第 3 章　Matlab 绘图及视频读写

司守奎
烟台市，海军航空大学
Email：sishoukui@163.com
微信：sishoukui

3.1　散点图

1. 基于 scatter 函数的二维散点图

给定平面上的 n 个不同点的直角坐标 (x_i, y_i) $(i=1,2,\cdots,n)$，两个坐标分量组成的向量分别用向量 $\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^T$ 和 $\boldsymbol{y}=[y_1,y_2,\cdots,y_n]^T$ 表示。使用 scatter 函数绘制散点图，常用的 3 种格式如下：

scatter(x, y)，scatter(x, y, sz)，scatter(x, y, sz, c)

每个离散点默认用圆圈表示。在第二种格式中，sz 表示圆圈的大小，若 s 为标量，则所有圆圈大小相同；若 sz 为 n 维向量，则其分量值越大，圆圈越大。在第三种格式中，c 表示颜色，当 c 为 n 维向量时，其分量取值越大，对应圆圈的颜色越红，反之越蓝；当 c 为 1×3 维向量时，它的分量分别表示 R、G、B 的值，这里 RGB 分别表示 3 种颜色 red、green、blue。

例 3.1　seamount.mat 是 Matlab 自带的某海山数据，其中向量 x 表示 294 个点的纬度（单位为°），y 向量表示经度（单位为°），z 是取值为负的深度向量（单位为 m）。绘出平面散点图（不考虑深度），圆圈颜色用深度向量 z 来表示。

为了对比，我们绘制了两种散点图如图 3.1 所示。

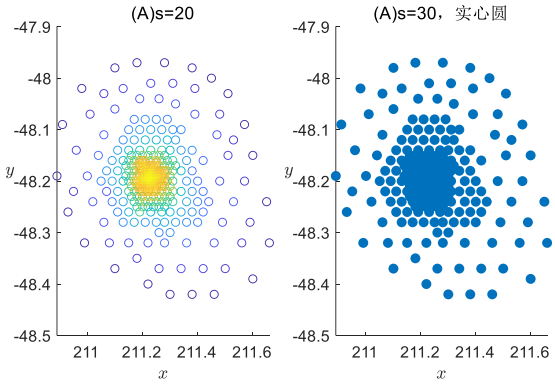


图 3.1　seamount 的平面散点图

```
clc, clear
load seamount          %加载 Matlab 内置文件 seamount.mat
subplot(121), scatter(x,y,20,z)   %圆圈大小为 20
title('(A)s=20'), xlabel('$x$','Interpreter','Latex')
ylabel('$y$','Interpreter','Latex','Rotation',0)
subplot(122), scatter(x,y,30,'filled')   %圆圈大小为 50 且为实心
title('(A)s=30，实心圆'), xlabel('$x$','Interpreter','Latex')
ylabel('$y$','Interpreter','Latex','Rotation',0)
```

2. 基于 scatter3 函数的三维散点图

在三维空间中，n 个点构成的横坐标、纵坐标和竖坐标向量分别为 x, y, z, 则绘制散点图的函数为 scatter3，它的常用格式为

scatter3(x, y, z)，scatter3(x, y, z, sz)，scatter3(x, y, z, sz, c)

其中 sz 和 c 的意义与 scatter 中的意义相同。

例 3.2　绘制 seamount.mat 的三维空间散点图。

绘制的三维散点图如图 3.2 所示。

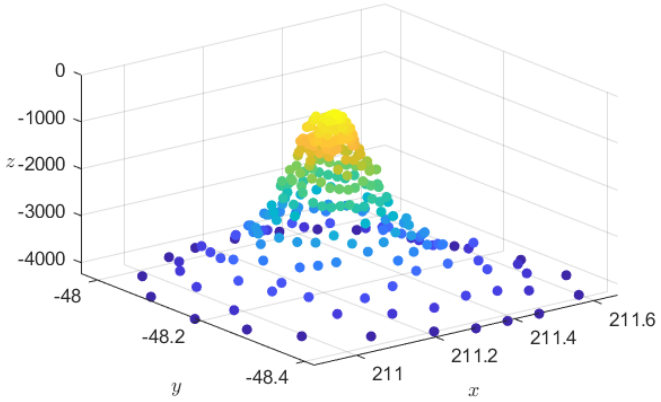


图 3.2　seamount.mat 的三维散点图

```
clc, clear
load seamount      %加载 Matlab 内置文件 seamount.mat
scatter3(x,y,z,30,z,'filled')
xlabel('$x$','Interpreter','Latex'), ylabel('$y$','Interpreter','Latex')
zlabel('$z$','Interpreter','Latex','Rotation',0)
```

例 3.3　将半径分别为 1、0.75、0.5，球心均在原点的球面离散化为 20×20 的网格点，并绘制出三维散点图，其中三类点的颜色值分别用 3、2、1 表示，离散点的正方形大小分别为 30、20、10。

所画的三维散点图如图 3.3 所示。

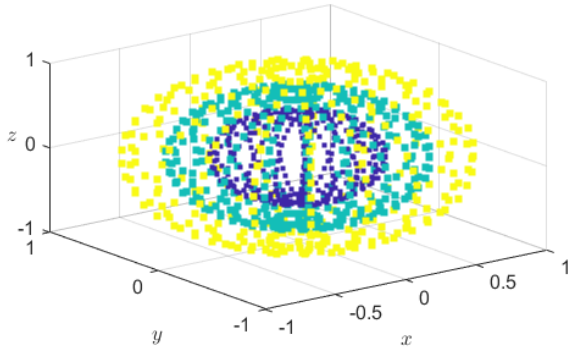


图 3.3　三维散点图

```
clc, clear, N=20;
%产生单位球面上的 N×N 网格点(对球面坐标的角度等间隔离散化）
[X,Y,Z]=sphere(N-1);
x=X(:); y=Y(:); z=Z(:);     %矩阵展开成列向量
x=[x;0.75*x;0.5*x];          %3 个球面上 x 坐标合并在一起
y=[y;0.75*y;0.5*y]; z=[z;0.75*z;0.5*z];
sz=[ones(N^2,1)*30;ones(N^2,1)*20;ones(N^2,1)*10];
c=[ones(N^2,1)*3;ones(N^2,1)*2;ones(N^2,1)];
scatter3(x,y,z,sz,c,'filled','Marker','s')   %用正方形表示离散点
xlabel('$x$','Interpreter','Latex'), ylabel('$y$','Interpreter','Latex')
```

zlabel('z','Interpreter','Latex','Rotation',0)

注 3.1　scatter 和 scatter3 默认用圆圈表示点，当然也可以用三角形、正方形、菱形、五角星和六角星等封闭符号来表示。若在 scatter 和 scatter3 中使用了 filled，则不建议使用非封闭的加号、黑点、叉号和星号等符号。

3.基于 gplotmatrix 和 gscatter 函数的散点图绘制

gplotmatrix 可以绘制分组数据变量对之间的散点图；gscatter 绘制分组数据的散点图。

例 3.4　fisheriris 数据集(fisheriris.csv 或 fisheriris.mat)由 Fisher 于 1936 收集整理。数据集包含 150 条数据，分为 3 类，每类 50 条数据，每条数据包含 4 个属性和一个类别标签值，数据格式如**表** 3.1 所示。

表 3.1　fisheriris **数据集数据（全部数据见数据文件** fisheriris.csv **或** fisheriris.mat）

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
⋮	⋮	⋮	⋮	⋮	⋮
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3	5.1	1.8	virginica

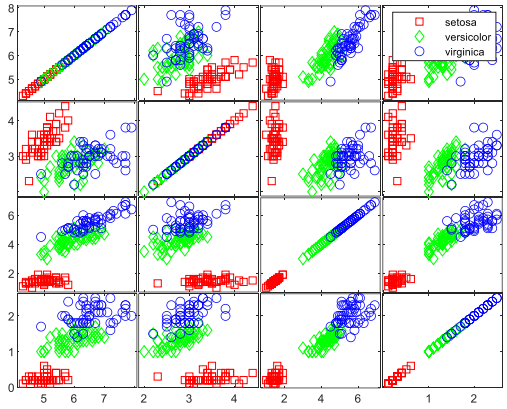


图 3.4　**根据** species **绘制的** 4 **个属性两两之间的散点图矩阵**

绘制的 4 个属性对之间的散点图如图 3.4 所示，粗略观察散点图后我们可以看出，setosa 类与其他两类非常不同。与此相反，其余两类在所有散点图中都存在大量重合。使用第 3 个属性和第 4 个属性的散点图重合较小。只使用第 3 个属性和第 4 个属性画出的散点图如图 3.5 所示。

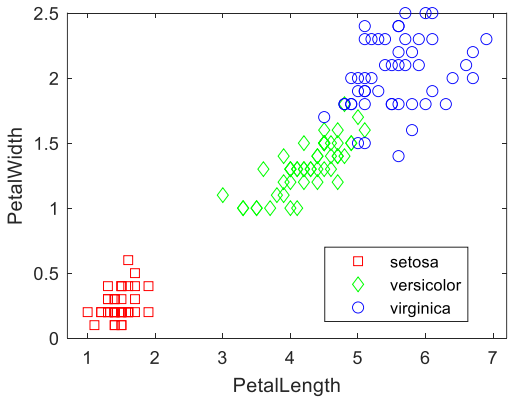


图 3.5　**根据属性** PetalLength **和** PetalWidth **绘制的散点图**

```
clc, clear, close all
load fisheriris
tabulate(species)    %频数表
```

```
gplotmatrix(meas,meas,species,'rgb','sdo')
figure, gscatter(meas(:,3),meas(:,4),species,'rgb','sdo')
xlabel('PetalLength'), ylabel('PetalWidth')
```

3.2 基于 plot 函数的散点图和平面曲线绘制

已知二维平面上的 n 个点 $(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$，构成的 x 坐标和 y 坐标向量分别为 x,y,把这 n 个点按照先后顺序用线段相连，就得到过这 n 个点的折线图。Matlab 无法绘制真正意义上的曲线，实际绘制的都是折线图；如果相邻两点之间不连线段，则绘制出散点图。

plot 函数是绘制二维图形的最基本函数，它是针对向量或矩阵的列来绘制曲线的，可以绘制线段和曲线。函数 plot 的最典型调用方式是三元组形式：

```
plot(x, y, 'Color|Linestyle|Marker')
```

其中 x，y 为同维数的向量（或矩阵），x 作为点的横坐标，y 作为点的纵坐标，plot 命令用直线连接相邻两数据点绘制图形。Color、Linestyle 和 Marker 分别是颜色、线型和数据点标记，它们之间没有先后顺序之分。

常用的颜色、线型和数据点符号如**错误!未找到引用源。**所示。

表 3.2　颜色、线型、数据点符号

颜色符号	颜色	线型符号	线型	数据点符号	标记
b（默认）	蓝色	-（默认）	实线	+	十字
r	红色	:	短虚线	*	星号
y	黄色	--	长虚线	o	圆圈
g	绿色	-.	点划线	x	叉号
c	蓝绿色			s	正方形
m	紫红色			d	菱形
k	黑色			p	五角星
w	白色			h	六角形

画二维曲线图时，当知道曲线的函数表达式时，可以使用 2 种方式画图：

（1）用描点画图命令 plot；

（2）用函数画图命令 fplot。

例 3.5（续**例** 3.1） 分别使用 plot 和 plot3 命令绘制数据集 seamount.mat 的散点图，其中 plot 不考虑深度 z。

绘制的散点图如图 3.6 所示。

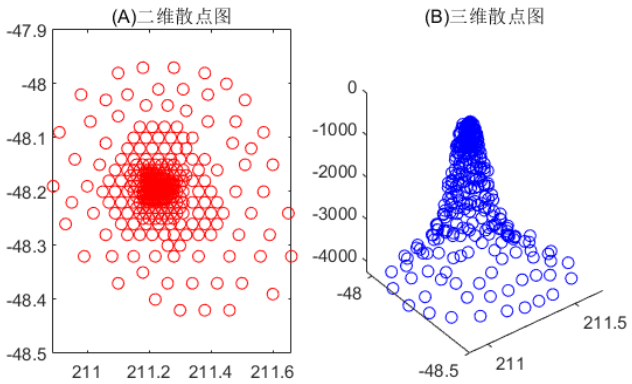


图 3.6　基于 plot 和 plot3 的 seamount 散点图

```
clc, clear, close all
load seamount
```

```
subplot(121), plot(x,y,'ro')        %绘制二维散点图
title('(A)二维散点图')
subplot(122), plot3(x,y,z,'bo')     %绘制三维散点图
title('(B)三维散点图')
```

例 3.6　绘制单位圆 $x^2+y^2=1$。

单位圆的参数方程为

$$\begin{cases} x=\cos t, \\ y=\sin t, \end{cases} \quad t\in[0,2\pi].$$

画出的单位圆如图 3.7 所示。

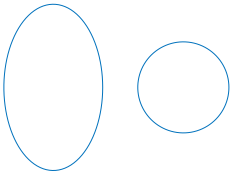


图 3.7　每个轴单位长度不等和相等情形下单位圆图形对比

```
clc, clear, close all
t=linspace(0,2*pi);                          %等间距取 100 个值
x=cos(t); y=sin(t)
subplot(121), plot(x,y), axis off            %不显示坐标轴
subplot(122), plot(x,y), axis equal off      %等价于 axis equal, axis off
```

例 3.7　画出单位圆的内接正 8 边形。

所画出的图形如图 3.8 所示。

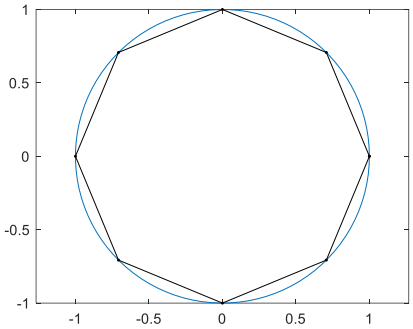


图 3.8　单位圆的内接正 8 边形

```
clc, clear, close all
t1=linspace(0,2*pi);          %等间距取 100 个值
x1=cos(t1); y1=sin(t1);
plot(x1,y1)                   %画单位圆
hold on                       %图形保持
t2=linspace(0,2*pi,9)         %等间距取 9 个不同点，0 和 2*pi 对应的点重合
x2=cos(t2); y2=sin(t2);
plot(x2,y2,'.k-'), axis equal
```

3.3　三维绘图命令

MATLAB 也提供了一些三维基本绘图命令，如三维曲线命令 plot3，三维网格图命令 mesh 和三维表面图命令 surf。

plot3(x,y,z)通过描点连线画出曲线，这里 x,y,z 都是 n 维向量，分别表示该曲线上点集的横坐标、纵坐标、竖坐标。

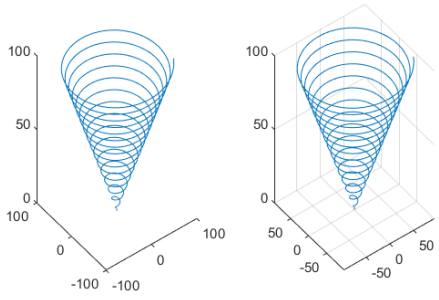
命令 mesh(x,y,z)画网格曲面。这里 x,y,z 分别表示数据点的横坐标、纵坐标、竖坐标，如果 x 和 y 是向量，x 是 m 维的向量，y 是 n 维的向量，则 z 是 n×m 的矩阵。x,y,z 也可以都是同维数的矩阵. 命令 mesh(x,y,z)将该数据点在空间中描出，并连成网格。

命令 surf(x,y,z)画三维表面图，这里 x,y,z 分别表示数据点的横坐标、纵坐标、竖坐标。

已知曲线或曲面的函数关系，提倡使用 fplot3，fzmesh，fsurf 等命令画图。

三维空间隐函数绘图命令为 fimplicit3。

例 3.8　画出三维螺旋线 $x=t\cos t$， $y=t\sin t$， $z=t$ 的图形。



（A）plot3 画图　（B）fplot3 画图

图 3.9　plot3 **和** fplot3 **画图对比**

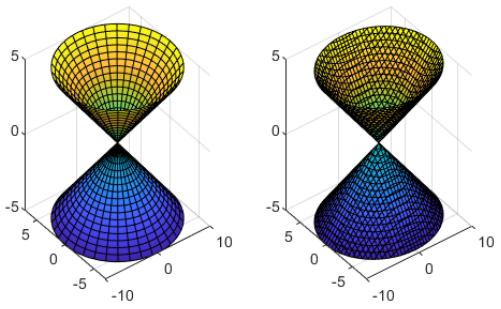
用 plot3 和 fplot3 绘制的图形如图 3.9 所示。

```
clc, clear, close all
t=0:0.01:100; x=t.*cos(t); y=t.*sin(t);
subplot(121), plot3(x,y,t)
x=@(t)t.*cos(t); y=@(t)t.*sin(t);
z=@(t)t; subplot(122), fplot3(x,y,z,[0,100])
```

例 3.9　绘制椭圆锥面 $\frac{x^2}{4}+\frac{y^2}{2}=z^2$ 的网格曲面图。

它的参数方程为 $\begin{cases} x=2z\cos t, \\ y=\sqrt{2}z\sin t, \\ z=z. \end{cases}$

用参数方程绘图和隐函数直接绘图得到的图形如图 3.10 所示。



(A) fmesh 绘图　（B）fimplicit3 绘图

图 3.10　参数方程和隐函数绘图比较

```
clc, clear, close all
subplot(121), x=@(t,z)2*z.*cos(t); y=@(t,z)sqrt(2)*z.*sin(t);
z=@(t,z)z; fsurf(x,y,z,[0,2*pi,-5,5]) , title('')
subplot(122), fimplicit3(@(x,y,z)x.^2/4+y.^2/2-z.^2,[-10,10,-10,10,-5,5])
```

例 3.10　莫比乌斯带是一种拓扑学结构，它只有一个面和一个边界，是 1858 年由德国数学家、天文学家莫比乌斯和约翰·李斯丁独立发现的。其参数方程为

$$\begin{cases} x=\left(2+\dfrac{s}{2}\cos\dfrac{t}{2}\right)\cos t, \\ y=\left(2+\dfrac{s}{2}\cos\dfrac{t}{2}\right)\sin t, \\ z=\dfrac{s}{2}\sin\dfrac{t}{2}, \end{cases}$$

其中，$-1\le s\le 1$，$0\le t\le 2\pi$。绘制莫比乌斯带。

绘制的图形如图 3. 11 所示。

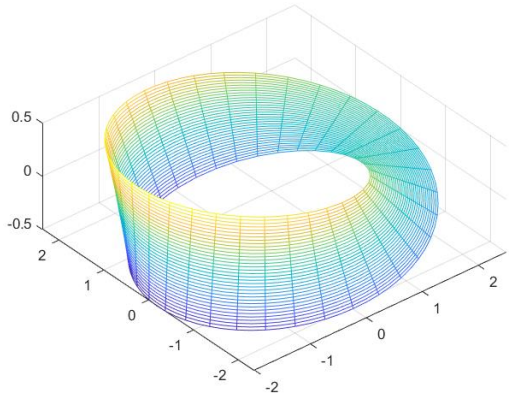


图 3.11　莫比乌斯带

```
clc, clear, close all
x=@(s,t)(2+s/2.*cos(t/2)).*cos(t);
y=@(s,t)(2+s/2.*cos(t/2)).*sin(t);
z=@(s,t)s/2.*sin(t/2);
fmesh(x,y,z,[-1,1,0,2*pi]), title('')
view(-40,60)     %设置视角
```

例 3.11　已知平面区域 $0\le x\le 1400$，$0\le y\le 1200$ 步长间隔为 100 的网格节点高程数据见表 3.3(单位：m)。

表 3.3　高程数据表

y/x	0	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400

1200	1350	1370	1390	1400	1410	960	940	880	800	690	570	430	290	210	150
1100	1370	1390	1410	1430	1440	1140	1110	1050	950	820	690	540	380	300	210
1000	1380	1410	1430	1450	1470	1320	1280	1200	1080	940	780	620	460	370	350
900	1420	1430	1450	1480	1500	1550	1510	1430	1300	1200	980	850	750	550	500
800	1430	1450	1460	1500	1550	1600	1550	1600	1600	1600	1550	1500	1500	1550	1550
700	950	1190	1370	1500	1200	1100	1550	1600	1550	1380	1070	900	1050	1150	1200
600	910	1090	1270	1500	1200	1100	1350	1450	1200	1150	1010	880	1000	1050	1100
500	880	1060	1230	1390	1500	1500	1400	900	1100	1060	950	870	900	936	950
400	830	980	1180	1320	1450	1420	400	1300	700	900	850	810	380	780	750
300	740	880	1080	1130	1250	1280	1230	1040	900	500	700	780	750	650	550
200	650	760	880	970	1020	1050	1020	830	800	700	300	500	550	480	350
100	510	620	730	800	850	870	850	780	720	650	500	200	300	350	320
0	370	470	550	600	670	690	670	620	580	450	400	300	100	150	250

（1）画出该区域的等高线。

（2）画出该区域的三维表面图。

画等高线及三维表面图的 MATLAB 程序如下：

```
clc, clear, close all
a=load('data3_11.txt');
x0=0:100:1400; y0=1200:-100:0;
subplot(1,2,1), c=contour(x0,y0,a,6); clabel(c)  %画 6 条等高线，并标注等高线
title('等高线图')
subplot(1,2,2), surf(x0,y0,a), title('三维表面图')
```

所画出的图形见图 3.12。

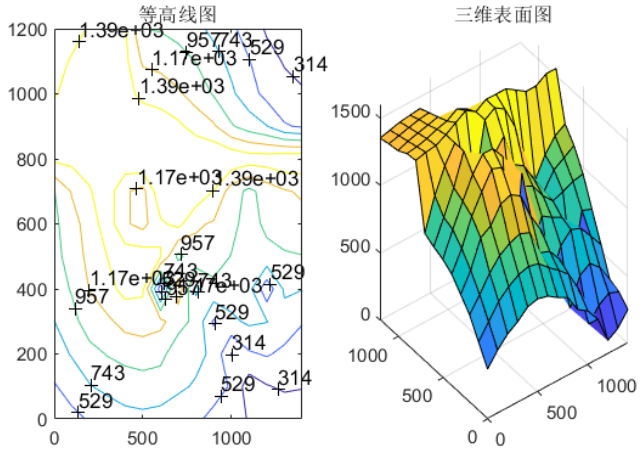


图 3.12　等高线图及三维表面图

例 3.12　绘制下列参数方程表示的曲面。

$$\begin{cases} x = e^{-|u|/10}\sin(5|v|), \\ y = e^{-|u|/10}\cos(5|v|), \\ z = u. \end{cases}$$

```
clc, clear, close all
x = @(u,v) exp(-abs(u)/10).*sin(5*abs(v));
y = @(u,v) exp(-abs(u)/10).*cos(5*abs(v));
z = @(u,v) u; fsurf(x,y,z)
```

绘制的曲面如图3.13所示。

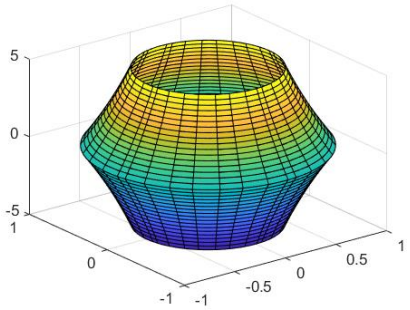


图 3.13　参数方程表示的曲面

例 3.13　S 型曲面由下面两部分组成：

$$\begin{cases} x=\cos t, \\ 0\le y\le 5, t\in[-\pi,\pi/2] \\ z=\sin t, \end{cases} 和 \begin{cases} x=-\cos t, \\ 0\le y\le 5, \quad t\in[\pi/2,-\pi]. \\ z=2-\sin t, \end{cases}$$

（1）绘制 S 形曲面的图形。

（2）绘制 S 形曲面的图形，要求相同的 t 值，颜色相同。

解　S 型曲面第一部分的参数方程为

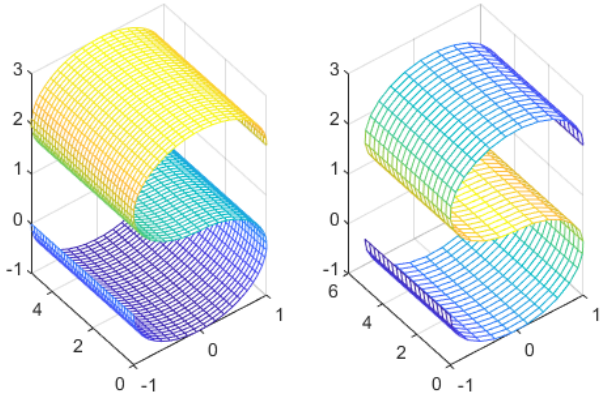
$$\begin{cases} x=\cos t, \\ y=y, \qquad t\in[-\pi,\pi/2], y\in[0,5]. \\ z=\sin t, \end{cases}$$

类似地可写出第二部分曲面的参数方程。

（1）利用 fmesh 直接绘制 S 形曲面的两个组成部分即可。

（2）对于每部分曲面，先分别使用 meshgrid 产生网格矩阵 T 和 Y，并计算矩阵 X 和 Z；再将两个曲面对应的 X、Y 和 Z 矩阵合并，进而利用 mesh 绘出整个网格曲线。

绘出的图形如图 3.14 所示。



（A）问题（1）图形　　（B）问题（2）图形

图 3.14　S 形曲面图形

```
clc, clear, close all
x=@(t,y)cos(t); y=@(t,y)y; z=@(t,y)sin(t);
subplot(121), fmesh(x,y,z,[-pi,pi/2,0,5])
x2=@(t,y)-cos(t); z2=@(t,y)2-sin(t)
hold on, fmesh(x2,y,z2,[-pi,pi/2,0,5])

t=linspace(-pi,pi/2,20); y=linspace(0,5,20);
```

```
[T,Y]=meshgrid(t,y);
X=cos(T); Z=sin(T);
t2=linspace(pi/2,-pi,20);
[T2,Y2]=meshgrid(t2,y);
X2=-cos(T2); Z2=2-sin(T2);
subplot(122), mesh([X,X2],[Y,Y2],[Z,Z2],[T,T2])
```

3.4 万花筒曲线

万花筒曲线可看成一条曲线绕另一条封闭曲线转动，前一条曲线上某点形成的轨迹。本节介绍各种圆形螺纹齿状万花筒花纹。

1. 内切圆

例 3.14 假设有两个不同的圆，第一个圆（大圆）的半径为 R，曲线方程为 $x^2+y^2=R^2$；第二个圆（小圆）的半径为 r，曲线方程为 $[x-(R-r)]^2+y^2=r^2$，其中 $r<R$。两个圆相切于点 $Q(R,0)$，现让小圆绕大圆按顺时针转动，求小圆上某个固定点在转动过程中形成的轨迹。

解 在初始状态，小圆的圆心 O_1 的坐标为 $(R-r,0)$，不失一般性地，在小圆上选取定点 P，其坐标为 $(R-2r,0)$。在转动过程中，O_1 的轨迹形成半径为 $R-r$ 的圆，对应的曲线方程为 $x^2+y^2=(R-r)^2$，该圆的参数方程为

$$\begin{cases} x=(R-r)\cos t, \\ y=(R-r)\sin t, \end{cases} \quad t\geq 0,$$

其中，$t=0$ 时的坐标为初始小圆的圆心坐标。

在初始位置（$t=0$），点 P 到切点 Q、按顺时针转动的圆弧长度为 $L=\left|PQ\right|=\pi r$。现在让小圆的圆心绕大圆按逆时针转动角度 Δt（单位为 rad），此时大圆上切点移动的长度为 $s=R\Delta t$，如图 3.15 所示。

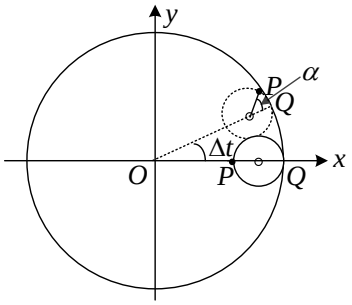


图 3.15 两圆内切示意图

注意：选取的 Δt 为比较小的正数，且满足 $s<\pi r$。因此点 P 到 Q 在小圆上按顺时针方向的长度为 $L-s$，弧 PQ 对应的角度为 $\alpha=(L-s)/r$。设小圆的圆心坐标为 $(x_{\text{center}},y_{\text{center}})$，其中，$x_{\text{center}}=(R-r)\cos\Delta t$，$y_{\text{center}}=(R-r)\sin\Delta t$，则点 P 对应的坐标为

$$\begin{cases} x=x_{\text{center}}+r\cos(\Delta t+\alpha), \\ y=y_{\text{center}}+r\sin(\Delta t+\alpha). \end{cases}$$

接着讨论小圆的圆心再次按逆时针转动角度 Δt，此时需要更新点 P 到 Q 在小圆上按顺时针方向的长度

$$L:=\begin{cases} L-s, & L-s>0, \\ 2\pi r+L-s, & L-s\leq 0, \end{cases}$$

其中 s 为常数，不需要更新。

在实际绘图时，取 $R=1$。绘图过程按如下三步进行：

（1）绘制大圆的 Matlab 程序如下：

```
clc, clear, close all
```

```
R=1; theta=0:0.01:2*pi;
x=R*cos(theta); y=R*sin(theta);
plot(x,y,'r-','LineWidth',3)          %线的粗细设置为 3
axis equal off, hold on
```

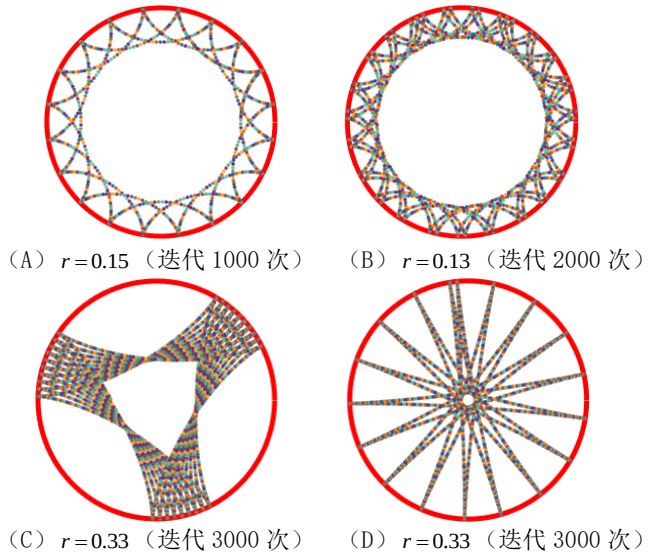
（2）在仿真中取 $r=0.15$，$\Delta t=0.02$，参数设置与初始化的程序如下：

```
r=0.15; delta=0.02;
t0=0;                                 %t 的初始值
xy=[R-2*r,0];                         %用于存储 P 点的坐标
L=pi*r; s=R*delta;
```

（3）以 Δt 为时间间隔，循环计算 P 点坐标，并绘出此点。程序如下：

```
for i=1:1000                          %迭代 1000 次
    x_center=(R-r)*cos(t0+delta);     %小圆圆心的 x 坐标
    y_center=(R-r)*sin(t0+delta);     %小圆圆心的 y 坐标
    alpha=(L-s)/r;
    x_new=x_center+r*cos(t0+delta+alpha);   %更新小圆上 P 点的 x 坐标
    y_new=y_center+r*sin(t0+delta+alpha);   %更新小圆上 P 点的 y 坐标
    xy=[xy;x_new,y_new];
    plot(x_new,y_new,'.')                   %绘制 P 点
    t0=t0+delta;                      %更新 t
    if L-s>=0
        L=L-s;                        %更新 L
    else
        L=2*pi*r+(L-s);
    end
end
```

四种 r 的取值对应的图形如图 3.16 所示。



（A）$r=0.15$（迭代 1000 次）　（B）$r=0.13$（迭代 2000 次）

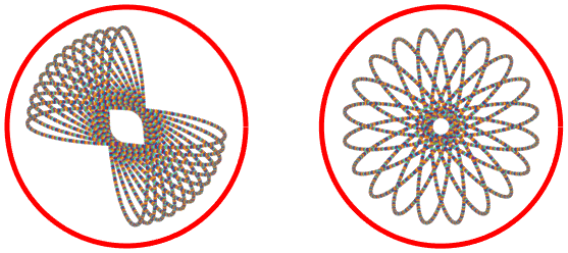
（C）$r=0.33$（迭代 3000 次）　（D）$r=0.33$（迭代 3000 次）

图 3.16　不同 r 值下的万花筒曲线图（内切圆）

在上述仿真中，点 P 在小圆上。下面考虑小圆域内某点 G 在运动中形成的轨迹。不妨设点 G 的初始坐标为 $(R-r-\rho r,0)$，其中参数 $\rho\in[0,1)$。在运动过程中，点 G、点 P 和小圆圆心三点共线。在绘制点 G 轨迹的程序中，只需要更改 x_new 和 y_new 的迭代公式即可：

```
x_new=x_center+rho*r*cos(t0+delta+alpha);
y_new=y_center+rho*r*sin(t0+delta+alpha);
```

在仿真中，取 $\rho=2/3$，迭代次数为 4000，两种 r 取值下的图形如图 3.17 所示。



（A） $r=0.51$　　（B） $r=0.55$

图 3.17　不同 r 值下小圆域内 G 点的轨迹

2. 外切圆

例 3.15　假设有两个圆，第一个圆（内圆）的方程为 $x^2+y^2=R^2$，第二个圆（外圆）的方程为 $[x-(R+r)]^2+y^2=r^2$。这两个圆外切于点 $Q(R,0)$，选取外圆的一点 P，其坐标为 $(R+2r,0)$，如图 3.18 所示。让外圆按逆时针绕内圆旋转，求点 P 形成的轨迹。

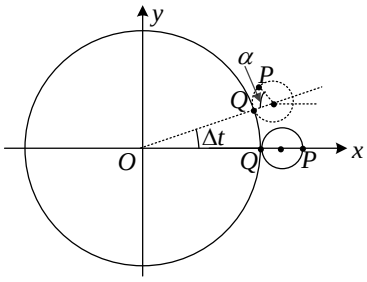


图 3.18　两圆外切示意图

解　外圆圆心所在的曲线是半径为 $R+r$ 的圆，参数方程为

$$\begin{cases} x=(R+r)\cos t, \\ y=(R+r)\sin t, \end{cases} \quad t\geq 0 .$$

在初始情形下，角度 t 对应的值为 0。当外圆的圆心对应的角度为 Δt 时，圆心坐标记为 $(x_{\text{center}},y_{\text{center}})$，其中，$x_{\text{center}}=(R+r)\cos\Delta t$，$y_{\text{center}}=(R+r)\sin\Delta t$。令 $L=\overset{\frown}{PQ}=\pi r$，$s=R\Delta t$，$\alpha=(L-s)/r$，于是点 P 的坐标为

$$\begin{cases} x=x_{\text{center}}+(R+r)\cos(\Delta t+\pi-\alpha), \\ y=y_{\text{center}}+(R+r)\sin(\Delta t+\pi-\alpha). \end{cases}$$

L 和 t 的更新公式与内切圆情形相同。

在仿真中取 $R=1$，$r=0.15$，$\Delta t=0.02$，绘制点 P 轨迹的 Matlab 程序如下：

（1）初始化参数及内圆曲线的绘制，程序如下：

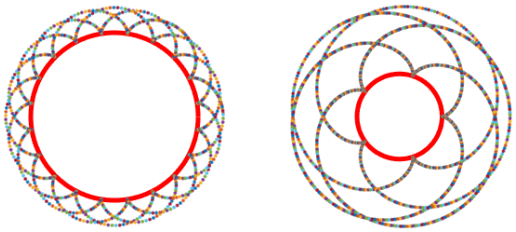
```
clc, clear, close all
R=1; r=0.15; delta=0.02;
theta=0:0.01:2*pi;
x=R*cos(theta); y=R*sin(theta);
plot(x,y,'r-','LineWidth',3)          %线的粗细设置为 3
axis equal off, hold on
```

（2）绘制 P 点的轨迹，程序如下：

```
t0=0;                                 %t 的初始值
L=pi*r; s=R*delta;
for i=1:1000                          %迭代 1000 次
    x_center=(R+r)*cos(t0+delta);     %小圆圆心的 x 坐标
    y_center=(R+r)*sin(t0+delta);     %小圆圆心的 y 坐标
```

```
        alpha=(L-s)/r;
        x_new=x_center+r*cos(t0+delta+pi-alpha);  %更新小圆上 P 点的 x 坐标
        y_new=y_center+r*sin(t0+delta+pi-alpha);  %更新小圆上 P 点的 y 坐标
        plot(x_new,y_new,'.')                    %绘制 P 点
        t0=t0+delta;                             %更新 t
        if L-s>=0
            L=L-s;                               %更新 L
        else
            L=2*pi*r+(L-s);
        end
    end
```

图 3.19 给出了 $r=0.15,0.8$ 时的曲线图。



（A）$r=0.15$（迭代 1000 次）　（B）$r=0.8$（迭代 2000 次）

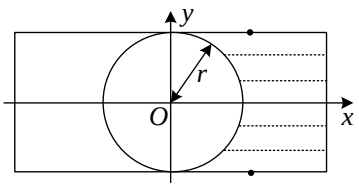
图 3.19　不同 r 值下的万花筒曲线图（外切圆）

3.5　折叠桌的设计

例 3.16　本题取自 2014 年全国大学生数学建模竞赛 B 题：创意平板折叠桌。由尺寸为 120cm×50cm×3cm 的长方形平板加工成可折叠的桌子，桌面呈圆形。桌腿由两组木条组成，每组各用一根钢筋将木条连接，钢筋两端分别固定在桌腿各组最外侧的两根木条的中心位置，并且沿木条有空槽以保证滑动的自由度，如图 3.20 所示。桌腿随着铰链的活动可以平摊成一张平板，每根木条宽 2.5cm，折叠后桌子的高度为 53cm。试建立模型描述此折叠桌的动态变化过程和桌脚边缘线。

图 3.20　折叠桌图片

解　虽然折叠桌设计是离散型的，但仍可以用连续型的方法进行建模。显然，圆形桌面在长方形木板的中心位置。基于圆形桌面下侧所在的平面建立直角坐标系 xoy，其中桌面圆心为原点 O，与长方形平板的长、宽平行的方向分别为 x 轴、y 轴，如图 3.21 所示，其中 y 轴右侧的两个黑点表示最外侧桌腿上钢筋所在的位置。桌面对应的圆的方程为 $x^2+y^2=r^2$，其中 r 为桌面的半径。记长方形的长为 $2l$，宽为 $2r$。过原点 O 且垂直于 xoy 平面、方向朝上的方向为 z 轴正向，建立空间坐标系 $Oxyz$。设桌面底侧距离地面的高度为 h，最外侧桌腿与桌面的夹角为 α，图 3.22 给出了二者之间的关系，即有 $\sin\alpha=h/l$。为简便起见，以下仅考虑桌面右侧的桌腿。



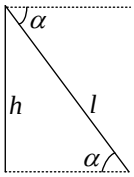


图 3.21　桌面所在平面的直角坐标系　　图 3.22　桌面高度与桌腿示意图

最外侧两个桌腿关于 xOz 平面对称，y 轴正向一侧的最外侧桌腿的线段方程为

$$\begin{cases} z=-x\tan\alpha, \\ y=r, \\ 0\le x\le l\cos\alpha. \end{cases}$$

此桌腿的中点 P_1 的坐标为 $\frac{1}{2}(l\cos\alpha, 2r, -l\sin\alpha)$，对应的另一条最外侧桌腿的中点 P_2 坐标为 $\frac{1}{2}(l\cos\alpha, 2r, -l\sin\alpha)$，因此钢筋所在的线段 P_1P_2 方程为

$$\begin{cases} x=\frac{1}{2}l\cos\alpha, \\ -r\le y\le r, \\ z=-\frac{1}{2}l\sin\alpha. \end{cases}$$

在桌面右侧边缘（半圆）上任选一点 Q_1，其坐标可表示为

$$\begin{cases} x=r\cos\theta, \\ y=r\sin\theta, \\ z=0, \end{cases}$$

其中，$-\pi/2\le\theta\le\pi/2$。

Q_1 点到 y 轴的距离为 $r\cos\theta$，过 Q_1 的桌腿与钢筋的交点 Q_2 的坐标为

$$\begin{cases} x=\frac{1}{2}l\cos\alpha, \\ y=r\sin\theta, \\ z=-\frac{1}{2}l\sin\alpha. \end{cases}$$

向量 $\overrightarrow{Q_1Q_2}=\left(\frac{1}{2}l\cos\alpha-r\cos\theta, 0, -\frac{1}{2}l\sin\alpha\right)$，其长度为

$$\left|\overrightarrow{Q_1Q_2}\right|=\sqrt{\left(\frac{1}{2}l\cos\alpha-r\cos\theta\right)^2+\left(-\frac{1}{2}l\sin\alpha\right)^2}.$$

Q_2 对应的桌腿长度为 $l-r\cos\theta$，它所在的线段方程为

$$\begin{cases} x=r\cos\theta+\frac{l-r\cos\theta}{\left|\overrightarrow{Q_1Q_2}\right|}\left(\frac{1}{2}l\cos\alpha-r\cos\theta\right)t, \\ y=r\sin\theta, \\ z=\frac{l-r\cos\theta}{\left|\overrightarrow{Q_1Q_2}\right|}\left(-\frac{1}{2}l\sin\alpha\right)t, \end{cases}$$

其中，$t\in[0,1]$，$t=0$ 对应 Q_1 点，$t=1$ 对应桌腿的底端。

由已知数据知 $l=0.6$，$r=0.25$，$h=0.5$。假设折叠桌的每侧均有 N 条桌腿，在仿真中需要计算每条桌腿两端的空间坐标。计算与绘图程序如下：

```
clc, clear, close all
L=0.6; r=0.25; h=0.5;
a=asin(h/L); N=21;                    %N 条桌腿
t=linspace(-pi/2,pi/2,N)';            %右侧边缘对应角度的离散化
Q1=[r*cos(t),r*sin(t),zeros(N,1)];                          %Q1 点坐标
D=[L*cos(a)/2-r*cos(t),zeros(N,1),-L*sin(a)/2*ones(N,1)]; %向量 Q1Q2
d=vecnorm(D,2,2);                                    %求 D 的逐行 2 范数
%下面计算桌腿底端坐标
Qe=[r*cos(t)+(L-r*cos(t))./d.*D(:,1),r*sin(t),(L-r*cos(t))./d.*D(:,3)];
Q2=(Q1+Qe)/2;                                        %钢筋端点坐标

hold on, plot3(Q1(:,1),Q1(:,2),Q1(:,3),'.-b')  %绘制桌腿上端 N 个点的连线
for i=1:N
    plot3([Q1(i,1),Qe(i,1)],[Q1(i,2),Qe(i,2)],[Q1(i,3),Qe(i,3)],...
        'r-','LineWidth',2)                  %绘制第 i 条桌腿
end
plot3(Qe(:,1),Qe(:,2),Qe(:,3),'.-b');    %绘制桌腿底端 N 个点的连线
plot3(Q2(:,1),Q2(:,2),Q2(:,3),'-k','LineWidth',2)  %绘制钢筋对应的连线
view(35,15);                                        %设置视角
xlabel('$x$','Interpreter','Latex'), ylabel('$y$','Interpreter','Latex')
zlabel('$z$','Interpreter','Latex','Rotation',0)
```

输出图形见图 3.23。在桌子折叠过程中，桌面高度 h 的变化范围为 0cm~50cm。对于不同的 h，可以绘制不同的桌腿，从而动态地演示桌子的折叠过程。

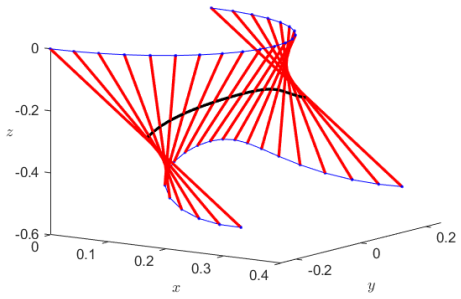


图 3.23　桌腿示意图

3.6　视频读写

视频是由若干帧图像组成的。对于视频文件 filename，Matlab 可构建与之相关的多媒体读取格式：

```
Obj=VideoReader(filename);
```

其中，变量 Obj 的类型为 VideoReader。文件 filename 的一些属性保存在变量 Obj 中，例如，Obj.Height 和 Obj.Width 分别为每帧图像的高和宽，Obj.FrameRate 为视频每秒播放的帧数，Obj.NumFrames 为视频包含图像的帧数。

命令 read 用于读取视频文件，使用格式为

```
video=read(Obj);
```

对于 RGB 彩色视频，得到的 video 是 4D 数组，大小为 Obj.Height×Obj.Width×3×Obj.NumFrames，数据类型一般为 uint8 性。

在视频写入时，先使用 VideoWriter 命令创建视频写入目标：

Obj=VideoWriter(filename)

再用 open 命令打开所创建的目标 open(Obj)，接着使用 writeVideo 命令将当前图像（frame）写入视频：

writeVideo(Obj,Frame);

最后关闭视频 close(Obj)。

例 3.17　Matlab 图像处理工具箱中包含视频文件 rhinos.avi，其中 avi 是视频的类型。完成下列实验：

（1）读取该视频，得到 4D 数组；

（2）由读取的每帧图像来演示视频；

（3）将演示的视频保存，名称为 rhinos2.avi。

解　程序如下：

```
clc, clear, close all
A=VideoReader('rhinos.avi');  %读取视频文件
B=read(A);                    %从 A 中读取数据，B 为 4D 数组
disp(size(B))  %输出 B 的维数为 240×320×3×114，即视频由 114 帧图像组成

n = A.NumFrames               %视频帧数：114
H = A.Height;                 %视频高度：240
W = A.Width;                  %视频宽度：320
%初始化结构型变量 mov,用于存储视频
mov(1:n)=struct('cdata',zeros(H,W,3,'uint8'),'colormap',[]);
for i=1:n
    mov(i).cdata=B(:,:,:,i);    %对 mov 的数据域 cdata 进行赋值
end
h=figure;
%左下角为[0,0],[150,150]表示显示器的起始像素坐标
set(h,'position',[150,150,W,H])
movie(h,mov)

writeObj=VideoWriter('rhinos2.avi');  %创建目标，用于写入视频
writeObj.FrameRate=A.FrameRate;       %对视频帧数进行赋值
open(writeObj);                       %打开文件
for i=1:n
    imshow(B(:,:,:,i));                %显示图像
    frame=getframe;                   %获取当前图像
    writeVideo(writeObj,frame);       %将获取的图像写入到 writeObj 中
end
close(writeObj);                      %关闭文件
```

例 3.18　对单位圆 $x^2+y^2=1$ 作 n 条直径，其中第 i 条直径与 x 轴正向夹角为 $\alpha_i=(i-1)\pi/n$，

$i=1,2,\cdots,n$。对于第 i 条直径，选某点在该直径上做简谐振动，其运动方程为

$$\begin{cases} x_i(t)=\cos\dfrac{(i-1)\pi}{n}\cdot\sin\left[t+\dfrac{(i-1)\pi}{n}\right], \\ y_i(t)=\sin\dfrac{(i-1)\pi}{n}\cdot\sin\left[t+\dfrac{(i-1)\pi}{n}\right], \end{cases}$$

其中 $t\in[0,2\pi]$ 为时间变量。将 n 个点的运动轨迹制作成视频，名称为 ex2.avi。

解　在绘图时，需要先绘制圆域和 n 条直线。圆的参数方程为

$$\begin{cases} x=\cos\theta, \\ y=\sin\theta, \end{cases} \quad \theta\in[0,2\pi].$$

将参数 θ 等间隔离散化，得到 k_1 个分点。使用 fill 命令来填充圆域对应的多边形（k_1 条边，即单位圆的近似）。并用 plot 命令绘制直径。将时间 $t\in[0,2\pi]$ 进行等分，得到 k_2 个分点，即视频帧数为 k_2。

在实验中取 $n=8$，$k_1=200$，$k_2=500$，每秒播放的帧数为 15。制作视频的程序如下：

```
clc, clear, close all
n=8; k1=200; k2=500;
theta=linspace(0,2*pi,k1);
xt=cos(theta); yt=sin(theta);    %单位圆上的离散点坐标
Obj=VideoWriter('ex2.avi');      %创建空的待写入的视频文件
Obj.FrameRate=15;                %设置视频每秒的帧数
open(Obj);                       %打开视频文件
for t=linspace(0,2*pi,k2)        %将时间离散化
    fill(xt,yt,'r')
    axis equal off, hold on
    for i=1:n
        X=cos((i-1)*pi/n); Y=sin((i-1)*pi/n);
        plot([X,-X],[Y,-Y],'k-','LineWidth',2)  %绘制 n 条直线
    end
    x=cos([0:n-1]*pi/n).*sin(t+[0:n-1]*pi/n);
    %给定时间 t，n 条直径上点的 x 坐标分量
    y=sin([0:n-1]*pi/n).*sin(t+[0:n-1]*pi/n);
    %给定时间 t，n 条直线上点的 y 坐标分量
    scatter(x,y,100,[1:n],'filled','MarkerEdgeColor','g')
    %t 时刻，n 个点的散点图
    hold off                     %关闭图形保持功能
    frame=getframe;              %获取当前图像
    writeVideo(Obj,frame);       %将获取图像写入 Obj 中
end
close(Obj);
```

在上述程序中，使用 scatter 命令绘制 n 个点的散点图，并将每次绘制的图形以图像形式保存到结构变量 frame 中，图 3.24 给出了 $t=6.2832$ 时的图形。

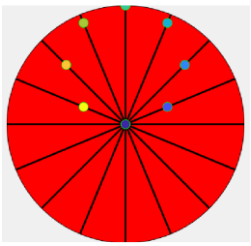


图 3.24　$t=6.2832$ 时的图形

习题 3

3.1　已知 4×5 维矩阵 $\boldsymbol{B}$ 的数据如表 3.4 所列，其第一行表示 x 坐标，第二行表示 y 坐标，第三行表示 z 坐标，第四行表示类别。

表 3.4　矩阵 $\boldsymbol{B}$ 的数据

7.7	5.1	5.4	5.1	5.1	5.5	6.1	5.5	6.7	7.7	6.4	6.2	4.9	5.4	6.9
2.8	2.5	3.4	3.4	3.7	4.2	3	2.6	3	2.6	2.7	2.8	3.1	3.9	3.2
6.7	3	1.5	1.5	1.5	1.4	4.6	4.4	5.2	6.9	5.3	4.8	1.5	1.7	5.7
3	2	1	1	1	1	2	2	3	3	3	3	1	1	3

（1）使用 scatter3 绘制散点图。对于类别为 1、2、3 的点，圆圈大小分别为 40、30、20；不同类别的点，其颜色不同。

（2）使用 x,y 坐标利用 gscatter 绘制散点图，对于类别为 1、2、3 的点，对应点分别用圆圈、正方形、三角形表示，颜色分别为红色、绿色和蓝色。

3.2　绘制平面 $3x-4y+z-10=0$，$x\in[-5,5]$，$y\in[-5,5]$。

3.3　绘制瑞士卷曲面，要求相同的 t 值，颜色相同。

$$\begin{cases} x=t\cos t, \\ 0\le y\le 3, \quad t\in[\pi,9\pi/2]. \\ z=t\sin t, \end{cases}$$

3.4　附件 1：区域高程数据.xlsx 给出了某区域 43.65×58.2（km）的高程数据，画出该区域的三维网格图和等高线图，在 A（30，0）和 B（43，30）（单位：km）点处建立了两个基地，在等高线图上标注出这两个点。并求该区域地表面积的近似值。