Ch. 08 三維空間的繪圖

曲面繪製的程序
三維的座標系統
三維的繪圖方法

SURFACE程序的語法

語法	說明
SURFACE, Z[,X,Y]	繪製立體的網格曲面

```
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色

z=DIST(26,26)
CONTOUR,z
SURFACE,z
end
```

SURFACE程序的關鍵字

關鍵字	說明
AX=degrees	設定X軸旋轉的角度, 預設為30°
AZ=degrees	設定Z軸旋轉的角度, 預設為30°
BOTTOM=index	設定曲面底部的顏色
/HORIZONTAL	設定繪製曲面的線條為平行線
MAX_VALUE=value	設定Z值繪製的最大值
MIN_VALUE=value	設定Z值繪製的最小值
/SAVE	儲存座標轉換的矩陣
SHADES=array	設定每個網格點的顏色
SKIRT=value	在曲面周邊繪製裙擺
其他繪圖關鍵字	與PLOT程序共用部分繪圖

```
;window,xsize=216,ysize=162
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色
z=DIST(26,26)
CONTOUR,z
SURFACE,z
surface,z,AX=60,AZ=60,/save
contour,z,zvalue=1,/T3D,/NOERASE   ;與PLOT程序共用的繪圖關鍵字
surface,z,/horizontal ;曲面網格線為平行線
surface,z,/SKIRT
grid=dist(26,26)*10
surface,z,shades=grid  ;z值越高,顏色越淡
surface,z,bottom=200  ;設定曲面底部顏色
SURFACE,z,min_value=6,max_value=14
end
```

SHADE_SURF程序

語法	說明
SHADE_SURF,Z[,X,Y]	繪製立體的平滑曲面，引數X和Y是引數Z對應的座標

SURFACE 顯示網格曲面
SHADE_SURF 顯示平滑曲面，有些關鍵字與SURFACE程序共用

```
;window,xsize=216,ysize=162
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色

z=dist(26,26)
shade_surf,z
shade_surf,z,min_value=6,max_value=14
contour,z,zvalue=1,/T3D,/noerase
;IDL系統每次繪圖時會自動讀入座標轉換矩陣!P.T和尺度變化向量  !X.S、!Y.S、!Z.S
print,!P.T   ;座標轉換矩陣
Print
print,!X.S & print,!Y.S  & print,!Z.S   ; X,Y,Z 座標尺度變換向量

end
```

SCALE3程序

語法	說明
SCALE3	建立三維繪圖的座標系統

視窗的預設資料範圍是 [0,1]，
但實際的資料範圍可能會超過預設值，因此需要調整回[0,1]。

尺度轉換向量
!X.S =[-Xmin, 1] / [Xmax – Xmin]
!Y.S = [-Ymin, 1] / [Ymax – Ymin]
!Z.S = [-Zmin, 1] / [Zmax – Zmin]
其中第一個元素用來調整平移(translate)量
第二個元素用來改變縮放(scale)的比例。

SCALE3程序的關鍵字

關鍵字	說明
XRANGE=[Xmin, Xmax]	設定X軸的資料範圍
YRANGE=[Ymin, Ymax]	設定Y軸的資料範圍
ZRANGE=[Zmin, Zmax]	設定Z軸的資料範圍
AX=degrees	設定X軸旋轉的角度
AZ=degrees	設定Y軸旋轉的角度

SCAL3程序的關鍵字用來控制各個座標軸的資料範圍和旋轉角度。

```
;window,xsize=216,ysize=162
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色
theta=findgen(360)*10
x=1.5*cos(theta*!dtor)
y=1.5*sin(theta*!dtor)
z=theta*0.001
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[0,4]
;window,xsize=216,ysize=216
plot,x,y,/nodata,xstyle=4,ystyle=4   ;產生繪圖座標系統，但不繪圖
plots,x,y,z,/T3D
erase
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[0,4],ax=60,az=60
plots,x,y,z,/T3D
tm=!P.T                          ;將座標轉換矩陣存到變數tm
xs=!X.S & ys=!Y.S & zs=!Z.s   ;將各座標軸尺度變換向量分別存入不同變數
save,/all,filename='save_scale3_all.sav'   ;將所有變數存到指定的檔案中
end
```

- Suppose that you have saved all the variables from a previous IDL session with the command:

```
SAVE, /VARIABLES, FILENAME = 'session1.sav'
```

- If the file session1.sav is located in your current working directory, the variable associated with the file can be restored by entering:

```
RESTORE, 'session1.sav'
```

- Note: To restore a file that is not in your current working directory, you must specify the file path.

```
x=findgen(10)
y=indgen(20)
z=[1:20:0.5]
save,/all,filename='save_all_vars.sav'  ;將所有變數儲存到save_all_vars.sav檔案中
save,x,y,filename='save_2_vars.sav'
delvar,x,y      ;刪除變數X,Y
;restore,'save_2_vars.sav'  ;將save_2_vars.sav的儲存內容讀回
;restore,filename='save_2_vars.sav'
end
```

T3D程序

語法	說明
T3D	建立三維繪圖的座標系統

T3D程序的關鍵字

關鍵字	說明
ROTATE=[x,y,z]	執行旋轉
SCALE-[x,y,z]	執行放大或縮小
TRANSLATE=[x,y,z]	執行平移
/RESET	重置轉換矩陣
MATRIX=variable	輸出轉換矩陣

在SCALE3程序宣告各軸範圍後，使用T3D程序控制投影區域的性質
SCALE3 和 T3D 程序的執行，會改變座標轉換矩陣!P.T

```
;window,xsize=216,ysize=216
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0 ;設定顏色不分解，避免白色地方變成紅色
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
restore,'save_scale3_all.sav'   ;重新載入已儲存的變數
!P.T=tm     ;將重新載入的變數tm存入座標轉換矩陣!P.T
!X.S=xs & !Y.S=ys  & !Z.S=zs    ;將各重新載入變數分別存入各座標軸尺度變換向量
T3D,scale=[0.5,0.5,0.5]
T3D,rotate=[0,30,0]
T3D,translate=[0.2,0,0]
plots,x,y,z,/T3D
erase
T3D,/reset
plots,x,y,z,/T3D
z2=dist(256)
surface,z2,/SAVE   ;將執行後的座標系統存入轉換矩陣!P.T中
end
```

三維繪圖

PLOYLSHADE函數

語法	說明
Result = POLYSHADE(Vertex, Polygon) 或 Result = POLYSHADE(X,Y,Z, Polygon)	將輸入的頂點(Vertex)和多邊形建構成立體形狀，然後將算圖結果儲存至變數Result中

POLYSHADE函數的關鍵字

關鍵字	說明
/T3D	採用三維的座標系統
XSIZE=columns	定義二維影像的行長度
YSIZE=rows	定義二維影像的列長度
PLOY_SHADES=array	定義每個多邊形的顏色
SHADES=array	定義每個頂點的顏色

```
;window,xsize=216,ysize=216
!P.BACKGROUND=255  ;設定視窗上的背景顏色為白色
!P.COLOR=0         ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0 ;設定顏色不分解，避免白色地方變
成紅色
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
x=[0.2,0.8,0.8,0.5,0.2]
y=[0.2,0.2,0.6,0.8,0.6]
z=[0.0,0.0,0.0,0.0,0.0]
v=fltarr(3,5)
v[0,*]=x & v[1,*]=y & v[2,*]=z
p=[4,0,1,2,4,3,2,3,4]  ;連接多邊形頂點的數量和點位
d=[p[1:4],p[1]]  ;第一個多邊形所有頂點及第1個頂點
plots,x[d],y[d],/t3d
;xyouts,x[0],y[0],'0',charsize=2
;xyouts,x[1],y[1],'1',charsize=2
;xyouts,x[2],y[2],'2',charsize=2
;xyouts,x[4],y[4],'4',charsize=2

d=[p[6:8],p[6]] ;第二個多邊形所有頂點及第1個頂點
plots,x[d],y[d],/t3d
i=indgen(5)
str=string(i,'(I1)')
xyouts,x,y,str,size=2
image1=polyshade(v,p)
;tv,image1
v1=fltarr(3,4)
v1[0,*]=[x[0:2],x[4]]
v1[1,*]=[y[0:2],y[4]]
v1[2,*]=[z[0:2],z[4]]
p1=[4,0,1,2,3]
;將頂點v1和連接頂點資訊變數p所構成的遮
蔽多邊形，最後算圖的結果儲存至影像
image2變數中
image2=polyshade(v1,p1)
tv,image2

color = bindgen(5)*60
image3=polyshade(v,p,shades=color)
tv,image3
image4=polyshade(v,p,poly_shades=[100,200])
tv,image4
t3d,/reset
t3d,rotate=[60,0,0]
;t3d,translate=[0,0,0]
image5=polyshade(v,p,/t3d)
tv,image5
end
```

練習

請自訂四個頂點位置，繪製稜形圖，再標示出各頂點的標示編號，再將這稜形內部以淺灰色塗滿，但仍保留各頂點的標示編號。

MESH_OBJ程序的語法和關鍵字

語法	說明
MESH_OBJ, Type, Vertex, Polygon, Array1 [,Array2][,P1, P2, P3, P4, P5]	建立一個構成表面的多邊形網格組 Type: 表面形態 Vertex: 頂點位置 Polygon: 連接頂點的資訊 Array1, 和Array2: 各種表面不同的輸入陣列 P1,P2,P3,P4,P5: 依照各表面需求而定

MESH_OBJ, Type

Type	Surface Type
0	Triangulated 三角形表面
1	Rectangular 長方表面
2	Polar 極表面
3	Cylindrical 圓柱表面
4	Spherical 球表面
5	Extrusion 擠壓(凸出)表面
6	Revolution 旋轉表面
7	Ruled 直紋表面
Other values	None

MESH_OBJ,P1,P2,P3,P4,P5

Surface Type	Keywords
Triangulated	P1 through P5 are ignored.
Rectangular	If *Array1* is an (*n, m*) array, and if P1 has n elements, then the values contained in P1 are the X coordinates for each column of vertices. Otherwise, FINDGEN(*n*) is used for the X coordinates. If P2 has *m* elements, then the values contained in P2 are the Y coordinates for each row of vertices. Otherwise, FINDGEN(*m*) is used for the Y coordinates. The polygon facing is reversed if the order of either P1 or P2 (but not both) is reversed. P3, P4, and P5 are ignored.
Polar	P1 specifies the polar angle of the first column of *Array1* (the default is 0). P2 specifies the polar angle of the last column of *Array1* (the default is 2*PI). If P2 is less than P1 then the polygon facing is reversed. P3 specifies the radius of the first row of *Array1* (the default is 0). P4 specifies the radius of the last row of *Array1* (the default is *m*-1). If P4 is less than P3 then the polygon facing is reversed. P5 is ignored.

MESH_OBJ,P1,P2,P3,P4,P5

Surface Type	Keywords
Cylindrical	P1 specifies the polar angle of the first column of *Array1* (the default is 0). P2 specifies the polar angle of the last column of *Array1* (the default is 2*PI). If P2 is less than P1 then the polygon facing is reversed. P3 specifies the Z coordinate of the first row of *Array1* (the default is 0). P4 specifies the Z coordinate of the last row of *Array1* (the default is *m*-1). If P4 is less than P3 then the polygon facing is reversed. P5 is ignored.
Spherical	P1 specifies the longitude of the first column of *Array1* (the default is 0). P2 specifies the longitude of the last column of *Array1* (the default is 2*PI). IF P2 is less than P1 then the polygon facing is reversed. P3 specifies the latitude of the first row of *Array1* (the default is -PI/2). P4 specifies the latitude of the last row of *Array1* (the default is +PI/2). If P4 is less than P3 then the polygon facing is reversed. P5 is ignored.
Extrusion	P1 specifies the number of steps in the extrusion (the default is 1). P2 is a three element vector specifying the direction (and length) of the extrusion (the default is [0, 0, 1]). P3, P4, and P5 are ignored.

MESH_OBJ,P1,P2,P3,P4,P5

Surface Type	Keywords
Revolution	P1 specifies the number of “facets” in the revolution (the default is 3). If P1 is less than 3 then 3 is used. P2 is a three element vector specifying a point that the rotation vector passes through (the default is [0, 0, 0]). P3 is a three element vector specifying the direction of the rotation vector (the default is [0, 0, 1]). P4 specifies the starting angle for the revolution (the default is 0). P5 specifies the ending angle for the revolution (the default is 2*PI). If P5 is less than P4 then the polygon facing is reversed.
Ruled	P1 specifies the number of “steps” in the ruling (the default is 1). P2, P3, P4, and P5 are ignored.

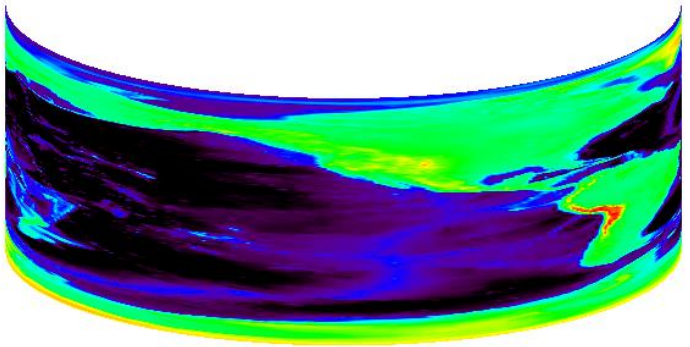
```
;window,xsize=216,ysize=216
!P.BACKGROUND=255  ;設定視窗上的背景顏色為白色
!P.COLOR=0         ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0 ;設定顏色不分解，避免
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[-2,2]
array1=replicate(2.0,360,360)
 ; 2.0 is the radius of cylinder or sphere
mesh_obj,4,v,p,array1      ;type=4 for spherical
image1=polyshade(v,p,/t3d)
tv,image1
mesh_obj,3,v,p,array1, P4=-3;,p3=0
  ;type=3 for cylindrical, p4=3 for cylinder length
; If P4  < P3 then the polygon facing is reversed
subdir=['examples','data']
path=filepath('worldelv.dat',subdirectory=subdir)
file=read_binary(path,data_dims=[360,360])
image2=polyshade(v,p,shades=file,/t3d)
tv,image2
array2=[[2,2],[0,0],[0,3]]
help,array2
array2=transpose(array2)
help,array2
mesh_obj,6,va,pa,array2,P1=180, $
P3=[0,0,1],p5=!pi*3./2.;,p4=!pi*3./2.
;mesh_obj,6,va,pa,[[2,0,0],[2,0,3]], $
   P1=180,P3=[0,0,1] ,p5=!pi*3./2.;,p4=!pi*3./2.
   ; type=6 for revolution(旋轉表面),
   ; P1=180表示用180個小平面構成圓柱,
   ;P3=[0,0,1]表示對Z軸旋轉
image3=polyshade(va,pa,/t3d)
tv,image3

array4=[[0.5,0.5],[0,0],[0,0.5]]
mesh_obj,0,va2,pa2,array4
image4=polyshade(va2,pa2,/t3d)
tv,image4

py=findgen(4)
array5=[[0.5,0.5],[0,0.5],[0,0],[0.5,0]]
mesh_obj,1,va3,pa3,array5;,p1=py
image5=polyshade(va3,pa3,/t3d)
tv,image5
END
```

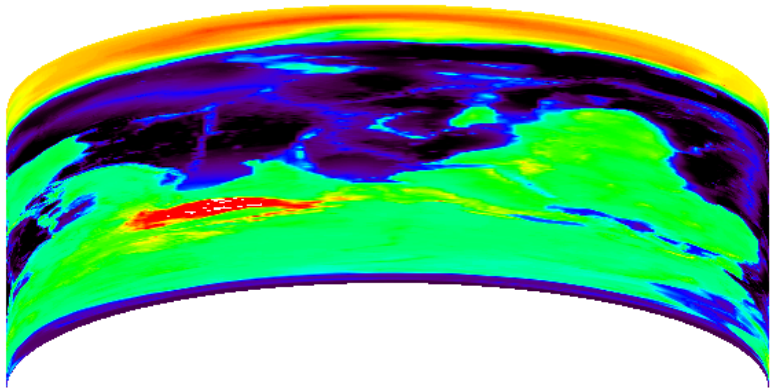
```
!P.BACKGROUND=255
!P.COLOR=0
DEVICE,DECOMPOSED=0
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[-2,2]

array1=fltarr(3,5)
array1[0,*]=randomu(seed,5)
array1[1,*]=randomu(seed,5)
array1[2,*]=randomu(seed,5)
help,array1
mesh_obj,0,v,p,array1
help,v,p
image_1=polyshade(v,p,/T3D)
help,image_1
TV,image_1

array2=[[0.5,0.5],[0,0.5],[0,0],[0.5,0]]
py=findgen(4)
mesh_obj,1,v1,p1,array2,p1=py
image_2=polyshade(v1,p1,/T3D)
TV,image_2

;array3=replicate(2.0,360,360)
array3=randomu(seed,5,5)
mesh_obj,2,v2,p2,array3;,p2=!pi
image_3=polyshade(v2,p2,/T3D)
TV,image_3

array4=replicate(2.0,360,360)
mesh_obj,3,v3,p3,array4,P3=0,P4=3
image_4=polyshade(v3,p3,/T3D)
TV,image_4

mesh_obj,4,v4,p4,array4,p3=-
180,p4=180
image_5=polyshade(v4,p4,/T3D)
TV,image_5
```

接上頁

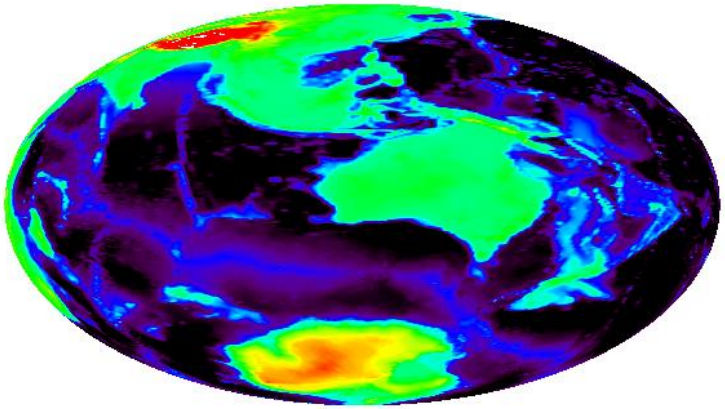
```
mesh_obj,5,v5,p5,array1
image_6=polyshade(v5,p5,/T3D)
TV,image_6

mesh_obj,6,v6,p6,array1,p1=300,p2=[0,0,20],p3=[0,0,1],p4=!pi/2,p5=!pi*3./2.
image_7=polyshade(v6,p6,/T3D)
TV,image_7

array7=randomu(seed,3,4)
mesh_obj,7,v7,p7,array1,array7
image_8=polyshade(v7,p7,/T3D)
TV,image_8
end
```

```
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[-2,2]
array1=replicate(2.0,360,360)
;mesh_obj,3,v,p,array1,P3=0, P4=-3
mesh_obj,3,v,p,array1,p3=0,P4=3
;type=3 for cylindrical,
;P1 --  the polar angle of the first column of Array1 (the default is 0).
;P2 --  the polar angle of the last column of Array1 (the default is 2*PI).
; p4=3 for cylinder length
; If P4  < P3 then the polygon facing is reversed
subdir=['examples','data']
path=filepath('worldelv.dat',subdirectory=subdir)
file=read_binary(path,data_dims=[360,360])
image2=polyshade(v,p,shades=file,/t3d)
loadct,39  & tv,image2
;loadct,0 & tv,image2
end
```





```
!P.BACKGROUND=255  ;設定視窗上的背景顏色為白色
!P.COLOR=0         ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[-2,2]
array1=replicate(2.0,360,360)
;mesh_obj,4,v,p,array1,p1=-!pi,p2=!pi,p3=-!pi,p4=!pi
;;type=4 for spherical
;P1 -- the longitude of the first column of Array1 (the default is 0).
;P2 -- the longitude of the last column of Array1 (the default is 2*PI).
;P3 -- the latitude of the first row of Array1 (the default is -PI/2).
;P4 -- the latitude of the last row of Array1 (the default is +PI/2).
;mesh_obj,4,v,p,array1,p1=0,p2=2.*!pi,p3=-!pi,p4=!pi
mesh_obj,4,v,p,array1,p1=!pi,p2=-!pi,p3=-!pi,p4=!pi
subdir=['examples','data']
path=filepath('worldelv.dat',subdirectory=subdir)
file=read_binary(path,data_dims=[360,360])
image2=polyshade(v,p,shades=file,/t3d)
loadct,39  & tv,image2
;loadct,0  & tv,image2
end
```

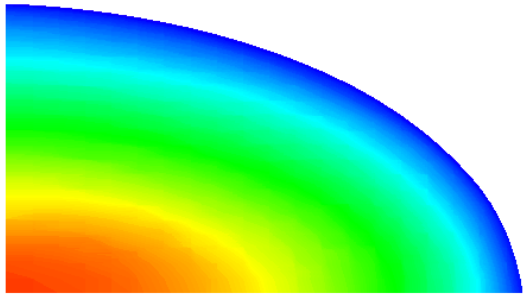


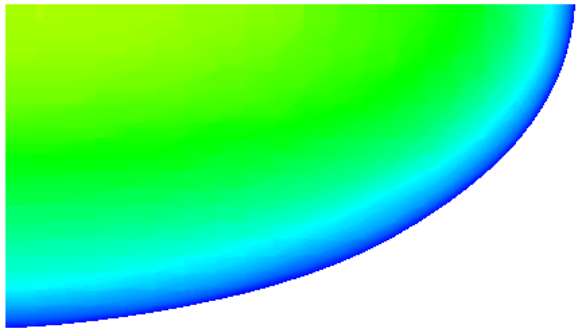
SHADE_VOLUME程序的語法和關鍵字

語法	說明
SHADE_VOLUME, Volume, Value, Vertex, Polygon	求出構成等值面的頂點和多邊形 Type:體資料 Value:等值面的數值 Vertex: 頂點的位置 Polygon:頂點連接的資訊

關鍵字	說明
/LOW	宣告演算等值面內側部分

```
;window,xsize=216,ysize=216
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
x=randomu(seed,100)
y=randomu(seed,100)
z=randomu(seed,100)
f=sqrt(x^2+y^2+z^2)
vol=GRID3(x,y,z,f)  ;將不規則的網格點轉換為規則網格點
s=size(vol)
scale3,xrange=[0,s[1]],yrange=[0,s[2]],AX=90,zrange=[0,s[3]],AZ=0
shade_volume,vol,0.8,v,p
image1=polyshade(v,p,/T3D)
tv,image1

scale3,xrange=[0,s[1]],yrange=[0,s[2]],ax=-90,zrange=[0,s[3]],az=0
shade_volume,vol,0.8,v2,p2,/LOW
image2=polyshade(v2,p2,/T3D)
tv,image2
end
```





SET_SHADING程序的語法和關鍵字

語法	說明
SET_SHADING	改變著色方式和效果

關鍵字	說明
/GOURAUD	使用Gouraud著色法
LIGHT=[x,y,z]	定義光線方向
VALUES=[darkest, brightest]	定義光線的明暗

會影響SHADE_SURF程序和POLYSHADE函數的輸出結果

```
window,xsize=216,ysize=216
!P.BACKGROUND=255   ;設定視窗上的背景顏色為白色
!P.COLOR=0          ;設定圖形是以黑色線條呈現
DEVICE, DECOMPOSED=0  ;設定顏色不分解，避免白色地方變成紅色
plot,[0,1],[0,1],/nodata,xstyle=4,ystyle=4
scale3,xrange=[-2,2],yrange=[-2,2],zrange=[-2,2]
array1=replicate(2.0,360,360)
mesh_obj,4,v,p,array1       ;type=4 for spherical

set_shading, light=[0,0,1];,values=[150,250]
image1=POLYSHADE(v,p,/T3D)
tv,image1
END
```