Numerical Methods for Solving Differential Equations
(一階微分方程式的電腦數值解)

南台電機 趙春棠 整理

重點 1 ： 任何一個 N 階 微分方程式，都可以表示爲 N 個 1 階 微分方程式 的 聯立。 如此一來，電腦只要會解 1 階 微分方程式 ，電腦 就可以 幫我們解 任何一個 N 階 微分方程式 了。

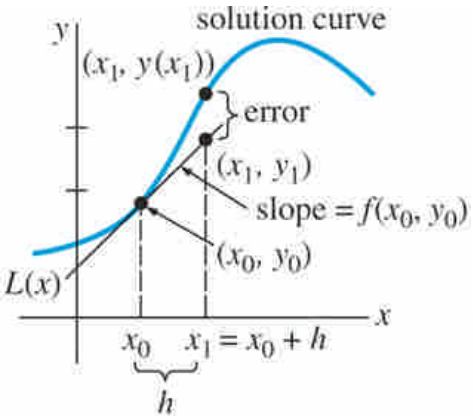
重點 2 ：利用電腦解 1 階 微分方程式，首先必須了解 Euler's Method (尤拉 法)。 尤拉法 非常簡單，例如已知初始值，先計算初始值該處的函數微分值，就可以估測出下一點 解函數 的位置了，再以下一點當做起點，則可估測下下一點解函數的位置 .....。 由於只利用到一點的函數微分值，故可視爲 Runge-Kutta-1方法。哈！由於只利用到一點的函數微分值，故誤差頗大喔！

重點 3 ：休恩法(Huen's Method) 改進了尤拉 法。休恩法 除了 採用 目前這一點的函數微分值，還採用了原本預估點位置的函數微分值喔，他最後真正是把兩個函數微分值取平均作爲預測斜率，來做估測的喔。由於它利用到兩點的函數微分值，故可視爲 Runge-Kutta-2方法。

重點 4 ：哈！如此說來，真正的 Runge-Kutta- 4法，可知是利用到四個點位置 (簡單來說，起點 1 個，中點取 2 個，終點取 1 個) 的函數微分值，給它們不同的權重，最後以算出的結果作爲預測斜率，來做估測的喔。效果非常準喔！

- Please briefly illustrate the Euler's Method for Solving Differential Equations.

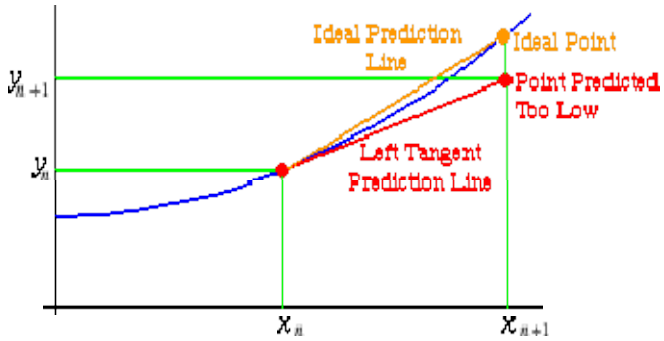
（http://calculuslab.deltacollege.edu/ODE/7-C-2/7-C-2-h.html , Home Page: http://calculuslab.deltacollege.edu/ ）



$$y_1 = y_0 + h f(x_0, y_0)$$

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

y0 -> y1　推廣爲　yn -> yn+1　(利用繪出三角形，很容易推導)



Euler's Method: The procedure of using successive "tangent lines".

$$y' = f(x,y) \qquad\qquad y_{n+1} = y_n + h \cdot f(x_n, y_n)$$

■ Consider the initial-value problem $y'+3y=5,\ \ y(0)=1$. Use Euler's method to obtain an approximation to y(1) using step size h=0.5

Sol. **(1)** $\left.\frac{dy}{dt}\right|_{(t=0,y(0))} = (5-3y)\Big|_{(t=0,y(0))} = (5-3\cdot 1) = 2$

故 $y(0.5) = y(0) + 0.5 \cdot \left.\frac{dy}{dt}\right|_{(t=0,y(0))} = 1 + 0.5 * 2 = 2$

【Actual Value: $y(0.5) = (-\frac{2}{3}e^{-3t} + \frac{5}{3})\Big|_{t=0.5} = 1.5179$】

(2) $\left.\frac{dy}{dt}\right|_{(t=0.5,y(0.5))} = (5-3y)\Big|_{(t=0.5,y(0.5))} = (5-3\cdot 2) = -1$

故 $y(1) = y(0.5) + 0.5 \cdot \left.\frac{dy}{dt}\right|_{(t=0.5,y(0.5))} = 2 + 0.5 * (-1) = 1.5$

【Actual Value: $y(1) = (-\frac{2}{3}e^{-3t} + \frac{5}{3})\Big|_{t=1} = 1.6335$】

■ 試利用 Euler's Method，撰寫 Matlab 程式解 $y'+3y=5,\ \ y(0)=2$（考慮 0<= t <= 4）。

euler1.m

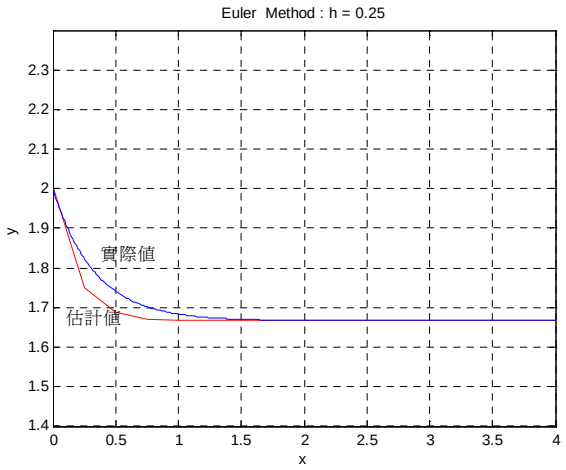
```
      clear;
% initial condition
      x0 = 0;                y0 = 2;
      xmax = 4;
      h = 0.25;
%  估計值計算
      n = (xmax-x0)/h;
for i=1:1:n+1
      if i == 1
          x(i) = x0;
          y(i) = y0;
      else
           x(i) = x0+(i-1)*h;
           % function      yd=f(x,y)
           y(i) = y(i-1) + h*yd(x(i-1),y(i-1));
      end
end
%  實際值
      x1 = x0:(xmax-x0)/300:xmax;
      y1 = ytrue(x1);
%
      figure;        plot(x, y, 'r-', x1, y1);
      grid on;
       axis([0 xmax    1.4    2.4]);
       xlabel('x');          ylabel('y');
      title('Euler    Method : h = 0.25');
      gtext('估計值');             gtext('實際值');
```

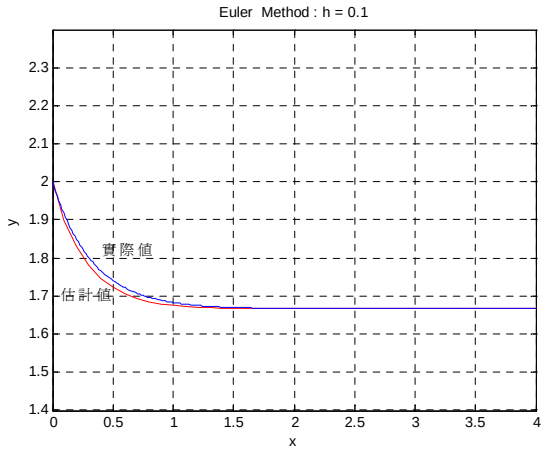
yd.m

```
function fxy=yd(x,y);
        fxy=-3*y+5;
%       fxy=-2*x.^3+12*x.^2-20*x+8.5;
%       fxy=sin(2*x)-tan(x)*y;
%       fxy=sin(x)./y;
%       fxy=2*x+y;
```

ytrue.m

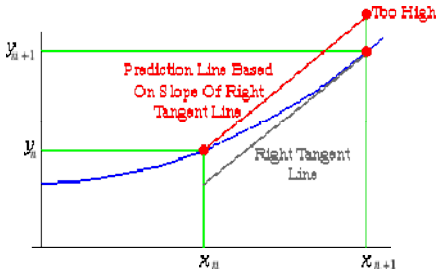
```
function fx=ytrue(x);
            fx= (1/3)*exp(-3*x)+(5/3);
%            fx=-0.5*x.^4+4*x.^3-10*x.^2+8.5*x+1;
%       fx=-2*cos(x).*cos(x)+3*cos(x);
%       fx=0.5*exp(1)*exp(-cos(x));
%       fx=0.75*x.^3-0.75*x.^-1;
```

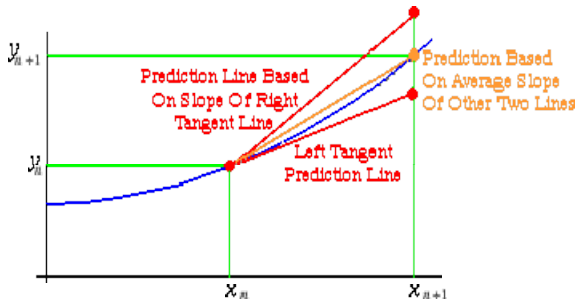




- Please briefly illustrate the **Heun's Method**（**Improved Euler's Method**） for Solving Differential Equations. （http://calculuslab.deltacollege.edu/ODE/7-C-2/7-C-2-h.html ）

Sol. Euler's Method has serious error accumulation for some problems! Heun's Method try to improve it!





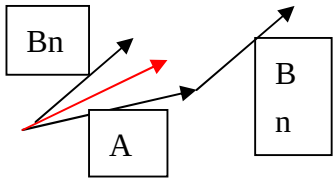
$$y' = f(x, y)$$

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

$$k_1 = f(x_n, y_n) \quad and \quad k_2 = f(x_{n+1}, y^*_{n+1}) = f(x_n + h, y_n + h \cdot k_1)$$

$$y_{n+1} = y_n + h \cdot \frac{k_1 + k_2}{2}$$

【圖解如下】



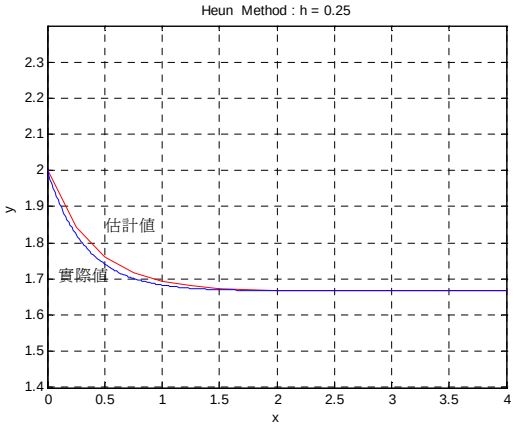
[中間紅線斜率爲 Average of An and Bn = (An+Bn)/2]

Note: 1. 本法又稱 RK2 method. (RK '2' : Second Order Method, 用了兩個斜率)

■ 試利用 Heun's Method，撰寫 Matlab 程式解 $y'+3y = 5$, $y(0) = 2$（考慮 0<= t <= 4）。

heun.m

```
%     heun
% initial condition
      clear;
      x0 = 0;            y0 = 2;
      xmax = 4;          h = 0.25;
%  估計值計算
      n = (xmax-x0)/h;
for j=1:1:n+1;
      x(j) = x0 + (j-1)*h;
end
       y(1) = y0;
for i=1:1:n
    % function
    y_star(i) = y(i) + h*yd(x(i), y(i));
    y1 = yd(x(i), y(i));
    y2 = yd(x(i+1),y_star(i));
    y(i+1) = y(i) + 0.5*h*(y1 + y2);
end
%  實際值（以下略）
```



■ Please describe "A Second-Order Runge-Kutta Method." 【Euler's Method is said to be a **first-order Runge-Kutta method**】（其實前述 **Heun's Method**，已算是 **RK2** 了。比較此法，與前述 **Heun's Method**，同樣是利用了兩個斜率，只是斜率的 權重 不同也！）

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

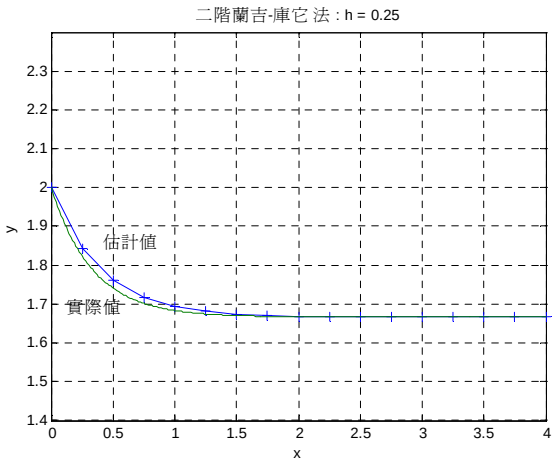
$$k_1 = f(x_n, y_n) \quad and \quad k_2 = f(x_n + 0.75 \cdot h,\ y_n + 0.75 \cdot h \cdot k_1)$$

$$y_{n+1} = y_n + h \cdot (\frac{1 \cdot k_1 + 2 \cdot k_2}{3})$$

■ 試利用 **A Second-Order Runge-Kutta Method**，撰寫 Matlab 程式解 $y'+3y = 5,\ \ y(0) = 2$（考慮 0<= t <= 4）。

rk2.m

```
%蘭吉-庫它 法 : 二階
  clear;
% initial condition
      x0 = 0;        y0 = 2;
      xmax = 4;     h = 0.25;
%  估計值計算
      n = (xmax-x0)/h;
for i=1:1:n+1
     x(i) = x0 + (i-1)*h;
end
   y(1) = y0;
for i=1:1:n;
   k1 = yd(x(i),y(i));
   k2 = yd(x(i) + 0.75*h,y(i) + 0.75*h*k1);
   y(i+1) = y(i) + h*(k1/3 + 2*k2/3);
end
%  實際值(實線) 【以下略】
```



■ Please describe **the Fourth-Order Runge-Kutta Method**.

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

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

$$k_2 = f(x_n + 0.5h, y_n + 0.5h \cdot k_1) \qquad k_3 = f(x_n + 0.5h, y_n + 0.5h \cdot k_2)$$

$$k_4 = f(x_n + h, y_n + h \cdot k_3)$$

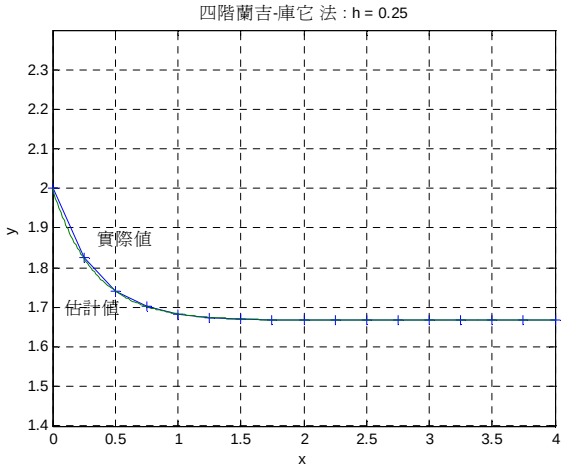
$$y_{n+1} = y_n + h \cdot \left(\frac{1 \cdot k_1 + 2 \cdot k_2 + 2 \cdot k_3 + 1 \cdot k_4}{6}\right)$$

【Note: 起點斜率取一次，中點斜率取 2 次，終點斜率取 1 次；各給不同的 weight。由於共取了四點的 斜率，所以 才稱 四階 蘭吉-庫它法 呀！ 這個 RK4 可是被公認爲非常精確的數值解喔！ ^_^】

■ 試利用 **A Fourth-Order Runge-Kutta Method（RK4）**，撰寫 Matlab 程式解 $y'+3y = 5, \quad y(0) = 2$（考慮 0<= t <= 4）。 【the RK3 method is omitted!】

rk4.m

```
%     蘭吉-庫它 法 : 四階
      clear;
% initial condition
      x0 = 0;       y0 = 2;
      xmax = 4;     h = 0.25;
%  估計值計算
      n = (xmax-x0)/h;
for i=1:1:n+1
    x(i) = x0 + (i-1)*h;
end
      y(1) = y0;
for i=1:1:n;
      k1 = yd(x(i), y(i));
      k2 = yd(x(i) + 0.5*h, y(i) + 0.5*h*k1);
      k3 = yd(x(i) + 0.5*h, y(i) + 0.5*h*k2);
      k4 = yd(x(i) + h, y(i) + h*k3);
      y(i+1) = y(i) + h*(k1 + 2*k2 + 2*k3 + k4)/6;
end
      x1 = x0:(xmax-x0)/300:xmax;              %  實際值(實線)
      y1 = ytrue(x1);
%
      plot(x,y,'-+',x1,y1,'-');                grid on;
      axis([0 xmax 1.4 2.4]);
      xlabel('x');          ylabel('y');
      title('四階蘭吉-庫它 法 : h = 0.25');
      gtext('估計值');          gtext('實際值');
```


四階蘭吉-庫它 法 : h = 0.25
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
y
實際值
估計值
0
0.5
1
1.5
2
2.5
3
3.5
4
x