Lecture 11: Differential Equations

Eigenvalue Problems

Abhinav Jha

Indian Institute of Technology, Gandhinagar

13^{th} April 2026

Outline

$\frac{dy}{dt} = \underbrace{f(t,y)}_{\text{Tangent value/ Direction}}$; $\underbrace{y(a)=\alpha}_{\text{Initial position}}$; $\underbrace{a \le t \le b}_{\text{when } t \text{ ranges and } y \text{ exist}}$.

$y(t)+C$]- Solution

$[a,b] \rightarrow \{t_i\}_{i=0}^{n}$

Euler: $y_{i+1} = y_i + h f(t_i, y_i)$ $i=0,\dots,n-1$; $h = \frac{b-a}{n}$

- $O(h)$ method

Higher order Taylor method: $y_{i+1} = y_i + h \underbrace{T^{(n)}(t_i, y_i)}_{\text{Taylor part}}$, $i=0,\dots,n-1$

$\underbrace{\{ f^{(k)}(t_i, y_i) \}}$

- $O(h^n)$ method ; depends on Taylor

Runge-Kutta (RK) methods: The idea is to use 2D Taylor expansion of $f(t,y)$ to compute the derivatives
-2

RK-2 : a) Midpoint

$y_0 = \alpha$

$$y_{i+1} = y_i + h f\left(t_i + \frac{h}{2}, \underbrace{y_i + \frac{h}{2} f(t_i, y_i)}_{K_1}\right) ; \quad i = 0, 1, \ldots, n-1.$$

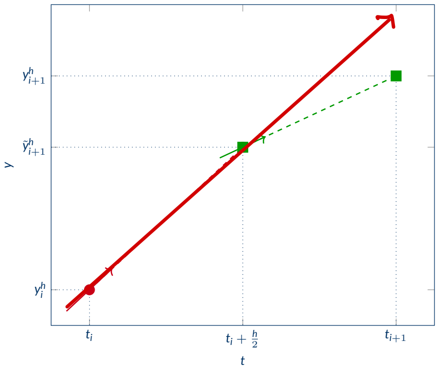
$$K_1 = y_i + \frac{h}{2} f(t_i, y_i) \qquad - O(h^2)$$

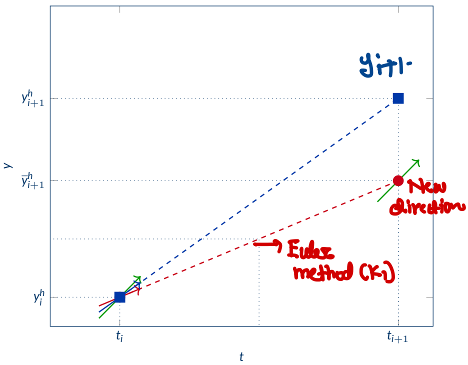
b) Modified Euler

$y_0 = \alpha$

$$y_{i+1} = y_i + h\left[\frac{f(t_i, y_i) + f(t_{i+1}, \underbrace{y_i + h f(t_i, y_i)}_{K_1})}{2}\right] ; \quad i = 0, 1, \ldots, n-1$$

$- O(h^2)$.





Geometrical view of the Midpoint and Modified Euler methods.

Example : Consider $y' = y - t^2 + 1$; $h = 0.5$; $y(0) = 0.5$, $0 \le t \le 2$.

Midpoint : $\bar{y}_{i+1} = y_i + \frac{h}{2} f(t_i, y_i)$

$$y_{i+1} = y_i + h f\left(t_i + \frac{h}{2}, \bar{y}_{i+1}\right)$$

$y_0 = 0.5$; $i = 0$

$$\bar{y}_1 = y_0 + \frac{0.5}{2} f(t_0, y_0) = 0.875$$

$$y_1 = y_0 + h f\left(t_0 + \frac{0.5}{2}, \bar{y}_1\right) = 1.40625$$

Then y_2 ; $\to \bar{y}_2$ and then y_2.

Generalized RK-ν method : y_{i+1} given y_i

$$y_0 = \alpha$$

$$k_1 = y_i$$

$$k_2 = y_i + h\,a_{2,1} f(t_i, k_1)$$

$$k_3 = y_i + h\,a_{3,1} f(t_i, k_1) + h\,a_{3,2} f(t_i + c_2 h, k_2)$$

$$\vdots$$

$$k_\nu = y_i + h \sum_{j=1}^{\nu-1} a_{\nu,j} f(t_i + c_j h, k_j)$$

$$y_{i+1} = y_i + h \sum_{j=1}^{\nu} b_j f(t_i + c_j h, k_j)$$

Butcher Table

$$\begin{array}{c|c} \underline{c} & A \\ \hline & \underline{b}^T \end{array}$$

;

A: RK-matrix

$\underline{b}$: RK-weights

$\underline{c}$: RK-nodes.

1. RK-1 : $y_{i+1} = y_i + h f(t_i, y_i) \rightarrow$

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

2. RK-2 :

$$\left.\begin{aligned} k_1 &= y_i \\ k_2 &= y_i + \frac{h}{2} f(t_i, k_1) \\ y_{i+1} &= y_i + h f\left(t_i + \frac{h}{2}, k_2\right) \end{aligned}\right\} \text{Mid-point}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

3. RK-3 : Heunn method

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

4. RK-4

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

1. $\sum_{j=1}^{s} b_j = 1$ → Time marching

2. $c_i \leq c_{i+1}$ → Improve the tangents using previous steps.

3. $c_i = \sum_{j=1}^{i-1} a_{i,j}$ → Time fraction is evaluated by sum of weights of prev. slopes.

Explicit RK

Implicit RK $\times$

$y' = y - t^2 + 1.$

t_i	$\nu=1$			$\nu=2$			$\nu=3$			$\nu=4$		
	y_i^h	y_i	$\lvert y_i - y_i^h\rvert$	y_i^h	y_i	$\lvert y_i - y_i^h\rvert$	y_i^h	y_i	$\lvert y_i - y_i^h\rvert$	y_i^h	y_i	$\lvert y_i - y_i^h\rvert$
0.00	0.50000	0.50000	0.00000	0.50000	0.50000	0.00000	0.50000	0.50000	0.00000	0.50000	0.50000	0.00000
0.20	0.80000	0.82930	0.02930	0.82800	0.82930	0.00130	0.82924	0.82930	0.00005	0.82929	0.82930	0.00001
0.40	1.15200	1.21409	0.06209	1.21136	1.21409	0.00273	1.21398	1.21409	0.00011	1.21408	1.21409	0.00001
0.60	1.55040	1.64894	0.09854	1.64466	1.64894	0.00428	1.64877	1.64894	0.00018	1.64892	1.64894	0.00002
0.80	1.98848	2.12723	0.13875	2.12128	2.12723	0.00595	2.12699	2.12723	0.00024	2.12720	2.12723	0.00003
1.00	2.45818	2.64086	0.18268	2.63317	2.64086	0.00769	2.64056	2.64086	0.00030	2.64082	2.64086	0.00004
1.20	2.94981	3.17994	0.23013	3.17046	3.17994	0.00948	3.17958	3.17994	0.00036	3.17989	3.17994	0.00005
1.40	3.45177	3.73240	0.28063	3.72117	3.73240	0.01123	3.73198	3.73240	0.00042	3.73234	3.73240	0.00006
1.60	3.95013	4.28348	0.33336	4.27062	4.28348	0.01286	4.28302	4.28348	0.00046	4.28341	4.28348	0.00007
1.80	4.42815	4.81518	0.38702	4.80096	4.81518	0.01422	4.81470	4.81518	0.00048	4.81509	4.81518	0.00009
2.00	4.86579	5.30547	0.43969	5.29037	5.30547	0.01510	5.30501	5.30547	0.00047	5.30536	5.30547	0.00011

Table 1: Comparison of computed values y_i^h and exact values y_i for different ν.

which RK-method is best?

RK2
3
4

Evaluation at time step	Errors
2	h^2
3	h^3
4	h^4 Best!
$5 \le n \le 7$	h^{n-1}
$8 \le n \le 9$	h^{n-2}
$n \ge 10$	h^{n-3}

A m-step linear multistep method is

$$y'(t) = f(t, y)$$

is defined by

$$y_{i+1} = \sum_{j=0}^{m-1} a_j\, y_{i+1-m+j} + h \sum_{j=0}^{m} b_j f(t_{i+1-m+j}, y_{i+1-m+j})$$

for $i = m-1, m, \ldots, n-1$, $m \geq 2$; $h = \frac{b-a}{n}$; $\{a_j\}, \{b_j\}$ are constants

$$y_0 = \alpha, \quad y_1 = \alpha_1, \ldots, y_{m-1} = \alpha_{m-1}.$$

<u>$m=3$</u>

$$y_{i+1} = a_2 y_i + a_1 y_{i-1} + a_0 y_{i-2} + h \sum_{j=0}^{3} b_j f(t_{i-2+j}, y_{i-2+j}).$$

Eg: $A = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}$; $\underline{x}^{(0)} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ max $\to x_1$

$$\underline{x}^{(1)} = A\underline{x}^{(0)} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}\begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \to \text{max} \to x_2$$

$$\lambda_1 = \frac{\phi(\underline{x}^{(1)})}{\phi(\underline{x}^{(0)})} \to \phi \to \text{denominator.}$$

Normalization:

$$\lambda_i = \frac{\phi(\underline{x}^{(i)})}{\phi(\underline{x}^{(i-1)})} \; ?$$

$$\underline{x}^{(i)} = A\hat{\underline{x}}^{(i-1)}$$

$$\hat{\underline{x}}^{(i-1)} \leftarrow \frac{\underline{x}^{(i-1)}}{\|\underline{x}^{(i-1)}\|_\infty}$$

$$= \frac{\phi(\underline{x}^{(i)})}{\phi(\hat{\underline{x}}^{(i-1)})} \; ? \; \times \to \text{Denominator has no } \lambda_1^k \text{ factor}$$