



数据和函数的可视化

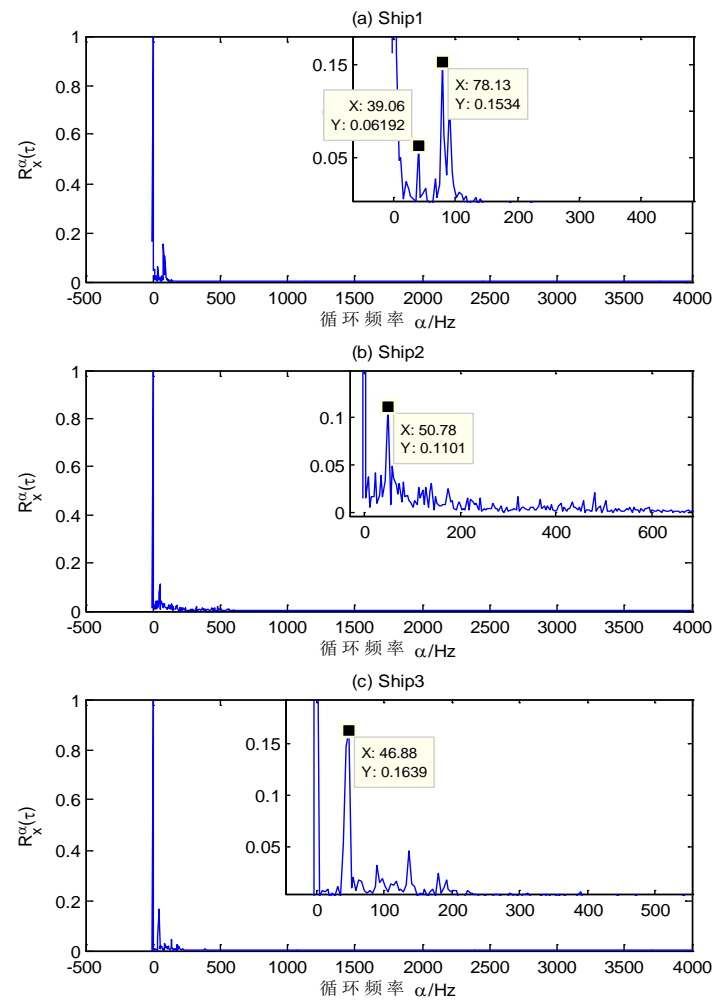
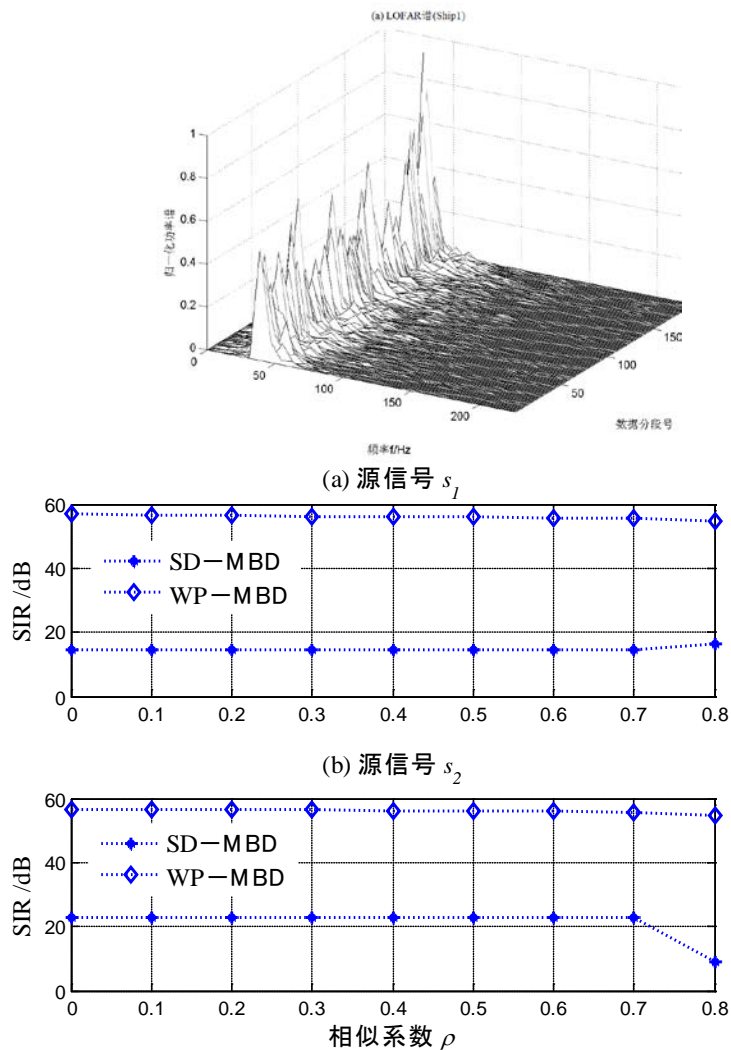
第7讲



引言

- MATLAB具有极其强大的数据可视化功能。
- 数据可视化命令plot。
- 详细介绍MATLAB这一部分的内容可以写一本书。我们只能介绍MATLAB数据可视化的基础，2-D数据可视化、3-D数据可视化初步。
- 二维图形是将平面坐标上的数据点连接起来的平面图形。可以采用直角坐标系、对数坐标、极坐标等形式。

引言





引言

➤ 绘图的基本步骤:

- ① 使用MATLAB的绘图函数绘制图形或者曲线;
- ② 设置曲线的属性, 例如线型、线宽等;
- ③ 设置绘图区域的属性, 并且添加数据网格线;
- ④ 为绘制的图形添加标题、轴标签或者标注文本等;



主要内容

- 7.1 二维绘图
- 7.2 图形控制
- 7.3 图形标识
- 7.4 特殊图形
- 7.5 三维绘图
- 7.6 函数绘图



7.1 二维绘图

■ plot指令的基本调用格式

(1) plot(x)

- **x为向量时，以该元素的下标为横坐标、元素值为纵坐标绘出曲线**
- **x为实数二维数组时，则按列绘制每列元素值相对其下标的曲线，曲线数等于x数组的列数。**
- **x如果为复数二维数组？**



7.1 二维绘图

■ plot指令的基本调用格式

(2) plot(x, y)

- **x、y为同维数组时，绘制以x、y元素为横纵坐标的曲线**
- **x为向量，y为二维数组、且其列数或行数等于x的元素数时，绘制多条不同颜色的曲线**
- **x为二维数组，y为向量时，情况与上相同，只是y仍为纵坐标。**



7.1 二维绘图

■ plot指令的基本调用格式

(3) `plot(x1, y1, x2, y2, ...)`

- 绘制以 $x1$ 为横坐标、 $y1$ 为纵坐标的曲线1，以 $x2$ 为横坐标、 $y2$ 为纵坐标的曲线2，等等。
- 其中 x 为横坐标， y 为纵坐标，绘制 $y=f(x)$ 函数曲线。

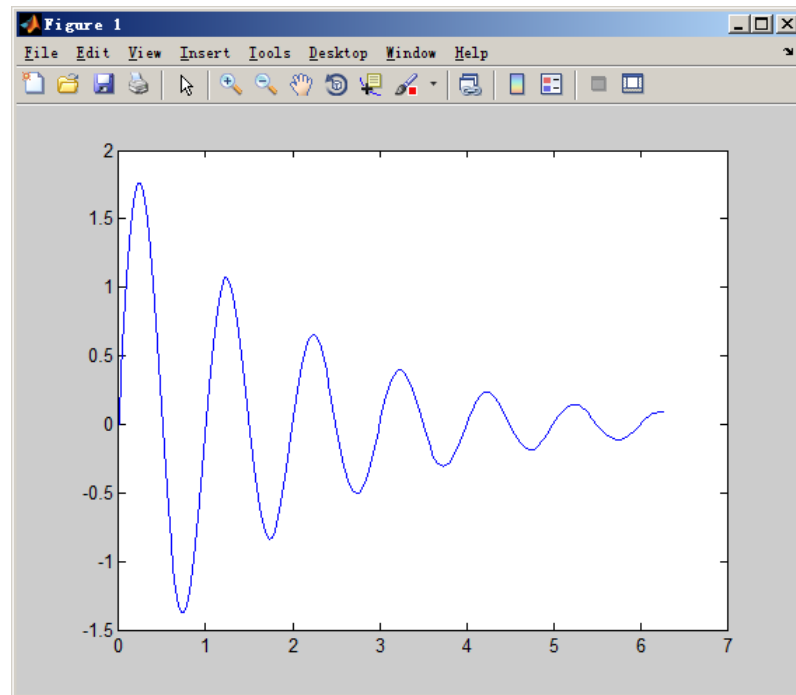
例:plot绘制曲线(一次一条曲线)

在 $[0, 2\pi]$ 区间内, 绘制曲线 $y = 2e^{-0.5x} \sin(2\pi x)$

```
x = 0:pi/100:2*pi;
```

```
y = 2*exp(-0.5*x).*sin(2*pi*x);
```

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



例:plot绘制曲线(参数方程)

绘制曲线

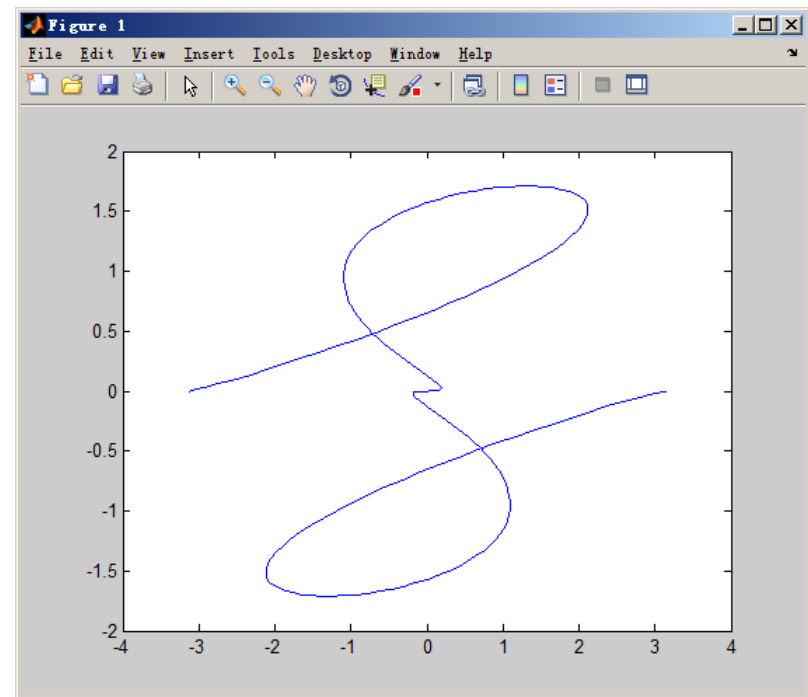
$$\begin{cases} x = t \cos(3t) \\ y = t \sin^2 t \end{cases}, -\pi \leq t \leq \pi$$

t = -pi:pi/100:pi;

x = t.*cos(3*t);

y = t.*sin(t).^2;

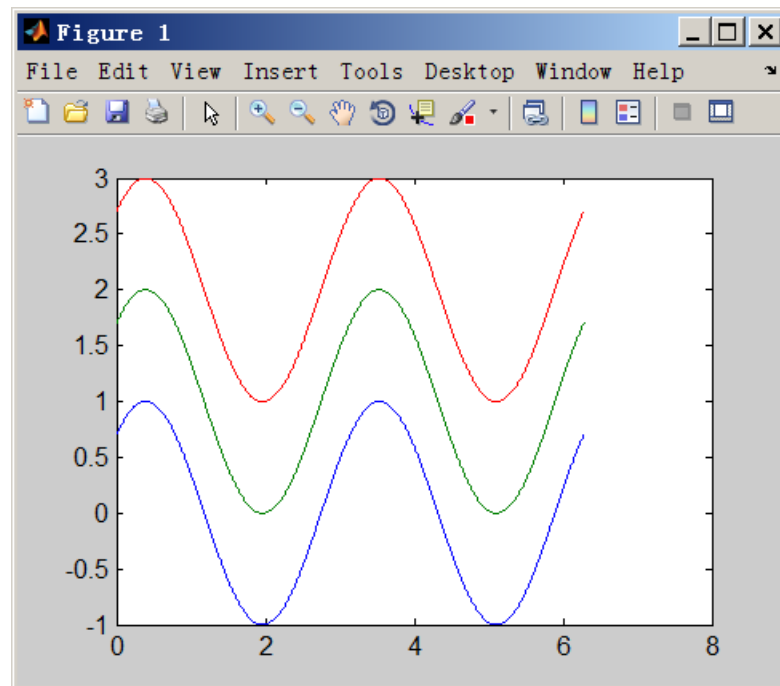
plot(x,y)



例:plot绘制曲线(一次多条曲线)

```
x = 0:pi/1000:2*pi;  
y = sin(2*x+pi/4);  
plot(x,y,x,y+1,x,y+2)
```

以 (x,y) 、 $(x,y+1)$ 、 $(x,y+2)$
为三组坐标绘制曲线
三条曲线不同颜色





多次叠绘、双纵坐标和多子图

■ 多次叠绘

- 多次调用plot命令在一幅图上绘制多条曲线，需要**hold**指令的配合。
- **hold on** 保持当前坐标轴和图形，并可以接受下一次绘制。
- **hold off** 取消当前坐标轴和图形保持，这种状态下，调用plot绘制完全新的图形，不保留以前的坐标格式、曲线。

例:叠绘曲线

叠绘波形曲线 $y = \sin(t) \sin(9t)$ 及其包络线。

`t=(0:pi/100:pi)';` %长度为101的时间采样列向量

`y1=sin(t)*[1,-1];` %包络线函数值, 是(101x2)的矩阵

`y2=sin(t).*sin(9*t);` %长度为101的调制波列向量

`t3=pi*(0:9)/9;`

`y3=sin(t3).*sin(9*t3);`

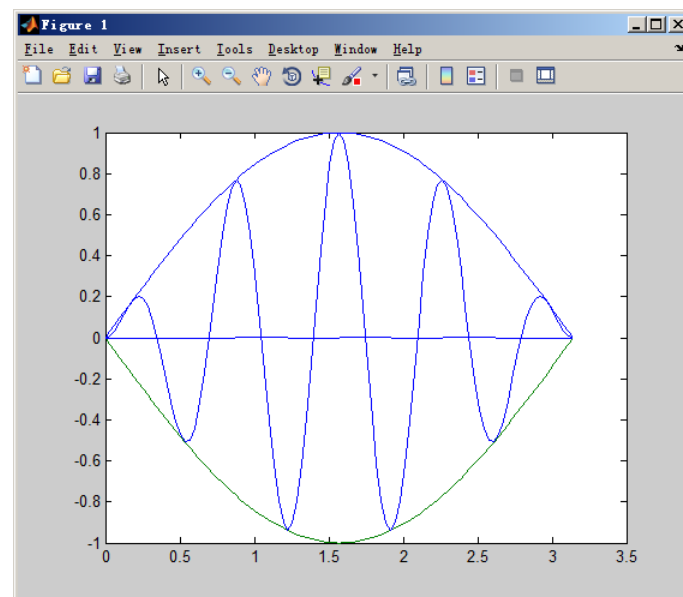
`plot(t,y1)`

`hold on`

`plot(t,y2)`

`plot(t3,y3)`

`hold off`





多子图

- MATLAB允许在同一图形窗口布置几幅独立的子图。

- `subplot(m, n, k)`

将图形窗口分割成的 m 行和 n 列子图，并设置其中第 k 个为当前绘图窗口

- **编号原则：** 左上角为第1子图，按行顺序排列。
- 产生的子图彼此独立。所有的绘图指令均可以在子图中使用。

subplot分割图形窗口

$t=0:0.1:2*\pi;$

$y1=\sin(t);$

$y2=\cos(10*t);$

$y3=\sin(t).*\sin(10*t);$

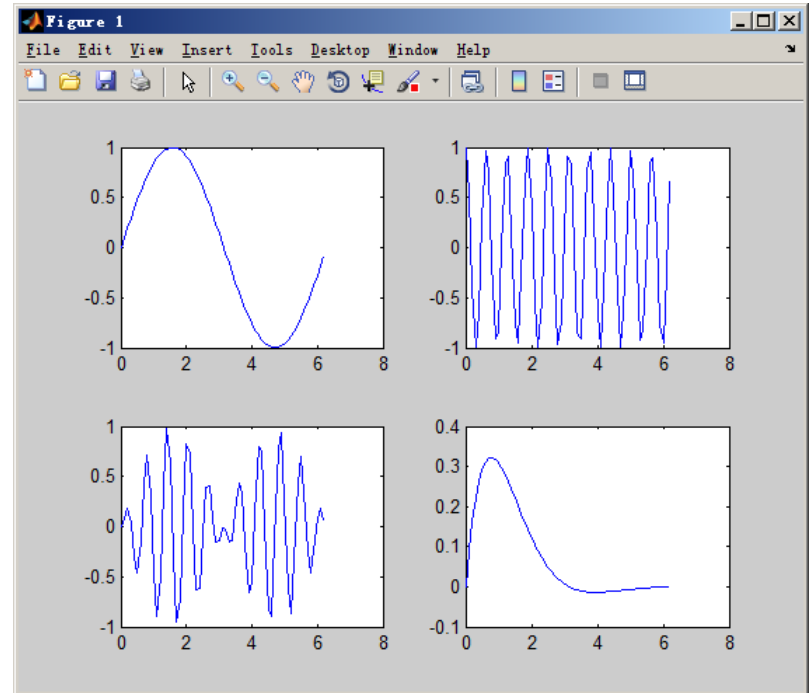
$y4=\sin(t).*\exp(-t);$

subplot(2,2,1),plot(t,y1);

subplot(2,2,2),plot(t,y2);

subplot(2,2,3),plot(t,y3);

subplot(2,2,4),plot(t,y4);



7.2.1 设置曲线的样式

- MATLAB通过不同的参数，控制曲线的颜色、线型和数据点的标识形状。

色彩 color	说明	标识 marker	说明	线型 linestyle	说明
r	红色	+	加号	-	实线
g	绿色	o	圆圈	--	虚线
b	蓝色	*	星号	:	点线
c	青	.	点	-o	点划线
m	洋红	x	十字		
y	黄色	s	矩形		
k	黑色	d	菱形		
w	白色	^	上三角		
		v	下三角		
		>	右三角		
		<	左三角		
		p	五边形		
		h	六边形		

7.2.2 曲线的色彩、线型和数据点形

其他曲线细节的设置：

LineWidth：曲线的粗细，单位为磅。

MarkerEdgeColor：数据点标识边缘的颜色。

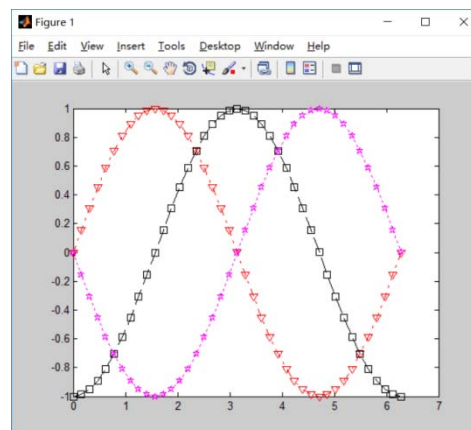
MarkerFaceColor：填充数据点标识的颜色。

MarkerSize：数据点标识的大小，单位为磅。

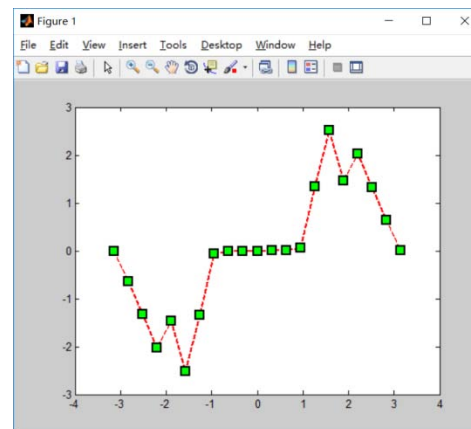
- 曲线的线形控制符、颜色控制符、数据点形控制符可以组合使用
- 其先后次序不影响绘图结果
- 也可以单独使用。

7.2.2 曲线的色彩、线型和数据点形

```
t = 0:pi/20:2*pi;  
y = sin(t);  
y2 = sin(t-pi/2);  
y3 = sin(t-pi);  
plot(t,y,'-rv',t,y2,'--ks',t,y3,':mp')  
% plot(t,y,'rv',t,y2,'ks',t,y3,'mp')
```



```
x = -pi:pi/10:pi;  
y = tan(sin(x)) - sin(tan(x));  
plot(x,y,'--rs','LineWidth',2, 'MarkerEdgeColor','k',...  
      'MarkerFaceColor','g ', 'MarkerSize',10)
```



7.2.3 坐标轴控制

MATLAB能自动根据数据调整坐标轴的显示范围。
`axis`函数可以修改当前坐标轴的范围，其语法格式：
`axis([xmin xmax ymin ymax])`

设置刻度

指令及格式：

`set(gca, 'XTick', xs, 'YTick', ys)`

`set(gca, 'XTickLabel', {'', ''}, ...)`

- `xs`、`ys`可以使任何合法的实数向量，用于分别设置`x`、`y`轴的刻度。
- `gca`表示当前`axis`对象的句柄。

例：设置刻度、刻度标签

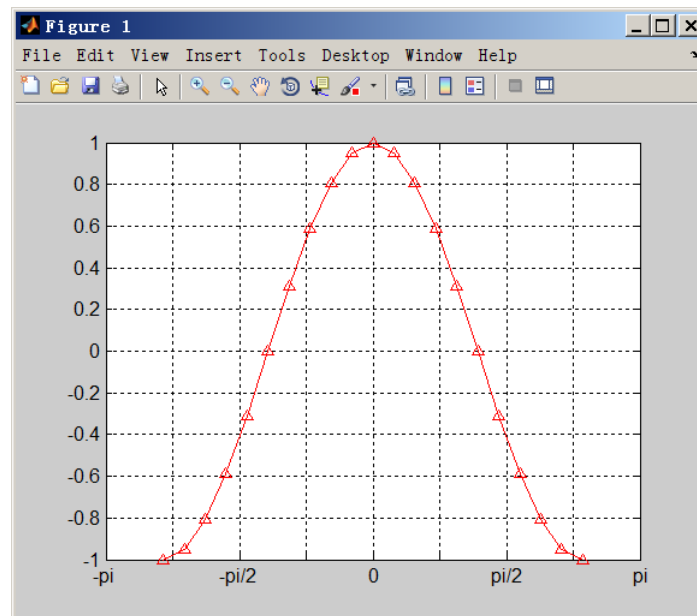
```
x = -pi:pi/10:pi;
```

```
y = cos(x);
```

```
plot(x,y,'-r^');
```

```
set(gca,'XTick', -pi:pi/4:pi)
```

```
set(gca, 'XTickLabel',{'-pi','', '-pi/2','', '0','', 'pi/2','', 'pi'});
```





7.2.3 分格线和坐标框

■ 分格线与grid指令

- grid on 画出分格线
- grid off 不画分格线
- MATLAB的缺省设置是不画分格线；分格线的疏密取决于坐标刻度（改变坐标刻度，可改变分格线的疏密）。

■ 坐标框

- box on 控制加边框线
- box off 控制不加边框线

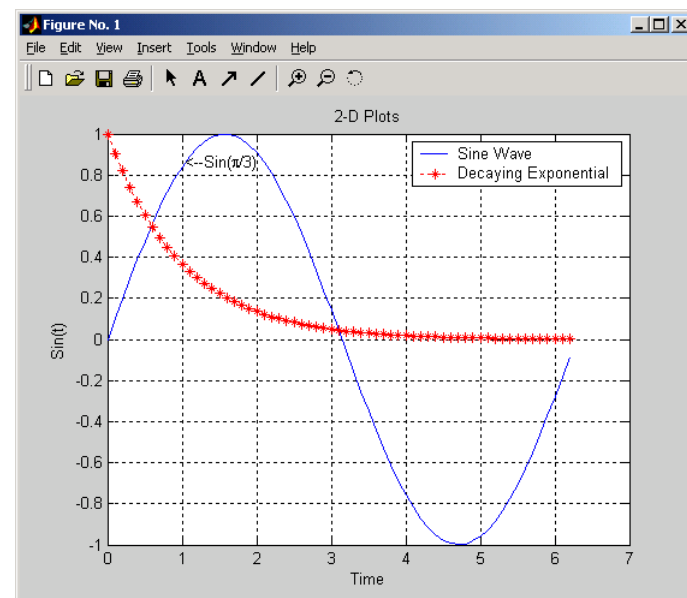
7.3 图形标识

■ 图形标识包括:

- 图名 (title)
- 坐标轴名 (xlabel、ylabel)
- 图形文本注释 (text)
- 图例 (legend)

■ 基本语法格式

- title(s) % s为字符串变量或常量
- xlabel(s)
- ylabel(s)
- legend(s)
- text(x, y, s) % 指定坐标 (x, y) 处加注文字



7.3.1 添加图名

- title函数的基本调用格式为：

`title('string')`

其中：字符串string为要添加的图名

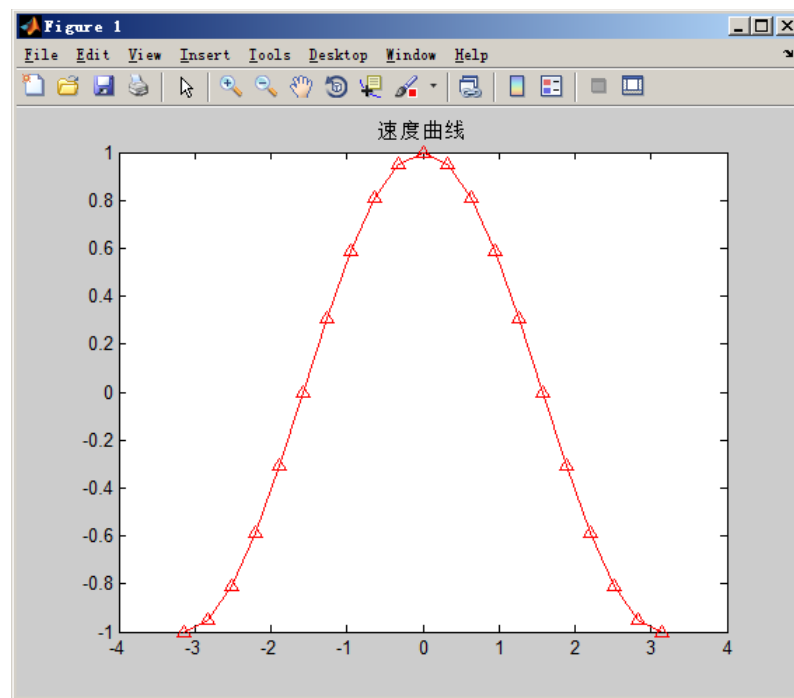
例如：

`x = -pi:pi/10:pi;`

`y = cos(x);`

`plot(x,y,'-r^');`

`title('速度曲线')`



7.3.2 添加坐标轴名

使用xlabel、ylabel和zlabel函数分别为图形窗体的X轴、Y轴和Z轴添加轴标签。以X轴为例，基本调用格式为：

xlabel('string')

- string就是坐标轴的标签。
- 坐标轴的标签自动与坐标轴居中对齐。

例如：

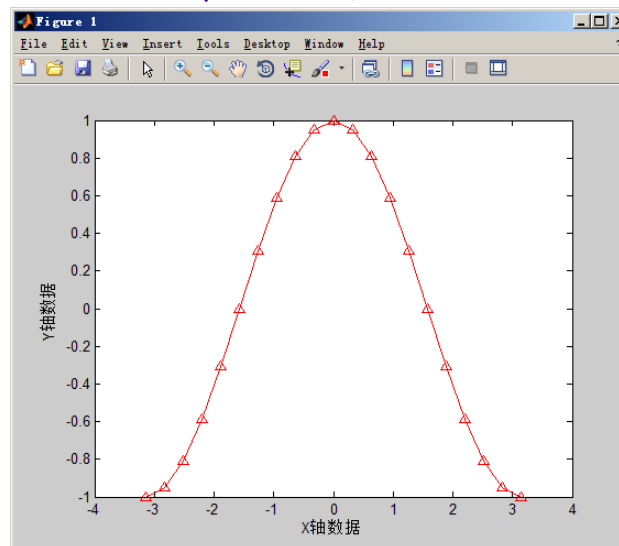
x = -pi:pi/10:pi;

y = cos(x);

plot(x,y,'-r^');

xlabel('X轴数据')

ylabel('Y轴数据')



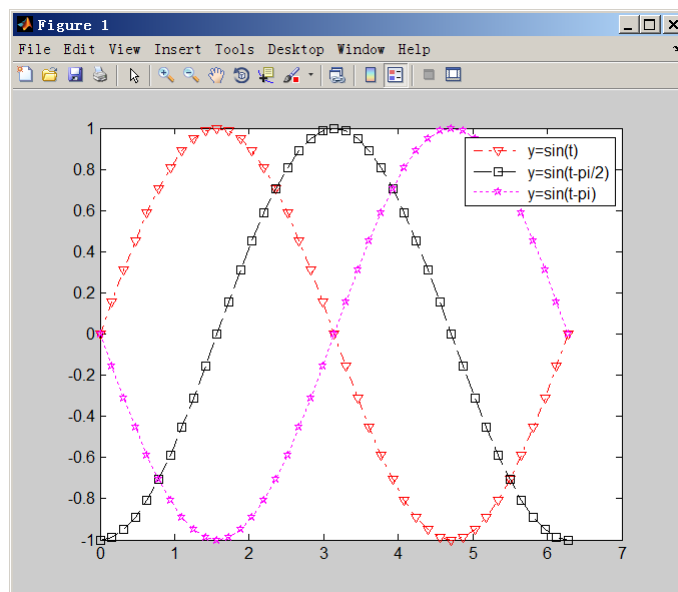
7.3.3 添加图例

添加轴的图例需要使用函数`legend`，基本调用格式为：

`legend('string1', 'string2'.....)`

- 字符串`string1`、`string2`为图例的说明性文本。
- 图例默认绘制在图形右上角处。

例：`legend('y=sin(t)', 'y=sin(t-pi/2)', 'y=sin(t-pi)')`





7.3.4 添加图形文本

text函数的基本调用格式为：

`text(x, y, 'string')`

- 字符串string为要添加的文本内容。
- x和y是文本注释添加的坐标值，该坐标值使用当前轴系的单位设置。

例如：

```
x = 0:0.1:2*pi;
```

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

```
plot(x,exp(-x),'r:');
```

```
text(pi/3,sin(pi/3), '<--Sin(\pi/3)')
```

7.3.5 图形标注属性设置

通过图形标注属性参数控制字体，包括：

FontName: 字体名称，例如新罗马、隶书等。

FontSize: 字体大小，整数值，默认为10磅。

FontWeight: 设置字体的加粗属性。

FontUnits: 字体大小的度量单位，默认为磅。

例如：

```
x = 0:1:2*pi;
```

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

```
plot(x,exp(-x),'r:');
```

```
text(pi/3,sin(pi/3), '<--Sin(\pi/3)', 'FontSize',12)
```



7.3.5 图形标注属性设置

```
x = 0:.1:2*pi;
```

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

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

```
grid on
```

```
hold on
```

```
plot(x,exp(-x),'r:*');
```

```
title('2-D Plots','FontName','Arial','FontSize',16)
```

```
xlabel('时间','FontName','隶书','FontSize',16)
```

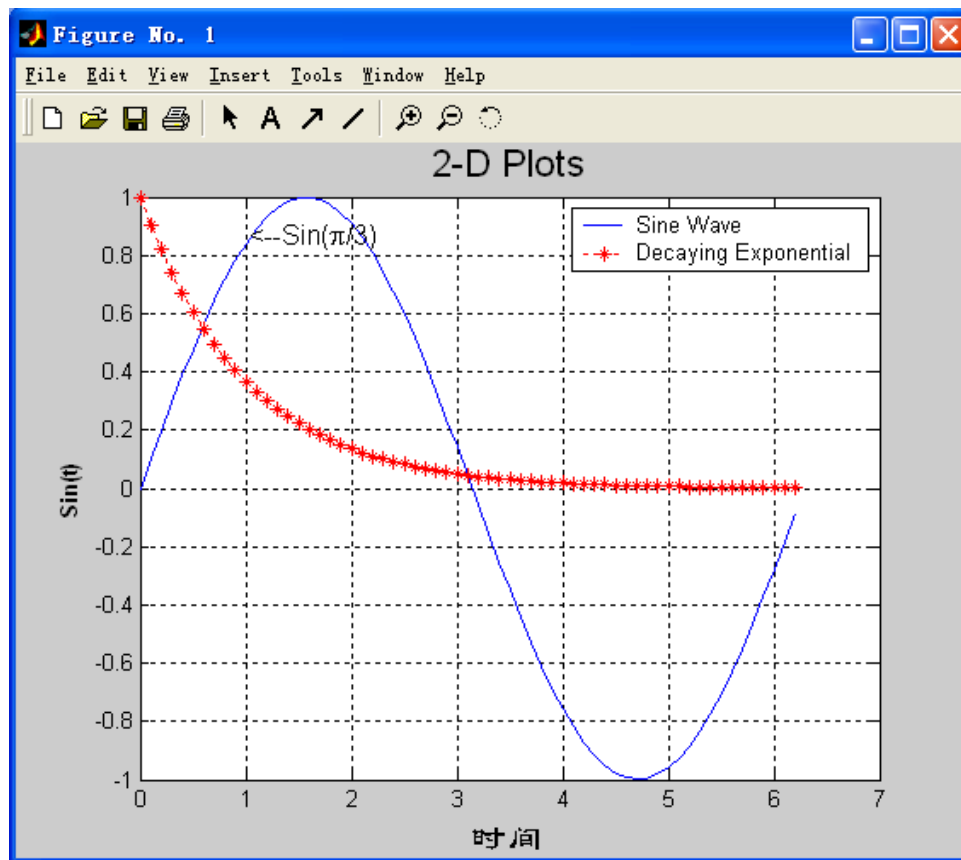
```
ylabel('Sin(t)','FontWeight','Bold')
```

```
text(pi/3,sin(pi/3), '<--Sin(\pi/3)','FontSize',12)
```

```
legend('Sine Wave', 'Decaying Exponential')
```

```
hold off
```

7.3.5 图形标注属性设置



设置的字体大小或者字体名称会一直对控制符后面的文本内容起作用，直到新字体控制符出现。

7.3.6 图形标注属性设置(特殊控制)

■ 精细指令形式

■ 字体样式设置:

\fontname{arg} \arg \fontsize {arg} string

其中，String为要输出的字符串，其前面的均为属性控制，使用方法见下表。

	指令	arg 取值	示例
指定字体	\fontname {arg}	Arial 宋体	'\fontname ('宋体')
指定风格	\arg	Bf (黑体) It 斜体 1 Sl 斜体 2 Rm 正体	'\bf example'
指定大小	\fontsize {arg}	正整数 (缺省 10P)	'\fontsize {16} Example1.'

7.3.6 图形标注属性设置(特殊控制)

上下角标的控制

	指令	Arg 取值	举例	
			示例指令	效果
上标	<code>^{\arg}</code>	任何合法字符	<code>'\exp^{(-t)}\sin(t)'</code>	$e^{-t} \sin(t)$
下标	<code>_{\arg}</code>	任何合法字符	<code>'U_{\alpha}'</code>	U_{α}

希腊字母与特殊字符

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
<code>\alpha</code>	α	<code>\upsilon</code>	υ	<code>\sim</code>	\sim
<code>\beta</code>	β	<code>\phi</code>	ϕ	<code>\leq</code>	\leq
<code>\gamma</code>	γ	<code>\chi</code>	χ	<code>\infty</code>	∞
<code>\delta</code>	δ	<code>\psi</code>	ψ	<code>\clubsuit</code>	\clubsuit
<code>\epsilon</code>	ϵ	<code>\omega</code>	ω	<code>\diamondsuit</code>	\diamondsuit
<code>\zeta</code>	ζ	<code>\Gamma</code>	Γ	<code>\heartsuit</code>	\heartsuit



例:绘制 $y = 50 \cdot \exp(-0.5 \cdot t) \cdot \sin(3 \cdot t)$

```
t = 0:.01:10;
```

```
y = 50*exp(-0.5*t).*sin(3*t);
```

```
plot(t,y);
```

```
title('\fontname{隶书}\fontsize{16}{隶书}  
      \fontname{Impact}{Impact}')
```

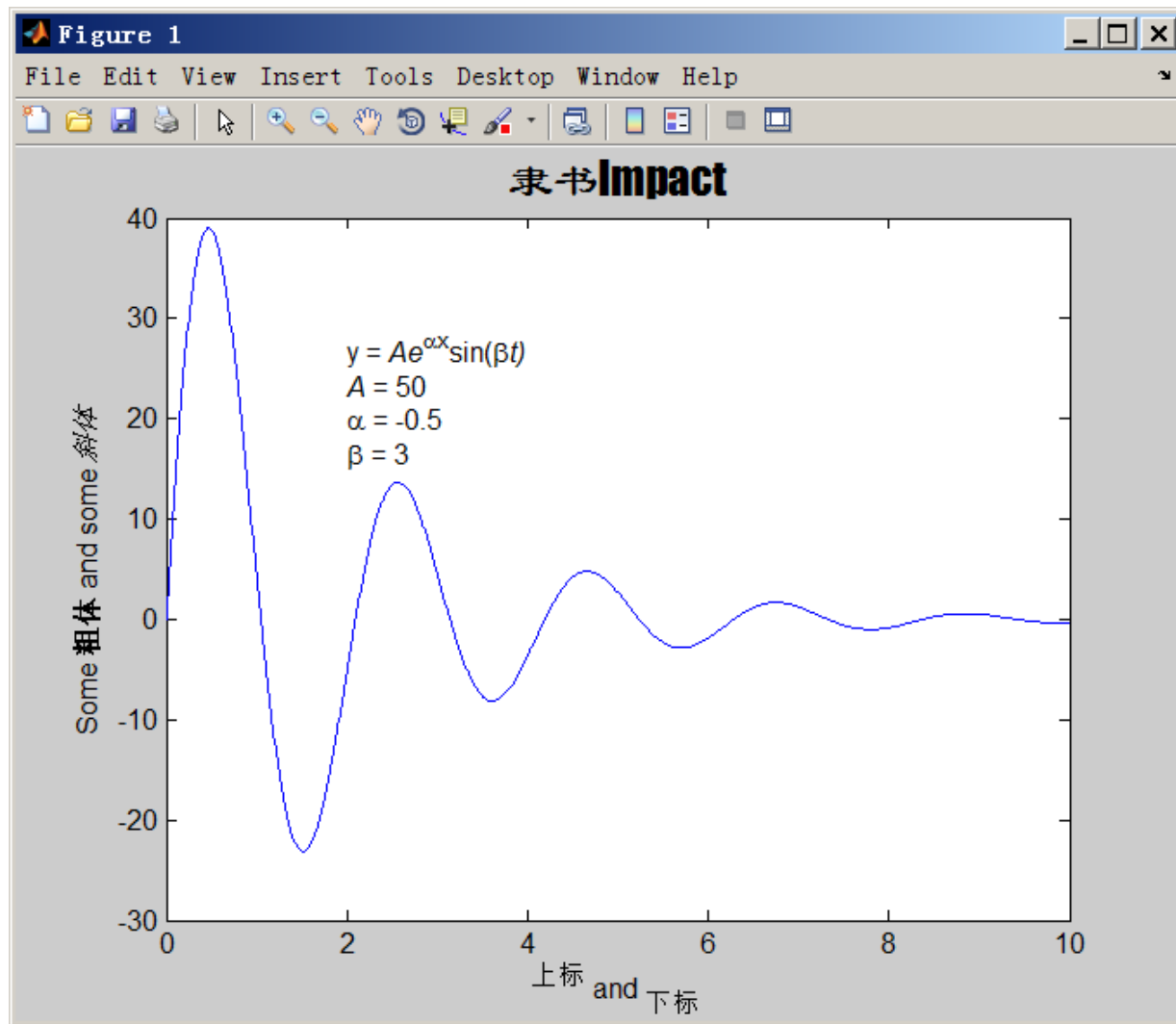
```
xlabel('^{上标} and _{下标}')
```

```
ylabel('Some \bf{粗体}\rm and some \it{斜体}')
```

```
txt = {'y = {\itAe}^{\alphax}sin(\beta\itt)',...  
      ['\itA\rm', ' = ', num2str(A)],...  
      ['\alpha = ', num2str(alpha)],...  
      ['\beta = ', num2str(beta)]]};
```

```
text(2,22,txt );
```


例: 绘制 $y = 50 * \exp(-0.5 * t) .* \sin(3 * t)$

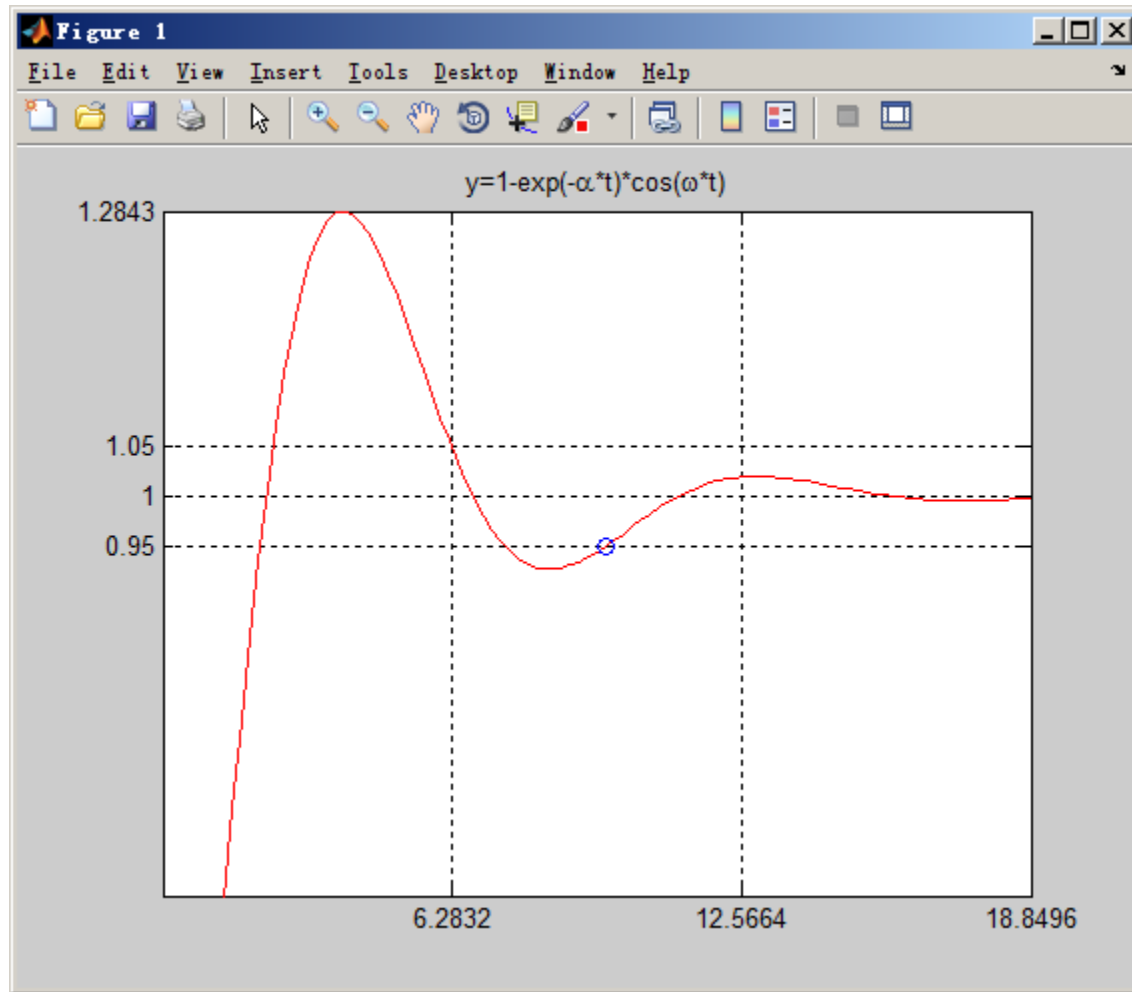




例：绘制 $y=1-\exp(-0.3*t).*\cos(0.7*t)$

```
t=6*pi*(0:100)/100;  
y=1-exp(-0.3*t).*cos(0.7*t);  
tt=t(find(abs(y-1)>0.05));  
ts=max(tt);  
plot(t,y,'r-');  
grid on;  
axis([0,6*pi,0.6,max(y)]);  
title('y=1-exp(-\alpha*t)*cos(\omega*t)');  
hold on;  
plot(ts,0.95,'bo');  
hold off;  
set(gca,'xtick',[2*pi,4*pi,6*pi],'ytick',[0.95,1,1.05,max(y)]);  
grid on;
```

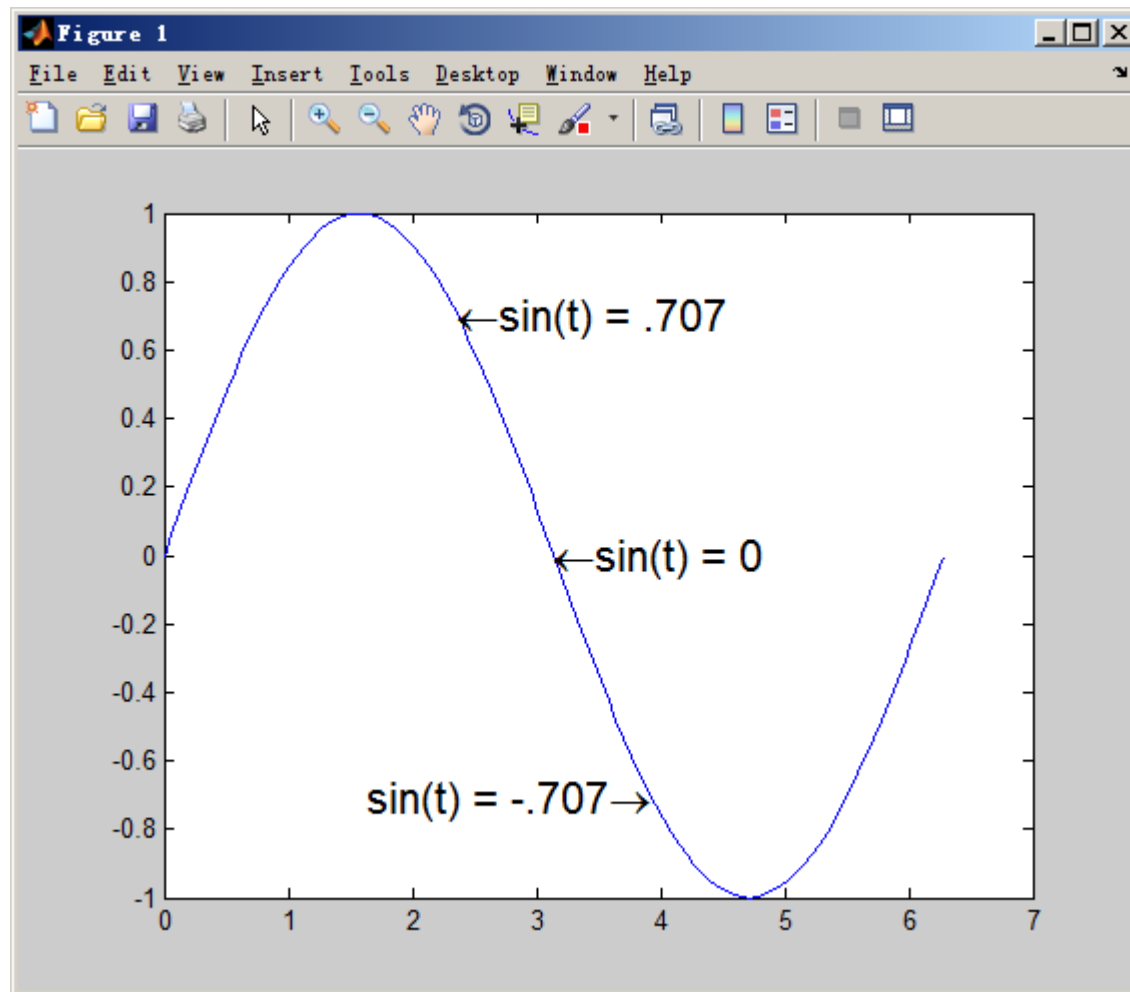
例：绘制 $y=1-\exp(-0.3*t).*\cos(0.7*t)$



例：在图形曲线上标注特殊值

```
t=(0:100)/100*2*pi;  
y=sin(t);  
plot(t, y)  
text(3*pi/4, sin(3*pi/4), '\fontsize{16}\leftarrowsin(t)  
= .707 ')  
text(pi, sin(pi), '\fontsize{16}\leftarrowsin(t) = 0 ')  
text(5*pi/4, sin(5*pi/4), '\fontsize{16}sin(t) = -  
.707\rightarrow',...  
    'HorizontalAlignment','right')  
➤ 其中, 'HorizontalAlignment','right'设置图形标识为  
水平右对齐
```

例：在图形曲线上标注特殊值



例：在图形曲线上标注特殊值(续)

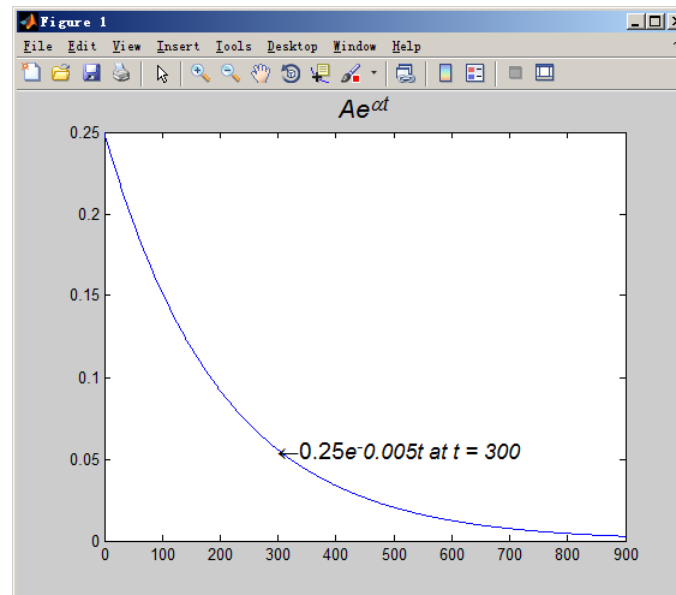
```
t = 0:900;
```

```
plot(t,0.25*exp(-0.005*t))
```

```
title('\fontsize{16}\it Ae^{\alphi}');
```

```
text(300,.25*exp(-0.005*300),...
```

```
'\fontsize{14}\leftarrow 0.25\ite^{-0.005\itt at \itt =  
300');
```



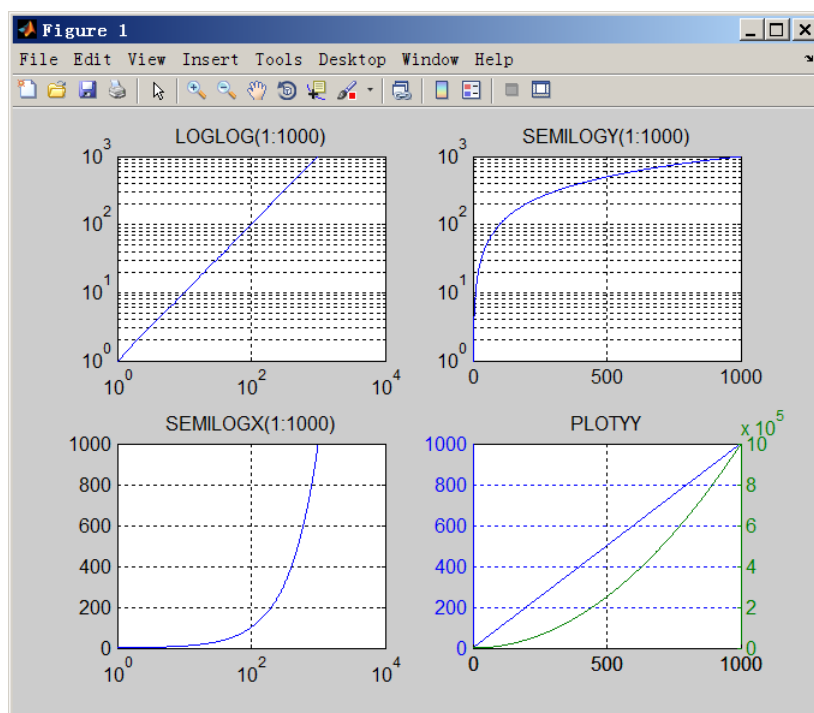
7.4 特殊图形

loglog: 两个坐标轴都使用对数刻度。

semilogx: x轴用对数刻度, y轴用普通线性刻度。

semilogy: y轴用对数刻度, x轴用普通线性刻度。

plotyy: 双y轴绘图。



例:plotyy绘制双纵坐标图

■ plotyy指令调用格式:

plotyy(x1, y1, x2, y2)

➤ **x1-y1曲线y轴在左， x2-y2曲线y轴在右。**

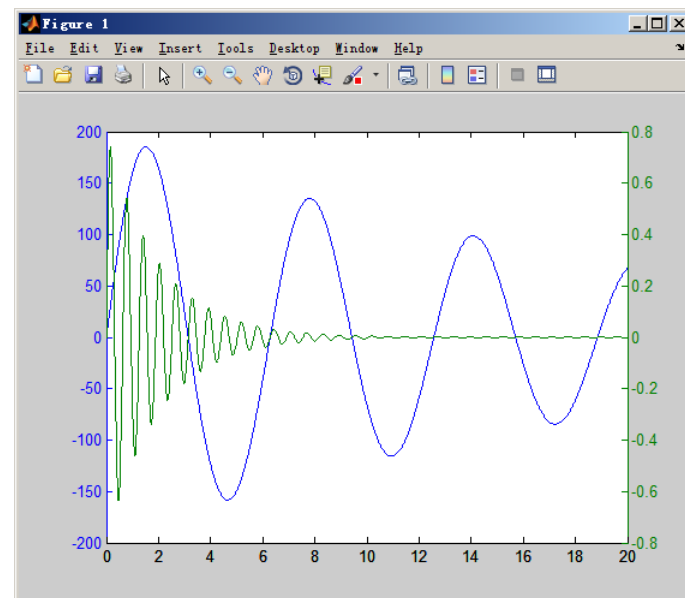
例如:

x = 0:0.01:20;

y1 = 200*exp(-0.05*x).*sin(x);

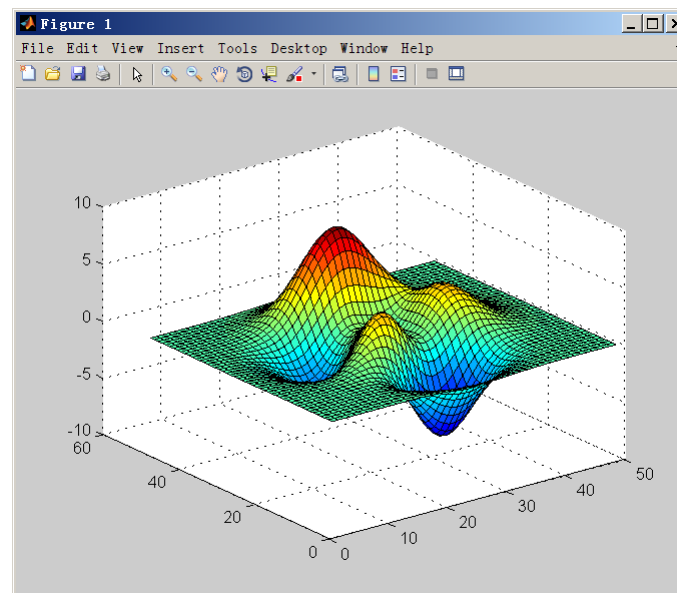
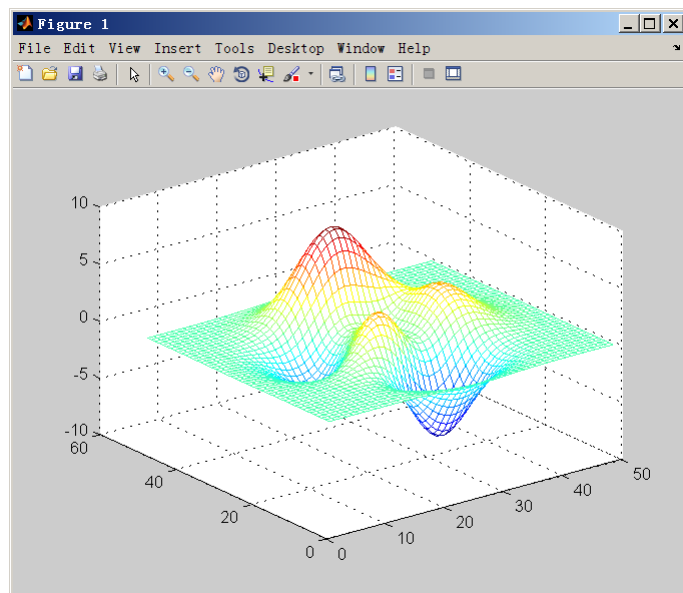
y2 = 0.8*exp(-0.5*x).*sin(10*x);

plotyy(x,y1,x,y2);



7.5 三维绘图

函数名	说明
<code>plot3(x,y,z)</code>	绘制三维曲线
<code>mesh(z)</code> 或 <code>mesh(x,y,z)</code> <code>surf(z)</code> 或 <code>surf(x,y,z)</code>	绘制三维曲面图





7.5 三维绘图

■ meshgrid 函数

用于生成二维或者三维网格矩阵。

➤ 基本调用格式:

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

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

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

其中x , y和z是一维数组

7.5 三维绘图

坐标网格化

$$z=f(x,y)=x^2+y^2$$

$$x=[1 \ 2 \ 3]; \ y=[4 \ 5 \ 6];$$

生成z轴数据

$$f(1,4), f(2,4), f(3,4);$$

$$f(1,5), f(2,5), f(3,5);$$

$$f(1,6), f(2,6), f(3,6);$$

$$[X \ Y]=\text{meshgrid}(x,y)$$

$$X =$$

$$\begin{matrix} 1 & 2 & 3 \end{matrix}$$

$$\begin{matrix} 1 & 2 & 3 \end{matrix}$$

$$\begin{matrix} 1 & 2 & 3 \end{matrix}$$

$$Y =$$

$$\begin{matrix} 4 & 4 & 4 \end{matrix}$$

$$\begin{matrix} 5 & 5 & 5 \end{matrix}$$

$$\begin{matrix} 6 & 6 & 6 \end{matrix}$$

$$Z=X.^2+Y.^2$$

7.5 三维绘图

■ `mesh(x,y,z)`

➤ 如果 x 、 y 为向量 $\text{length}(X) = n$, $\text{length}(Y) = m$, $\text{size}(Z) = [m,n]$ 。网格节点坐标为 $(X(j), Y(i), Z(i,j))$

➤ 如果 x 、 y 、 z 为 m 行, n 列矩阵。网格节点坐标为 $(X(i), Y(j), Z(i,j))$

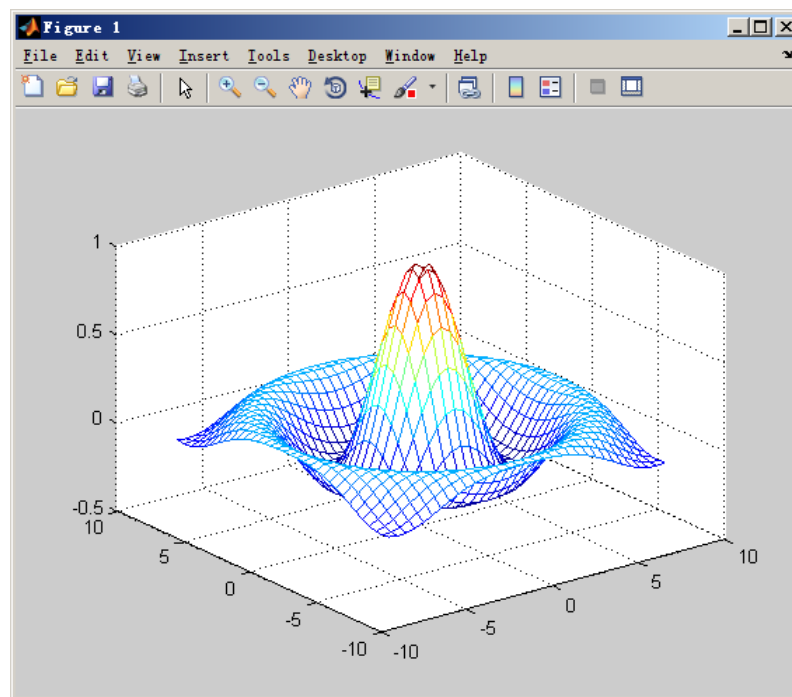
例如:

```
[X,Y] = meshgrid(-8:0.5:8);
```

```
R = sqrt(X.^2 + Y.^2);
```

```
Z = sin(R)./R;
```

```
mesh(X,Y,Z) %mesh(x,y,Z)
```



7.5 三维绘图

■ **mesh(z)**, 其中 z 为 m 行, n 列矩阵

➤ x 坐标的取值从1取到 n , 间距为1, y 坐标的取值是从1取到 m 构成的网格节点, 曲面的高度就是 z 中各矩阵元素的值。

例如:

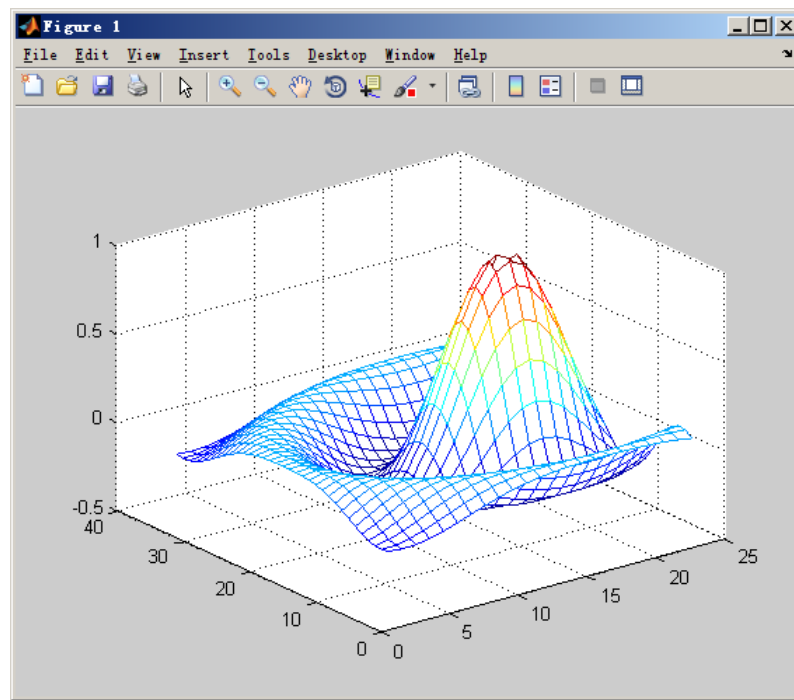
```
[X,Y] = meshgrid(-8:0.5:8);
```

```
R = sqrt(X.^2 + Y.^2);
```

```
Z = sin(R)./R;
```

```
Z1=Z(:,1:end-10);
```

```
mesh(Z1)
```



7.6.1 ezplot绘制二维图

函数**ezplot**无需数据准备，可以直接画出函数的图形。

➤ 基本调用格式：

ezplot (<隐函数表达式>,[x最小值, x最大值, y最小值, y最大值])

➤ **ezplot(f)** %f是关于x的函数

例如： **ezplot('sin(x)')**

➤ **ezplot(f(x,y))** %f(x,y)=0,隐函数

例如： **ezplot('x^3+y^3-3*x*y')**

➤ **ezplot(f1,f2)** %f1、f2是关于变量t的参数方程

ezplot('3*sin(t)','3*cos(t)')

7.6.2 ezsurf绘制三维图

ezsurf用于创建由 $f(x,y)$ 确定的曲面图， f 为关于 x 和 y 两个变量的函数的字符串。

➤ 基本调用格式:

ezsurf(f,n)

➤ 默认区域是 $-2\pi < x < 2\pi$ 、 $-2\pi < y < 2\pi$

➤ n 表示绘图网格为 $n*n$

例如:

```
fh = @(x,y) x.*exp(-x.^2-y.^2);  
ezmesh(fh,40)
```

