MATH 128A Numerical Analysis Discussion Section

Raehyun Kim*

* Department of Mathematics, UC Berkeley

Outline

- (Forward) Euler's Method
  - $x_{n+1} = x_n + hf(x_n, t_n)$
- High-order Taylor Method
  - $x_{n+1} = x_n + hT(x_n, t_n)$
- Runge-Kutta Method
  - Midpoint Euler, RK4
- Multi-step Method
  - Adams-Bashforth / Adams-Moulton

Euler's Method

- Forward Euler's method
  - $y_{n+1} = y_n + hf(t_n, y_n)$

- Error analysis(w/ truncation error)

$$|y(t_{j+1}) - u_{j+1}| \leq \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left(e^{L(t_{j+1}-a)} - 1\right) + \delta\, e^{L(t_{j+1}-a)}.$$

  - As t grows, the error increases exponentially.
  - Requires really huge N for adequate error bounds.
  - We can not take arbitrary small h.

Euler's Method

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.5$

High-order Taylor Method

- High-order Taylor Method
  - $x_{n+1} = x_n + hT(x_n, t_n)$
  - $T(x_n, t_n) = f(x_n, t_n) + \frac{h}{2}\frac{df}{dt}(x_n, t_n) + \cdots + \frac{h^{k-1}}{k!}\frac{d^{k-1}f}{dx^{k-1}}(x_n, t_n)$
- LTE

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)),$$

  - $O(h^n)$

High-order Taylor Method

a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1,$ with $h = 0.5$

Runge-Kutta Method

- Runge-Kutta method
  - Similar to the high-order Taylor method but it uses the function values instead of the value of derivatives
  - Approximate the value of derivatives using the Taylor polynomial in two variables
  - (OR) Divide 1 time step into multiple stages

Runge-Kutta Method

- Example
  - Midpoint method

$$w_0 = \alpha,$$

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right), \quad \text{for } i = 0, 1, \ldots, N-1.$$

  - Modified Euler

$$w_0 = \alpha,$$

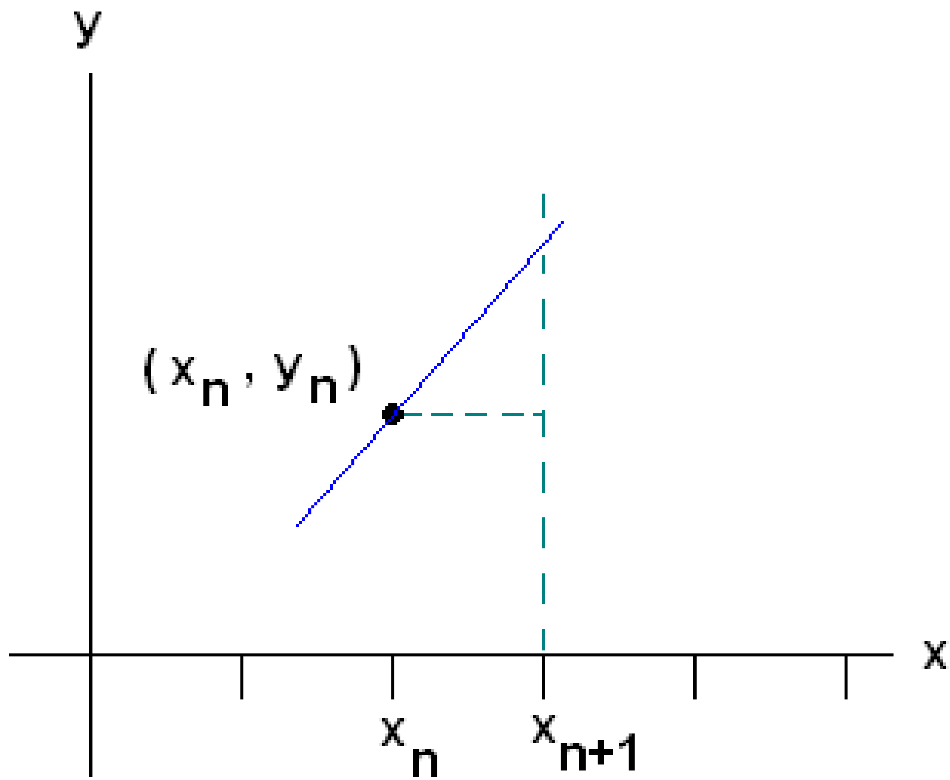
$$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))], \quad \text{for} \quad i = 0, 1, \ldots, N-1.$$

Runge-Kutta Method

- Example
  - Mi

    $w_{($

    w_{i+} . . , $N-1$.



  - Mo

    w

    w_{i+} 1, . . . , $N-1$.

Runge-Kutta Method

- Example
  - RK-4th order

$$w_0 = \alpha,$$

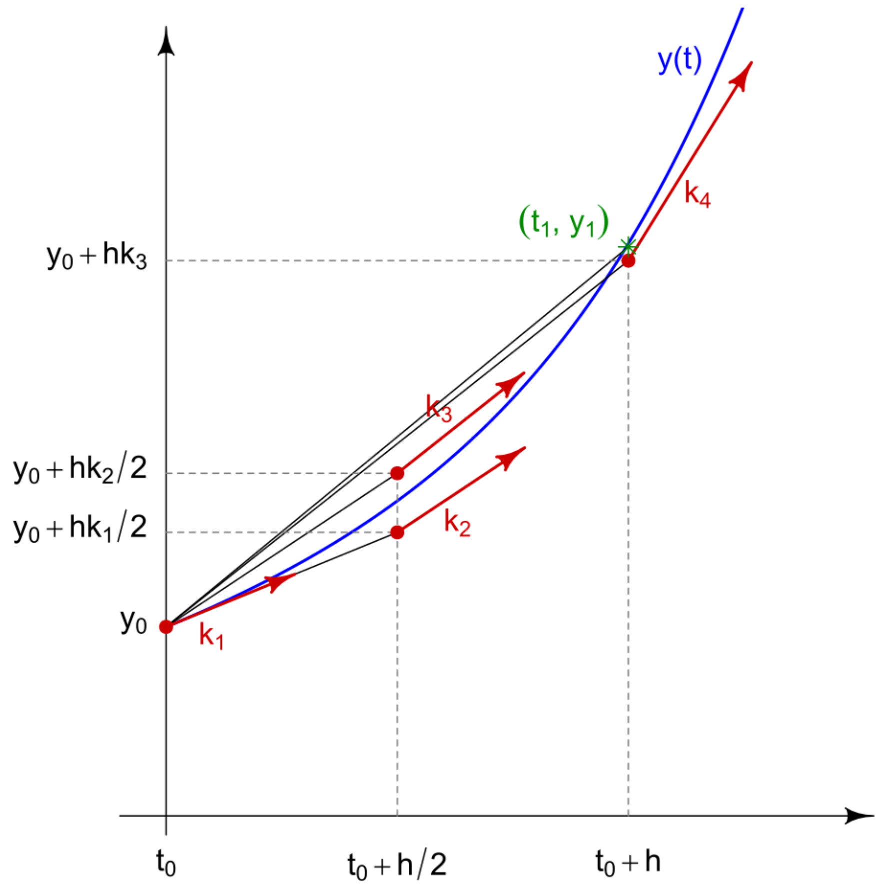
$$k_1 = hf(t_i, w_i),$$

$$k_2 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right)$$

$$k_4 = hf(t_{i+1}, w_i + k_3),$$

$$w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k$$



Runge-Kutta Method

a. $y' = e^{t-y}$, $0 \le t \le 1$, $y(0) = 1$, with $h = 0.5$; actual solution $y(t) = \ln(e^t + e - 1)$.

Multi-step method

- Numerical scheme using multiple $f\big(t_k, y(t_k)\big)$
  - (Forward) Euler

$$y_{k+1} = y_k + hf(t_k, y_k)$$

  - Trapezoidal

$$y_{k+1} = y_k + \frac{h}{2}\big(f(t_k, y_k) + f(t_{k+1}, y_{k+1})\big)$$

  - Adams-Bashforth
  - Adams-Moulton

Multi-step method

- Basic idea of the construction
  - Approximate

$$\int_{t_n}^{t_{n+1}} f\big(t, y(t)\big)dt$$

  - Using the interpolation -> LMM
    - Excluding the right endpoint -> explicit (open)
    - Including the right endpoint -> implicit (close)
  - Using the quadrature rule -> RK mthd / GL mthd

Multi-step method

- Adams-Bashforth

$$y_{n+1} = y_n + hf(t_n, y_n), \qquad \text{(This is the Euler method)}$$

$$y_{n+2} = y_{n+1} + h\left(\frac{3}{2}f(t_{n+1}, y_{n+1}) - \frac{1}{2}f(t_n, y_n)\right),$$

$$y_{n+3} = y_{n+2} + h\left(\frac{23}{12}f(t_{n+2}, y_{n+2}) - \frac{16}{12}f(t_{n+1}, y_{n+1}) + \frac{5}{12}f(t_n, y_n)\right),$$

$$y_{n+4} = y_{n+3} + h\left(\frac{55}{24}f(t_{n+3}, y_{n+3}) - \frac{59}{24}f(t_{n+2}, y_{n+2}) + \frac{37}{24}f(t_{n+1}, y_{n+1}) - \frac{9}{24}f(t_n, y_n)\right),$$

$$y_{n+5} = y_{n+4} + h\left(\frac{1901}{720}f(t_{n+4}, y_{n+4}) - \frac{2774}{720}f(t_{n+3}, y_{n+3}) + \frac{2616}{720}f(t_{n+2}, y_{n+2}) - \frac{1274}{720}f(t_{n+1}, y_{n+1}) + \frac{251}{720}f(t_n, y_n)\right).$$

(Wikipedia)

Multi-step method

- Adams-Moulton

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}),$$ This is the backward Euler method

$$y_{n+1} = y_n + \frac{1}{2}h\left(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)\right),$$ This is the trapezoidal rule

$$y_{n+2} = y_{n+1} + h\left(\frac{5}{12}f(t_{n+2}, y_{n+2}) + \frac{2}{3}f(t_{n+1}, y_{n+1}) - \frac{1}{12}f(t_n, y_n)\right),$$

$$y_{n+3} = y_{n+2} + h\left(\frac{9}{24}f(t_{n+3}, y_{n+3}) + \frac{19}{24}f(t_{n+2}, y_{n+2}) - \frac{5}{24}f(t_{n+1}, y_{n+1}) + \frac{1}{24}f(t_n, y_n)\right),$$

$$y_{n+4} = y_{n+3} + h\left(\frac{251}{720}f(t_{n+4}, y_{n+4}) + \frac{646}{720}f(t_{n+3}, y_{n+3}) - \frac{264}{720}f(t_{n+2}, y_{n+2}) + \frac{106}{720}f(t_{n+1}, y_{n+1}) - \frac{19}{720}f(t_n, y_n)\right).$$

(Wikipedia)

Multi-step method

a. $y' = e^{t-y}$, $0 \le t \le 1$, $y(0) = 1$, with $h = 0.5$; actual solution $y(t) = \ln(e^t + e - 1)$.