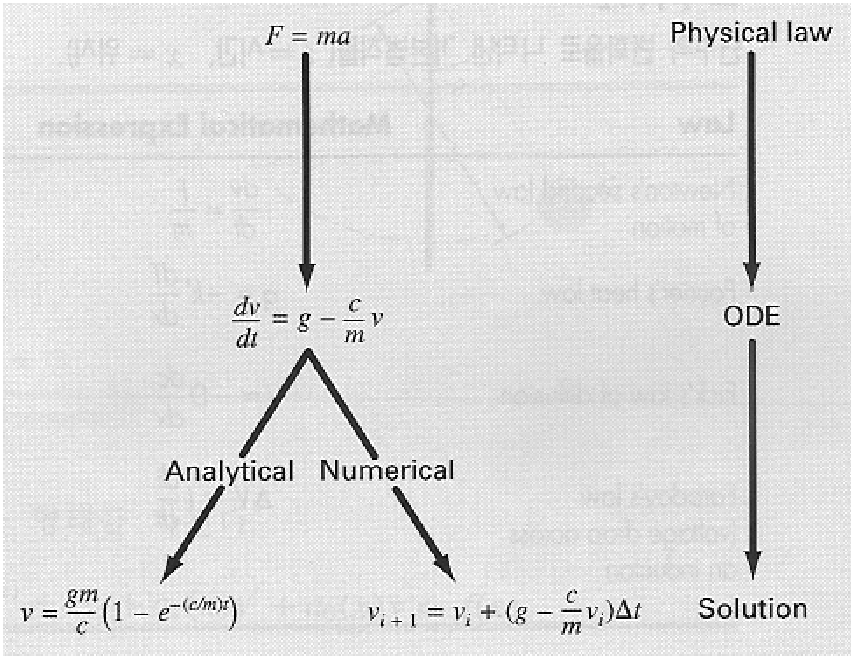
7. 상미분 방정식 (ODE)

1. 서 론

자연 현상 → 물리법칙 적용 → 수학적 표현 → 미분방정식
Natural phenomenon → Physics Law → Mathematical expression → DE



자연현상뿐 아니라 공학적인 문제에서도 미분방정식이 많이 활용되나
공학분야 대부분의 미분방정식들은 해석적으로는 풀리지 않고
수치적인 접근 방법을 필요로 한다.
Many engineering problems are analytically unsolvable.

일반적으로 공학 분야 미분방정식은
1. Simplified 된 equation 을 해석적으로 푸는 방법
2. Original Equation 을 approximate 하게 푸는 방법
등 두 가지 방법으로 해를 찾을 수 있는데,
정확한 Error estimation 이 가능한 두 번째 방법이 선호되며
수치해석적 방법이 이에 해당한다.

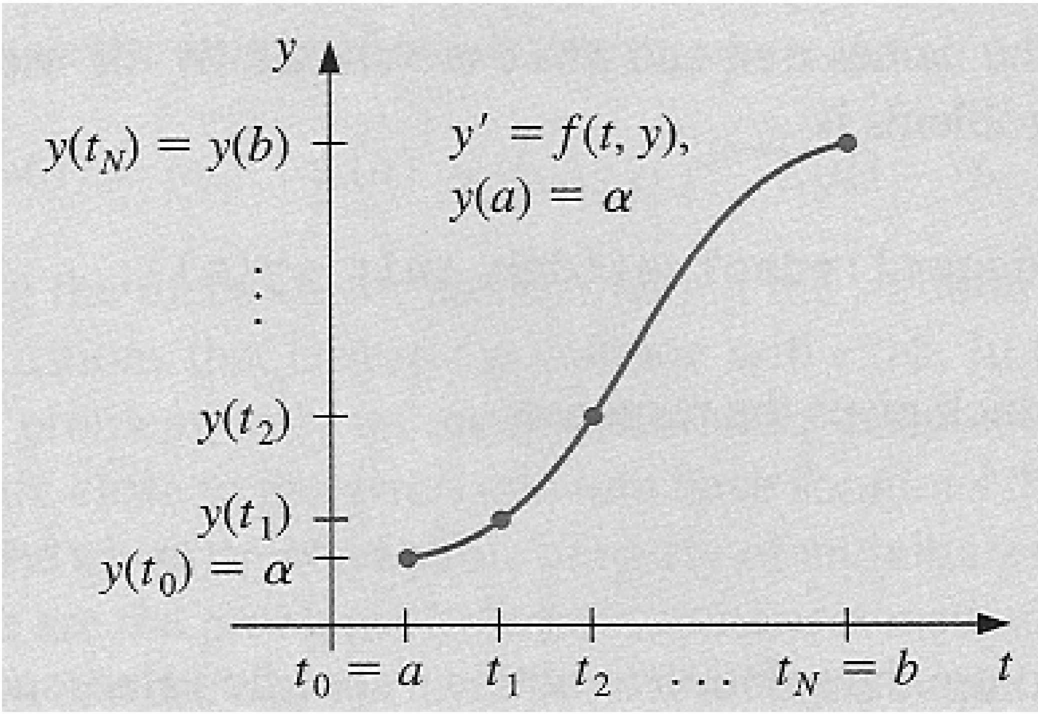
- 상미분/편미분 ordinary/partial, 선형/비선형 linear/nonlinear
- homogeneous/nonhomogeneous
- Initial value problem/boundary value problem

2. Initial value Problems

$\frac{dy}{dt} = f(t, y)$ for $a \le t \le b$, with $y(a) = \alpha$

$t_i = a + ih$, for each I = 0, 1, 2, ⋯, N where $h = (b-a)/N$

*t*를 *a* 부터 *h* 만큼씩 증가시켜 가면서 y 값을 구하는 문제
To find *y* values increasing *t* from *a* by *h* step by step.



t = a 에서 *y, dy/dt* 값을 알고 있다고 할 때,
t = a + h 에서의 y 값을 어떠한 방법으로 구할 것인가가 궁극적인 문제

How to obtain *y* value at $t = a + h$, when knowing *y* and *dy*/*dt* at $t = a$

2.1 Taylor Method

$$y(t_o+h)=y(t_o)+y_o^{'}h+\frac{y_o^{''}}{2!}h^2+\cdots+\frac{y_o^{n}}{n!}h^n+R_n$$

- Euler Method

$$y_{i+1}=y_i+f(t_i,y_i)h$$

Example) 다음 방정식에 대해 x=0 에서 주어진 초기 조건을 이용, 구간 간격 h=0.5 로 x=4 에서의 y 값을 구하시오.

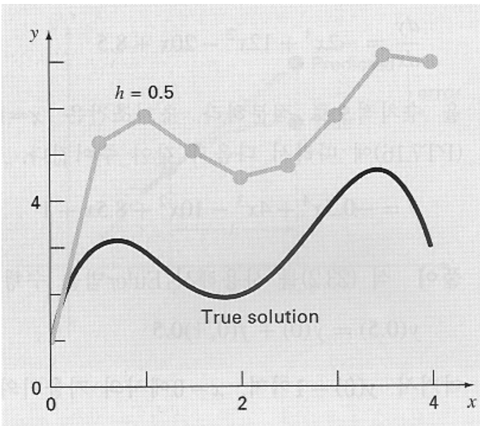
$$\frac{dy}{dx}=-2x^3+12x^2-20x+8.5$$

초기 조건 initial condition: x=0 에서 y=1

y(0.5) = y(0) + f(0,1) * 0.5
f(0,1) = 8.5
y(0.5) = 1 + 8.5 * 0.5 = 5.25 (real value = 3.21875)

y(1.0) = y(0.5) + f(0.5,5.25) * 0.5
= 5.25 + [$-2*0.5^3 + 12*0.5^2 - 20*0.5 + 8.5$] * 0.5
= 5.875 (real value = 3.0)

x	y_{true}	y_{Euler}	Percent Relative Error Global	Percent Relative Error Local
0.0	1.00000	1.00000		
0.5	3.21875	5.25000	−63.1	−63.1
1.0	3.00000	5.87500	−95.8	−28.0
1.5	2.21875	5.12500	131.0	−1.41
2.0	2.00000	4.50000	−125.0	20.5
2.5	2.71875	4.75000	−74.7	17.3
3.0	4.00000	5.87500	46.9	4.0
3.5	4.71875	7.12500	−51.0	−11.3
4.0	3.00000	7.00000	−133.3	−53.0



"세련되지 않은" Euler 법의 가상 code

```
C     set integration range
         xi = 0
         xf = 4
C     Initialize variables
         x = xi
         y = 1
C     set step size and determine number of calculation steps
         dx = 0.5
         nc = (xf – xi) / dx
C     output initial condition
         PRINT x, y
C     Loop to implement Euler's method and display results
         DO i = 1, nc
            dydx = −2 x^3 + 12 x^2 – 20x + 8.5
            y = y + dydx * dx
            x = x + dx
            PRINT x, y
         ENDDO
```

"세련된" (모듈화된) Euler 법 가상 code

(a) Main Program
```
C     Assign values for
        y = [initial value of dependent variable]
        xi = [initial value of independent variable]
        xf = [final value of independent variable]
        dx = [calculation step size]
        xout = [output interval]
C
        x = xi
        m = 0
        xp(m) = x
        yp(m) = y
        DO
            xend = x + xout
            IF (xend > xf) THEN xend = xf
            h = dx
            CALL Integrator (x, y, h, xend)
            m = m + 1
            xp(m) = x
            yp(m) = y
            IF(x ≥ xf) EXIT
        ENDDO
        DISPLAY RESULTS
        END
```

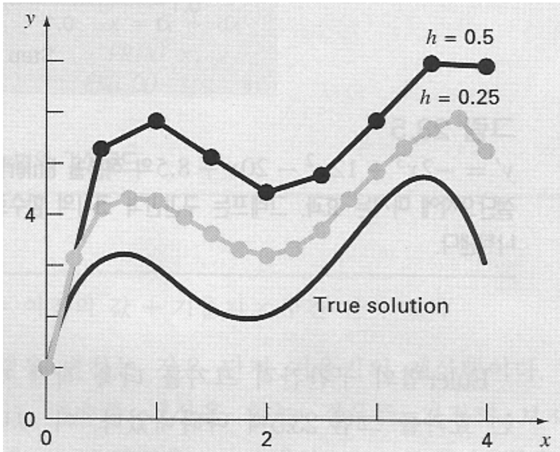
(b) Routine to take one output step
```
    SUB Integrator (x, y, h, xend)
      DO
          IF (xend - x < h) THEN h = xend - x
          CALL Euler (x, y, h, ynew)
          x = x + h
          y = ynew
          IF (x ≥ xf) EXIT
      ENDDO
    ENDSUB
```

(c) Euler's method for a single ODE
```
    SUB Euler (x, y, h, ynew)
      CALL Derivs (x, y, dydx)
      ynew = y + dydx * h
    ENDSUB
```

(d) Obtain Derivative
```
    SUB Derivs (x, y, dydx)
      dydx = …
    ENDSUB
```

Euler method 에서 error 를 줄이는 방법은 (How to reduce error)
(1) 구간 간격 (h)를 줄인다. (Reduce the size of h)
(2) Taylor series 에서 고차 항을 이용한다. (Use higher order term)

(1) 앞의 문제에서 h 를 0.25 로 줄였을 때 효과



(2) 고차 Taylor method

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{f'(t_i, y_i)}{2!}h^2$$

$$f'(t, y(t)) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y}\frac{dy(t)}{dt}$$

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

Example) $y' = y - t^2 + 1$, for 0 ≤ t ≤ 2, with y(0) = 0.5 & h = 0.2

t_i	Exact $y(t_i)$	Taylor Order 1 w_i	Error $\lvert y_i - w_i \rvert$	Taylor Order 2 w_i	Error $\lvert y(t_i) - w_i \rvert$	Taylor Order 4 w_i	Error $\lvert y(t_i) - w_i \rvert$
0.0	0.5000000	0.5000000	0.0000000	0.5000000	0	0.5000000	0
0.2	0.8292986	0.8000000	0.0292986	0.8300000	0.0007014	0.8293000	0.0000014
0.4	1.2140877	1.1520000	0.0620877	1.2158000	0.0017123	1.2140910	0.0000034
0.6	1.6489406	1.5504000	0.0985406	1.6520760	0.0031354	1.6489468	0.0000062
0.8	2.1272295	1.9884800	0.1387495	2.1323327	0.0051032	2.1272396	0.0000101
1.0	2.6408591	2.4581760	0.1826831	2.6486459	0.0077868	2.6408744	0.0000153
1.2	3.1799415	2.9498112	0.2301303	3.1913480	0.0114065	3.1799640	0.0000225
1.4	3.7324000	3.4517734	0.2806266	3.7486446	0.0162446	3.7324321	0.0000321
1.6	4.2834838	3.9501281	0.3333557	4.3061464	0.0226626	4.2835285	0.0000447
1.8	4.8151763	4.4281538	0.3870225	4.8462986	0.0311223	4.8152377	0.0000615
2.0	5.3054720	4.8657845	0.4396874	5.3476843	0.0422123	5.3055554	0.0000834

2.2 Modification of the Euler Method

※ Euler method 에서 오차가 발생하는 근본 원인은 구간 시작점에서의 도함수를 전체 구간에 적용하기 때문이다.
이를 보완하는 두 가지 방법: Heun method, Midpoint method
Fundamental reason for the error comes from that the derivative at initial point was used in the whole interval, $t=a \sim t= a+h$

- Heun Method
  구간의 시작점과 끝점에서 도함수를 구하고 두 값의 평균을 취함
  Use average of derivatives at the two end points of the interval

  예측자 방정식 (predictor equation)

$$y_{i+1}^{o} = y_i + f(x_i, y_i)h$$

  y_{i+1} 의 추정값을 사용 구간의 끝점에서의 도함수를 계산

$$y'_{i+1} = f(x_{i+1}, y_{i+1}^{o})$$

  구간 평균 기울기를 계산

$$\bar{y}' = \frac{y'_i + y'_{i+1}}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{o})}{2}$$

  평균 기울기를 이용하여 y_{i+1} 값을 근사

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{o})}{2} h$$

  (수정자 방정식 corrector equation)

Example) $y' = 4e^{0.8x} - 0.5y$ 인 함수 x=4 에서의 적분 값을 구하라.

구간 간격은 1 이고 초기 값은 x = 0 에서 y = 2.

$$y'_o = 4e^{0} - 0.5 \cdot 2 = 3 \qquad y_1^{o} = y_o + y'_o h = 2 + 3 \cdot 1 = 5$$

$$y'_1 = f(x_1, y_1^{o}) = 4e^{0.8} - 0.5 \cdot 5 = 6.402164$$
$$y' = (y'_o + y'_1)/2 = (3 + 6.402164)/2 = 4.701082$$
$$y_1 = y_o + y' h = 2 + 4.701082 \cdot 1 = 6.701082$$

더 이상의 y_1 값 변화가 없을 때까지 y’ 및 y_1 의 계산을 반복할 수 있다. One may repeat until y1 converges

		Iterations of Heun's Method			
		1		15	
x	y_{true}	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)
0	2.0000000	2.0000000	0.00	2.0000000	0.00
1	6.1946314	6.7010819	8.18	6.3608655	2.68
2	14.8439219	16.3197819	9.94	15.3022367	3.09
3	33.6771718	37.1992489	10.46	34.7432761	3.17
4	75.3389626	83.3377674	10.62	77.7350962	3.18

※ 도함수가 종속변수 (*t* 또는 *x*) 만의 함수인 상미분 방정식에서는 예측자 계산이 필요 없고 이 기법은 다음과 같이 간략하게 표현된다.

$$y_{i+1} = y_i + \frac{f(x_i) + f(x_{i+1})}{2} h$$

이는 사다리꼴 적분 공식과 정확하게 같은 표현이다.
When *f* is a function of only *x* not *y* -> trapezoidal integration

- Midpoint Method (Use *y'* value at the midpoint)
  구간의 평균 도함수 값을 구하기 위해 Heun method 에서는 양 끝점 값의 평균을 내는 반면, Midpoint 법에서는 중앙점에서의 y 값을 Euler 법으로 예측하고 y' 값을 구한 후 이 y' 값을 이용하여 구간 끝점 (x_{i+1})에서의 y 값을 선형 보간으로 예측한다.

중앙점에서의 y 값 예측 $y_{i+1/2} = y_i + f(x_i, y_i)\frac{h}{2}$

중앙점에서의 기울기 계산 $y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$

중앙점 기울기를 평균 기울기로 가정, y_{i+1} 값 근사

$$y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h$$

※ 기울기가 x 만의 함수일 경우 적분에서의 midpoint 법과 같은 표현이 됨.

$$y_{i+1} = y_i + f(x_{i+1/2})h$$

(0) Euler's method for a single ODE

```
SUB Euler (x, y, h, ynew)
      CALL Derivs (x, y, dydx)
      ynew = y + dydx * h
ENDSUB
```

(a) Simple Heun without Corrector

```
SUB Heun (x, y, h, ynew)
      CALL Derivs (x, y, dy1dx)
      ye = y + dy1dx * h
      CALL Derivs (x+h, ye, dy2dx)
      Slope = (dy1dx +dy2dx) / 2
      ynew = y + Slope * h
ENDSUB
```

(b) Midpoint method

```
SUB Midpoint (x, y, h, ynew)
      CALL Derivs (x, y, dydx)
      ym = y + dydx * h/2
      CALL Derivs (x+h/2, ym, dymdx)
      ynew = y + dymdx * h
ENDSUB
```

(c) Heun with Corrector

```
SUB HeunIter (x, y, h, ynew)
      es = 0.01
      maxit = 20
      CALL Derivs (x, y, dy1dx)
      ye = y + dy1dx * h
      Iter = 0
      DO
          yeold = ye
          CALL Derivs (x+h, ye, dy2dx)
          Slope = (dy1dx +dy2dx) / 2
          ye = y + Slope * h
          iter = iter + 1
          ea = | (ye – yeold)/ye | * 100%
          IF (ea ≤ es OR iter > maxit) EXIT
      ENDDO
      ynew = ye
ENDSUB
```

2.3 Runge-Kutta Method

RK 법은 고차 도함수를 계산하지 않으면서 높은 Taylor 급수의 정확도를 얻을 수 있는 기법으로, 앞의 Euler, Heun, Midpoint 법 모두 RK 라 불리는 single-step method 의 일종이며, 다음의 일반적인 형태를 가진다.

$$y_{i+1} = y_i + \phi(x_i, y_i, h)h \qquad \text{Eq.(1)}$$

여기서 $\phi(x_i, y_i, h)$ 는 구간 상에서 대표적인 기울기를 나타내는 증분함수 (increment function)이며 다음의 일반적 표현을 가짐.

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

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$
$$k_3 = f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h)$$
$$\vdots$$
$$k_n = f(x_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h)$$

n=1 인 1 차 RK 법은 Euler 법이 되고, n 이 선택되면 a, p, q 값들은 Eq.(1)을 Taylor 급수 전개하여 같은 식이 되도록 하여 결정한다.

- 2 차 RK 법의 유도

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h \qquad \text{Eq.(2)}$$

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

y_{i+1} 을 y_i 와 $f(x_i,y_i)$에 대해 2 차 Taylor 전개

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{f'(x_i, y_i)}{2!}h^2$$

$$f'(x_i, y_i) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}\frac{dy}{dx}$$

$$y_{i+1} = y_i + f(x_i, y_i)h + \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx}\right)\frac{h^2}{2!} \qquad \text{Eq.(3)}$$

RK 법의 기본 개념은 식 (2)와 (3)이 같아지도록 a, p, q 값들을 구하는 것

$$f(x_i + p_1 h, y_i + q_{11} k_1 h) = f(x_i, y_i) + p_1 h \frac{\partial f}{\partial x} + q_{11} k_1 h \frac{\partial f}{\partial y} + O(h^2)$$

$$y_{i+1} = y_i + a_1 h f(x_i, y_i) + a_2 h f(x_i, y_i) + a_2 p_1 h^2 \frac{\partial f}{\partial x} + a_2 q_{11} h^2 f(x_i, y_i)\frac{\partial f}{\partial y} + O(h^3)$$

$$y_{i+1} = y_i + [a_1 f(x_i, y_i) + a_2 f(x_i, y_i)]h + [a_2 p_1 \frac{\partial f}{\partial x} + a_2 q_{11} f(x_i, y_i)\frac{\partial f}{\partial y}]h^2 + O(h^3)$$

Eq.(4)

Eq.(3) = Eq.(4) 관계로부터

$$a_1 + a_2 = 1, \qquad a_2 p_1 = \frac{1}{2}, \qquad a_2 q_{11} = \frac{1}{2}$$

세 개의 연립 방정식이 4 개의 미지수를 가지므로 미지수 중 하나의 값을 가정해야 하고, 이에 따라 여러 개의 2 차 RK 법이 존재하게 된다.

- $a_2 = \frac{1}{2}$ 일 때, $a_1 = \frac{1}{2}$, $p_1 = 1$, $q_{11} = 1$

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h \qquad k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + h, y_i + k_1 h)$$

수정자가 한 번 반복되는 Heun 법에 해당

- $a_2 = 1$ 일 때, $a_1 = 0$, $p_1 = \frac{1}{2}$, $q_{11} = \frac{1}{2}$

$$y_{i+1} = y_i + k_2 h \qquad k_1 = f(x_i, y_i)$$

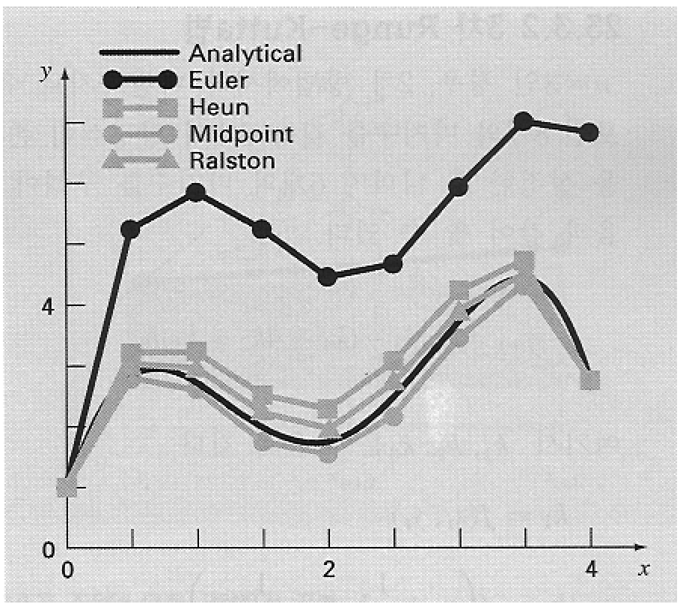
$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h)$$

Midpoint 법에 해당

- $a_2 = \frac{2}{3}$ 일 때, $a_1 = \frac{1}{3}$, $p_1 = q_{11} = \frac{3}{4}$ (Ralston 법)

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h \qquad k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1 h)$$



- 3 차 Runge-Kutta 법

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 4k_2 + k_3)h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h)$$

$$k_3 = f(x_i + h, y_i - k_1 h + 2k_2 h)$$

도함수가 x 만의 함수라면 Simpson 1/3 공식에 해당

- 4 차 Runge-Kutta 법 (classical 4 차 RK 법)

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h)$$

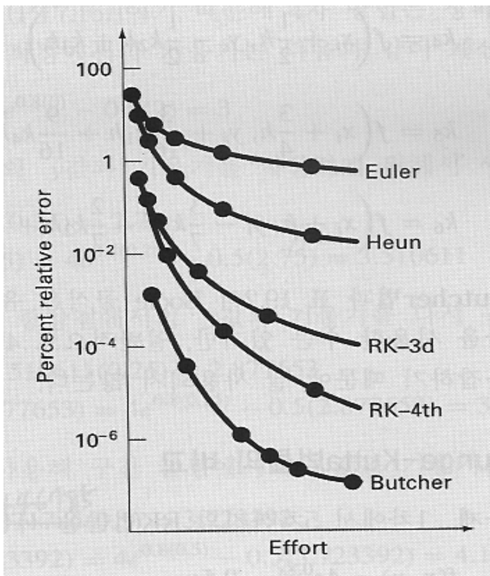
$$k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h)$$

$$k_4 = f(x_i + h, y_i + k_3 h)$$

도함수가 x 만의 함수이면 역시 Simpson 1/3 공식에 해당

classical 4 차 RK

```
SUB RK4 (x, y, h, ynew)
    CALL Derivs (x, y, k1)
    ym = y + k1 * h / 2
    CALL Derivs (x+h/2, ym, k2)
    ym = y + k2 * h / 2
    CALL Derivs (x+h/2, ym, k3)
    ye = y + k3 * h
    CALL Derivs (x+h, ye, k4)
    Slope = (k1+2*k2+2*k3 +k4)/6
    ynew = y + Slope * h
ENDSUB
```



2.4 Methods for Systems of Equations

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \cdots, y_n)$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \cdots, y_n)$$

$$\vdots$$

$$\frac{dy_n}{dx} = f_n(x, y_1, y_2, \cdots, y_n)$$

n 개의 초기 조건을 필요

Example) 다음 연립미분방정식을 구간간격 0.5 로 x=2.0 에서 풀어라

$$\frac{dy_1}{dx} = -0.5y_1$$

$$\frac{dy_2}{dx} = 4 - 0.3y_2 - 0.1y_1$$

초기조건

$y_1(x=0) = 4$

$y_2(x=0) = 6$

- Euler 법

  $y_1(0.5) = 4 + [-0.5 * 4] * 0.5 = 3$

  $y_2(0.5) = 6 + [4 - 0.3*6- 0.1*4] * 0.5 = 6.9$

  두 번째 식에서 y_1 에, 첫 번째 식에서 계산한 3 이 아닌 $y_1(x=0) = 4$ 를 사용했음에 유의

x	y_1	y_2
0	4	6
0.5	3	6.9
1.0	2.25	7.715
1.5	1.6875	8.44525
2.0	1.265625	9.094087

- 4 차 RK 법

x	y_1	y_2
0	4	6
0.5	3.115234	6.857670
1.0	2.426171	7.632106
1.5	1.889523	8.326886
2.0	1.471577	8.946865

(a) Main Program

```
C   Assign values for
      n = number of equations
      y(i) = [initial values of n
               dependent variables]
      xi = [initial value of independent
             variable]
      xf = [final value of independent
             variable]
      dx = [calculation step size]
      xout = [output interval]
C
      x = xi
      m = 0
      xp(m) = x
      DO i = 1, n
         yp(i,m) = y(i)
      ENDDO
      DO
        xend = x + xout
        IF (xend > xf) THEN xend = xf
        h = dx
        CALL Integrator (x, y, h, xend)
        m = m + 1
        xp(m) = x
        DO i = 1, n
           yp(i,m) = y
        ENDDO
        IF(x ≥ xf) EXIT
      ENDDO
      DISPLAY RESULTS
      END
```

(b) Routine to take one output step

```
      SUB Integrator (x, y, n, h, xend)
      DO
        IF (xend-x< h) THEN h=xend-x
          CALL RK4 (x, y, n, h)
          x = x + h
          IF (x ≥ xf) EXIT
      ENDDO
      ENDSUB
```

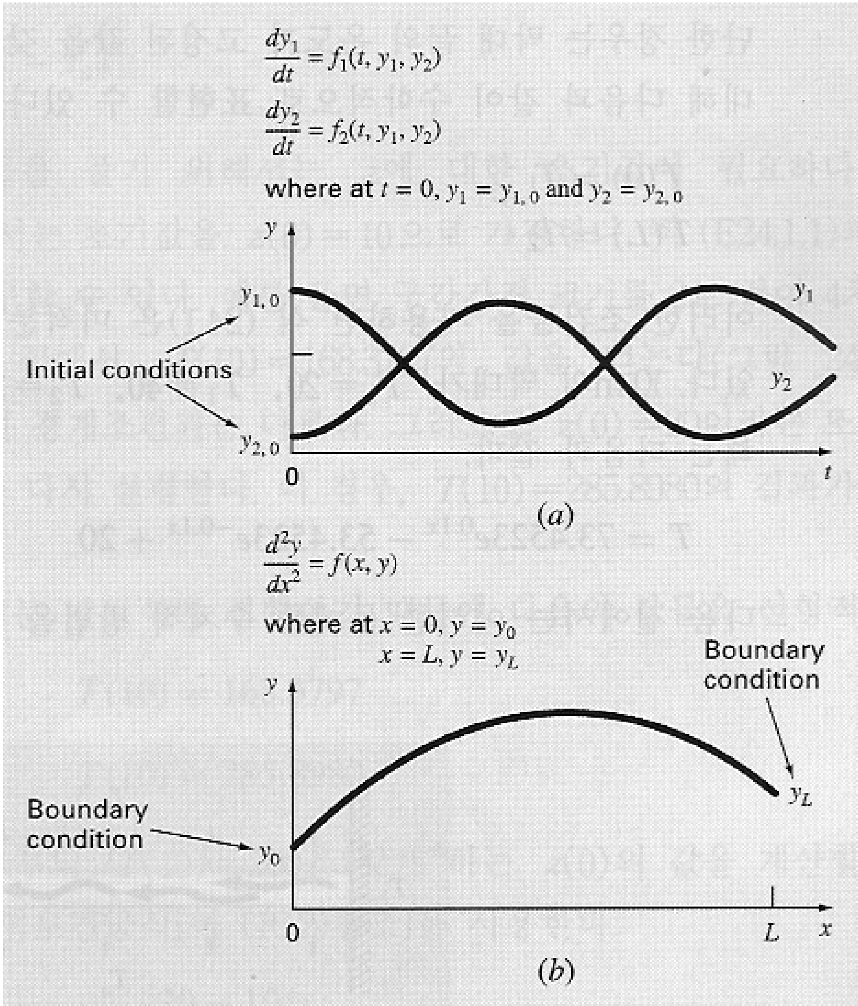
(c) Classical 4 차 RK

```
    SUB RK4 (x, y, h, ynew)
      CALL Derivs (x, y, k1)
      DO i = 1, n
         ym(i) = y(i) + k1(i) * h /2
      ENDDO
      CALL Derivs (x+h/2, ym, k2)
      DO i = 1, n
         ym(i) = y(i) + k2(i) * h /2
      ENDDO
      CALL Derivs (x+h/2, ym, k3)
      DO i = 1, n
         ye(i) = y(i) + k3(i) * h
      ENDDO
      CALL Derivs (x+h, ye, k4)
      DO i = 1, n
         Slope(i) = (k1(i)+2*k2(i)
                      +2*k3(i) +k4(i))/6
         y(i) = y(i) + Slope(i) * h
      ENDDO
     ENDSUB
```

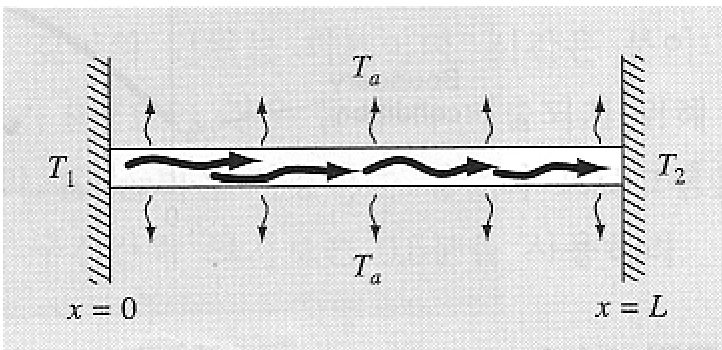
(d) Obtain Derivative

```
    SUB Derivs (x, y, dy)
      dy(1) = …
      dy(2) = …
      …
     ENDSUB
```

3. Boundary value Problems



Example) $\frac{d^2T}{dx^2} + h'(T_a - T) = 0$



$T(0) = T_1$ (= 40)
$T(L) = T_2$ (=200)
$T_a = 20$
$h' = 0.01$
$L = 10$

Exact solution : $T = 73.4523e^{0.1x} - 53.4523\ e^{-0.1x} + 20$

3.1 Shooting Method

- Linear Shooting Method

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

문제의 2 차 방정식을 두 개의 1 차 상미분 방정식으로 분해

$$\frac{dT}{dx} = z$$

$$\frac{dz}{dx} = h'(T - T_a)$$

위 방정식들을 풀기 위해서는 z 의 초기값이 필요하다.
초기값 z(0)=10 을 가정하고 두 개의 연립 방정식을 4 차 RK 로 풀면 T(10) = 168.3797 이 얻어진다. 이는 경계조건 T(10)=200 과 다르다.
따라서 z(0)=20 인 또 다른 초기값을 가정하여 계산을 실행한다.
이때는 T(10) = 285.8980 이 얻어진다.
본래의 상미분 방정식은 선형이므로 다음 값들은 선형적으로 변한다.

z(0) = 10　　　　T(10) = 168.3797
z(0) = 20　　　　T(10) = 285.8980

따라서 이들로부터 T(10) = 200 갖게 하는 z(0) 값을 계산할 수 있다.

$$z(0) = 10 + \frac{20-10}{285.8980 - 168.3797}(200 - 168.3797) = 12.6907$$

- Nonlinear Shooting Method

비선형 문제의 경우 T(10) 값이 z(0)에 따라 선형적으로 변하지 않기 때문에 위의 마지막 식으로 주어진 T(10) 값을 갖게 하는 z(0) 값을 결정할 수 없다.
비선형 방법의 근간은 주어진 z(0)로부터 T(10) 값이 구해지는 과정을 하나의 함수로 간주하고

T(10) = f(z(0))

$f(z_0)$ – T(10) = 0 인 nonlinear equation 을 푸는 것이다.

고차방정식에 적용하는 경우 상당히 복잡하고 어렵다.
다른 대안을 찾을 필요가 있다.
→ finite-difference method (유한차분법)

3.2 Finite Difference Method

- Linear Finite Difference Method

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

방정식에 있는 도함수를 유한제차분들로 근사시키는 것이다.
미분 방정식이 연립방정식으로 변환된다.

$$\frac{d^2T}{dx^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}$$

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} - h'(T_i - T_a) = 0$$

방정식에 1 차 도함수가 포함되어 있으면 이 역시 차분화한 후 T_i 항 별로 재정리한다.

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_a$$

$$-T_i + (2 + h'\Delta x^2)T_{i+1} - T_{i+2} = h'\Delta x^2 T_a$$

$$\vdots$$

앞의 열전달 문제에 네 개의 내부 절점 ($\Delta x = 2$)을 적용하면

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ & -1 & 2.04 & -1 \\ & & -1 & 2.04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{Bmatrix}$$

$\{T\}^T = \{65.9698 \quad 93.7785 \quad 124.5382 \quad 159.4795\}$

Shooting법과 유한차분법을 사용해 얻은 결과와 정확한 해석해와의 비교.

x	True	Shooting Method	Finite Difference
0	40	40	40
2	65.9518	65.9520	65.9698
4	93.7478	93.7481	93.7785
6	124.5036	124.5039	124.5382
8	159.4534	159.4538	159.4795
10	200	200	200

- Nonlinear Finite Difference Method

원래 미분방정식이 선형이 아닐 때는 위에서 얻었던 연립방정식

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_a$$

$$-T_i + (2 + h'\Delta x^2)T_{i+1} - T_{i+2} = h'\Delta x^2 T_a$$

$$\vdots$$

도 비선형이 된다.
이 방정식들은 비선형 연립 방정식 풀이 기법으로 푼다.

Problem) 다음의 비선형 미분방정식을 FDM 을 이용해 푸시오.

$$y'' = \frac{1}{8}(32 + 2x^3 - yy') \qquad \text{for} \quad 1 \le x \le 3$$

y(1) = 17 y(3) = 43/3