

MATLAB 语言及应用

MATLAB Language and Its Applications

第三章 MATLAB 绘图

Ch.3 Plotting

刘世东

Shidong Liu

School of Physics and Physical Engineering
Qufu Normal University



June 29, 2020

本章目录

1. 二维绘图

1.1 2D 绘图 plot

1.2 一幅图中画多条曲线

1.3 子图

1.4 对数坐标

1.5 隐函数绘图

1.6 极坐标

2. 三维绘图

2.1 空间曲线

2.2 空间曲面

3. 其他绘图

3.1 长条图

3.2 面积图

3.3 扇形图

3.4 针头图

3.5 阶梯图

3.6 场图

4. 动画

MATLAB 绘图

MATLAB 拥有大量简单、灵活、易用的绘图命令,使得数学计算结果可以方便地、多样性地实现了可视化,这是其它语言所不能比拟的. 可以搜索: [Types of Matlab Plots](#) 查看绘图命令.

计算机无法画出真正连续的函数: 1) 点足够密集,看上去就连续了; 2) 相邻点用直线连接,看上去就连续了.

plot 绘图

plot 绘图的一般步骤

- ① 定义函数
- ② 函数图形的值范围
- ③ 调用 MATLAB `plot(x,y)`

```
x=-2:.001:2; %设置自变量范围
y=(sqrt(cos(x)).*cos(200*x)+sqrt(abs(x))-0.7).*(4-x.*x).^0.01; % 求对应x的y值. ...
    可以用自定义的函数
plot(x,y) % !surprise! %
```

- `plot(x,y)` 默认画线图.
- `plot(x,y,'-o')` 数据用圈 (Circle) 表示并用线连接.
- `plot(x,y,'r-o')` 数据用圈 (Circle) 表示并用红线连接.

线性/颜色/标志的命令如下:

线形, 颜色等命令

Line Style	Description
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

Color	Description
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

Marker	Description
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Pentagram
h	Hexagram

其他设置

- 1 `title('add title of the figure')` 加标题
- 2 `xlabel('x axis')`
`ylabel('y axis')` 设置 x, y 轴标注
- 3 `grid on` 加网格
- 4 `axis[x1,x2,y1,y2]` 同时设置 x, y 轴范围; 单独设置时用 `xlim, ylim`
- 5 `axis equal` 设置 x, y 轴同一尺度, 让圆看起来是个圆

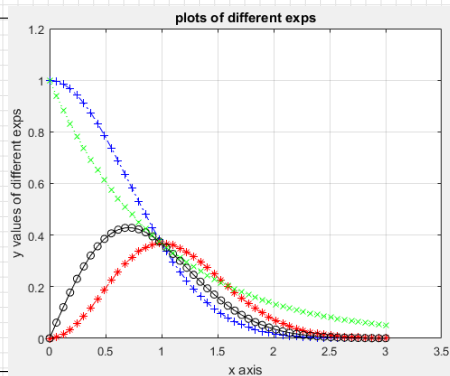
也可以用函数句柄形式或者 Name-Value Pair Arguments 形式进行设置, 更多设置参看`doc plot`帮助文档

也可以在 figure 窗口中用菜单设置.

绘制多条曲线

方法一: `plot(x1,y1,x2,y2)`

```
x=linspace(0,3,50); % Here put es in one line to ...
    save paper
e1=exp(-x.^2); e2=x.^2.*exp(-x.^2);
e3=x.*exp(-x.^2); e4=exp(-x);
plot(x,e1,'b-+',x,e2,'r--*',x,e3,'k-o',x,e4,'g:x')
% NOTE ABOVE!
title('plots of different exps')
xlabel('x axis')
ylabel('y values of different exps')
grid on
axis([0,3.5,0,1.2]) % save paper also
```



方法二: `plot(x,y)` 与 `hold on`: 画图时, 添加 `hold on` 命令即可多次使用 `plot`, 否则先前的图片会被清除. 例如:

```
x=linspace(0,3,50);  
e1=exp(-x.^2);  
e2=x.^2.*exp(-x.^2);  
plot(x,e1,'b-.+')  
plot(x,e2,'r--*')
```

上述代码的图形中只有 e2 图形, e1 图形被清除.

```
x=linspace(0,3,50);  
e1=exp(-x.^2);  
e2=x.^2.*exp(-x.^2);  
plot(x,e1,'b-.+')  
hold on % 第一个plot之前  
plot(x,e2,'r--*')  
hold off % 关闭 hold on的作用.
```

`hold on` 还有一个重要的作用是联合 `pause` 画动图.

绘制多条曲线 — 双坐标轴绘图

`yyaxis left/right` 命令画双坐标轴图 — 高版本 MATLAB 使用此命令

或者

`plotyy (X1, Y1, X2, Y2)`：以左、右不同的纵轴分别绘制 X1-Y1(左) 和 X2-Y2(右) 两条曲线。
但是高版本 MATLAB 不建议使用了。

高版本 MATLAB

```
x = linspace(0,10);
y = sin(3*x);
yyaxis left % NOTE HERE!!!
plot(x,y)
z = sin(3*x).*exp(0.5*x);
yyaxis right % NOTE HERE!!!
plot(x,z)
ylim([-150 150])
```

低版本 MATLAB

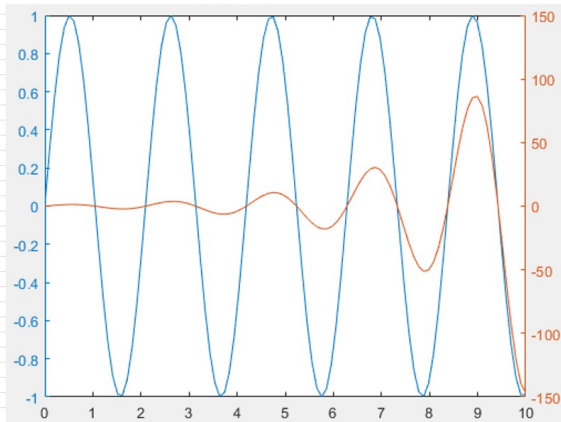
```
x = linspace(0,10);
y = sin(3*x);
z = sin(3*x).*exp(0.5*x);
plotyy(x,y,x,z)
```

一般, 新版本 MATLAB 都会向下兼容不是很久远的 MATLAB.

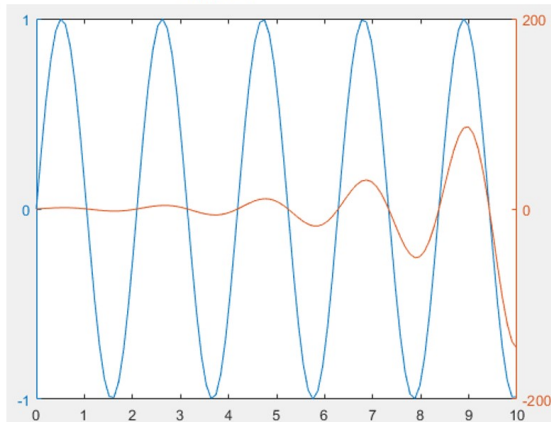
若使用了不建议使用的命令, 运行后, MATLAB 一般会给出提示...

`yyaxis` 模式设置相关参数更简单方便.

高版本MATLAB



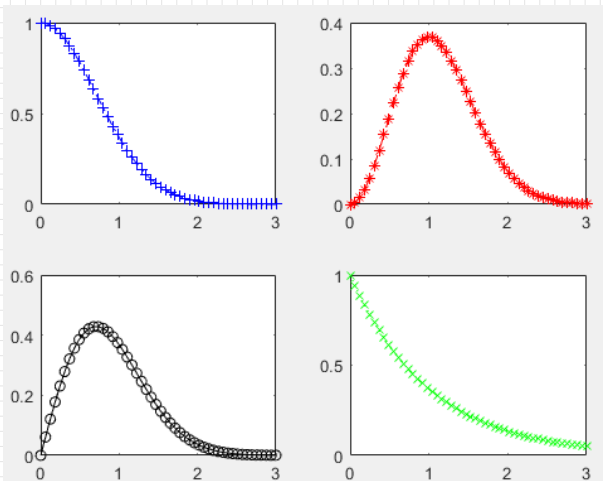
低版本MATLAB



一个画布中多副图-子图

`subplot(m,n,p)`: 将画布分成 $m \times n$ 个图片区域, 其中 m 为行数, n 为列数. p 表示将当前图放在第 p 个图片位置. 图片顺序的位置是 Z 字形顺序.

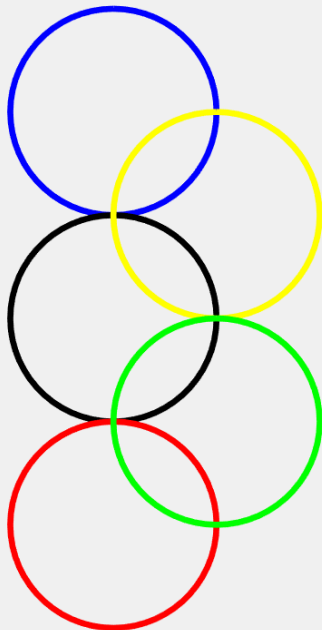
```
x=linspace(0,3,50);  
e1=exp(-x.^2); e2=x.^2.*exp(-x.^2);  
e3=x.*exp(-x.^2); e4=exp(-x);  
subplot(2,2,1), plot(x,e1,'b-.+')  
subplot(2,2,2), plot(x,e2,'r--*')  
subplot(2,2,3), plot(x,e3,'k-o')  
subplot(2,2,4), plot(x,e4,'g:x')
```



plot 函数可以画复数数组

```
t = 0:pi/30:2*pi;
z = cos(t)+1i*sin(t);
x0 = 2; y0 = 1i;
bluez = z + x0;
blackz = z;
redz = z - x0;
yellowz = z + (x0/2-y0);
greenz = z + (-x0/2-y0);
plot(bluez, 'b', 'Linewidth',5), hold on
plot(blackz, 'k', 'Linewidth',5)
plot(redz, 'r', 'Linewidth',5)
plot(yellowz, 'y', 'Linewidth',5)
plot(greenz, 'g', 'Linewidth',5), hold off
axis equal, axis off
text(-0.4,1.2, 'a↗ wuhuan')
```

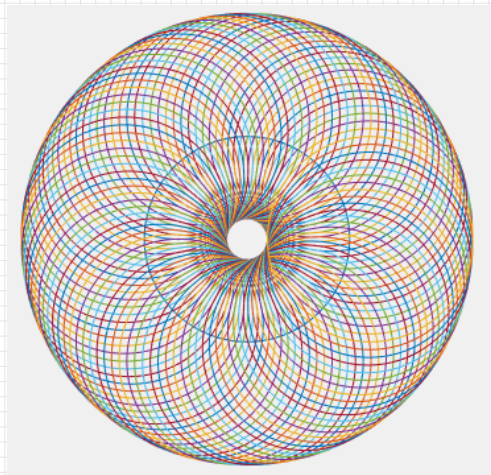
a↗ wuhuan



是时候展现一个好看的图了

```
t=0:.01:2*pi;  
z=cos(t)+1i*sin(t); %修改  
plot(z)  
axis equal  
tz=linspace(0,2*pi,50); %修改  
hold on  
R=1.2; % 修改  
for k=1:length(tz)  
    z1=R*cos(tz(k))+1i*sin(tz(k))*R;  
    plot(z+z1)  
end
```

呸 你是不是对好看有什么误会...
(手动白眼)



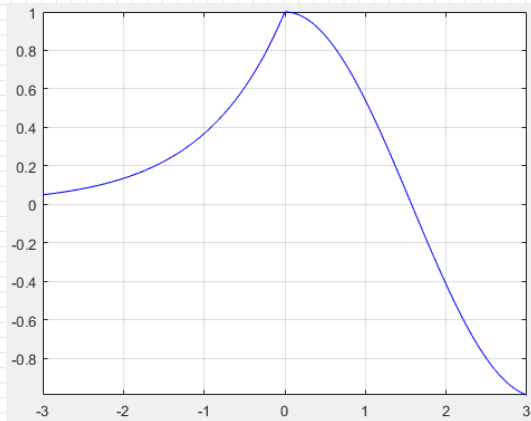
函数表达式绘图 fplot

`fplot(fun, [xmin, xmax])`: 函数 `fun` 的图像, 区间缺省时, 默认为 `[-5,5]`.

```
fplot(@(x) exp(x), [-3 0], 'b')  
hold on  
fplot(@(x) cos(x), [0 3], 'b')  
hold off  
grid on
```

fplot 优点: 自适应函数的变化趋势而取点, 从而能够在不可预知函数变化 (陡峭缓和) 趋势时保证取点疏密度使曲线准确.

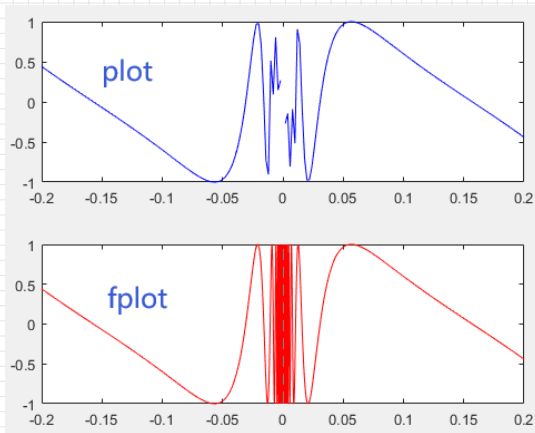
(see 下一页)



fplot 对变化陡峭函数的优势

函数图像 $f = \sin[\tan^{-1}(10x)]$

```
fun = @(x) sin(1./tan(10*x)); % 匿名函数
x = -.2:0.002:.2; y = fun(x); % 取点
subplot(2,1,1)
plot(x,y, 'b-') % 散点画图
subplot(2,1,2)
fplot(fun, [-.2, .2], 'r') % 函数表达式画图
```



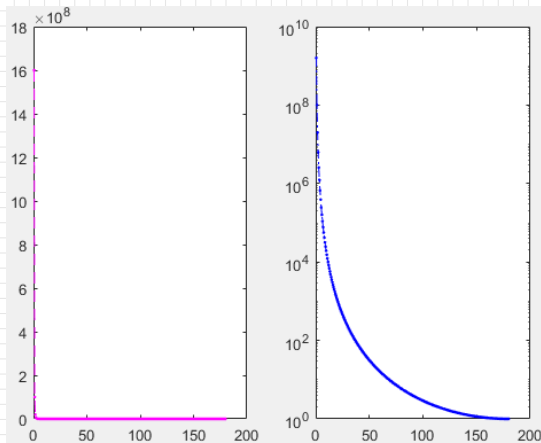
对数坐标

当 x 与 y 的数值变化差异很大或数值量级很大时, 往往会使用对数坐标.

- `semilogx(x,y)` x 轴对数;
- `semilogy(x,y)` y 轴对数;
- `loglog(x,y)` x,y 轴均对数.

$$\frac{d\sigma}{d\Omega} = \frac{1}{\sin^4(\theta/2)}$$

```
theta=0:0.01:pi;
dsdo=1./(sin(theta/2).^4);
subplot(1,2,1),
plot(rad2deg(theta),dsdo,'m-.-')
% 线性坐标
subplot(1,2,2),
semilogy(rad2deg(theta),dsdo,'b-.-')
% 对数y坐标
```



隐函数绘图 ezplot

显函数: $y = f(x)$ 或参数方程 $x = f_1(t), y = f_2(t)$ 的函数优先建议使用 plot 画图.

隐函数, 无法产生对应 x 的 y 数组, plot 无能为力. MATLAB 提供了隐函数的画图命令 `ezplot` (Easy-to-use function plot).

M 语句格式: `ezplot(fun)` 默认画 $x \in [-2\pi, 2\pi]$ 之间 $y = \text{fun}(x)$ 或 $x \in [-2\pi, 2\pi], y \in [-2\pi, 2\pi]$ 之间 $\text{fun}(x, y) = 0$ 的图像, 可以自定义区间.

这里的 fun 可以是函数句柄或者匿名函数或者字符串或者 M 函数文件.

```
ezplot('x^2-y^4') % recommend use ezplot('x.^2-y.^4')
%% or use
ezplot(@(x,y)x.^2-y.^4)
%% or use
fh = @(x,y)x.^2-y.^4
ezplot(fh)
```

`doc ezplot` 告之高版本的 MATLAB16a 不建议使用 `ezplot`, 所以建议不再使用, 无论 (你觉得) 多好用...

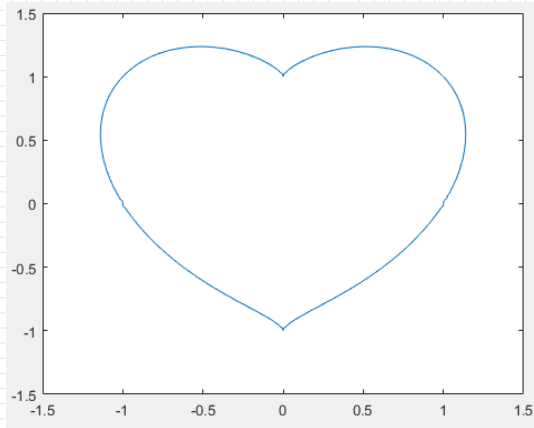
隐函数绘图 fimplicit

高版本 MATLAB2016b 画隐函数图

像 `fimplicit`, 用法类似 `ezplot`, 但是默认区间为 $[-5, 5]$ 且不建议再使用字符串的形式.

MATLAB: 简单的函数, 优先建议使用匿名函数, 复杂的使用 M 函数文件.

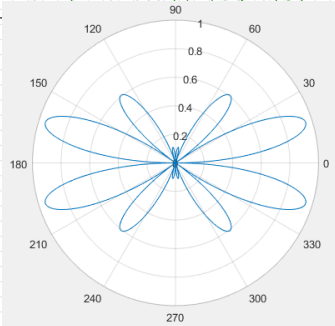
```
f = @(x,y) (x.^2+y.^2-1).^3-x.^2.*y.^3;  
fimplicit(f,[-1.5,1.5,-1.5,1.5])
```



极坐标 polarplot

`plot` 函数是在直角坐标系中画图, 极坐标画图时, 使用 `polarplot`¹ 相关参数设定与`plot`相似, 可以通过`doc polarplot`查看学习.

```
theta = 0:0.01:2*pi; rho = sin(5*theta).*cos(theta);  
polarplot(theta,rho) % try rho =sin(k*theta)+2+sqrt(5) k=1,2,...
```



符号函数可以用`ezpolar`, 注意它不能画隐函数.

¹低版本的 MATLAB 用 `polar`, `polarplot`是 2016a 引入的, 之后不建议使用`polar`.

空间曲线 plot3

`plot3(x,y,z)` 画空间曲线, 相关参数设置同 `plot`. (空间曲线方程一般由两个曲面方程联立得到; 或参数方程)

参数方程决定的空间曲线

```
t = 0:pi/50:10*pi;  
x= sin(t); y= cos(t); z= exp(-t/6);  
plot3(x,y,z, 'b'), xlabel('x'), ylabel('y'), zlabel('z')
```



空间曲面 surf

`surf(X,Y,Z)` 和 `surf(X,Y,Z,C)`

```
[X,Y] = meshgrid(-4:.3:4); %产生网格数据点
Z = (X.^2-Y.^2)/2;
C = gradient(Z); % Z的梯度
subplot(1,2,1), surf(X,Y,Z), xlabel('x'), ylabel('y'), zlabel('z')
colorbar %颜色显示Z的值, 同Z轴值一样
subplot(1,2,2), surf(X,Y,Z,C); xlabel('x'), ylabel('y'), zlabel('z')
colorbar % 颜色显示Z的梯度, 即C, Z轴为Z的值 (相当于四维图)
```

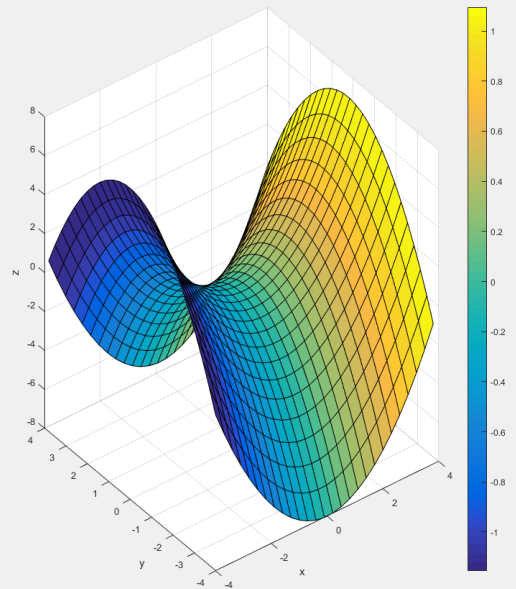
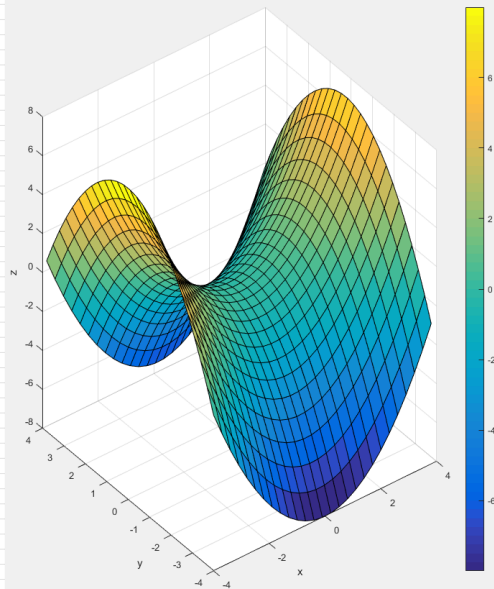
图形在下一页.

`colorbar` 显示 colormap, 默认显示在图形右边. colormap 默认情况下不显示. 在显示状态下 (如上) 时关闭可以使用命令 `colorbar('off')`

`surfc` 等于 `surf` 函数和 `contour` 函数. 画出空间曲面, 同时画出等高线 (XY 平面画等 Z 线), 例如

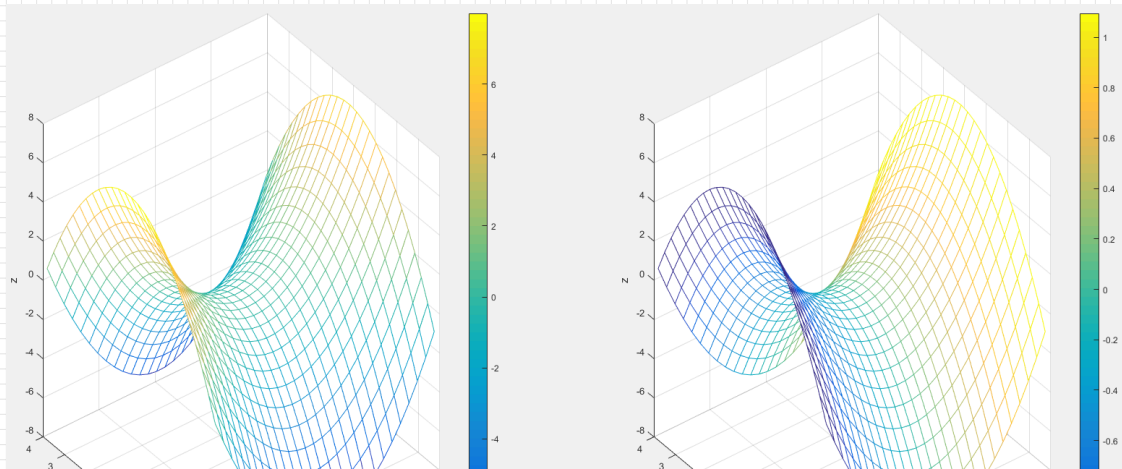
`surfc(X,Y,Z)`

`doc contour`.



空间曲面 mesh

`mesh`函数与 `surf` 函数的使用是一样的. 都是用矩形画曲面, 但是`surf`是以填充矩形, `mesh`是非填充矩形. 同理`meshc`.



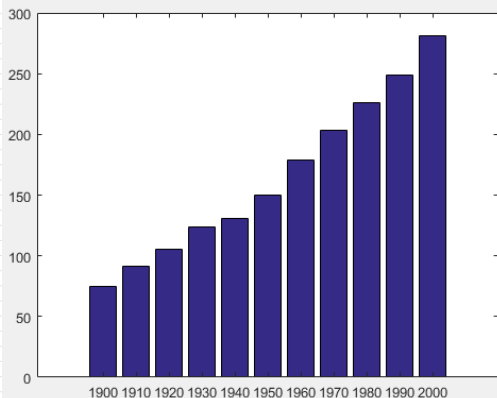
have a try

```
[x,y] = meshgrid(linspace(-2*pi,2*pi,60));  
z = real(atan(x+1i.*y));  
surf(x,y,z)
```


长条图 bar

`bar(x,y)` x 为横坐标, y 为纵坐标的长条图. 适合数据量少的画图方式.

```
x = 1900:10:2000;
y = [75 91 105 123.5 131 150 179 203 226 249 281.5];
bar(x,y)
```



- `bar(x,y,'r')`: 填充红色
- `bar(y,'FaceColor',[0 .5 ...
.5], 'EdgeColor',[0 .9 ...
.9], 'LineWidth',1.5)`

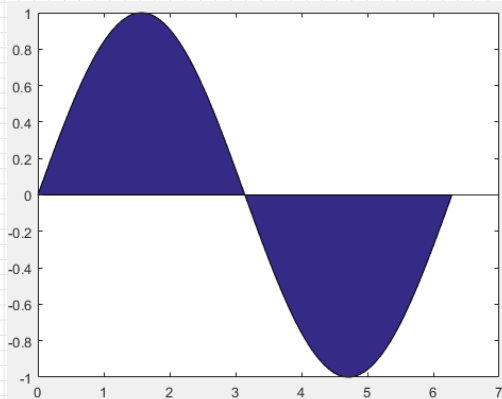
颜色和线宽

- `bar(x,y,0.4)`: bin 的相对宽度
- 更多 `doc bar` and see also `bar3`(三维立体), `bar3h`(水平长条图)

面积图 area

`area(x,y)` x 为横坐标, y 为纵坐标的填充面积图.

```
x=0:pi/100:2*pi;  
y=sin(x);  
area(x,y)
```

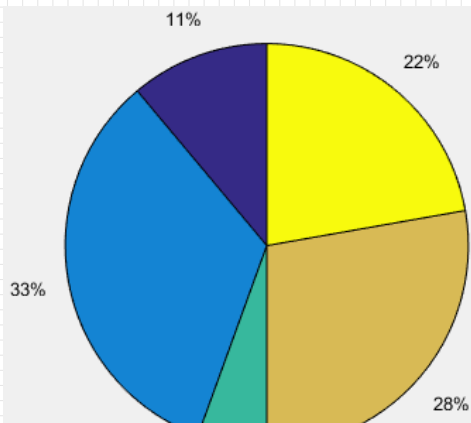


- 修改线形或者填充色...
- 更多 `doc area`

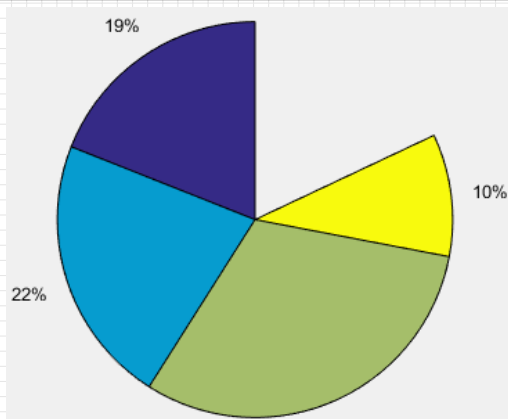
扇形图 pie

`pie(X)` 以扇形画 X 里的数据. 当 $\text{sum}(X) < 1$, 则画实际, 若大于 1 则画相对.

```
X = [1 3 0.5 2.5 2];  
pie(X)
```



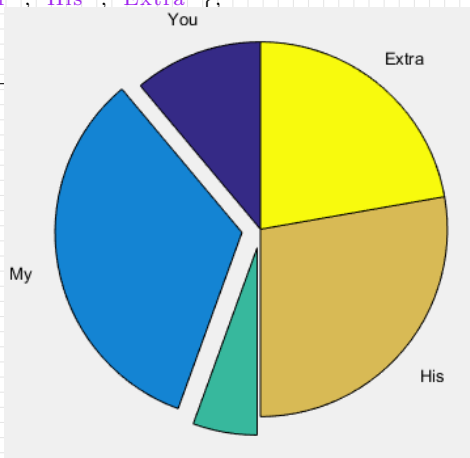
```
X = [0.19 0.22 0.31, 0.1];  
pie(X)
```



扇形图

扇形图可以突出部分扇形并命名部分扇形区

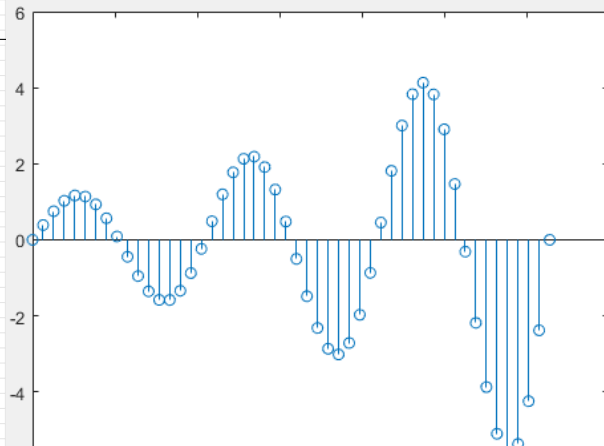
```
X = [1 3 0.5 2.5 2]; %更多设置参看 doc pie, see also --pie3画立体扇形图  
labels = {'You','My','Her','His','Extra'};  
explode = [0 1 1 0 0];  
pie(X,explode,labels)
```



针头图 stem

`stem(X,Y)` 针头图常用于工程科学.

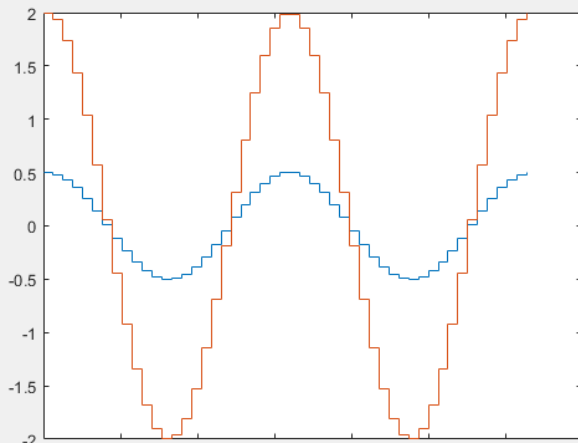
```
X = linspace(0,2*pi,50); %更多设置命令 doc stem --- stem3画立体针头图  
Y = exp(0.3*X).*sin(3*X);  
h = stem(X,Y);
```



阶线图 stairs

stairs(X,Y) 常用于统计分析

```
X = linspace(0,4*pi,50)'; %更多设置命令 doc stairs  
Y = [0.5*cos(X), 2*cos(X)];  
stairs(X,Y)
```

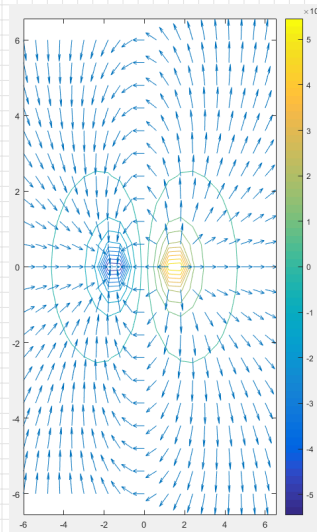


场图 quiver

展示电偶极子等势面和电场分布

```
q=2e-6;k=9e9;a=1.5;b=0; [X,Y]=meshgrid(-6:.6:6);
rp=sqrt((X-a).^2+(Y-b).^2); rm=sqrt((X+a).^2+(Y+b).^2);
V = q*k*(1./rp-1./rm);
[Ex,Ey]=gradient(-V);
AE=sqrt(Ex.^2+Ey.^2); Ex=Ex./AE; Ey=Ey./AE;
cv=linspace(min(min(V)),max(max(V)),49);
contour(X,Y,V,cv)
colorbar
hold on
quiver(X,Y,Ex,Ey,0.7)
plot(a,b,'wo',a,b,'w+'), plot(-a,-b,'wo',-a,-b,'w-')
hold off
```

`doc quiver` see also `quiver3`(三维场分布)



其他 (高级) 绘图命令或设置

参看 < 精通 Matlab R2011a > 张志涌等编著北京航空航天大学出版社, 第六章数据可视及探索. 略, 以我仅有的经验, 初级阶段用到最多的还是二维绘图.

万一不小心遇到特殊情况 (高级绘图), 可以通过百度查命令, 然后`doc`命令看用法和设置, 注意 see also. 一般都可以解决, 如果无法解决, 多半情况下 MATLAB 对此特殊情况的处理比较复杂或者鲜少有人会如你所想的那样处理...

放开那女孩, 你连 MATLAB 都不会, 你不配拥有她

```

clear % 以下程序来自知乎
% heart: https://www.zhihu.com/question/35654835/answer/64189073
[x,y,z]=meshgrid(linspace(-1.3,1.3));
val=(x.^2 + (9/4)*y.^2 + z.^2 - 1).^3 - x.^2.*z.^3 - (1/9)*y.^2.*z.^3;
p = patch(isosurface(x,y,z,val,0));
isonormals(x,y,z,val,p)
set(p, 'FaceColor', 'red', 'EdgeColor', 'none');
axis equal
view(-10,24)
colormap(hot)
axis off
camlight
rotate3d on
% flower:
% ...
https://www.zhihu.com/question/35057370/answer/87250282?group\_id=712348138785996800
[x,t] = meshgrid((0:24)/24,(0:.5:575)/575*17*pi-2*pi);
p = (pi/2)*exp(-t/(8*pi));

```

MATLAB 中的动画

MATLAB 中的动画必须依靠循环

MATLAB 中的动画实现方式有很多种, 下面只简单介绍几种我常用的实际上也是我只会的方法...:-)(-:...

点沿着曲线运动

```
t = linspace(0,15,200);
y = sin(5*t).*exp(-t/3);
plot(t,y,'r','LineWidth',1.5)
hold on
p = plot(t(1),y(1),'o','MarkerFaceColor','blue'); % 只画第一个点
hold off
axis manual % 冻结坐标轴刻度
for k = 2:length(t)
p.XData = t(k); % also set(p,'XData',t(k))
p.YData = y(k);
drawnow
%drawnow limitrate Can plot fast but may lost frames
end
```

Note: Starting in R2014b, you can use **dot notation** to set properties. If you are using an earlier release, use the set function instead, such as `set(p,'XData',x(k))`.

[doc Trace Marker Along Line AND Related Examples](#)

动画曲线

一. animatedline 函数形式

2014b 之后, MATLAB 引入了新的动画曲线函数. `h = animatedline;`

```
h = animatedline; % 产生一个空的动画线句柄.
axis([0,4*pi,-1,1]) % 定义坐标轴, 否则图片会变动
% 可以通过以下方式修改动画曲线属性, 更多参考 doc animatedline
h.LineStyle = '-.'; % 使用虚线, 缺省默认是实线
h.Color = 'r'; % 缺省默认为黑色
h.LineWidth = .75; % 缺省默认0.5
h.Marker = 'o'; % 缺省默认为 none

x = linspace(0,4*pi,1000); % 点数太少(10), 看不出动画效果或太快, 点数太多动画效果太慢
y = sin(5*x).*exp(-x/5);
for k = 1:length(x)
    addpoints(h,x(k),y(k));
    drawnow % 可以用drawnow limitrate 提速
end
```

二. hold on 与 pause/drawnow 联合形式

本质上而言, `animatedline` 使用的是不断增加数据点到画布上实现动画形式. 因此, 可以借助于 `hold on` 和 `pause` 命令进行画图.

仍然以前面的动画为例.

```
x = linspace(0,4*pi,1000);
y = sin(5*x).*exp(-x/5);
hold on
for k = 1:length(x)
plot(x(k),y(k),'r:o', 'LineWidth',2)% 因为每次只增加一个点, 故线形式在这里不起作用
axis([0,4*pi,-1,1])% 应该放在for循环里面, 否则plot范围会发生变化
pause(0.01)
end
hold off
```

使用 `hold on` 命令时, 建议均在合适位置关闭 `hold on`.

本人建议使用 `pause` 命令, 因为命令含义更明确, 且更改动画速度更方便.

三. 不断覆盖形式

所有的动画都可以使用消除, 覆盖进行实现.

```
x = linspace(0,4*pi,1000);  
y = sin(5*x).*exp(-x/5);  
for k = 1:length(x)  
    plot(x(1:k),y(1:k),'r:o', 'LineWidth',2)  
    axis([0,4*pi,-1,1])  
    pause(0.01)  
end
```

接上一个动画曲线—谁还不曾心动

```
h = animatedline;  
axis([-2 2 -2 1.5])  
  
h.Color = 'r';  
h.LineWidth = 2; % 线粗一点简直完美  
  
x = linspace(-2,2,1000);  
y=real((sqrt(cos(x)).*cos(200*x)+sqrt(abs(x))-0.7).*(4-x.*x).^0.01);% 要取实部  
for k = 1:length(x)  
    addpoints(h,x(k),y(k)); % addpoints 不支持复数，不会自动忽略虚部，故 y 取实部  
drawnow  
end
```

椭圆演示

椭圆方程 $x^2/a^2 + y^2/b^2 = 1$, 其参数方程为 $x = a \cos t$, $y = b \sin t$, 焦距为 $d = 2\sqrt{|a^2 - b^2|}$

动画显示椭圆的由来.

四: movie 函数

类似 (远古时代, 其实好多优秀的动画片) 放电影, 一帧一帧的图片放映

%movie函数制作向右运动的正弦波

```
t=0:pi/20:4*pi;
```

```
x=0:0.05*pi:4*pi;
```

```
m=moviein(40);%用moviein创建40列的矩阵
```

```
for i=1:40
```

```
y=sin(x-t(i));%将函数sin(x)往右移动t(i)
```

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

```
axis([0,12,-1,1]);
```

```
m(i)=getframe;%用getframe函数把制作好的第i帧画面以列的形式保存在矩阵m中
```

```
end
```

```
movie(m,30);
```

更多详细用法参考 `doc movie`.

致谢

本课程参考资料包括但不限于 (排名不分先后):

- 1 Matlab 帮助文档
- 2 Experiments with Matlab, C.Moler
- 3 Matlab 高效编程技巧与应用: 25 个案例分析, 吴鹏 (rocwoods)
- 4 精通 Matlab R2011a, 张志勇
- 5 Matlab 2014a 完全自学一本通, 刘浩/韩晶
- 6 Elementary Mathematical and Computational Tools for Electrical and Computer Engineers using Matlab, Jamal T.Manassah
- 7 Physical Modeling in Matlab, Allen B.Downey
- 8 一些网络资源 (知乎, Matlab 中文论坛等)
- 9 ...