

图线的绘制与装饰

使用 `plot()` 函数绘制图线

装饰图线

控制坐标轴,边框与网格

绘制多条图线

在一个图像上绘制多条图线

在一个窗口内绘制多个图像

图形对象的操作

获取图形句柄

通过图形句柄操作图形属性

将图形保存到文件

学习一门技术最好的方式就是阅读官方文档,可以查看[MATLAB官方文档](#)

图线的绘制与装饰

使用 `plot()` 函数绘制图线

在MATLAB中,使用 `plot()` 函数绘制图线,其语法为:

```
1 | plot(x,y,LineStyle)
```

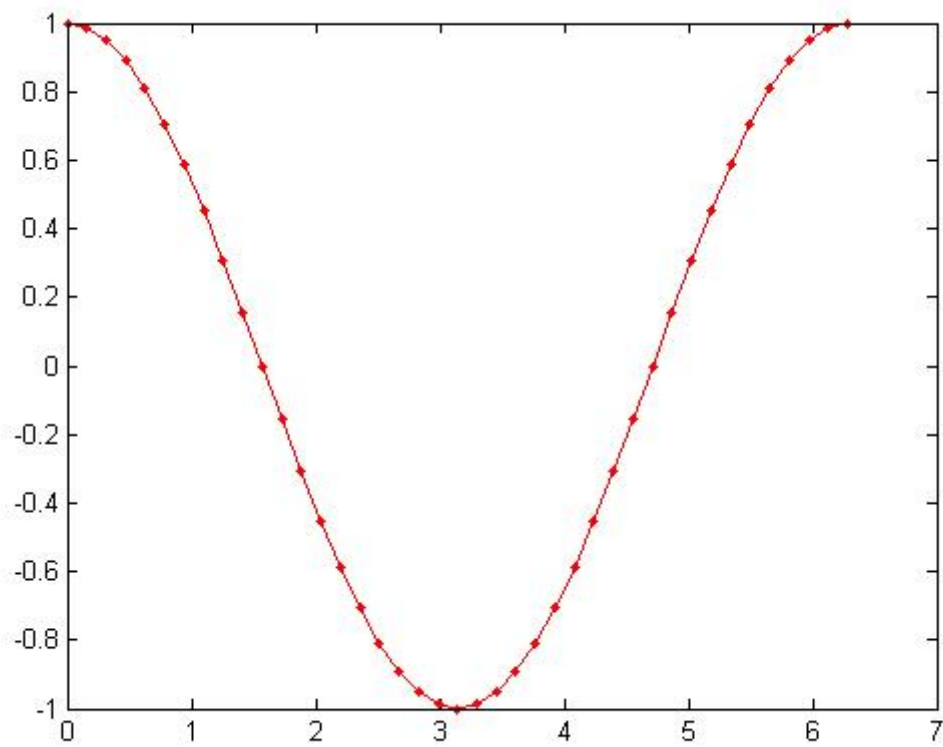
各参数意义如下:

- `x`: 图线上点的x坐标
- `y`: 图线上点的y坐标
- `LineStyle`: 图线的线条设定,三个指定**线型**,**标记符号**和**颜色**的**设定符**组成一个字符串,设定符不分先后.具体细节请参考[官方文档](#).

线型符号	线型设定符	标记	标记设定符	颜色	颜色设定符
-	实线（默认）	o	圆圈	y	黄色
--	虚线	+	加号	m	品红色
:	点线	*	星号	c	青蓝色
-.	点划线	.	点	r	红色
		x	叉号	g	绿色
		s	方形	b	蓝色
		d	菱形	w	白色
		^	上三角	k	黑色
		v	下三角		
		>	右三角		
		<	左三角		
		p	五角形		
		h	六角形		

下面例子演示了绘制 $(0, 2\pi)$ 内余弦函数的图像:

```
1 x = 0:pi/20:2*pi;  
2 y = cos(x);  
3 plot(x, y, 'r.-')
```

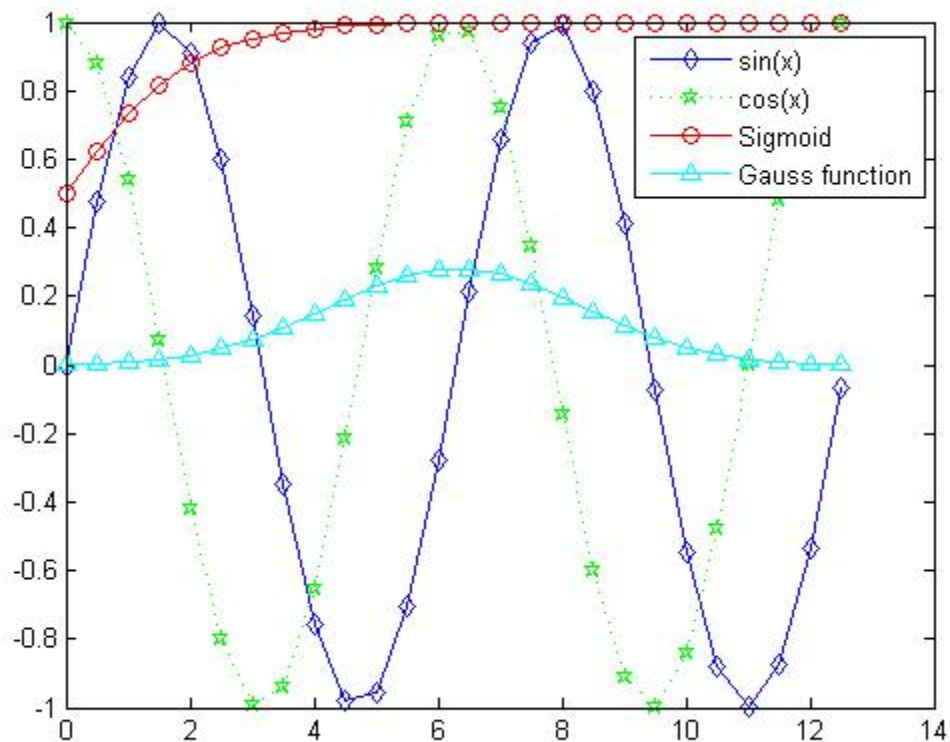


装饰图线

- 使用 `legend()` 函数为图片增加图例

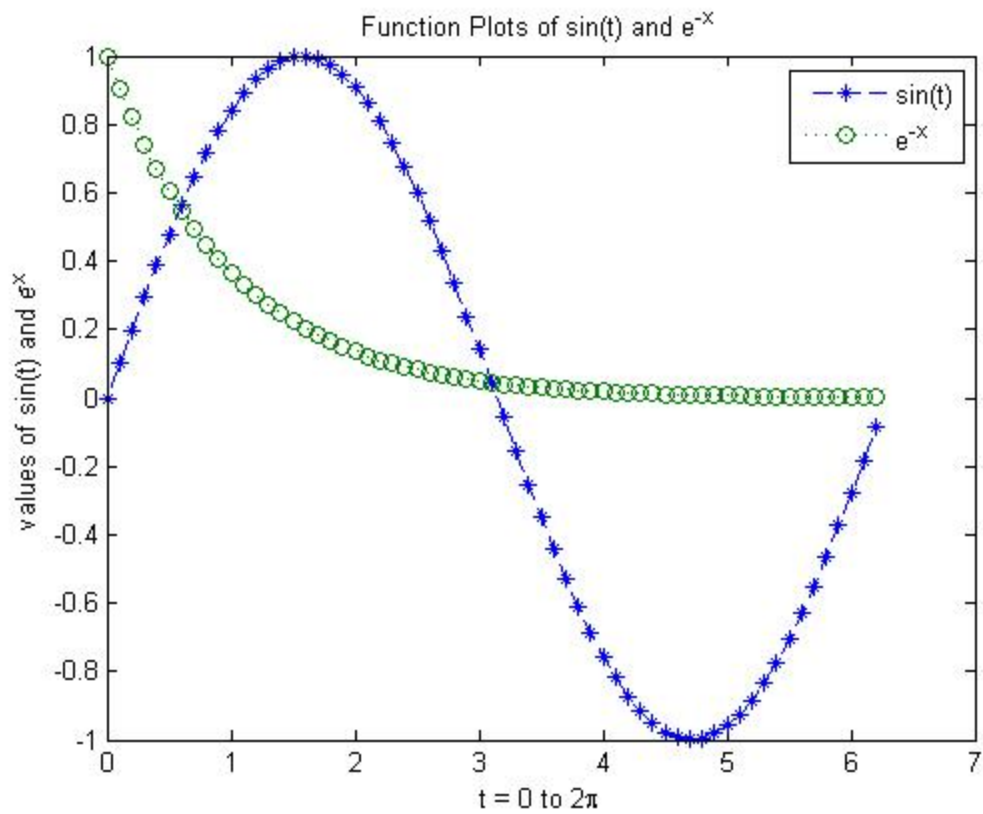
使用 `legend(label1, ..., labelN)` 函数可以为图片添加图例.

```
1 x=0:0.5:4*pi;  
2 y=sin(x); h=cos(x); w=1./(1+exp(-x)); g=(1/(2*pi*2)^0.5).*exp((-1.*(x-  
3 2*pi).^2)./(2*2^2));  
4 plot(x,y,'bd-',x,h,'gp:',x,w,'ro-',x,g,'c^--'); % 绘制多条图线  
5 legend('sin(x)','cos(x)','Sigmoid','Gauss function'); % 添加图例
```



- 使用 `title()` 和 `*label()` 为图片增加标题和标签

```
1 x = 0:0.1:2*pi; y1 = sin(x); y2 = exp(-x);  
2 plot(x, y1, '--*', x, y2, ':o');  
3 xlabel('t = 0 to 2\pi');  
4 ylabel('values of sin(t) and e^{-x}');  
5 title('Function Plots of sin(t) and e^{-x}');  
6 legend('sin(t)', 'e^{-x}');
```

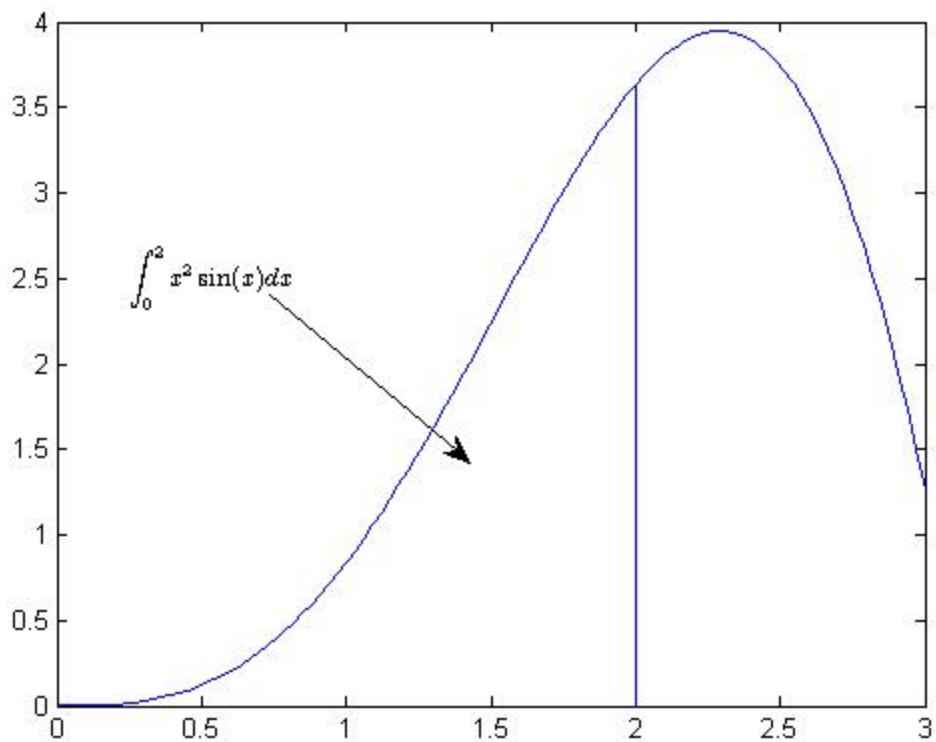


- 使用 `text()` 和 `annotation()` 为图片增加注解

```

1 x = linspace(0,3); y = x.^2.*sin(x); plot(x,y);
2 line([2,2],[0,2^2*sin(2)]);
3 str = '$$ \int_{0}^{2} x^2 \sin(x) dx $$';
4 text(0.25,2.5,str,'Interpreter','latex');
5 annotation('arrow','x',[0.32,0.5],'y',[0.6,0.4]);

```



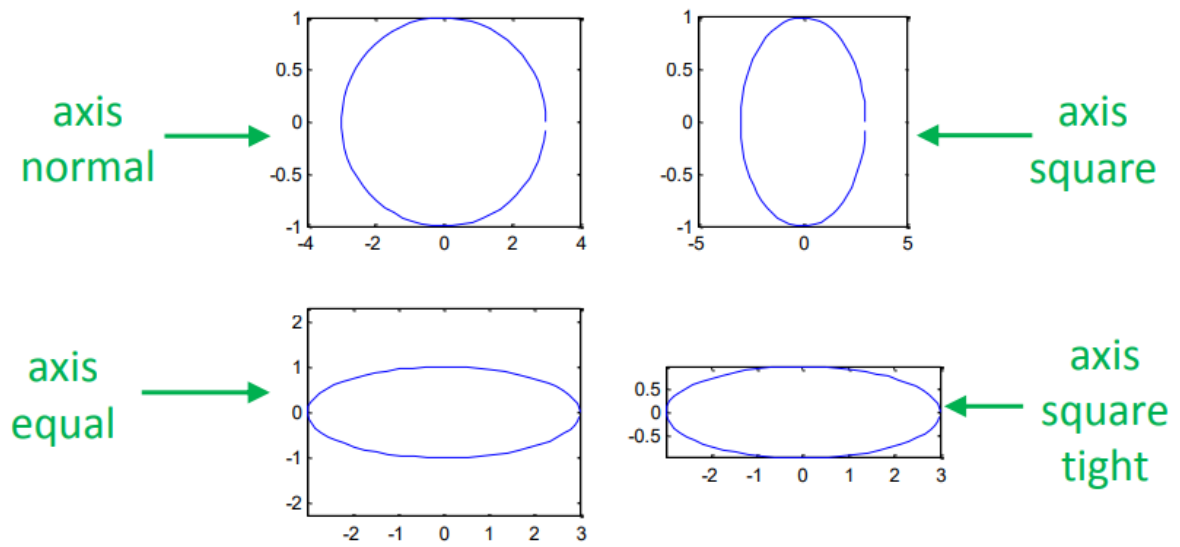
控制坐标轴,边框与网格

使用下列命令可以控制坐标轴,边框与网格.

命令	作用
<code>grid on/off</code>	设置网格可见性
<code>box on/off</code>	设置边框可见性
<code>axis on/off</code>	设置坐标轴可见性
<code>axis normal</code>	还原默认行为,将图框纵横比模式和数据纵横比模式的属性设置为自动
<code>axis square</code>	使用相同长度的坐标轴线,相应调整数据单位之间的增量
<code>axis equal</code>	沿每个坐标轴使用相同的数据单位长度
<code>axis tight</code>	将坐标轴范围设置为等同于数据范围,使轴框紧密围绕数据

下面的例子演示 `axis` 命令的效果:

```
1 t = 0:0.1:2*pi; x = 3*cos(t); y = sin(t);
2 subplot(2, 2, 1); plot(x, y); axis normal
3 subplot(2, 2, 2); plot(x, y); axis square
4 subplot(2, 2, 3); plot(x, y); axis equal
5 subplot(2, 2, 4); plot(x, y); axis equal tight
```

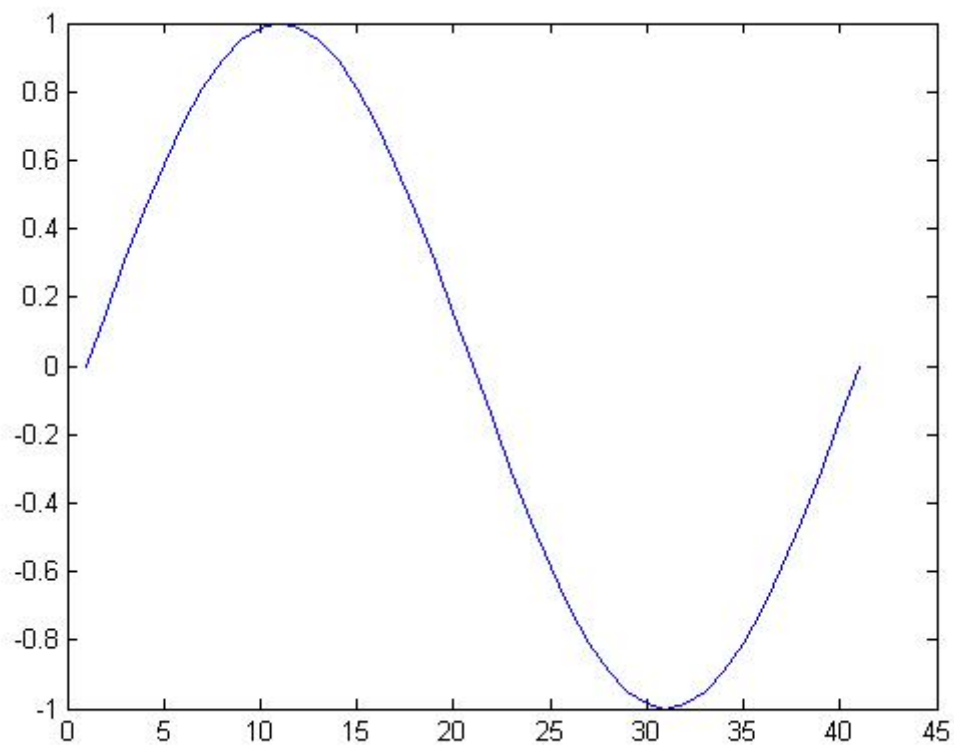


绘制多条图线

在一个图像上绘制多条图线

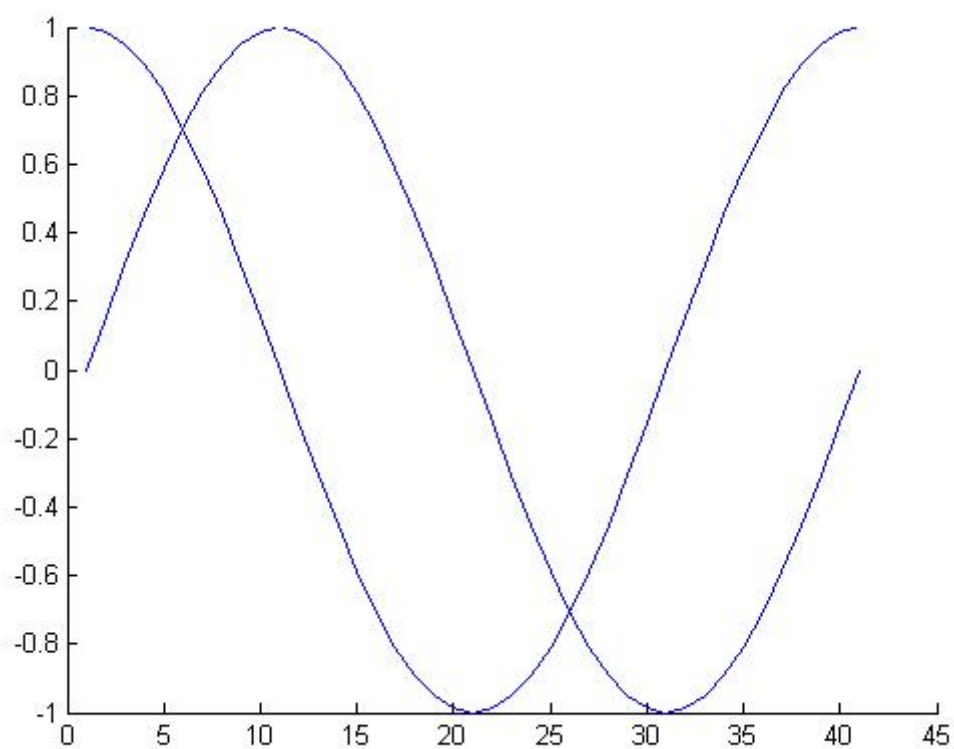
默认情况下,每次执行 `plot()` 函数都会清除上一次绘图的结果,多次执行 `plot()` 只会保留最后一次绘制的图形.

```
1 plot(cos(0:pi/20:2*pi));
2 plot(sin(0:pi/20:2*pi));
```



我们可以使用 `hold on` 和 `hold off` 命令控制绘图区域的刷新,使得多个绘图结果同时保留在绘图区域中.

```
1 hold on      % 提起画笔,开始绘制一组图片
2 plot(cos(0:pi/20:2*pi));
3 plot(sin(0:pi/20:2*pi));
4 hold off     % 放下画笔,该组图片绘制完毕
```



在一个窗口内绘制多个图像

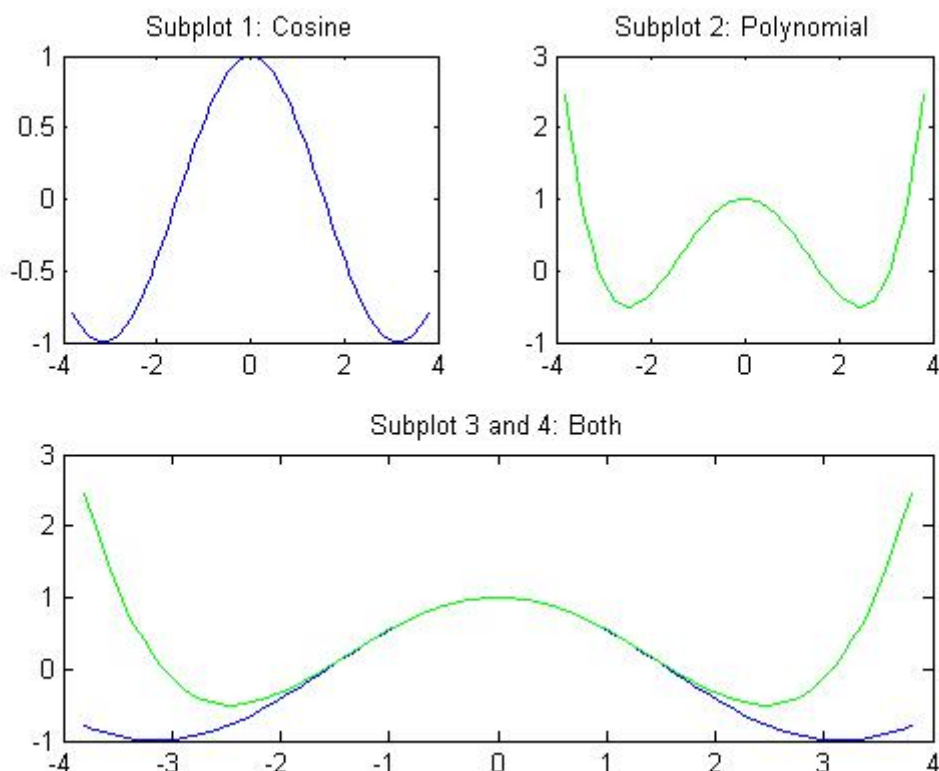
使用 `subplot()` 函数可以在一个窗口内绘制多个图像.其语法为:

```
1 subplot(m,n,p)
```

该命令表示将当前图窗划分为 $m \times n$ 个网格,并在第 p 个网格内绘制图像.

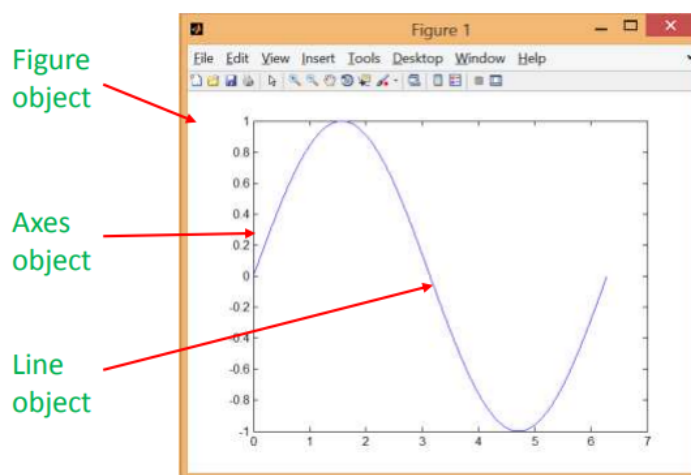
示例如下:

```
1 subplot(2,2,1);
2 x = linspace(-3.8,3.8);
3 y_cos = cos(x);
4 plot(x,y_cos);
5 title('Subplot 1: Cosine')
6
7 subplot(2,2,2);
8 y_poly = 1 - x.^2./2 + x.^4./24;
9 plot(x,y_poly,'g');
10 title('Subplot 2: Polynomial')
11
12 subplot(2,2,[3,4]);
13 plot(x,y_cos,'b',x,y_poly,'g');
14 title('Subplot 3 and 4: Both')
```

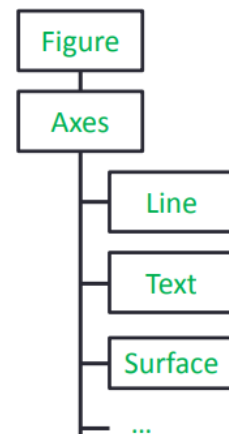


图形对象的操作

在MATLAB中,图形都是以对象的形式储存在内存中,通过获取其图形句柄可以对其进行操作.



Hierarchy

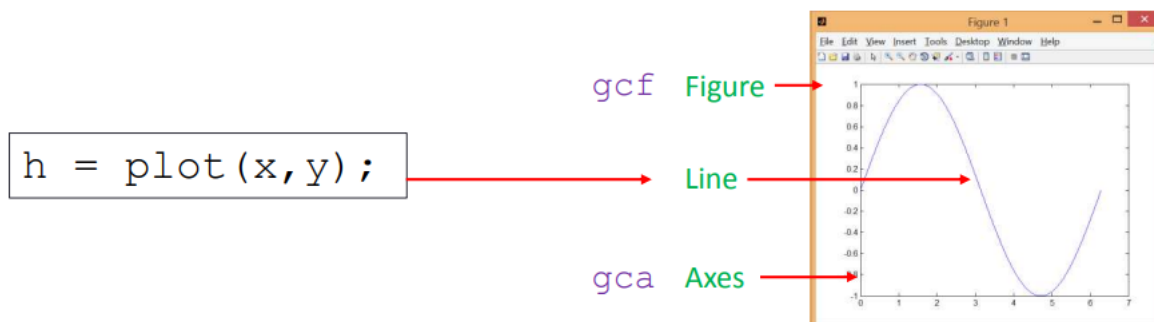


获取图形句柄

图形句柄本质上就是一个浮点数,可以唯一确定一个图形对象.下面几个函数用于获取图形句柄.

Function	Purpose
<code>gca()</code>	获取当前坐标轴的句柄
<code>gcf()</code>	获取当前图像的句柄
<code>allchild(handle_list)</code>	获取该对象的所有子对象的句柄
<code>ancestor(h,type)</code>	获取对象最近的 type 类型的祖先节点
<code>delete(h)</code>	删除某对象
<code>findall(handle_list)</code>	获取该对象的后代对象

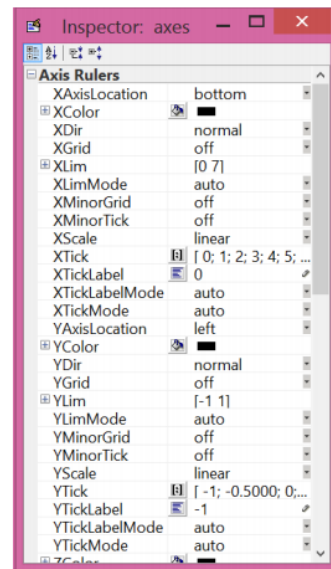
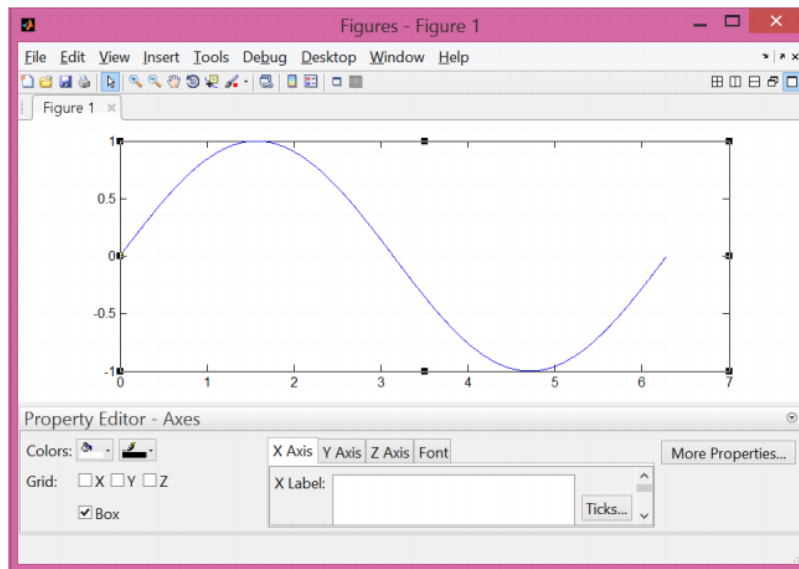
所有绘图函数也会返回图形对象的句柄.



通过图形句柄操作图形属性

使用 `get()` 和 `set()` 函数可以对图形对象的属性进行访问和修改.访问[官方文档](#)可以查看所有图形对象的属性.

- `set(H,Name,Value)`
- `v = get(h,propertyName)`



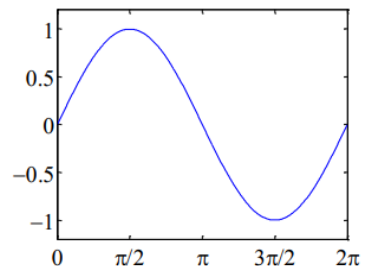
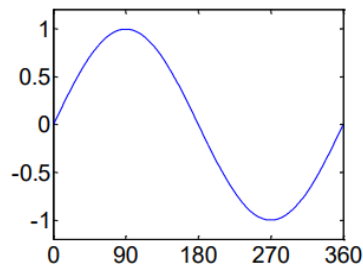
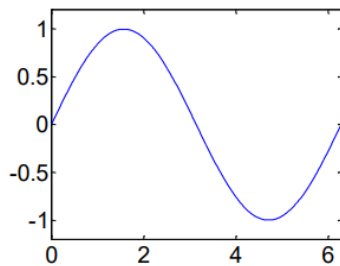
下面两个例子演示使用图形句柄操作图形对象:

1. 改变坐标轴属性:

```

1 % 第一张图
2 set(gca, 'FontSize', 25);
3
4 % 第二张图
5 set(gca, 'XTick', 0:pi/2:2*pi);
6 set(gca, 'XTickLabel', 0:90:360);
7
8 % 第三张图
9 set(gca, 'FontName', 'symbol');
10 set(gca, 'XTickLabel', {'0', 'p/2', 'p', '3p/2', '2p'});

```

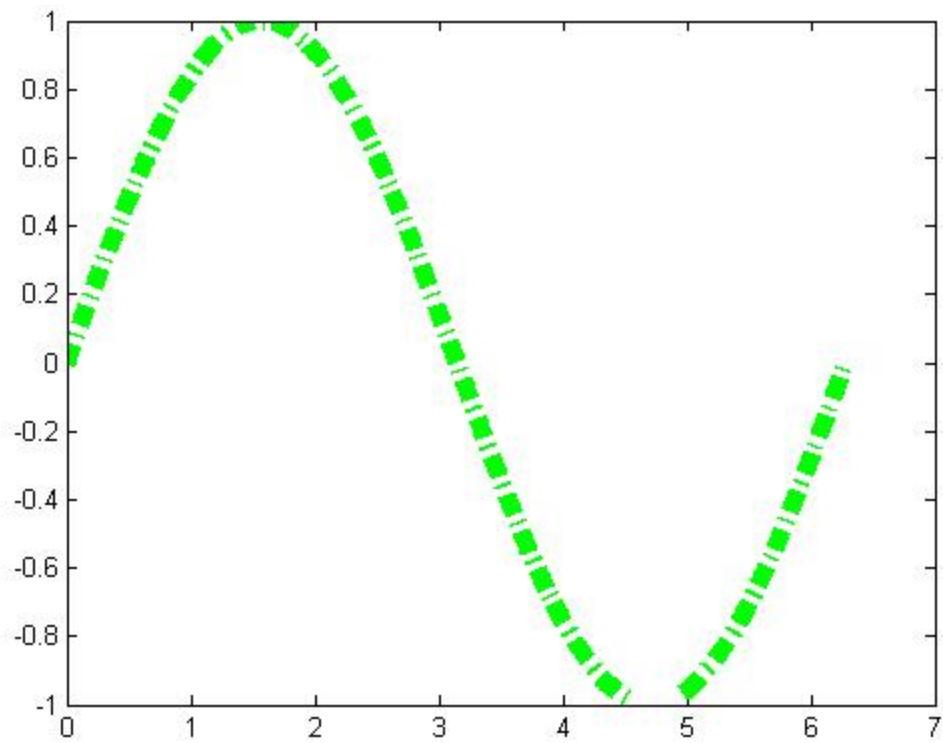


2. 改变线型

```

1 h = plot(x,y);
2 set(h, 'LineStyle', '-.', ...
3       'Linewidth', 7.0, ...
4       'color', 'g');

```



将图形保存到文件

使用 `saveas(fig, filename)` 命令可以将图形对象保存到文件中,其中 `fig` 为图形句柄, `filename` 为文件名.

```
1 | saveas(gcf, 'myfigure.png')
```

使用 `saveas()` 函数将图像保存成位图时,会发生失真.要精确控制生成图片的质量,可以使用 `print()` 函数,见[官方文档](#)