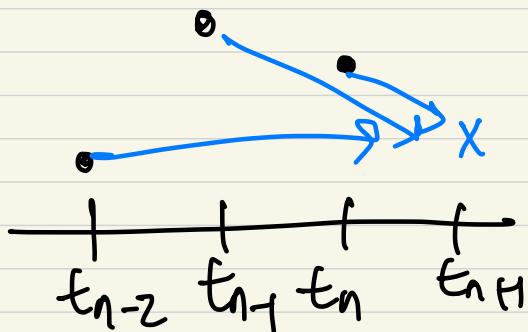


4.9

Multistep methods

$$t_{n-1} \rightarrow t_{n+1}$$

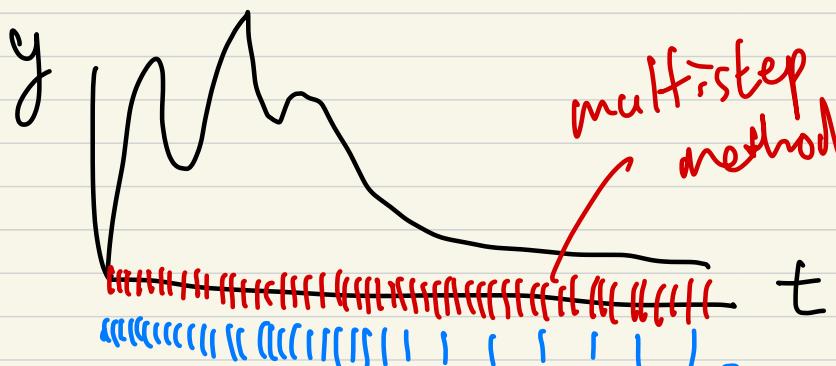
Higher-order accuracy is achieved by using data at previous timesteps, $t_{n-1}, t_{n-2}, t_{n-3}, \dots, t_0$.



price: storage & memory
not self-starting

should use a scheme for starting such as E-E

$(y_{n-2}, y_{n-1}, y_n \rightarrow y_{n+1})$
 $n=0, y_{n-1} \text{ & } y_{n-2} \text{ do not exist}$



Cannot change timestep size during computation
($\because \delta t$ is fixed)

RK4 \leftarrow change of δt during computation can

- Leapfrog method (LF)

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

↓

$$\left(\frac{y_{n+1} - y_n}{h} = f_n \quad \text{EE} \atop f_{n+1} \quad \text{IE} \right)$$

$$\frac{1}{2}(f_{n+1} + f_n) \quad \text{TR}$$

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

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

leapfrog
method

not self-starting (get y_1 using
EE or R(2...))

1 ft. evaluation → 2nd-order
accurate

too good to be true!

model prob. $y' = \lambda y$

$$\xrightarrow{\text{LF}} y_{n+1} = y_{n-1} + 2h \lambda y_n$$

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

$$\text{Assume } y_n = 6^n y_0 \rightarrow 6^{n+1} y_0 - 2\lambda h 6^n y_0 - 6^{n-1} y_0 = 0$$

$$\rightarrow 6^2 - 2\lambda h 6 - 1 = 0 \rightarrow 6 = \lambda h \pm \sqrt{\lambda^2 h^2 + 1}$$

two roots! $\rightarrow y_n = c_1 6_1^n + c_2 6_2^n$

$$6_1 = \lambda h + \sqrt{\lambda^2 h^2 + 1} = 1 + \lambda h + \frac{1}{2} \lambda^2 h^2 - \frac{1}{8} \lambda^4 h^4 + \dots$$

$$(e^{\lambda h} = 1 + \lambda h + \frac{1}{2} \lambda^2 h^2 + \frac{1}{6} \lambda^3 h^3 + \dots) \therefore \text{2nd-order accurate}$$

$$6_2 = \lambda h - \sqrt{\lambda^2 h^2 + 1} = -1 + \lambda h - \frac{1}{2} \lambda^2 h^2 + \frac{1}{8} \lambda^4 h^4 + \dots$$

$\Rightarrow 6_1$ shows that the method is 2nd-order accurate.

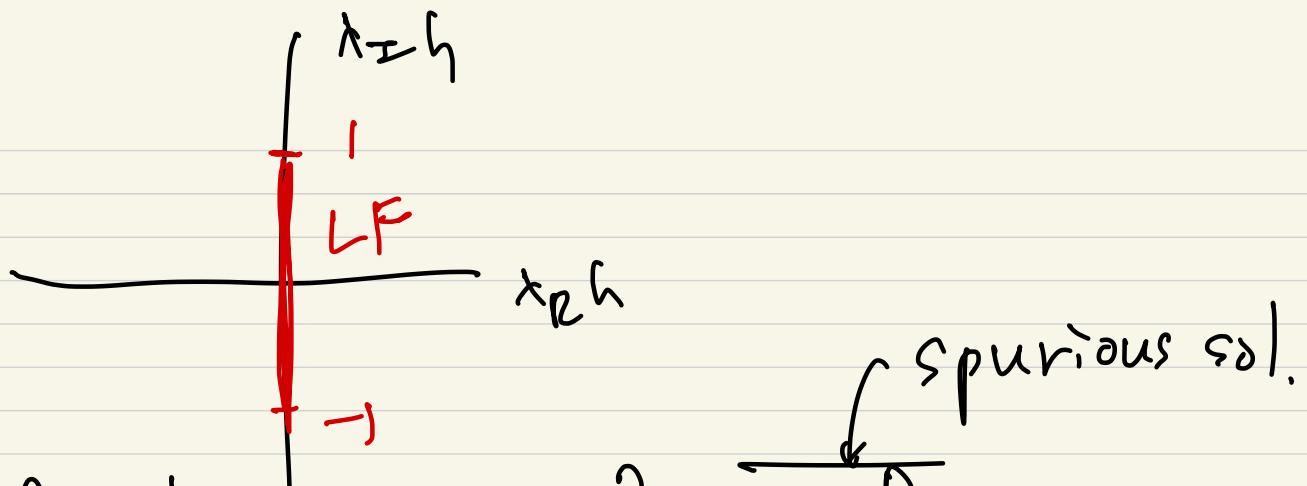
6_2 is the "spurious" root and has no physical meaning.

for real & negative λ , $|6_2| > 1 \rightarrow$ LF is unstable.

$$\text{for } \lambda = i\omega, 6 = \lambda h \pm \sqrt{\lambda^2 h^2 + 1} = i\omega h \pm \sqrt{1 - \omega^2 h^2}$$

$$\text{if } |\omega h| < 1, |6|^2 = (\omega h)^2 + (1 - \omega^2 h^2) = 1 \quad \begin{matrix} \text{"no amp."} \\ \text{error!} \end{matrix}$$

$$|\omega h| > 1, |6|^2 = |\omega h \pm \sqrt{\omega^2 h^2 - 1}|^2 > 1 \quad \therefore \text{unstable}$$



General sol. $y_n = c_1 \theta_1^n + c_2 \theta_2^n$

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.

$$c_1 = \frac{y_1 - y_0 \theta_2}{\theta_1 - \theta_2}$$

$$c_2 = \frac{-y_1 + y_0 \theta_1}{\theta_1 - \theta_2}$$

If we choose $y_1 = y_0 \theta_1$, $c_2 = 0$.

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 errors can restart it.

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

$$y_{n+1} = y_n + h \underbrace{y'_n}_{f(y_n, t_n)} + \frac{1}{2} h^2 \underbrace{y''_n}_{\frac{y'_n - y'_{n-1}}{h}} + \frac{1}{6} h^3 \underbrace{y'''_n}_{\Theta(h)} + \dots$$

$\Theta(h^3)$

r. neglect $\Theta(h)$ term

$$= y_n + h y'_n + \frac{1}{2} h (y'_n - y'_{n-1}) + \Theta(h^3)$$

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

$$\rightarrow \boxed{y_{n+1} = y_n + \frac{1}{2} h (3 f(y_n, t_n) - f(y_{n-1}, t_{n-1})) + \Theta(h^3)}$$

AB₂

$$\text{or } \frac{y_{n+1} - y_n}{h} = \frac{1}{2} (3 f(y_n, t_n) - f(y_{n-1}, t_{n-1})) + \Theta(h^2)$$

globally 2nd-order accurate
explicit
not self-starting

$$g' = \lambda y : q_{n+1} = q_n + \frac{1}{2} h (3\lambda q_n - \lambda q_{n-1})$$

$$\rightarrow q_{n+1} - (1 + \frac{3}{2}\lambda h) q_n + \frac{1}{2}\lambda h q_{n-1} = 0$$

$$\text{Assume } q_n = \delta^n q_0. \rightarrow \delta^2 - (1 + \frac{3}{2}\lambda h) \delta + \frac{1}{2}\lambda h = 0$$

$$\rightarrow \delta = \frac{1}{2} \left[(1 + \frac{3}{2}\lambda h) \pm \sqrt{1 + \lambda h + \frac{9}{4}\lambda^2 h^2} \right]$$

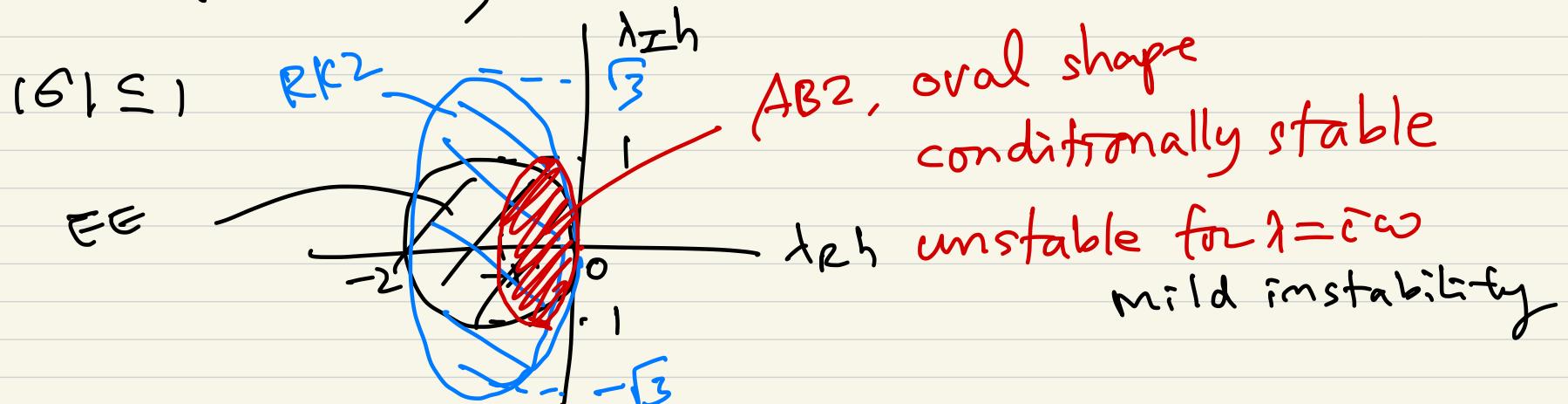
$(\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{3}{64}x^3 + \dots)$

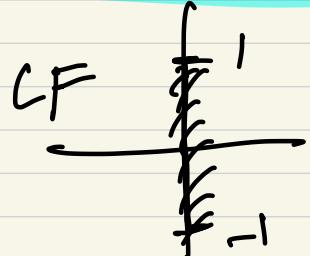
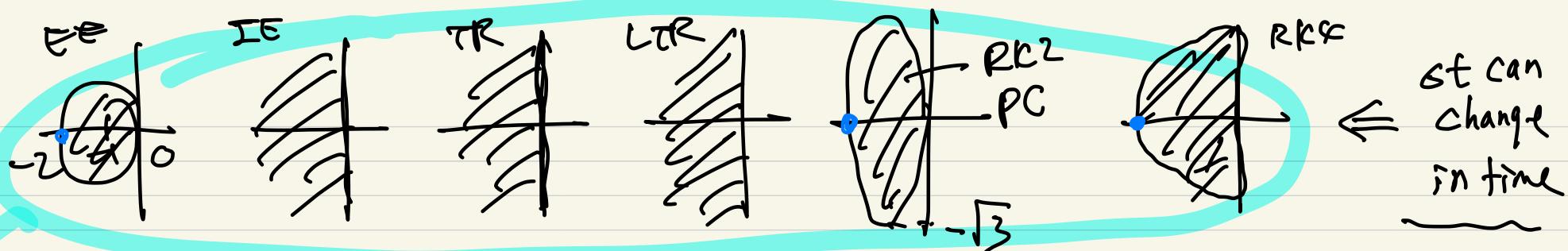
$$\rightarrow \delta_1 = 1 + \lambda h + \frac{1}{2}\lambda^2 h^2 \quad \text{[+ --]} \quad \text{2nd-order accurate}$$

$$\delta_2 = \frac{1}{2}\lambda h - \frac{1}{2}\lambda^2 h^2 + \dots, \text{ spurious root.}$$

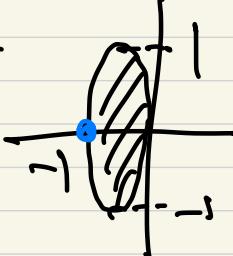
$$q_n = c_1 \delta_1^n + c_2 \delta_2^n$$

As $h \rightarrow 0$, $\delta_2 \rightarrow 0$ less dangerous.



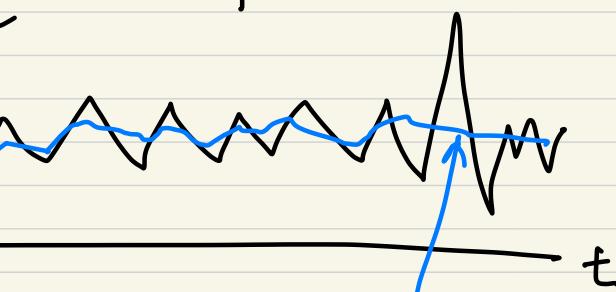


AB^2



$$\begin{cases} y' = \lambda y \\ y' = f(y, \epsilon) \end{cases}$$

$\sigma t = h$



$$h = \text{hold } \alpha + h_{\text{new}}(1-\alpha)$$

$$0 < \alpha \leq 1$$

λ real & negative
stability $\rightarrow |\lambda h| \leq *$

$$\begin{aligned} \rightarrow h &\leq \frac{*}{|\lambda|} \quad \xrightarrow{\lambda(t)} \lambda(t) \\ h_{\max} &= \frac{*}{|\lambda|} \quad \rightarrow h_{\max}^* = 0.9 \frac{*}{|\lambda|} \\ &\sim 0.95 \end{aligned}$$

4.10

System of first-order ODEs

ex) chemical reaction

$$\left\{ \begin{array}{l} \frac{dy_1}{dt} = \alpha_{11} y_1^2 + \alpha_{12} y_1 y_2 + \dots \\ \frac{dy_2}{dt} = \alpha_{21} y_1 y_2 + \alpha_{22} y_2^2 + \dots \\ \dots \\ \dots \end{array} \right.$$

higher-order
ODE

→ System of
1st-order ODEs

ex) Blasius e.g. (laminar
boundary
flow)

$$f''' + ff'' = 0$$

$$y_1 = f$$

$$y_2 = y_1' = f'$$

$$y_3 = y_2' = f''$$

$$y_3' = f''' = -y_1 y_3$$

$$\rightarrow \left\{ \begin{array}{l} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = -y_1 y_3 \end{array} \right.$$

From the conceptual point of view, there is only one fundamental difference between numerical sols. of one SDE and that of a system of ODEs. → stiffness

single ODE

$$\frac{dy}{dt} = f(y, t)$$

vs. system of ODE's

$$\frac{dy_i}{dt} = f_i(t, y_1, y_2, \dots, y_m)$$

$$i = 1, 2, \dots, m$$

model prob.

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

$$\frac{dy}{dt} = Ay$$

Assume that A has a complete set of eigen vectors.

$$\rightarrow A = S^{-1}\Lambda S \quad \Lambda: \text{eigenvalue matrix}$$

$$\frac{dy}{dt} = Ay = S^{-1}\Lambda Sy$$

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix}$$

$$S \frac{dy}{dt} = \Lambda Sy \rightarrow \frac{d}{dt}(Sy) = \Lambda Sy, \quad u \equiv Sy$$

$$\rightarrow \frac{dy}{dt} = \Lambda u \rightarrow \underbrace{\frac{du_i}{dt}}_{\text{EE}} = \lambda_i u_i \quad i=1,2,\dots,m$$

$$\text{EE: } u_i^n = (1 + \lambda_i h)^n u_i^0$$

$$\text{for stability } |1 + \lambda_i h| \leq 1 \quad i=1,2,\dots,m$$

\downarrow
largest eigenvalue

\rightarrow smallest h

$$\frac{dy}{dt} = Ay$$

$$\text{EE: } y_{n+1} = y_n + h A y_n = (I + hA) y_n$$

$$\rightarrow y_n = (I + hA)^n y_0 \equiv B^n y_0$$

$$\text{for stability } |\alpha_i| \leq 1 \quad \alpha_i : \text{eigenvalues of } B$$

$$\alpha_i = 1 + h \lambda_i \quad \lambda_i : \text{eigenvalues of } A$$

$$\rightarrow |\alpha_i| = |1 + h \lambda_i| \leq 1$$

$$\text{for real & negative } \lambda_i, -1 \leq 1 + h \lambda_i \leq 1$$

$$\rightarrow h \leq \frac{2}{|\lambda_{\max}|}$$

$$\text{Stiffness} \equiv \frac{|\lambda_{\max}|}{|\lambda_{\min}|} \gg 1 \rightarrow \text{System is stiff.}$$

Stiffness can arise in physical systems with several degrees of freedom, but with widely different response times.

ex) a system composed of two springs, one very stiff and the other very flexible

ex) a mixture of chemical species with very different reaction times.

$$\begin{cases} u' = 998u - 1998v \\ v' = -999u - 1999v \end{cases} \quad u(0) = v(0) = 1$$

$$\begin{cases} u' \\ v' \end{cases} = \begin{pmatrix} 998 & -1998 \\ -999 & -1999 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

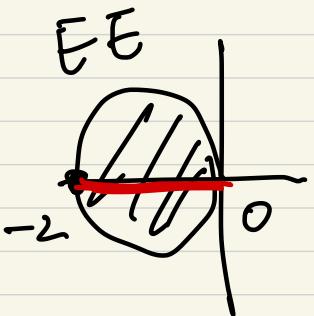
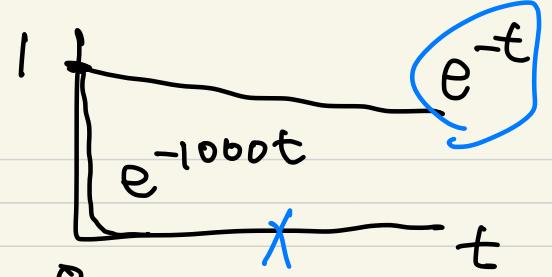
$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \underbrace{\begin{pmatrix} 998 & -1998 \\ -999 & -1999 \end{pmatrix}}_A \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\det(A - xI) = 0$$

$$\rightarrow \begin{aligned} \lambda_1 &= -1 \\ \lambda_2 &= -1000 \end{aligned} \quad) \text{stiff}$$

exact sol.

$$\begin{cases} u = 4e^{-t} - 3e^{-1000t} \\ v = -2e^{-t} + 3e^{-1000t} \end{cases}$$



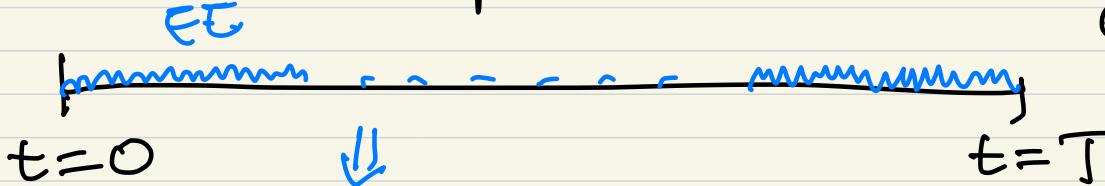
λ real & negative

$$|\lambda_R h| \leq 2 \rightarrow h \leq \frac{2}{|\lambda_1|} \text{ or } \frac{2}{|\lambda_2|} \Rightarrow h_{\max} = \frac{2}{1000} = \frac{1}{500}$$

Advance 5 timesteps, $t = 5 \cdot \frac{1}{500} = 0.01$

$$\rightarrow e^{-1000t} = 4.5 \times 10^{-5}, e^{-t} = 0.99$$

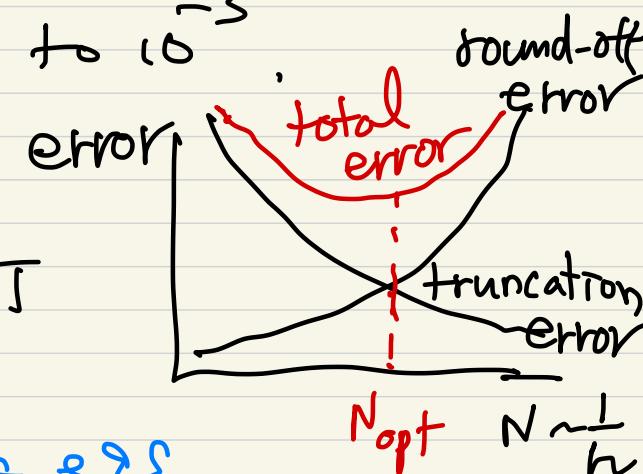
take 3500 timesteps to drive e^{-t} to 10^{-3} .



use implicit methods!

↳ nonlinear algebraic eqs.

↓
iteration or linearize to avoid iteration



$$\frac{dy}{dt} = \underline{f}(y_1, y_2, \dots, y_m) = \underline{f}(y)$$

TR : $y_{n+1} - y_n = \frac{h}{2} [\underline{f}(y_{n+1}) + \underline{f}(y_n)] + O(h^3)$

① Linearization

$$f_i(y_{n+1}) = f_i(y_n) + (y_{i,n+1} - y_{i,n}) \left. \frac{\partial f_i}{\partial y_i} \right|_n + (y_{2,n+1} - y_{2,n}) \left. \frac{\partial f_i}{\partial y_2} \right|_n$$

$$+ \dots + (y_{m,n+1} - y_{m,n}) \left. \frac{\partial f_i}{\partial y_m} \right|_n + O(h^2), \quad i=1,2,\dots,m$$

$$\rightarrow \underline{f}(y_{n+1}) = \underline{f}(y_n) + A(y_{n+1} - y_n) + O(h^2) \quad \text{neglect}$$

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \dots & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \dots & \frac{\partial f_2}{\partial y_m} \\ \vdots & & & \\ \frac{\partial f_m}{\partial y_1} & \frac{\partial f_m}{\partial y_2} & \dots & \frac{\partial f_m}{\partial y_m} \end{pmatrix}_{t_n}$$

Jacobian matrix
full matrix

$$\rightarrow \underline{y}_{n+1} - \underline{y}_n = \frac{h}{2} \left[f(\underline{y}_n) + A(\underline{y}_{n+1} - \underline{y}_n) + f(\underline{y}_n) \right] + O(h^3)$$

$$\rightarrow \boxed{\left(I - \frac{h}{2} A \right) \underline{y}_{n+1} = \left(I - \frac{h}{2} A \right) \underline{y}_n + h f(\underline{y}_n)}$$

linearized
TR method

full matrix \rightarrow inverting $(I - \frac{h}{2} A)$ requires $O(m^3)$ operations.

\rightarrow may require iterative methods to solve this eq. at each time step.

linearization may have errors for strongly nonlinear prob.

② Iterative method w/o linearization

$$TR : \boxed{\underline{y}_{n+1} - \underline{y}_n = \frac{h}{2} [f(\underline{y}_{n+1}) + f(\underline{y}_n)] + O(h^3)}$$

Newton-iterative method $\frac{F(x) = 0}{k: \text{iteration index}}$

$$\frac{dF}{dx} = \frac{F^{k+1} - F^k}{x^{k+1} - x^k}$$

$$\rightarrow \boxed{\frac{dF^k}{dx} (x^{k+1} - x^k) = F^{k+1} - F^k}$$

$$\underline{F} = \underline{y}_{n+1} - \underline{y}_n - \frac{h}{2} [f(\underline{y}_{n+1}) + f(\underline{y}_n)] = 0$$

$$\rightarrow \left. \frac{\partial F_i}{\partial y_j} \right|^k (\underline{y}_j^{k+1} - \underline{y}_j^k) = -F_i^k \quad i=1, 2, \dots, M$$

k: iteration index

$$\left. \frac{\partial F_i}{\partial y_j} \right|_0^k = \left. \frac{\partial y_i}{\partial y_j} \right|_0^k - 0 - \frac{h}{2} \left[\left. \frac{\partial f_i}{\partial y_j} \right|_0^k + 0 \right]$$

$$= \delta_{ij} - \frac{h}{2} \left. \frac{\partial f_i}{\partial y_j} \right|_0^k$$

$$\rightarrow \boxed{(\underline{I} - \frac{h}{2} \underline{A}^k) (\underline{y}^{k+1} - \underline{y}^k) = -\underline{F}^k} = -\underline{y}^k + \underline{y}_n + \frac{h}{2} [f(\underline{y}^k) + f(\underline{y}_n)]$$

full matrix

$$k=0 : \underline{y}^0 = \underline{y}_n$$

Solve this system of eqs. iteratively at each time step,

→ 3~4 iterations / time step if initial guess

\underline{y}^0 is good.

(but it is good)

* Inherent instability

Consider $y'' - k^2 y = 0$, $y(0) = y_0$, $y'(0) = -ky_0$ ($k > 0$)

→ exact sol. $y = y_0 e^{-kx}$ well-behaved

numerical sol. $y = c_1 e^{-kx} + \underline{c_2 e^{+kx}}$

truncation and round-off errors will be exponentially amplified → sol. diverges.

⇒ Inherently unstable

None of the standard numerical methods will provide the correct sol.!