Homework #5,　　　2021/06/17 11：59 pm

1. (25 %) 使用 classdef 方式重新定義上課投影片用@polynom 定義的類別，此類別名稱為 polynom，在 methods 區塊中要定義以下之函式：
   (a) constructor 函式，輸入多項式係數建立多項式物件 (例如：p = polynom([1 3 2 4])。
   (b) display 函式，顯示多項式。
   (c) 多項式的 加、減、乘、除 函式。
   (d) 計算多項式值的函式 polyval (例如：a = polyval(p, 2.0)，亦即在 x=2.0 處函式的值)。
   (e) plot 函式，畫出多項式在給定 x 的範圍內、f(x) vs. x 的圖形。

   完成類別定義後，重新做上課投影片中用@polynom 定義的範例，並畫出函式、一次微分與二次微分在 x=0 ~ 10 範圍內的圖形。

答：

polynom.m

```
classdef polynom
   properties
            c {mustBeNumeric}
   end
   methods
      function poly = polynom(vec)
       %POLYNOM Polynomial class constructor
       if isa(vec, 'polynom')
          poly = vec;
       else
          poly.c = vec(:).';
       end
      end
      function display(poly)
       % POLYNOM/DISPLAY Display of a polynom
       disp(' ');
       disp([inputname(1),' = '])
       disp(' ');
       disp(['     ', polyAsString(poly)])
       disp(' ');
      end
```

```
function s = polyAsString(poly)
% POLYNOM/POLYASSTRING String representation of a polynom
    degree=length(poly.c)-1;
    s = sprintf('%d*x^%d', poly.c(1), degree);
    for i=degree-1:-1:0
      coef = poly.c(degree-i+1);
      if coef >=0
        s=sprintf('%s + %d*x^%d', s, coef, i);
      else
        s=sprintf('%s - %d*x^%d', s, -coef, i);
      end
    end
end
function r = plus(p, q)
% POLYNOM/PLUS    Implement p + q for polynoms.
  p = polynom(p);
  q = polynom(q);
  k = length(q.c) - length(p.c);
  r = polynom([zeros(1,k) p.c] + [zeros(1,-k) q.c]);
end
function r = minus(p,q)
% POLYNOM/MINUS Implement p - q for polynoms.
  p = polynom(p);
  q = polynom(q);
  k = length(q.c) - length(p.c);
  r = polynom([zeros(1,k) p.c] - [zeros(1,-k) q.c]);
end
function r = mtimes(p, q)
% POLYNOM/MTIMES Implement p*q for polynoms.
  p = polynom(p);
  q = polynom(q);
  r = polynom(conv(p.c, q.c));
end
```

```
    function [q, r] = mrdivide(a, b)
    % POLYNOM/MRDIVIDE Implement a/b for polynoms.
      a = polynom(a);
      b = polynom(b);
      [q, r] = deconv(a.c, b.c);
      q = polynom(q);
      r = polynom(r);
    end
    function c = polyCoef(p)
      % POLYNOM/POLYCOEF Convert polynom object to coefficient vector.
      c = p.c;
    end
    function y = polyval(p, x)
    % POLYNOM/POLYVAL    POLYVAL(p, x) evaluates p at the points x.
      y = polyval(p.c, x);
    end
    function plot(p, range)
    % POLYNOM/PLOT    PLOT(p) plots the polynom p.
      if nargin<2
        range = max(abs(roots(p)))*[-1 1];
      end
      x = linspace(range(1), range(2));
      y = polyval(p, x);
      plot(x, y);
      title(polyAsString(p))
      grid on
    end
    function q = polyder(p)
    % POLYNOM/POLYDER POLYDER(p) is the derivative of the polynom p.
      q = polynom(polyder(p.c))
    end
  end
end
```

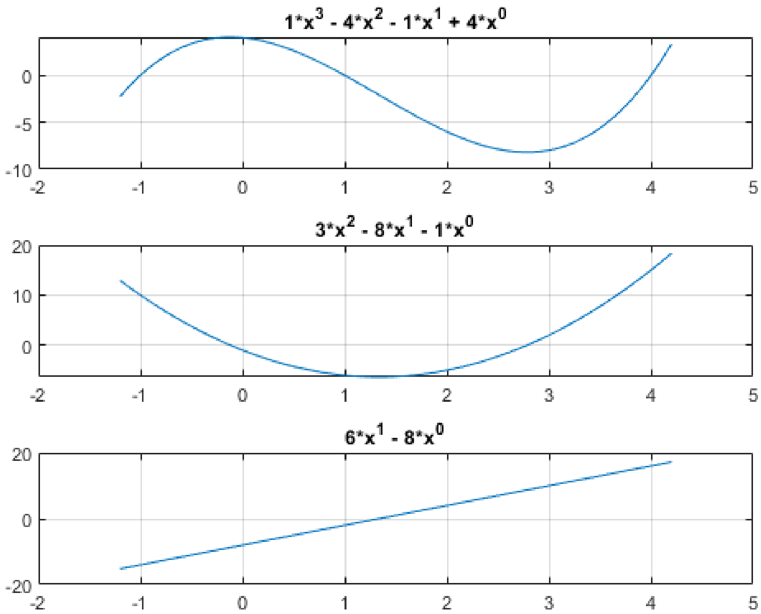
----

test_poly.m

```
% test 1
p = polynom([3 4 2 1])
```

```
% test 2
p = polynom([3 4 2 1]);
q = polynom([-1, 2]);
r = p + q
s = r + [2, 3]
% test 3
p = polynom([1, 1]);
q = polynom([1, 2]);
r = (p+1)*(q+2)
[a, b] = r/[1, 1]
% test 4
p = polynom([1 2 3]);
x = polyval(p, 1)
y = polyval(p, [1 2 3 4])
% test 5
p = polynom([1 -4 -1 4]);
range = [-1.2, 4.2];
subplot(3,1,1); plot(p, range);
p2 = polyder(p);
subplot(3,1,2); plot(p2, range);
p3 = polyder(p2);
subplot(3,1,3); plot(p3, range);
```

Results：



2. (20 %)

雷利分布。許多實際問題都會發現到另一種亂數分布——雷利分布。雷利分布的亂數值，可對常態分布下的兩個亂數平方和取平方根得到。換句話說，要產生雷利分布的亂數值 r，先要取得兩個常態分布下的亂數值（n_1 與 n_2），然後再依下式計算：

$$r = \sqrt{n_1^{\ 2} + n_2^{\ 2}} \tag{6.23}$$

(a) 產生一個函式 rayleigh(n,m)，可以傳回一個雷利分布的 n × m 亂數陣列，若只有提供一個引數給 [rayleigh(n)]，傳回一個雷利分布的 n × n 亂數陣列。請小心處理你函式的輸入引數個數，並為 MATLAB 說明系統編寫這函式適當的說明。

(b) 藉著產生 20,000 個雷利分布的亂數值，來測試你的函式，並繪製此分布的直方圖。這個分布看起來像什麼？

(c) 請計算這個雷利分布的平均值與標準差。

答：

random_rayleigh.m

```
function res = random_rayleigh(n,m)
% RANDOM_RAYLEIGH Return samples from a Rayleigh distribution.
% Function RANDOM_RAYLEIGH generates an array of Rayleigh-
% distributed random numbers. The usage is:
%
% random_rayleigh() -- Generate a single value
% random_rayleigh(n) -- Generate an n x n array
% random_rayleigh(n,m) -- Generate an n x m array
%
% Define variables:
% arr1 -- Normally-distributed array
% arr2 -- Normally-distributed array
% res -- Results
%
% Check for a legal number of input arguments.
msg = nargchk(0,2,nargin);
error(msg);
```

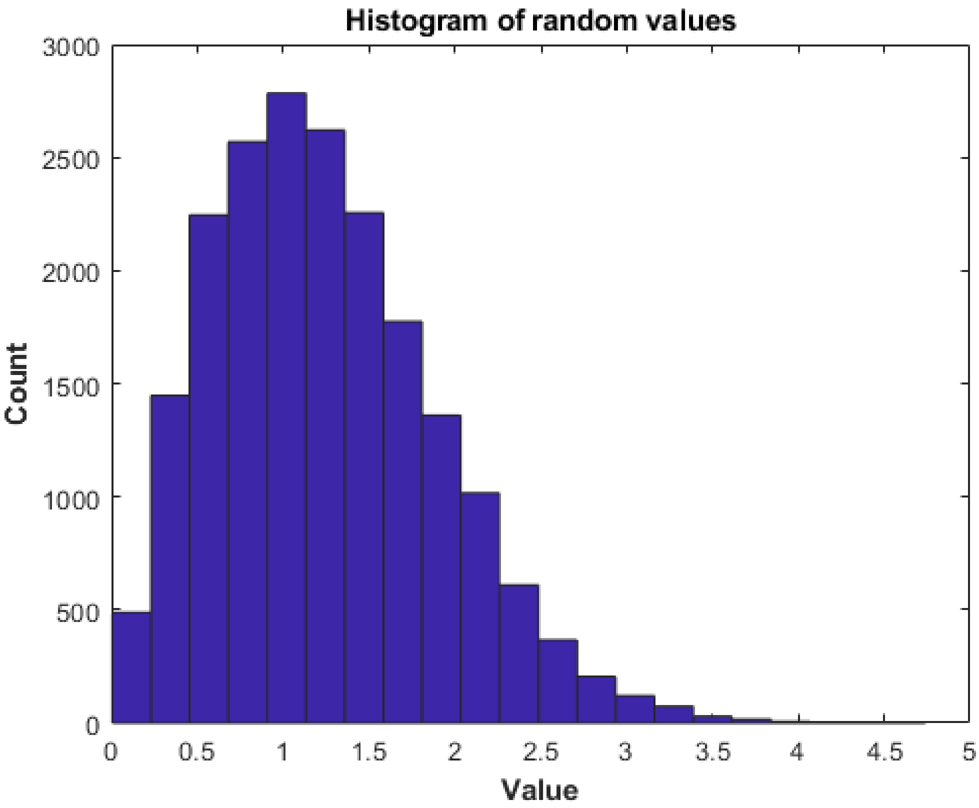
```
% If both arguments are missing, set to 1.
% If the m argument is missing, set it to n.
if nargin < 1
m = 1;
n = 1;
elseif nargin < 2
m = n;
end
% Calculate data
arr1 = randn(n,m);
arr2 = randn(n,m);
res = sqrt( arr1.^2 + arr2.^2 );
```

----

test_random_rayleigh.m

```
% Purpose:
% To test function random_rayleigh by getting 20,000
% values, calculating the mean and standard
% deviation, and plotting a histogram of the values.
%
% Define variables:
% ave -- Average (mean) of distribution
% dist -- Distribution
% sd -- Standard deviation of distribution
%
% Get 20,000 values
dist = random_rayleigh(1,20000);
% Calculate mean and standard deviation
ave = mean(dist);
sd = std(dist);
% Tell user
fprintf('Average = %.4f\n',ave);
fprintf('Std Dev = %.4f\n',sd);
% Create histogram
hist(dist,21);
title('\bfHistogram of random values');
xlabel('\bfValue');
ylabel('\bfCount');
```

Results：




```
>> test_random_rayleigh
Average = 1.2452
Std Dev = 0.6454
```

3. (15 %)

寫出一個程式，產生 3 個匿名函式表示 3 個函數 $f(x) = 10 \cos x$，$g(x) = 5 \sin x$，以及 $h(a,b) = \sqrt{a^2 + b^2}$，並在 $-10 \le x \le 10$，畫出 $h(f(x), g(x))$ 圖形。

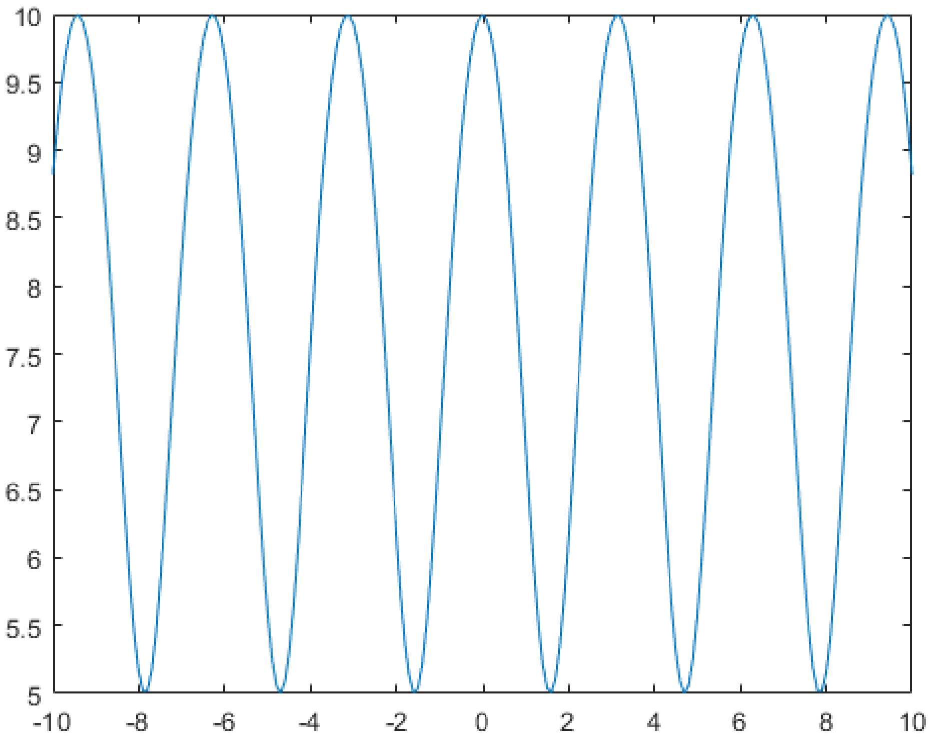
答：

test_anonymous.m

```
% Purpose:
% To create three anonymous functions, and then create a
% plot using them.
```

```
% Define variables:
% f -- Function handle
% g -- Function handle
% h -- Function handle
% x -- Input data samples
%
% Create anonymous functions
f = @ (x) 10 * cos(x);
g = @ (x) 5 * sin(x);
h = @ (a,b) sqrt(a.^2 + b.^2);
% Plot the functiion h(f(x),g(x))
x = -10:0.1:10;
plot(x,h(f(x),g(x)));
```

Results：



4. (20 %)

請寫出三個 MATLAB 函式，分別計算雙曲正弦、餘弦和正切函數：

$$\sinh(x) = \frac{e^x - e^{-x}}{2} \quad \cosh(x) = \frac{e^x + e^{-x}}{2} \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

使用你的函數畫出雙曲正弦、餘弦和正切函數圖形。

(請寫一個程序檔 script file(主程式)去呼叫這三個函數)

答：

sinh1.m

```
function out = sinh1(x)
% SINH1 Calculate hyperbolic sine function
% Function SINH1 calculates the hyperbolic sine function
% Define variables:
% x -- Input value
% Calculate value
out = (exp(x) - exp(-x))/2;
```

----

cosh1.m

```
function out = cosh1(x)
% COSH1 Calculate hyperbolic cosine function
% Function COSH1 calculates the hyperbolic cosine function
% Define variables:
% x -- Input value
% Calculate value
out = (exp(x) + exp(-x))/2;
```

----

tanh1.m

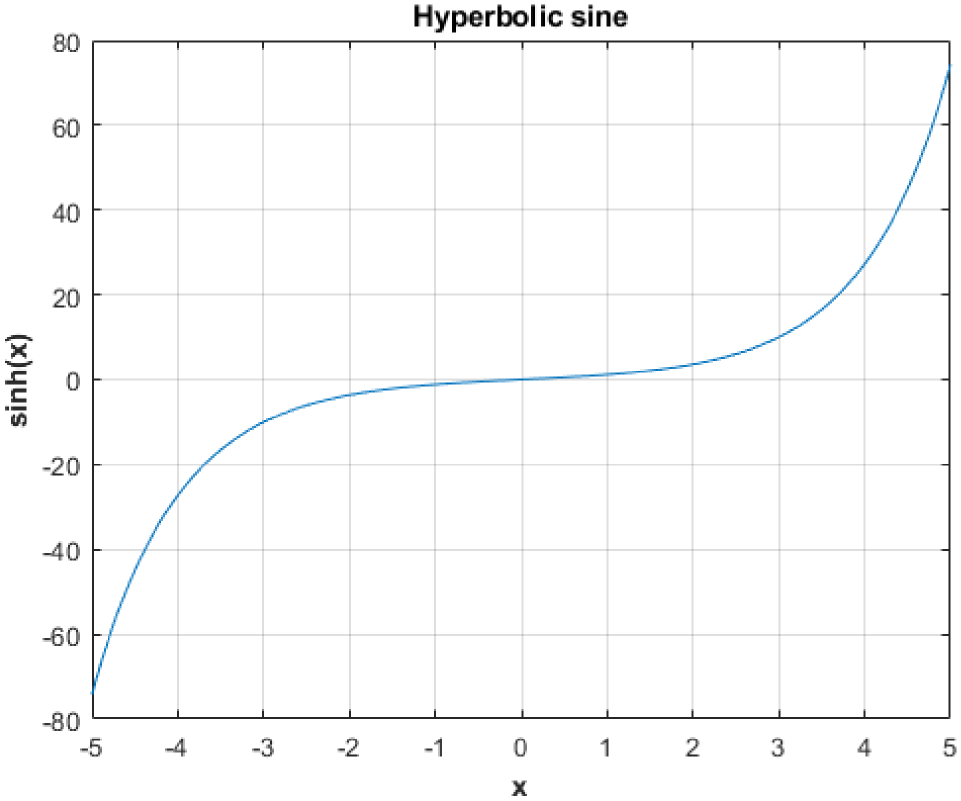
```
function out = tanh1(x)
% TANH1 Calculate hyperbolic tangent function
% Function TANH1 calculates the hyperbolic tangent function
% Define variables:
% x -- Input value
% Calculate value
out = (exp(x) - exp(-x)) ./ (exp(x) + exp(-x));
```

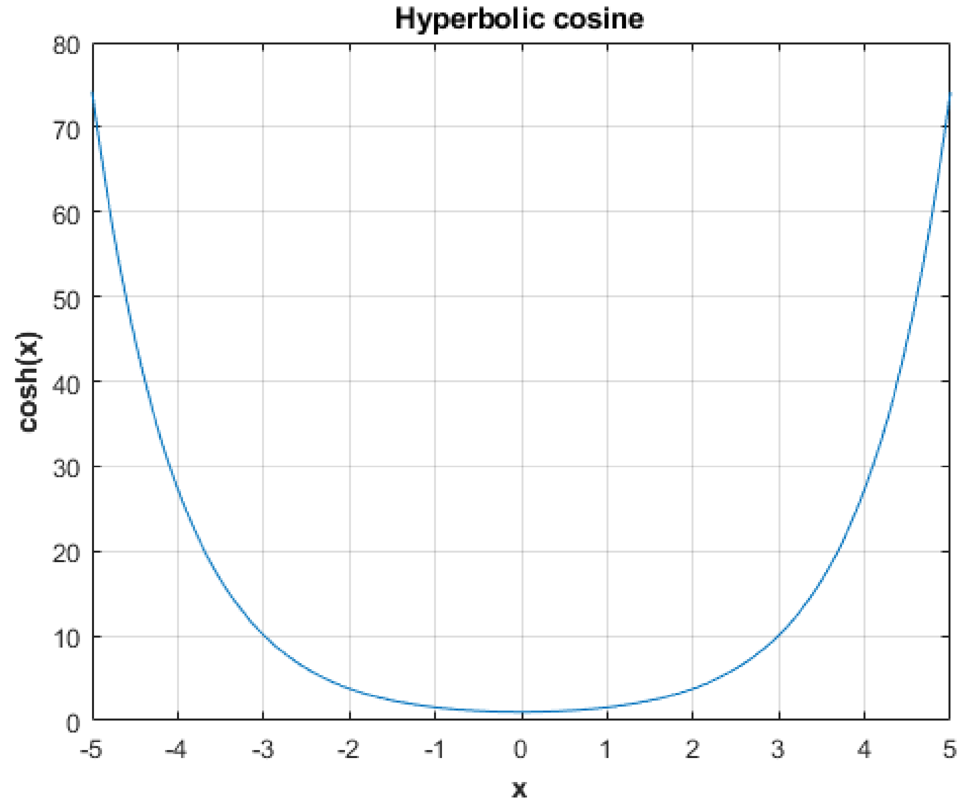
----

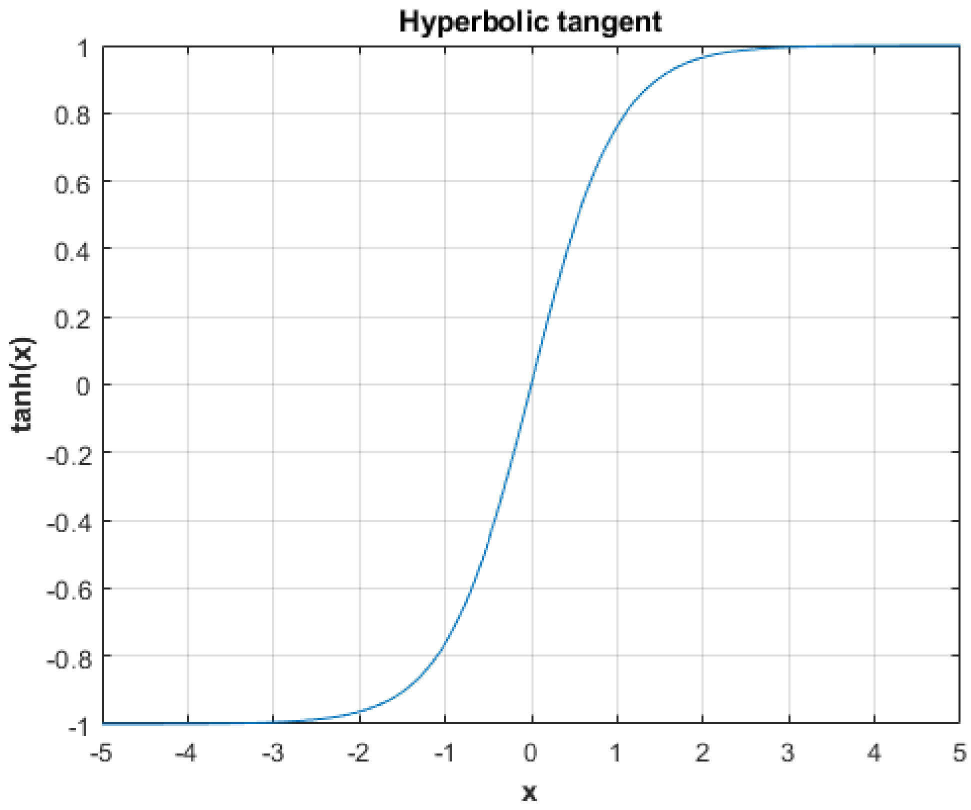
test_hyperbolic_functions.m

```
% Purpose:
% To plot the hyperbolic functions sinh, cosh, abd tanh.
% Define variables:
% out_cosh -- Hyperbolic cosine
% out_sinh -- Hyperbolic sine
% out_tanh -- Hyperbolic tangent
% Calculate results
x = -5:0.05:5;
out_sinh = sinh1(x);
out_cosh = cosh1(x);
out_tanh = tanh1(x);

% Display results
figure(1);
plot(x,out_sinh);
title('\bfHyperbolic sine');
xlabel('\bfx');
ylabel('\bfsinh(x)');
grid on;

figure(2);
plot(x,out_cosh);
title('\bfHyperbolic cosine');
xlabel('\bfx');
ylabel('\bfcosh(x)');
grid on;

figure(3);
plot(x,out_tanh);
title('\bfHyperbolic tangent');
xlabel('\bfx');
ylabel('\bftanh(x)');
grid on;
```

Results：


Hyperbolic sine
sinh(x)
80
60
40
20
0
-20
-40
-60
-80
-5
-4
-3
-2
-1
0
1
2
3
4
5
x


Hyperbolic cosine
cosh(x)
80
70
60
50
40
30
20
10
0
-5
-4
-3
-2
-1
0
1
2
3
4
5
x



5. (20 %) 使用 ode45 函式求解下列初始值常微分方程式系統在 t = 0 ~ 10 (時間間隔為 0.001)的數值解：

$$x' = -sx + sy$$
$$y' = -xz + rx - y \qquad (x' = \frac{dx}{dt} 、 y' = \frac{dy}{dt} 、 z' = \frac{dz}{dt})$$
$$z' = xy - bz$$

其中 $s = 10, r = 28, b = 8/3$。考慮兩個差異極小的初始條件：

(1) $x(0) = 5.0, y(0) = 5.0, z(0) = 5.0$,

以及 (2) $x(0) = 5.00001, y(0) = 5.0, z(0) = 5.0$ 。

請用 plot 指令畫出此系統在兩個初始條件下的解在 xz plane 上隨時間的軌跡圖(即 x(t) vs. z(t)圖，分別用藍色虛線與紅色點線代表，並加上圖說明 legend 於右下角、x 軸 lable 為‘X’，y 軸 lable 為‘Y’，x 軸範圍為-25 ~ 25，y 範圍為 0 ~ 50)，並比較它們的差異。(此初始值常微分方程式系統為有名的勞倫茲方程式，在本題給定的參數下，解的軌跡為著名的勞倫茲吸子。)

答：

fun5.m

```
function yprime = fun5(t,y)
s=10;
r=28;
b=8.0/3.0;
yprime = [-1.0*s*y(1)+s*y(2)
          -1.0*y(1)*y(3)+r*y(1)-y(2)
          y(1)*y(2)-b*y(3)];
```

----

hw5_ode45.m

```
%   Purpose:
%       This program solves a differential equation of the
%       form dy/dt + 2 * y = 0, with the initial condition
%       y(0) = 1.
%
% Define variables:
%     odefun_handle -- Handle to function that defines the derivative
%     tspan          -- Duration to solve equation for
%     yo             -- Initial condition for equation
%     t              -- Array of solution times
%     y              -- Array of solution values

% Get a handle to the function that defines the
% derivative.
odefun_handle = @fun4;

% Solve the equation over the period 0 to 5 seconds
tspan = 0:0.001:20;

% Set the initial conditions
y01 = [5 5 5];
y02 = [5.00001 5 5];

% Call the differential equation solver.
%[t,y] = ode45(odefun_handle,tspan,y0);
[t1,y1] = ode45('fun5',tspan,y01);
[t2,y2] = ode45('fun5',tspan,y02);
```

```
% Plot the result
plot(y1(:,1),y1(:,3),'b--','LineWidth',1);
grid on;
hold on
plot(y2(:,1),y2(:,3),'r:','LineWidth',1);
xlabel('X');
ylabel('Z');
set(gca,'XLim',[-25 25],'YLim',[0 50])
legend({'IC1','IC2'},'Location','southeast')
```

Results：