§ Initial-Value Problems for ODEs

GIVEN: $\frac{dy}{dt} = y'(t) = f(t,y), \quad y(a) = y_a$

FIND: $y(t)$ for $a < t \le b$

- numerical errors (round-off and truncation errors)

  Consider a perturbed system:

$$\frac{dz}{dt} = f(t,z) + \delta(t), \quad a < t \le b$$
$$z(a) = y_a + \varepsilon_0$$

- Does $z(t) \approx y(t)$?

  (i) (uniqueness) a unique solution $y(t)$ exists

(ii) (well-posed) for any $\varepsilon>0, \exists \kappa(\varepsilon) > 0$ such that

whenever $|\varepsilon_0| < \varepsilon$ and $\delta(t) \in C^0[a,b]$ with $|\delta(t)| < \varepsilon$,

a unique soltion $z(t)$ to the problem

$$\frac{dz}{dt} = f(t,z) + \delta(t), \quad a < t \le b$$
$$z(a) = y_a + \varepsilon_0$$

exists with

$$|z(t) - y(t)| < \kappa(\varepsilon) \cdot \varepsilon, \quad a < t \le b$$

GIVEN: $\frac{dy}{dt} = y'(t) = f(t,y), \quad y(a) = y_a$

FIND: $y(t)$ for $a < t \le b$

- sufficient conditions for the problem to be well posed:

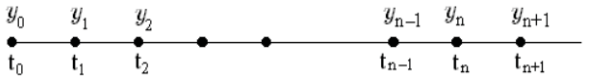
1 if $|f(t,y_2) - f(t,y_1)| \le L|y_2 - y_1|$, (Lipschitz condition);

2 if $f \in C^0[a,b]$ and satisfies $\left|\frac{\partial f}{\partial y}\right| \le L$ for some $L > 0$

GIVEN: $\frac{dy}{dt} = y'(t) = f(t,y), \quad y(a) = y_a$

FIND: $y(t)$ for $a < t \le b$

- Discretization: provide $y_n = y(t_n)$



GIVEN: $y_0, y_1, y_2, \cdots, y_n$

FIND: y_{n+1}

• Classification:

$$\begin{cases}\text{Explicit Schemes: } y_{n+1}=F\left(y_0,y_1,y_2,\cdots,y_n\right)\\ \text{Implicit Schemes: } y_{n+1}=F\left(y_0,y_1,y_2,\cdots,y_n,y_{n+1}\right)\end{cases}$$

$$\begin{cases}\text{one-step method:} & y_{n+1}=F\left(y_n\right)\\ & \text{or} \quad y_{n+1}=F\left(y_n,y_{n+1}\right)\\ \text{multi-step method:} & y_{n+1}=F\left(y_{n-m+1},\cdots,y_n\right)\\ & \text{or} \quad y_{n+1}=F\left(y_{n-m+1},\cdots,y_n,y_{n+1}\right)\end{cases}$$

§ Taylor methods (explicit, one-step method)

GIVEN: y_n and $y'(t)=f(t,y)$

FIND: y_{n+1}

• Taylor method of order *k*:

$$y_{n+1}=y\left(t_{n+1}\right)=y\left(t_n+dt\right)$$

$$=y\left(t_n\right)+dt\ y'\left(t_n\right)+\frac{1}{2}dt^2\ y''\left(t_n\right)+\cdots+\frac{1}{k!}dt^k\ y^{(k)}\left(t_n\right)+O\left(dt^{k+1}\right)$$

• need compute $y\left(t_n\right),y'\left(t_n\right),y''\left(t_n\right),\cdots,y^{(k)}\left(t_n\right)$

$$y\left(t_n\right)=y_n$$

$$y'\left(t_n\right)=f\left(t_n,y_n\right)=f_n$$

$$y''\left(t_n\right)=\left(\frac{d}{dt}y'(t)\right)_{t=t_n}=\left(\frac{df}{dt}\right)_{t=t_n,y=y_n}\equiv f_n'$$

$$y'''\left(t_n\right)=\left(\frac{d}{dt}y''(t)\right)_{t=t_n}=\left(\frac{d^2f}{dt^2}\right)_{t=t_n,y=y_n}\equiv f_n''$$

• notice that $f=f(t,y)=f\left(t,y(t)\right)$

$$\frac{df}{dt}=\frac{d}{dt}f\left(t,y(t)\right)=\frac{\partial f}{\partial t}+\frac{dy}{dt}\cdot\frac{\partial f}{\partial y}=\left(\frac{\partial}{\partial t}+f\cdot\frac{\partial}{\partial y}\right)f(t,y)$$

$$\frac{d^2f}{dt^2}=\left(\frac{\partial}{\partial t}+f\frac{\partial}{\partial y}\right)\left(\frac{\partial}{\partial t}+f\frac{\partial}{\partial y}\right)f(t,y)$$

$$=\left(\frac{\partial^2}{\partial t^2}+\frac{\partial}{\partial t}\left(f\frac{\partial}{\partial y}\right)+\left(f\frac{\partial}{\partial y}\right)\frac{\partial}{\partial t}+f\frac{\partial}{\partial y}\left(f\frac{\partial}{\partial y}\right)\right)f$$

$$=\frac{\partial^2}{\partial t^2}+\frac{\partial f}{\partial t}\cdot\frac{\partial f}{\partial y}+2f\cdot\frac{\partial^2 f}{\partial t\partial y}+f\left(\frac{\partial f}{\partial y}\right)^2+f^2\frac{\partial^2 f}{\partial y^2}$$

• Taylor method of order k:

$$y_{n+1} \approx y(t_n) + dt\, y'(t_n) + \frac{1}{2} dt^2\, y''(t_n) + \cdots + \frac{1}{k!} dt^k\, y^{(k)}(t_n)$$

• truncation error per time step (from t_n to t_{n+1}) $\sim O(dt^{k+1})$

• accumulated truncation error from $t = 0$ to $t = T$:

$$T/dt \cdot O(dt^{k+1}) \sim O(T \cdot dt^k)$$

• **forward Euler method** ($k = 1$, explicit, one-step):

$$y_{n+1} \approx y_n + dt \cdot f(t_n, y_n)$$

accumulated truncation error $\sim O(T \cdot dt)$

simple but poor accuracy

i.e. $y'(t_n) = f(t_n, y_n) \approx \frac{y_{n+1} - y_n}{dt}$

forward difference

• higher order (higher k): better accuracy but too much trouble

seldom in use

• **backward Euler method** (implicit, one-step):

$$y'(t_{n+1}) = f(t_{n+1}, y_{n+1}) \approx \frac{y_{n+1} - y_n}{dt}$$

backward difference

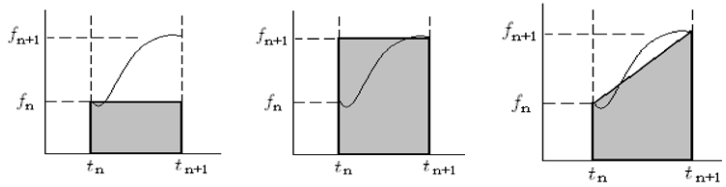
$$y_{n+1} \approx y_n + dt \cdot f(t_{n+1}, y_{n+1})$$

accumulated truncation error $\sim O(T \cdot dt)$

GIVEN: $\frac{dy}{dt} = y'(t) = f(t, y), \quad y(a) = y_a$

FIND: $y(t)$ for $a < t \le b$

exact: $y_{n+1} = y(t_n + dt) = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t))\, dt$



forward $dt \cdot f(t_n, y_n)$

backward $dt \cdot f(t_{n+1}, y_{n+1})$

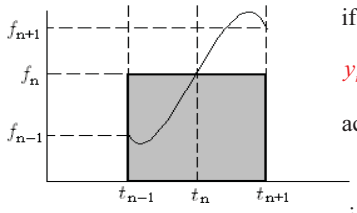
trapezoidal

• **Crank-Nilcoson method** (trapezoidal, implicit, one-step):

$$y_{n+1} \approx y_n + \frac{dt}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$$

accumulated truncation error $\sim O\left(T \cdot dt^2\right)$

• **leap-frog method** (explicit, two-step):



if $t_{n+1} - t_n = t_n - t_{n-1} = dt$

$$y_{n+1} \approx y_{n-1} + 2dt\, f(t_n, y_n)$$

accumulated truncation error $\sim O\left(T \cdot dt^2\right)$

i.e. $y'(t_n) = f(t_n, y_n) \approx \dfrac{y_{n+1} - y_{n-1}}{2dt}$

~ central difference

§ **Runge-Kutta methods** (explicit, one-step)

§ 2^{nd} order method: accumulated truncation error ~ $O(\ T \cdot dt^2\ )$

$$k_1 \equiv dt \cdot f(t_n, y_n)$$

$$k_2 \equiv dt \cdot f(t_n + \beta\, dt, y_n + \alpha\, k_1)$$

$$y_{n+1} \approx y_n + \gamma_1 k_1 + \gamma_2 k_2$$

• What values of α, β, γ_1, γ_2 are required to obtain an accuracy of order 2?

$$k_1 = dt \cdot f_n$$

$$k_2 = dt \cdot \left(f_n + (\beta\, dt)\frac{\partial f_n}{\partial t} + (\alpha\, k_1)\frac{\partial f_n}{\partial y} + O\left(dt^2\right)\right)$$

$$= dt \cdot \left(f_n + (\beta\, dt)\frac{\partial f_n}{\partial t} + (\alpha\, dt\, f_n)\frac{\partial f_n}{\partial y} + O\left(dt^2\right)\right)$$

P.S. $O\left(dt^2\right) = \frac{1}{2}(\beta\, dt)^2 \frac{\partial^2 f_n}{\partial t^2} + (\beta\, dt)(\alpha k_1)\frac{\partial^2 f_n}{\partial t \partial y} + \frac{1}{2}(\alpha k_1)^2 \frac{\partial^2 f_n}{\partial y^2} + O\left(dt^3\right)$

$$y_{n+1} \approx y_n + \gamma_1 (dt\, f_n) + \gamma_2 dt\left(f_n + (\beta dt)\frac{\partial f_n}{\partial t} + (\alpha dt\, f_n)\frac{\partial f_n}{\partial y} + O\left(dt^2\right)\right)$$

$$= y_n + (\gamma_1 + \gamma_2) dt\, f_n + (\beta\gamma_2) dt^2 \frac{\partial f_n}{\partial t} + (\alpha\gamma_2) dt^2 f_n \frac{\partial f_n}{\partial y} + O\left(dt^3\right)$$

On the other hand, the Taylor series expansion of y_{n+1} about t_n is:

$$y_{n+1} = y_n + dt\, f_n + \frac{1}{2} dt^2 \frac{df_n}{dt} + O\left(dt^3\right)$$

$$= y_n + dt\, f_n + \frac{1}{2} dt^2 \left(\frac{\partial f_n}{\partial t} + f_n \frac{\partial f_n}{\partial y}\right) + O\left(dt^3\right)$$

Thus, to have a scheme with a truncation error/step ~ $O(dt^3)$, we need

$$(\gamma_1 + \gamma_2) = 1$$

$$\alpha\gamma_2 = \beta\gamma_2 = \frac{1}{2}$$

• **2nd Runge-Kutta method:**

$$k_1 \equiv dt \cdot f(t_n, y_n)$$

$$k_2 \equiv dt \cdot f(t_n + \beta\, dt, y_n + \alpha\, k_1)$$

$$y_{n+1} \approx y_n + \gamma_1 k_1 + \gamma_2 k_2$$

$$(\gamma_1 + \gamma_2) = 1$$

$$\alpha\gamma_2 = \beta\gamma_2 = \frac{1}{2}$$

truncation error $\sim dt^3 \left\{ \frac{1}{4}\left(\frac{2}{3} - \alpha\right)\left(\frac{\partial}{\partial t} + f\frac{\partial}{\partial y}\right)^2 f + \frac{1}{6}\frac{\partial f}{\partial y}\left(\frac{\partial}{\partial t} + f\frac{\partial}{\partial y}\right) f \right\}$

- **2nd Runge-Kutta method:**

$$y_{n+1} \approx y_n + \gamma_1 \cdot dtf\left(t_n, y_n\right) + \gamma_2 \cdot dtf\left(t_n + \beta\, dt, y_n + \alpha\, k_1\right)$$

(1) $\gamma_1 = 0,\ \gamma_2 = 1,\ \alpha = \beta = 1/2$ (midpoint method)

$$y_{n+1} = y_n + dt\, f\left(t_n + \frac{dt}{2}, y_n + \frac{dt}{2} f_n\right)$$

(2) $\gamma_1 = \gamma_2 = 1/2,\ \alpha = \beta = 1$ (midpoint Euler method)

$$y_{n+1} = y_n + \frac{dt}{2} f_n + \frac{dt}{2} f\left(t_n + dt,\ y_n + dt\, f_n\right)$$

(3) $\gamma_1 = 1/4, \gamma_2 = 3/4,\ \alpha = \beta = 2/3$ (Heun's method)

$$y_{n+1} = y_n + \frac{dt}{4} f_n + \frac{3dt}{4} f\left(t_n + \frac{2dt}{3},\ y_n + \frac{2dt}{3} f_n\right)$$

- 3rd Runge-Kutta method:

$$k_1 = dt \cdot f\left(t_n, y_n\right)$$
$$k_2 = dt \cdot f\left(t_n + \beta_1\, dt, y_n + \alpha_1 k_1\right)$$
$$k_3 = dt \cdot f\left(t_n + \beta_2\, dt, y_n + \alpha_2 k_1 + \alpha_3 k_2\right)$$

$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2 + \gamma_3 k_3$$

8 parameters!

6 constraints for O(dt^4) per time step!

- 4th Runge-Kutta method (the most common one):

$$k_1 = dt \cdot f\left(t_n, y_n\right)$$
$$k_2 = dt \cdot f\left(t_n + \frac{dt}{2}, y_n + \frac{k_1}{2}\right)$$
$$k_3 = dt \cdot f\left(t_n + \frac{dt}{2}, y_n + \frac{k_2}{2}\right)$$
$$k_4 = dt \cdot f\left(t_n + dt, y_n + k_3\right)$$

$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) + O\left(dt^5\right)$$

accumulated truncation error $\sim O\left(T \cdot dt^4\right)$

§ **Adams-Bashforth methods** of order m (explicit, multi-step)

$f_j \equiv f\left(t_j, y_j\right)$

$$y_{n+1} = y_n + dt \sum_{j=0}^{m-1} b_j f_{n-j} + O\left(dt^{m+1}\right)$$

$$= y_n + dt \cdot \left(b_0 f_n + b_1 f_{n-1} + \cdots + b_{m-1} f_{n-m+1}\right) + O\left(dt^{m+1}\right)$$

e.g. Adams-Bashforth methods of order 3 (explicit, multi-step):

$$y_{n+1} = y_n + dt \cdot \left(b_0 f_n + b_1 f_{n-1} + b_2 f_{n-2}\right)$$

• Adams-Bashforth methods of order 3 (explicit, multi-step):

$$y_{n+1} = y_n + dt\cdot\left(b_0 f_n + b_1 f_{n-1} + b_2 f_{n-2}\right)$$

$$= y_n + dt\cdot\left\{\begin{array}{l} b_0 f_n \\ +b_1\left(f_n - dt\, f_n' + \frac{1}{2}dt^2 f_n'' - \frac{1}{6}dt^3 f_n''' + O\left(dt^4\right)\right) \\ +b_2\left(f_n - 2dt\, f_n' + 2dt^2 f_n'' - \frac{4}{3}dt^3 f_n''' + O\left(dt^4\right)\right)\end{array}\right\}$$

$$= y_n + \left(b_0 + b_1 + b_2\right)dt\, f_n + \left(-b_1 - 2b_2\right)dt^2 f_n' + \frac{\left(b_1 + 4b_2\right)}{2}dt^3 f_n'' + O\left(dt^4\right)$$

$$= y_n + dt\, f_n + \frac{1}{2}dt^2 f_n' + \frac{1}{6}dt^3 f_n'' + O\left(dt^4\right)$$

Therefore,

$$\begin{cases} b_0 + b_1 + b_2 = 1 \\ -b_1 - 2b_2 = 1/2 \\ b_1 + 4b_2 = 1/3 \end{cases} \Rightarrow \begin{cases} b_0 = 23/12 \\ b_1 = -16/12 \\ b_2 = 5/12 \end{cases}$$

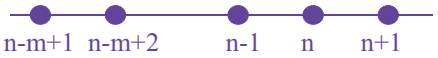
Conclusion:

$$y_{n+1} \approx y_n + \frac{dt}{12}\left(23 f_n - 16 f_{n-1} + 5 f_{n-2}\right)$$

⇨ Adams-Bashforth three-step method

⇨ accumulated truncation error ~ $O(T\cdot dt^3)$

§ **Adams-Moulton methods** of order m+1 (implicit, multi-step)



$$y_{n+1} = y_n + dt\sum_{j=1}^{m-1} b_j f_{n-j} + O\left(dt^{m+2}\right)$$

$$= y_n + dt\cdot\left(b_{-1} f_{n+1} + b_0 f_n + b_1 f_{n-1} + \cdots + b_{m-1} f_{n-m+1}\right) + O\left(dt^{m+2}\right)$$

e.g. Adams-Moulton methods of order m+1=3 (implicit, multi-step):

$$y_{n+1} = y_n + dt\cdot\left(b_{-1} f_{n+1} + b_0 f_n + b_1 f_{n-1}\right)$$

• Adams-Moulton methods of order m+1=3 (implicit, multi-step):

$$y_{n+1} = y_n + dt\cdot\left(b_{-1} f_{n+1} + b_0 f_n + b_1 f_{n-1}\right)$$

$$= y_n + dt\cdot\left\{\begin{array}{l} b_{-1}\left(f_n + dt\, f_n' + \frac{1}{2}dt^2 f_n'' + \frac{1}{6}dt^3 f_n''' + O\left(dt^4\right)\right) \\ +b_0 f_n \\ +b_1\left(f_n - dt\, f_n' + \frac{1}{2}dt^2 f_n'' - \frac{1}{6}dt^3 f_n''' + O\left(dt^4\right)\right)\end{array}\right\}$$

$$= y_n + \left(b_{-1} + b_0 + b_1\right)dt\, f_n + \left(b_{-1} - b_1\right)dt^2 f_n' + \frac{\left(b_{-1} + b_1\right)}{2}dt^3 f_n'' + O\left(dt^4\right)$$

$$= y_n + dt\, f_n + \frac{1}{2}dt^2 f_n' + \frac{1}{6}dt^3 f_n'' + O\left(dt^4\right)$$

Therefore,

$$\begin{cases} b_{-1}+b_0+b_1=1 \\ b_{-1}-b_1=1/2 \\ b_{-1}+b_1=1/3 \end{cases} \Rightarrow \begin{cases} b_{-1}=5/12 \\ b_0=8/12 \\ b_1=-1/12 \end{cases}$$

Conclusion:

$$y_{n+1} \approx y_n + \frac{dt}{12}(5f_{n+1}+8f_n-f_{n-1})$$

⇨ Adams-Moulton two-step method

⇨ accumulated truncation error ~ $O(T\cdot dt^3)$

§ predict-corrector methods

PREDICTOR: use Adams-Bashforth m-step method to predict y^*_{n+1} :

$$y^*_{n+1} = y_n + dt\sum_{j=0}^{m-1} b_j f_{n-j} + O\left(dt^{m+1}\right)$$

CORRECTOR: use Adams-Moulton m-step method to predict y_{n+1} :

$$y_{n+1} = y_n + dt\left(b_{-1}f^*_{n+1} + \sum_{j=0}^{m-1} b_j f_{n-j}\right) + O\left(dt^{m+2}\right)$$

$$f^*_{n+1} = f\left(t_{n+1}, y^*_{n+1}\right)$$

accumulated truncation error ~ $O(T\cdot dt^{m+1})$

advantage: explicit

disadvantage: need many more computations

§ General multi-step methods:

$$y_{n+1} \approx a_0 y_n + a_1 y_{n-1} + \cdots + a_{m-1} y_{n-m+1}$$
$$+ dt\left(b_{-1}f_{n+1} + b_0 f_n + b_1 f_{n-1} + \cdots + b_{m-1} f_{n-m+1}\right)$$
$$= y_n + dt\, f_n' + \frac{1}{2}dt^2 f_n'' + \frac{1}{6}dt^3 f_n''' + \cdots \quad \text{(expected)}$$

- $\begin{cases} b_{-1}=0 & \text{explicit schemes} \\ b_{-1}\neq 0 & \text{implicit schemes} \end{cases}$
- AB/AM methods: $a_0=1, a_j=0$ for $j\geq 1$

§ General multi-step methods:

$$y_{n+1} \approx a_0 y_n + a_1 y_{n-1} + \cdots + a_{m-1} y_{n-m+1}$$
$$+ dt\left(b_{-1}f_{n+1} + b_0 f_n + b_1 f_{n-1} + \cdots + b_{m-1} f_{n-m+1}\right)$$
$$= y_n + dt\, f_n' + \frac{1}{2}dt^2 f_n'' + \frac{1}{6}dt^3 f_n''' + \cdots \quad \text{(expected)}$$

- degrees of freedom (# of adjustable variables): $\begin{cases} 2m & \text{explicit schemes} \\ 2m+1 & \text{implicit schemes} \end{cases}$
- maximum order of accuracy attainable: $\begin{cases} 2m-1 & \text{explicit schemes} \\ 2m & \text{implicit schemes} \end{cases}$

§ Systems of differential equations

$$\begin{cases} \dfrac{du_1}{dt} = f_1(t,u_1,u_2,\cdots,u_N) \\ \dfrac{du_2}{dt} = f_2(t,u_1,u_2,\cdots,u_N) \\ \quad\vdots \\ \dfrac{du_N}{dt} = f_N(t,u_1,u_2,\cdots,u_N) \end{cases} \qquad \frac{d}{dt}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} f_1(t,u_1,u_2,\cdots,u_N) \\ f_2(t,u_1,u_2,\cdots,u_N) \\ \vdots \\ f_N(t,u_1,u_2,\cdots,u_N) \end{pmatrix}$$

I.C.s: $u_1(0)=\alpha_1,\ u_2(0)=\alpha_2,\cdots,\ u_N(0)=\alpha_N$

- Let $U \equiv (u_1,u_2,\cdots,u_N)^T$ and $F \equiv (f_1,f_2,\cdots,f_N)^T$, Then

$$\frac{dU}{dt} = F(t,U)$$
$$U(0) = (\alpha_1,\alpha_2,\cdots,\alpha_N)^T$$

e.g. Adams-Moulton two-step method (3rd order)

$$y_{n+1} \approx y_n + \frac{dt}{12}(5f_{n+1} + 8f_n - f_{n-1})$$

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}_{n+1} \approx \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}_n + \frac{dt}{12}\left\{5\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}_{n+1} + 8\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}_n - \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}_{n-1}\right\}$$

$$\begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_N^{n+1} \end{pmatrix}_{n+1} \approx \begin{pmatrix} u_1^{n} \\ u_2^{n} \\ \vdots \\ u_N^{n} \end{pmatrix} + \frac{dt}{12}\left\{5\begin{pmatrix} f_1(t_{n+1},u_1^{n+1},u_2^{n+1},\cdots,u_N^{n+1}) \\ f_2(t_{n+1},u_1^{n+1},u_2^{n+1},\cdots,u_N^{n+1}) \\ \vdots \\ f_N(t_{n+1},u_1^{n+1},u_2^{n+1},\cdots,u_N^{n+1}) \end{pmatrix}\begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{matrix}_{n+1}\right.$$
$$\left. +8\begin{pmatrix} f_1(t_n,u_1^{n},u_2^{n},\cdots,u_N^{n}) \\ f_2(t_n,u_1^{n},u_2^{n},\cdots,u_N^{n}) \\ \vdots \\ f_N(t_n,u_1^{n},u_2^{n},\cdots,u_N^{n}) \end{pmatrix}\begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{matrix}_{n} -\begin{pmatrix} f_1(t_{n-1},u_1^{n-1},u_2^{n-1},\cdots,u_N^{n-1}) \\ f_2(t_{n-1},u_1^{n-1},u_2^{n-1},\cdots,u_N^{n-1}) \\ \vdots \\ f_N(t_{n-1},u_1^{n-1},u_2^{n-1},\cdots,u_N^{n-1}) \end{pmatrix}\begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{matrix}_{n-1}\right\}$$

§ Higher order equations $\dfrac{d^m y}{dt^m} = f\left(t, y, \dfrac{dy}{dt}, \dfrac{d^2y}{dt^2}\cdots, \dfrac{d^{m-1}y}{dt^{m-1}}\right)$

I.C.s $y(0)=\alpha_1, \dfrac{dy}{dt}(0)=\alpha_2, \dfrac{d^2y}{dt^2}(0)=\alpha_3,\cdots, \dfrac{d^{m-1}y}{dt^{m-1}}(0)=\alpha_m$

- Let

$$\begin{cases} u_1(t) \equiv y(t) \\ u_2(t) \equiv \dfrac{du_1}{dt} = \dfrac{dy}{dt} \\ u_3(t) \equiv \dfrac{du_2}{dt} = \dfrac{d^2y}{dt^2} \\ \quad\vdots \\ u_m(t) \equiv \dfrac{du_{m-1}}{dt} = \dfrac{d^{m-1}y}{dt^m} \end{cases} \Rightarrow \begin{cases} \dfrac{du_1}{dt} = u_2(t) \\ \dfrac{du_2}{dt} = u_3(t) \\ \quad\vdots \\ \dfrac{du_{m-1}}{dt} = u_m(t) \\ \dfrac{du_m}{dt} = f(t,u_1,u_2,\cdots,u_m) \end{cases}$$

I.C.s: $u_1(0)=\alpha_1,\ u_2(0)=\alpha_2,\cdots,\ u_m(0)=\alpha_m$

Consistency: difference equation → differential equation

as $dt \to 0$? (truncation errors → 0 as $dt \to 0$)

Stability: computed solution to the difference equation
→ exact solution to the difference equation?
(round-off errors under control)

Convergence: computed solution to the difference equation
→ exact solution to the differential equation
as $dt \to 0$?

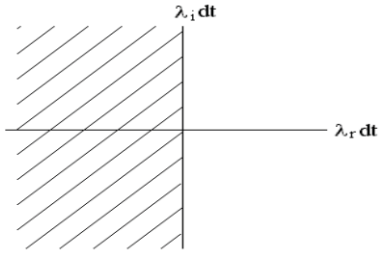
§ **test problem**

$$y'(t) = f(t,y) = \lambda y(t)$$
$$y(0) = y_0$$
$$\lambda = \lambda_r + i\lambda_i \in C$$

the exact solution is:

$$y(t) = y_0 \exp(\lambda_r t)(\cos(\lambda_i t) + i\sin(\lambda_i t))$$

where $|y(t)|$ is bounded for all $t \geq 0$ as long as $\lambda_r \leq 0$.



§ Schemes stability $y'(t) = f(t,y) = \lambda y(t)$

(1) forward Euler method:

$$y_{n+1} = y_n + dt\, f_n = y_n + dt\, \lambda y_n = (1+\lambda dt)\, y_n \quad \text{or}$$

$$f\ell(y_{n+1}) = (1+\lambda dt)\, f\ell(y_n)$$

Suppose $f\ell(y_n) = y_n + e_n$, where e_n is the round-off error. Then

$$y_{n+1} + e_{n+1} = (1+\lambda dt)(y_n + e_n) \quad \text{Thus} \quad e_{n+1} = (1+\lambda dt)\, e_n$$

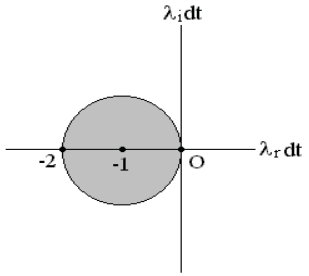
$$\Rightarrow\ e_n = (1+\lambda dt)^n\, e_0 \to \infty \text{ if } |1+\lambda dt| > 1.$$

Alternatively, the solution of the difference equation is

$$y_n = (1+\lambda dt)^n\, y_0 \to \infty \text{ if } |1+\lambda dt| > 1 \text{ even if } \lambda_r \leq 0.$$

(numerical instability instead of physical instability)

Numerical Stability



(1) forward Euler method

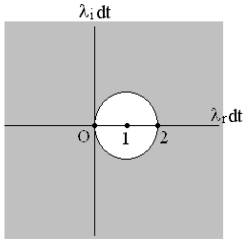
stability criterion: $|1+\lambda dt| < 1$

~ a limitation on the magnitude of dt
besides the consideration of accuracy

(2) backward Euler method:

$$y_{n+1} = y_n + dt\, f_{n+1} = y_n + dt\,\lambda y_{n+1}$$

$$y_{n+1} = (1-\lambda dt)^{-1} y_n \Rightarrow y_n = (1-\lambda dt)^{-n} y_0$$



$\Rightarrow$ stable if $|1-\lambda dt| \geq 1$.

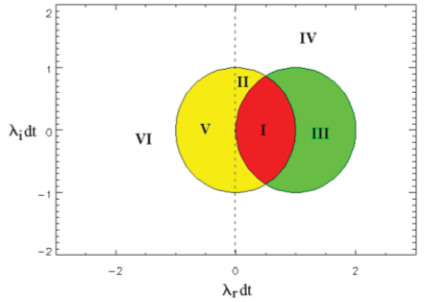
Inconsistent with the exact solution when $\lambda_r dt > 0$ and $|1-\lambda dt| \geq 1$!

because the Taylor's secies of $\frac{1}{1-z}$ has a convergence radius of 1.

§ backward Euler method

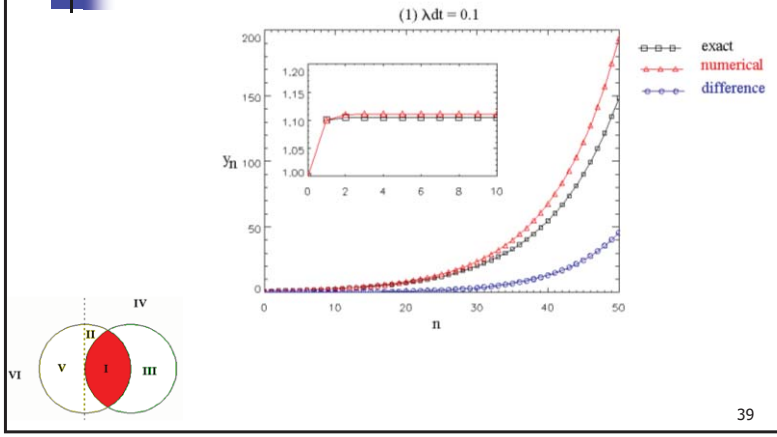
$$e^z = 1 + z + \frac{1}{2!}z^2 + \frac{1}{3!}z^3 + \cdots \quad \text{for } |z| < \infty \qquad (1)$$

$$\frac{1}{1-z} = 1 + z + z^2 + \cdots \quad \text{for } |z| < 1 \qquad (2)$$

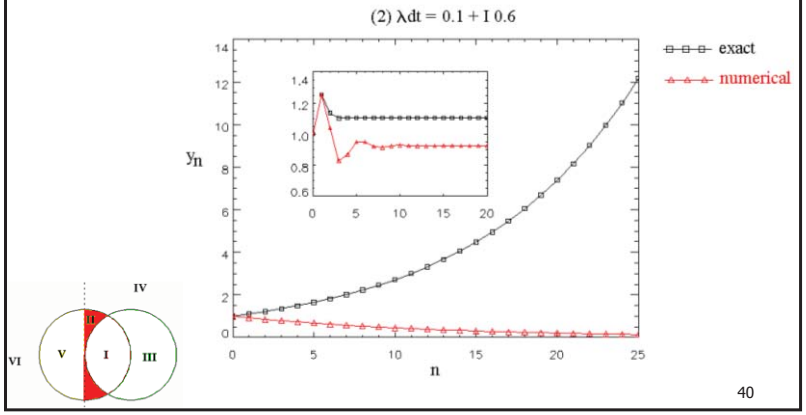


Regions I, II, and V: truncation errors under control.
Regions I, and III: rounding errors out of control.

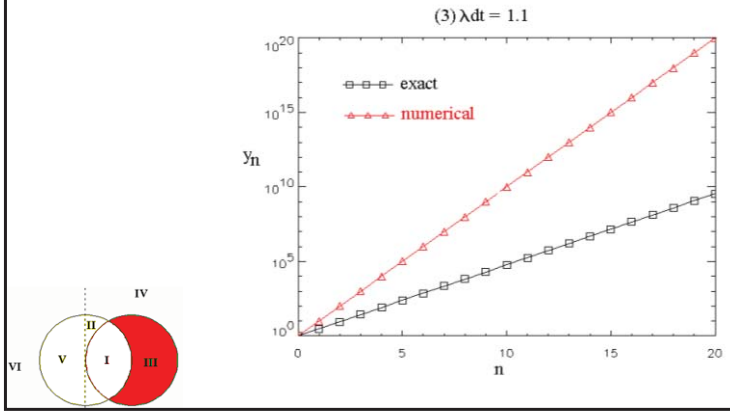
♥ Region I: Both series converge to a value > 1
Rounding error diverges.



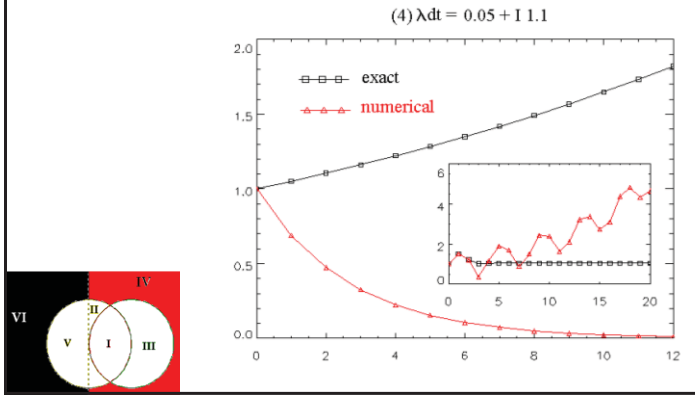
♥ Region II: Exact series converge to a value > 1
Numerical series converge to a value < 1.
Rounding error converges.



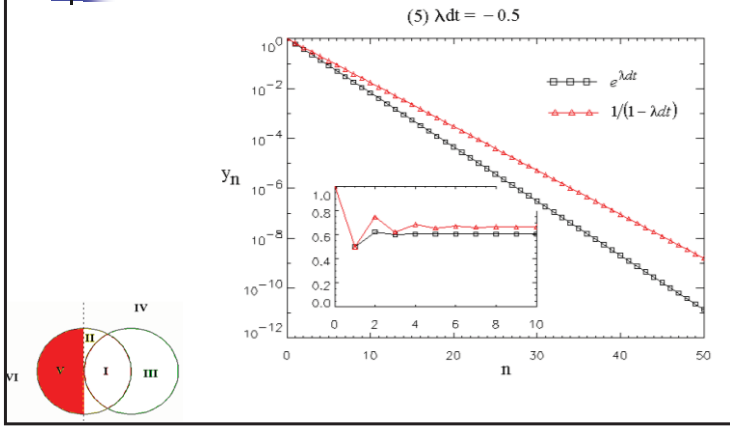
♥ Region III: Exact series converges to a value > 1.
Numerical series does not converge.
Rounding error diverges.



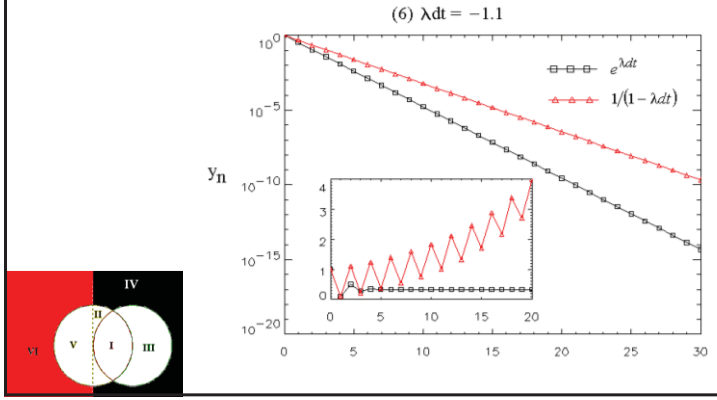
♥ Region IV: Exact series converges to a value > 1.
Numerical series does not converge.
Rounding error converges.



♥ Region V: Both series converge to a value < 1.
Rounding error converges.



♥ Region VI: Exact series converges to a value < 1.
Numerical series does not converge.
Rounding error converges.



(3) Trapezoidal (Crank-Nilcoson) method:

$$y_{n+1} = y_n + \frac{1}{2}dt\left(f_n + f_{n+1}\right) = y_n + \frac{1}{2}\lambda dt\left(y_n + y_{n+1}\right)$$

$$y_{n+1} = \left(\frac{1+\frac{\lambda dt}{2}}{1-\frac{\lambda dt}{2}}\right) y_n \qquad y_n = \left(\frac{1+\frac{\lambda dt}{2}}{1-\frac{\lambda dt}{2}}\right)^n y_0$$

stable if $\left|1+\frac{\lambda dt}{2}\right| \le \left|1-\frac{\lambda dt}{2}\right| \Leftrightarrow \lambda_r\, dt \le 0$

(4) leapfrog method:

$$y_{n+1} = y_{n-1} + 2dt\, f_n = y_{n-1} + 2\lambda dt\, y_n$$

Suppose $y_n = \rho^n$. Substitute into the difference equation to oftain

$\rho^2 - 2\lambda dt\,\rho - 1 = 0 \quad \Rightarrow \quad \rho_{1,2} = \lambda dt \pm \sqrt{\lambda^2 dt^2 + 1}$

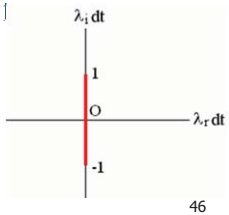
$y_n = C_1\rho_1^n + C_2\rho_2^n \quad \Rightarrow$ stable if $\left|\rho_{1,2}\right| \le 1$

However, since $\rho_1 \cdot \rho_2 = -1 \quad \Rightarrow$ stable if $\left|\rho_{1,2}\right| = 1$

Write $\rho_1 = \exp(i\theta)$. Then $\rho_2 = -\exp(-i\theta)$.

Moreover, $\rho_1 + \rho_2 = 2\lambda dt = 2i\sin\theta$

$\Rightarrow$ stable if $\lambda_r dt = 0$ *and* $\left|\lambda_i dt\right| \le 1$



(5) Runge-Kutta methods:

2nd method: $\left|1+\lambda dt+\frac{1}{2}(\lambda dt)^2\right| \le 1$

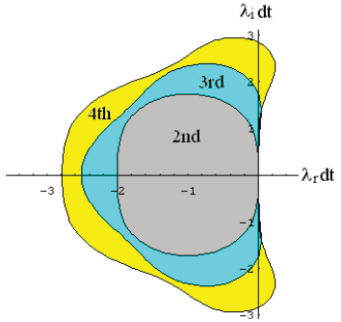
3rd method: $\left|1+\lambda dt+\frac{1}{2}(\lambda dt)^2+\frac{1}{6}(\lambda dt)^3\right| \le 1$

4nd method: $\left|1+\lambda dt+\frac{1}{2}(\lambda dt)^2+\frac{1}{6}(\lambda dt)^3+\frac{1}{24}(\lambda dt)^4\right| \le 1$

2nd: -2

3rd: -2.51275; 1.73205

4th: -2.78529; 2.82843



- 2nd order Runge-Kutta methods:

$$k_1 = dt \cdot f_n = dt \cdot \lambda\, y_n$$

$$k_2 = dt \cdot f\left(t_n + \beta\, dt, y_n + \alpha\, k_1\right) = dt \cdot \lambda\left(y_n + \alpha\, k_1\right) = \lambda dt\left(y_n + \alpha\lambda dt\, y_n\right)$$

$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2 = y_n + \gamma_1\lambda dt\, y_n + \gamma_2\lambda dt\left(y_n + \alpha\lambda dt\, y_n\right)$$

$$= \left(1 + (\gamma_1 + \gamma_2)\lambda dt + \alpha\gamma_2(\lambda dt)^2\right) y_n$$

$$= \left\{1 + \lambda dt + \frac{1}{2}(\lambda dt)^2\right\} y_n$$

$(\gamma_1 + \gamma_2) = 1$

$\alpha\gamma_2 = \beta\gamma_2 = \frac{1}{2}$

stable if $\left|1+\lambda dt+\frac{1}{2}(\lambda dt)^2\right| < 1$

• In general:

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + \cdots + a_{m-1} y_{n-m+1} + dt\left(b_{-1} f_{n+1} + b_0 f_n + b_1 f_{n-1} + \cdots + b_{m-1} f_{n-m+1}\right)$$

Define $\begin{cases} p(z) = z^m - a_0 z^{m-1} - a_1 z^{m-2} - \cdots - a_{m-1} \\ q(z) = b_{-1} z^m + b_0 z^{m-1} + \cdots + b_{m-1} \end{cases}$

- The multi-step method is said to be **stable** if all roofs of $p(z)$ lie in the disk $|z| \le 1$ and if each root of modulus 1 is simple.
- The method is said to be **consistent** if $p(1) = 0$ and $p'(1) = q(1)$

Theorem For the multi-step method to be convergent, it is necessary and sufficient that it be stable and consistent.

Conclusion:

implicit schemes: more stable – allow a larger time increment *dt*

troublesome

explicit schemes: less stable – need a small *dt*

easy to implement.

§ system of equations:

test problem: $\frac{dU}{dt} = AU$

where A is a constant complex matrix.

suppose $\{\lambda_k\}_{k=1}^N$ are the complex eigenvalues of the matrix A

~ stable if all $\lambda dt \in$ stable region

§ Boundary-Value Problems for ODEs

$$y'' = g(x, y, y'),\ a \le x \le b$$

$$y(a) = \alpha,\ y(b) = \beta$$

~ shooting method and finite difference method

§ **shooting method** — IV ODEs $y'' = g(x, y, y'),\ a \le x \le b$

$y(a) = \alpha,\ y(b) = \beta$

$u_1(x) \equiv y(x)$

$y'(a) = t$

$\frac{du_1}{dx} = u_2$

$\frac{du_2}{dx} = g(x, u_1, u_2)$

$u_1(a) = \alpha,\ u_2(a) = t$

STEP1: guess $y'(a) = t$ and integrate the ODEs until $x = b$:

STEP2: check if the relative error $\frac{|y(b) - \beta|}{\beta} \le \varepsilon$? if not, re-guess $y'(a)$.

• How to re-guess? Notice $y(b)$ = fn. of $t = y_b(t)$

Define $f(t) \equiv y_b(t) - \beta$.

Thus we are looking for a value of t such that $f(t)$=0

$\Rightarrow$ root-searching mehods. e.g., Secant method:

STEP1: take two initial guesses $y'(a)$=t_0 and $y'(a)$=t_1.

STEP2: if $\frac{|y_b(t_i) - \beta|}{\beta} = \frac{|u_1(b, t_i) - \beta|}{\beta} > \varepsilon$, then

$$t_{i+1} = t_i - \frac{f(t_i)(t_i - t_{i-1})}{f(t_i) - f(t_{i-1})} = t_i - \frac{(y_b(t_i) - \beta)(t_i - t_{i-1})}{y_b(t_i) - y_b(t_{i-1})} = t_i - \frac{(u_1(b, t_i) - \beta)(t_i - t_{i-1})}{u_1(b, t_i) - u_1(b, t_{i-1})}$$

• A special case: the ODE is linear

$y'' = g(x, y, y') = p(x)y' + q(x)y + r(x)$

$y(a) = \alpha \qquad y(b) = \beta$

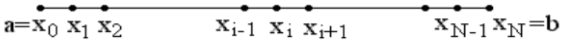
(1) $\begin{cases} y'' = p(x)y' + q(x)y + r(x) \\ y(a) = \alpha \\ y'(a) = 0 \end{cases} \Rightarrow \quad y_1(x)$

(2) $\begin{cases} y'' = p(x)y' + q(x)y \\ y(a) = 0 \\ y'(a) = 1 \end{cases} \Rightarrow \quad y_2(x)$

Then $y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} \cdot y_2(x)$

§ **Finite-difference methods** $y'' = g(x, y, y'),\ a \le x \le b$

$y(a) = \alpha, y(b) = \beta$



$y''(x_i) = g(x_i, y(x_i), y'(x_i))$

e.g. central difference: $\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \approx g\left(x_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right)$

truncation error ~ $O(h^2)$

for i = 1,2,…, N–1

BC's: $y_0 = \alpha, y_N = \beta$

~ implicit equations for $y_i, i = 1, 2, 3, \cdots, N-1$

~ root-searching for multi-variable system

linear equation: $g(x,y,y') = p(x)y' + q(x)y + r(x)$

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \approx p(x_i)\cdot\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i + r(x_i)$$

$$\left(\frac{1}{h^2} + \frac{p_i}{2h}\right)y_{i-1} - \left(\frac{2}{h^2} + q_i\right)y_i + \left(\frac{1}{h^2} - \frac{p_i}{2h}\right)y_{i+1} = r_i$$

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{h^2}+\frac{p_1}{2h} & -\left(\frac{2}{h^2}+q_1\right) & \frac{1}{h^2}-\frac{p_1}{2h} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{1}{h^2}+\frac{p_2}{2h} & -\left(\frac{2}{h^2}+q_2\right) & \frac{1}{h^2}-\frac{p_2}{2h} & 0 & \cdots & 0 \\ & & \vdots & & & & \\ 0 & \cdots & 0 & \frac{1}{h^2}+\frac{p_{N-1}}{2h} & -\left(\frac{2}{h^2}+q_{N-1}\right) & \frac{1}{h^2}-\frac{p_{N-1}}{2h} & \\ 0 & \cdots & & & 0 & 1 & \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} \alpha \\ r_1 \\ r_2 \\ \vdots \\ r_{N-1} \\ \beta \end{pmatrix}$$