

Chapter 8

複數與三維繪圖

Hung-Yuan Fan (范洪源)

Department of Mathematics,
National Taiwan Normal University, Taiwan

Spring 2017



- 8.1 複數資料
- 8.2 多維陣列
- 8.3 三維圖形



Section 8.1

複數資料



- 複數 (**complex numbers**) 是包含實數及虛數的一種數字。
- 大多數描述電機與機械系統行為的微分方程式，也都會產生複數型態的解。
- 複數的使用很廣泛，我們必須深入了解複數的使用及其處理方法，才能順利解決問題。
- 複數的一般形式為：

$$c = a + bi \quad (\text{直角座標表示法})$$

$$= z \cos \theta + (z \sin \theta)i = ze^{i\theta}, \quad (\text{極座標表示法})$$

其中 $i = \sqrt{-1}$ 或是 $i^2 = -1$ 。



複數的座標示意圖

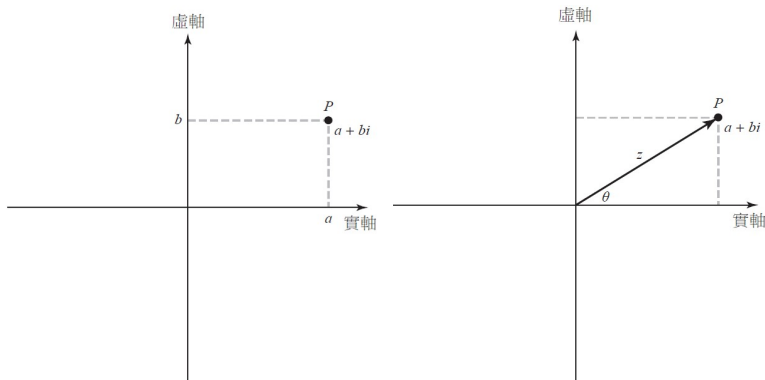


Figure: 直角座標的複數表示 (左) 和極座標的複數表示 (右)



兩座標間的轉換

- 極座標轉直角坐標 $(z, \theta) \mapsto (a, b)$:

$$a = z \cos \theta,$$

$$b = z \sin \theta$$

- 直角坐標轉極座標 $(a, b) \mapsto (z, \theta)$:

$$z = \sqrt{a^2 + b^2},$$

$$\theta = \tan^{-1} \frac{b}{a},$$

其中幅角 θ 可以由函式 `atan2(b, a)` 或 `atan2d(b, a)` 求得，且輸出角度範圍是 $-\pi \leq \theta \leq \pi$ 或 $-180^\circ \leq \theta \leq 180^\circ$ 。



複數的基本運算

對於任何兩個複數 $c_1 = a_1 + b_1 i = z_1 e^{i\theta_1}$ 和 $c_2 = a_2 + b_2 i = z_2 e^{i\theta_2}$, 定義加減乘除如下:

- 直角坐標形式:

$$c_1 \pm c_2 = (a_1 \pm a_2) + (b_1 \pm b_2)i,$$

$$c_1 \times c_2 = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2)i,$$

$$\frac{c_1}{c_2} = \frac{a_1 a_2 + b_1 b_2}{a_2^2 + b_2^2} + \frac{b_1 a_2 - a_1 b_2}{a_2^2 + b_2^2} i$$

- 極坐標形式:

$$c_1 \pm c_2 = (z_1 \cos \theta_1 \pm z_2 \cos \theta_2) + (z_1 \sin \theta_1 \pm z_2 \sin \theta_2)i,$$

$$c_1 \times c_2 = (z_1 z_2) e^{i(\theta_1 + \theta_2)}, \quad \frac{c_1}{c_2} = \frac{z_1}{z_2} e^{i(\theta_1 - \theta_2)}$$



複數變數的初始化

- 1 使用內建值 i 或 j 來產生複數值，這兩個值在 MATLAB 都被預設為 $\sqrt{-1}$ 。

```
>> c1 = 4 + i*3
```

```
c1 =
```

```
4.0000 + 3.0000i
```

- 2 直接在數字的虛數部分之後，加上 i 或 j 來指定複數的虛數部分。

```
>> c1 = 4 + 3i
```

```
c1 =
```

```
4.0000 + 3.0000i
```



在複數間使用關係運算子

- 關係運算子 $>$, $>=$, $<$ 與 $<=$ 只會比較複數的實數部分。

```
>> c1 = 4 + 3i; c2 = 3 + 8i;
```

```
>> c1 > c2
```

```
ans =
```

```
1 % 代表敘述式 c1 > c2 是對的 (true)!
```

- 較合理的關係運算是比較複數的絕對值

$$|c| = z = \sqrt{a^2 + b^2}$$

的大小，而不是比較複數實部的大小。

```
>> compare_abs = [abs(c1), abs(c2)]
```

```
compare_abs =
```

```
5.0000 8.5440
```



複數函式

- MATLAB 包含了許多支援複數運算的函式。
- 這些函式可以分成三種類別：
 - ① 型態轉換函式: `real`、`imag` 等函式。
 - ② 絕對值與角度函式: `abs`、`angle` 等函式。
 - ③ 數學函數: 包含了指數函數、對數函數、三角函數與平方根函數。例如, MATLAB 的 `sin`、`cos`、`log`、`sqrt` 等函式也可以處理複數資料型態。



一些支援複數資料型態的函式

函式	描述
<code>conj(c)</code>	計算 c 的共軛複數。如果 $c = a + bi$ ，則 $\text{conj}(c) = a - bi$ 。
<code>real(c)</code>	傳回複數 c 的實數部份 (real part)。
<code>imag(c)</code>	傳回複數 c 的虛數部份 (imaginary part)。
<code>isreal(c)</code>	如果陣列 c 內的元素沒有虛數部分，則傳回 true 值 (1)。
<code>abs(c)</code>	傳回複數 c 的絕對值大小。
<code>angle(c)</code>	由 $\text{atan2}(\text{imag}(c), \text{real}(c))$ 的式子，計算並傳回複數 c 的角度。



MATLAB 複數資料的圖形與實數資料的圖形會有些微的不同。
以繪製複數值函數 (complex-valued function)

$$y(t) = e^{-0.2t}(\cos t + i\sin t), \quad 0 \leq t \leq 4\pi$$

的圖形為例。

複數值函數的繪圖

```
t = 0:pi/20:4*pi;  
y = exp(-0.2*t).*(cos(t)+i*sin(t));  
plot(t,y,'LineWidth',2);  
title('\bf Plot of Complex Function vs Time');  
xlabel('\bf \it t');  
ylabel('\bf \it y(t)');
```



範例的繪圖結果 (承上頁)

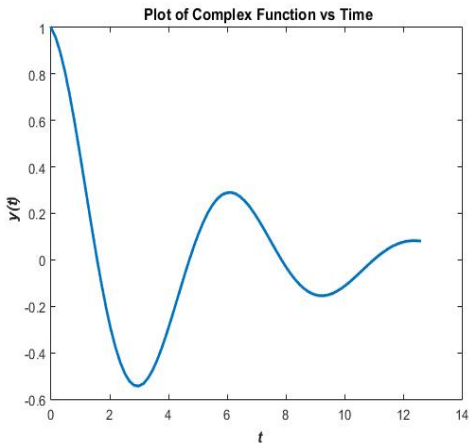


Figure: 函式 plot 不能畫出複數值函數的圖形!



方法一：複數值函數圖形的呈現

```
t = 0:pi/20:4*pi;  
y = exp(-0.2*t).*(cos(t)+i*sin(t));  
plot(t,real(y),'b-','LineWidth',2);  
hold on;  
plot(t,imag(y),'r--','LineWidth',2);  
title('\bf Plot of Complex Function vs Time');  
xlabel('\bf \it t');  
ylabel('\bf \it y(t)');  
legend('real part','imag. part');  
hold off;
```



範例的繪圖結果 (承上頁)

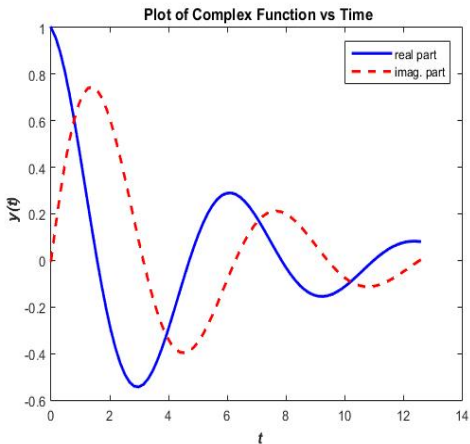


Figure: 在同一張圖形中呈現複數值函數的實部和虛部



方法二：複數值函數圖形的呈現

```
t = 0:pi/20:4*pi;  
y = exp(-0.2*t).*(cos(t)+i*sin(t));  
plot(y,'b-','LineWidth',2);  
title('\bf Plot of Phase Portrait (相位圖)');  
xlabel('\bf Real Part');  
ylabel('\bf Imaginary Part');
```



範例的繪圖結果 (承上頁)

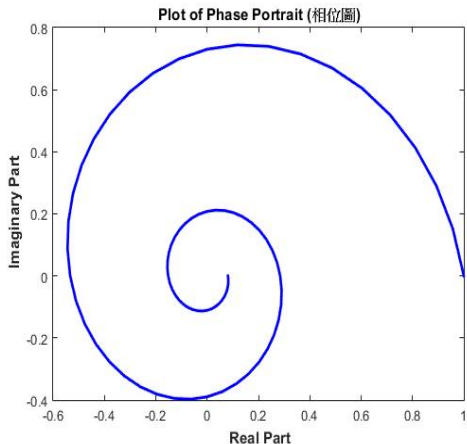


Figure: 複數值函數的實部對虛部之相位圖



方法三：複數值函數圖形的呈現

```
t = 0:pi/20:4*pi;  
y = exp(-0.2*t).*(cos(t)+i*sin(t));  
polar(angle(y),abs(y));  
title('\bf Plot of Phase Portrait (極座標圖)');
```



範例的繪圖結果 (承上頁)

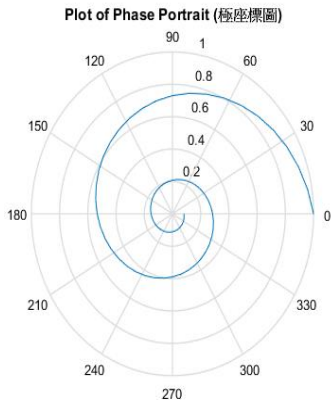


Figure: 以極座標表示複數值函數的相位圖



方法四：複數值函數圖形的呈現 (動畫版)

```
t = linspace(0,4*pi,4e4);  
y = exp(-0.2*t).*(cos(t)+i*sin(t));  
comet(real(y),imag(y));  
title('\bf Plot of Phase Portrait (相位圖動畫版)');  
xlabel('\bf Real Part');  
ylabel('\bf Imaginary Part');
```



範例的繪圖結果 (承上頁)

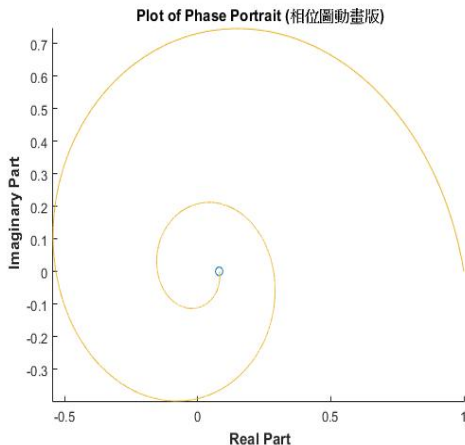


Figure: 複數值函數的實部對虛部之相位圖動畫版



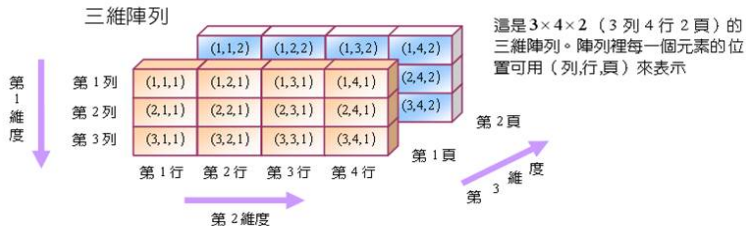
Section 8.2

多維陣列



二維與三維陣列

- 陣列的維度多於二維，稱為**多維陣列**。
- 三維陣來必須以**列、行與頁**三個維度來描述。



三維陣列的建立

- 1 由一維或是二維陣列所生成。

```
>> a = [1 2 3 4; 5 6 7 8; 9 10 11 12];  
>> a(:, :, 2) = zeros(3,4)  
% 變數 a 是一個  $3 \times 4 \times 2$  的陣列。
```

- 2 由 MATLAB 內建函式生成。

```
>> b = ones(4,4,2)  
>> c = randn(2,2,3)
```



檢查多維陣列的維度和大小

- 函式 `ndims`: 用來取得多維陣列的維度。

```
>> ndims(c)
```

```
ans =
```

```
3
```

- 函式 `size`: 用來取得多維陣列的大小。

```
>> size(c)
```

```
ans =
```

```
2 2 3
```



Section 8.3

三維圖形



三維空間曲線的繪圖

表 6.4.1 空間曲線繪圖函數

函數	說明
<code>plot3(x,y,z)</code>	分別以向量 x,y 與 z 代表資料點在每一個座標軸的位置，繪製三維空間曲線
<code>plot3(x,y,z,'str')</code>	以控制字串 str 所指定的格式繪出三維空間曲線



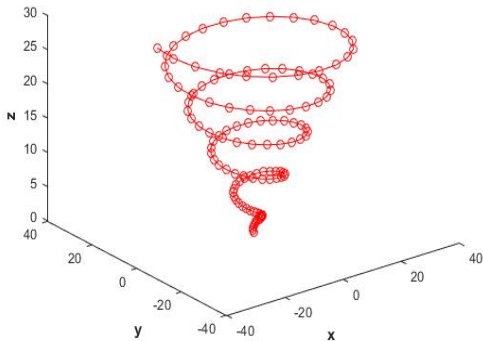
函式 plot3 的範例 (1/2)

```
t = linspace(0,30,120);  
plot3(t.*sin(t),t.*cos(t),t,'-ro');  
title('\bf Plot of 3D Parametric Curve');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)

Plot of 3D Parametric Curve



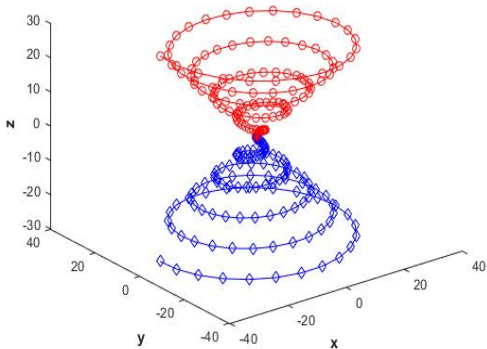
函式 plot3 的範例 (2/2)

```
t = linspace(0,30,120);  
plot3(t.*sin(t),t.*cos(t),t, '-ro', ...  
t.*sin(t),t.*cos(t),-t, '-bd');  
title('\bf Plot of 3D Parametric Curve');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)

Plot of 3D Parametric Curve



如何繪製函數 $z = f(x, y)$ 的圖形?

- 對於函數 $f(x, y)$ 而言，每給一組 (x, y) ，便能計算其函數值 $z = f(x, y)$ 。
- 只要給予 (x, y) 的組數夠多，即可繪出函數的三維曲面圖。

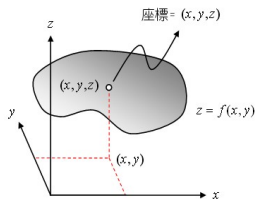


圖 6.1.1

函數圖形於三維空間
的示意圖



描繪三維圖形的步驟

Step 1 建立 x 軸和 y 軸方向上的二維陣列 xx 與 yy 。

- 直接使用宣告式由鍵盤輸入兩陣列的所有元素。
- 使用函式 `meshgrid` 自動生成兩陣列。

Step 2 建立 z 軸方向上的二維陣列 zz 。

- 直接使用宣告式由鍵盤輸入陣列 zz 的元素。
- 由函數 $z = f(x, y)$ 的形式與陣列運算建立二維陣列 zz 。

Step 3 使用內建函式繪製三維圖形和等高線圖。

- `mesh(xx,yy,zz)`: 產生一個網格圖 (**mesh**) 或是線框圖 (**wireframe**) 。
- `surf(xx,yy,zz)`: 產生一個曲面圖 (**surface**) 。
- `contour(xx,yy,zz)`: 產生一個等高線圖 (**contour**) 。
- 建議使用 `title`、`xlabel`、`ylabel`、`zlabel` 等函式增加圖形的可讀性。



MATLAB 如何定義這些二維陣列? (1/2)

對於 x 、 y 軸上取樣的網格點 (grid points) 或資料點

$$x = [x_1, x_2, \dots, x_n], \quad y = [y_1, y_2, \dots, y_m],$$

MATLAB 定義二維陣列 xx 與 yy 具有下列矩陣形式:

$$xx = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_1 & x_2 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad yy = \begin{bmatrix} y_1 & y_1 & \cdots & y_1 \\ y_2 & y_2 & \cdots & y_2 \\ \vdots & \vdots & & \vdots \\ y_m & y_m & \cdots & y_m \end{bmatrix} \in \mathbb{R}^{m \times n}$$



MATLAB 如何定義這些二維陣列? (2/2)

因此，與 z 軸方向有關的二維陣列 zz 應當定義為

$$zz = \begin{bmatrix} f(x_1, y_1) & f(x_2, y_1) & \cdots & f(x_n, y_1) \\ f(x_1, y_2) & f(x_2, y_2) & \cdots & f(x_n, y_2) \\ \vdots & \vdots & & \vdots \\ f(x_1, y_m) & f(x_2, y_m) & \cdots & f(x_n, y_m) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

其中， m 是向量 y 的元素個數且 n 是向量 x 的元素個數。

Note: 使用元素對元素的陣列運算和 f 的形式，我們可以輕易地建構二維陣列 zz !



範例

若考慮在 $x = 1, 2, 3$, 和 $y = 1, 2, 3, 4$ 點上繪出雙自變量函數

$$z = f(x, y) = \sqrt{x^2 + y^2}$$

的圖形, 則 MATLAB 會如何設定陣列 xx 和 yy ?



範例的解答 (承上頁)

```
>> xx = [1 2 3; 1 2 3; 1 2 3; 1 2 3]
```

```
xx =
```

```
1 2 3
```

```
1 2 3
```

```
1 2 3
```

```
1 2 3
```

```
>> yy = [1 1 1; 2 2 2; 3 3 3; 4 4 4]
```

```
yy =
```

```
1 1 1
```

```
2 2 2
```

```
3 3 3
```

```
4 4 4
```



使用 meshgrid 函式建立陣列

表 6.1.2 meshgrid() 的使用

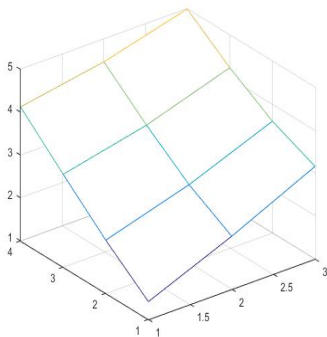
函數	說明
meshgrid(vx,vy)	以向量 vx 代表 x 軸方向資料點的位置，以 vy 代表 y 軸方向資料點的位置，建構出兩個二維矩陣 xx 與 yy，以供三維繪圖所需

```
>> vx = 1:3;   vy = 1:4;  
>> [xx,yy] = meshgrid(vx,vy)  
% 輸出二維陣列xx、yy與上頁範例的結果相同!
```



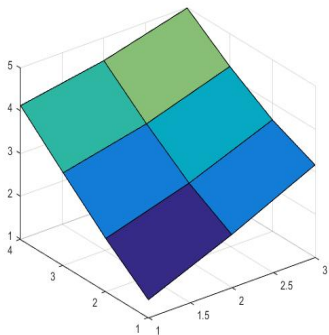
承上例，我們可以畫出 $z = f(x, y)$ 的網格圖：

```
>> zz = sqrt(xx.^2+yy.^2); % 建立函數值的二維陣列 zz  
>> mesh(xx,yy,zz)
```



同理，我們也可以畫出 $z = f(x, y)$ 的曲面圖：

```
>> surf(xx,yy,zz)
```



範例

嘗試在 $-4 \leq x \leq 4$ 以及 $-3 \leq y \leq 3$ 區間上，設定資料點的取樣間距為 **0.1**，繪製函數

$$z = f(x, y) = e^{-0.5(x^2 + 0.5(x-y)^2)}$$

的網格圖、曲面圖和等高線圖。

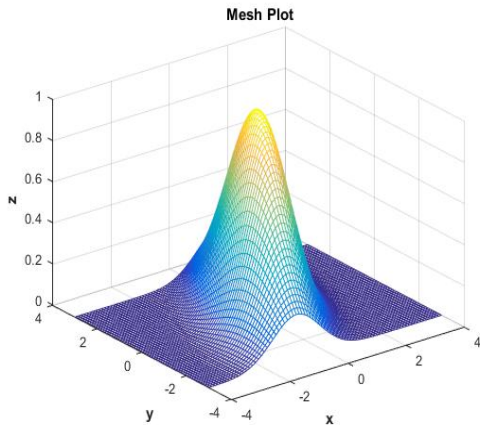


使用函式 `mesh` 的程式碼 (承上例)

```
[xx,yy] = meshgrid(-4:0.1:4,-3:0.1:3);  
zz = exp(-0.5*(xx.^2+0.5*(xx-yy).^2));  
mesh(xx,yy,zz);  
title('\bf Mesh Plot');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)

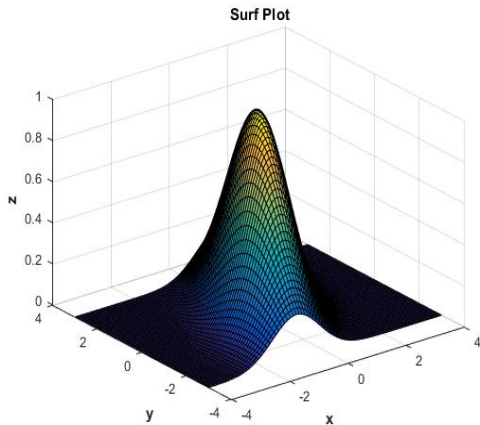


使用函式 surf 的程式碼 (承上例)

```
[xx,yy] = meshgrid(-4:0.1:4,-3:0.1:3);  
zz = exp(-0.5*(xx.^2+0.5*(xx-yy).^2));  
surf(xx,yy,zz);  
title('\bf Surf Plot');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)



使用函式 `contour` 的程式碼 (承上例)

```
[xx,yy] = meshgrid(-4:0.1:4,-3:0.1:3);  
zz = exp(-0.5*(xx.^2+0.5*(xx-yy).^2));  
contour(xx,yy,zz);  
title('\bf Contour Plot');  
xlabel('\bf x');  
ylabel('\bf y');
```



範例的繪圖結果 (承上頁)

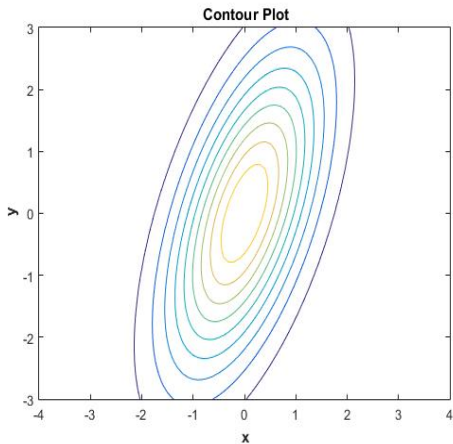


表 6.1.3 meshc() 與 waterfall() 函數的使用

函數	說明
<code>meshc(xx,yy,zz)</code>	繪出網格圖，但在網格圖下方會附帶繪出等高線圖
<code>waterfall(xx,yy,zz)</code>	以切片的方式來繪製三維的立體圖

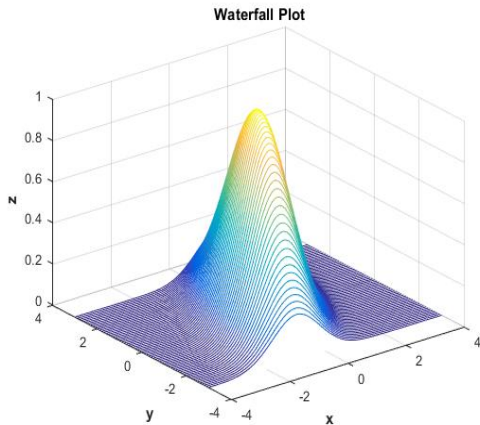


使用函式 `waterfall` 的程式碼 (承上例)

```
[xx,yy] = meshgrid(-4:0.1:4,-3:0.1:3);  
zz = exp(-0.5*(xx.^2+0.5*(xx-yy).^2));  
waterfall(xx,yy,zz);  
title('\bf Waterfall Plot');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)



使用函式 meshc 的程式碼

```
x = linspace(-8,8,30);  
y = x; [xx,yy] = meshgrid(x,y);  
expr = sqrt(xx.^2+yy.^2);  
zz = sin(expr)./(expr+eps);  
meshc(xx,yy,zz);  
title('\bf Meshc Plot');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)

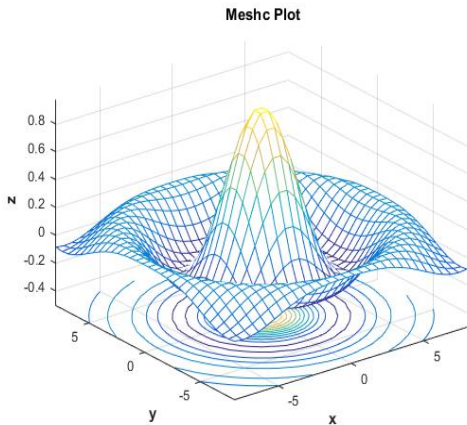


表 6.1.4 surf() 與 surfc() 函數的使用

函數	說明
<code>surf(xx,yy,zz)</code>	分別以資料點的 x 、 y 與 z 座標之集合所組成的矩陣 xx 、 yy 與 zz 來繪出三維的曲面圖
<code>surfc(xx,yy,zz)</code>	同 <code>surf</code> ，但在圖形下方會顯示出函數圖形的等高線圖

Note: 若 \mathbf{vx} 、 \mathbf{vy} 分別為 x 、 y 軸資料點所形成的向量，則

- `mesh(vx,vy,zz)`、`meshc(vx,vy,zz)` 分別和指令 `mesh(xx,yy,zz)`、`meshc(xx,yy,zz)` 繪圖效果一樣。
- `surf(vx,vy,zz)`、`suurfc(vx,vy,zz)` 分別和指令 `surf(xx,yy,zz)`、`surfc(xx,yy,zz)` 繪圖效果一樣。

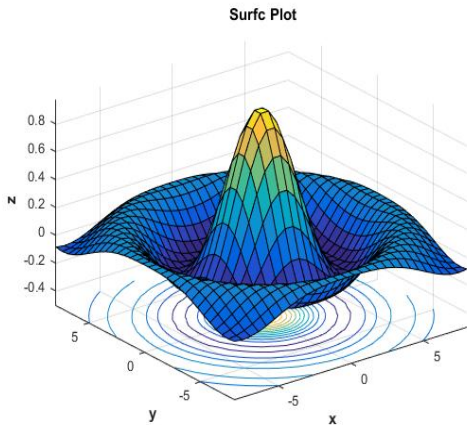


使用函式 `surf` 的程式碼 (承上例)

```
x = linspace(-8,8,30);  
y = x; [xx,yy] = meshgrid(x,y);  
expr = sqrt(xx.^2+yy.^2);  
zz = sin(expr)./(expr+eps);  
surf(xx,yy,zz);  
title('\bf Surf Plot');  
xlabel('\bf x');  
ylabel('\bf y');  
zlabel('\bf z');
```



範例的繪圖結果 (承上頁)



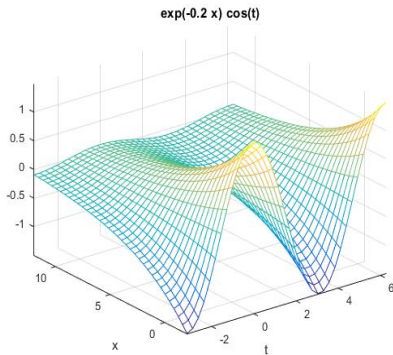
只要給予函數字串與範圍，下列函式可快速的繪出三維的圖形：

函式	說明
<code>ezmesh('f', [xmin,xmax,ymin,ymax], n)</code>	以 $n \times n$ 個網格點繪出 f 的三維圖形，若省略 x, y 範圍或 n 值，則預設值是 $-2\pi \leq x, y \leq 2\pi$ 且 $n = 60$
<code>ezmeshc('f', [xmin,xmax,ymin,ymax], n)</code>	同上，但圖形下方顯示等高線圖
<code>ezsurf('f', [xmin,xmax,ymin,ymax], n)</code>	同 <code>ezmesh</code> ，但網格面會塗上顏色
<code>ezsurfc('f', [xmin,xmax,ymin,ymax], n)</code>	同上，但圖形下方顯示等高線圖



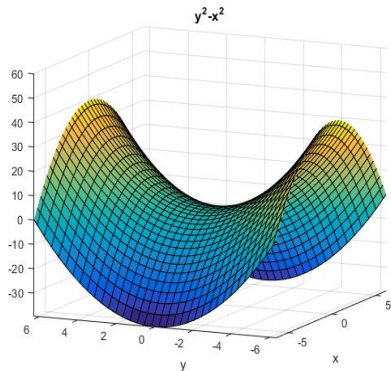
函式 ezmesh 的範例

```
>> ezmesh('exp(-0.2*x)*cos(t)', [-pi, 2*pi, -2, 12], 36)
```



函式 ezsurf 的範例

```
>> ezsurf('y^2-x^2',36) %  $z = f(x, y) = y^2 - x^2$  的圖形  
% 函數圖形是一個雙曲拋物面 (Hyperbolic Paraboloid) 。  
% 原點 (0,0) 是一個鞍點 (saddle point) 。
```



若空間曲面上任何一點 (x, y, z) 滿足下列雙參數方程式:

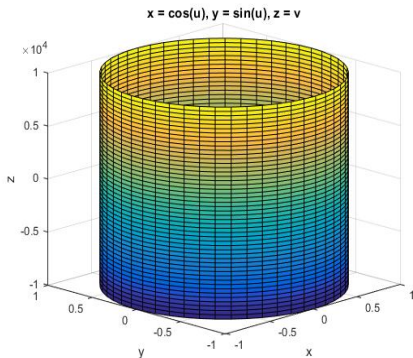
$$x = f(u, v), \quad y = g(u, v), \quad z = h(u, v),$$

其中參數區間為 $u_{\min} \leq u \leq u_{\max}$ 和 $v_{\min} \leq v \leq v_{\max}$ ，我們可利用函式 `ezsurf` 與 `ezmesh` 輕易地繪出參數曲面的圖形。

- `ezsurf('f','g','h',[umin, umax,vmin,vmax])`: 繪製參數曲面圖
- `ezmesh('f','g','h',[umin, umax,vmin,vmax])`: 繪製參數網格圖
- 曲面透明度可用函式 `alpha(value)` 調整，其中 `value` 是介於 0 到 1 的實數且預設值是 1。

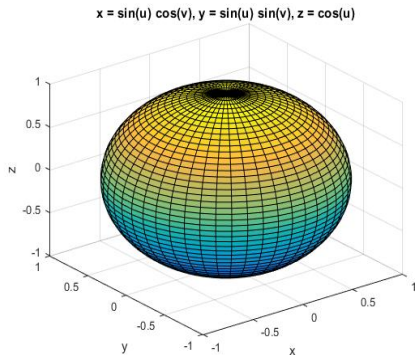


```
>> ezsurf('cos(u)', 'sin(u)', 'v', [0, 2*pi, -1e4, 1e4])
```



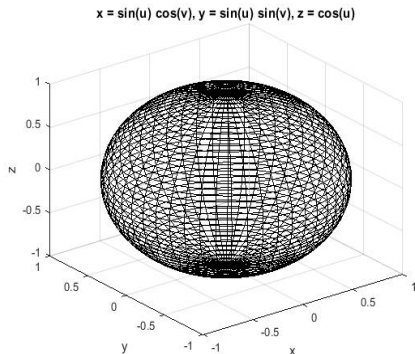
單位球體的曲面圖 (1/2)

```
>> ezsurf('sin(u)*cos(v)', 'sin(u)*sin(v)', 'cos(u)',  
[0,pi,0,2*pi])
```



單位球體的曲面圖 (2/2)

```
>> ezsurf('sin(u)*cos(v)', 'sin(u)*sin(v)', 'cos(u)',  
[0,pi,0,2*pi])  
>> alpha(0) % 完全透明的單位球體!
```



Thank you for your attention!

