

# 数学实验与数学软件

## 第十五课：MATLAB数据可视化 (二)

李嘉

中山大学数学学院

教师微信号：Recksic

课程邮箱：[sysumatlab@163.com](mailto:sysumatlab@163.com)



## 多子图绘图-例5.2-8

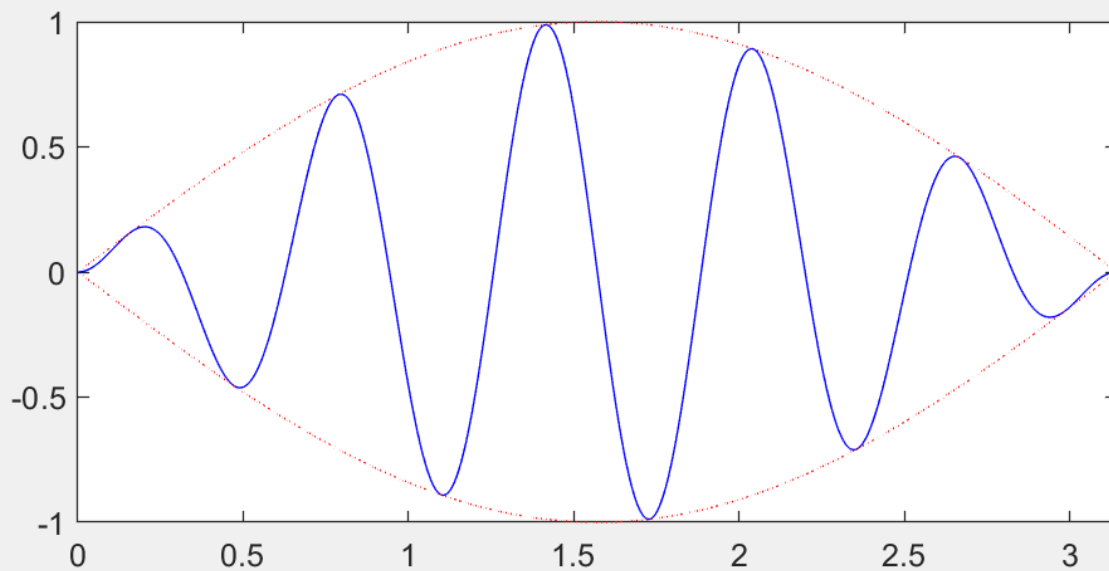
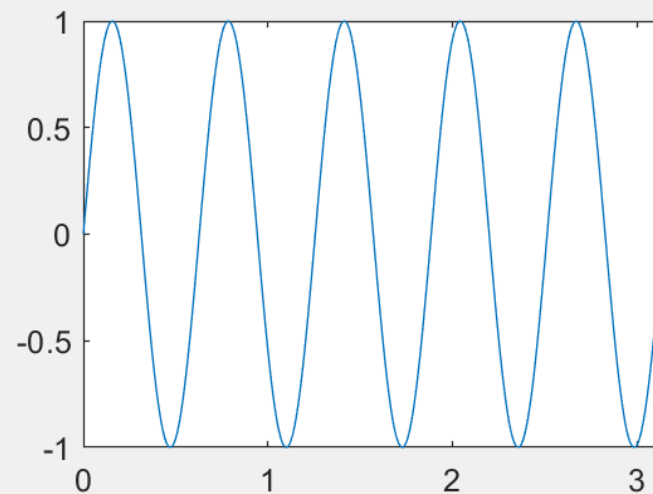
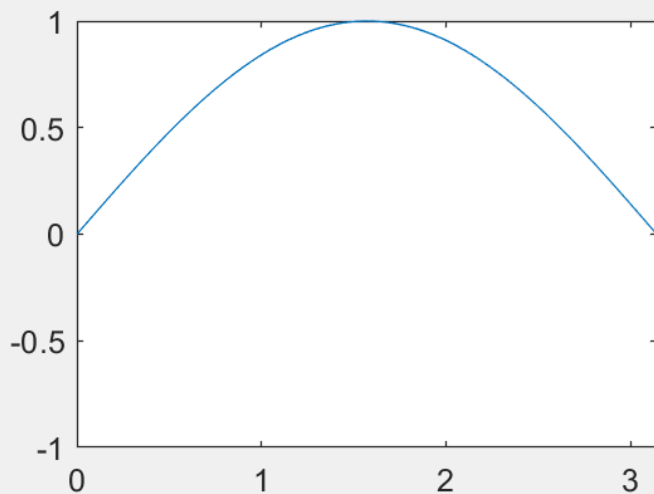
- ❑ `subplot(m,n,k)` 为MATLAB设定mXn的子图中第k个位置画图
- ❑ `subplot('position',[left bottom width height])`  
为MATLAB按指定位置画图，其中，四个未知参数范围均在0~1之间，左下角为0位，右上角为1位

### 【例5.2-8】

```
clf;t=(pi*(0:1000)/1000)';  
y1=sin(t);y2=sin(10*t);y12=sin(t).*sin(10*t);  
subplot(2,2,1),plot(t,y1);axis([0,pi,-1,1])  
subplot(2,2,2),plot(t,y2);axis([0,pi,-1,1])  
%以上命令为左上角与右上角位置分别绘制一幅函数曲线图
```

```
subplot('position',[0.2,0.1,0.6,0.40])  
%上端不超过0.5就不会覆盖之前绘制的两幅图  
plot(t,y12,'b-',t,[y1,-y1],'r:')  
axis([0,pi,-1,1])
```

# 多子图绘图-例5.2-8

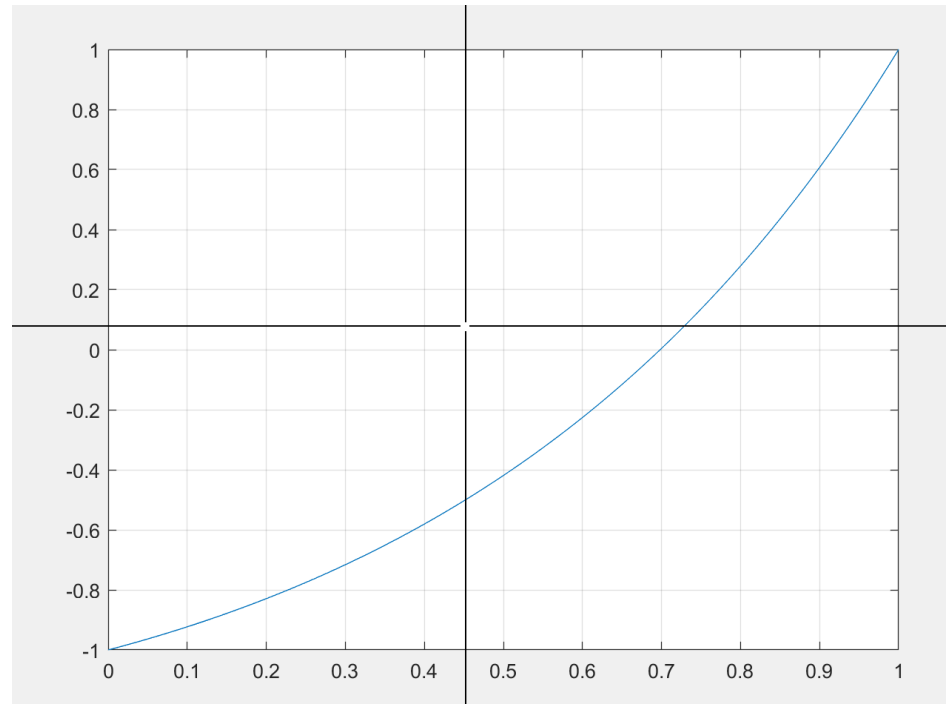


# 获取屏幕坐标-例5.2-9

□ `[x,y]=ginput(n)` 采用鼠标点击的方法获取二维图形坐标，  
n为坐标点的个数，多个点的x和y以列向量的形式存储

【例5.2-9】

```
clf,x=0:0.01:1;y=(x+2).^x-2;  
plot(x,y);grid on
```



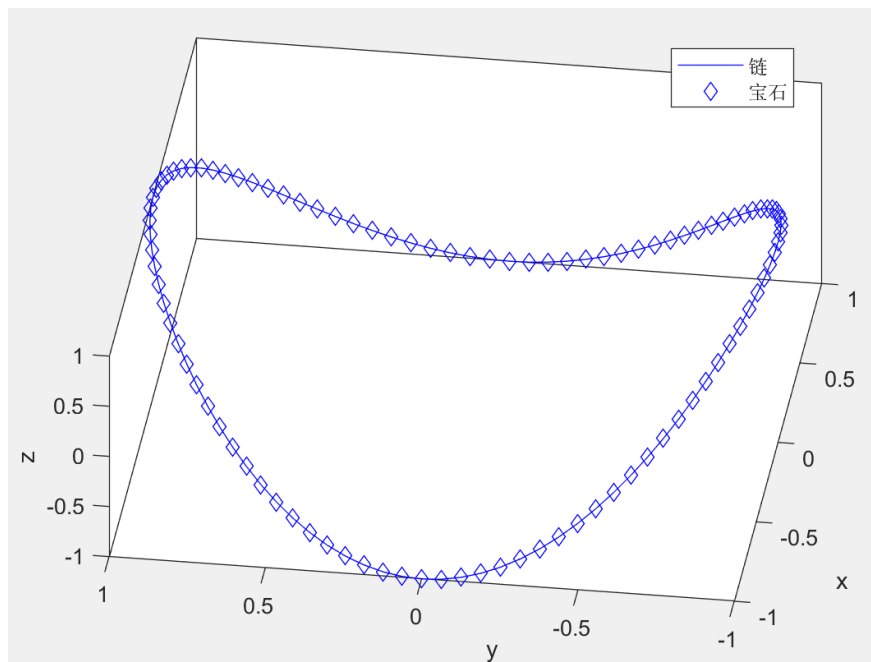
```
[x,y]=ginput(1);  
format long,x,y  
x = 0.69828692903537  
y = -5.884401711382421e-010
```

# 三维曲线的绘制-例5.3-1

□ `plot3(X,Y,Z,'s')` 分别代表三维坐标序列和线型等因素，使用方法以及多条曲线绘制的扩展与 `plot` 函数完全类似。

【例5.3-1】

```
t=(0:0.02:2)*pi; x=sin(t);y=cos(t);z=cos(2*t);  
plot3(x,y,z,'b-','x,y,z','bd')  
view([-82,58]),box on %观察角度(经纬度)，并加边框  
xlabel('x'),ylabel('y'),zlabel('z')  
legend('链','宝石','Location','best')
```



# 三维曲面或网线

- ❑ 与`ezsurf`和`ezmesh`不同，基于数值数组的曲面或网格线在绘制前需要指定好采样的函数值。通常来讲，需要准备一个方格型的取值区域并将所有点的横纵坐标命名为`x`与`y`（用`meshgrid`函数与横纵坐标向量生成，`x`与`y`均为矩阵），再定义网格区域内每个点的函数值 $z=f(x,y)$
- ❑ `surf(x,y,z)`为三维曲面的绘图方式，若`x`与`y`缺省则会以矩阵`z`的两个维度下标来代替。
- ❑ `mesh(x,y,z)`为三维网格线的绘图方式，若`x`与`y`缺省则会以矩阵`z`的两个维度下标来代替。
- ❑ `surf(x,y,z,c)`在曲面绘制的基础上增加了颜色函数 $C=c(x,y)$ ，以决定各个点的颜色变化。默认的色彩设置为 $C=z$ ，即函数按照曲面的高度来决定每个点的颜色。
- ❑ `mesh(x,y,z,c)`的原理与`surf`函数完全类似。

## 例5.3-2

【例5.3-2】。

```
clf
```

```
x=-4:4;y=x;
```

```
[X,Y]=meshgrid(x,y);
```

```
Z=X.^2+Y.^2;
```

%以1为步长设立2维网格

%此时x与y均为9x9矩阵，储存所有点横纵坐标

%z同样为9x9的矩阵

```
surf(X,Y,Z);
```

```
colormap(hot)
```

%设置绘图的颜色分布

```
hold on
```

```
stem3(X,Y,Z,'bo')
```

%再曲面下方再绘制三维杆图，每个点都连接到xoy坐标面

```
hold off
```

```
xlabel('x'),ylabel('y'),zlabel('z')
```

```
axis([-5,5,-5,5,0,inf])
```

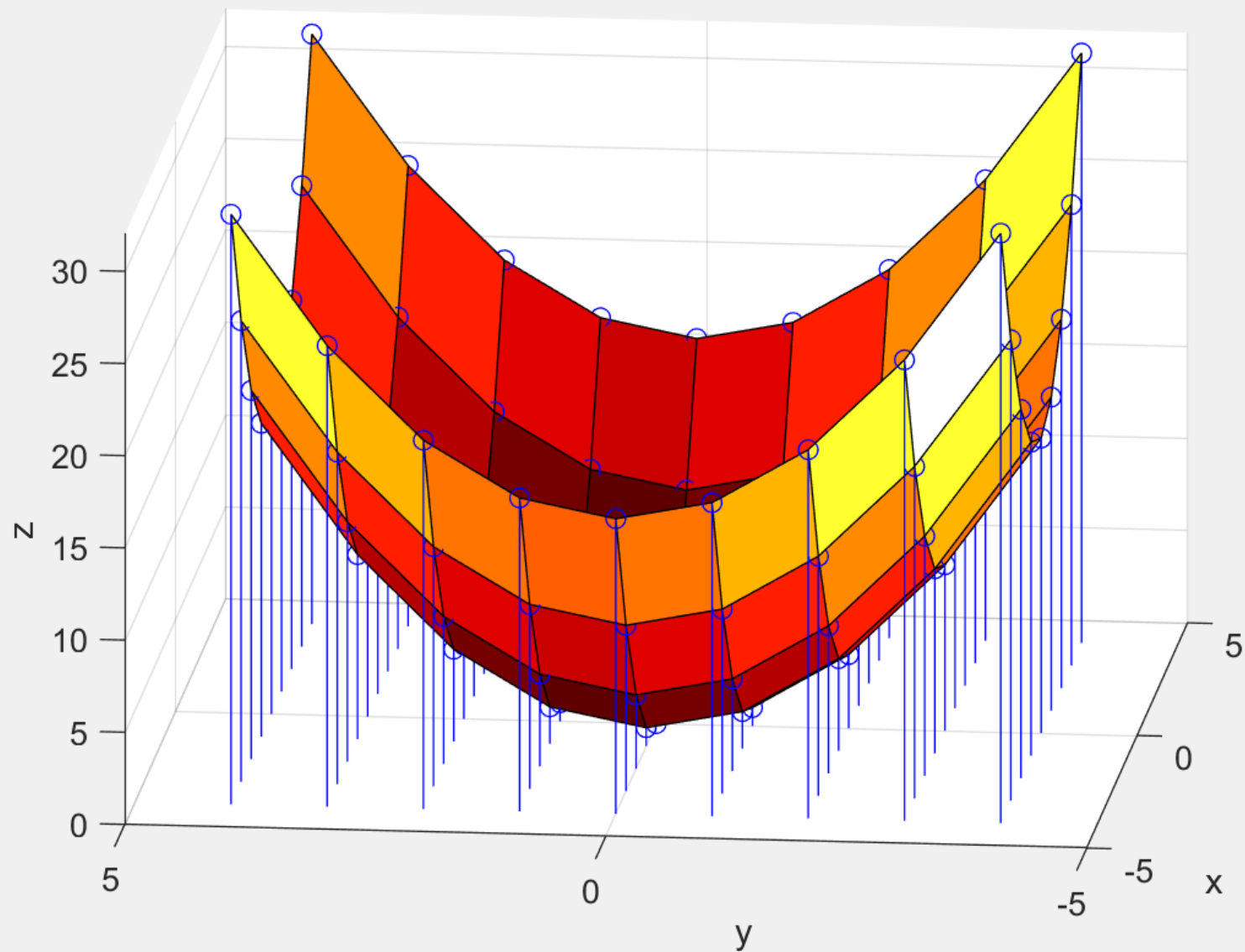
%上侧边界由MATLAB来默认指定，并不是真的到inf

```
view([-84,21])
```

%经度-84度，纬度21度

%容易发现，绘图的颜色是分块的，并没有ezsurf那样的渐变色

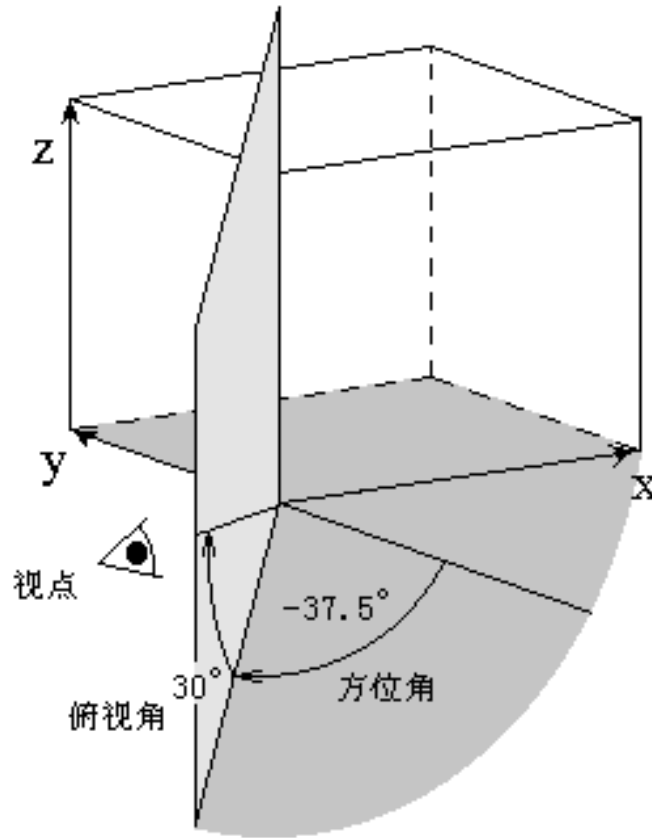
## 例5.3-2





# 视角与颜色控制
















□ `view([longi, lati])` 是一种视角设置方法，同时还可以通过 `view([x, y, z])` 来控制视角， $[x, y, z]$  为观察点位置



□ `colormap(CM)` 或 `colormap CM` 可以改变全图或子图的颜色

# 颜色控制

❑ 旧版本的默认`colormap`为`jet`，新版本默认为`parula` (森莺)

parula	
jet	
hsv	
hot	
cool	
spring	
summer	
autumn	
winter	
gray	
bone	
copper	
pink	
lines	
colorcube	

# 浓淡控制

□ **shading options**, 其中option是包含选项如下

指令	功能
<b>shading faceted</b>	“瓷砖”的颜色是均匀一致的，而且同时显色“瓷砖”交接的“边”。(此为缺省值),ezplot会含黑色分隔框
<b>shading flat</b>	“瓷砖”的颜色是均匀一致的。
<b>shading interp</b>	使用双线性插值来使“瓷砖”根据四顶点的颜色产生连续的变化。 ezplot函数因采用足够精细的采样，这句更多的还是去分隔框的效果。

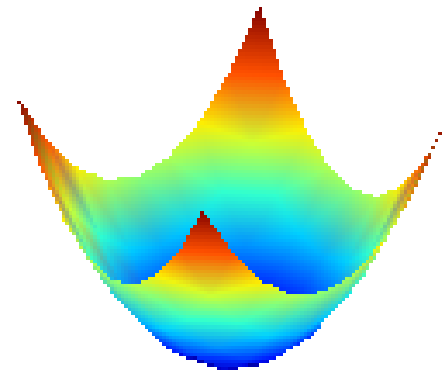
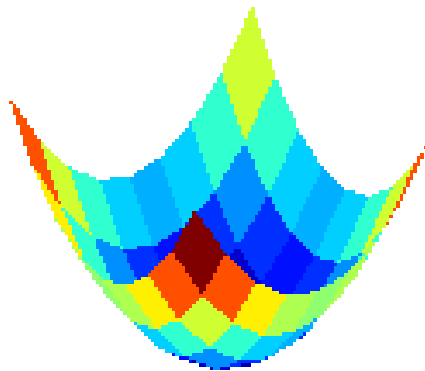
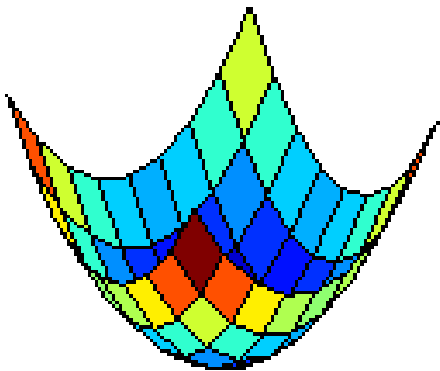
## 例5.3-3

### 【例5.3-3】

```
clf,x=-4:4;y=x;;[X,Y]=meshgrid(x,y);Z=X.^2+Y.^2;  
surf(X,Y,Z) %本句绘制的图像会被后面的绘图覆盖掉  
colormap(jet) %此后绘图都使用这一色图
```

```
subplot(1,3,1),surf(Z),axis off  
subplot(1,3,2),surf(Z),axis off,shading flat  
subplot(1,3,3),surf(Z),axis off,shading interp
```

```
set(gcf,'Color','w') %设置图形窗底色为白色,默认为灰色  
%MATLAB语言中,gcf为当前Figure句柄值,gca为当前坐标轴句柄值
```



# 透明度控制-例5.3-4

□ **alpha(v)** 可以设置目标的透明度,  $0 \leq v \leq 1$ ,  $v=0$ 代表完全透明,  $v=1$ 代表完全不透明

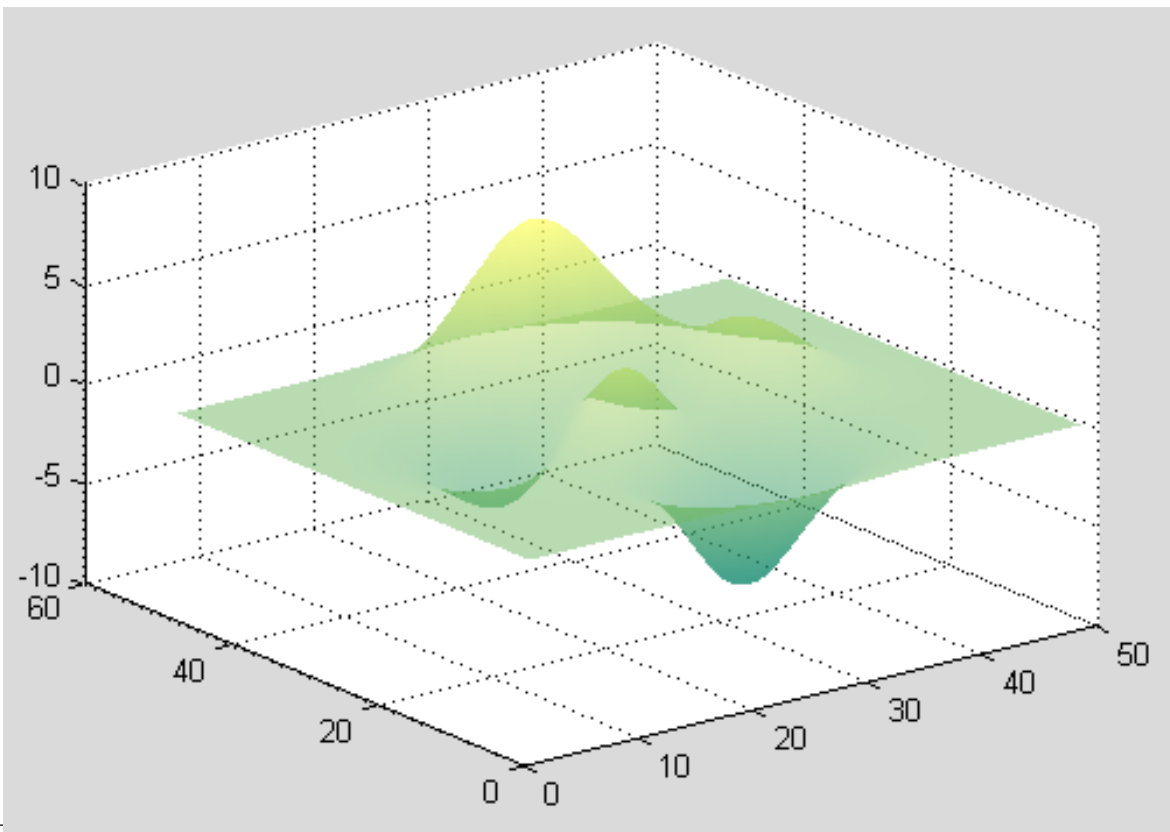
【例5.3-4】

**clf, surf(peaks)**

**%peaks**为MATLAB自带的函数

$$\%f(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{1}{5}x - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}$$

**shading interp, alpha(0.5), colormap(summer)**    **%半透明**



# 光照与照明模式控制

- ❑ `light('Name', value, ...)` 为灯光设置函数
- ❑ `'color', [1, 0, 0]`, 含义为红色光, 默认为白光
- ❑ `'style'` 的两种 `value` 包括 `'infinite'` (平行光), 或 `'local'` (点光源)
- ❑ `'position'` 代表光源位置, 平行光为穿过该点到原点的方向
- ❑ `lighting options` 设置照明模式, `options` 有以下选项

指令	功能
flat	入射光均匀的洒在图形对象的每个面上 (缺省模式)
gouraud	先对顶点颜色进行插补, 再对顶点勾画的面色进行插补, 用于曲面表现。
phong	对顶点处法线进行插值, 再计算各像素的反光, 表现效果最好, 但费时。
none	关闭所有的光源。

# 光反射材质设定

❑ `material[ka kd ks n ns]` 的五个参数可以分别设置光反射材质的环境/漫反射/镜面反射强度、镜面反射指数和镜面反射颜色反射率

❑ `material options` 也可以利用MATLAB自带的参数设置

指令	功能
shiny	使对象比较明亮，镜面反射额较大，反射光的颜色取决于光源的颜色。
dull	使对象比较暗淡，漫反射额较大，反射光的颜色取决于光源的颜色。
metal	使对象有金属光泽，反射光的颜色取决于光源的颜色和图形表面的颜色（此为缺省值）
default	返回到Matlab缺省值

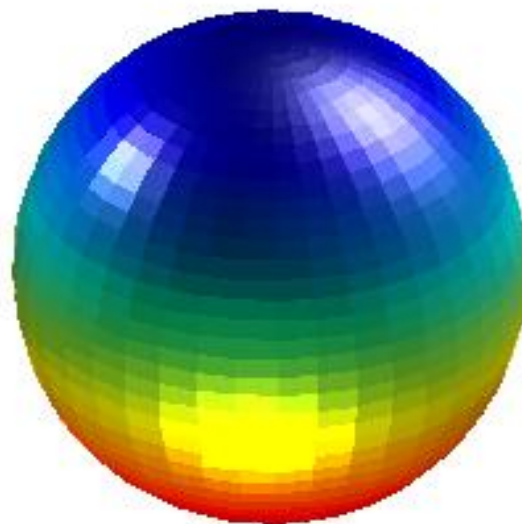
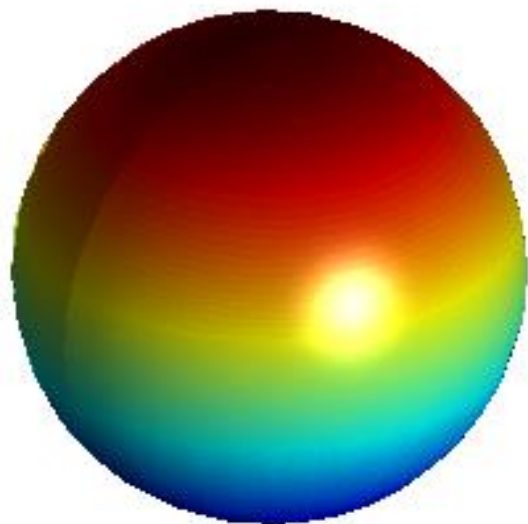
## 例5.3-5

【例5.3-5】（图5.3-6）

```
clf;  
[X,Y,Z]=sphere(40); %生成每一维度40等分的三维球面，x,y,z均为41x41矩阵  
  
colormap(jet),subplot(1,2,1),surf(X,Y,Z)  
axis equal off,shading interp %实现一种渐变效果  
  
light('position',[0 -10 1.5],'style','infinite')%后偏上方平行光  
lighting phong %法线插值后的反射计算，慢，效果好  
material shiny %明亮的镜面反射模式  
  
subplot(1,2,2),surf(X,Y,Z,-Z)  
axis equal off,shading flat %每片各自有不同颜色  
  
light;lighting flat  
%默认的光源([1,0,1]到原点方向平行白光)，默认的均匀照明模式  
light('position',[-1,-1,-2],'color','y')%左后下方的黄光  
light('position',[-1,0.5,1],'style','local','color','w')  
%左前上方补充白色点光源  
set(gcf,'Color','w') %背景底色为白色
```



## 例5.3-5



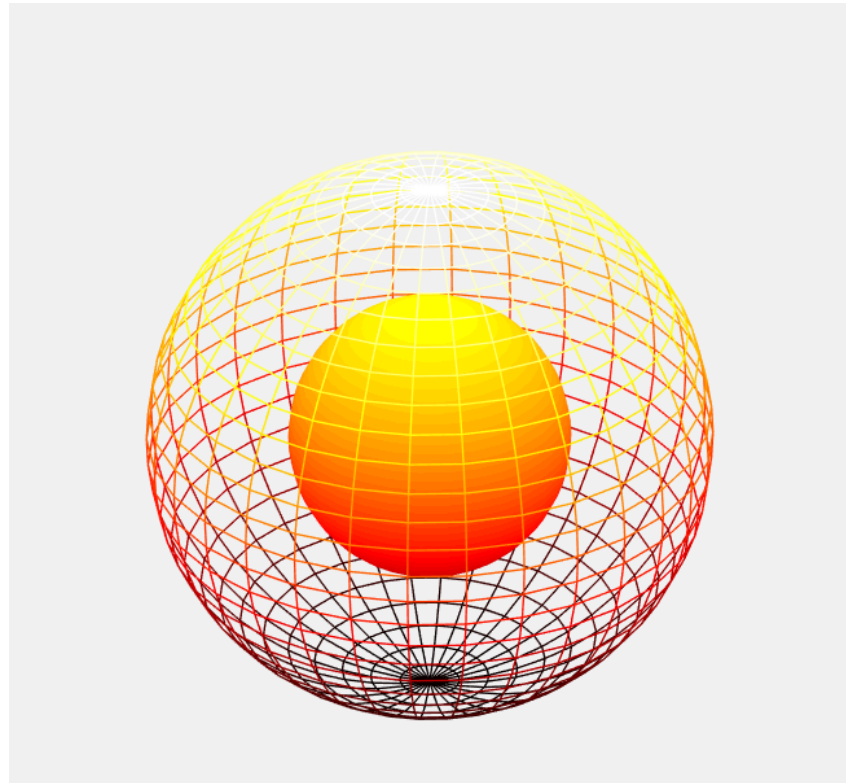
- 注意到两幅绘图的`colormap`都被设置为了`jet`，如果需要分别设置不同的色图，需要使用`h1=subplot(1,2,1)`，`colormap(h1,jet)`，`h2=subplot(1,2,2)`，`colormap(h2,...)`以实现不同色图的效果。

# 图形的透视-例5.3-6

□ **hidden on/off** 可以控制 (如网格) 叠压图形的消隐和透视

【例5.3-6】 (图5.3-7)。

```
[X0,Y0,Z0]=sphere(30);           % [x0,y0,z0]代表单位球  
X=2*X0;Y=2*Y0;Z=2*Z0;           % [x,y,z]代表半径为2的球  
surf(X0,Y0,Z0);shading interp  
hold on,mesh(X,Y,Z),colormap(hot),hold off  
hidden off,axis equal,axis off    %确保外部网格可以被透视
```

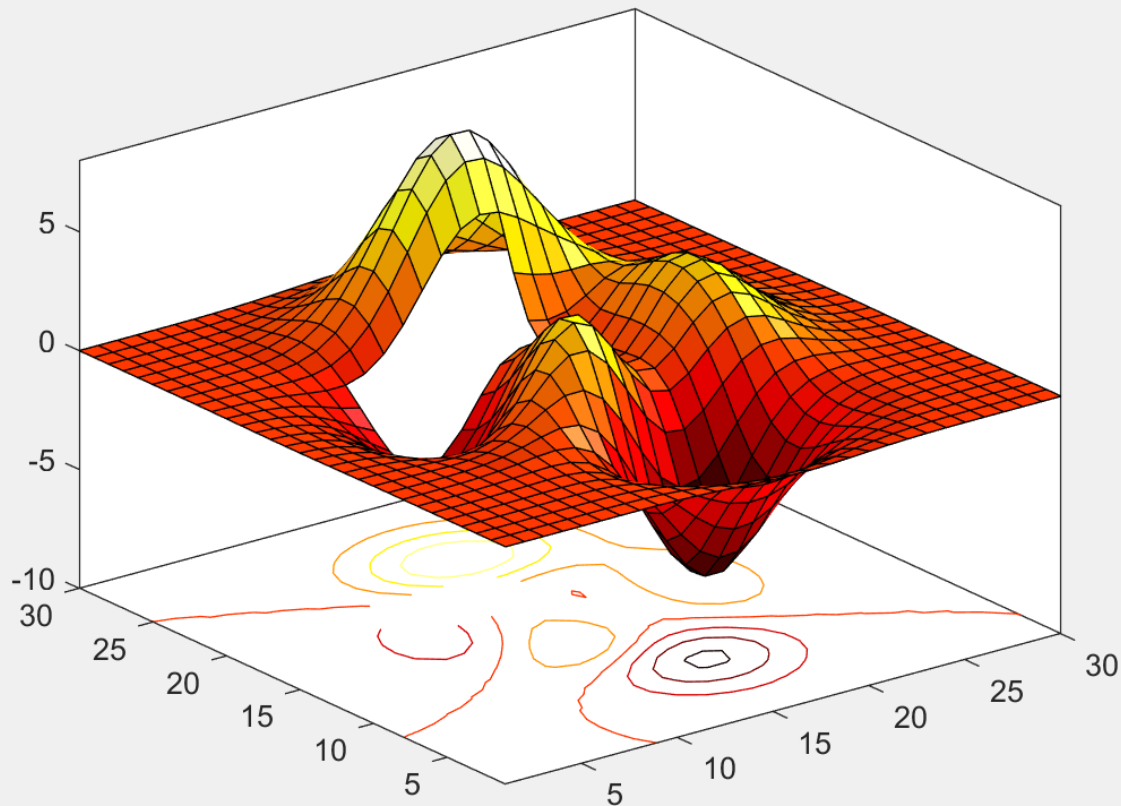


# 图形的镂空-例5.3-7

□ 设置z值为NaN可以实现简单镂空效果

【例5.3-7】（图5.3-8）

```
P=peaks(30);P(18:20,9:15)=NaN;surfc(P); %镂空一个矩形区域,有等位线  
colormap(hot),light('position',[50,-10,5])  
material([0.9,0.9,0.6,15,0.4])  
grid off,box on
```

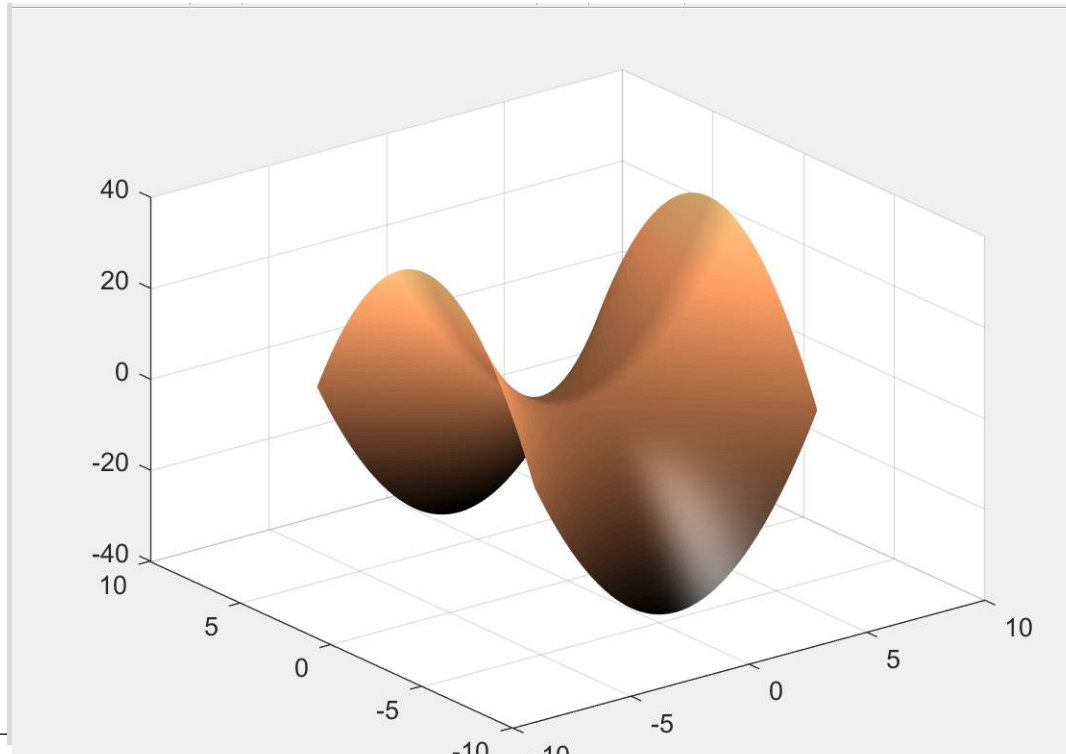


# 图形的裁切-例5.3-8

□ 裁切如果不希望其余部分被“镂空”，可以设置为0

【例5.3-8】（图5.3-9）。

```
clf,x=[-8:0.1:8];y=x;[X,Y]=meshgrid(x,y);ZZ=X.^2-Y.^2;  
ii=find(abs(X)>6|abs(Y)>6);           %对[-6,6]x[-6,6]之外地方进行忽略  
ZZ(ii)=zeros(size(ii));               %设置为0  
surf(X,Y,ZZ),shading interp;colormap(copper)  
light('position',[0,-15,1]);lighting phong  
material([0.8,0.8,0.5,10,0.5])
```



# 伪彩图与等位线-例5.4-1

❑ 函数 `pcolor` 可以完成伪彩图绘制，`contour` 可以绘制等位线，而 `contourf` 可以绘制带有颜色填充的等位线图

## 【例5.4-1】

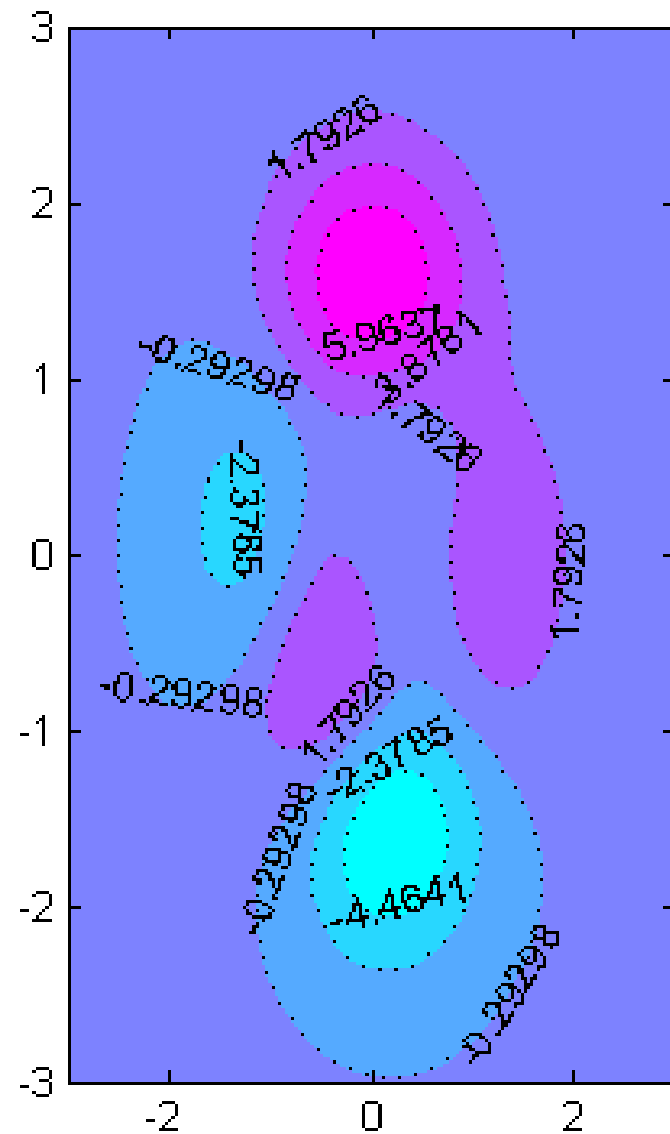
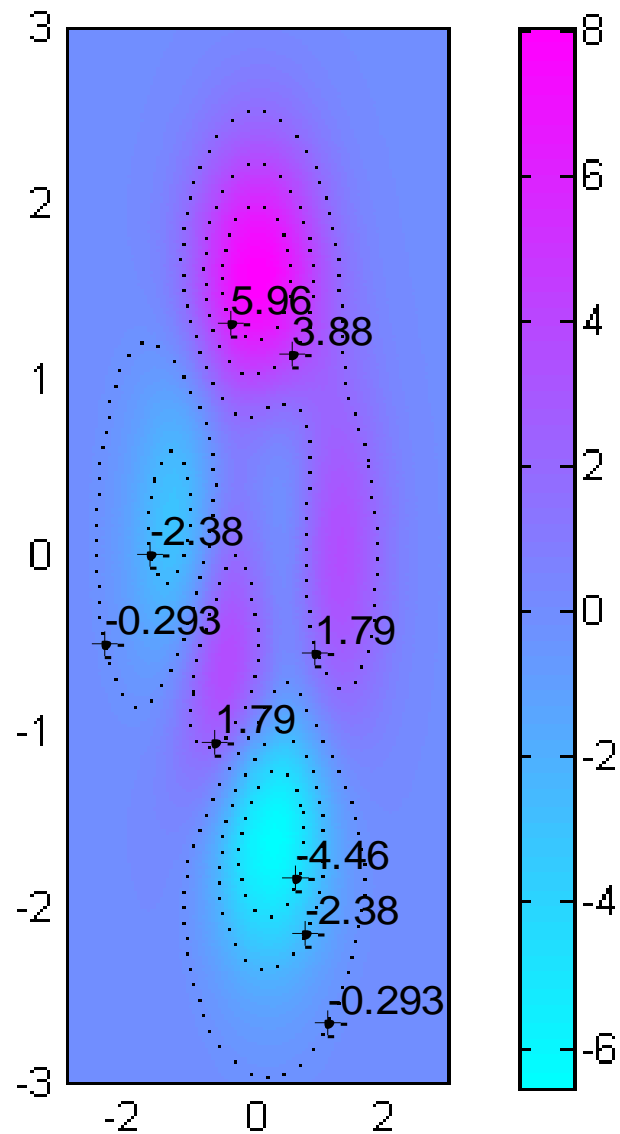
```
clf;clear;[X,Y,Z]=peaks(40);%获取peaks的坐标值，每一维40点等分
n=6;subplot(1,2,1)
pcolor(X,Y,Z),shading interp%伪彩图的绘制，后面一句可以去掉黑色网格

zmax=max(max(Z));zmin=min(min(Z));%查找并记录z的最大值和最小值
caxis([zmin,zmax]), colorbar
%在绘图的旁边添加色标尺，表明颜色与取值之间的一一对应关系

hold on,C=contour(X,Y,Z,n,'k:');%在伪彩图上加黑色等位线，分n=6级
clabel(C), hold off %为等位线C添加随机标识，以指明每条线的对应值

subplot(1,2,2)
[C,h]=contourf(X,Y,Z,n,'k:'); %直接绘制填色等位线，线性黑虚线
clabel(C,h),colormap(cool)
%利用等位线矩阵C与对象h添加标识，并将两幅图的色图都设置为cool
set(gcf,'Color','w')
```

# 伪彩图与等位线-例5.4-1



# 颜色变化表示准四维信息-例5.4-2

□ 双自变量与双函数的构成，称之为准四维表现

【例5.4-2】（图5.4-2）

```
clf,x=3*pi*(-1:1/15:1);y=x;[X,Y]=meshgrid(x,y);
```

```
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R; %防除0操作
```

```
[dzdx,dzdy]=gradient(Z);
```

```
dzdr=sqrt(dzdx.^2+dzdy.^2);
```

```
dz2=del2(Z);
```

%计算各点梯度的模

%拉普拉斯算子计算近似曲率

```
subplot(1,2,1),surf(X,Y,Z,abs(dzdr)),shading faceted;
```

%分片同色+黑色线条分隔的绘制方法

```
colorbar('SouthOutside')
```

%下方色标尺

```
brighten(0.6);colormap hsv
```

%变量后，色图用128色调变化

```
title('No. 1 surf(X,Y,Z,abs(dzdr))')
```

```
subplot(1,2,2);surf(X,Y,Z,abs(dz2))
```

```
shading faceted
```

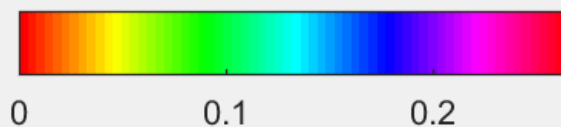
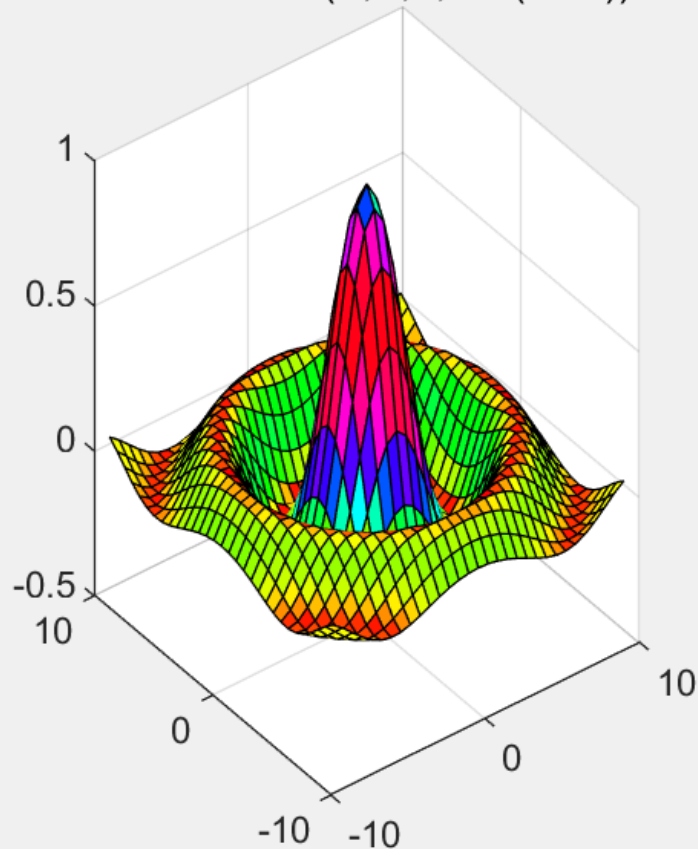
```
colorbar('NorthOutside')
```

%曲率图的色标尺放置于上方

```
title('No. 2 surf(X,Y,Z,abs(dz2))')
```

# 颜色变化表示准四维信息-例5.4-2

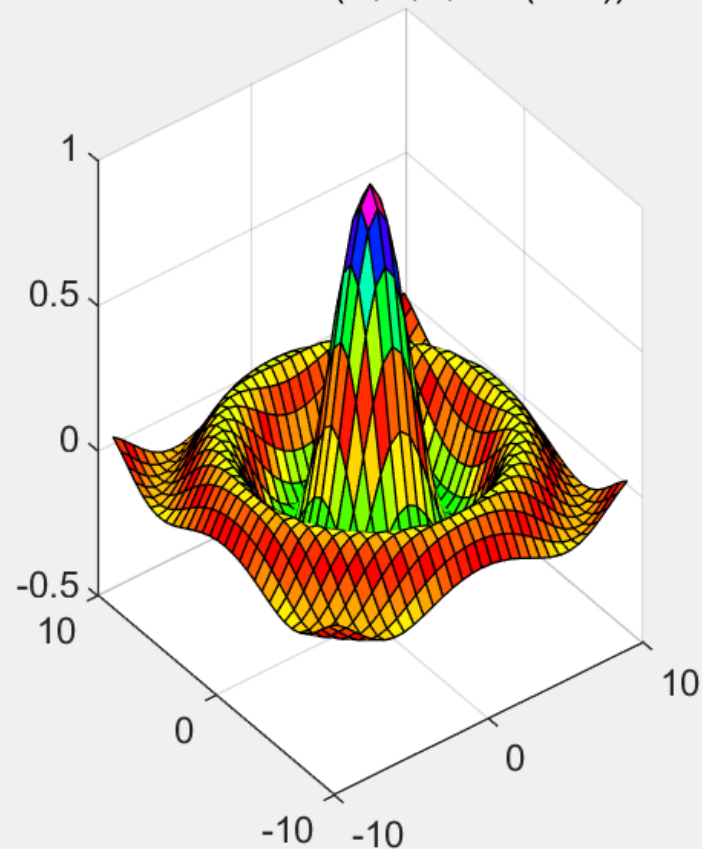
No. 1 `surf(X,Y,Z,abs(dzdr))`



0.02 0.04 0.06



No. 2 `surf(X,Y,Z,abs(dz2))`





# 切片图表示部分四维信息-例5.4-3

❑ `slice(X,Y,Z,V,sx,sy,sz)` 可以绘制切片图，其中，  
`x,y,z,v`为同型三维矩阵，表示一组点的各维度坐标和颜色  
函数值。切片方向与坐标面平行，每一个`sx,sy,sz`的取值都  
会带来一个新的切片绘制。

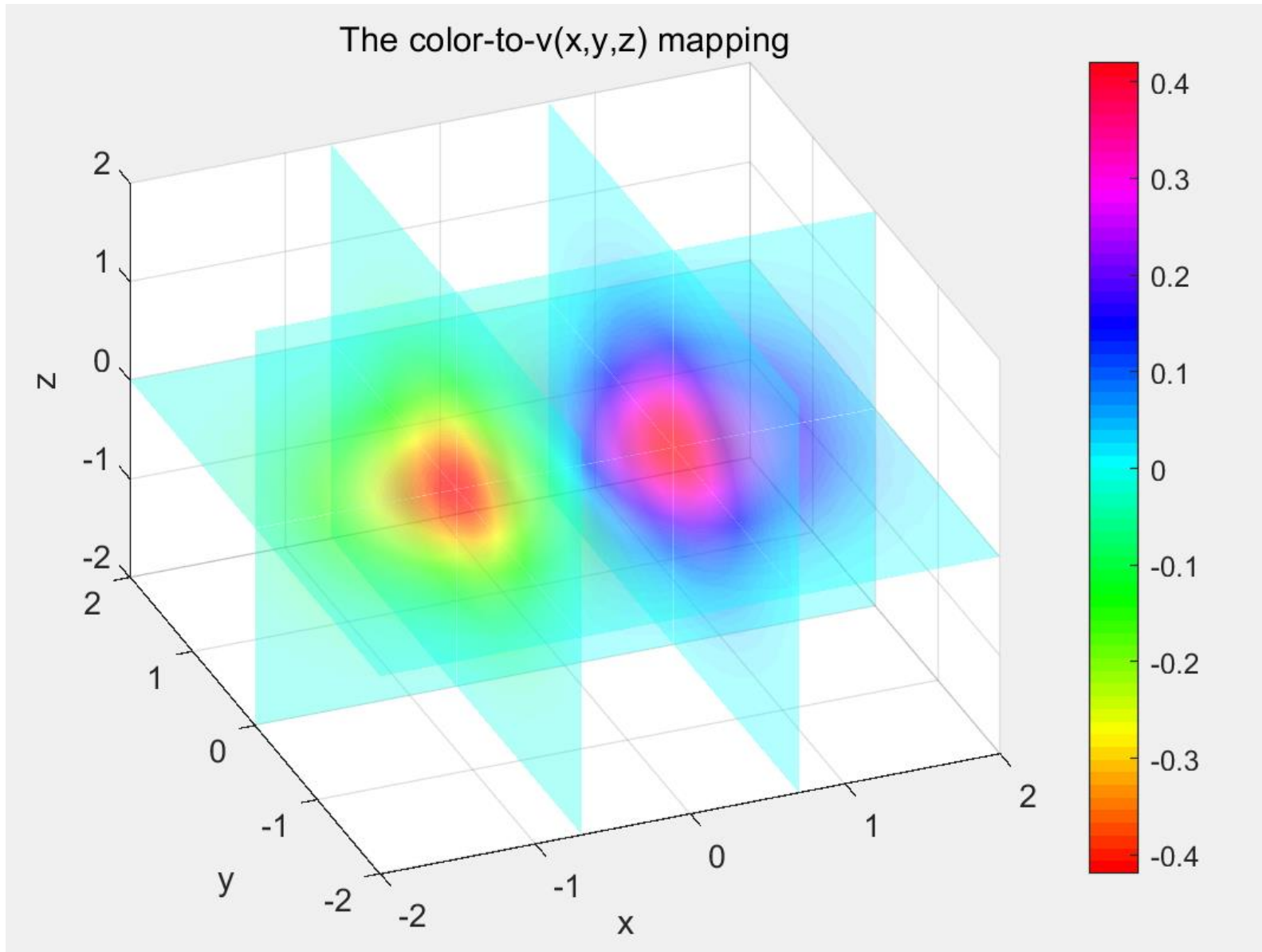
【例5.4-3】

```
clf
[x,y,z] = meshgrid(-2:.2:2,-2:.25:2,-2:.16:2);
%17x21x26的点阵构成了三维矩阵
v = x.*exp(-x.^2-y.^2-z.^2);           %对应的颜色函数值
xs = [-0.7,0.7]; ys = 0; zs = 0;       %共四个切面x = ±0.7,y = 0,z = 0
slice(x,y,z,v,xs,ys,zs)

colorbar, shading interp, colormap hsv
xlabel('x'), ylabel('y'), zlabel('z')
title('The color-to-v(x,y,z) mapping')

view([-22,39])                          %视角与透明度
alpha(0.3)
```

# 切片图表示部分四维信息-例5.4-3



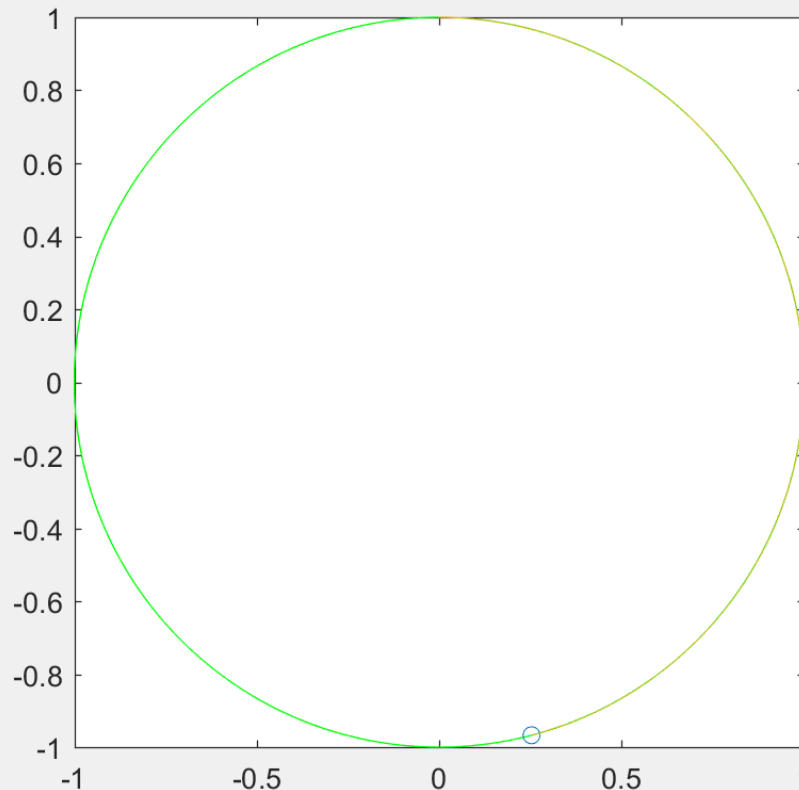
# 彗星线的绘制-例5.4-4

□ `comet(x,y,p)` 或 `comet3(x,y,z,p)` 可绘制二维或三维彗星线, `p*length(y)` 或 `p*length(z)` 为彗星线长度

【例 5.4-4】

```
shg;n=2;t=n*pi*(0:0.000005:1);x=sin(t);y=cos(t);  
plot(x,y,'g');axis square  
hold on,comet(x,y,0.0001),hold off
```

%已有绿色的原型圈  
%从上方开始的长度为20的短彗星

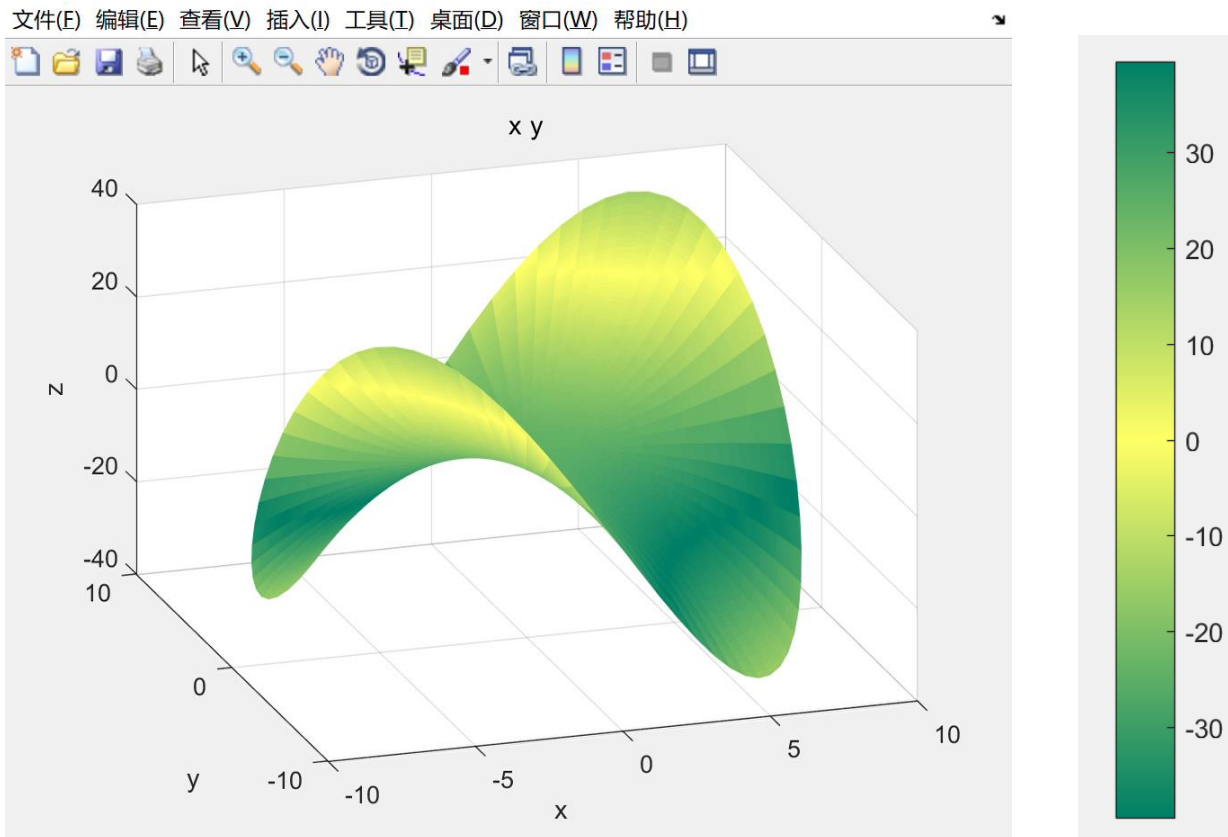


# 色图的变幻-例5.4-5

□ **spinmap(t,inc)** 可以使色图与图像颜色动态变化，**t**为变化时间（默认为3），**inc**代表变幻的速度，其缺省值为2

【例 5.4-5】（图5.4-4）

```
ezsurf('x*y','circ'); shading flat; view([-18,28]) % 双曲抛物面 (鞍面)  
C=summer; CC=[C; flipud(C)]; % C为64x3矩阵，上下对称延拓  
colormap(CC), spinmap(30,4) % 变幻30秒，每次变幻4行
```



# 电影动画

- ❑ `M(i)=getframe`可以在当前的绘图窗口中对当前图形进行拍照并且保存在一个结构体数组中，`i`为帧序号
- ❑ `M(i)`的存储类型为一种特殊的结构体，包含了两个域，`cdata`与`colormap`
- ❑ `movie(M,k,fps)`可以将之前定义好的画面按照顺序以每秒`fps` (默认为12) 帧的速度进行播放，并且共重复`k`次。若计算机出线卡顿，则会尽可能的接近`fps`帧来播放
- ❑ 简单的动画一般可以通过变化的线条绘制，观察角的变化，图形的刚体运动，图像函数的变化，或图形颜色的变化来进行生成。
- ❑ 影片动画的优点是便于整理和多次反复播放，也可以进行剪辑处理，缺点则是非常耗费存储空间。

## 例5.4-6 (有改动)

### 【例5.4-6】

```
clear all
```

```
figure('Renderer','zbuffer')
```

%z缓存渲染技术

```
x=3*pi*(-1:0.05:1);y=x;[X,Y]=meshgrid(x,y);
```

```
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;
```

%前例的二维sinc函数

```
h=surf(X,Y,Z);colormap(jet);
```

```
axis tight off
```

```
n=48;
```

%制作四十八帧的动画

```
MF(n)=struct('cdata',[],'colormap',[]);
```

%事先为每一个画面申请存储空间,申请最后一个元素为结构体,自动前面的也申请了

```
for ii=1:n
```

```
    rotate(h,[0 0 1],5);
```

%绕z轴旋转5度

```
    MF(ii)=getframe;
```

%每次旋转后获取画面并存储

```
end
```

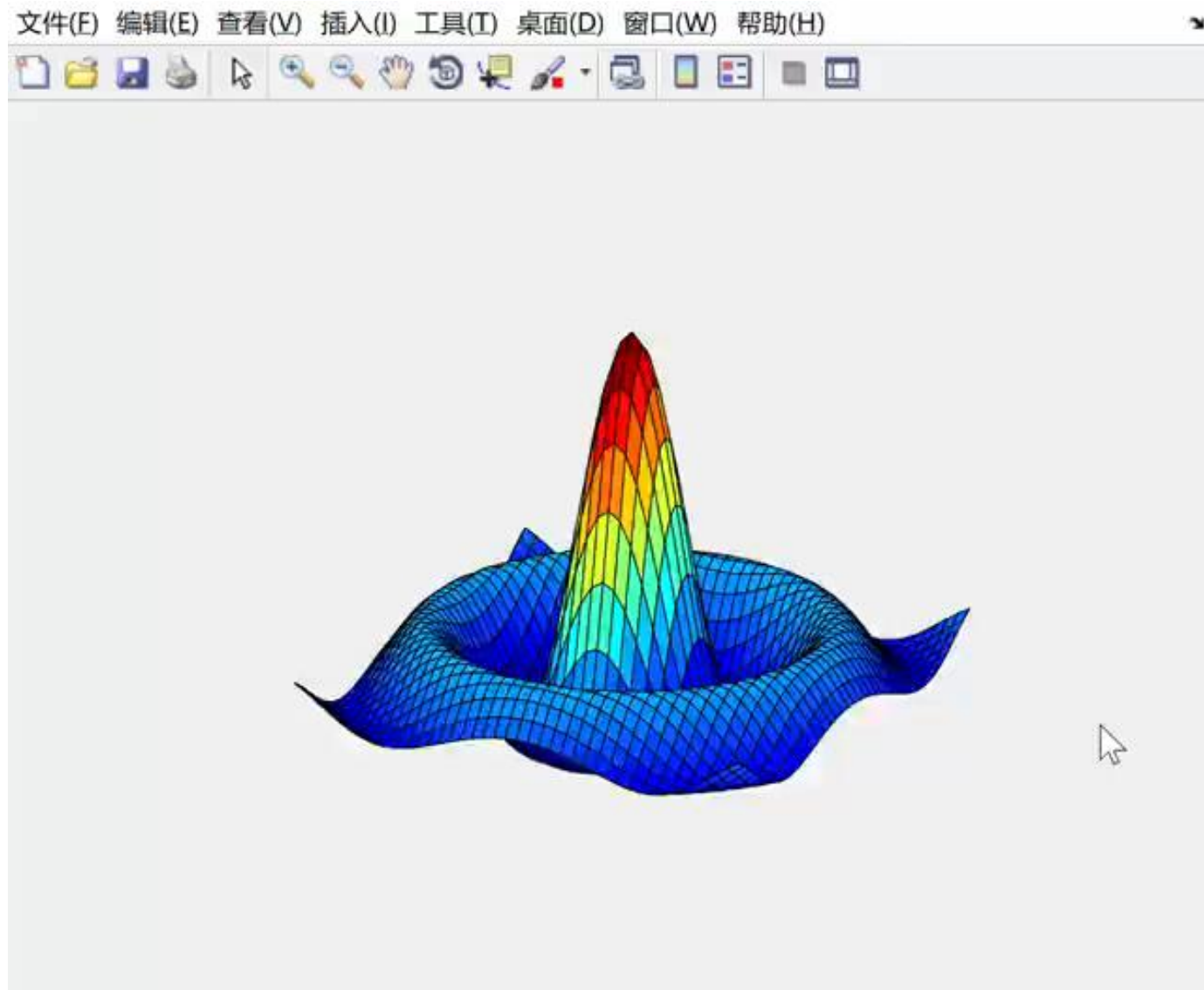
```
shg,axis off
```

```
movie(MF,1,24)
```

%每秒24帧,播放一次

%实际运行时会看到两次,第一次为计算和存储,第二次才是播放(恰为两秒)

## 例5.4-6 (可自行搜索gif或avi储存方法)



# 实时动画-例5.4-7

- ❑ **drawnow**可以随时将动态的绘图展示在绘图窗口，函数 **pause**可以辅助控制两次绘图之间的时间差，但受到绘图运算时间的影响，实际的绘图时间间隔要更长一点。
- ❑ **EraseMode**已经不被新版本的MATLAB支持了，这个选项可以直接去掉，不会影响绘图的效果与质量。

【例 5.4-7】制作红色小球沿一条带封闭路径的下旋螺旋线运动的实时动画

```
function f=anim_zzy1(K,ki)
```

```
% anim_zzy1.m 演示红色小球沿一条封闭螺旋线运动的实时动画
```

```
% 仅演示实时动画的调用格式为          anim_zzy1(K)      此时默认 ki=n/2
```

```
% 既演示实时动画又拍摄照片的调用格式为      f=anim_zzy1(K,ki)
```

```
% K          红球运动的循环数（不小于1）
```

```
% ki         指定拍摄照片的瞬间，取 1 到“自变量采样总点数 n ”间的任意整数。
```

```
% f          存储拍摄的照片数据，可用image(f.cdata)观察照片。
```

```
t1=(0:1000)/1000*10*pi;x1=cos(t1);y1=sin(t1);z1=-t1;%向下螺旋线
```

```
t2=(0:10)/10;x2=x1(end)*(1-t2);
```

```
y2=y1(end)*(1-t2);z2=z1(end)*ones(size(x2));          %下端到圆心
```

```
t3=t2;z3=(1-t3)*z1(end);x3=zeros(size(z3));y3=x3;      %中心向上
```

```
t4=t2;x4=t4;y4=zeros(size(x4));z4=y4;                  %上端到圆上
```



# 实时动画-例5.4-7

【例 5.4-7】制作红色小球沿一条带封闭路径的下旋螺线运动的实时动画

```
x=[x1 x2 x3 x4]; n=length(x); %总轨迹长度，显然螺旋线时间长
if nargin<2 %若第二变量缺省则ki取中间但不会拍照
    ki=fix(n/2);
end
y=[y1 y2 y3 y4]; z=[z1 z2 z3 z4];

shg
plot3(x,y,z,'Color',[1,0.6,0.4],'LineWidth',2.5) %调出橘红色
axis off

h=line('xdata',x(1),'ydata',y(1),'zdata',z(1),'Color',
[1 0 0],'Marker','.', 'MarkerSize', 40);
%去掉了EraseMode，初始小球（红点）位置

KK=K*n;
text(-1,-0.85,-36,'倒计时') %文本框显示
KK=KK-1;
htext=text(-1,-1,-40,int2str(KK)); %初始倒计数为(线长x循环数-1)
```

# 实时动画-例5.4-7

【例 5.4-7】制作红色小球沿一条带封闭路径的下旋螺线运动的实时动画

% 使小球运动

```
i=2;j=1;
```

```
while 1 %无限循环，需要break来确保不会死循环
```

```
set(h,'xdata',x(i),'ydata',y(i),'zdata',z(i)); %改变小球位置
```

```
drawnow; % 实时动画的核心语句，立即绘图
```

```
pause(0.0005) % 停顿时间万分之五秒
```

```
i=i+1;KK=KK-1;set(htext,'string',int2str(KK)) %倒计时更新
```

```
if nargin==2 && nargout==1 %
```

```
if(i==ki&&j==1);f=getframe(gcf);end %拍照
```

```
end
```

```
if i>n %如果完成了一个周期，判断周期是否过量，过量则结束循环体
```

```
i=1;j=j+1;
```

```
if j>K;break;end
```

```
end
```

```
end
```

```
shg,f=anim_zzy1(2,450);
```

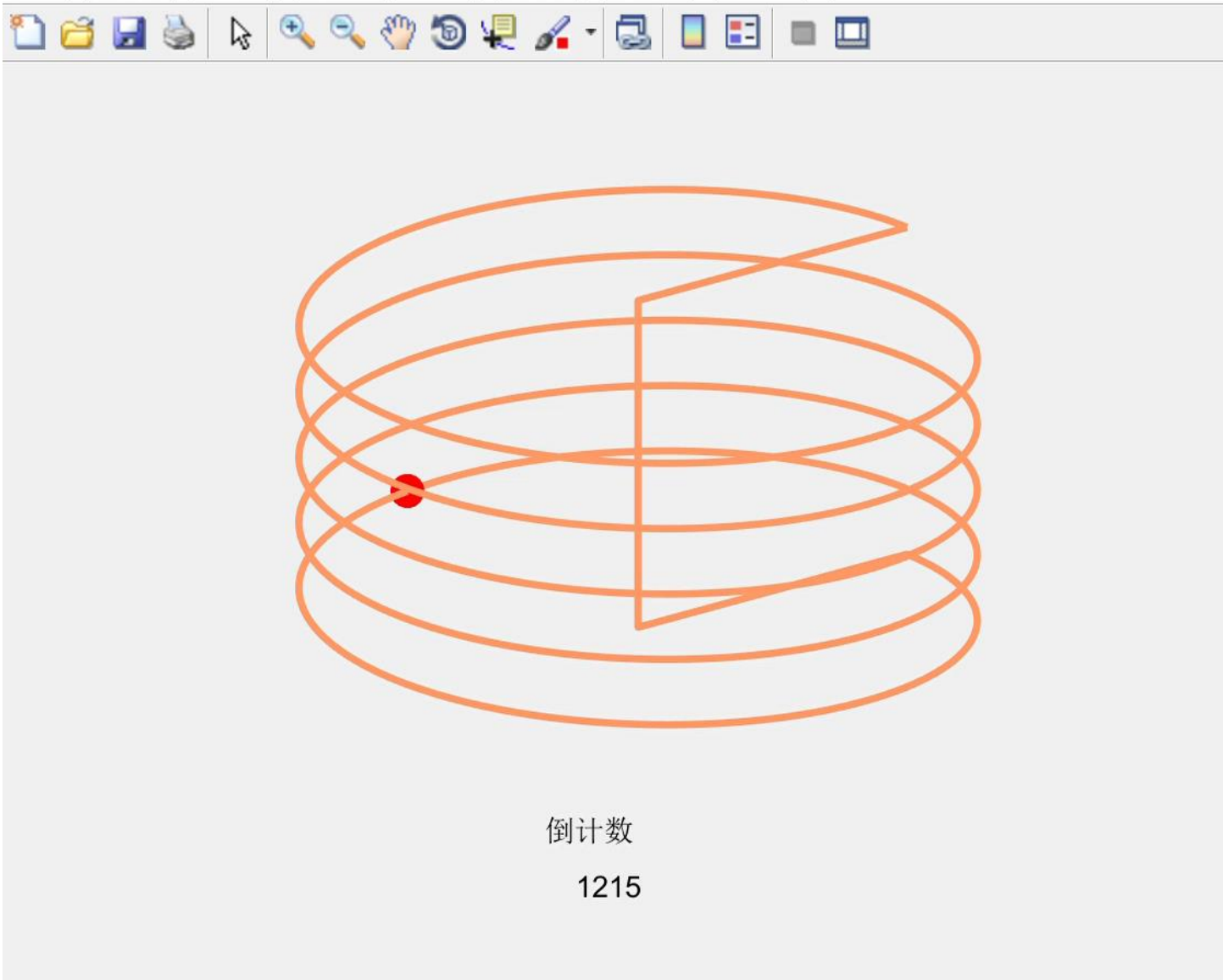
```
image(f.cdata),axis off
```

%最终停留在第450个画面(倒计时1819)

%演示并保存第450个画面

# 例5.4-7

文件(E) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)



# GIF制作实例-借鉴自 “百度经验”

```
clear;clc;
figure %新建一张图
axis([0 5 0 2])%定义x轴（从0到5）和y轴的范围（从0到2）
for i=1:4
    if i==1
        text(i,1,'数','fontsize',40,'color','red');%第一帧： ‘数’ 字
    end
    if i==2
        text(i,1,'学','fontsize',40,'color','red');%第二帧： ‘学’ 字
    end
    if i==3
        text(i,1,'实','fontsize',40,'color','red');%第三帧： ‘实’ 字
    end
    if i==4
        text(i,1,'验','fontsize',40,'color','red');%第四帧： ‘验’ 字
    end
    picname=[num2str(i) '.fig'];%每一帧暂存的文件名(如第一帧1.fig)
    hold on % 写后面的字时，不把前面的字冲掉
    saveas(gcf,picname) %进行上述暂存，每一帧存为一个fig
end
```

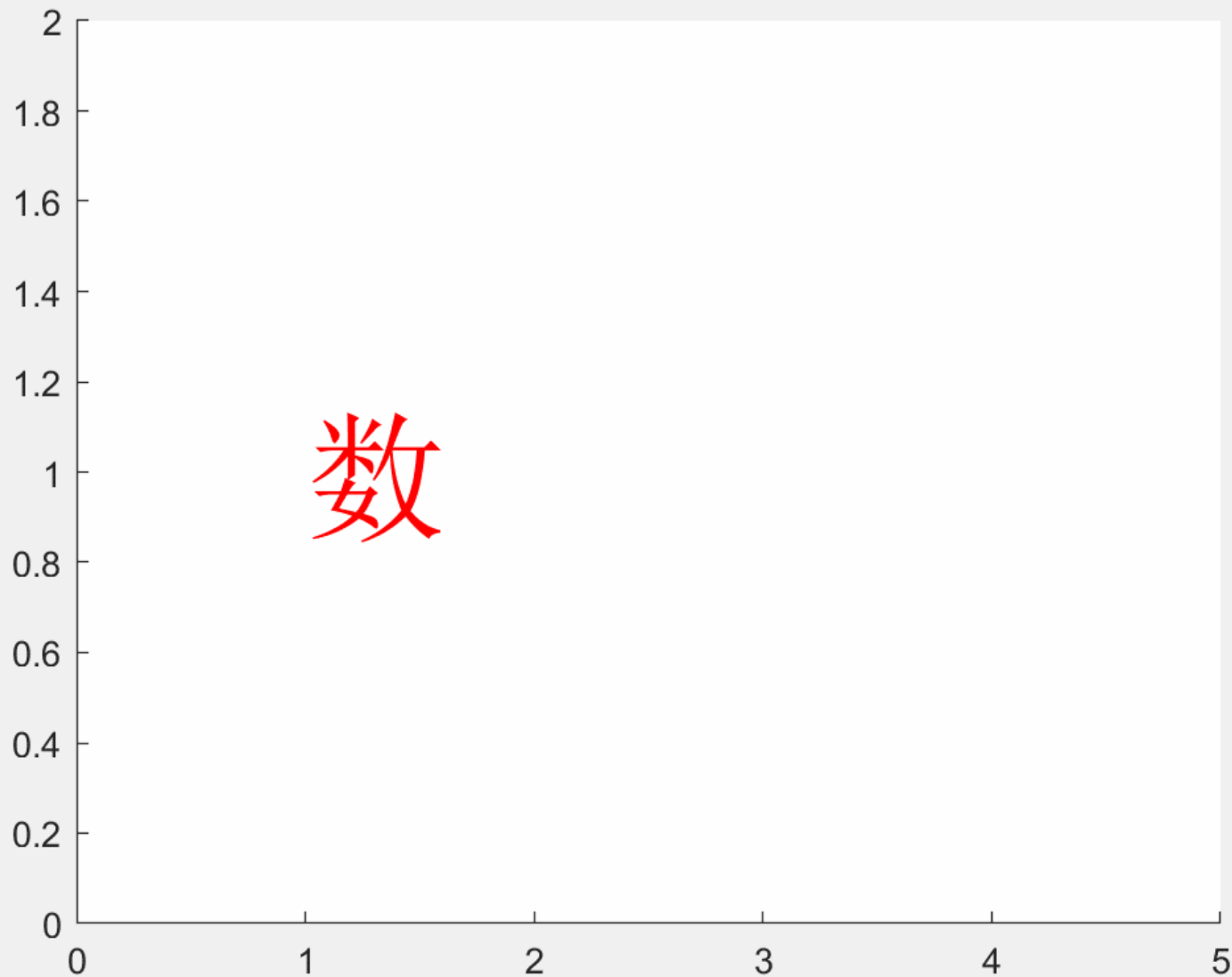
# GIF制作实例-借鉴自 “百度经验”

```
for i=1:4
    picname=[num2str(i) '.fig'];
    open(picname) %读取第i帧
    frame=getframe(gcf); %从当前图（刚读取的）提取帧（frame）

    im=frame2im(frame);%先将帧转化为彩色图像
    [I,map]=rgb2ind(im,20);
    %制作gif文件，图像必须是index索引图像，所以需要再次转化

    if i==1
        imwrite(I,map,'shuxueshiyan.gif','gif',...
            'Loopcount',inf,'DelayTime',0.5);%第一帧创建新gif，无限循环
    else
        imwrite(I,map,'shuxueshiyan.gif','gif',...
            'WriteMode','append','DelayTime',0.5);%后面帧的追加
    end;
    close all
end
```

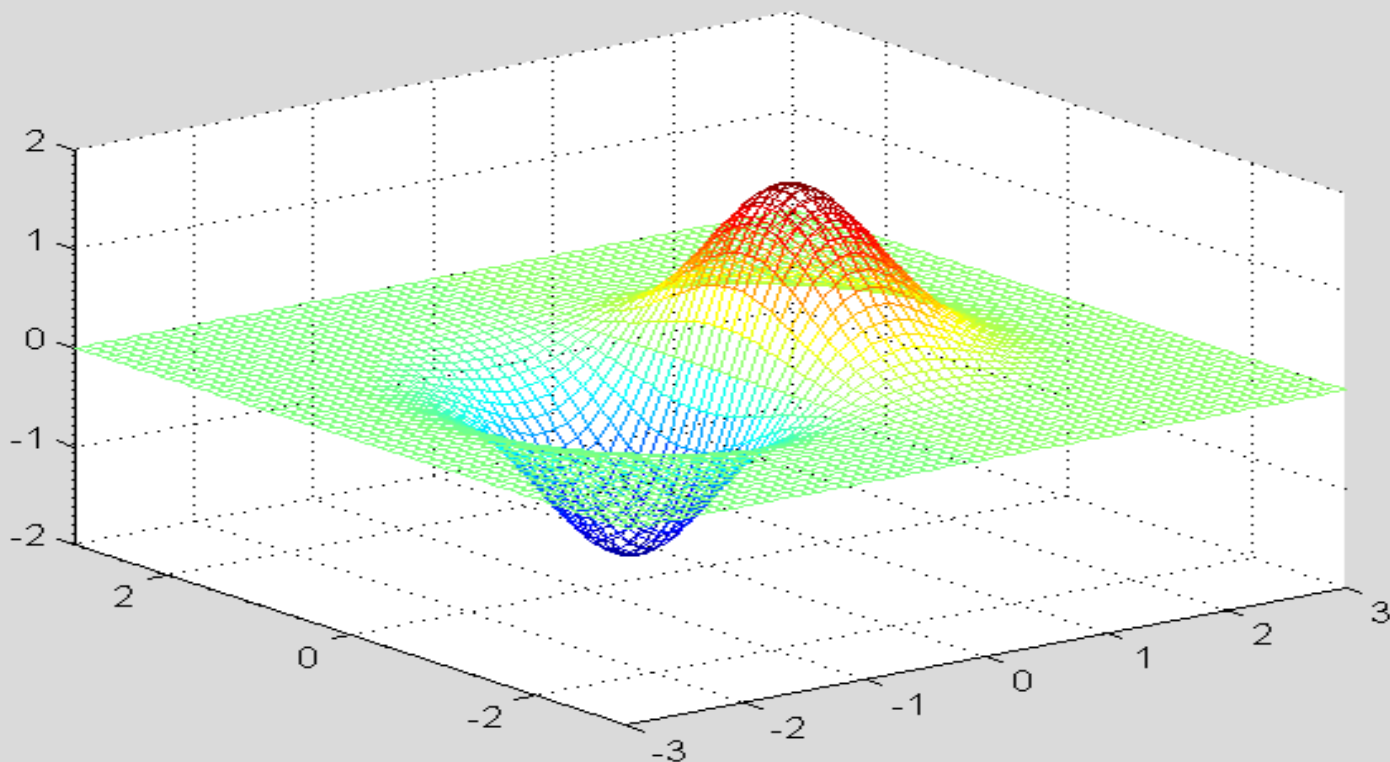
# GIF制作实例-借鉴自 “百度经验”



# 第17周参考作业（无需提交）

## □ Q1. 习题5第6题（**写出两种方法的MATLAB源代码，注意透视**）

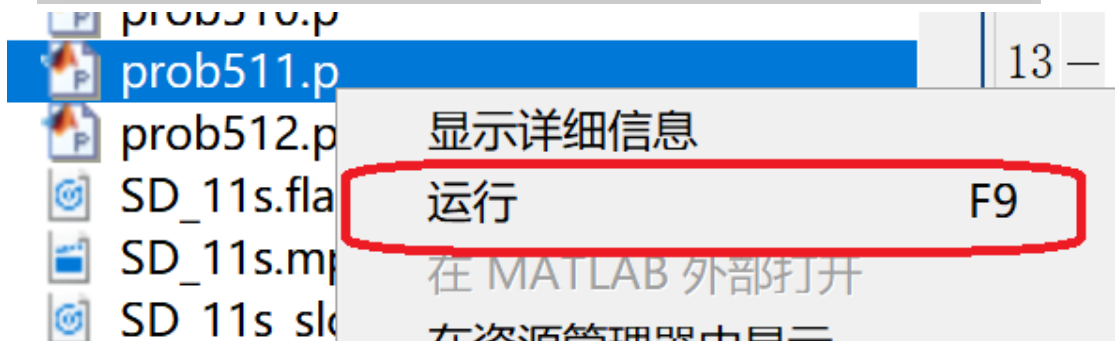
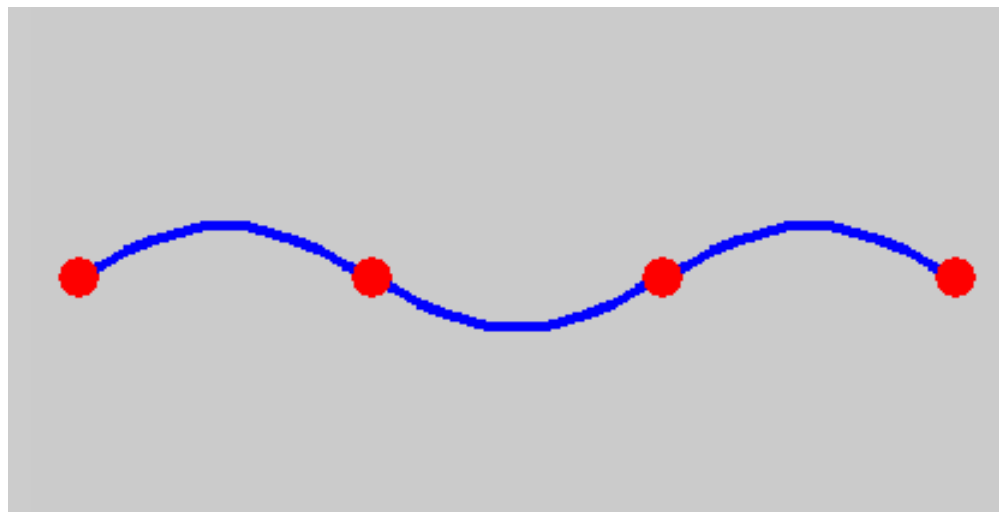
采用两种不同方法绘制  $z = 4xe^{-x^2-y^2}$  在  $x, y \in [-3, 3]$  的如图 p5-6 的三维（透视）网格曲面。（提示：ezmesh; mesh; hidden）



# 第17周参考作业（无需提交）

□ Q2. 习题5第11题（**写出M函数源代码, prob511.p文件为已编译不可见源码的MATLAB代码, 右键运行即可**）

利用影片动画法, 据函数  $f(x, t) = \sin(x)\sin(t)$  制作如图 p5-10 所示驻波动画。在做题前, 先运行 prob511.p 产生的演示动画。（提示：用 2 个 line 分别产生带图柄的线和点对象；**不用擦除模式(否则会警告)**；用 set 通过线图柄操作线位置；getframe; movie）





# 第17周参考作业（无需提交）

- Q3. 设函数  $z(x, y, t) = e^{-\frac{[(x-1)^2+y^2]}{0.1t+1}} + e^{-\frac{[(x+1)^2+y^2]}{0.02t^2+1}}$ ,  $-3 \leq x \leq 3, -3 \leq y \leq 3, 0 \leq t \leq 10$ , 容易发现这是一个不断向外扩散的函数。
- 请按照时间  $t$  的变化范围将  $[0, 10]$  分成若干等分的时间段, 并且在每个时间点  $t = t_i$  以涂色等位线的方式绘制函数  $z(x, y, t_i)$  的实时情况, 最终将等位线以实时动画的方式展示出来。并在  $t = 10$  的最终状态上加上合适的标识数据。
- 请在作业纸上提供定义函数与绘图的代码（关键步骤可以写上一些注释），无需写出绘图结果。

# 不用交的作业（重要）

- 将今天讲过的例题尝试自己键入并运行一遍
- 学习电影动画的gif或avi的存储方式，并顺利完成期末作业的第二题。
- 课本习题5第5, 7, 8, 9, 10, 12题请自行阅读、思考、测试完成。

# 感谢同学们认真听课!

---

欢迎同学们积极提问、交流

[sysumatlab@163.com](mailto:sysumatlab@163.com)