⊙ Runge-Kutta methods

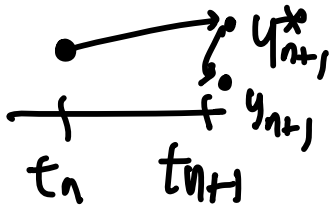
노트 제목 2019-10-16

advantages : ① good stability properties

② timestep size can be changed during computation.

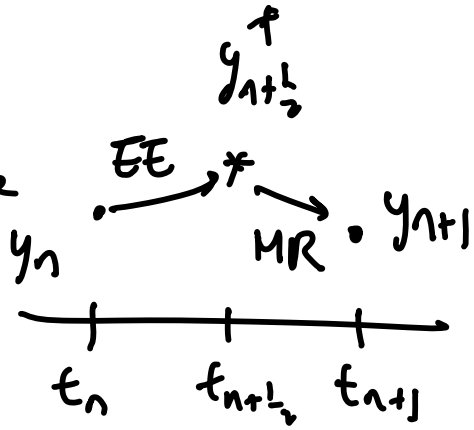
③ self starting

PC



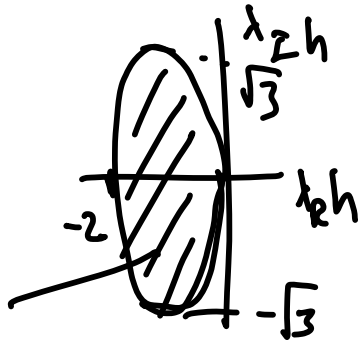
- 2nd-order Runge-Kutta method (RK2)

$$\begin{cases} y^*_{n+\frac{1}{2}} = y_n + \frac{h}{2} f(y_n, t_n) : \text{EE} \\ y_{n+1} = y_n + h f(y^*_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) : \text{mid-point rule} \end{cases}$$



model prob. $y' = \lambda y \longrightarrow y_{n+1} = y_n\left(1 + \lambda h + \frac{1}{2}\lambda^2 h^2\right)$

2nd-order accurate



For $\lambda = i\omega$, unstable $\left(\sigma = 1 + i\omega h - \frac{1}{2}\omega^2 h^2\right)$ RK2

$$\sigma = |\sigma| e^{i\theta}, \quad \theta = \tan^{-1}\frac{\omega h}{1 - \frac{1}{2}\omega^2 h^2}$$

$$|\sigma| > 1$$

$$\text{phase error} = \omega h - \theta = -\frac{1}{6}\omega^3 h^3 + \cdots$$

- RK4 (most popular scheme) $\quad y' = f(y, t)$

$$\begin{cases} y^*_{n+\frac{1}{2}} = y_n + \frac{h}{2} f(y_n, t_n) \\ y^{**}_{n+\frac{1}{2}} = y_n + \frac{h}{2} f\left(y^*_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}\right) \end{cases} \longrightarrow$$

requires 4 function evaluations

$\rightarrow$ expansive

$$\begin{cases} y_{n+1}^{***} = y_n + h f(y_{n+\frac{1}{2}}^{**}, t_{n+\frac{1}{2}}) \\ y_{n+1} = y_n + h\left[\frac{1}{6} f(y_n, t_n) + \frac{1}{3} f(y_{n+\frac{1}{2}}^{*}, t_{n+\frac{1}{2}}) + \frac{1}{3} f(y_{n+\frac{1}{2}}^{**}, t_{n+\frac{1}{2}}) + \frac{1}{6} f(y_{n+1}^{***}, t_{n+1})\right] \end{cases}$$

explicit

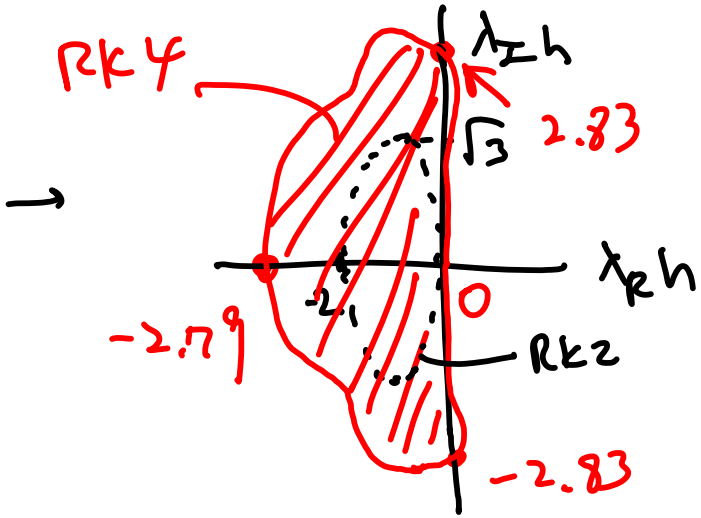
$y' = \lambda y$: $y_{n+1} = \sigma y_n$

$\rightarrow \sigma = 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4$

$e^{\lambda h} = \cdot \; - \; - \; -$

$\leftarrow$ 4th-order accurate

Stability : $|\sigma| \leq 1$ $\leftarrow$ $1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \frac{1}{24}\lambda^4 h^4 = e^{i\theta}$



- conditionally stable
- " " even for $\lambda = i\omega$.

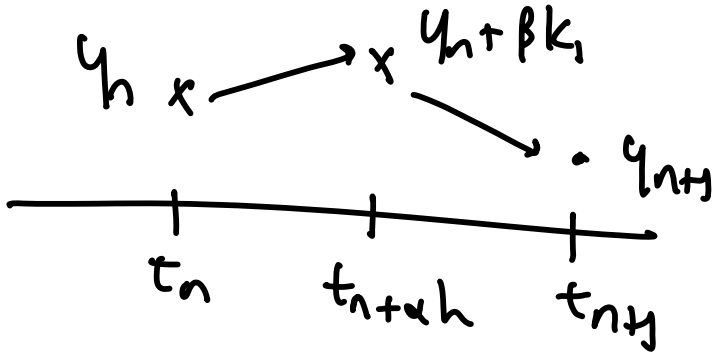
$|\lambda_I h| \leq 2.83$

$\lambda = \lambda_R$: $|\lambda_R h| \leq 2.79$

RK3

- How to construct RK 2?

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



$$\begin{cases} k_1 = h f(y_n, t_n) \\ k_2 = h f(y_n + \beta k_1, t_n + \alpha h) \\ y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2 \end{cases}$$

Find α, β, γ_1 and γ_2 to ensure the highest order of accuracy for the method.

Taylor series for k_2

$$k_2 = h\left[f(y_n, t_n) + \beta k_1 \frac{\partial f}{\partial y}\Big|_n + \alpha h \frac{\partial f}{\partial t}\Big|_n + \cdots \right]$$

$$\rightarrow y_{n+1} = y_n + \gamma_1 h \underline{f_n} + \gamma_2 h \left(\underline{f_n} + \beta h f_n f_{y_n} + \alpha h f_{t_n} + \cdots\right)$$

$$= y_n + (\gamma_1 + \gamma_2) h f_n + \gamma_2 \beta h^2 f_n f_{y_n} + \gamma_2 \alpha h^2 f_{t_n} + \cdots$$

$$\left(\begin{aligned} y(t_{n+1}) &= y(t_n) + h y'(t_n) + \tfrac{1}{2} h^2 y''(t_n) + \cdots \\ &= y_n + h f_n + \tfrac{1}{2} h^2 (f_{t_n} + f_{y_n} f_n) + \cdots \end{aligned} \right)$$

Match the coeffs.

$$\left(\begin{aligned} &\gamma_1 + \gamma_2 = 1 \\ &\gamma_2 \beta = \tfrac{1}{2} \\ &\gamma_2 \alpha = \tfrac{1}{2} \end{aligned} \right. \rightarrow$$

4 unknowns and 3 eqs

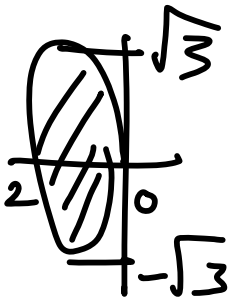
$\downarrow$

α as a free parameter.

$$\rightarrow \gamma_2 = \frac{1}{2\alpha}, \quad \beta = \alpha, \quad \gamma_1 = 1 - \frac{1}{2\alpha}$$

$$\text{RK2}: \begin{cases} k_1 = h f(y_n, t_n) \\ k_2 = h f(y_n + \alpha k_1, t_n + \alpha h) \\ y_{n+1} = y_n + \left(1 - \frac{1}{2\alpha}\right) k_1 + \frac{1}{2\alpha} k_2 \end{cases}$$

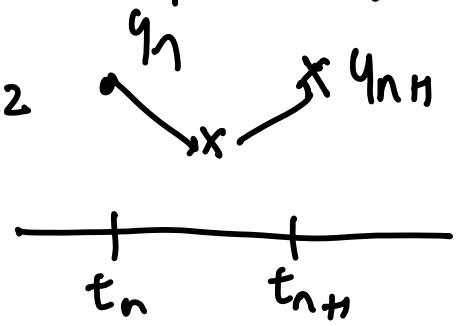
$0 < \alpha < 1$



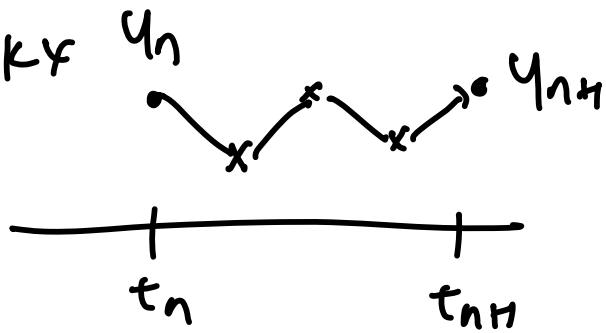
model prob. $y' = \lambda y \longrightarrow y_{n+1} = y_n \left(1 + \lambda h + \frac{1}{2}\lambda^2 h^2\right)$

* 2nd-order accurate

RK2



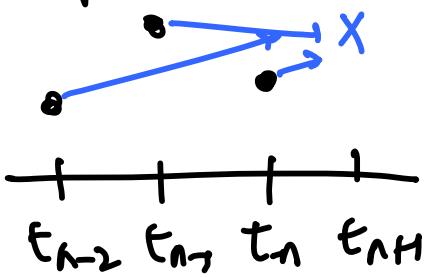
RK4



$y' = f(y, t)$: EE, IE, TR, LTR, PC, RK2, RK4

4.9 Multistep methods

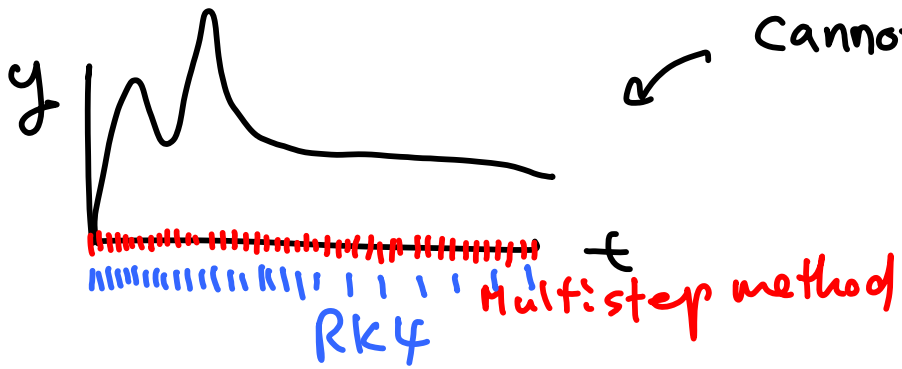
Higher-order accuracy is achieved by using data at previous timesteps, $t_{n-1}, t_{n-2}, t_{n-2}, \cdots$ & t_n



price : storage & memory

not self-starting

cannot change timestep size during computation.



- Leapfrog method (LF)

$$y' = f(y,t) \longrightarrow \frac{y_{n+1}-y_n}{h} = \begin{cases} f_n & \text{EE} \\ f_{n+1} & \text{IE} \\ \frac{1}{2}(f_n+f_{n+1}) & \text{TR} \end{cases}$$

$$\downarrow$$

$$\frac{y_{n+1}-y_{n-1}}{2h} = f(y_n,t_n) \rightarrow \boxed{y_{n+1} = y_{n-1} + 2h\, f(y_n,t_n)}$$

(CD2)

leapfrog method

not self-starting (get y_1 using EE or RK2...)

1 ft. evaluation → 2nd-order acurate

$\Downarrow$

too good to be true!

model prob. $y' = \lambda y \xrightarrow[LF]{} y_{n+1} = y_{n-1} + 2h\lambda y_n$

$\rightarrow y_{n+1} - 2\lambda h y_n - y_{n-1} = 0 \rightarrow \sigma^{n+1} - 2\lambda h \sigma^n - \sigma^{n-1} = 0$

Assume $y_n = \sigma^n y_0$

$\rightarrow \sigma^2 - 2\lambda h \sigma - 1 = 0 \rightarrow \sigma = \lambda h \pm \sqrt{\lambda^2 h^2 + 1}$

two roots! $\rightarrow y_n = c_1 \sigma_1^n + c_2 \sigma_2^n$

$\sigma_1 = \lambda h + \sqrt{\lambda^2 h^2 + 1} = 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 \boxed{- \frac{1}{8}\lambda^4 h^4} + \cdots$ 2nd-order accurate

$\left(e^{\lambda h} = 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 + \frac{1}{6}\lambda^3 h^3 + \cdots \right)$

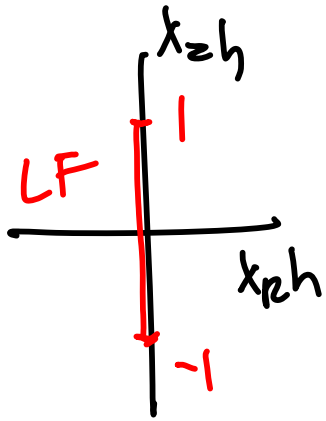
$\sigma_2 = \lambda h - \sqrt{\lambda^2 h^2 + 1} = -1 + \lambda h - \frac{1}{2}\lambda^2 h^2 + \frac{1}{8}\lambda^4 h^4 + \cdots$

$\Rightarrow$ σ_1 shows that the method is 2nd-order accurate.

σ_2 is the spurious root and has no physical meaning.

for real & negative λ, $|\sigma_2| > 1$ $\rightarrow$ LF is unstable.

for $\lambda = i\omega$, $\sigma = \lambda h \pm \sqrt{\lambda^2 h^2 + 1} = i\omega h \pm \sqrt{1-\omega^2 h^2}$



if $|\omega h| < 1$, $|\sigma|^2 = (\omega h)^2 + (1-\omega^2 h^2) = 1$ $\therefore$ no amplitude error

$|\omega h| > 1$, $|\sigma|^2 = |\omega h \pm \sqrt{\omega^2 h^2 - 1}| > 1$, unstable.

general sol. $y_n = c_1 \sigma_1^n + \underbrace{c_2 \sigma_2^n}_{\text{spurious sol.}}$

Find c_1 & c_2

$n=0$: $y_0 = c_1 + c_2$

let y_1 be the sol. @ $n=1$ obtained by some other numerical sol.

$n=1$: $y_1 = c_1 \sigma_1 + c_2 \sigma_2$

$$c_1 = \frac{y_1 - y_0 \sigma_2}{\sigma_1 - \sigma_2}, \quad c_2 = \frac{-y_1 + y_0 \sigma_1}{\sigma_1 - \sigma_2}$$

If we choose $y_1 = y_0 \sigma_1$, $c_2 = 0$

Then, the spurious root is completely suppressed.

In general, the starting scheme plays a role in determining the level of contribution of the spurious root. However, even if the spurious root is suppressed initially,

the round-off error can restart it.

- 2nd-order Adams-Bashforth method (AB2)