

## 第 6 讲 绘制二维图

(第 5 章 MATLAB 绘图)

目的:

1. 掌握绘制二维图形的方法。
2. 掌握绘制图形的辅助操作。

---

一、最常用的绘制二维图形的函数：plot 函数

(一) plot 函数是最常用的绘制图形的函数，其调用格式为：

**plot(x,y,name1,value1,name2,value2···)**

**plot 函数采用的是描点法做函数的图形。**

格式中的 x,y 是点的 x 坐标与 y 坐标，x 与 y 必须是同维向量，这样才能一一对应匹配后产生点。例如，如果 x=[1,2,3],y=[0, 3,1]则 plot 将描绘(1,0),(2,3),(3,1)这三个点。

格式中的 name 和 value 需成对设置，name 是描绘时的属性名称，value 是对应的属性的值。例如 plot (x,y,'color','r') 意思是将描绘的颜色设置成红色。详见书上 P107 页。

注：属性名称需用单引号括起来，属性值如果是数字，就不需要用单引号，如果不是数字，需要用单引号括起来。例如 plot (x, y, 'linestyle', '--', 'color', 'b', 'linewidth', 1.2) 表示点与点之间的连接线（属性名为 linestyle）设置为虚线（即属性值为--），颜色属性的值设置成蓝色，连接线的宽度设置成 1.2 宽。在这里 1.2 是数字，所以没有用单引号括起来。

如果不设置属性名称和属性值，只输入 plot(x,y)则 matlab 将默认使用蓝色、连接线默认使用实线、线宽默认使用 0.5。例如 plot(1:5,2:6)。

(二)、关于描点时 x 与 y 对应生成点的说明

**1、plot(x,y):当 x 与 y 是向量时，以 x 与 y 匹配产生的点描点作图,不需要同为行或者同为列，但维度需要一致。（最基本也是最重要的）**

**例 x=[1,2,3]; y=[3,5,7]'; plot(x,y); 这里 x 是行，y 是列，但维度一致，所以仍可以匹配产生点。**

2、plot(x,y):当 x 与 y 是矩阵时，可以同时绘制多条曲线（不常用，不推荐使用，重点掌握上面的 1，本身也可以用 1 的方法多绘制几次来替代，但可以了解下）

产生描点的原则是：

(1) 当  $x$  与  $y$  中有一个为向量，另一个为矩阵时，比如  $x$  是向量， $y$  是矩阵，则以  $x$  的维度去匹配  $y$  的行或者列，优先匹配列。

(2) 当  $x$  与  $y$  都是矩阵时，检查  $x$  的列向量的维度和  $y$  矩阵列向量的维度是否一致，如果一致，相对应匹配点作图；但要注意，此时  $x$  的列数需要等于  $y$  矩阵的列数，否则，就会有列找不到对应列，从而导致系统报错。

(3) 前后两矩阵优先进行列配对，如果列无法配对，则可以前行配后列或者前列配后行，只要维度相同。

注：plot( $x$ )是以向量  $x$  中元素的位标当横坐标， $x$  的元素当纵坐标作图。

例题 1：  $x=[2, 3, 6]$ ; plot( $x$ ); 等同于 plot([1,2,3],[2,3,6])

### (三)关于线型颜色和数据点显示的说明

plot( $x,y$ , '属性 1 名称', '属性 1 值', '属性 2 名称', '属性 2 值', 依次类推)

例如 plot( $x,y$ , 'linestyle', '--', 'color', 'r', 'marker', 'o')

意思是用点与点之间用双划线连接，颜色是红色，点的标记形状用  $o$  表示；如果不指定 linestyle

那么系统默认用实线连接，实线连接等同于 'linestyle', '-'。具体 linestyle 的值请查帮助文件。

如果不想用线连接点只想描点，那么使用 'linestyle', 'none'

系统允许缺省 'linestyle'、'color'、'marker' 这三个属性名称，比如

plot( $x,y$ , 'linestyle', '--', 'color', 'r', 'marker', 'o') 可简写为：

plot( $x,y$ , '--ro') 或者 plot( $x,y$ , 'r--o') 或者 plot( $x,y$ , 'or--') 属性值的顺序无所谓。

除了上述三个常用属性外，还有如下属性

'linewidth', 1.5 意思是线宽 1.5

'markersize', 6 意思是点的标记大小为 6。**注意：这两属性的值是数据标量，前后不加单引号。**

'markeredgecolor', 'g' 意思是点标记的边缘颜色是绿色。

'markerfacecolor', 'b' 意思是点标记的内部填充是蓝色。

### 练习 1：绘制一条连续曲线

设  $y = \left[ 0.5 + \frac{3 \sin x}{1 + x^2} \right] \cos x$ ，在  $x=0 \sim 2\pi$  区间取 101 点，绘制函数的曲线。

#### 给出程序和图形：

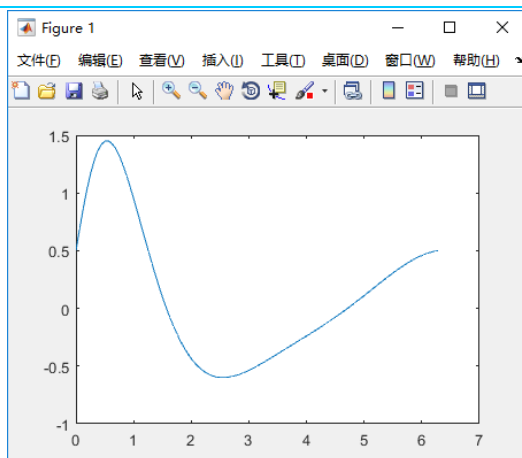
```
x=linspace(0,2*pi,101);
```

%这里使用了linspace函数来产生向量，其作用是在 $[0, 2\pi]$ 上等分的产生包括端点0与 $2\pi$ 在内

%的101个点，此时 $[0, 2\pi]$ 区间被分为了100等份。

```
y=(0.5+3*sin(x)./(1+x.^2)).*cos(x); %注意，这里要用点运算
```

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



**【提示】** 注意使用/和\*。

### 练习 2：在同一坐标系下用不同的颜色和线型绘制 3 条曲线

已知  $y_1 = x^2$ ， $y_2 = \cos(2x)$ ， $y_3 = y_1 * y_2$ ，其中  $x$  可取  $0 \sim 2\pi$ ，取 101 个点。

#### 给出程序和图形

```
x=linspace(0,2*pi,101);
```

```
y1=x.^2; y2=cos(2*x); y3=y1.*y2;
```

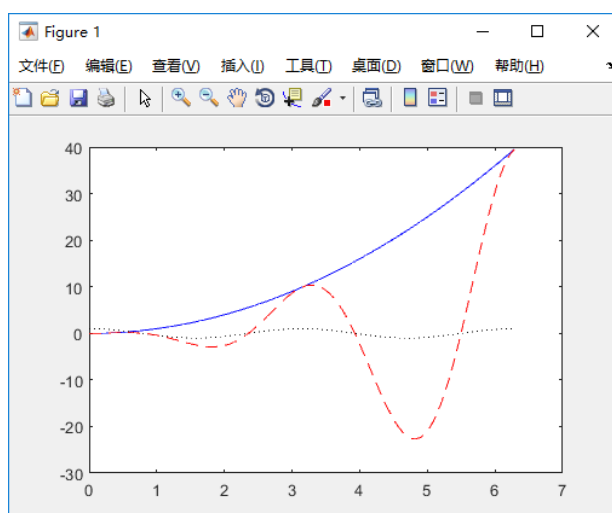
```
plot(x,y1,'b-');
```

```
hold on;
```

%使用 hold on 保持画面，即保留前面曲线不被后面曲线冲掉，这样可以在同一窗口显示多条曲线。关闭保持画面使用 hold off。

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

```
hold off;
```



**【提示】** 用到 linspace 生成线性间距向量函数(p24)，取 101 个点。

**练习 3：** 绘制一条不连续曲线

$$\text{已知 } y = \begin{cases} \frac{x + \sqrt{\pi}}{e^2} & x \leq 0 \\ \frac{1}{2} \ln(x + \sqrt{1 + x^2}) & x > 0 \end{cases}$$

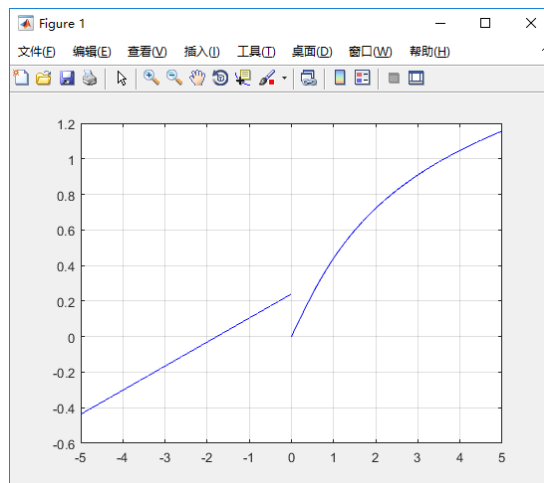
在  $-5 \leq x \leq 5$  区间绘制函数曲线。（注意：本曲线不连续！）

**给出程序和图形：**

```
x1=-5:0.1:0; y1=(x1+sqrt(pi))/exp(2);
x2=0:0.1:5; y2=log(x2+sqrt(1+x2.^2))/2;
plot(x1,y1,'b');
hold on
plot(x2,y2,'b');
```

grid on; % 显示格子以便更好观察曲线，关闭格子线使用grid off。

hold off;



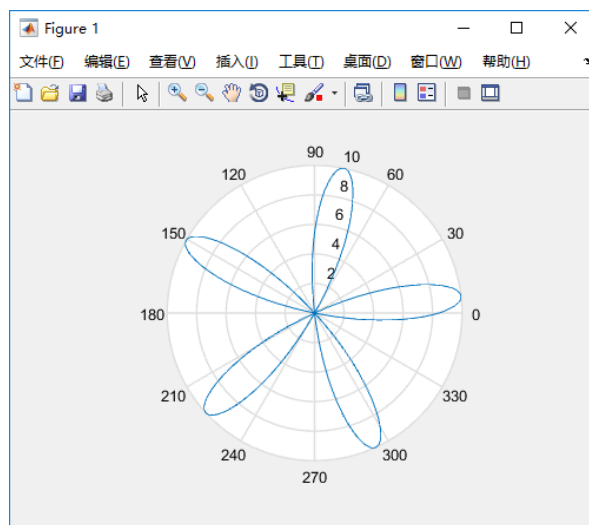
【提示】也可以使用 `plot(x1,y1,'b',x2,y2,'b')` 同时绘制两条线，这样可以省掉 `hold on` 语句，但用上面这种一条语句绘制多条曲线的方式只能设置颜色连接线和点标记这三个属性，其他属性无法设置。例如，可以使用 `plot(x1,y1,'r--*',x2,y2,'g-o')` 绘制两条曲线，但使用 `plot(x1,y1,'r--*', 'linewidth', 1.5, x2,y2,'g-o', 'markersize',6)` 进行额外设置会出错。

练习 4：使用 `polar` 函数绘制极坐标曲线

$$\rho = 10 \sin(1 + 5\theta), \quad 0 \leq \theta \leq 2\pi$$

给出程序和图形：

```
theta=0:0.01:2*pi;  
rho=10*sin(1+5*theta);  
polar(theta,rho);
```



### 【提示】

$\theta$  取的点要足够多。P116 例。

接下来，通过练习 4 学习设置坐标轴名称、图像标题、图例、坐标轴范围等等。

**练习 5：** 在同一坐标系下用不同的颜色和线型绘制 3 条曲线，

已知  $y_1 = x^2$ ， $y_2 = \cos(2x)$ ， $y_3 = y_1 * y_2$ ，其中  $x$  可取  $0 \sim 2\pi$ ，取 101 个点。

(1) 绘制函数图像

同练习 1

(2) 设置  $x$  轴与  $y$  轴标签

```
xlabel('X'); ylabel('Y');
```

(3) 设置  $x$  轴的显示范围是  $[-\pi, 2\pi]$ ， $y$  轴的显示范围为  $[-40, 50]$

```
xlim([-pi, 2*pi]); ylim([-40, 50]);
```

(4) 设置图像的图例

```
legend('y1=x^2', 'y2=cos(2x)', 'y3=x^2cos(2x)')
```

(5) 设置图像标题

```
title('我是图像标题');
```

%如果文字乱码，可在绘图窗口点左键选中标题后点右键，在属性检查器菜单里更改字体

(6) 在图像的 (0.1, 0.2) 位置处显示一个文本框

```
text(0.1,0.2,'我是文本框')
```

(7) 设置坐标轴等距显示

```
>>axis equal
```

(8) 设置坐标轴为常见的十字交叉轴 (此处需要使用绘图对象属性来进行设置)

```
a=gca;
```

%上面语句中 gca 代表当前绘图窗口, 将 gca 赋给 a, 那么 a 就是当前绘图窗口

%不能直接使用 gca 进行图像设置,

%需把 gca 赋值给一个变量, 再使用该对象进行图像设置。

```
a.XAxisLocation='origin'; %x 轴的位置除 origin 外, 还有 top 和 bottom
```

```
a.YAxisLocation='origin'; %y 轴的位置除 origin 外, 还有 left 和 right
```

%给 a 对象的属性 XAxisLocation 和 YAxisLocation 赋值 (**属性名字区别大小写**)

%如果不知道 a 对象有哪些属性和属性名字的大小写情况

%可以在命令行窗口>>a 回车就会显示 a 对象的属性介绍。

-----我是 1 号华丽的分割线-----

除了 plot、polar 绘图函数外, 还有其他绘图函数, 例如

**fplot(fun,[xlim])**、**fimplicit(fun,[xlim,ylim])**(隐函数绘图), 其中 fun 是函数句柄, 使用这两函数绘图时可以不用给出点的坐标矩阵 x 和 y (这和 plot 不同), 只需给出函数句柄 fun, fun 可以是 m 文件函数或者匿名函数

例如

```
>>myfun1=@(x)sin(x);
```

```
>>fplot(myfun1); %等同于 fplot(myfun1,[-5,5]), 此处缺省了[xlim],缺省值为[-5,5]
```

上面语句也可以直接在使用 fplot 时再定义函数句柄, 例如

```
>> fplot(@(x)sin(x))
```

```
>> fimPLICIT(@(x,y)x.^2+y.^2-1,[-1,1,-1,1])绘制单位圆图。
```

其它常见的二维绘图函数见下表，具体使用方法可以查看帮助文档。

表 7.3 特殊的二维绘图函数

函数名称	说明	函数名称	说明
bar	条形图	comet	彗星流动图
errorbar	图形上加误差范围	histogram	直方图（累计图）
polarhistogram	极坐标系下的直方图	stair	阶梯图
stem	针状图	fill	多边形填充图
feather	羽毛图	compass	罗盘图
quiver	向量场图	scatter	散点图
pie	饼图	refline	画直线
refcurve	画曲线		

这里比较常用函数是 `refline`，其基本调用格式是 `refline(k,b)`：绘制斜率为  $k$  截距为  $b$  的直线，常用来绘制图形参考线。

比较有趣的是 `comet` 函数，绘图时会显示一个从开始点到最终点的动画，常用于演示轨迹。

例如：`x=0:0.01:5;y=x+3;comet(x,y);`

-----我是 2 号华丽的分割线-----

在同一绘图窗口绘制多个坐标系。

1、可以用 `subplot(m,n,p)` 在同一个绘图窗口绘制多个坐标系。

其中  $m$ ， $n$  是指把绘图窗口分割成  $m*n$  个子区域， $p$  的取值范围是  $1 \sim m*n$ ，代表这  $m*n$  个子区域中的第  $p$  块区域，例如

`subplot(2,2,1);plot (...)`

`subplot(2,2,2);plot(...)`

`subplot(2,2,3);plot(...)`

`subplot(2,2,4);plot(...)`

这四条语句分别在绘图窗口的 1，2，3，4 块区域上绘图。

**练习 6：**在子图上绘制  $y_1 = x^2$ ， $y_2 = \cos(2x)$ ， $y_3 = y_1 * y_2$  的函数图，

其中  $x$  可取  $0 \sim 2\pi$ ，取 101 个点。

给出程序和图形

```
x=linspace(0,2*pi,101);
```

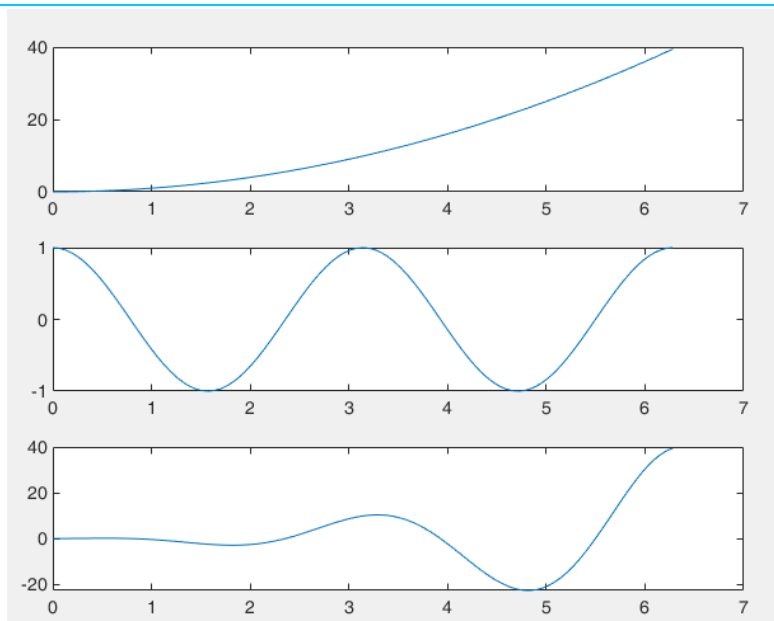


```
y1=x.^2; y2=cos(2*x); y3=y1.*y2;
```

```
subplot(3,1,1); plot(x,y1); %将绘图窗口分为3行一列，这是第一块，即第一行
```

```
subplot(3,1,2); plot(x,y2); %这是第二块，即第二行
```

```
subplot(3,1,3); plot(x,y3); %这是第三块，即第三行
```



### 练习 7：分别用条形图、阶梯图、杆图和填充图绘制 3 条曲线

将图形窗口分成 3×4 个绘图区，

第 1 行绘制  $y1 = x^2$  的 4 个图形

第 2 行绘制  $y2 = \cos(2x)$  的 4 个图形

第 3 行绘制  $y3 = y1 * y2$  的 4 个图形。

**给出程序和图形（要求： $x$  取  $0 \sim 2\pi$ ，取 10 个点。）：**

```
x=linspace(0,2*pi,10);
```

```
y1=x.^2;
```

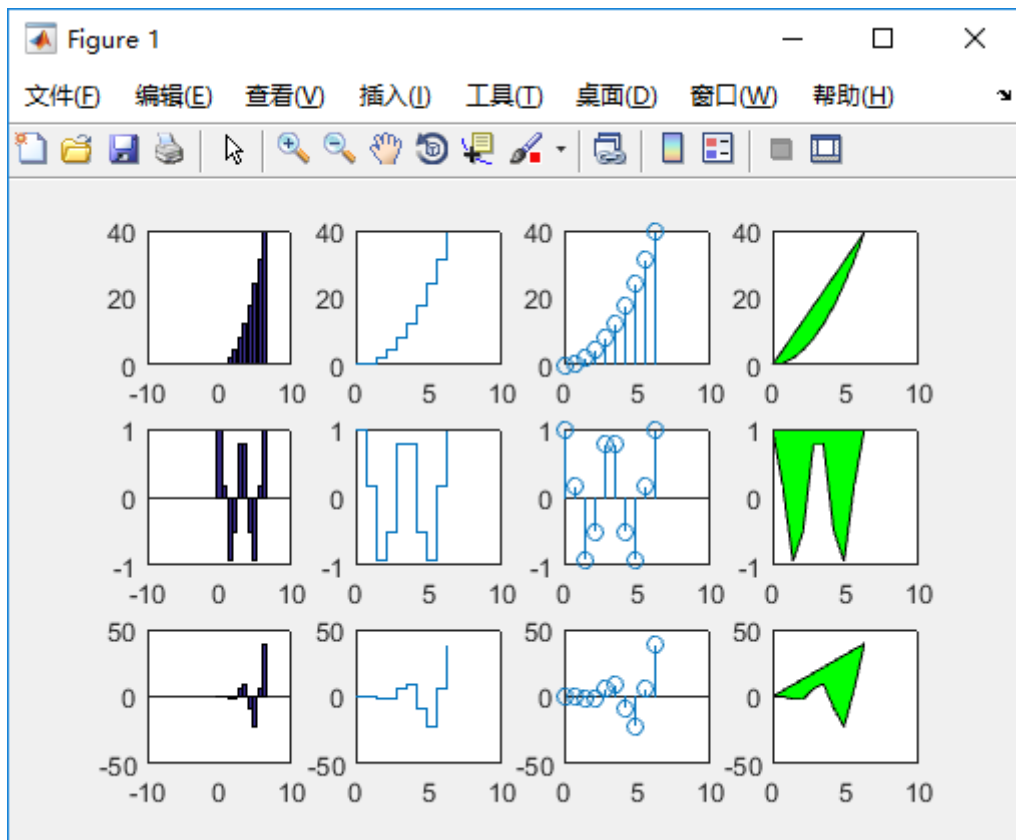
```
subplot(3,4,1);bar(x,y1); % 条形图
```

```
subplot(3,4,2);stairs(x,y1);% 阶梯图
```

```

subplot(3,4,3);stem(x,y1);% 杆图
subplot(3,4,4);fill(x,y1,'g');% 填充图
y2=cos(2*x);
subplot(3,4,5);bar(x,y2); % 条形图
subplot(3,4,6);stairs(x,y2);% 阶梯图
subplot(3,4,7);stem(x,y2);% 杆图
subplot(3,4,8);fill(x,y2,'g');% 填充图
y3=y1.*y2;
subplot(3,4,9);bar(x,y3); % 条形图
subplot(3,4,10); stairs(x,y3);% 阶梯图
subplot(3,4,11); stem(x,y3);% 杆图
subplot(3,4,12); fill(x,y3,'g');% 填充图

```



## 2、也可以使用 tiledlayout(m,n) 函数来替代 subplot 绘制多个子图。

```
x=linspace(0,2*pi,30);
y1=x.^2;
y2=cos(2*x);
y3=y1.*y2;
T=tiledlayout(3,4); %将绘图窗口分为 3*4=12 块区域
nexttile; %该命令使下一个坐标区获得焦点。
bar(x,y1); %在点 x 的位置 (x 位于条的正中间) 用对应的 y 值做高作矩形。
nexttile;
stairs(x,y1,'-','linewidth',1); %点和点之间不是直接连接而是用横竖折线连接
nexttile;
stem(x,y1,'markerfacecolor','r','markeredgecolor','g'); %图形类似于火柴棍，火柴棍的头部高度是 y 的值。
nexttile;
area(x,y1);
nexttile;
bar(x,y2);
nexttile;
stairs(x,y2);
nexttile;
stem(x,y2);
nexttile;
area(x,y2);
nexttile;
bar(x,y3);
nexttile;
stairs(x,y3);
nexttile;
stem(x,y3);
nexttile;
area(x,y3);
```

## 3、可以使用 figure 语句建立一个新的绘图窗口。

>>figure %创建名字叫 figure 1 的绘图窗口

>>figure %如果 figure 1 没关闭，则创建的绘图窗口会被命名为 figure 2，否则命名为 figure 1

注：当有多个绘图窗口时比如有 figure 1、figure 2、figure 3，如果想要在 figure 2 上绘图，需先用语句>>figure(2) 激活该窗口（用鼠标点击激活也可以），然后再运行绘图语句进行绘图。