

Chapter 3

二維繪圖的進階功能

Hung-Yuan Fan (范洪源)

Department of Mathematics,
National Taiwan Normal University, Taiwan

Spring 2020



- L1 二維繪圖額外的繪圖功能
- L2 極座標圖
- L3 註解並儲存圖形
- L4 額外的二維圖形類別
- L5 利用 `plot` 函式對二維陣列繪圖



Lecture 1

二維繪圖額外的繪圖功能



對數座標的二維圖形

在 x 軸及 y 軸上，線性座標和對數座標將產生四種可能的組合：

- 1 `plot` 函式將 xy 資料都畫在線性軸上。

表 5.1.1 `plot` 函數的使用

| 函 數 | 說 明 |
|------------------------|--|
| <code>plot(x,y)</code> | 以 x 為資料點的橫座標所組成的向量， y 為縱座標所組成的向量，描點繪出 (x,y) 的曲線圖 |
| <code>plot(y)</code> | x 的間距為 1，描點繪出 (x,y) 的曲線圖 |



對數座標的二維圖形

在 x 軸及 y 軸上，線性座標和對數座標將產生四種可能的組合：

- 1 `plot` 函式將 xy 資料都畫在線性軸上。

表 5.1.1 `plot` 函數的使用

| 函 數 | 說 明 |
|------------------------|--|
| <code>plot(x,y)</code> | 以 x 為資料點的橫座標所組成的向量， y 為縱座標所組成的向量，描點繪出 (x,y) 的曲線圖 |
| <code>plot(y)</code> | x 的間距為 1，描點繪出 (x,y) 的曲線圖 |

- 2 `semilogx`、`semilogy` 和 `loglog` 函式的語法說明如下：

表 7.1.2 對數繪圖函數的使用

| 函 數 | 說 明 |
|----------------------------|----------------------------------|
| <code>semilogx(x,y)</code> | x 軸為對數座標，繪出 $x-y$ 的對數圖 |
| <code>semilogy(x,y)</code> | y 軸為對數座標，繪出 $x-y$ 的對數圖 |
| <code>loglog(x,y)</code> | x 軸與 y 軸皆為對數座標，繪出 $x-y$ 的對數圖 |



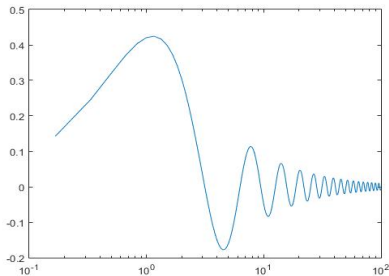
函式 semilogx 的範例

```
x = linspace(0,100,600);  
semilogx(x,sin(x) ./ (x+1));
```



函式 semilogx 的範例

```
x = linspace(0,100,600);  
semilogx(x,sin(x) ./ (x+1));
```



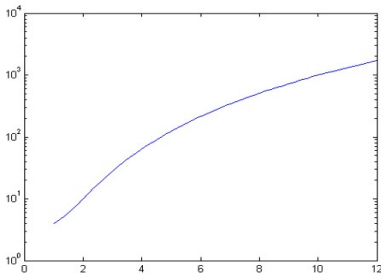
函式 semilogy 的範例

```
x = 1:0.2:12;  
y = x.^3 - x + 4;  
semilogy(x,y);
```



函式 semilogy 的範例

```
x = 1:0.2:12;  
y = x.^3 - x + 4;  
semilogy(x,y);
```



如何在閉區間 $I = [\text{start}, \text{end}]$ 上建立等間距的資料點？

① **linspace**: 在陣列點之間建立**線性**的間隔。

- **`linspace(start,end)`**: 在 I 上產生 **100** 點等間距的陣列。
- **`linspace(start,end,n)`**: 在 I 上產生 **n** 點等間距的陣列。



如何在閉區間 $I = [\text{start}, \text{end}]$ 上建立等間距的資料點？

① **linspace**: 在陣列點之間建立**線性**的間隔。

- `linspace(start,end)`: 在 I 上產生 **100** 點等間距的陣列。
- `linspace(start,end,n)`: 在 I 上產生 **n** 點等間距的陣列。

② **logspace**: 在陣列點之間建立**對數**的間隔。

- 函式的形式如下:

```
y = logspace(se,ee); % 50 點對數尺度上等間距的陣列  
y = logspace(se,ee,n);
```

- `se` 是起始值 `start` 以 10 為底的次方 (或是指數)。
- `ee` 是終了值 `end` 以 10 為底的次方 (或是指數)。
- `n` 是所要產生陣列的**點數**。



```
>> linspace(1,10,10)
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```



```
>> linspace(1,10,10)
```

```
ans =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>> logspace(0,1,10) % 因為  $10^0 = 1$  且  $10^1 = 10$ 
```

```
ans =
```

```
1.0000 1.2915 1.6681 2.1544 2.7826 3.5938  
4.6416 5.9948 7.7426 10.0000
```





良好的程式設計

如果繪圖數據的大小範圍涵蓋幾個級數的話，使用對數圖形來正確地呈現數據。如果繪圖數據的大小範圍只有一個級數或更小的話，那就使用線性圖形。



程式設計的陷阱

不要嘗試將負數資料畫在對數圖形上，這些資料會被忽略。



雙 y 軸繪圖 (1/2)

表 7.1.3 `plotyy()` 函數的使用

| 函 數 | 說 明 |
|---|--|
| <code>plotyy(x_1, y_1, x_2, y_2)</code> | 以圖形左邊的刻度當成 x_1 - y_1 資料點的 y 軸，以圖形右邊的刻度當成 x_2 - y_2 資料點的 y 軸，繪出雙 y 軸圖 |



雙 y 軸繪圖 (1/2)

表 7.1.3 `plotyy()` 函數的使用

| 函 數 | 說 明 |
|--|--|
| <code>plotyy(x₁,y₁,x₂,y₂)</code> | 以圖形左邊的刻度當成 x_1 - y_1 資料點的 y 軸，以圖形右邊的刻度當成 x_2 - y_2 資料點的 y 軸，繪出雙 y 軸圖 |

函式 `plotyy` 的範例

```
x = linspace(0,6,120);  
figure(1);  
plot(x,sqrt(x)+sin(6*x),x,exp(x));  
figure(2);  
plotyy(x,sqrt(x)+sin(6*x),x,exp(x));
```



雙 y 軸繪圖 (2/2)

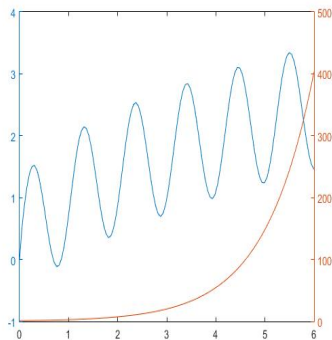
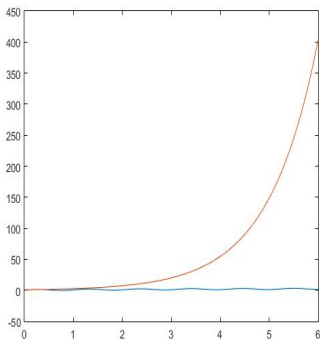


Figure: 函式 `plot`(左) 和 `plotyy`(右) 的繪圖結果



控制 x 、 y 軸的繪圖範圍

| 指令 | 說明 |
|--|---|
| <code>v = axis;</code> | 這個函式傳回四個元素的列向量 <code>[xmin xmax ymin ymax]</code> ，分別代表目前圖形的顯示範圍。 |
| <code>axis ([xmin xmax ymin ymax]);</code> | 這個函式設定面圖形的 x 和 y 軸顯示範圍。 |
| <code>axis equal</code> | 這個指令設定兩座標軸的刻度間距相等。 |
| <code>axis square</code> | 這個指令使得現在的座標圈變成正方形。 |
| <code>axis normal</code> | 這個指令取消 <code>axis equal</code> 及 <code>axis square</code> 產生的效果。 |
| <code>axis off</code> | 這指令關掉所有的座標軸名稱、刻度、及背景。 |
| <code>axis on</code> | 這指令開啟所有的座標軸名稱、刻度、及背景（預設值） |



函式 axis 的範例

```
x = linspace(0,10,64);
```

```
y = x.*cos(4*x)./12;
```

```
figure(1);
```

```
plot(x,y,'-ro');
```

```
figure(2);
```

```
plot(x,y,'-ro');
```

```
axis([0,6,-0.6,0.6]);
```

% 只觀察在 $0 \leq x \leq 6$ 和 $-0.6 \leq y \leq 0.6$ 範圍內的函數圖形



使用 `axis` 指令的結果

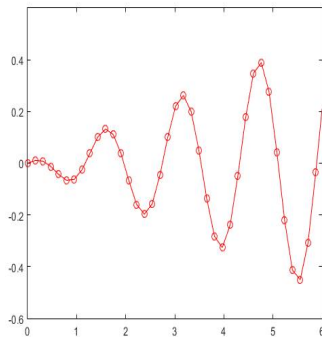
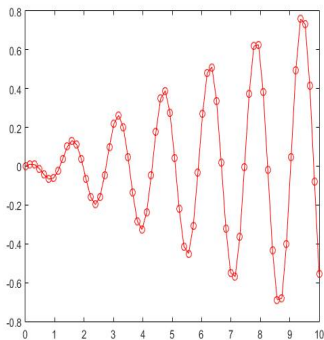


Figure: 原函數圖形 (左) 和使用 `axis`(右) 的繪圖結果



在相同軸上繪製多個圖形

- 正常的情況下，每次執行 `plot` 指令時，都會產生一個新的圖形，同時也會遺失之前顯示在圖上的繪圖資料。



在相同軸上繪製多個圖形

- 正常的情況下，每次執行 `plot` 指令時，都會產生一個新的圖形，同時也會遺失之前顯示在圖上的繪圖資料。
- 使用 `hold on` 指令之後，所有其後所產生的圖形，都可以疊在之前的圖形上。
- 使用 `hold off` 指令，可以切換圖形視窗回到預設的狀態。

表 5.2.4 設定圖形產生的方式

| 指令 | 說明 |
|-------------------|---|
| <code>hold</code> | 設定 <code>hold</code> 為 <code>on</code> 時，則新產生的圖形會疊加在原有圖形的上面，若是設定 <code>off</code> ，則原有的圖形會被新產生的圖形覆蓋掉。 |



使用 hold on/off 指令的範例

```
x = linspace(0,2*pi,36);  
y1 = sqrt(x).*sin(2*x);  
y2 = sqrt(x).*cos(2*x);  
plot(x,y1,'-rs');  
hold on;  
plot(x,y2,'-bo');  
hold off;
```



使用 hold on/off 指令的結果

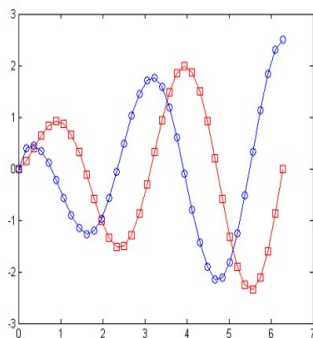
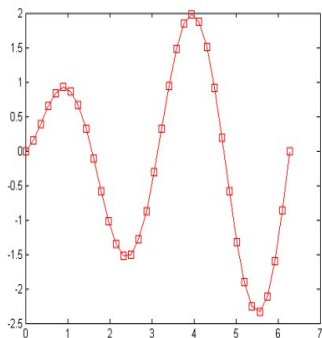


Figure: 紅色 x-y1 圖形 (左) 和使用 hold on 疊加藍色 x-y2 曲線 (右)



表 5.2.5 設定圖形產生的方式

| 函 數 | 說 明 |
|--------------------|--|
| figure | 建立一個新的繪圖視窗，視窗的標題為 Matlab 自動設定 |
| figure(<i>n</i>) | 建立一個新的繪圖視窗，視窗的標題為 Figure <i>n</i> 。若 Figure <i>n</i> 為已經存在的視窗，則 figure(<i>n</i>) 會把此視窗變成作用中視窗 |



函式 figure 的撰寫格式

⋮

`figure(1);`

第 1 個圖形視窗的繪圖程式碼

`figure(2);`

第 2 個圖形視窗的繪圖程式碼

⋮

`figure(n);`

第 n 個圖形視窗的繪圖程式碼

⋮



表 5.2.6 subplot() 函數的用法

| 函 數 | 說 明 |
|---|--|
| <code>subplot(m,n,p)</code> | 把繪圖視窗分成 $m \times n$ 個區域，並在第 p 個位置建立一個子繪圖區。位置 p 的計算方式是由左而右，由上而下來排列 |
| <code>subplot(m,n,p, 'replace')</code> | 於第 p 個位置建立一個子繪圖區，若此繪圖區內已有其它圖形存在，則新繪的圖會取代掉原有的圖 |



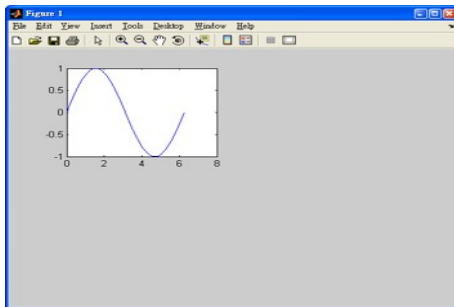
函式 subplot 的範例 (1/4)

```
>> x = linspace(0,2*pi,50);  
>> subplot(2,2,1); plot(x,sin(x));
```



函式 subplot 的範例 (1/4)

```
>> x = linspace(0,2*pi,50);  
>> subplot(2,2,1); plot(x,sin(x));
```



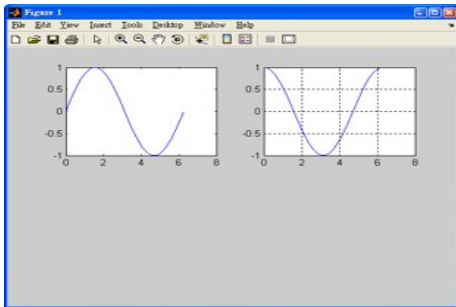
函式 subplot 的範例 (2/4)

```
>> subplot(2,2,2); plot(x,cos(x));  
>> grid on;
```



函式 subplot 的範例 (2/4)

```
>> subplot(2,2,2); plot(x,cos(x));  
>> grid on;
```



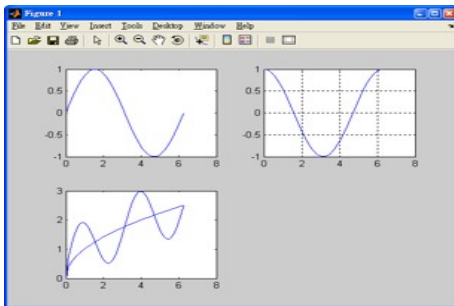
函式 subplot 的範例 (3/4)

```
>> subplot(2,2,3); plot(x,sqrt(x));  
>> hold on;  
>> plot(x,sqrt(x)+sin(2*x));  
>>hold off;
```



函式 subplot 的範例 (3/4)

```
>> subplot(2,2,3); plot(x,sqrt(x));  
>> hold on;  
>> plot(x,sqrt(x)+sin(2*x));  
>>hold off;
```



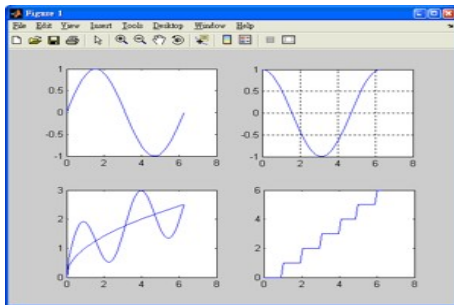
函式 subplot 的範例 (4/4)

```
>> subplot(2,2,4); plot(x,floor(x));
```



函式 subplot 的範例 (4/4)

```
>> subplot(2,2,4); plot(x,floor(x));
```



繪製線型的進階控制

使用 `plot` 函式繪製二維圖形的線型控制:

```
plot(x,y, 'Property',value,...)
```

| Property | 說明 |
|-----------------|--------------------|
| LineWidth | 設定每條線的寬度，以點為單位。 |
| MarkerEdgeColor | 設定標記顏色，或填滿標記的邊緣顏色。 |
| MarkerFaceColor | 設定填滿標記的表面顏色。 |
| MarkerSize | 設定標記的大小，以點為單位。 |

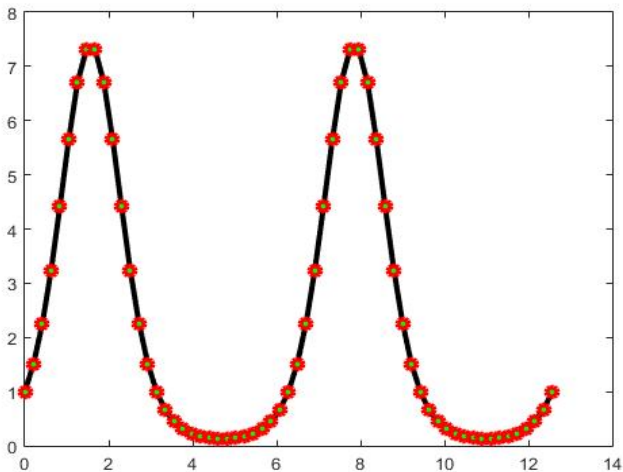


線型控制的範例

```
x = 0:pi/15:4*pi;  
y = exp(2*sin(x));  
plot(x,y,'-ko','LineWidth',3.0,'MarkerSize',6,...  
      'MarkerEdgeColor','r','MarkerFaceColor','g')
```



範例的繪圖結果 (承上頁)



圖形視窗內的文字字串 (標題、座標軸名等) 的格式，可以使用**流線修飾符號 (stream modifiers)** 修改顯示文字的字體格式。

- `\bf`: 粗體文字。
- `\it`: 斜體文字。
- `\rm`: 取消流線修飾符號，恢復預設的文字字體。
- `\fontname{ fontname }`: 設定使用字型的名稱。
- `\fontsize{ fontsize }`: 設定使用字體的大小。
- `_ {xxx}`: 下標符號，例如: `x_{3}` 會顯示為 x_3 。
- `^ {xxx}`: 上標符號，例如: `x^{3}` 會顯示為 x^3 。



常用的希臘文字和數學符號

| 字元序列 | 符號 | 字元序列 | 符號 | 字元序列 | 符號 |
|-----------------------|------------|----------------------|-----------|------------------------------|-------------------|
| <code>\alpha</code> | α | | | <code>\int</code> | \int |
| <code>\beta</code> | β | | | <code>\cong</code> | \cong |
| <code>\gamma</code> | γ | <code>\Gamma</code> | Γ | <code>\sim</code> | \sim |
| <code>\delta</code> | δ | <code>\Delta</code> | Δ | <code>\infty</code> | ∞ |
| <code>\epsilon</code> | ϵ | | | <code>\pm</code> | \pm |
| <code>\eta</code> | η | | | <code>\leq</code> | \leq |
| <code>\theta</code> | θ | | | <code>\geq</code> | \geq |
| <code>\lambda</code> | λ | <code>\Lambda</code> | Λ | <code>\neq</code> | \neq |
| <code>\mu</code> | μ | | | <code>\propto</code> | \propto |
| <code>\nu</code> | ν | | | <code>\div</code> | \div |
| <code>\pi</code> | π | <code>\Pi</code> | Π | <code>\circ</code> | \circ |
| <code>\phi</code> | ϕ | | | <code>\leftrightarrow</code> | \leftrightarrow |
| <code>\rho</code> | ρ | | | <code>\leftarrow</code> | \leftarrow |
| <code>\sigma</code> | σ | <code>\Sigma</code> | Σ | <code>\rightarrow</code> | \rightarrow |
| <code>\tau</code> | τ | | | <code>\uparrow</code> | \uparrow |
| <code>\omega</code> | ω | <code>\Omega</code> | Ω | <code>\downarrow</code> | \downarrow |

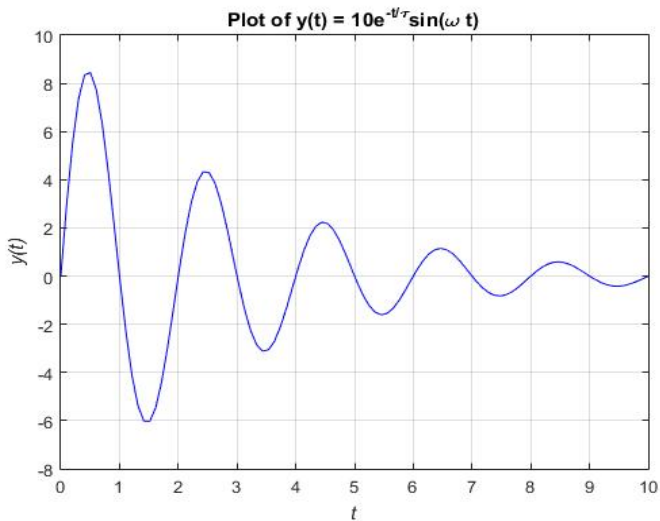


範例：圖形視窗出現特殊字體與數學符號

```
tau = 3;  
omega = pi;  
t = linspace(0,10);  
y = 10 * exp(-t./tau) .* sin(omega*t);  
plot(t,y,'b-');  
title('Plot of  $y(t) = 10e^{-t/\tau}\sin(\omega t)$ ');  
xlabel('\it{t}');  
ylabel('\it{y(t)}') ;  
grid on;
```



範例的繪圖結果 (承上頁)



Lecture 2

極座標圖



表 7.1.1 `polar()` 函數的使用

| 函 數 | 說 明 |
|-------------------------------------|-------------------------------------|
| <code>polar(theta, r)</code> | 根據角度向量 $theta$ ，以及距原點的長度 r 繪製極座標圖 |
| <code>polar(theta, r, 'str')</code> | 依據格式字串 str 所指定的格式繪製極座標圖 |

- 變數 `theta` 是弧度 (或徑度) 角的陣列，逆時針方向為正。
- 變數 `r` 為離原點的距離陣列。
- 函式 `polar` 可以輕易地繪製極座標函數

$$r = f(\theta), \quad \alpha \leq \theta \leq \beta$$

的二維圖形。

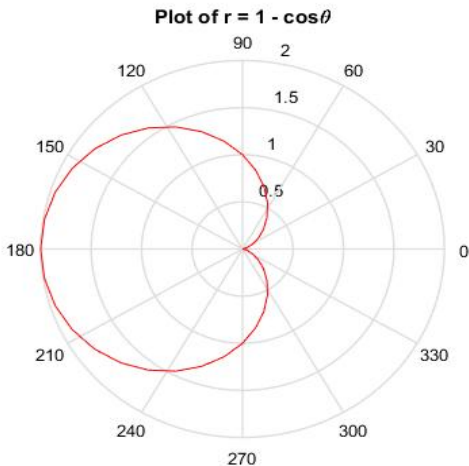


範例：心臟線 (Cardioid)

```
% 繪製函數  $r = f(\theta) = 1 - \cos \theta$ ,  $0 \leq \theta \leq 2\pi$ , 的圖形。  
theta = linspace(0,2*pi, 41);  
r = 1 - cos(theta);  
polar(theta,r,'r-');  
title('Plot of  $r = 1 - \cos \theta$ ');
```



範例的繪圖結果 (承上頁)



函式 fplot 的語法

表 5.4.1 繪圖函數 fplot() 的用法

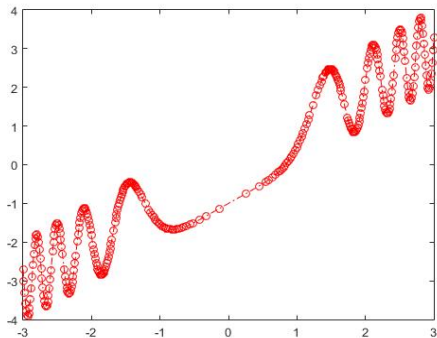
| 函 數 | 說 明 |
|--|--|
| <code>fplot('f_str', [xmin,xmax])</code> | 繪出函數 f_str 的圖形， x 軸的範圍取 $xmin$ 到 $xmax$ |
| <code>fplot('f_str', [xmin,xmax,ymin,ymax])</code> | 繪出函數 f_str 的圖形， x 軸的範圍取 $xmin$ 到 $xmax$ ， y 軸的範圍取 $ymin$ 到 $ymax$ |



```
>> fplot('x - cos(x^3)', [-3,3], 'ro-.')
```




```
>> fplot('x - cos(x^3)', [-3,3], 'ro-.')
```



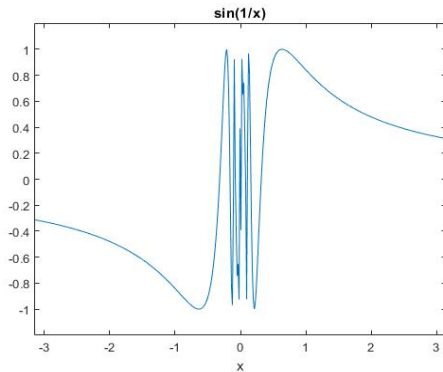
除了繪製一般函數圖形之外，`ezplot` 還可以處理隱函數與參數方程式的圖形，其使用語法如下：

表 5.4.2 繪圖函數 `ezplot()` 的用法

| 函 數 | 說 明 |
|--|--|
| <code>ezplot('f_str', [xmin, xmax])</code> | 繪出函數 f_str 的圖形，繪圖範圍在 x 與 y 方向均取 $xmin$ 到 $xmax$ |
| <code>ezplot('f_str', [xmin, xmax, ymin, ymax])</code> | 繪出函數 f_str 的圖形，繪圖範圍在 x 方向取 $xmin$ 到 $xmax$ 在 y 方向均取 $ymin$ 到 $ymax$ |
| <code>ezplot('fx', 'fy', [tmin, tmax])</code> | 參數繪圖，繪出 $(fx(t), fy(t))$ ， t 從取 $tmin$ 到 $tmax$ 的參數圖 |



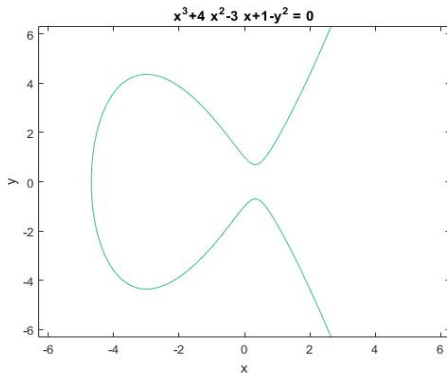
```
>> ezplot('sin(1/x)', [-pi, pi])
```



```
>> ezplot('x^3+4*x^2-3*x+1-y^2')
```



```
>> ezplot('x^3+4*x^2-3*x+1-y^2')
```



```
>> ezplot('3*cos(t)', '4*sin(t)', [0, 3*pi/2])
```



```
>> ezplot('3*cos(t)', '4*sin(t)', [0, 3*pi/2])
```

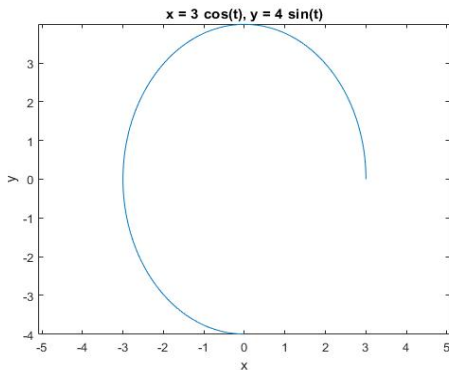


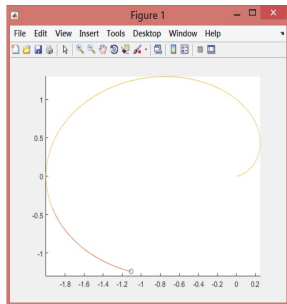
表 7.4.1 使用 `comet()` 函數

| 函 數 | 說 明 |
|------------------------------|--|
| <code>comet(x,y,p)</code> | 繪出彗星軌跡圖，彗星尾巴拖的長度為 $p \cdot \text{length}(y)$ ，若 p 省略，則 p 的預設值為 0.1 |
| <code>comet3(x,y,z,p)</code> | 同上，但繪出三維的彗星軌跡圖 |



範例：心臟線的動畫版本

```
>> t = linspace(0,2*pi,4e4);  
>> r = 1 - cos(t);  
>> comet(r.*cos(t),r.*sin(t))  
% 心臟線將慢慢呈現在您眼前！
```



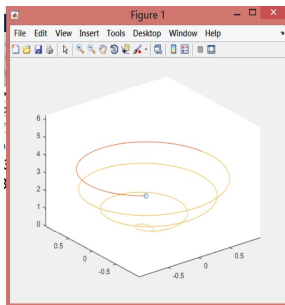
範例：三維參數曲線的動畫版本

```
>> t = linspace(0,2*pi,4e4);  
>> comet3(sin(t/2).*cos(6*t),sin(t/2).*sin(6*t),t)  
% 請試試看，見證奇蹟的時刻即將開始！
```



範例：三維參數曲線的動畫版本

```
>> t = linspace(0,2*pi,4e4);  
>> comet3(sin(t/2).*cos(6*t),sin(t/2).*sin(6*t),t)  
% 請試試看，見證奇蹟的時刻即將開始！
```



Lecture 3

註解並儲存圖形



Recall: 設定圖形標題和座標軸的解說文字，可使用下列函式：

表 5.3.1 於圖形內加入文字

| 函 數 | 說 明 |
|-----------------------------|--------------------------------------|
| <code>title('text')</code> | 設定圖形的標題文字為 <i>text</i> |
| <code>xlabel('text')</code> | 設定 x 軸的解說文字為 <i>text</i> |
| <code>ylabel('text')</code> | 設定 y 軸的解說文字為 <i>text</i> |
| <code>zlabel('text')</code> | 設定 z 軸的解說文字為 <i>text</i> （用於三維的繪圖） |



為了提高繪製圖形的可讀性，MATLAB 提供下列函式：

表 5.3.2 加入圖形的註解

| 函 數 | 說 明 |
|--|--|
| <code>legend(str₁,str₂,...)</code> | 加入曲線說明的文字 |
| <code>legend(str₁,str₂,...,pos)</code> | 設定曲線說明文字的位置，pos 設 1 代表將說明文字放在右上角，2 是左上，3 是左下，4 則是放在右下角 |
| <code>legend off</code> | 清除曲線說明文字 |
| <code>text(x,y,'text')</code> | 在圖形中位置為 (x,y) 之處加入註解文字 |
| <code>gtext('text')</code> | 利用滑鼠來設定文字輸入的位置 |

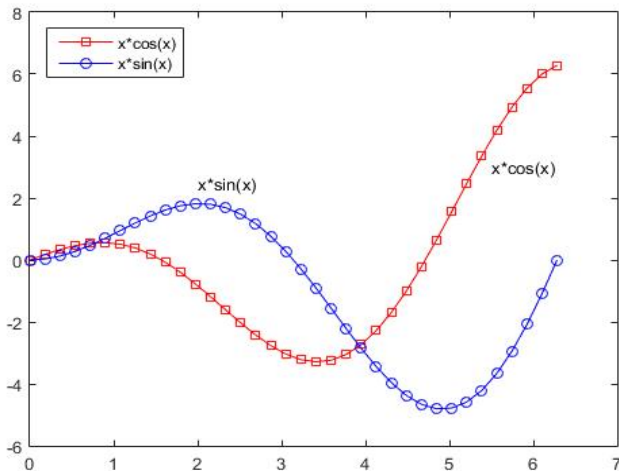


使用 legend 和 text 的範例

```
x = linspace(0,2*pi,36);  
y1 = x.*cos(x);  
y2 = x.*sin(x);  
plot(x, y1, '-rs', x, y2, '-bo');  
legend('x*cos(x)', 'x*sin(x)', 'Location', 'NW');  
text(2, 2.5, 'x*sin(x)');  
text(5.5, 3, 'x*cos(x)');
```



範例的繪圖結果 (承上頁)



Lecture 4

額外的二維圖形類別



MATLAB 除了提供一些基本的二維繪圖函式以外，還另外提供了下列常見的統計圖形：

- 長桿圖 (stem plots)
- 長條圖、直條圖 (bar plots)
- 圓形圖 (pie plots)
- 直方圖 (histogram)
- 階梯圖 (stair plots)
- 羅盤圖 (compass plots)



統計繪圖函式的語法

| 函式 | 說明 |
|---|---|
| <code>bar(x, y)</code> | 產生直條圖，使用 x 值來標示每根長條，而使用 y 值來決定長條的垂直高度。 |
| <code>barh(x, y)</code> | 產生橫條圖，使用 x 值來標示每根長條，而使用 y 值來決定長條的水平長度。 |
| <code>compass(x, y)</code> | 產生極座標圖，並畫出從座標原點到資料點 (x, y) 的箭頭。請注意圓形內的資料點位置，是用直角座標表示，而不是用極座標表示。 |
| <code>pie(x)</code> <code>pie(x, explode)</code> | 產生圓形圖。這函式計算每個 x 值相對於全部的比例，並依此比例畫出對等大小的扇形區域。選項陣列 <code>explode</code> 可以決定是否要把個別的扇形區域，與其他區域隔開顯示。 |
| <code>stairs(x, y)</code> | 產生階梯圖，而每個階梯的中心將落在資料點 (x, y) 上。 |
| <code>stem(x, y)</code> | 產生長桿圖，在每個資料點 (x, y) 上有個標記，並且由該點垂直畫一條線連接到 x 軸上。 |

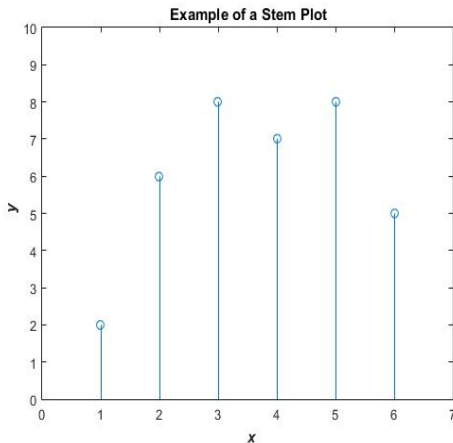


長桿圖的範例

```
x = [1 2 3 4 5 6];  
y = [2 6 8 7 8 5];  
stem(x,y);  
title('\bf Example of a Stem Plot');  
xlabel('\bf \it x');  
ylabel('\bf \it y');  
axis([0 7 0 10]);
```



範例的圖結果 (承上頁)

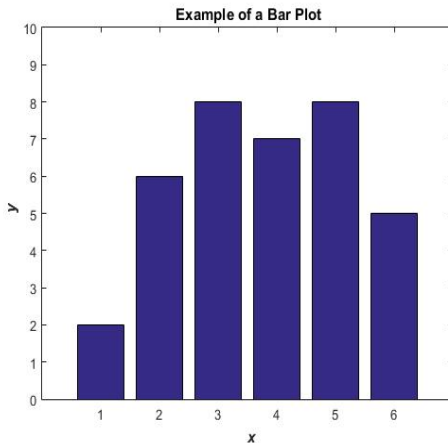


直條圖的範例

```
x = [1 2 3 4 5 6];  
y = [2 6 8 7 8 5];  
bar(x,y);  
title('\bf Example of a Bar Plot');  
xlabel('\bf \it x');  
ylabel('\bf \it y');  
axis([0 7 0 10]);
```



範例的繪圖結果 (承上頁)

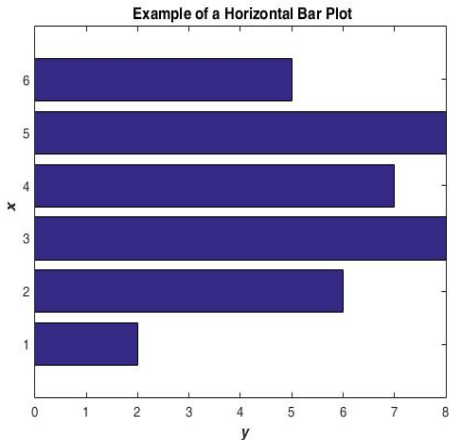


橫條圖的範例

```
x = [1 2 3 4 5 6];  
y = [2 6 8 7 8 5];  
barh(x,y);  
title('\bf Example of a Horizontal Bar Plot');  
xlabel('\bf \it y');  
ylabel('\bf \it x');
```



範例的繪圖結果 (承上頁)

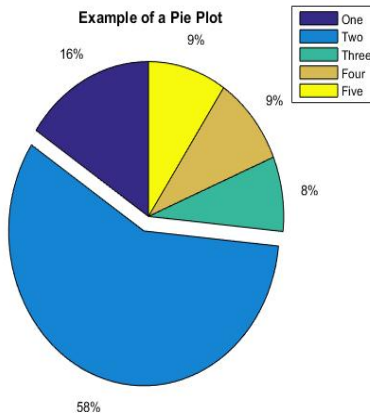


圓形圖的範例

```
data = [10 37 5 6 6];  
explode = [0 1 0 0 0];  
pie(data,explode);  
title('\bf Example of a Pie Plot');  
legend('One','Two','Three','Four','Five');
```



範例的繪圖結果 (承上頁)



三維圓形圖的範例

```
data = [10 37 5 6 6];  
explode = [0 1 0 0 0];  
pie3(data,explode);  
title('\bf Example of a 3D Pie Plot');  
legend('One','Two','Three','Four','Five');
```



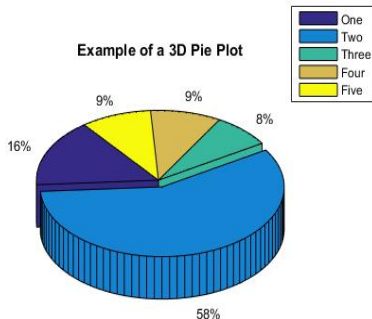


表 7.3.4 直方圖繪圖函數 hist()

| 函 數 | 說 明 |
|-----------------------------|--|
| <code>v=hist(data)</code> | 將向量 <i>data</i> 按數據大小分成 10 個等距的區間，然後將這 10 個區間內元素的個數傳回給向量 <i>v</i> 。若 <i>data</i> 為一矩陣，則會把同一行的元素視為同一筆資料來進行統計。另外，如果沒有設計傳回值，則直接繪出其直方圖 |
| <code>v=hist(data,n)</code> | 同上，但區間數為 <i>n</i> |



```
>> data = [0 3 3 4 5 3 7 4 2 8 2 8 10];  
>> v = hist(data)
```

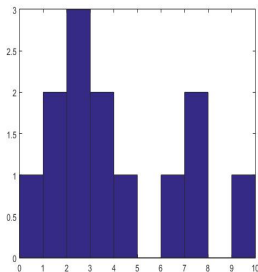


```
>> data = [0 3 3 4 5 3 7 4 2 8 2 8 10];
```

```
>> v = hist(data)
```

```
v =
```

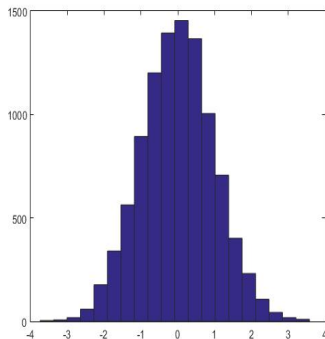
```
1 2 3 2 1 0 1 2 0 1
```




```
>> data2 = randn(10000,1);  
>> hist(data2,20)
```



```
>> data2 = randn(10000,1);  
>> hist(data2,20)
```

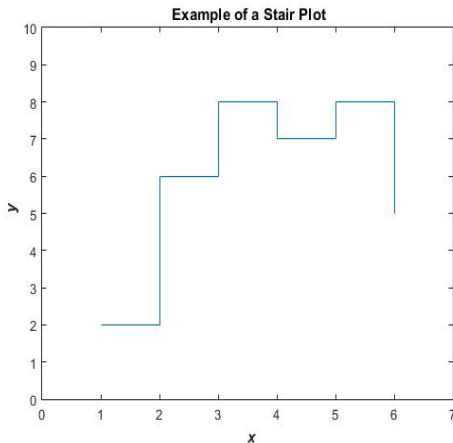


階梯圖的範例

```
x = [1 2 3 4 5 6];  
y = [2 6 8 7 8 5];  
stairs(x,y);  
title('\bf Example of a Stair Plot');  
xlabel('\bf \it x');  
ylabel('\bf \it y');  
axis([0 7 0 10]);
```



範例的繪圖結果 (承上頁)



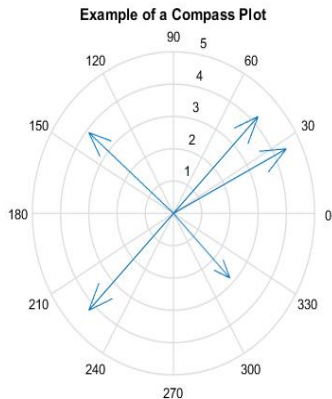
羅盤圖的範例

```
x = [3 4 2 -3 -3];  
y = [3 2 -2 -3 2.5];  
compass(x,y);  
title('\bf Example of a Compass Plot');
```

- 與極座標圖形相似，但不是取 (r, θ) 為繪圖資料點。
- 仍是以直角坐標 (x, y) 作為繪圖資料點，並將原點與資料點之間的連線段以**箭號**表示。



範例的繪圖結果 (承上頁)



Lecture 5

利用 `plot` 函式對二維陣列繪圖



- 一般而言，繪圖資料點都是儲存在一維陣列或向量。
- 如果我們擁有的是二維陣列資料而不是向量資料，那會怎樣？
- MATLAB 會把二維陣列的**每一行視為一條線**，而且資料集合內有多少行，就會畫出多少條線。

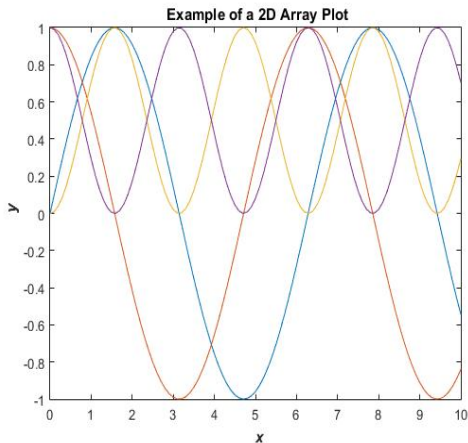


輸入引數為二維陣列的範例

```
x = 0:0.1:10;  
y = zeros(length(x),4);  
y(:,1) = sin(x);  
y(:,2) = cos(x);  
y(:,3) = sin(x).^2;  
y(:,4) = cos(x).^2;  
plot(x,y);  
title('\bf Example of a 2D Array Plot');  
xlabel('\bf \it x');  
ylabel('\bf \it y');
```



範例的繪圖結果 (承上頁)

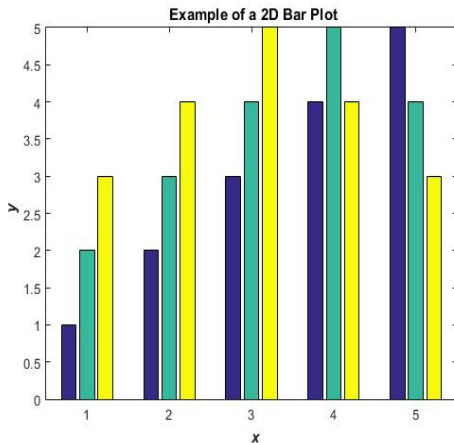


輸入引數為二維陣列的直條圖

```
x = 1:5;  
y = zeros(5,3);  
y(1,:) = [1 2 3];  
y(2,:) = [2 3 4];  
y(3,:) = [3 4 5];  
y(4,:) = [4 5 4];  
y(5,:) = [5 4 3];  
bar(x,y);  
title('\bf Example of a 2D Bar Plot');  
xlabel('\bf \it x');  
ylabel('\bf \it y');
```



範例的繪圖結果 (承上頁)



Thank you for your attention!

