})}{2} h$$

Note:
- This iterative procedure does not converge on the true answer
- Converges to a finite truncation error

Huen's Method – Example

Solve: $y' = x - y$ Subject to the I.C. $y(x=0)=0$
at $x=0.4$ with $h=0.4$ using Heun's method

1. Begin with Predictor Equation ($i=0$, for initial conditions):

$$y^o_{i+1} = y_i + f(x_i, y_i)h$$
$$y^o_1 = y_0 + f(x_0, y_0)h$$
$$y^o_1 = y_0 + (x_0 - y_0)h = 0$$

2. Calculate an average slope

$$\overline{y'} = 0.5\left(f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})\right)$$
$$\overline{y'} = 0.5\left((x_0 - y_0) + (x_1, y^o_1)\right)$$
$$\overline{y'} = 0.5\left((0-0) + (0.4-0)\right) = 0.2$$

4. Use corrector equation

$$y_{i+1} = y_i + 0.5\left(f(x_i, y_i) + f(x_{i+1}, y^o_{i+1})\right)h$$
$$y_1 = 0 + (0.2)0.4 = 0.08$$

Huen's Method – Example

Solve: $y' = x - y$ Subject to the I.C. $y(x=0)=0$
at x=0.4 with h=0.4 using Heun's method

Exact Solution: $y_e = x + e^{-x} - 1$

At x = 0.4 $y_e = 0.4 + e^{-0.4} - 1 = 0.07032$

True Error ⟶ $E_t = \left| \frac{.07032 - .08}{.07032} \right| \bullet 100\% = 13\%$

- Method is exact for 2nd order polynomials
- 2nd order accurate
- *Local truncation* error is O(h^3)
- *Global truncation* error is O(h^2)

Huen's Method – Example

Solve: $y' = x - y$ Subject to the I.C. $y(x=0)=0$
at x=0.4 with h=0.4 using Heun's method

Exact Solution: $y_e = x + e^{-x} - 1$

At x = 0.4 $y_e = 0.4 + e^{-0.4} - 1 = 0.07032$

True Error ⟶ $E_t = \left| \frac{.07032 - .08}{.07032} \right| \bullet 100\% = 13\%$

- Method is exact for 2nd order polynomials
- 2nd order accurate
- *Local truncation* error is O(h^3)
- *Global truncation* error is O(h^2)

Compare to Euler?

Huen's Method – Matlab Code Example

See matlab code

Note that if *y'* is only a function of the independent variable *x*, there is no need to iterate and the following equation holds for Huen's method:

$$y_{i+1} = y_i + \frac{f(x_i) + f(x_{i+1})}{2} h$$

⟵ Directly related to the trapezoidal rule

Runge-Kutta Methods – CH 25

Can achieve Taylor Series accuracy without evaluating higher order derivatives.

General form: $y_{i+1} = y_i + \phi(x_i, y_i, h)h$ (1)

$\phi(x_i, y_i, h)$ - *Increment function* & is like a slope over the interval

$\phi = a_1 k_1 + a_2 k_2 + ... + a_n k_n$

- *a*'s are constants & *k*'s are recurrence relationships
- n=1 → Euler's method

Runge-Kutta Methods – CH 25

Can achieve Taylor Series accuracy without evaluating higher order derivatives.

General form: $y_{i+1} = y_i + \phi(x_i, y_i, h)h$ (1)

$\phi(x_i, y_i, h)$ - *Increment function* & is like a slope over the interval

$$\phi = a_1 k_1 + a_2 k_2 + ... + a_n k_n$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

$$k_3 = f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h)$$

$$k_n = f(x_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + ... + q_{n-1,n-1} k_{n-1} h)$$

- *a*'s are constants & *k*'s are recurrence relationships
- n=1 → Euler's method

Runge-Kutta Methods

To Determine the final form of (1)

1. Select *n*
2. Evaluate *a's,p's,q's* by setting the general form equal to terms in the T-S expansion.
3. For low-order forms
   - Number of terms *n*=order of the method
   - *Local truncation error* is $O(h^{n+1})$
   - *Global truncation error* is $O(h^n)$

2nd- Order Runge-Kutta Methods

See Box 25.1 in Text

General Form: $y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$ (2)

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

By setting (2) equal to a T-S expansion through the 2nd order term, we can solve for a_1, a_2, p_1, q_{11}

$$\left.\begin{aligned} a_1 + a_2 &= 1 \\ a_2 p_1 &= 1/2 \\ a_2 q_{11} &= 1/2 \end{aligned}\right\}$$

3 Eqns & 4 unknows
Specify a_2 value →

$$\left\{\begin{aligned} a_1 &= 1 - a_2 \\ p_1 &= 1/(2a_2) \\ q_{11} &= 1/(2a_2) \end{aligned}\right.$$

**Since there are an infinite number of choices for a_2 there will be an infinite number of 2nd order R-K Methods*

2nd- Order Runge-Kutta Methods

A) Huen Method without iteration

($a_2 = ½$): $a_1 = ½$, $p_1 = 1$, $q_{11} = 1$

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + h, y_i + k_1 h)$$

k_1 slope at start of interval
k_2 slope at end of interval

$$a_1 = 1 - a_2$$
$$p_1 = 1/(2a_2)$$
$$q_{11} = 1/(2a_2)$$

Global Truncation Error ~ $O(h^2)$

2nd- Order Runge-Kutta Methods

B) Midpoint Method ($a_2 = 1$): $a_1 = 0$, $p_1 = 1/2$, $q_{11} = 1/2$

$$y_{i+1} = y_i + k_2 h$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + 0.5h, y_i + 0.5k_1 h)$$

$$a_1 = 1 - a_2$$
$$p_1 = 1/(2a_2)$$
$$q_{11} = 1/(2a_2)$$

Global Truncation Error ~ $O(h^2)$

2^{nd}- Order Runge-Kutta Methods

C) Ralston's Method (a_2 = 2/3): a_1 = 1/3 , p_1 = 3/4, q_{11} =3/4

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + 0.75h, y_i + .75k_1 h)$$

$$a_1 = 1 - a_2$$
$$p_1 = 1/(2a_2)$$
$$q_{11} = 1/(2a_2)$$

Global Truncation Error ~ O(h^2)

4^{th} - Order Runge-Kutta Methods

Classical 4^{th} order RK Method – most commonly used RK method

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

$$y_{i+1} = y_i + \phi h$$

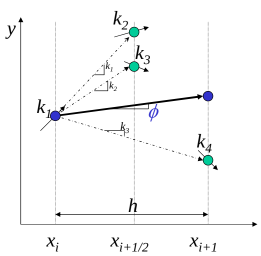
Slope Estimates:

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + 0.5h, y_i + .5k_1 h)$$

$$k_3 = f(x_i + 0.5h, y_i + .5k_2 h)$$

$$k_4 = f(x_i + h, y_i + k_3 h)$$



Global Truncation Error ~ O(h^4)

4^{th} - Order Runge-Kutta Methods –

Example: Use classical RK4 to determine *y @ x=0.4* for *y'=x-y* and *h=0.4*

Recall the exact solution is:

$$y = x + e^{-x} - 1$$
$$y(0.4) = 0.070320$$

RK4 Solution:

$$\left.\begin{array}{l} x_0 = 0 \\ y_0 = 0 \end{array}\right\} \text{Initial Conditions}$$

$$k_1 = f(x_i, y_i) = x_0 - y_0 = 0$$

$$k_2 = f(x_i + 0.5h, y_i + .5k_1 h) = (0 + 0.4/2) - (0 + 0) = 0.2$$

$$k_3 = f(x_i + 0.5h, y_i + .5k_2 h) = (0 + 0.4/2) - (0 + (0.5)(0.2)(0.4)) = 0.16$$

$$k_4 = f(x_i + 0.5h, y_i + k_3 h) = (0 + 0.4) - (0 + 0.16(0.4)) = 0.336$$

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h = 0 + \frac{1}{6}(0 + 2(.2) + 2(.16) + .336)0.4$$

$$y_1 = 0.07040$$

4^{th} - Order Runge-Kutta Methods –

Example: Use classical RK4 to determine y @ $x=0.4$ for $y'=x-y$ and $h=0.4$

Error:

$$E_t = \left| \frac{.07032 - .07040}{.07032} \right| \bullet 100\% = .11\%$$

See Matlab Sample Matlab RK4 method

Method Comparison

- Higher order methods produce better accuracy
- Effort for the higher order methods is similar to low-order methods (much of the effort goes into evaluating the function)
- Classical 4^{th} order RK is most widely used as it produces accurate results with reasonable effort.

Systems of Equations

- Recall, Any n^{th} order ODE can be represented as a system of n 1^{st} order ODEs

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, ..., y_n)$$
$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, ..., y_n)$$
$$\vdots$$
$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, ..., y_n)$$

- To solve the system requires n initial conditions at $x = x_0$

Systems of Equations – RK4 Example

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2)$$
$$\frac{dy_2}{dx} = f_2(x, y_1, y_2)$$

For example:

$$\frac{dy}{dx} = f_1(x, y, z) = -y$$
$$\frac{dz}{dx} = f_2(x, y, z) = 3 - 4\cos z + y$$

Subject to initial conditions

$$y_{1,0} = y_1(x = 0) = Y_1$$
$$y_{2,0} = y_2(x = 0) = Y_2$$

Systems of Equations – RK4 Example

Solve for slopes $k_{i,j}$

*i*th value of k for the *j*th dependant variable

For RK-4 *i=1,2,3 and 4* while *j=1, 2, …* number of dependant variables

Systems of Equations – RK4 Example

Solve for slopes
Start with *i=0*
The initial condition

$$k_{1,1} = f_1(x_i, y_{1i}, y_{2i})$$
$$k_{1,2} = f_2(x_i, y_{1i}, y_{2i})$$
$$k_{2,1} = f_1\left(x_i + \frac{1}{2}h, y_{1i} + \frac{1}{2}k_{11}h, y_{2i} + \frac{1}{2}k_{12}\right)$$
$$k_{2,2} = f_2\left(x_i + \frac{1}{2}h, y_{1i} + \frac{1}{2}k_{11}h, y_{2i} + \frac{1}{2}k_{12}h\right)$$
$$k_{3,1} = f_1\left(x_i + \frac{1}{2}h, y_{1i} + \frac{1}{2}k_{21}h, y_{2i} + \frac{1}{2}k_{22}h\right)$$
$$k_{3,2} = f_2\left(x_i + \frac{1}{2}h, y_{1i} + \frac{1}{2}k_{21}h, y_{2i} + \frac{1}{2}k_{22}h\right)$$
$$k_{4,1} = f_1(x_i + h, y_{1i} + k_{31}h, y_{2i} + k_{32}h)$$
$$k_{4,2} = f_2(x_i + h, y_{1i} + k_{31}h, y_{2i} + k_{32}h)$$

Systems of Equations – RK4 Example

$$y_{1,i+1} = y_{1i} + \frac{1}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41})h$$
$$y_{2,i+1} = y_{2i} + \frac{1}{6}(k_{12} + 2k_{22} + 2k_{32} + k_{42})h$$

Show Matlab Systems of Equations RK4 Example

$$\frac{dy_1}{dx} = y_2 \qquad y_1(x=0) = 4$$
$$\frac{dy_2}{dx} = -\frac{y_2}{2} - 7y_1 \qquad y_2(x=0) = 0$$

Matlab ODE solvers

ODE23 and ODE45 are RK solvers that combine 2nd and 3rd order RK and 4th and 5th order RK methods.

See Chapter 8 in Palm Text.