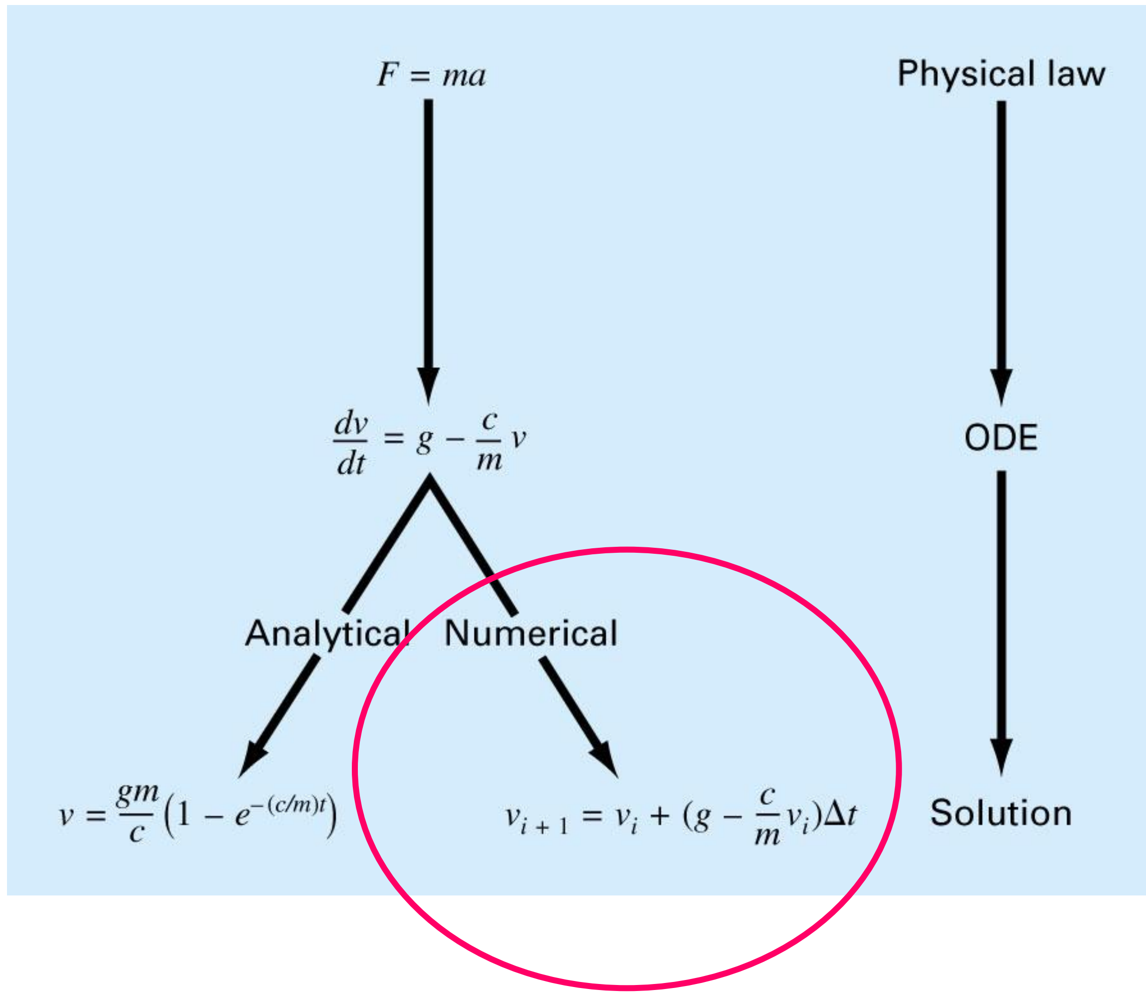
Numerical Analysis – Differential Equation

Hanyang University

Jong-Il Park

Differential Equation



Solving Differential Equation

- Differential Equation
  - Ordinary D.E.
  - Partial D.E.
  - Ordinary D.E.
    - Linear eg. $y'' + y = f(t)$
    - Nonlinear eg. $y' y'' + y = f(t)$

      Usually no closed–form solution

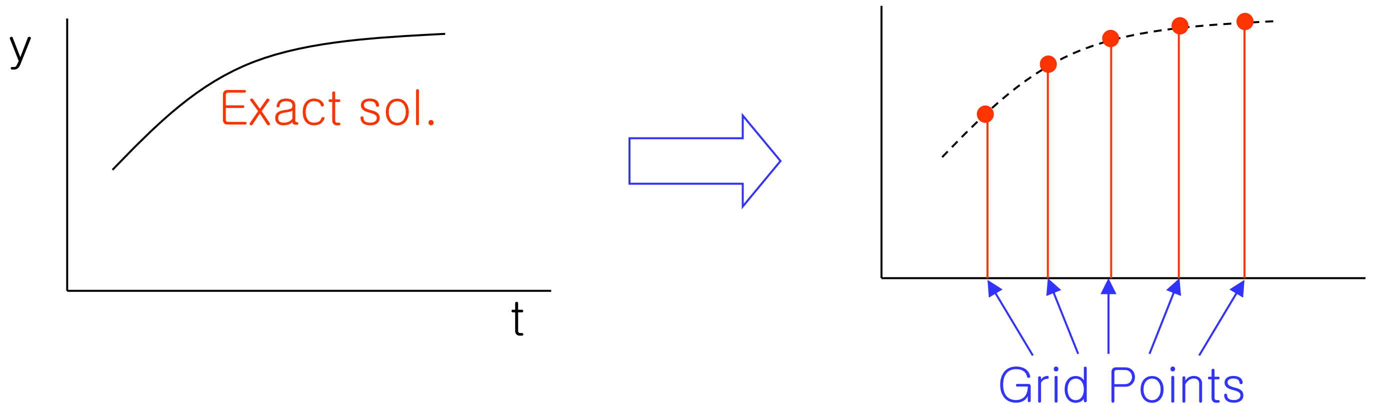
      ⇒ linearization / numerical solution
    - Initial value problem

      $$eg.\ y'' + y = 0,\ y(0) = y'(0) = 0$$
    - Boundary value problem

      $$eg.\ y'' + 4y' + 5y = 10,\ y(0) = 0,\ y(1) = 3$$

Discretization in solving D.E.

- Discretization



- Errors in Numerical Approach
  - Discretization error

$$e_D = y_e - y_d$$

  - Stability error

$$e_S = y_d - y_n$$

y_e : exact sol.
y_d : discretized sol.
y_n : numerical sol.

Errors

- Total error

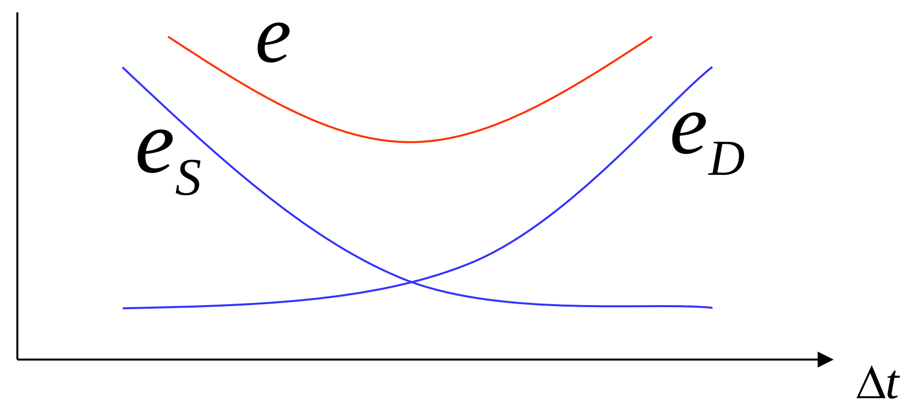
$$e = e_D + e_S$$

truncation   round-off

$e_D \longrightarrow 0$ as $\Delta t \longrightarrow 0$

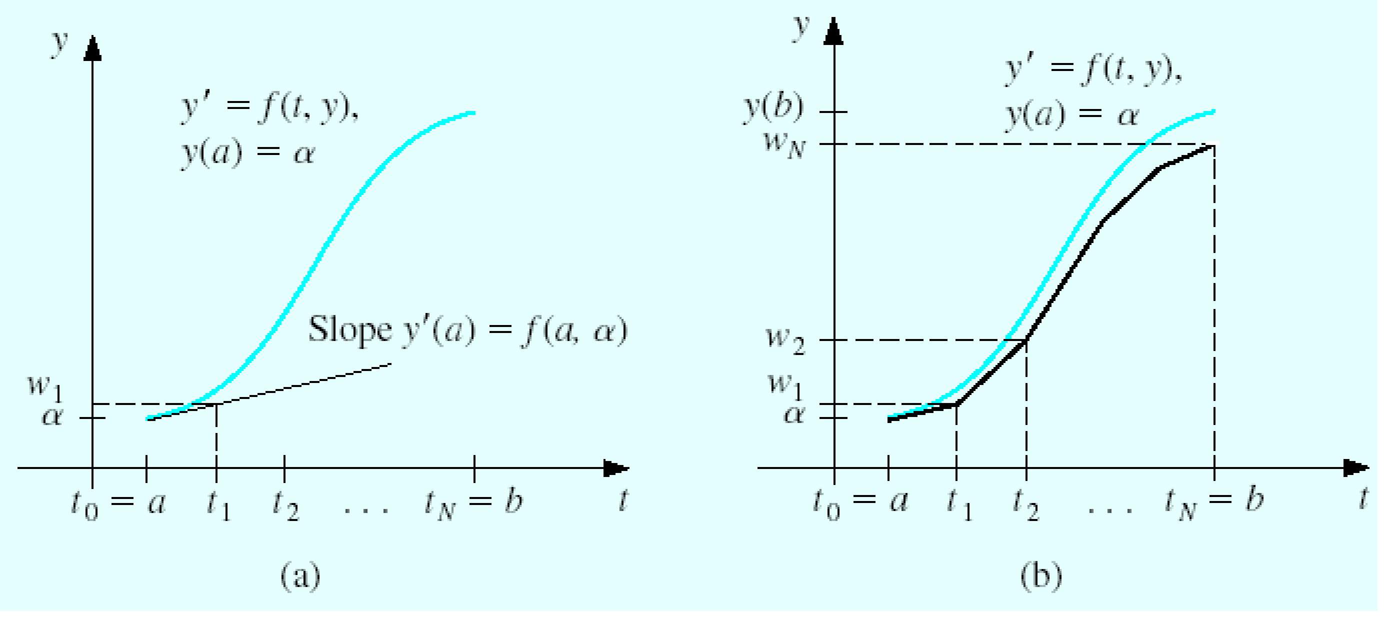
$e_S \longrightarrow$ increase as $\Delta t \longrightarrow 0$

trade-off



Local error & global error

- Local error
  - The error at the given step if it is assumed that all the previous results are all exact
- Global error
  - The true, or accumulated, error



Useful concepts(I)

- Useful concepts in discretization
  - Consistency

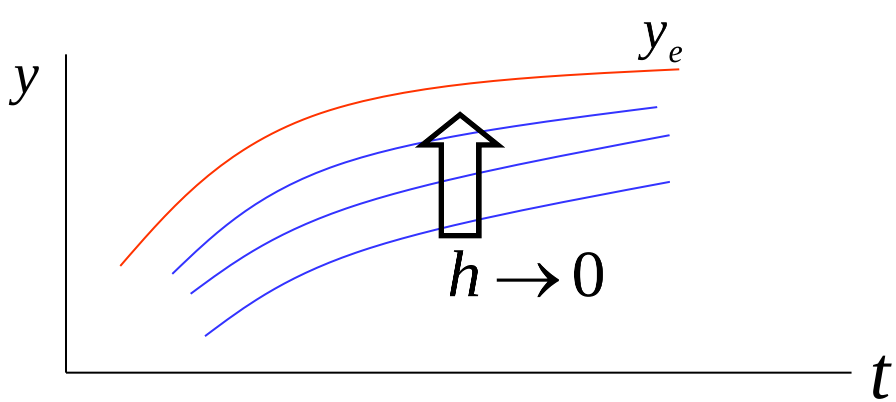
$$\Delta t(=h) \rightarrow 0 \;\Rightarrow\; e_D \rightarrow 0$$

  - Order

$$O(h^2) \Rightarrow e_D \propto h^2$$

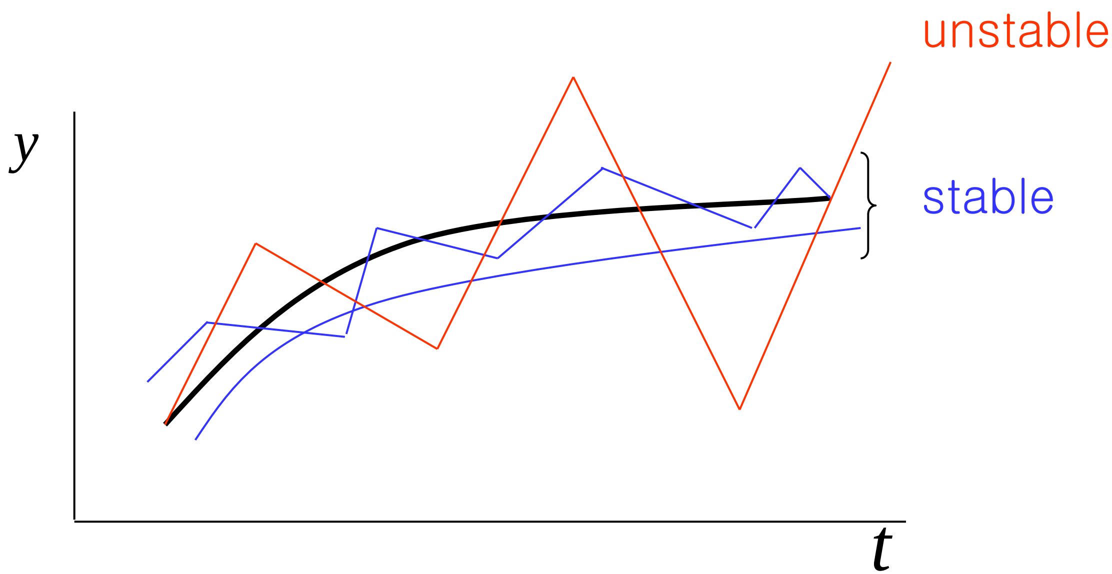
$$O(h^5) \Rightarrow e_D \propto h^5$$

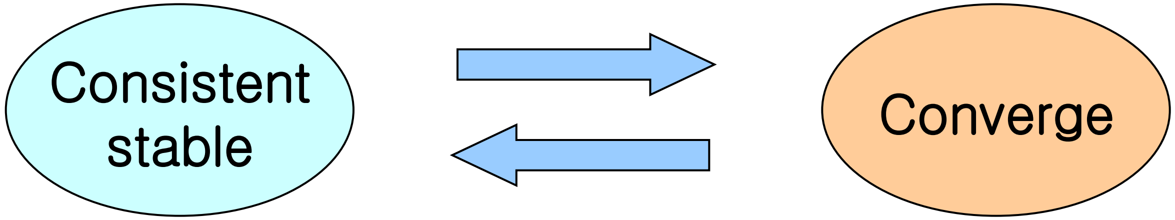
  - Convergence



Useful concepts(II)

- stability





Stability

- Stability condition

eg. $y' = -Ay, \quad y(0) = y_0$

Exact sol. $y = y_0 e^{-At}$

Euler method $y_{n=1} = y_n - hAy_n$

$$= (1 - hA) y_n$$

$$= (\underline{1 - hA})^{n+1} y_0$$

Amplification factor

For stability

$$|1 - hA| \leq 1 \quad \rightarrow \quad 0 < h \leq \frac{2}{A}$$

Implicit vs. Explicit Method

eg. $y' + y = 1.2, \quad y(0) = 0.2 \Rightarrow y' = \underline{1.2 - y}$

$= f$

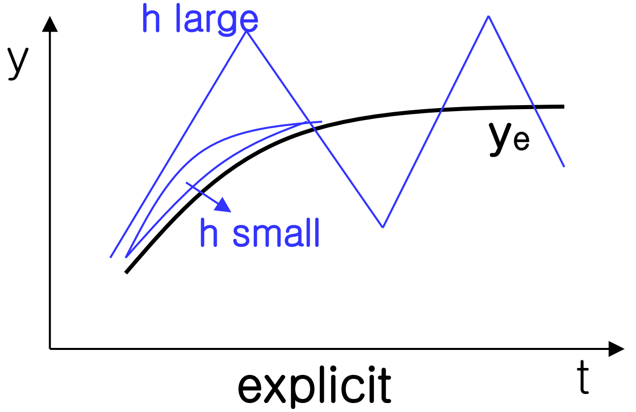
Explicit : $\underline{y_{n+1} = y_n + hf_n}$

$\therefore y_{n+1} = y_n + h(1.2 - y_n) = 1.2h - (1-h)y_n$

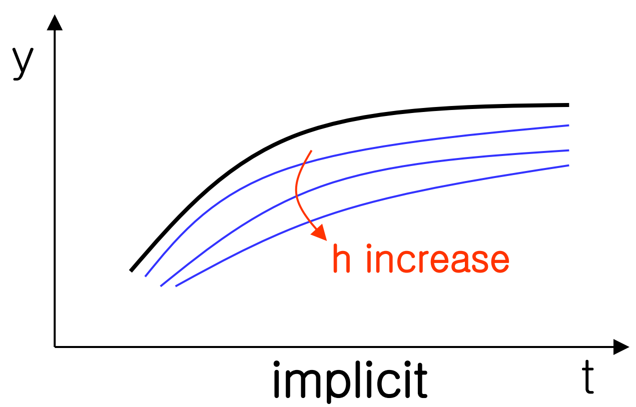
Implicit : $\underline{y_{n+1} = y_n + hf_{n+1}}$

$\therefore y_{n+1} = y_n + h(1.2 - y_{n+1})$

$$= \frac{y_n + 1.2h}{1+h}$$



"conditionally stable"



"stable"

Modification to solve D.E.

- Modified Differential Eq.

Diff. eq. —Discretization→ Modified D.E.

$$eg. \quad y' + Ay = g(t)$$

Discretization by Euler method

$$y_{n+1} = y_n + h(g_n - Ay_n)$$

<Consistency check>

$$y_{n+1} = y_n + hy'|_n + \tfrac{1}{2!}h^2 y''|_n + \cdots$$

$$\cancel{y_n} + hy'|_n + \tfrac{1}{2!}h^2 y''|_n + \cdots = \cancel{y_n} + h(g_n - Ay_n)$$

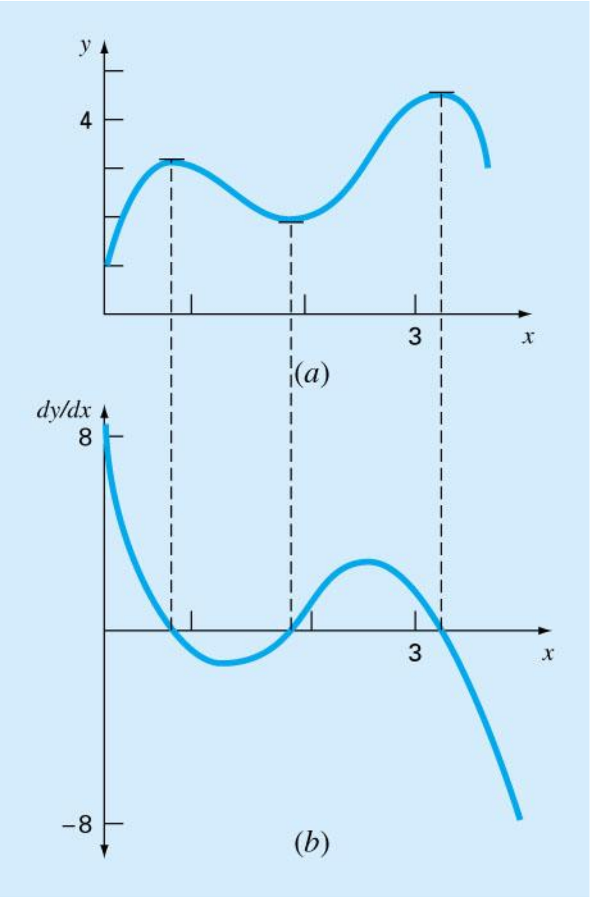
$$y'|_n + Ay_n = g_n - \tfrac{1}{2!}\underline{hy''} - \cdots$$

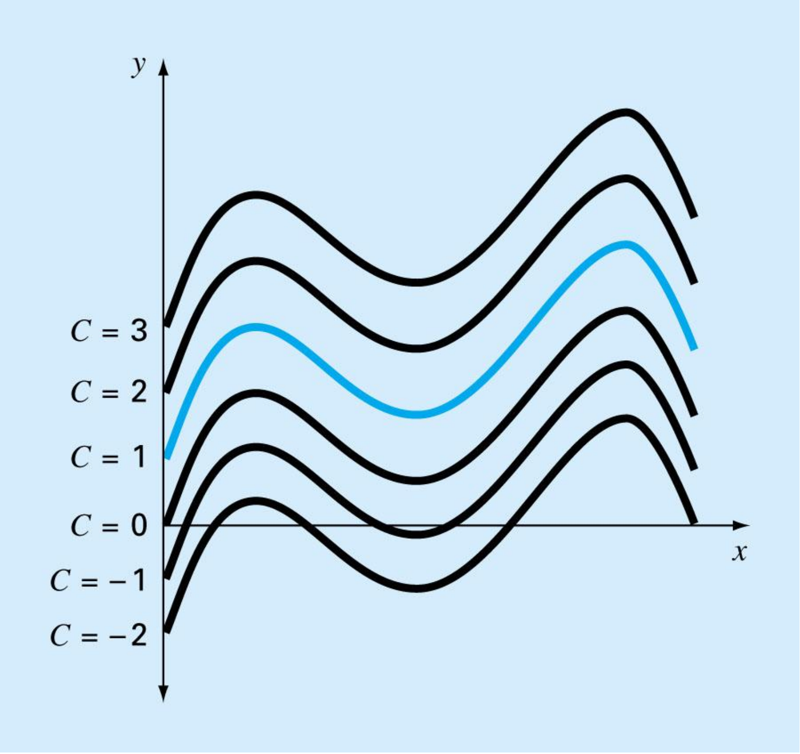
Let $h \to 0$ $\quad y'|_n + Ay_n = g_n$; consistent

<Order>

$$y'|_n + Ay_n = g_n + O(h)$$

Initial Value Problem: Concept





Initial value problem

- Initial Value Problem

$$y' = \frac{dy}{dt} = f(t, y) \;,\; y(t_0) = y_0$$

  - Simultaneous D.E.

$$\frac{dy_1}{dt} = f_1(t, y_1, y_{2,} \cdots, y_n), \quad y_1(t_0) = y_{10}$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\frac{dy_n}{dt} = f_n(t, y_1, y_{2,} \cdots, y_n), \quad y_n(t_0) = y_{n0}$$

  - High-order D.E.

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

$$y^{(t_0)} = y_0, \; y'(t_0) = y_0', \cdots, y^{(n-1)}(t_0) = y_0^{(n-1)}$$

Well-posed condition

Suppose that f and f_y, its first partial derivative with respect to y, are continuous for t in $[a, b]$ and for all y. Then the initial-value problem

$$y' = f(t, y), \quad \text{for } a \leq t \leq b, \quad \text{with } y(a) = \alpha,$$

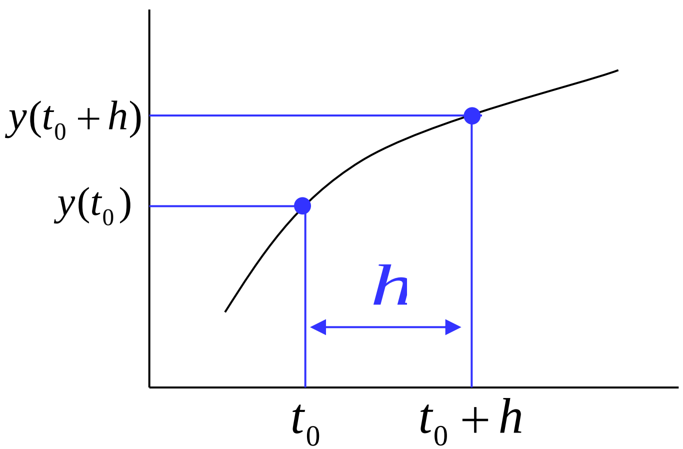
has a unique solution $y(t)$ for $a \leq t \leq b$, and the problem is well-posed.

Taylor series method(I)

- Taylor Series Method

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

$$y(t_0 + h) = y_0 + y_0' h + \tfrac{1}{2!} y_0'' h^2 + \cdots + \tfrac{1}{n!} y_0^{(n)} h^n + R_n$$



Truncation error

$$R_n = \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi),$$

$$t_0 \le \xi \le t_0 + h$$

Taylor series method(II)

- High order differentiation

$$y'' = \frac{d}{dt}[f(t, y(t))] = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = f_t + f_y f$$

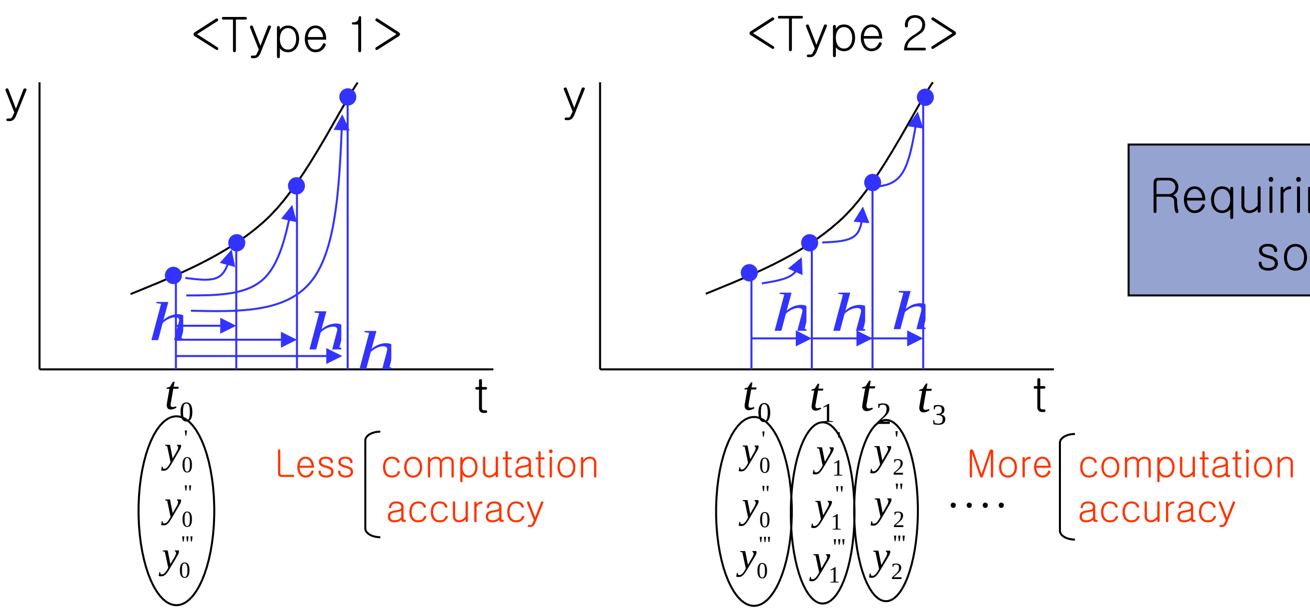
$$y''' = \frac{d}{dt}[f_t + f_y f] = f_{tt} + f_{ty} f + \frac{\partial}{\partial t}(f_y f) + \frac{\partial}{\partial y}(f_y f) f$$

$$= f_{tt} + 2 f_{ty} f + f_y f_t + f_y^2 f + f_{yy} f^2$$

$$\vdots$$

→ Complicated computation

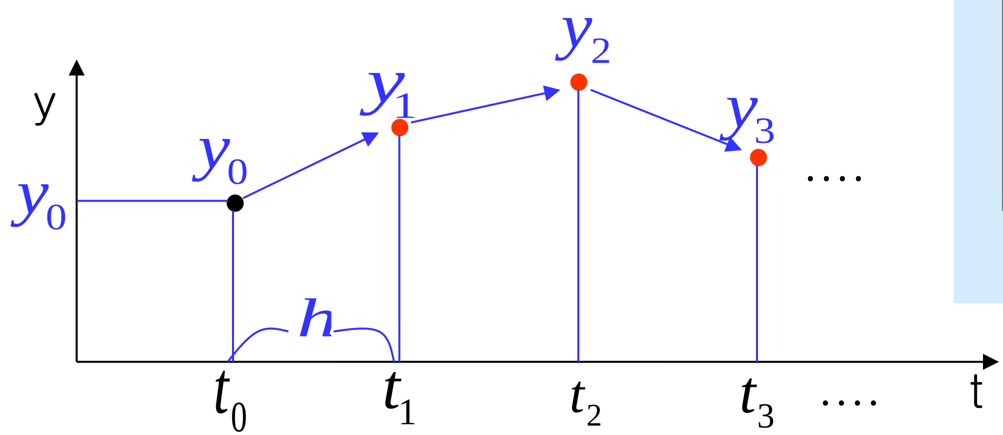
- Implementation

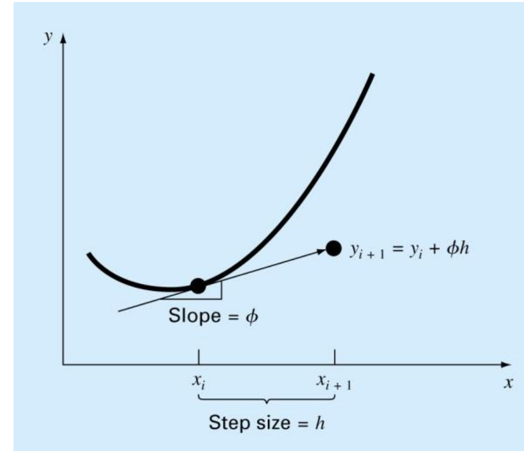


Requiring complicated source codes

Euler method(I)

- Euler Method





$$y' = f(t, y),\ y(t_0) = y_0$$

Talyor series expansion at t_o

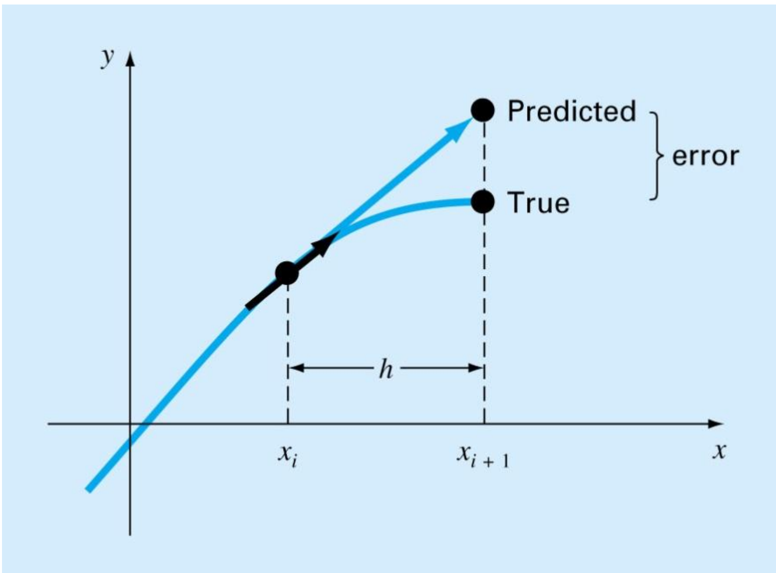
$$y_1 = y_0 + \underbrace{y_0'}h + \frac{1}{2!} y''(\xi_0) h^2, \quad t_0 \le \xi_0 \le t_1$$

$$= f(t_0, y_0)$$

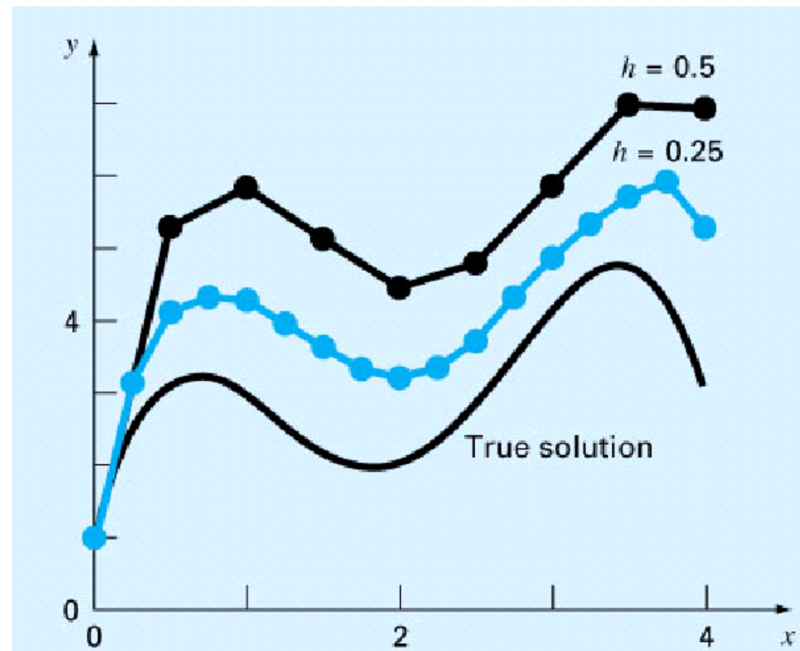
$$\equiv f_0$$

Euler method(II)

Error



Eg. $y' = -2x^3+12x^2-20x+8.5$, $y(0)=1$



Euler method(III)

Generalizing the relationship

$$y_{n+1} = y_n + f_n h + \tfrac{1}{2!} y''(\xi_n) h^2$$

$$= \underbrace{y_n + f_n h}_{\text{Euler's approx.}} + \underbrace{O(h^2)}_{\text{truncation error}}, \quad t_n \le \xi_n \le t_{n+1}$$

Error Analysis

$$y_n = y_0 + (y_1 - y_0) + (y_2 - y_1) + \cdots + (y_n - y_{n-1}) = y_0 + \sum_{i=0}^{n-1} (y_{i+1} - y_i)$$

Accumulated truncation error

$$e_t = \sum_{i=0}^{n-1} \tfrac{1}{2} y''(\xi_i) h^2$$

$$\bar{y}''(\xi) = \frac{1}{n} \sum_{i=0}^{n-1} y''(\xi_i), \quad t_0 \le \xi \le t_n$$

$$n = \frac{t_n - t_0}{h}$$

$$= \frac{t_n - t_0}{h} \frac{1}{2} \bar{y}''(\xi) h^2$$

$$= \frac{1}{2}(t_n - t_0)\, \bar{y}''(\xi) h = O(h)$$

; 1st order

Eg. Euler method

Suppose that Euler's method is used to approximate the solution to the initial-value problem

$$y' = y - t^2 + 1, \quad \text{for } 0 \le t \le 2, \quad \text{with } y(0) = 0.5,$$

assuming that $N = 10$. Then $h = 0.2$ and $t_i = 0.2i$.

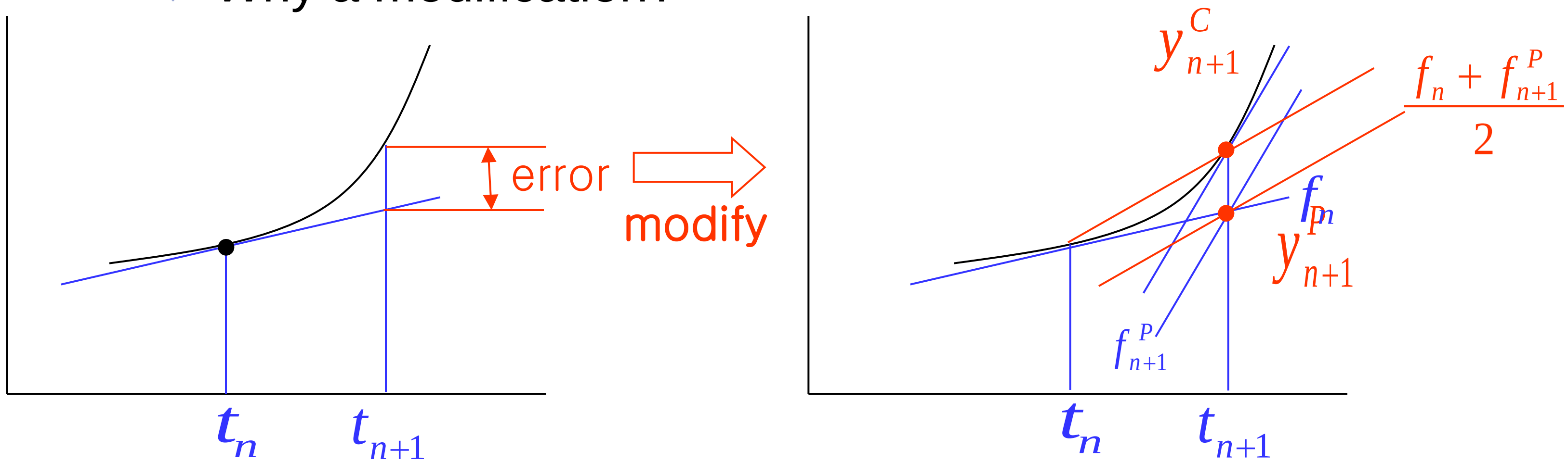
Since $f(t, y) = y - t^2 + 1$ and $w_0 = y(0) = 0.5$, we have

$$w_{i+1} = w_i + h(w_i - t_i^2 + 1) = w_i + 0.2[w_i - 0.04i^2 + 1] = 1.2w_i - 0.008i^2 + 0.2$$

t_i	$y_i = y(t_i)$	w_i	$\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0.0000000
0.2	0.8292986	0.8000000	0.0292986
0.4	1.2140877	1.1520000	0.0620877
0.6	1.6489406	1.5504000	0.0985406
0.8	2.1272295	1.9884800	0.1387495
1.0	2.6408591	2.4581760	0.1826831
1.2	3.1799415	2.9498112	0.2301303
1.4	3.7324000	3.4517734	0.2806266
1.6	4.2834838	3.9501281	0.3333557
1.8	4.8151763	4.4281538	0.3870225
2.0	5.3054720	4.8657845	0.4396874

Modified Euler method: Heun's method

- Modified Euler's Method
  - Why a modification?

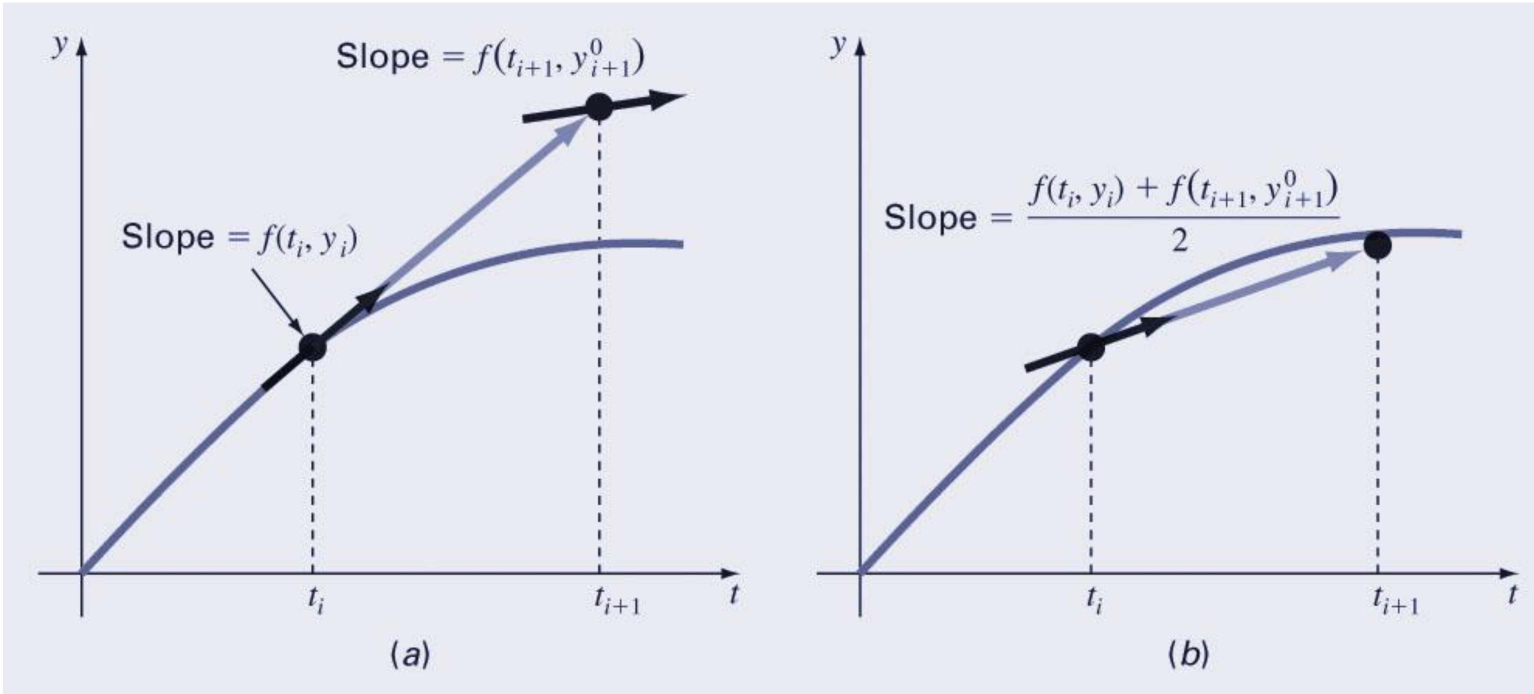


Predictor $\quad y^P_{n+1} = y_n + hf_n$

Average slope $\quad \bar{y}' = \frac{y_n' + y_{n+1}'}{2} = \frac{f(t_n, y_n) + f(t_{n+1}, y^P_{n+1})}{2}$

Corrector $\quad y^C_{n+1} = y_n + h\bar{y}' = y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_{n+1})]$

Heun's method with iteration



Iteration

$$y_{i+1}^{j} \longleftarrow y_i^m + \frac{f(t_i, y_i^m) + f(t_{i+1}, y_{i+1}^{j-1})}{2} h$$

significant improvement

Error analysis

- Error Analysis
  - Taylor series

$$y_{n+1} = y_n + hy'_n + \frac{1}{2}h^2 y''_n + \frac{1}{3!}h^3 y'''(\xi)$$

$$= y_n + hy'_n + \frac{1}{2}h^2 \left\{ \frac{y'_{n+1} - y'_n}{h} + O(h) \right\} + \frac{1}{3!}h^3 y'''(\xi)$$

$$= y_n + \frac{h}{2}[y'_n + y'_{n+1}] + \underline{O(h^3)}$$

truncation
3rd order

  - Total error

$O(h^2)$; 2nd order method

※ Significant improvement over Euler's method!

Eg. Euler vs. Modified Euler

Modified − Euler ∊

Suppose we apply the Runge-Kutta methods of order 2 to our usual example,

$$y' = y - t^2 + 1, \quad \text{for } 0 \le t \le 2, \quad \text{with } y(0) = 0.5,$$

where $N = 10$, $h = 0.2$, $t_i = 0.2i$, and $w_0 = 0.5$ in each case. The difference equations produced from the various formulas are

		Euler Method		Modified Euler	
t_i	$y_i = y(t_i)$	w_i	$\lvert y_i - w_i \rvert$	Modified Euler Method	Error
0.0	0.5000000	0.5000000	0.0000000	0.5000000	0
0.2	0.8292986	0.8000000	0.0292986	0.8260000	0.0032986
0.4	1.2140877	1.1520000	0.0620877	1.2069200	0.0071677
0.6	1.6489406	1.5504000	0.0985406	1.6372424	0.0116982
0.8	2.1272295	1.9884800	0.1387495	2.1102357	0.0169938
1.0	2.6408591	2.4581760	0.1826831	2.6176876	0.0231715
1.2	3.1799415	2.9498112	0.2301303	3.1495789	0.0303627
1.4	3.7324000	3.4517734	0.2806266	3.6936862	0.0387138
1.6	4.2834838	3.9501281	0.3333557	4.2350972	0.0483866
1.8	4.8151763	4.4281538	0.3870225	4.7556185	0.0595577
2.0	5.3054720	4.8657845	0.4396874	5.2330546	0.0724173

improvement

Runge-Kutta method

- Runge-Kutta Method
  - Simple computation

    no $y', y'', \cdots$.   Easy source code
  - very accurate

- The idea

$$y_{n+1} = y_n + h\phi(t_n, y_n, h)$$

where

$$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f(t_n + \alpha_1, y_n + \beta_1)$$

$$\vdots \qquad \qquad \vdots$$

$$k_n = f(t_n + \alpha_{n-1}, y_n + \beta_{n-1})$$

Second-order Runge-Kutta method

- Second-order Runge-Kutta method

$$y_{n+1} = y_n + h(a_1 k_1 + a_2 k_2) \quad \text{①}$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f(t_n + \alpha_1, y_n + \beta_1)$$

Taylor series expansion

$$y_{n+1} = y_n + h(f)_n + \frac{h^2}{2!}(f_t + f_y f)_n + \frac{h^3}{3!} y'''(\xi) \quad \text{②}$$

$$k_2 = (f)_n + \alpha_1 \left(\frac{\partial f}{\partial t}\right)_n + \beta_1 \left(\frac{\partial f}{\partial y}\right)_n + R_n \quad \text{③}$$

③→① $$y_{n+1} = y_n + h(a_1 + a_2)(f)_n + h a_2 (\alpha_1 f_t + \beta_1 f_y)_n + h R_n \quad \text{④}$$

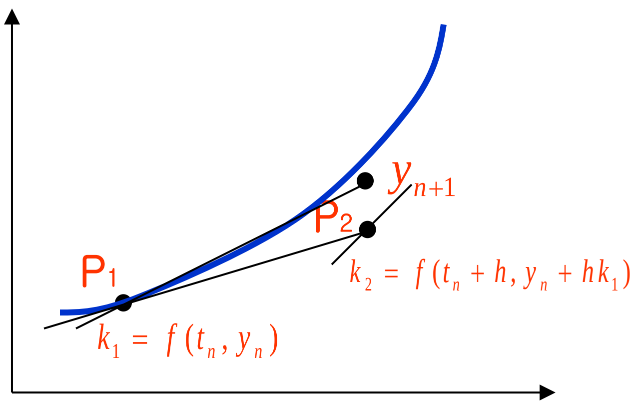
Equating ② and ④

$$a_1 + a_2 = 1, \; a_2 \alpha_1 = \frac{h}{2}, \; a_2 \beta_1 = \frac{h}{2} f(t_n, y_n)$$

Modified Euler - revisited

set $a_2 = \frac{1}{2}$

$$a_1 = \frac{1}{2},\ \alpha_1 = h,\ \beta_1 = hk_1$$



$$\therefore \quad \begin{aligned} y_{n+1} &= y_n + \tfrac{h}{2}(k_1 + k_2) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h, y_n + hk_1) \end{aligned}$$

➔ Modified Euler method

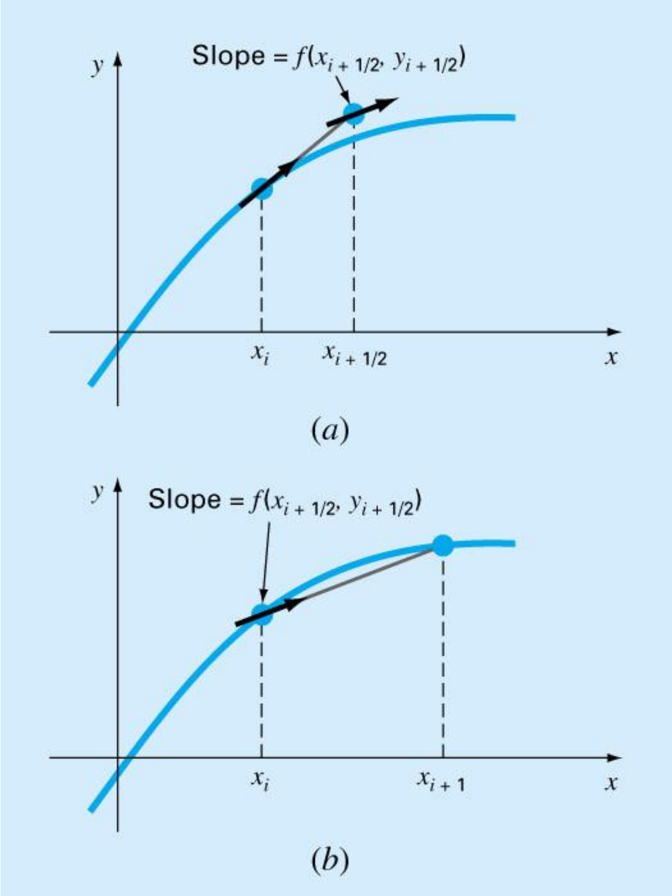
Modified Euler method is a kind of 2nd-order Runge-Kutta method.

Other 2nd order Runge-Kutta methods

- Midpoint method

$$a_1 = 0,\ a_2 = 1,\ \alpha_1 = \frac{h}{2},\ \beta_1 = \frac{h}{2}k_1$$

$$y_{n+1} = y_n + h\left(f\left(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2} f(t_n, y_n)\right)\right)$$



- Ralston's method

$$y_{n+1} = y_n + \tfrac{h}{4}(k_1 + 3k_2)$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}hk_1\right)$$

Comparison: 2^{nd} order R-K method

Suppose we apply the Runge-Kutta methods of order 2 to our usual example,

$$y' = y - t^2 + 1, \quad \text{for } 0 \le t \le 2, \quad \text{with } y(0) = 0.5,$$

where $N = 10$, $h = 0.2$, $t_i = 0.2i$, and $w_0 = 0.5$ in each case. The difference equations produced from the various formulas are

$$\text{Midpoint method:} \quad w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.218;$$

$$\text{Modified Euler method:} \quad w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.216;$$

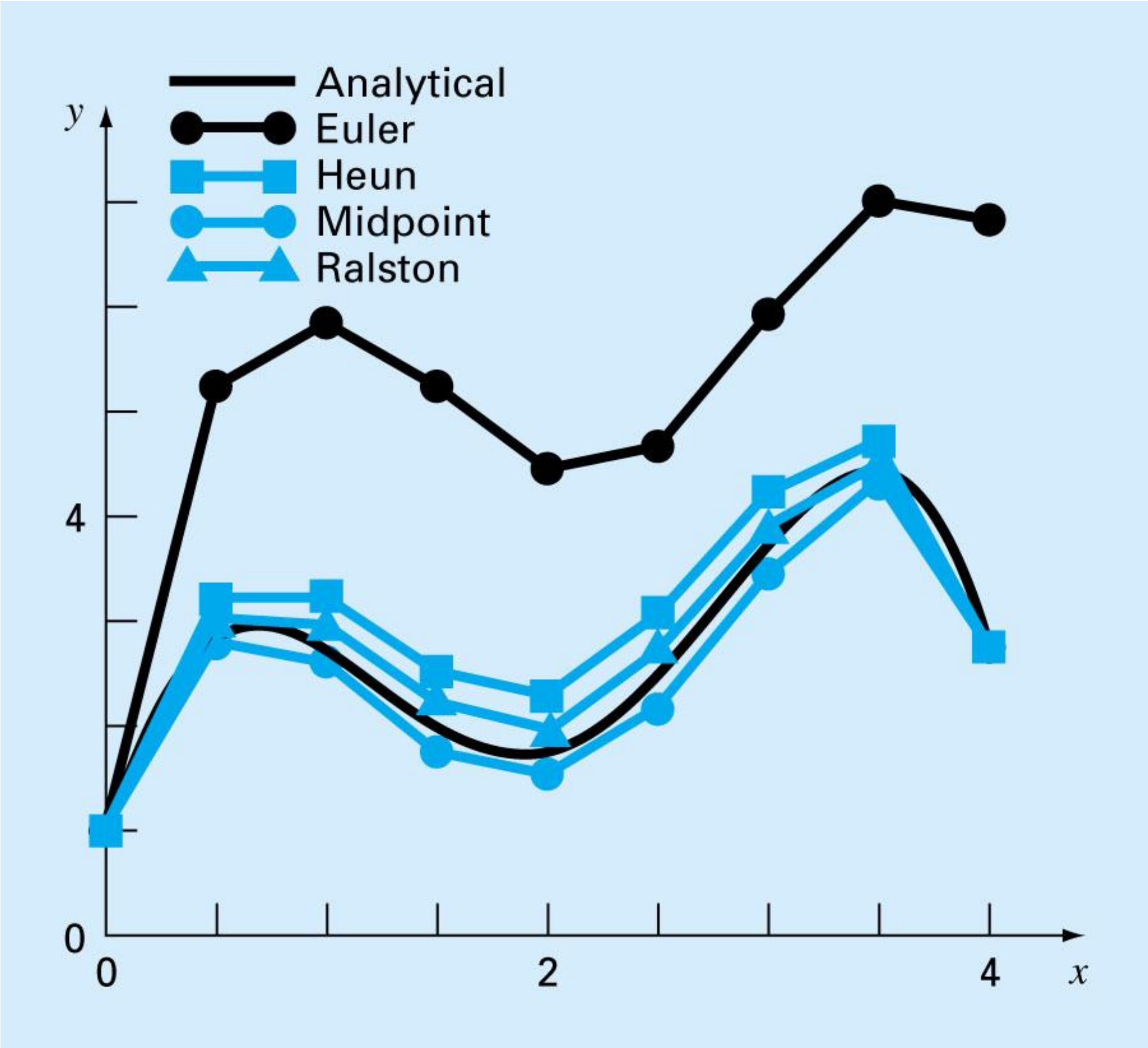
$$\text{Heun's method:} \quad w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.217\overline{3};$$

for each $i = 0, 1, \ldots, 9$. Table 5.5 on page 196 lists the results of these calculations. ■

t_i	$y(t_i)$	Midpoint Method	Error	Modified Euler Method	Error	Heun's Method	Error
0.0	0.5000000	0.5000000	0	0.5000000	0	0.5000000	0
0.2	0.8292986	0.8280000	0.0012986	0.8260000	0.0032986	0.8273333	0.0019653
0.4	1.2140877	1.2113600	0.0027277	1.2069200	0.0071677	1.2098800	0.0042077
0.6	1.6489406	1.6446592	0.0042814	1.6372424	0.0116982	1.6421869	0.0067537
0.8	2.1272295	2.1212842	0.0059453	2.1102357	0.0169938	2.1176014	0.0096281
1.0	2.6408591	2.6331668	0.0076923	2.6176876	0.0231715	2.6280070	0.0128521
1.2	3.1799415	3.1704634	0.0094781	3.1495789	0.0303627	3.1635019	0.0164396
1.4	3.7324000	3.7211654	0.0112346	3.6936862	0.0387138	3.7120057	0.0203944
1.6	4.2834838	4.2706218	0.0128620	4.2350972	0.0483866	4.2587802	0.0247035
1.8	4.8151763	4.8009586	0.0142177	4.7556185	0.0595577	4.7858452	0.0293310
2.0	5.3054720	5.2903695	0.0151025	5.2330546	0.0724173	5.2712645	0.0342074

Comparison: 2nd order R-K method

Eg. $y' = -2x^3+12x^2-20x+8.5$, $y(0)=1$

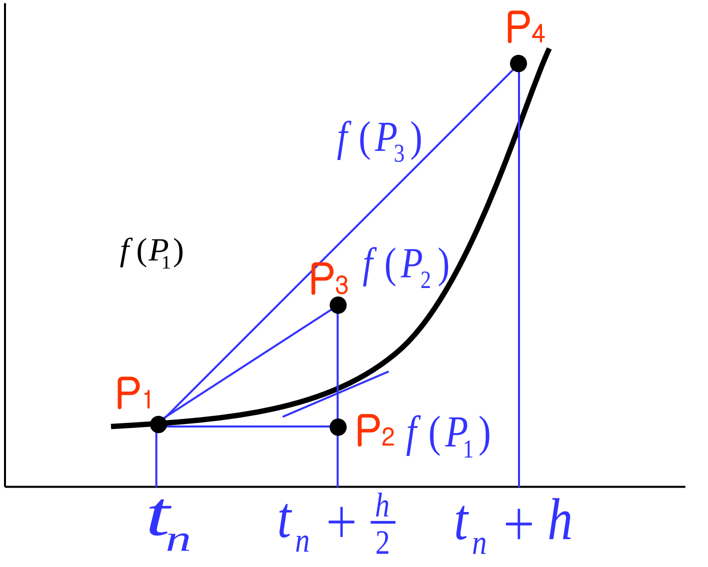


4-th order Runge-Kutta methods

- Fourth-order Runge-Kutta
  - Taylor series expansion to 4-th order
  - accurate
  - short, straight, easy to use

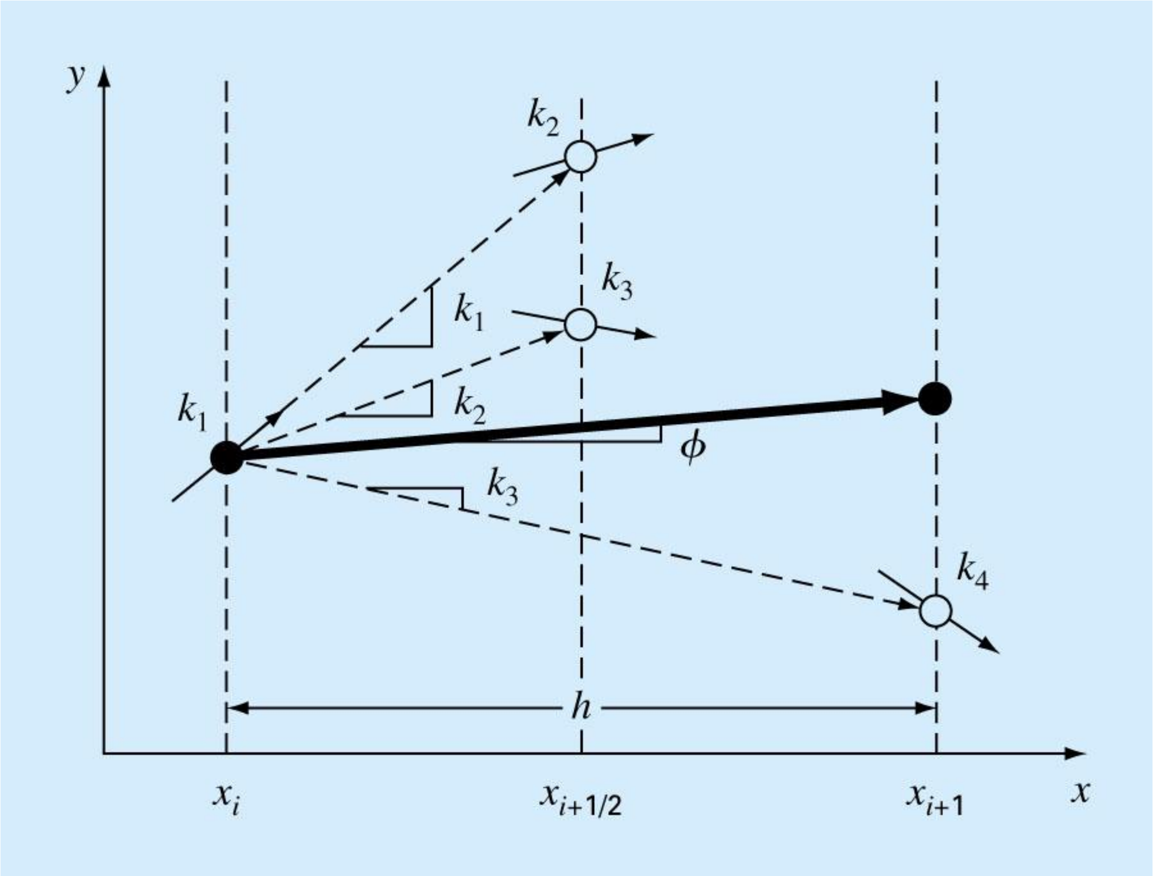
$$y_{n+1} = y_n + \tfrac{1}{6}h\{k_1 + 2(k_2 + k_3) + k_4\}$$

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(t_n + h, y_n + hk_3) \end{cases}$$



※ significant improvement over modified Euler's method

Runge-Kutta method



Eg. 4-th order R-K method

The Runge-Kutta method of order 4 applied to the initial-value problem

$$y' = y - t^2 + 1, \quad \text{for } 0 \le t \le 2, \quad \text{with } y(0) = 0.5,$$

with $h = 0.2$, $N = 10$, and $t_i = 0.2i$, gives the results and errors listed in Table 5.6.

t_i	Exact $y_i = y(t_i)$	Runge-Kutta Order 4 w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272027	0.0000269
1.0	2.6408591	2.6408227	0.0000364
1.2	3.1799415	3.1798942	0.0000474
1.4	3.7324000	3.7323401	0.0000599
1.6	4.2834838	4.2834095	0.0000743
1.8	4.8151763	4.8150857	0.0000906
2.0	5.3054720	5.3053630	0.0001089

← Significant improvement

Midpoint Method	Error
0.5000000	0
0.8280000	0.0012986
1.2113600	0.0027277
1.6446592	0.0042814
2.1212842	0.0059453
2.6331668	0.0076923
3.1704634	0.0094781
3.7211654	0.0112346
4.2706218	0.0128620
4.8009586	0.0142177
5.2903695	0.0151025

Discussion

For the problem

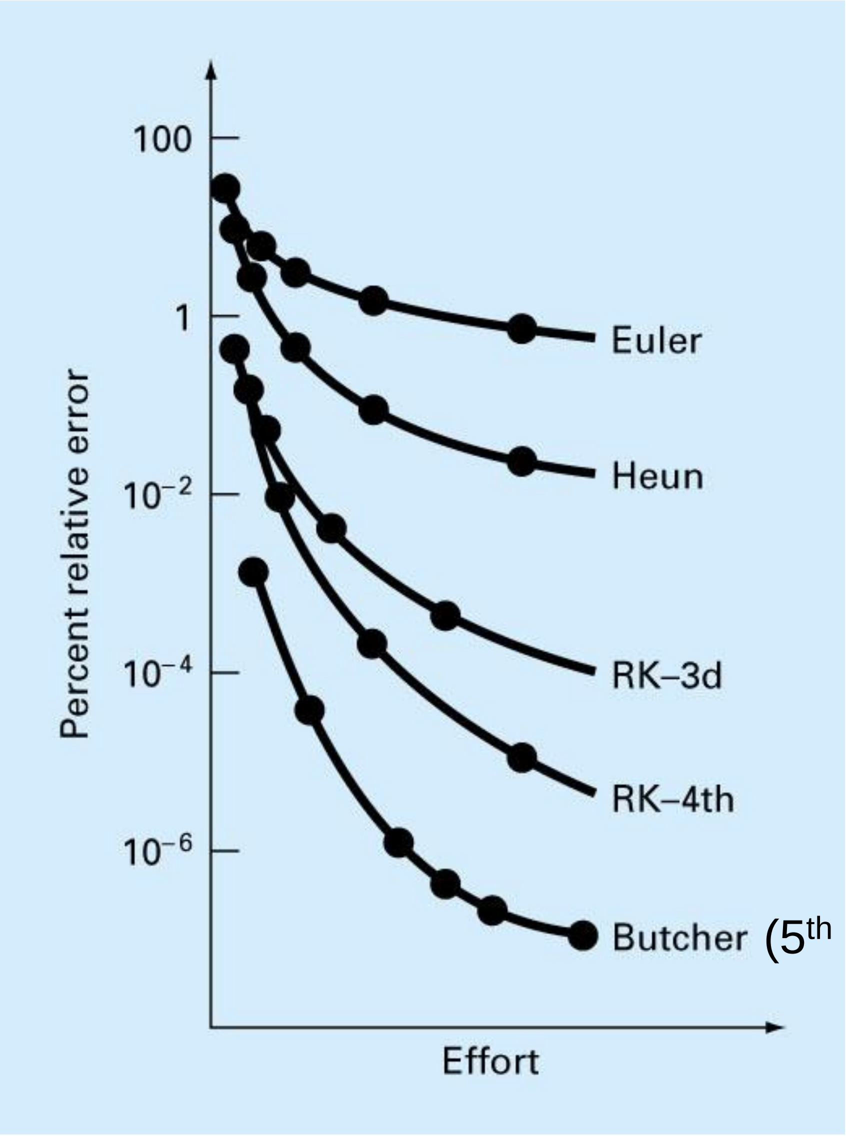
$$y' = y - t^2 + 1, \quad \text{for } 0 \le t \le 2, \quad \text{with } y(0) = 0.5,$$

Euler's method with $h = 0.025$, the Modified Euler's method with $h = 0.05$, and the Runge-Kutta method of order 4 with $h = 0.1$ are compared at the common mesh points of the three methods, 0.1, 0.2, 0.3, 0.4, and 0.5. Each of these techniques requires 20 functional evaluations to approximate $y(0.5)$. (See Table 5.8.) In this example, the fourth-order method is clearly superior, as it is in most situations. ■

t_i	Exact	Euler $h = 0.025$	Modified Euler $h = 0.05$	Runge-Kutta Order 4 $h = 0.1$
0.0	0.5000000	0.5000000	0.5000000	0.5000000
0.1	0.6574145	0.6554982	0.6573085	0.6574144
0.2	0.8292986	0.8253385	0.8290778	0.8292983
0.3	1.0150706	1.0089334	1.0147254	1.0150701
0.4	1.2140877	1.2056345	1.2136079	1.2140869
0.5	1.4256394	1.4147264	1.4250141	1.4256384

→ Better!

Comparison



(5th order)