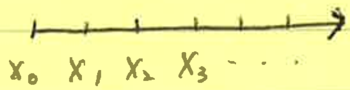
(P1)

Numerical Method for ODE.

△ Euler's method.

$$y' = f(x, y).$$



$$x_1 = x_0 + h.$$
$$x_2 = x_0 + 2h.$$
$$x_n = x_0 + nh.$$

where h is the step size.

Talyor expansion of $y(x+h)$ at x.

$$y(x+h) = y(x) + h y'(x) + \frac{h^2}{2} y''(x) + \cdots$$

For small h, $h^2, h^3 \ll h$.

$$\Rightarrow \quad y(x+h) \approx y(x) + h y'(x).$$

The numerical scheme (Euler method).

$$y_{n+1} = y_n + h f(x_n, y_n).$$

(Forward Euler method.
Euler-Cauchy method)

P2

Error of the Euler method.

~~Talyer~~ Taylor series. $\exists\, \xi \in [x, x+h)$

$$y(x+h) = y(x) + h y'(x) + \underbrace{\tfrac{1}{2} h^2 y''(\xi)}_{\text{truncation error}}.$$

global error $O(h)$.

△ Improved Euler method.

$$y^*_{n+1} = y_n + h f(x_n, y_n).$$

$$y_{n+1} = y_n + \tfrac{1}{2} h \left(f(x_n, y_n) + f(x_{n+1}, y^*_{n+1}) \right)$$

truncation error $O(h^3)$.

global error $O(h^2)$.

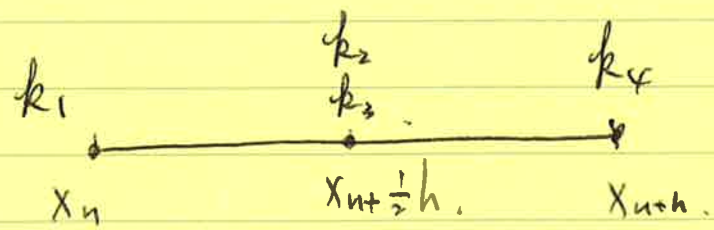
combine the ~~@~~ Taylor expansions ~~of~~ at both points.

△ "Further improved Euler method"

Runge - Kutta Method.

Algorithm.

$$k_1 = h f(x_n, y_n).$$
$$k_2 = h f(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}k_1)$$
$$k_3 = h f(x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}k_2).$$
$$k_4 = h f(x_n + h,\ y_n + k_3).$$
$$x_{n+1} = x_n + h.$$
$$y_{n+1} = y_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$



Truncation error $O(h^5)$
global error $O(h^4)$.

You can design your own higher order numerical ~~sch~~ method!

(P4)

△ Backward Euler Method.

$$y_{n+1} = y_n + h\, f(x_{n+1}, y_{n+1}).$$

More stable, larger ~~converge~~ range of convergence.

△ ~~Central diff.~~
Finite difference method.

$$y' = f(x, y).$$

$$\frac{y_{n+1} - y_n}{h} = f(x_n, y_n)$$

$$\frac{y_{n+1} - y_n}{h} = f(x_{n+1}, y_{n+1}).$$

$$? \quad \frac{y_{n+1} - y_{n-1}}{2h} = f(x_n, y_n). \qquad (PDE).$$

P5

y''.

$$\boxed{\frac{\dfrac{y_{n+1} - y_n}{h} - \dfrac{y_n - y_{n-1}}{h}}{h}}$$