

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

#### polynom.m Gist

```
classdef polynom
    properties
        coe
    end
    methods
        function poly = polynom(vec)
            if isa(vec,'polynom')
                poly = vec;
            else
                poly.coe = vec(:).';
            end
        end

        function str = poly_as_str(poly)
            deg = length(poly.coe)-1;
            str = sprintf('%d*x^%d',poly.coe(1),deg);
            for c = deg-1:-1:0
                coef = poly.coe(deg-c+1);
                if coef >= 0
                    str = sprintf('%s + %d*x^%d',str,coef,c);
                else
                    str = sprintf('%s - %d*x^%d',str,-coef,c);
                end
            end
        end
    end
end
```

```

function display(poly)
    disp([newline,inputname(1),'=']);
    disp([newline, ' ',poly_as_str(poly),newline]);
end

function r = plus(a, b)
    a = polynom(a);
    b = polynom(b);
    k = length(a.coe)-length(b.coe);
    r = polynom([zeros(1,k) a.coe] + [zeros(1,-k) b.coe]);
end

function r = minus(a, b)
    if isa(b,'polynom')
        b.coe = -1*b.coe;
    else
        b = -1*b;
    end
    r = a+b;
end

function r = mtimes(a, b)
    a = polynom(a);
    b = polynom(b);
    r = polynom(conv(a.coe, b.coe));
end

function [q, r] = mrdivide(a, b)
    a = polynom(a);
    b = polynom(b);
    [q, r] = deconv(a.coe, b.coe);
    q = polynom(q);
    r = polynom(r);
end

function y = polyval(p, x)
    y = polyval(p.coe, x);
end

function y = polyder(p)
    y = polynom(polyder(p.coe));
end

function y = polyint(p)
    y = polynom(polyint(p.coe));
end

```

```

end
function r = roots(p)
    r = roots(p.coe);
end
function plot(p, range)
    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(poly_as_str(p))
    grid on
end
function c = polyCoef(p)
    c = p.coe;
end
end
end

```

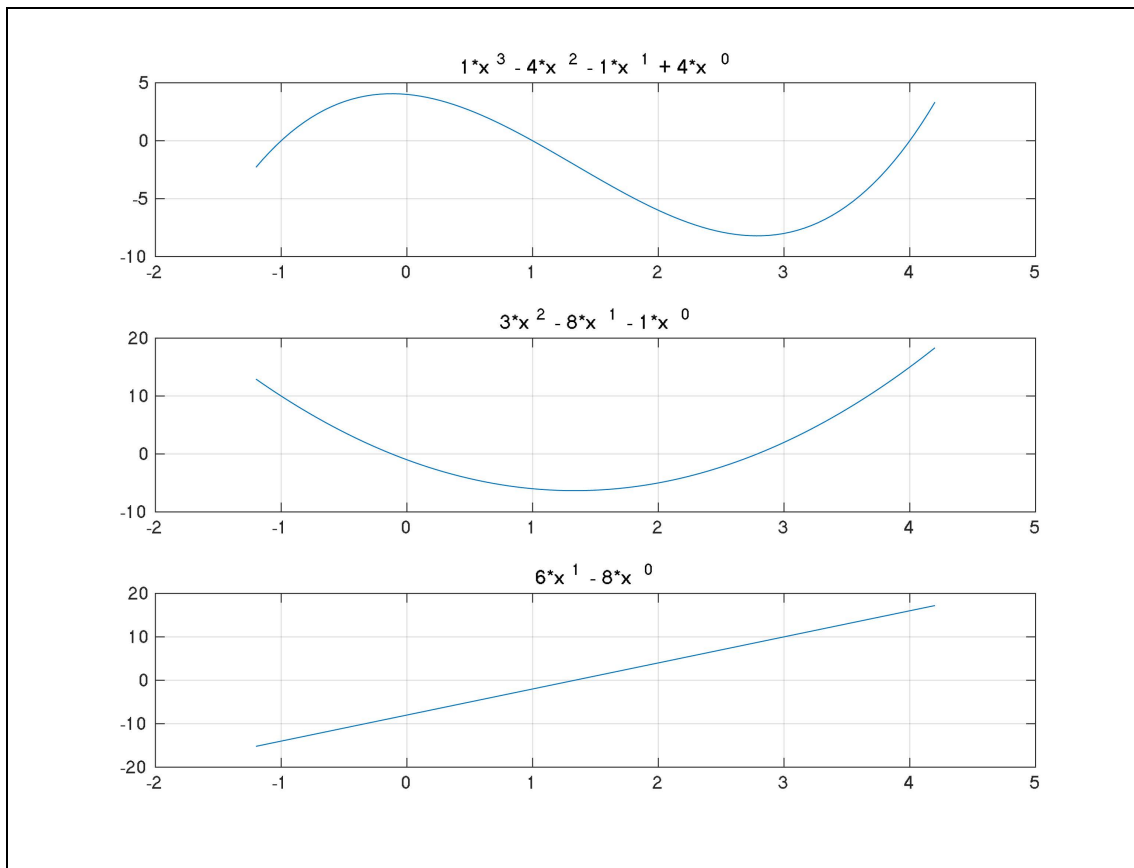
hw5\_1.m

```

p = polynom([1 -4 -1 4]);
range = [0, 10];
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);

```

Figure



## 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：

$$r = \sqrt{n_1^2 + n_2^2} \quad (6.23)$$

- 產生一個函式 `rayleigh(n,m)`，可以傳回一個雷利分布的  $n \times m$  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  $n \times n$  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。
- 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？
- 請計算這個雷利分布的平均值與標準差。

## rayleigh.m

```
function r = rayleigh(n,m)
%Rayleigh distribution Random arrays
% R = rayleigh(n) returns an array of n*n rayleigh random numbers
% R = rayleigh(n,m) returns an array of n*m rayleigh random numbers
if nargin < 1
    error(message('stats:raylrnd:TooFewInputs'));
elseif nargin == 1
    m = n;
elseif nargin > 2
    error(message('stats:raylrnd:TooMuchInputs'));
end
% Generate random values and take their sqrt.
r = sqrt(randn([n m]).^2 + randn([n m]).^2);
end
```

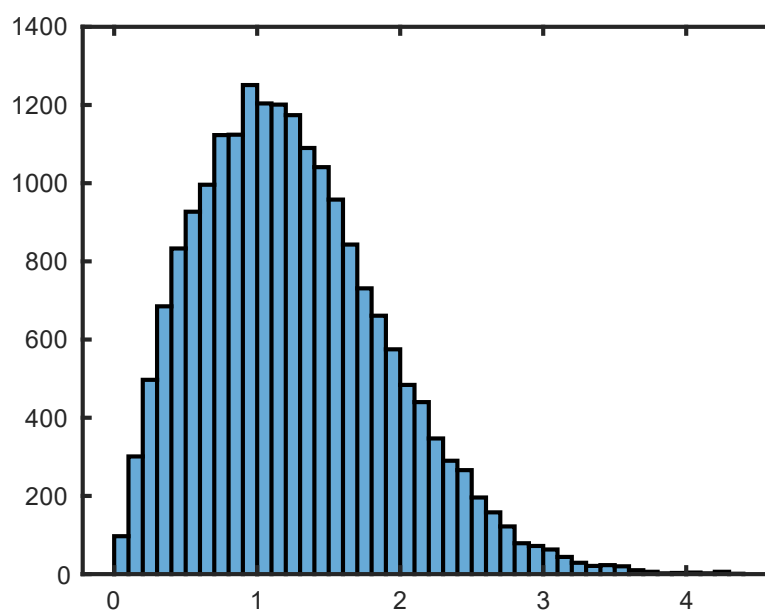
## hw5\_2.m

```
r = rayleigh(1,20000);
histogram(r)
fprintf("rayleigh avg = %f\n",mean(r));
fprintf("rayleigh std = %f\n",std(r));
```

## Output:

```
rayleigh avg = 1.252460
rayleigh std = 0.655801
```

## Figure



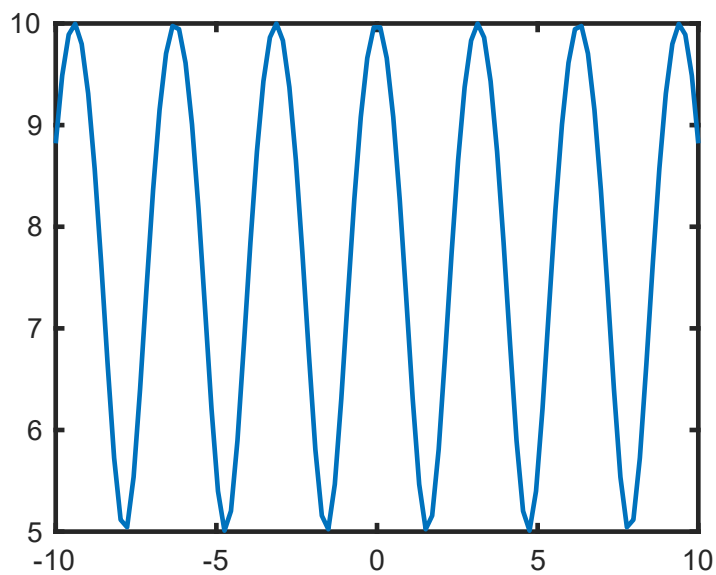
3. (15 %)

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  $-10 \leq x \leq 10$ ，畫出  $h(f(x), g(x))$  圖形。

hw5\_3.m

```
f = @(x) 10*cos(x);  
g = @(x) 5*sin(x);  
h = @(a,b) sqrt(a.^2+b.^2);  
range = linspace(-10,10,100);  
plot(range,h(f(range),g(range)))
```

Figure



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：

請寫一個程序檔

script file(主程式)

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

去呼叫這三個函數 使用你的函數畫出雙曲正弦、餘弦和正切函數圖形。

sinh.m

```
function r = sinh(x)
    r = (exp(x)-exp(-x))/2;
end
```

cosh.m

```
function r = cosh(x)
    r = (exp(x)+exp(-x))/2;
end
```

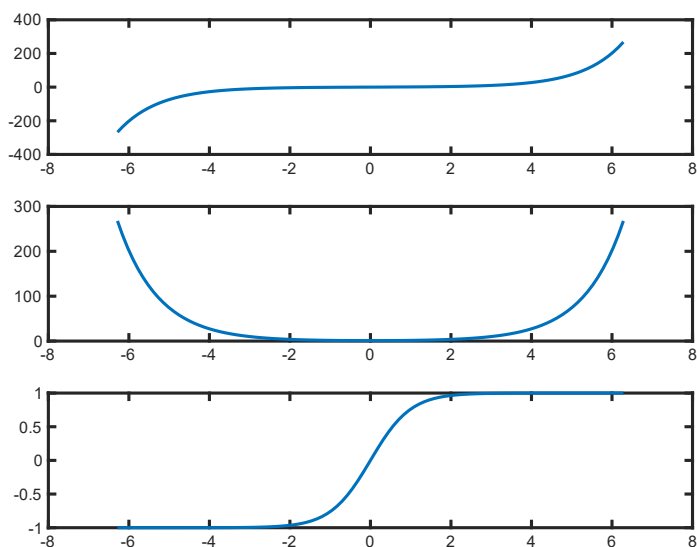
tanh.m

```
function r = tanh(x)
    r = sinh(x)/cosh(x);
end
```

hw5\_4.m

```
range = linspace(-2*pi,2*pi,100)
subplot(3,1,1)
plot(range,sinh(range));
subplot(3,1,2)
plot(range,cosh(range));
subplot(3,1,3)
plot(range,tanh(range));
```

Figure



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

$$x' = -sx + sy$$

$$y' = -xz + rx - y \quad \left( x' = \frac{dx}{dt}, y' = \frac{dy}{dt}, z' = \frac{dz}{dt} \right)$$

$$z' = xy - bz$$

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 label 為 'X', y 軸 label 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。)

func.m

```
function f = func(t,x)%x(1)->x; x(2)->y; x(3)->z
    s = 10;r = 28;b = 8/3;
    f(1) = -s*x(1)+s*x(2);
    f(2) = -x(1)*x(3)+r*x(1)-x(2);
    f(3) = x(1)*x(2)-b*x(3);
    f = f(:);
end
```

hw5\_5.m

```
[t1,x1] = ode45(@func,[0:0.001:10],[5;5;5]);
[t2,x2] = ode45(@func,[0:0.001:10],[5.00001;5;5]);
plot(x1(:,1),x1(:,3),'--b')
hold on
plot(x2(:,1),x2(:,3),':r')
hold off
xlabel('X')
xlim([-25,25])
ylabel('Y')
ylim([0,50])
legend({'x init 5','x init 5.00001'},'Location','southeast')
```

Figure



