Numerical Methods for Solving the Second-Order Differential Equations
(二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，都可以表示為 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，故可視為 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為預測斜率，來做估測的喔。由於它利用到兩點的函數微分值，故可視為 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為預測斜率，來做估測的喔。效果非常準喔！

■ 試利用 Euler’s Method，撰寫 Matlab 程式解聯立方程組（假定 step size h=0.25）

$$\begin{cases} y'_1 + 3y_1 = 5 & \quad y_1(0) = 1 \\ y'_2 - 3y_2 = 5 & \quad y_2(0) = 2 \end{cases}$$

【看吧！如以上重點 1 所說，一個二階系統變成兩個一階系統的聯立式了！ 】

eulerp_2A.m

```
%     歐拉  法 ：  解聯立方程組
         clear;
% initial condition
        x0 = 0;
        y10 = 1;
        y20 = 2;
        xmax = 2;
        h = 0.25;
% calculate
        n = (xmax-x0)/h;
for i=1:1:n+1
if i==1
      x(i) = x0;
      y1(i) = y10;
      y2(i) = y20;
else
        x(i) = x0+(i-1)*h;
        % function      yd=f(x,y)
        y1(i) = y1(i-1)+h*yd_1A(x(i-1),y1(i-1),y2(i-1));
        y2(i) = y2(i-1)+h*yd_2A(x(i-1),y1(i-1),y2(i-1));
end
end
        plotyy(x,y1,x,y2);grid
%      plotyy(x,y1,'-+',x,y2,'-*');grid
        gtext('y1');   gtext('y2');
        xlabel('x');   ylabel('y');
        title('Euler    Method   ：    解聯立方程組   '),
```

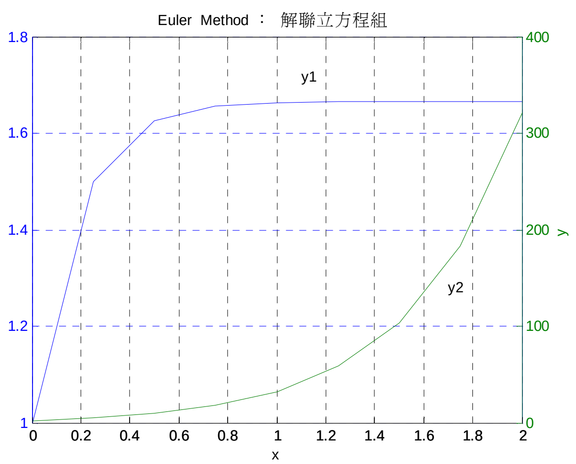
yd_1A.m

```
function fxy=yd_1(x,y1,y2);
        fxy = 5-3*y1;
```

yd_2A.m

```
function fxy = yd_2(x,y1,y2);
        fxy = 5+3*y2;
```

$y_1 = -\frac{2}{3}e^{-3t} + \frac{5}{3}$ （初值 1，以下藍線） $y_2 = \frac{11}{3}e^{3t} - \frac{5}{3}$ （初值 **2**，以下綠線【使用右邊綠色座標喔！】） 請於下圖比較之。(由 y1、y2 的刻度差距甚大，可知本系統有點 stiff 喔！)



■ 試利用 Euler's Method，撰寫 Matlab 程式解聯立方程組（假定 step size h=0.5）

$$\begin{cases} y'_1 = -0.5y_1 & y_1(0) = 4 \\ y'_2 = -0.1y_1 - 0.3y_2 + 4 & y_2(0) = 6 \end{cases}$$

eulerp_2.m

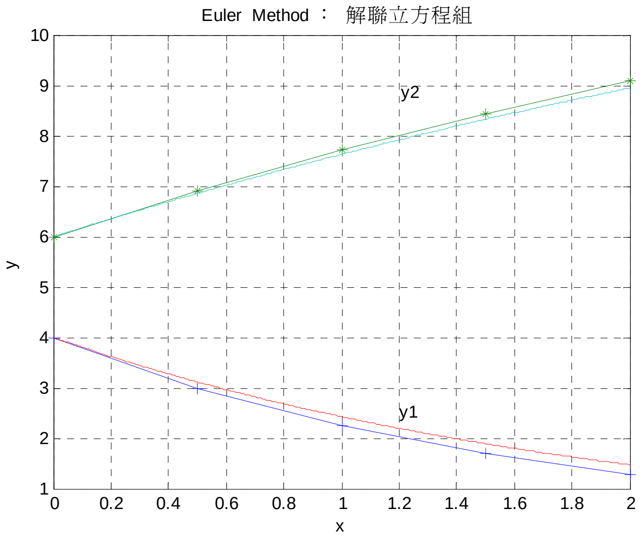
```
%     歐拉  法 :   解聯立方程組
      clear;
% initial condition
      x0 = 0;
      y10 = 4;
      y20 = 6;
      xmax = 2;
      h = 0.5;
% cal
      n = (xmax-x0)/h;
for i=1:1:n+1
if i==1
     x(i) = x0;
     y1(i) = y10;
     y2(i) = y20;
else
      x(i) = x0+(i-1)*h;
      % function     yd=f(x,y)
       y1(i) = y1(i-1)+h*yd_1(x(i-1),y1(i-1),y2(i-1));
      y2(i) = y2(i-1)+h*yd_2(x(i-1),y1(i-1),y2(i-1));
end
end
      x1 = x0:(xmax-x0)/300:xmax;
      y1true = 4*exp(-0.5*x1);
      y2true = 13.333-9.333*exp(-0.3*x1)+2*exp(-0.5*x1);
      plot(x,y1,'-+',x,y2,'-*',x1,y1true,x1,y2true,'-');grid
      gtext('y1');   gtext('y2');
      xlabel('x');   ylabel('y');
      title('Euler   Method  :   解聯立方程組 '),
```

yd_1.m

```
function fxy=yd_1(x,y1,y2);
     fxy = -0.5*y1;
```

yd_2.m

```
function fxy = yd_2(x,y1,y2);
     fxy = 4-0.3*y2 - 0.1*y1;
```



Note: Matlab 指令　　　>> dy=@(t,y)　[-0.5*y(1) ;　-0.1*y(1)-0.3*y(2)+4];

>> ode45(dy,[0,2], [4, 6])　　　　　% 繪出結果同上

■ $y'' = 0.5(x + y + y' + 2),\quad y(0) = 0,\ y'(0) = 0 \quad 0 \le x \le 1$　像這樣的 2 階 ODE，另可直接利用 Taylor Series Expansion method 求解喔！