Basics of Initial Value Problems for 1st-Order ODEs

Initial Value Problem: Find $y = y(t)$ satisfying

$$\boxed{\begin{array}{ll} \text{(ODE)} & y' = f(t, y), \quad [t_0, T] \\ \text{(IC)} & y(t_0) = y_0 \end{array}}$$

- System: find $y_1, y_2, \cdots, y_n$ such that

$$\text{(ODEs)} \quad \left\{ \begin{array}{l} y_1' = f_1(t, y_1, y_2, \cdots, y_n) \\ \vdots \\ y_n' = f_n(t, y_1, y_2, \cdots, y_n) \end{array} \right\} \quad [t_0, T]$$

$$\text{(ICs)} \quad y_1(t_0) = y_{1,0}, \ \cdots, \ y_n(t_0) = y_{n,0}$$

$$\Longrightarrow \quad \boxed{\vec{y}\,' = \vec{F}(t, \vec{y}) \text{ on } [t_0, T], \quad \vec{y}(0) = \vec{y}_0}$$

- Example 1: when $f = f(t)$

Error in video fixed

$$\left\{ \begin{array}{l} y' = f(t), \quad [t_0, T] \\ y(t_0) = y_0 \end{array} \right.$$

$$\xrightarrow{\text{FTC}} \quad y(t) = y_0 + \int_{t_0}^{t} f(s)\, ds$$

$$t_0 \le t \le T.$$

- Example 2: when $f = f(y)$

$$\left\{ \begin{array}{l} y' = f(y) \\ y(t_0) = y_0 \end{array} \right. \qquad \text{"autonomous"}$$

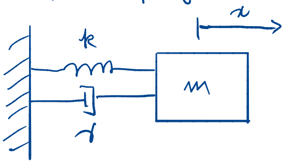
In addition, if f is linear, i.e.

$$y' = Ay, \quad y(t_0) = y_0 \qquad \text{"expm"}$$

$$\hookrightarrow \quad y(t) = y_0 e^{At} \quad \text{(exact)}$$

Is studying 1^{st}-order ODEs enough?

Example (spring-mass-damper)



(Newton) $mx'' = -kx - \gamma x'$

$\hookrightarrow \boxed{x''} + \underbrace{\frac{\gamma}{m}}_{= 2\zeta\omega_0} x' + \underbrace{\frac{k}{m}}_{= \omega_0^2} x = 0$

where $\omega_0 = \sqrt{\frac{k}{m}}$ natural freq

$\zeta = \frac{\gamma}{2\sqrt{km}}$ damping ratio

To convert it into a 1^{st}-order system:

$$\left.\begin{aligned} y_1 &= x \\ y_2 &= x' \end{aligned}\right\} \xrightarrow{\frac{d}{dt}} \left\{\begin{aligned} y_1' &= x' = y_2 \\ y_2' &= x'' \\ &= -\omega_0^2 x - 2\zeta\omega_0 x' \\ &= -\omega_0^2 y_1 - 2\zeta\omega_0 y_2 \end{aligned}\right.$$

$$\therefore \frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$\vec{y}\,' = A\,\vec{y}$$

Higher-Order Derivatives

In general: n^{th}-order ODE

$$x^{(n)} = f(t, x, x', \cdots, x^{(n-1)})$$

$$\left.\begin{aligned} y_1 &= x \\ y_2 &= x' \\ &\vdots \\ y_n &= x^{(n-1)} \end{aligned}\right\} \xrightarrow{\frac{d}{dt}} \left\{\begin{aligned} y_1' &= x' &&= y_2 \\ y_2' &= x'' &&= y_3 \\ &\vdots \\ y_n' &= x^{(n)} &&= f(t, y_1, y_2, \cdots, y_n) \end{aligned}\right.$$

$$\downarrow$$

$$\boxed{\vec{y}\,' = \vec{F}(t, \vec{y})}$$

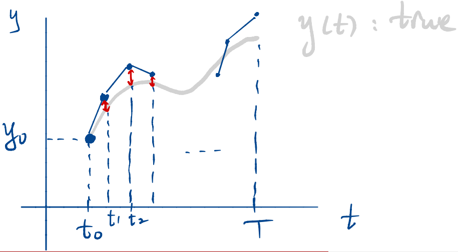
Notation: scalar notation

(y instead of $\vec{y}$)

Original Problem: Find $y = y(t)$

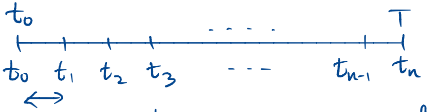
s.t.

$$\boxed{\begin{aligned} y' &= f(t, y), \quad [t_0, T] \\ y(t_0) &= y_0 \end{aligned}}$$



Numerically

- Divide $[t_0, T]$ into n equispaced subint.



$$h = \frac{T - t_0}{n}, \qquad t_j = t_0 + jh \qquad 0 \le j \le n.$$

- Start w/ initial data y_0 and iterate to obtain

$$y_0 \to y_1 \to y_2 \to \cdots \to y_n$$

where $\quad y_j \approx y(t_j)$

Euler's Method

- Simplest method
- "one-step" method.

Idea: Replace $y'(t_j)$ by FD.

$$y'(t_j) = f(t_j, y_j)$$
$$\approx$$
$$\frac{y_{j+1} - y_j}{h}$$

$$\Rightarrow \boxed{y_{j+1} = y_j + f(t_j, y_j)\, h}$$

(Forward) Euler's Method

"not very stable"

cf) Backward Euler

"more stable than Forw. Euler"

$$y'(t_{j+1}) = f(t_{j+1}, y_{j+1})$$
$$\approx$$
$$\frac{y_{j+1} - y_j}{h}$$

Implicit

$$\Rightarrow \boxed{y_{j+1} = y_j + f(t_{j+1}, y_{j+1})\, h}$$

"Backward Euler"

In case $\vec{f}(t, \vec{y}) = A\vec{y}$:

$$\vec{y}_{j+1} = \vec{y}_j + hA\vec{y}_{j+1}$$

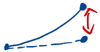
$$\vec{y}_{j+1} - hA\vec{y}_{j+1} = \vec{y}_j$$

$$(I - hA)\vec{y}_{j+1} = \vec{y}_j \Rightarrow \vec{y}_{j+1} = (I - hA)^{-1}\vec{y}_j$$

Error of Euler's Method (Forward Euler)

"Accuracy" → Taylor series expansion.

- Local error : 2^{nd} - order



$y(t_{j+1}) \leftarrow$ exact

$y_{j+1} \leftarrow$ F.E. (Num)

$t_j \quad t_{j+1}$

$$\underbrace{y(t_{j+1})}_{\substack{\| \\ t_j + h}} - y_{j+1} = \Big(\underbrace{y(t_j)}_{y_j} + \underbrace{y'(t_j)}_{f(t_j, y_j)} h + \frac{y''(t_j)}{2} h^2 + \cdots \Big) - y_{j+1}$$

$$= \underbrace{\left(y_j + f(t_j, y_j) h \right)}_{\text{Forw. Euler}} - \cancel{y_{j+1}} + O(h^2)$$

- Global error : 1^{st} - order

"at the end of trajectory"

accum

$$n \; O(h^2)$$

$$= \frac{T - t_0}{\cancel{h}} \, O(h^{\cancel{2}})$$

$$= O(h) \quad ✓$$