

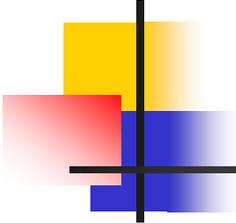
Matlab编程与应用

第三讲

中国科技大学信息学院

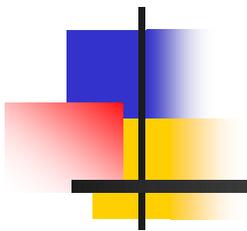
陆伟

luwei@ustc.edu.cn



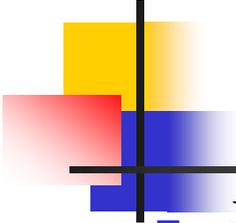
本讲内容

- part1: 二维绘图
- part2: 三维绘图
- part3: 动态显示



Part1:

二维绘图



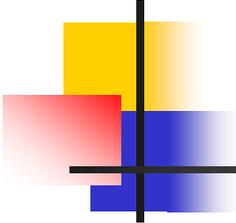
主要内容

- 基本绘图函数：**plot**函数

- 线型 (Line Style)、点型 (Marker)、颜色 (Color)

- 二维绘图的辅助操作

- 标注：图形名称、坐标轴名称、曲线标注、图例
- 坐标轴控制
- 图形保持 (同一坐标轴绘制多个图形)
- 窗口分割 (同一窗口含有多个坐标轴)
- 复数情况



plot基本绘图

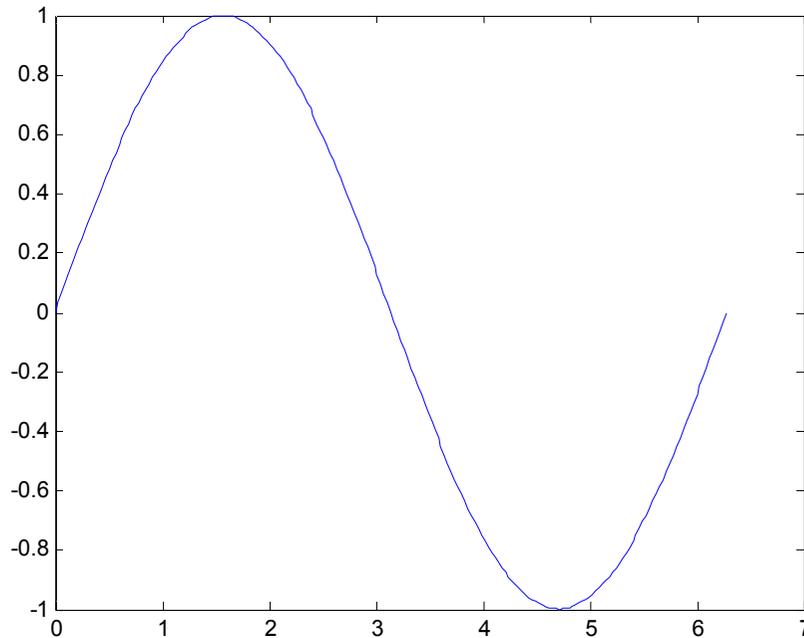
- **plot**: 最基本的绘图指令
- 对**x**坐标及对应的**y**坐标绘图
- 例:

```
x = 0 : 0.01*pi : 2*pi ;
```

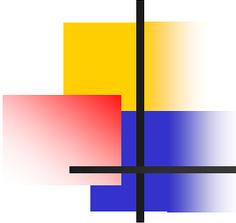
```
y = sin(x) ;
```

```
plot(x,y) ;
```

plot基本绘图



- 要求x向量、y向量长度必须相同！
- 若只给定一个向量。则横坐标是索引值
`plot(y)`

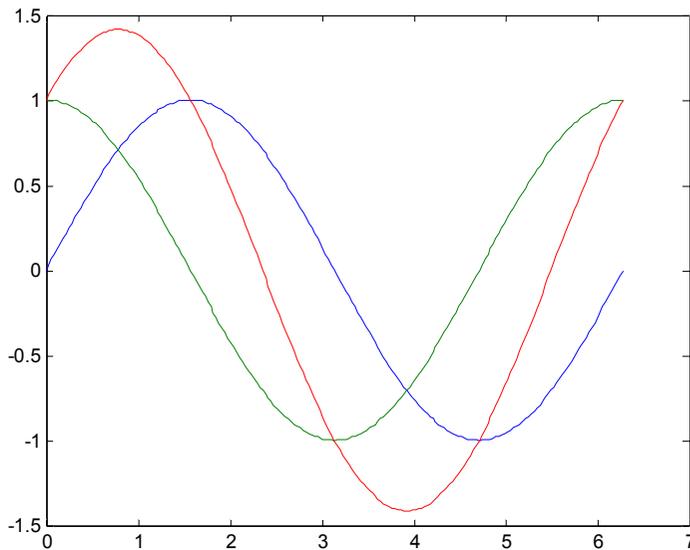


plot基本绘图

- 一次画出多条曲线
- 例:

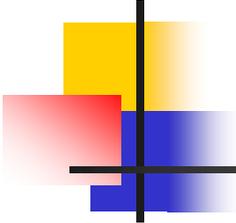
```
x = 0 : 0.01*pi : 2*pi ;  
y1 = sin(x);  
y2 = cos(x);  
y3 = sin(x)+cos(x);  
plot(x, y1, x, y2, x, y3);
```

plot基本绘图



- 画多条曲线时，Matlab自动赋予每条曲线不同的颜色。
- 也可以把y1、y2、y3组成一个矩阵，此时会对矩阵的每个行向量作图。

```
z = [y1 ; y2 ; y3];  
plot(x,z);
```



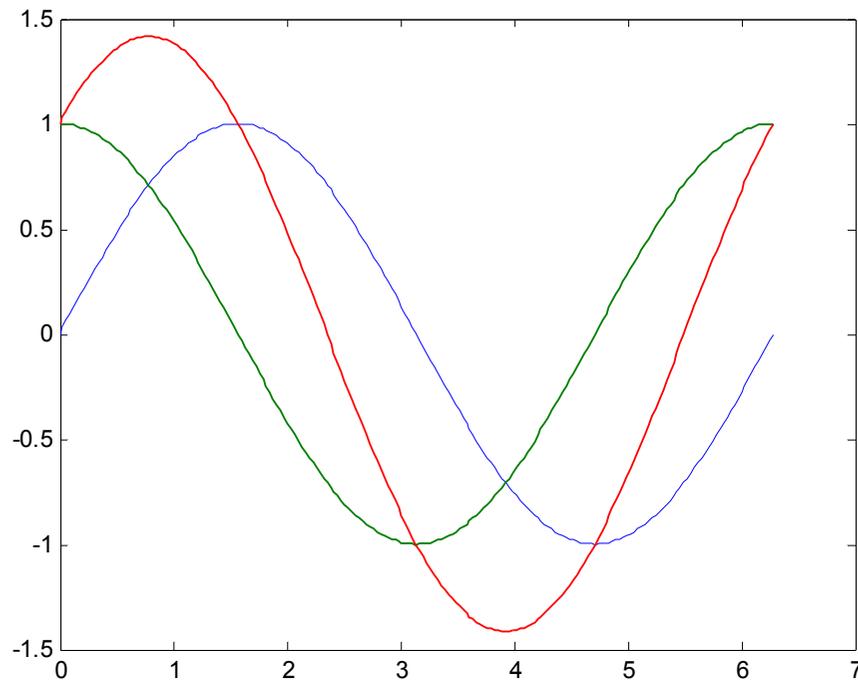
plot基本绘图

- x 和 y 都是矩阵
- 则 $\text{plot}(x,y)$ 用 x 的每一个行向量与 y 对应行向量作图

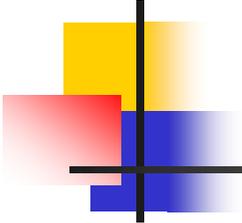
plot基本绘图

—设置曲线的线型 (Line Style)

例:



```
plot(x,y1,'-',x,y2,'--',x,y3,':');
```



plot基本绘图

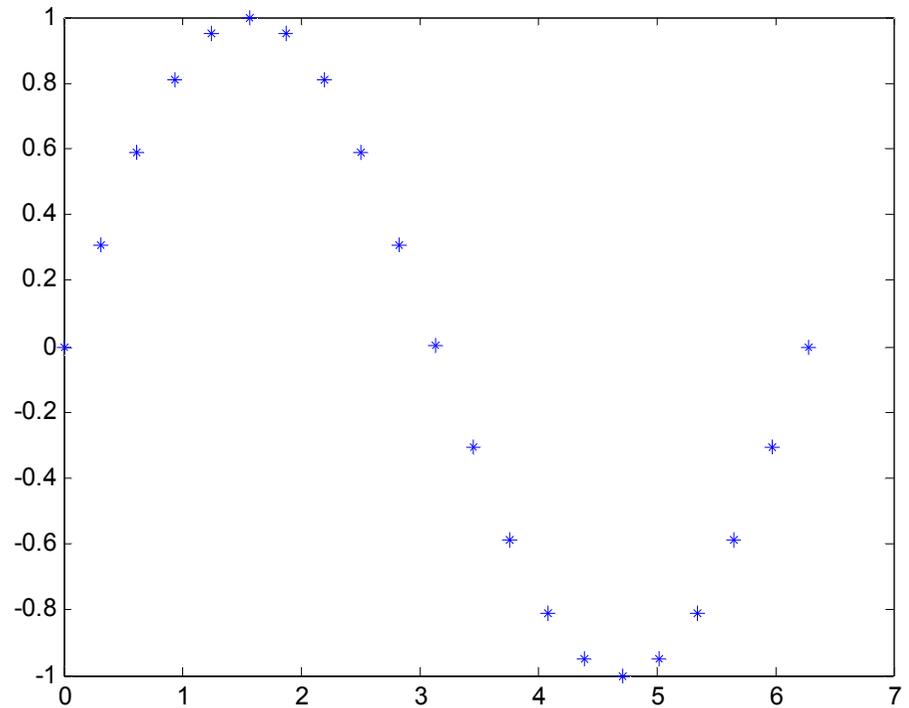
—设置曲线的线型 (Line Style)

plot 函数的Line Style	说明
-	实线(默认值)
--	虚线
:	点线
-.	点虚线

plot基本绘图

—设置曲线的点型 (Marker)

```
x = 0 : pi /10 : 2*pi;  
y = sin(x);  
plot(x,y,'*')
```

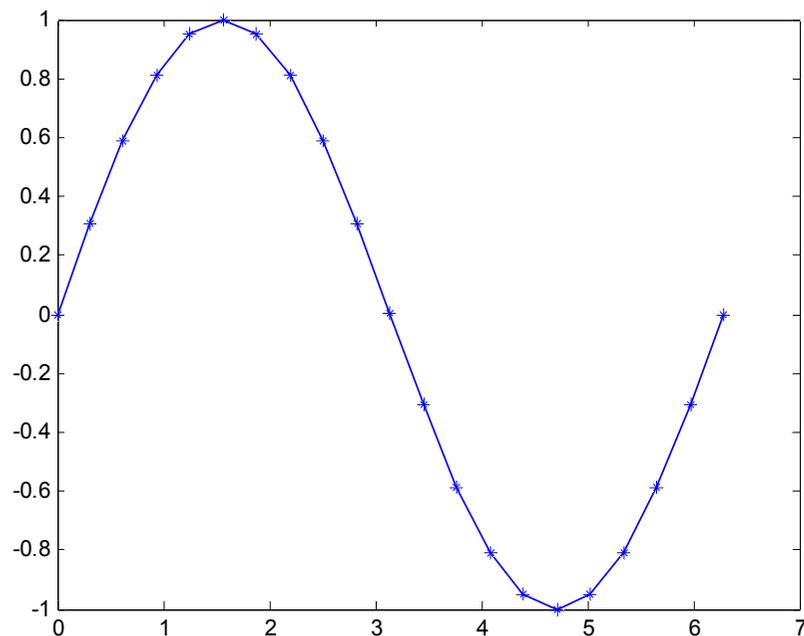


plot基本绘图

—设置曲线的点型 (Marker)

- 也可以同时设置画线与点型

```
x = 0 : pi /10 : 2*pi;  
y = sin(x);  
plot(x,y,'*-')
```



plot基本绘图

—设置曲线的点型 (Marker)

plot 函数的点型 (Marker)

说明

O

圆形

+

加号

X

叉号

*

星号

.

点号

^

朝上三角形

v

朝下三角形

plot基本绘图

—设置曲线的点型 (Marker)

plot 函数的点型 (Marker)	说明
>	朝右三角形
<	朝左三角形
square	方形
diamond	菱形
pentagram	五角星形
hexagram	六角星形
None	无点型(默认值)

plot基本绘图

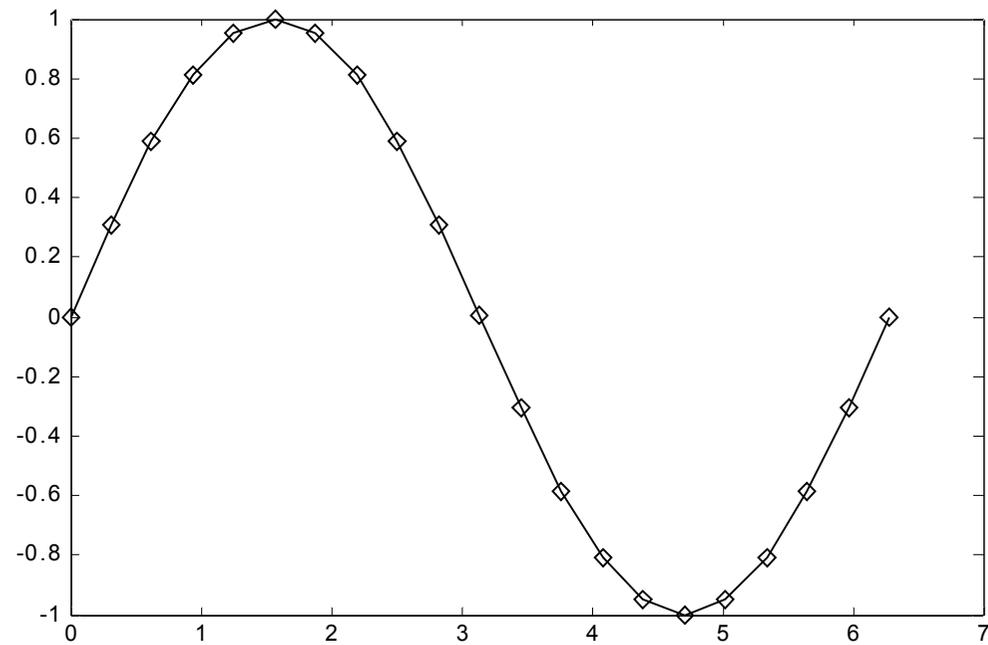
—设置曲线的颜色 (Color)

- 用黑色點線畫出正弦波
- 每一資料點畫上一個小菱形

```
x = 0 : pi /10 : 2*pi;  
y = sin(x);  
plot(x,y,'k:diamond') %其中 “k”代表黑色,  
% “: ” 代表线型  
% “diamond”代表菱形的点
```

plot基本绘图

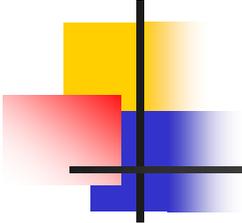
—设置曲线的颜色 (Color)



plot基本绘图

—设置曲线的颜色 (Color)

Plot函数的曲线颜色	曲线颜色	RGB值
b	蓝色(Blue)	(0,0,1)
c	青蓝色(Cyan)	(0,1,1)
g	绿色(Green)	(0,1,0)
k	黑色(Black)	(0,0,0)
m	紫黑色(Magenta)	(1,0,1)
r	红色(Red)	(1,0,0)
w	白色	(1,1,1)
y	黄色(Yellow)	(1,1,0)



plot基本绘图

—加入说明文字

- 为增进图形的可读性，常常需要对图形或坐标轴加入说明。

指令	说明
title	图形的标题
xlabel	x 轴的说明
ylabel	y 轴的说明
legend	标注图例
text	图形中加入文字
gtext	使用鼠标定位文字位置

plot基本绘图

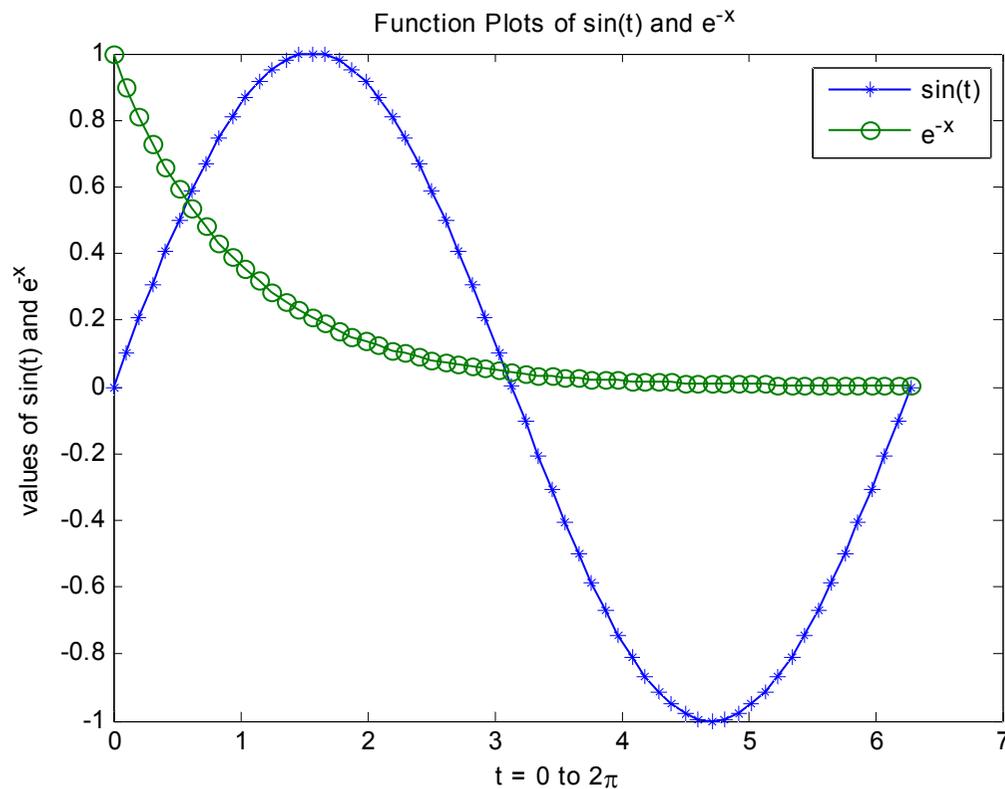
—加入说明文字

例

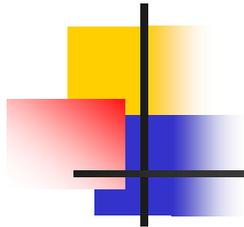
```
x = 0:pi/30:2*pi;  
y1 = sin(x);  
y2 = exp(-x);  
plot(x, y1, '--*', x, y2, ':o');  
xlabel('t = 0 to 2\pi');  
ylabel('values of sin(t) and e^{-x}');  
title('Function Plots of sin(t) and e^{-x}');  
legend('sin(t)', 'e^{-x}');
```

plot基本绘图

—加入说明文字



- **legend**指令：画出一小方块，给出每条曲线的说明
- “\”为特殊符号，用于产生上标、下标、希腊字母、数学符号等；遵循LaTeX数学模式
- 标题、坐标轴说明等的文字大小、位置也是可以修改的，见Matlab Help



■ 常用Latex用法

a^2 `a^{2}`

a_2 `a_{2}`

∞ `\inf ty`

\times `\times`

\oplus `\oplus`

\otimes `\otimes`

α `\alpha`

β `\beta`

γ `\gamma`

π `\pi`

τ `\tau`

Δ `\Delta`

δ `\delta`

Ω `\Omega`

plot基本绘图

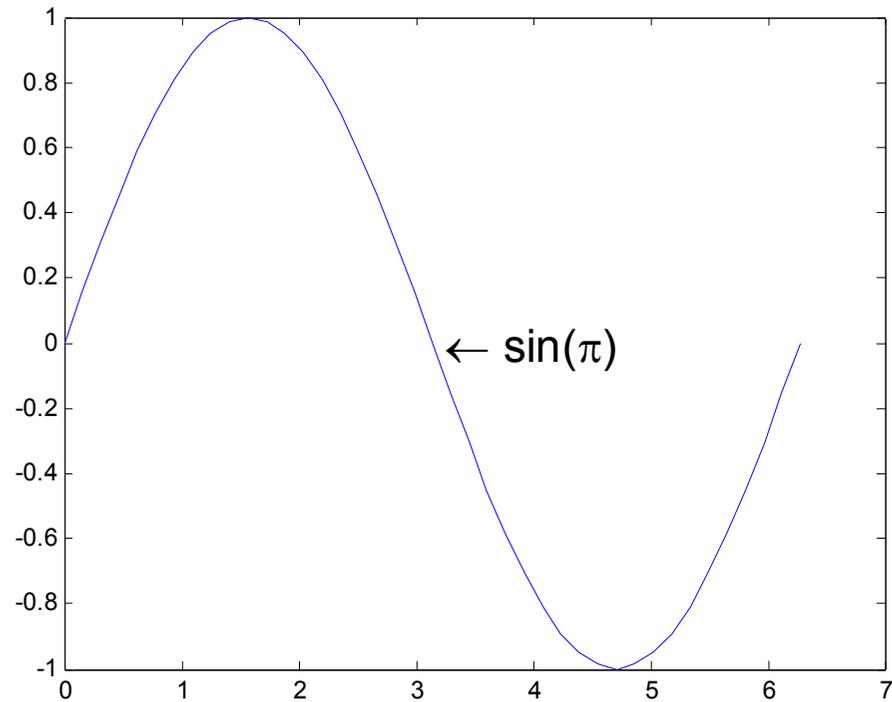
—加入说明文字

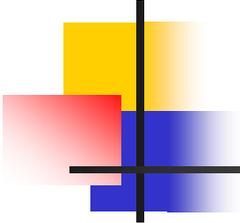
- `text(x, y, 'string')`
x、y : 文字起始坐标位置
string : 文字内容
- 例

```
x = 0:pi/20:2*pi;  
plot(x,sin(x))  
text(pi,0,' \leftarrow  
sin(\pi)', 'FontSize',18)
```

plot基本绘图

—加入说明文字





plot基本绘图

—加入说明文字

- 有时不确定说明文字位置，可用鼠标确定文字位置。
- `gtext('string')`

用鼠标单击图形，在选中点处放入说明文字 `string`

plot基本绘图

—hold函数

- 有时希望在已经画了一条曲线的图上再添加一条曲线。

```
x = 0:pi/20:2*pi;
```

```
plot(x,sin(x),'g');
```

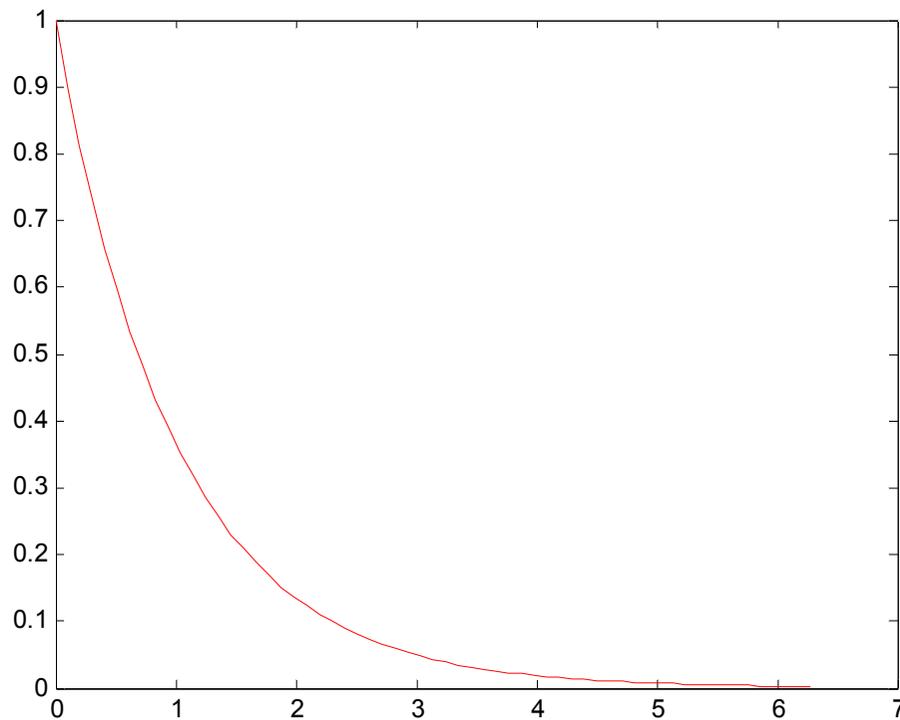
```
%再添加一条曲线
```

```
plot(x,cos(x),'r');
```

- 会得到想要的结果吗？

plot基本绘图

—hold函数



只画了后面一条
曲线，第一条被
擦除了。

plot基本绘图

—hold函数

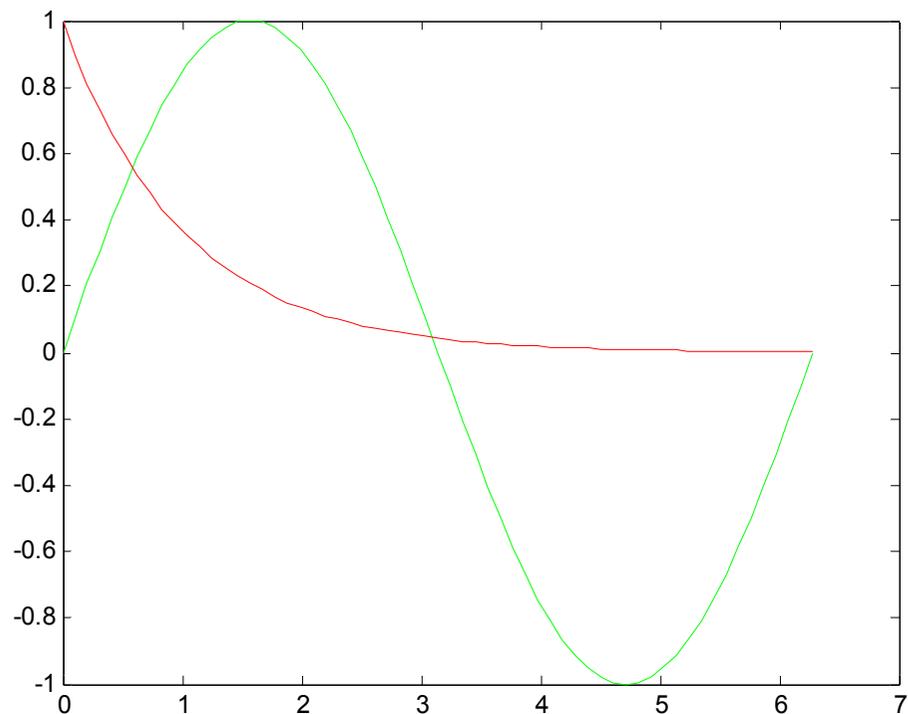
```
x = 0:pi/20:2*pi;
```

```
plot(x,sin(x),'g');
```

hold on

```
plot(x,cos(x),'r');
```

hold off



plot基本绘图

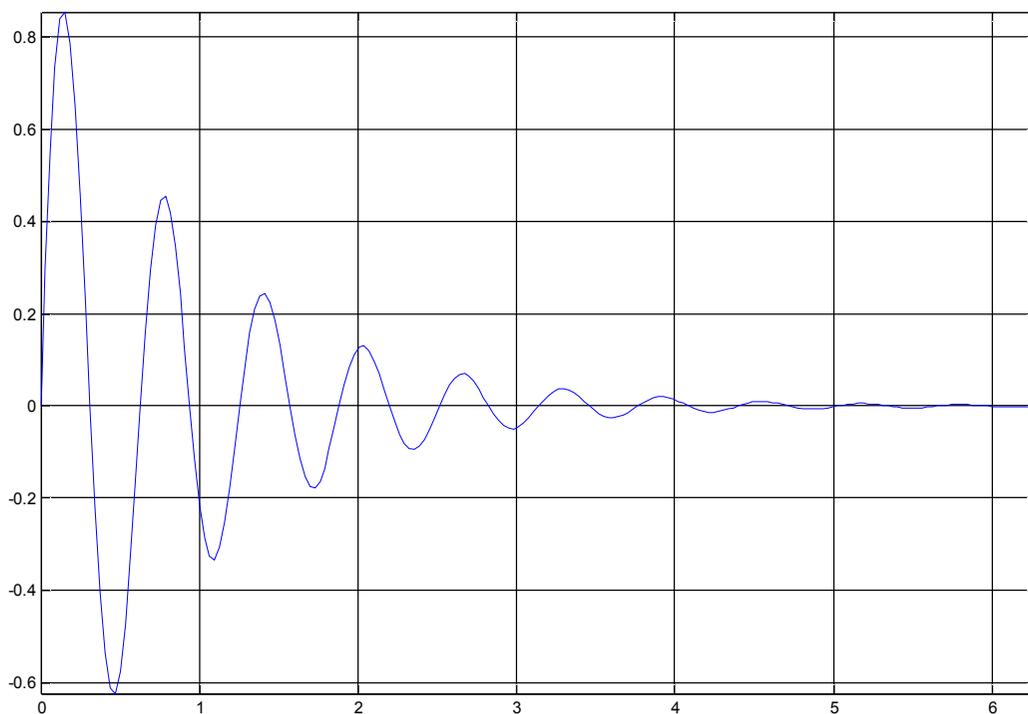
—给图形加上网格

- 有时希望在图形上加上网格，以便更好地观察波形的变化。
- **grid**函数

```
x = 0:pi/100:2*pi;  
y = exp(-x).*sin(10*x);  
plot(x,y)  
grid on %给图形加上网格
```

plot基本绘图

—给图形加上网格



- 去掉网格:
`grid off`
- 只用`grid` 则在
`on`、`off`状态间切
换
- `grid minor`

plot基本绘图

—双坐标轴

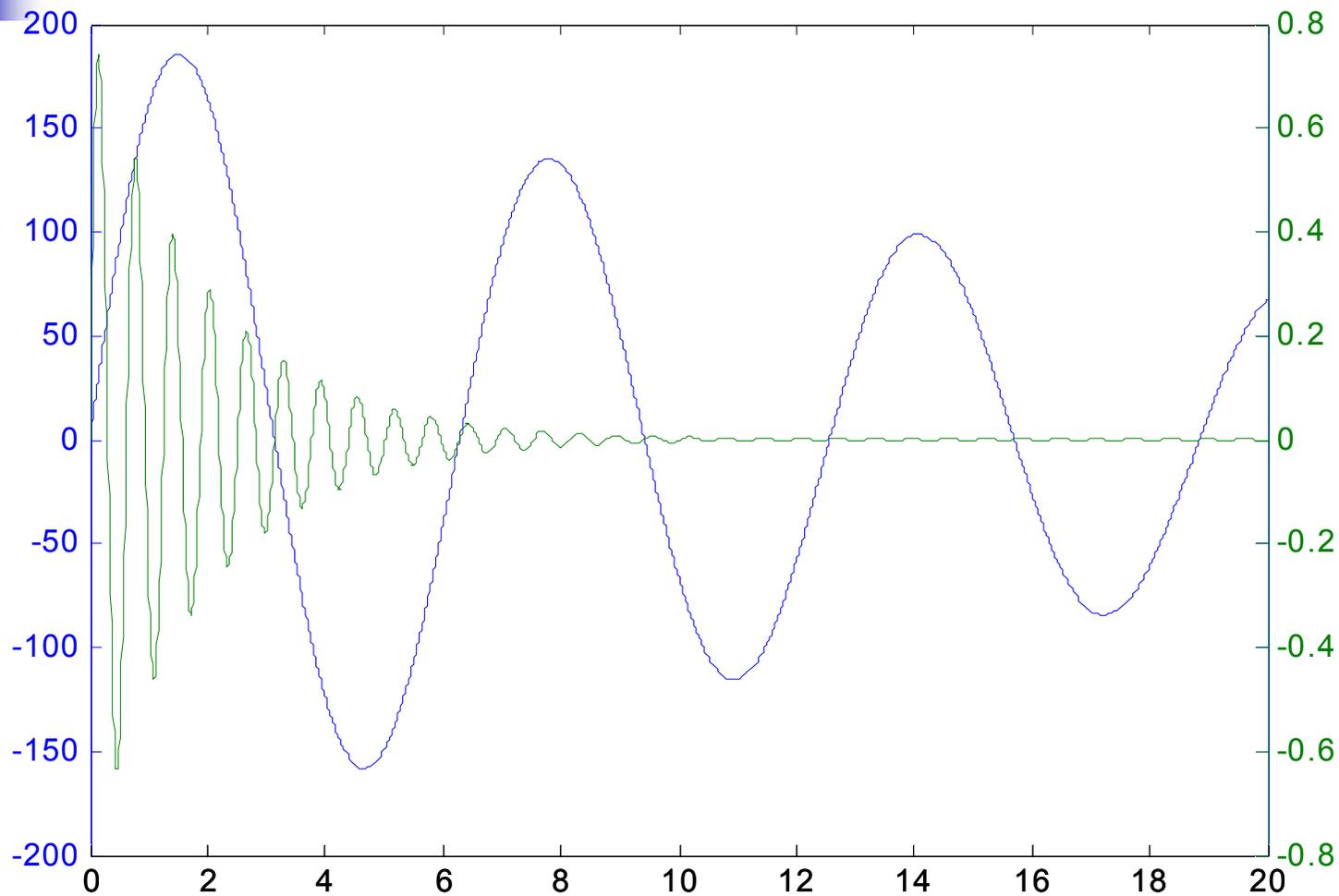
- **plotyy**函数

在同一窗口画幅度相差很大的两条曲线，采用不同的y轴刻度。

```
x = 0:0.01:20;  
y1 = 200*exp(-0.05*x).*sin(x);  
y2 = 0.8*exp(-0.5*x).*sin(10*x);  
plotyy(x,y1,x,y2,'plot');
```

plot基本绘图

—双坐标轴



plot基本绘图

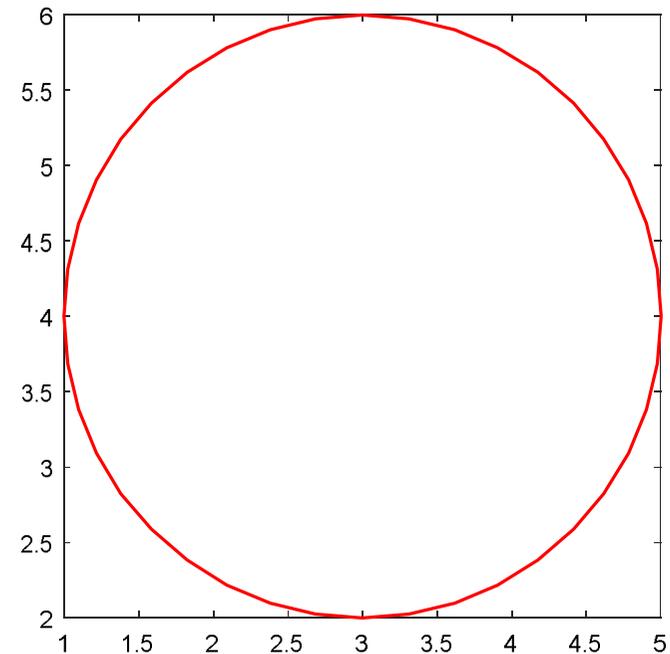
—坐标轴控制

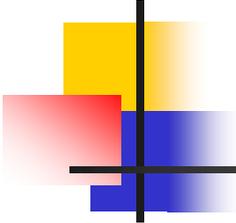
- 坐标轴刻度与长宽比控制
 - 默认坐标轴长宽比是窗口的长宽比
 - 可用**axis**指令加以修改

axis	([xmin xmax ymin ymax])	
axis normal		默认的长宽比
axis square		长宽比为1
axis equal		长宽比不变，但两轴刻度比例一致
axis tight		图轴紧贴图形
axis off		
axis on		

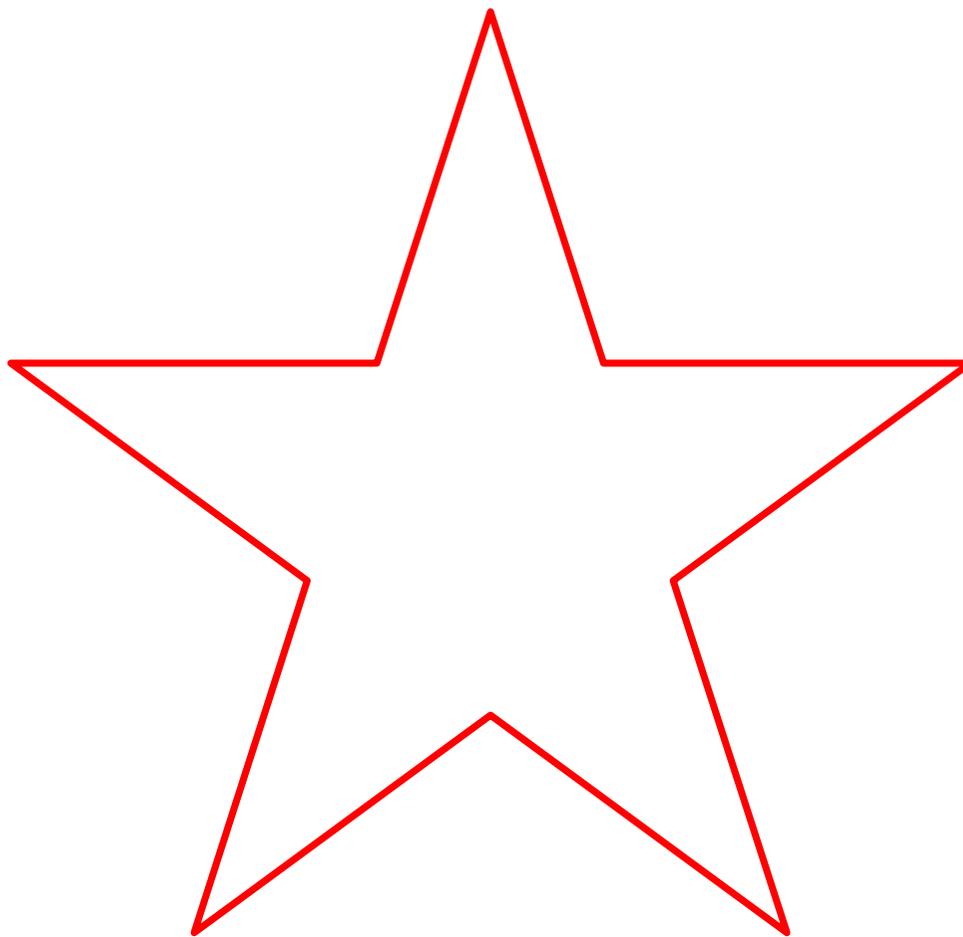
画一个圆

```
r = 2 ;  
theta = 0:pi/20:2*pi;  
x = r*cos(theta) + 3;  
y = r*sin(theta) + 4;  
plot(x,y,'r');  
axis equal
```





练习：画一个五角星



plot基本绘图

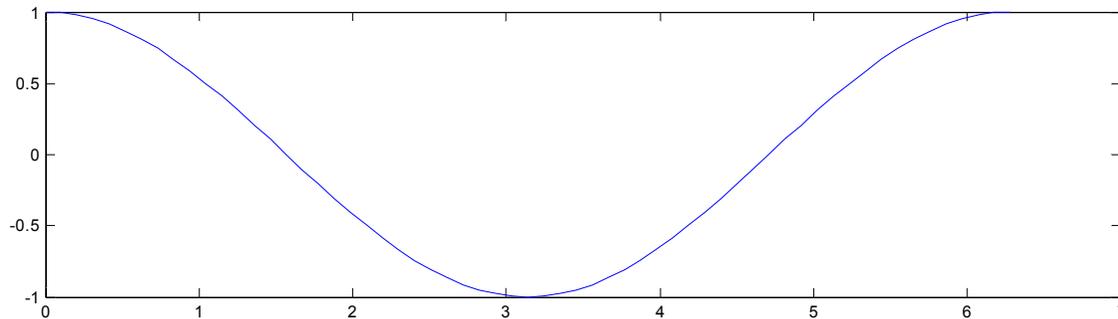
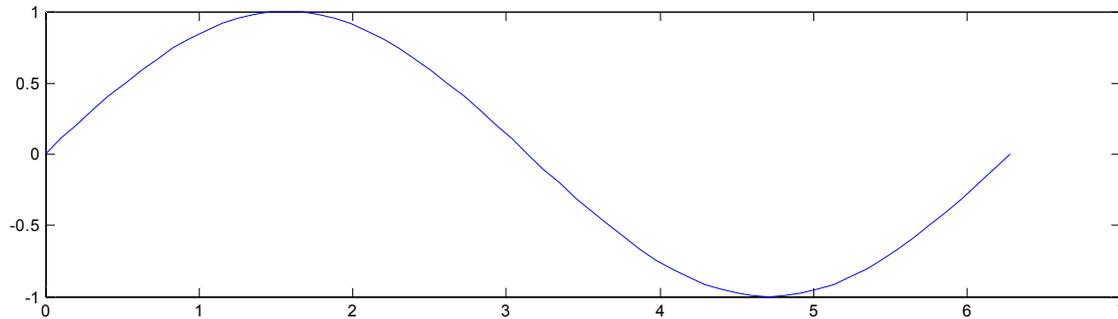
—在一个窗口中画多个图形

- 若要画多条曲线，我们可以：
 - 在同一窗口中画多条曲线
 - 打开多个窗口，在每个窗口中画一条曲线
 - 还可以利用**subplot**函数，在同一窗口中开设多个子窗口，每个子窗口画一条曲线。

```
x=0:pi/30:2*pi;  
subplot(2,1,1); plot(x,sin(x))  
subplot(2,1,2); plot(x,cos(x))
```

plot基本绘图

— 在一个窗口中画多个图形



plot基本绘图

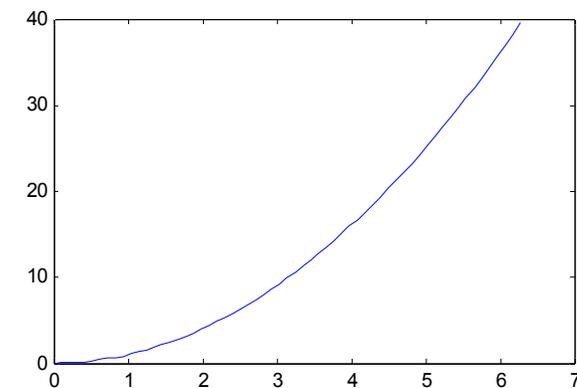
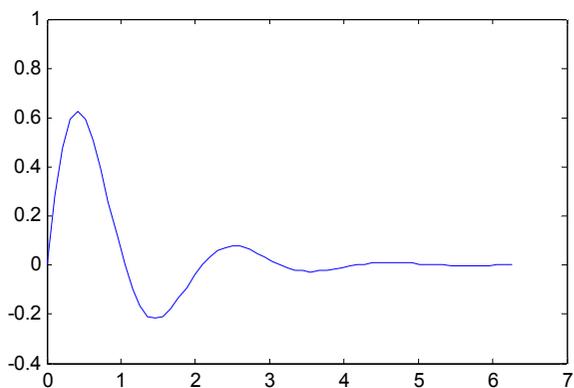
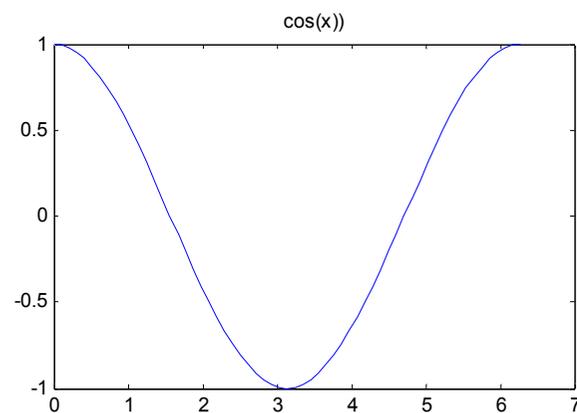
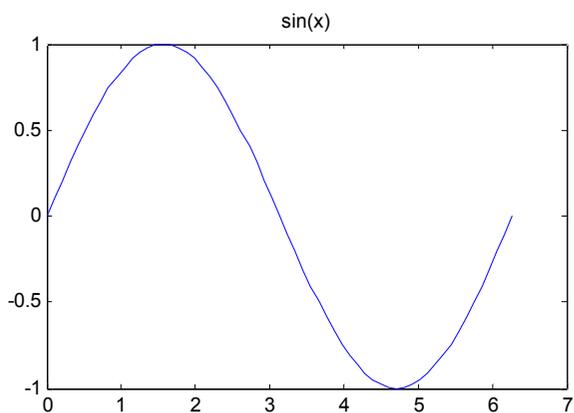
—在一个窗口中画多个图形

- `subplot(m,n,p)`
 - 将窗口分成 $m \times n$ 个子窗口
 - 下一个 `plot` 指令绘图于第 p 个子窗口
 - p 的算法为由左至右，一列一列
 - 也可写成: `subplot(mnp)`

```
x = 0:pi/30:2*pi
subplot(2,2,1) ;
plot(x,sin(x)) ;title('sin(x)')    %左上角
subplot(2,2,2) ;
plot(x,cos(x)) ;title('cos(x)')    %右上角
subplot(2,2,3); plot(x,exp(-x).*sin(3*x));
subplot(2,2,4); plot(x, x.^2);
```

plot基本绘图

— 在一个窗口中画多个图形

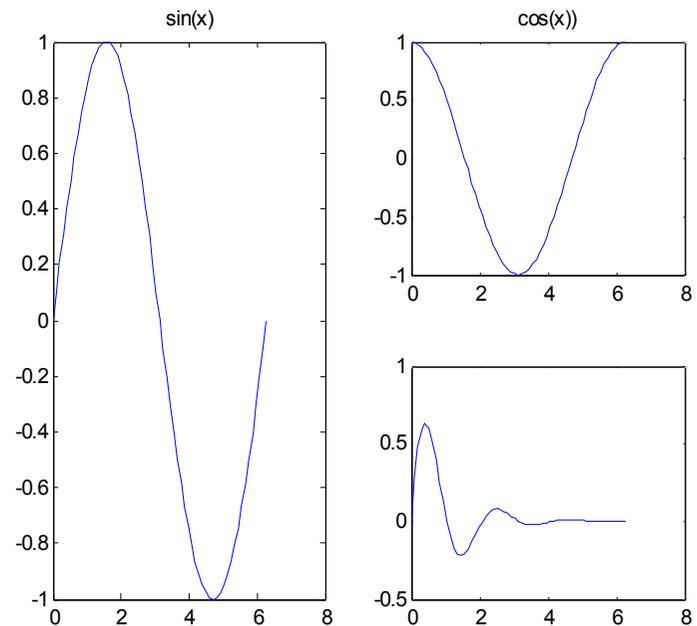


plot基本绘图

—在一个窗口中画多个图形

- 还可以组合子窗口

```
x = 0:pi/30:2*pi;  
subplot(2,2,[1,3]) ;  
plot(x,sin(x)) ; title('sin(x)')  
subplot(2,2,2) ;  
plot(x,cos(x)) ; title('cos(x)')  
subplot(2,2,4);  
plot(x,exp(-x).*sin(3*x));
```

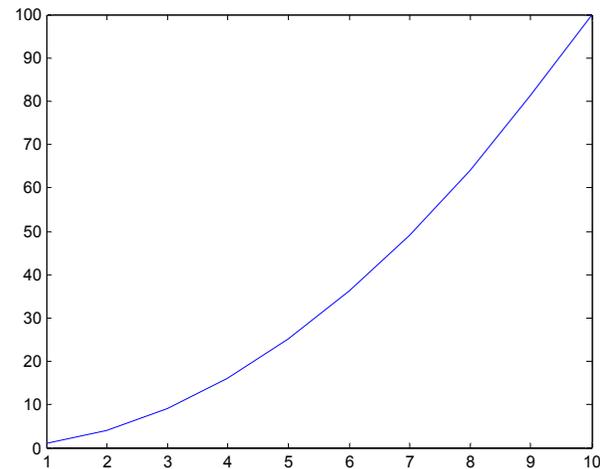


plot基本绘图

—当数组值为复数

- z 是一个复数向量
- `plot(z)`将 z 的实部和虚部当成 x 坐标和 y 坐标来画图,即
`plot(real(z), imag(z))`

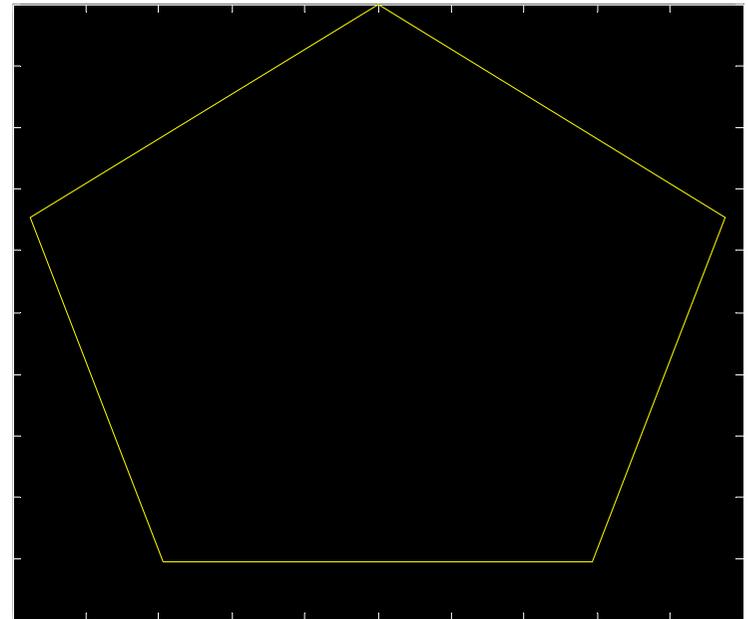
```
>> x = 1:10;  
>> y = x.^2;  
>> z =  
complex(x,y)  
>>plot(z)
```

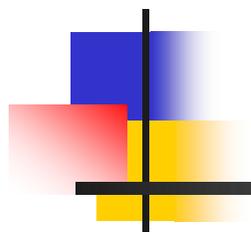


plot基本绘图

—当数组值为复数

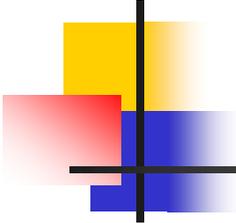
```
theta = pi/2 :  
2*pi/5 : (2*pi+pi/2) ;  
z = exp(j*theta) ;  
colordef black  
plot(z)  
axis equal tight
```





Part2:

三维绘图



三维作图

- 三维曲线
- 空间曲面
- 等高线
- $v=f(x,y,z)$ 的可视化

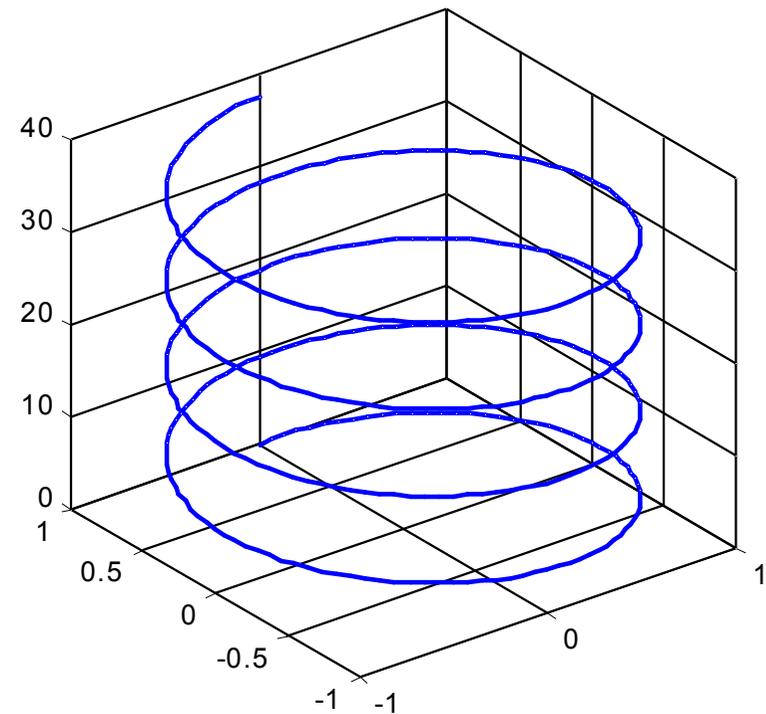
三维作图

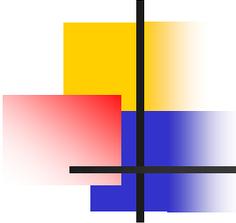
- 绘制三维曲线： $x=x(t)$, $y=y(t)$, $z=z(t)$

```
plot3(x,y,z,)
```

例：绘制三维螺旋线：

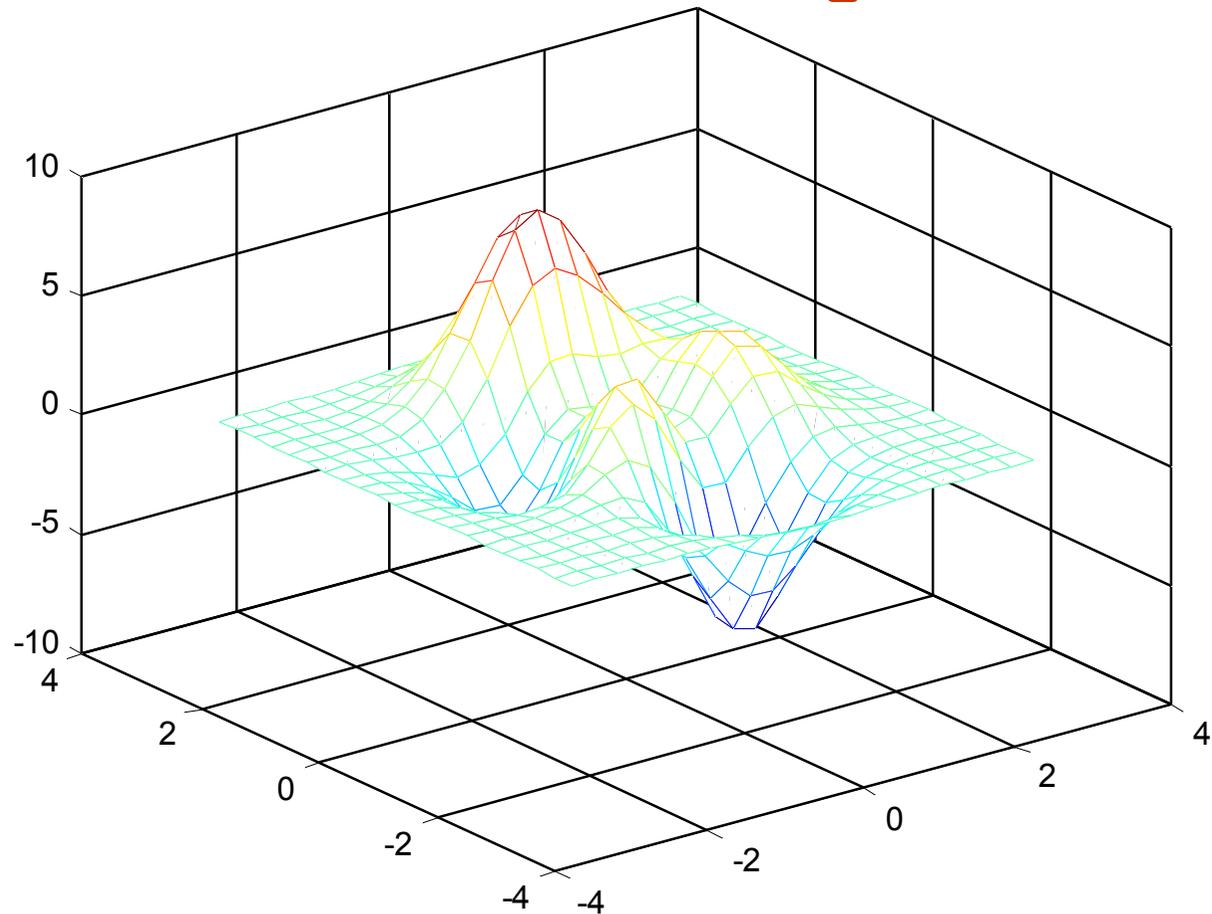
```
t = 0:pi/20:8*pi;  
x = sin(t);  
y = cos(t);  
z = 1.5*t;  
hp = plot3(x,y,z);  
set(hp,'linewidth',2,'color','b');  
grid on  
axis square
```

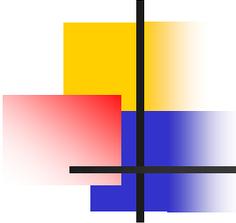




三维作图

- 空间曲面: $z = f(x, y)$





三维作图

- **mesh (X, Y, Z)**

- 网格生成函数: **meshgrid**

x, y 为给定的向量, **X, Y** 是网格划分后得到的网格矩阵

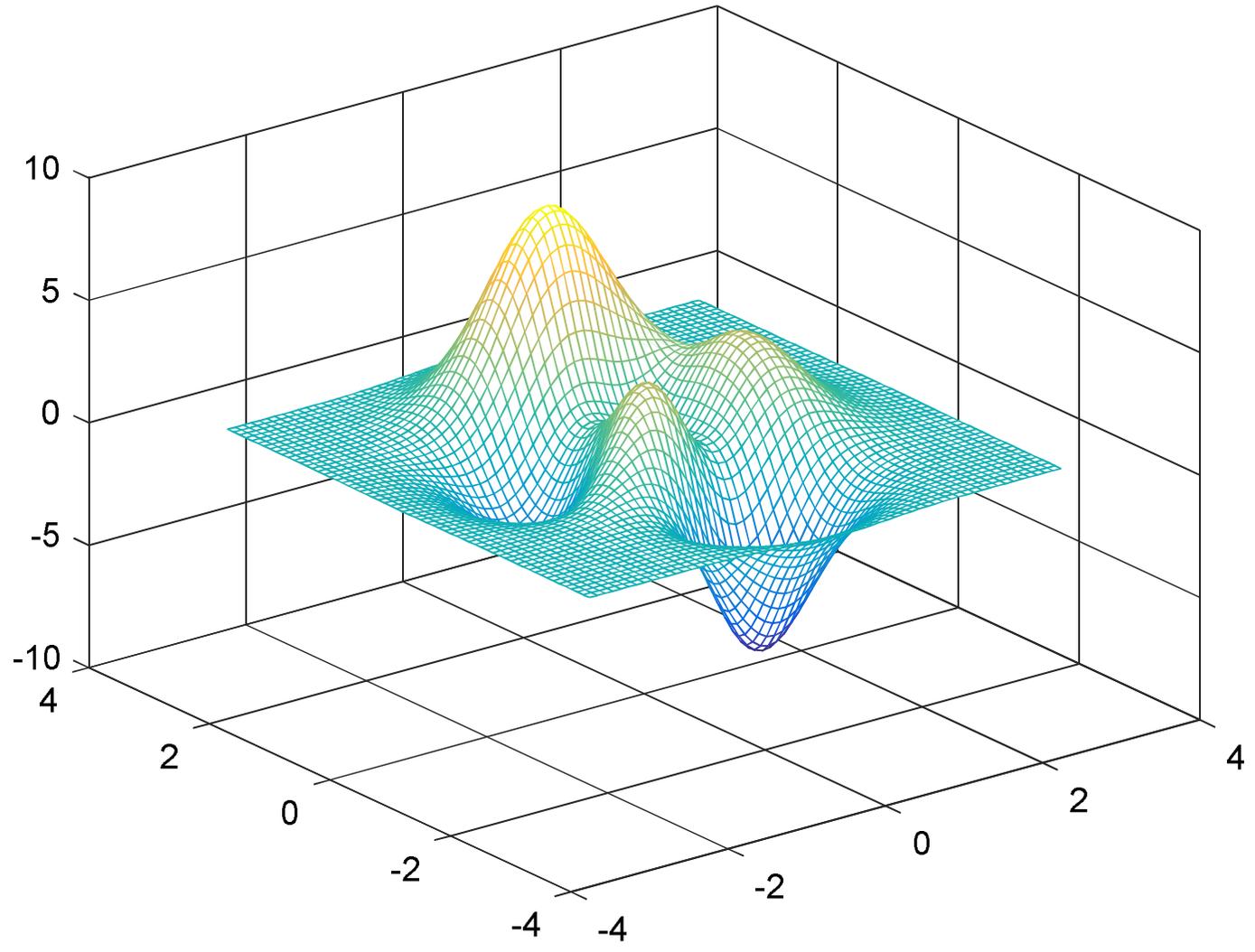
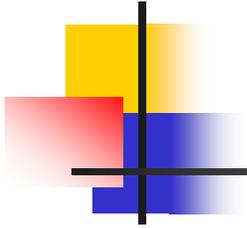
若 **x = y**, 则可简写为 **[X, Y] = meshgrid(x)**

例:

```
[X, Y] = meshgrid(-3 : 0.1 : 3);  
Z = peaks(X, Y);  
mesh(X, Y, Z)
```

>> **peaks** %matlab自带的测试函数

```
z = 3*(1-x).^2.*exp(-(x.^2) - (y+1).^2) ...  
- 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2) ...  
- 1/3*exp(-(x+1).^2 - y.^2)
```



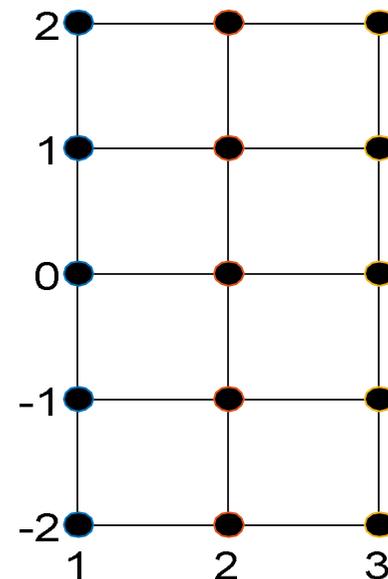
meshgrid 函数

```
gx = [1 2 3];  
gy = [-2:2];  
[x,y] =  
meshgrid(gx,gy)  
whos x y
```

```
x 5x3 120 double  
y 5x3 120 double
```

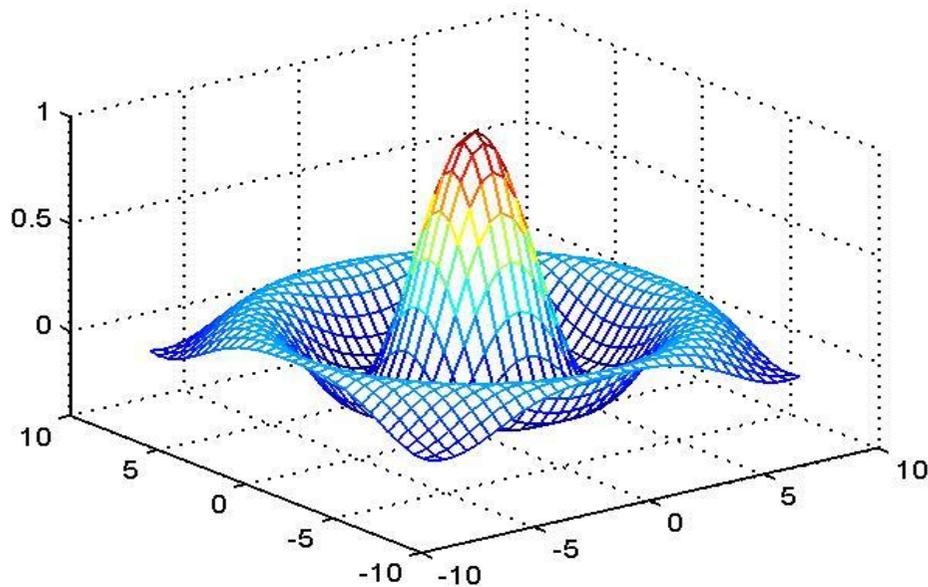
```
x = 1 2 3  
1 2 3  
1 2 3  
1 2 3  
1 2 3
```

```
y = -2 -2 -2  
-1 -1 -1  
0 0 0  
1 1 1  
2 2 2
```

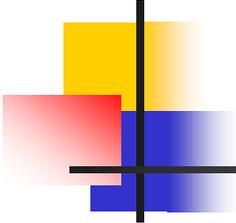


三维作图

■ 练习：“墨西哥帽子”

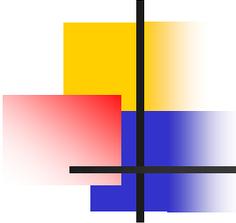


由函数 $z = \sin(r)/r$, 其中 $r = \sqrt{x^2 + y^2}$ 确定的曲面



三维作图

- 练习：画一个球体



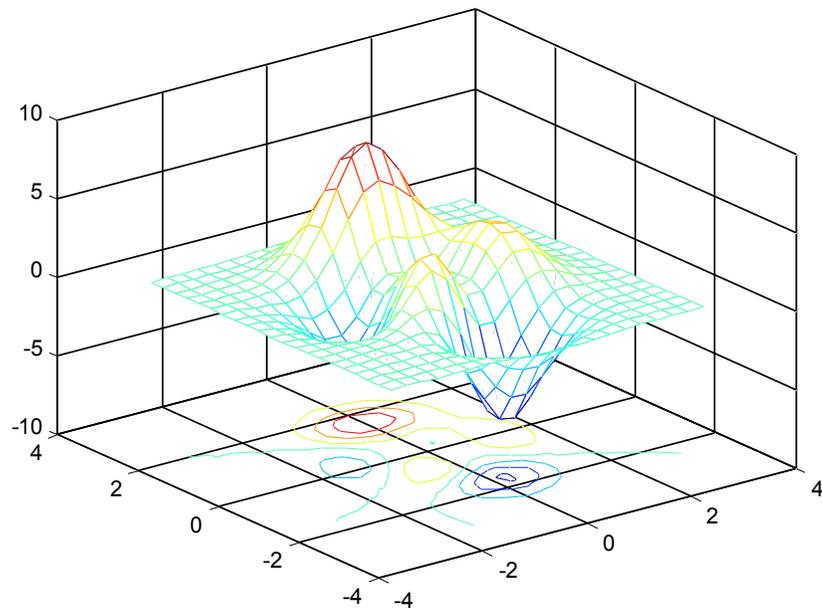
三维作图

- **meshc**: 绘制有登高线的空间曲面
- **meshz**: 绘制含0平面的空间曲面

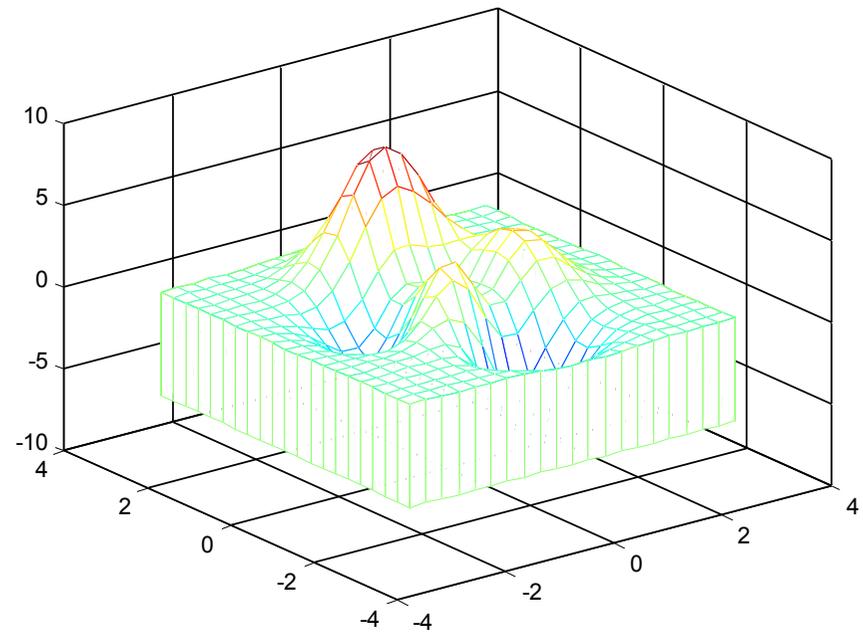
```
[X,Y] = meshgrid(-3:.4:3);  
Z = peaks(X,Y);  
meshc(X,Y,Z),title('meshc')  
figure  
meshz(X,Y,Z),title('meshz')
```

三维作图

meshc

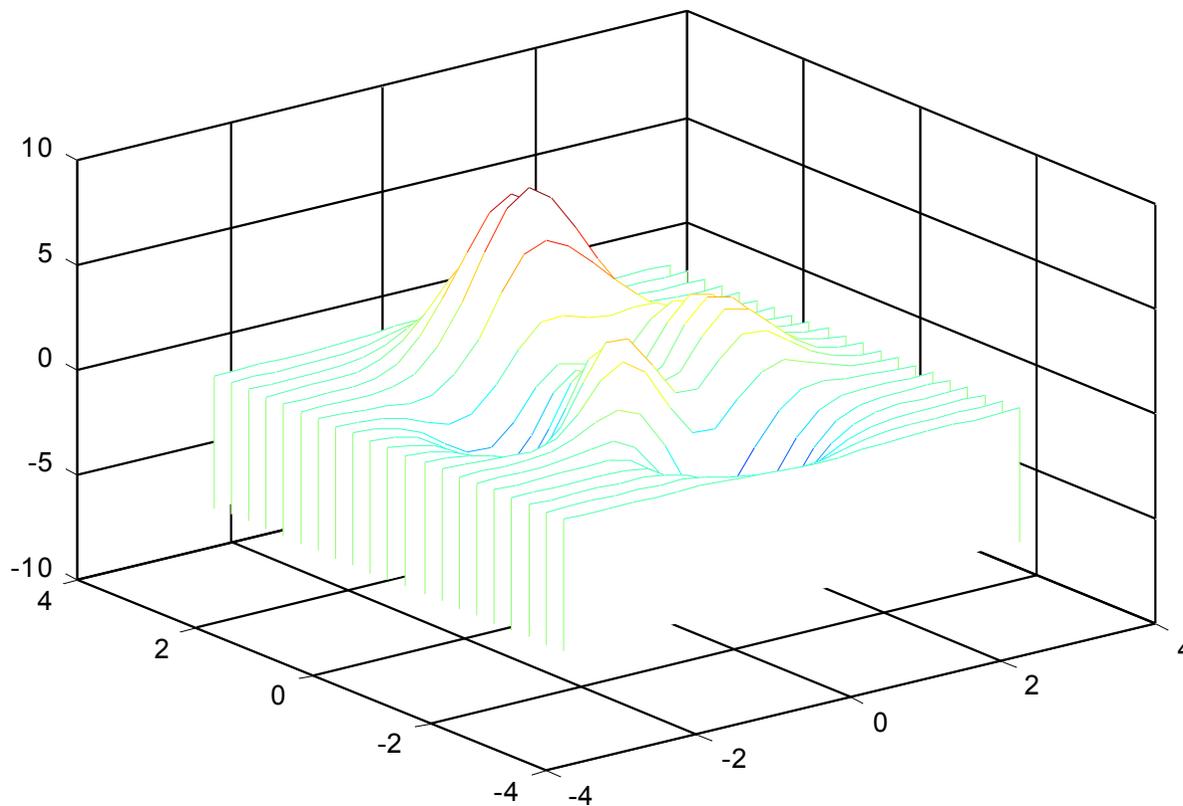


meshz



三维作图

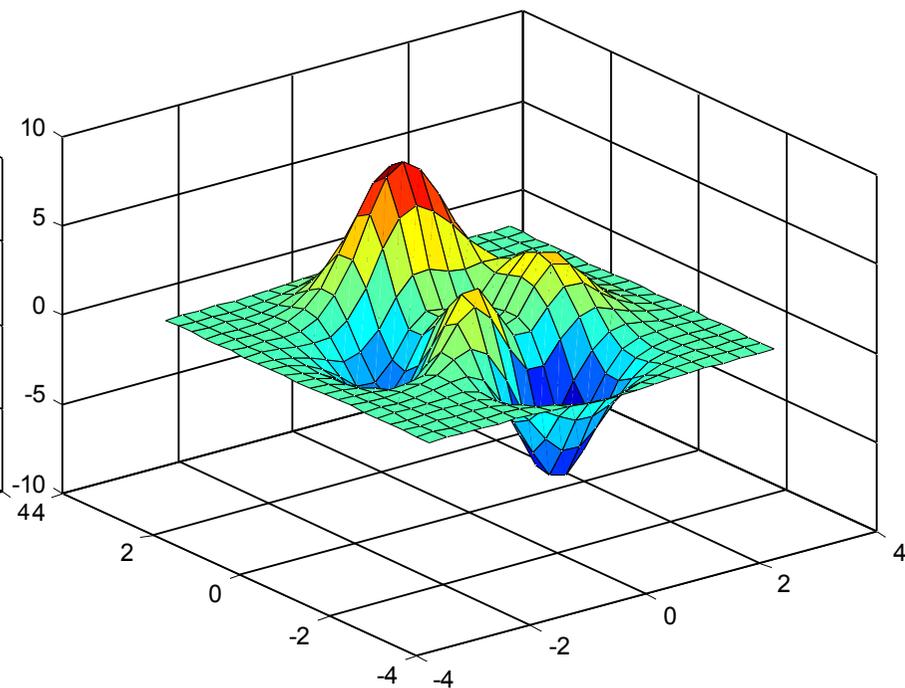
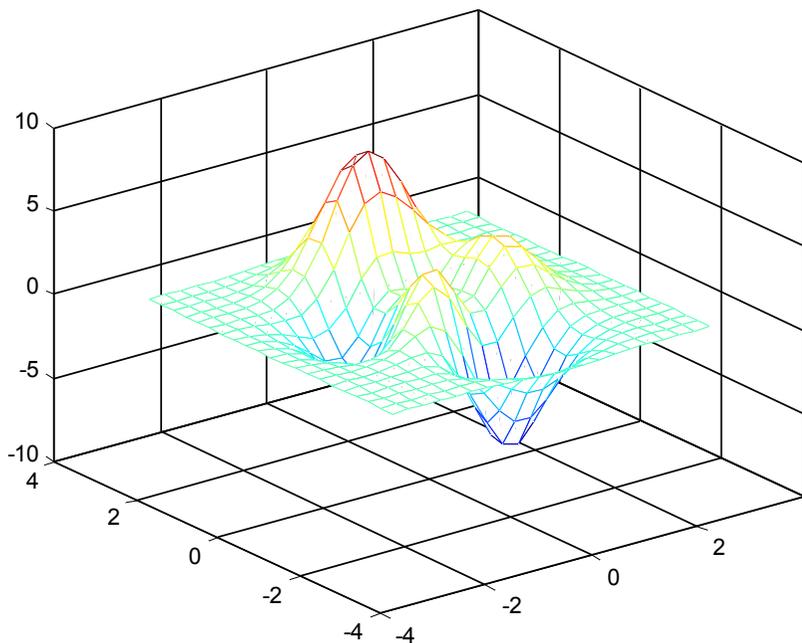
waterfall : 只绘制x轴方向的网格线



三维作图

- **surf(X,Y,Z)** 绘制由矩阵 X, Y, Z 所确定的曲面图, 参数含义同 mesh。

mesh 绘制网格图, **surf** 绘制着色的三维表面图

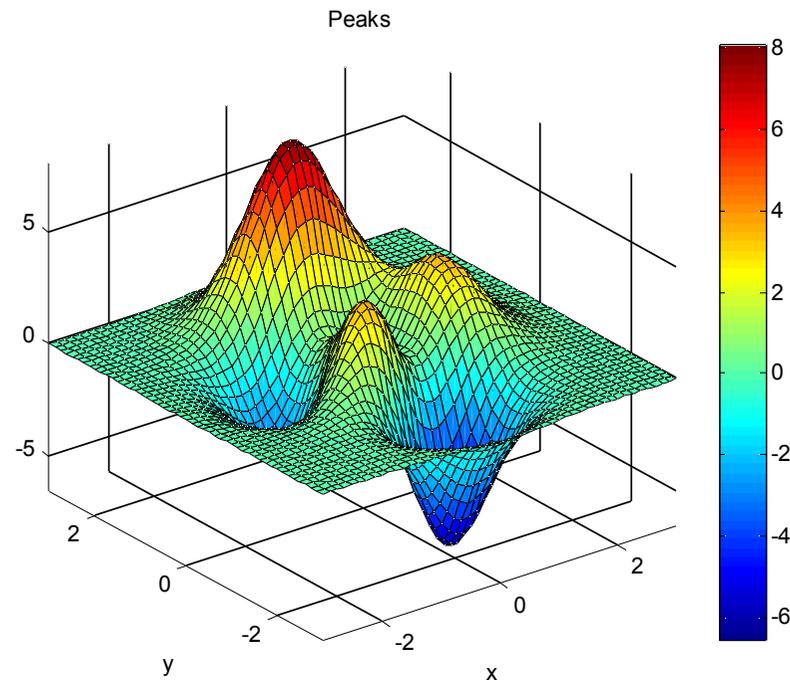


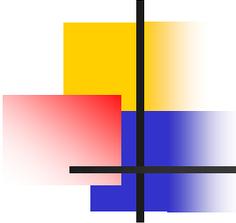
三维作图

- 曲面颜色的控制:
- **colorbar**: 显示 MATLAB 如何以不同颜色代表曲面的高度。

例:

```
peaks;  
colorbar;
```

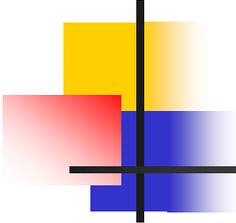




三维作图

- 曲面颜色的控制:

顏色	Red (紅色)	Green (綠色)	Blue (藍色)
black (黑)	0	0	0
white (白)	1	1	1
red (紅)	1	0	0
green (綠)	0	1	0
blue (藍)	0	0	1
yellow (黃)	1	1	0
magenta (鎳紫)	1	0	1
cyan (青藍)	0	1	1
gray (灰)	0.5	0.5	0.5
dark red (暗紅)	0.5	0	0
copper (銅色)	1	0.62	0.4
aquamarine (碧綠)	0.49	1	0.83



三维作图

- 颜色映射图 **colormap**

```
cm = colormap;  
size(cm)  
ans =  
      64      3
```

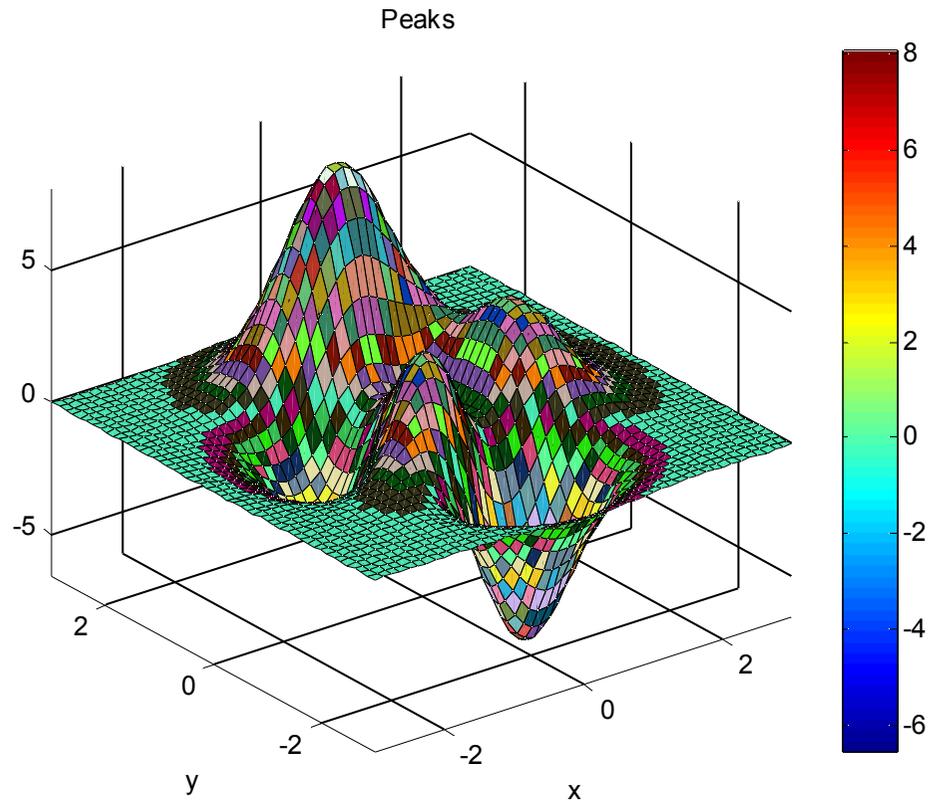
- **cm**是64*3的矩阵，每行代表**RGB**的成分，即一种颜色。
- **matlab**在画图时，把第一行的颜色对应曲面的最高点，最后一行颜色对应曲面最低点。其余高度颜色依照线性内插法给出。

三维作图

- 改变颜色映射图，可得到不同颜色的曲面。

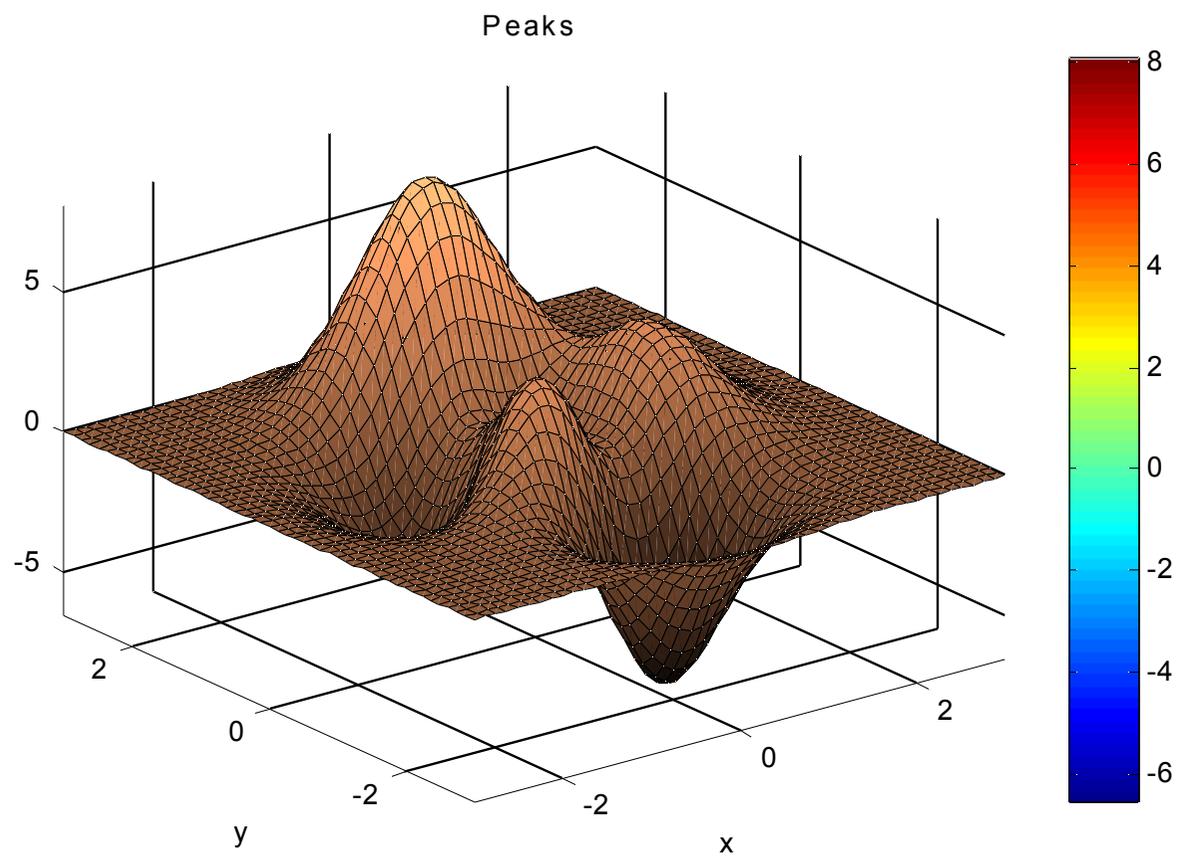
例：

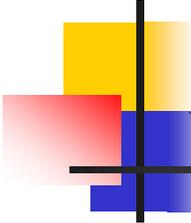
```
peaks ;  
colormap (rand (64 , 3) ) ;  
colorbar ;
```



三维作图

■ `colormap('copper')`

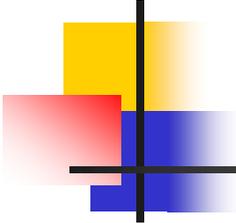




三维作图

■ matlab预设的颜色映射图

指令	说明
<code>colormap hsv</code>	HSV 的颜色对应表（预设值）
<code>colormap hot</code>	代表“热”的颜色对应表
<code>colormap cool</code>	代表“冷”的颜色对应表
<code>colormap summer</code>	代表“夏天”的颜色对应表
<code>colormap gray</code>	代表“灰阶”的颜色对应表
<code>colormap copper</code>	代表“铜色”的颜色对应表
<code>colormap autumn</code>	代表“秋天”的颜色对应表
<code>colormap winter</code>	代表“冬天”的颜色对应表
<code>colormap spring</code>	代表“春天”的颜色对应表
<code>colormap bone</code>	代表“X光片”的颜色对应表
<code>colormap pink</code>	代表“粉红”的颜色对应表
<code>colormap flag</code>	代表“旗帜”的颜色对应表



三维作图

- 以曲面梯度大小设定颜色:

```
[x,y,z] = peaks;  
surf(x,y,z,gradient(z));
```

- 以曲面曲率大小设定颜色:

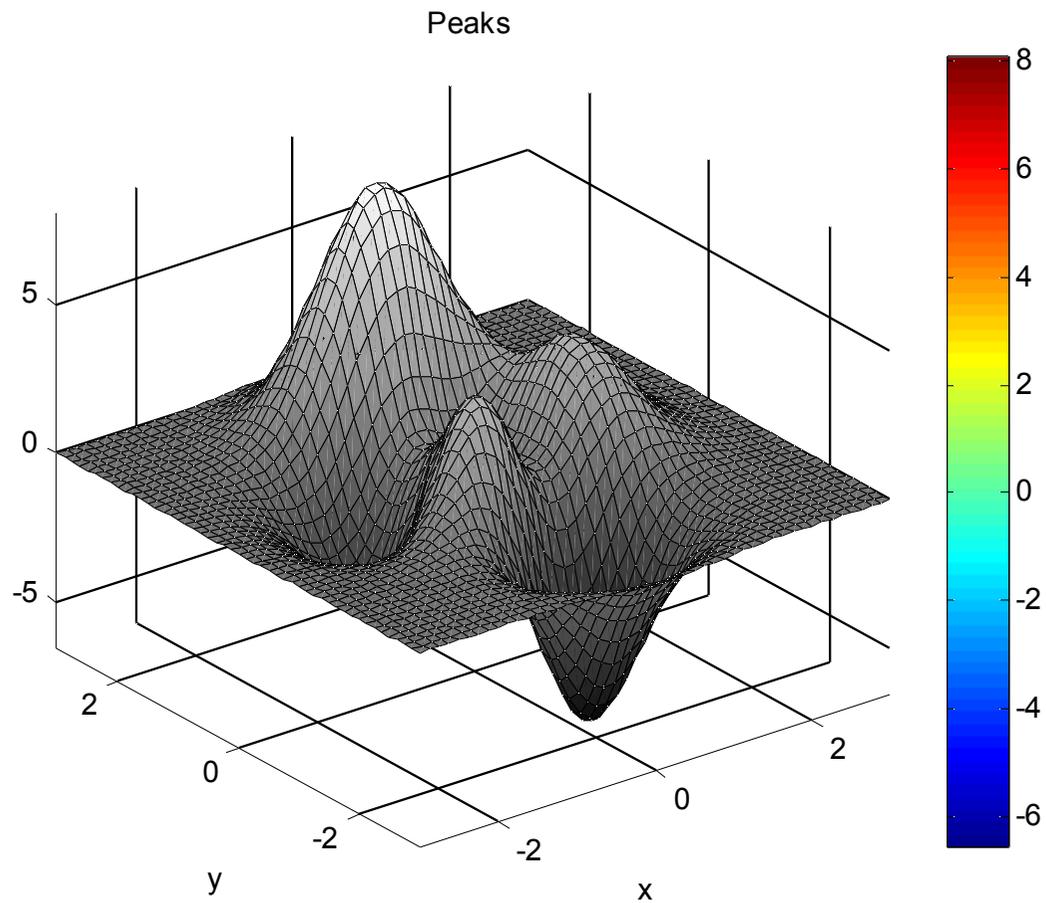
```
[x,y,z] = peaks;  
surf(x,y,z,del2(z));
```

- 使得表面颜色连续变化

```
surf(peaks)  
axis tight  
shading interp
```

三维作图

■ colormap gray



三维作图

- 改变图形的观察视角：
- 同一物体，从不同角度看，图形也不同，比如一个椭球体。

```
view(az,el);
```

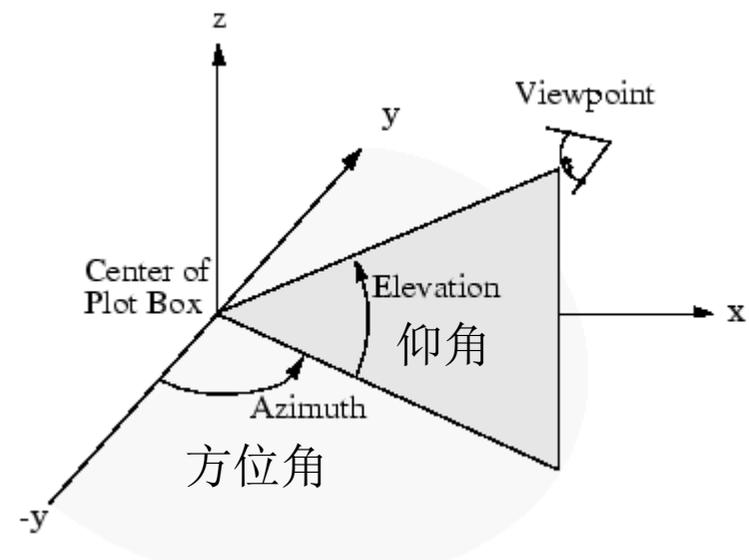
az: 方位角

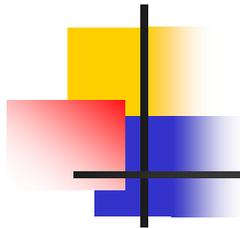
el: 仰角

```
view(x,y,z)
```

```
view(2); az=0, el=90
```

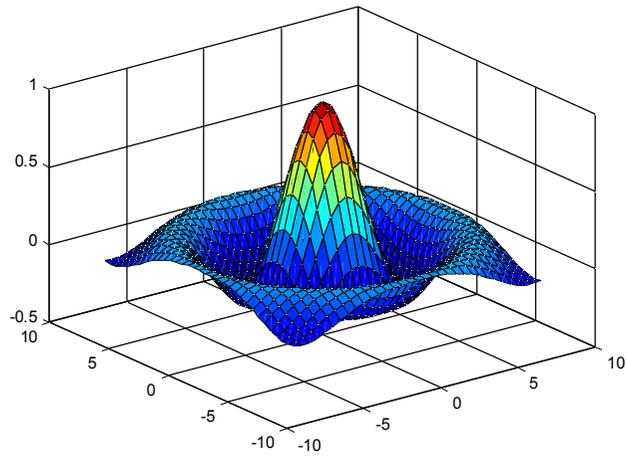
```
view(3); az=-37.5, el=30
```



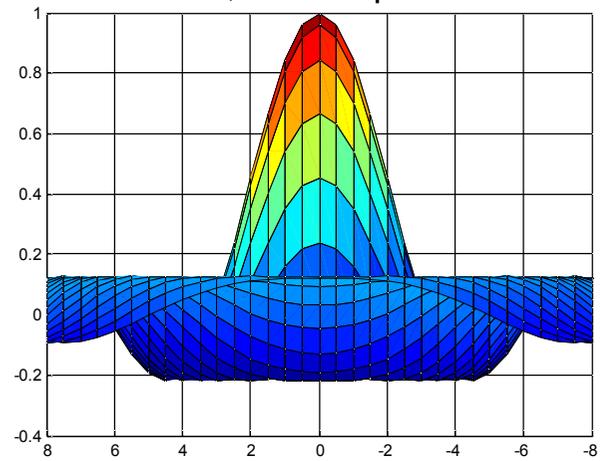


三维作图

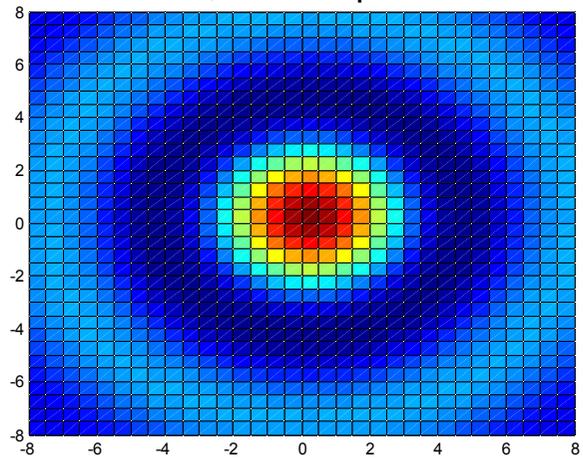
-37.5, 30

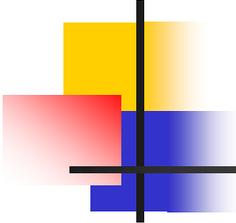


-90, 0 YZ plane



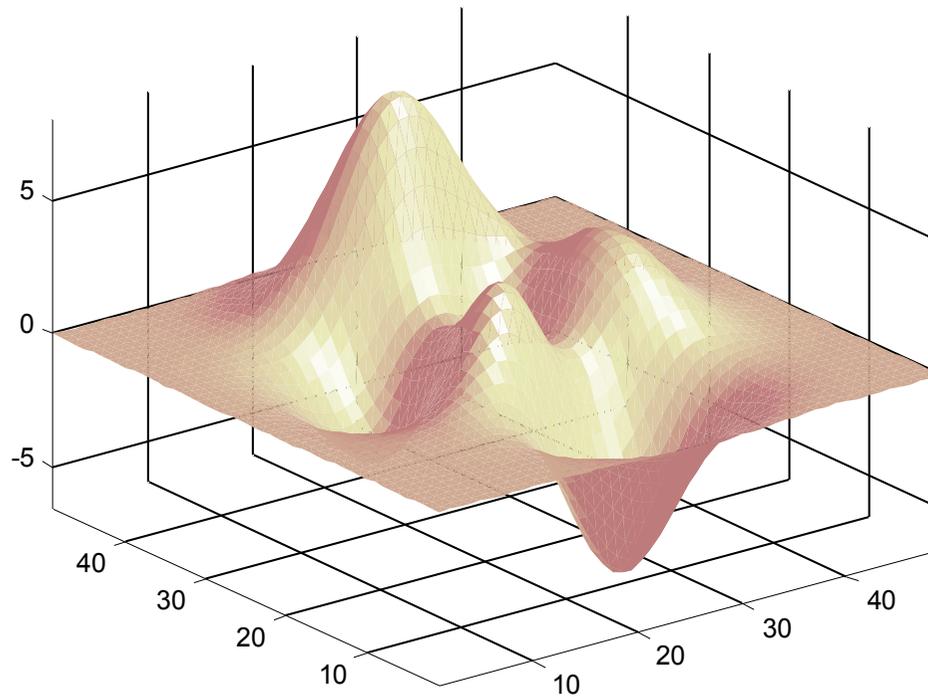
-90, 0 XY plane

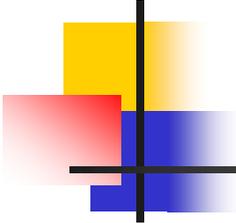




三维作图

- 试试这个效果:





三维作图

- 绘制等高线图:

contour: 平面上的等高线

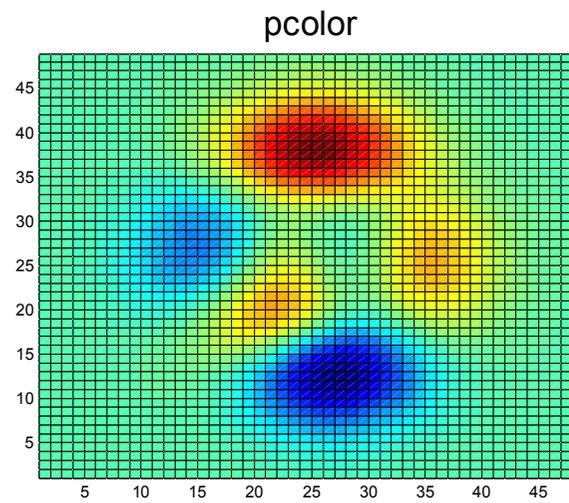
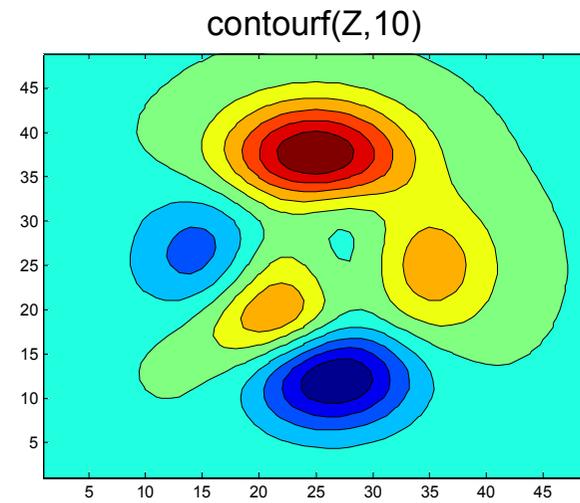
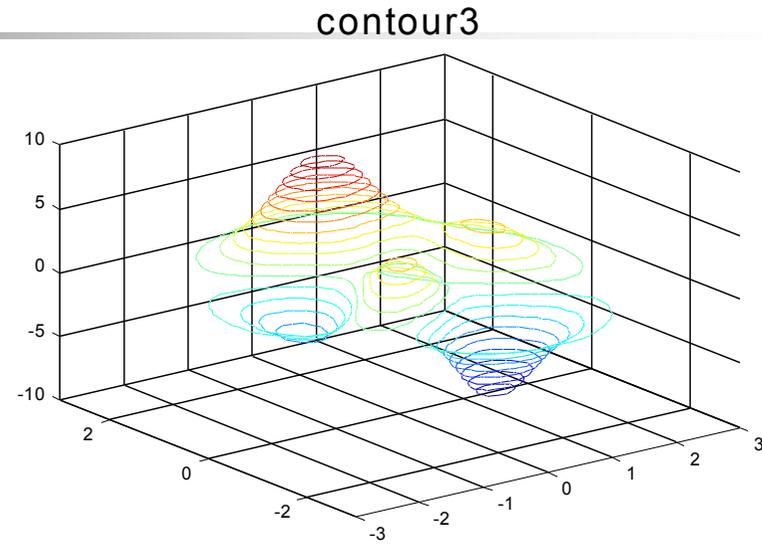
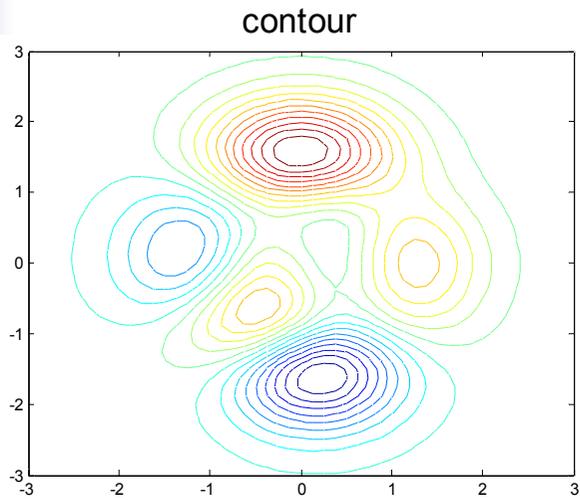
contour3: 空间中的等高线

pcolor : 在二维平面中以颜色表示曲面的高度

contourf :

clabel: 标注等高线高度

三维作图

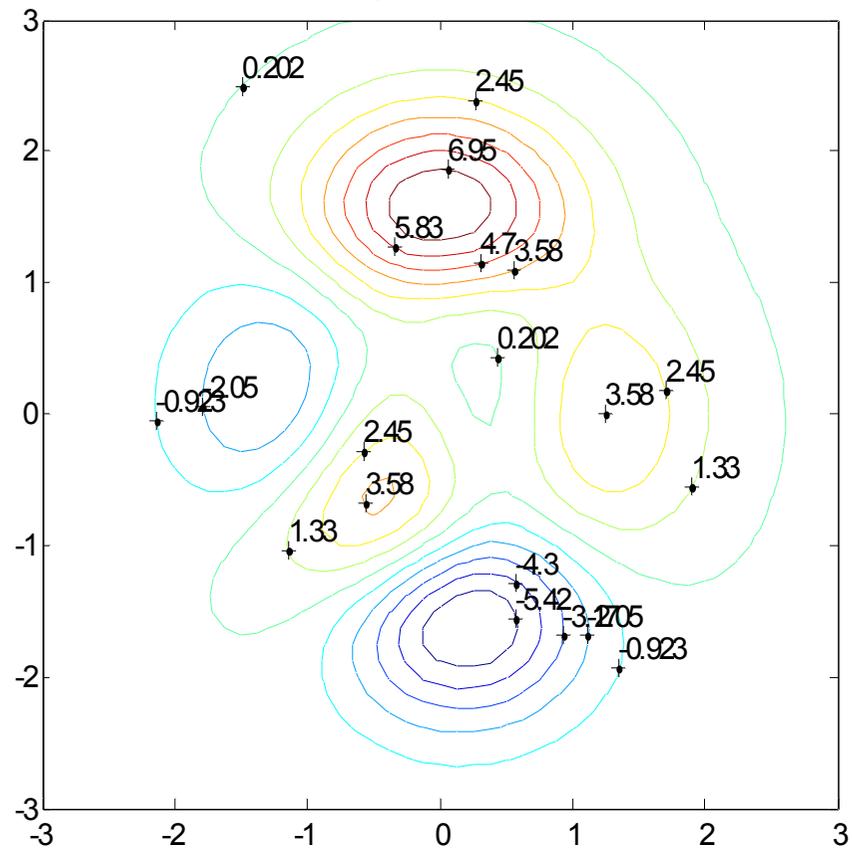


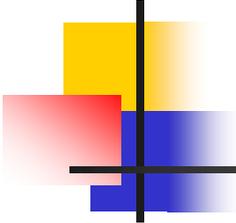
三维作图

- 给等高线加高度标注

```
[X,Y,Z] = peaks;  
C = contour(X,Y,Z,12);  
clabel(C)
```

Contour plot with labels





三维作图

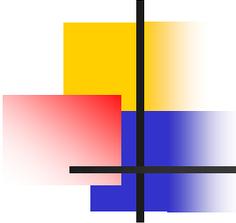
- $v = f(x, y, z)$ 的可视化:

比如：空间各点的温度（标量场）；

海洋中各处水的流速与方向（矢量场）。

- 常用方法：对三维体切片，在截面处用不同颜色表征在该点的函数值大小。

```
slice(X, Y, Z, V, sx, sy, sz)
```

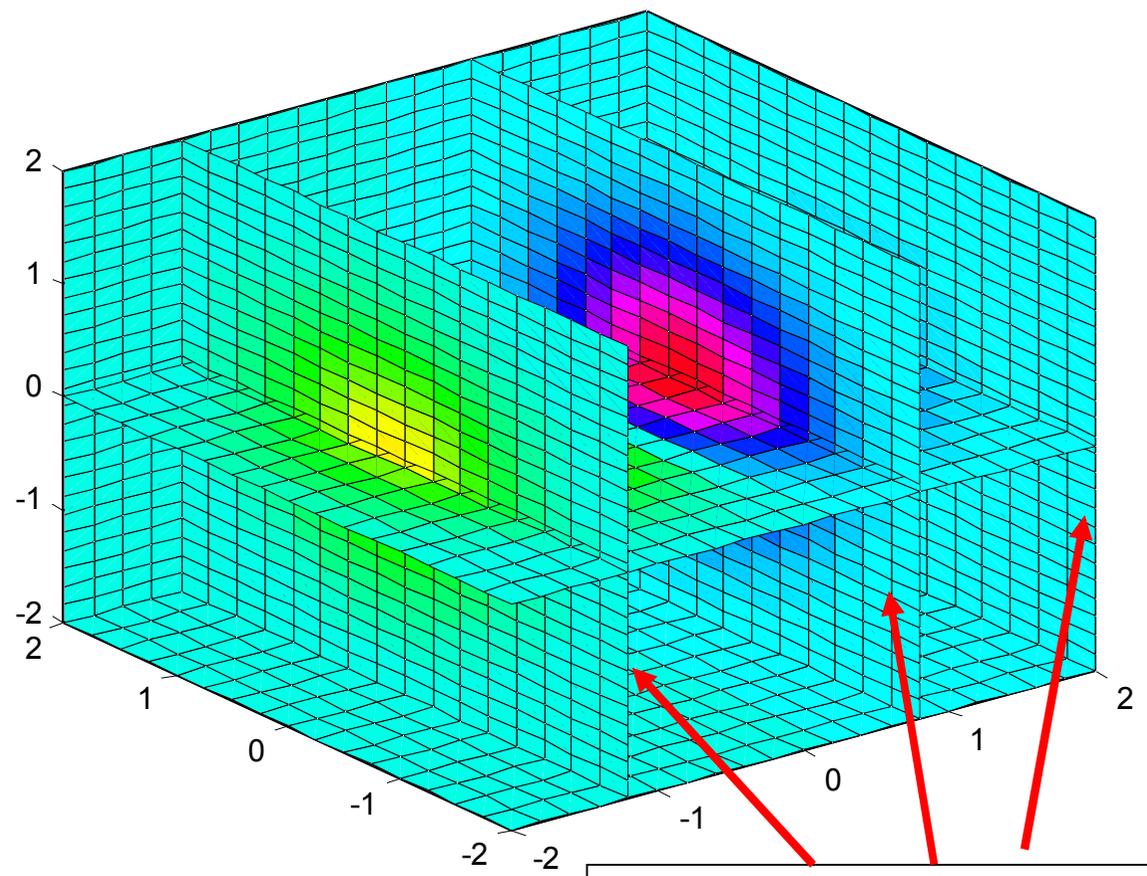


三维作图

$$v = x e^{(-x^2 - y^2 - z^2)}$$

```
[x,y,z] = meshgrid(-2:.2:2,-2:.25:2, ...  
-2:.16:2); % 产生三维网格  
v = x.*exp(-x.^2-y.^2-z.^2); % 待绘制的函数  
xslice = [-1.2,.8,2]; % x=-1.2平面...  
yslice = 2;  
zslice = [-2,0];  
slice(x,y,z,v,xslice,yslice,zslice)  
colormap hsv
```

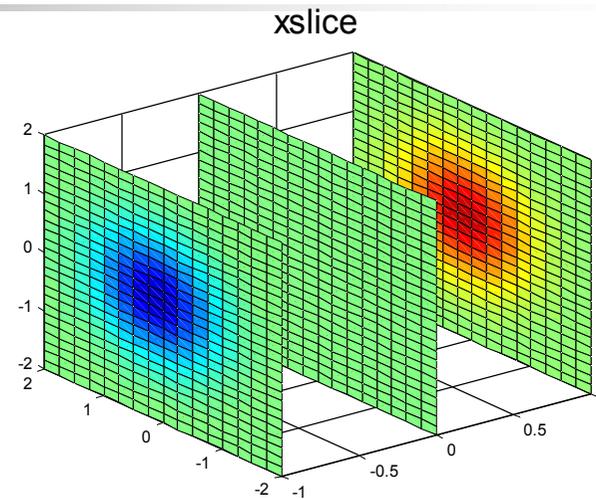
三维作图



`xslice=[-1.2 ,0.8 2]`

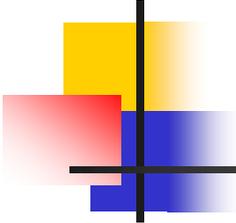
三维作图

```
slice(x,y,z,v,...  
[-1 0 1],[],[])
```



```
slice(x,y,z,v,...  
[],[-1 0 1],[])
```

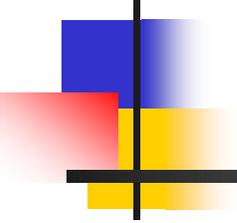




练习

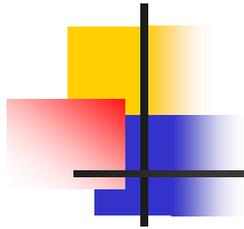
- 用 **ezmesh** 和 **ezsurf** 分别绘制一个圆环面，并将它们放在一个图形界面内，观察它们的三维内插不同之处。

$$\begin{cases} x = (R + r \cos u) \cos v \\ y = (R + r \cos u) \sin v \\ z = r \sin u \end{cases}$$

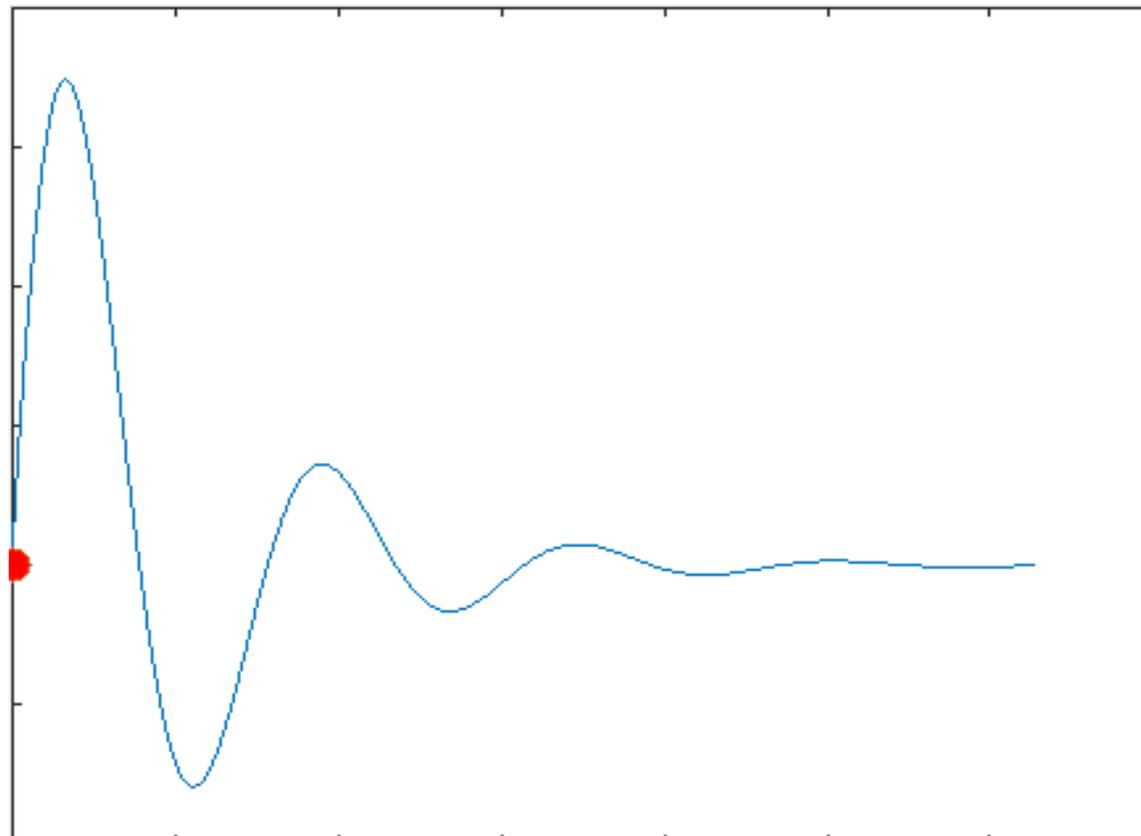


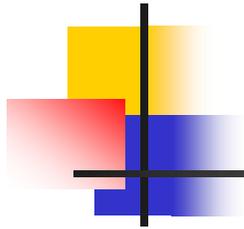
Part 3:

动态演示



- 一个点在曲线上移动:

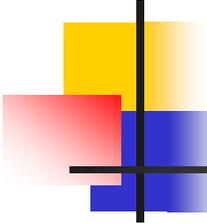




- 一个点在曲线上移动:

```
t=0:0.05:4*pi;  
N = length(t);  
y = sin(2*t).*exp(-t/2);  
h=plot(t , y);  
hold on
```

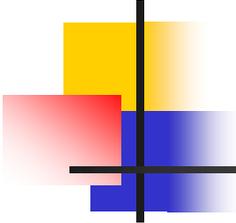
```
hp = plot(t(1),y(1),'marker','o',  
          'markersize',10,  
          'markerfacecolor','r');  
for k=2:N  
    set(hp,'xdata',t(k),'ydata',y(k));  
    %drawnow  
    pause(0.05)  
end
```



将动画存为GIF文件

■ `imwrite` 函数

```
F = getframe;  
im = frame2im(F);  
[I,map] = rgb2ind(im,256); %Gif,只能用256色  
    %写入 GIF89a 格式文件  
if k == 1;  
    imwrite(I,map,'test.gif','GIF','Loopcount',inf,'DelayTime',0.1);  
else  
imwrite(I,map, 'test.gif' , 'GIF' , 'WriteMode' , 'append' ,  
    'DelayTime' ,0.1) ;  
end
```



内摆线Hypocycloid

- 定义：一个动圆紧贴另一个圆的内部滚动时，动圆上一定点P的运动轨迹。

$$\begin{cases} x = (R - r)\cos\theta + r\cos\left(\frac{R - r}{r}\theta\right) \\ y = (R - r)\sin\theta - r\sin\left(\frac{R - r}{r}\theta\right) \end{cases}$$

内摆线

