第 5 章 MATLAB的科学可视化

CHAPTER 5

MATLAB Scientific Visualization

图形绘制与可视化是MATLAB语言的一大特色。MATLAB提供了一系列直观、简单的二维、三维图形绘制命令与函数，可以将实验结果和仿真结果用可视的形式显示出来。

从R2014b版本开始MATLAB提供全新的绘图体系与绘图函数，虽然早期版本的一些命令仍能使用，但在以后的版本中这些命令可能会被逐渐淘汰，所以本书将尽量按照新的体例介绍图形绘制的方法。

5.1节介绍简单二维曲线的绘制方法，包括由数据绘制曲线的方法和由数学表达式绘制曲线的方法，还介绍二维曲线的修饰方法。5.2节介绍特殊二维曲线的绘制方法，如极坐标曲线的绘制方法、离散点的表示方法、统计图的绘制方法及动态图形的绘制方法。5.3节介绍三维图形的绘制方法，包括三维曲线图与曲面图的绘制方法、视角设置方法与三维动态图形的处理方法。5.4节介绍二元与三元隐函数图形的绘制方法。

5.1 简单二维图形

二维图形是科学研究中最常见，也是最实用的图形表示。本节首先介绍将数据用二维曲线表示出来的方法，然后介绍将数学函数用曲线表示出来的方法。

5.1.1 基于数据的绘图

假设用户已经获得了一些实验数据。例如，已知各个时刻$t=t_1,t_2,\cdots,t_n$，测得这些时刻的函数值$y=y_1,y_2,\cdots,y_n$，则可以构成向量$\boldsymbol{t}$=[t_1,t_2,$\cdots$,t_n]和$\boldsymbol{y}$=[y_1,y_2,$\cdots$,y_n]，将数据输入MATLAB环境中。如果用户想用图形表示二者之间的关系，则用`plot(t,y)`命令即可绘制二维图形。可以看出，该函数的调用是相当直观的。

例5-1 试绘制出显函数$y=\sin(\tan x)-\tan(\sin x)$在$x\in[-\pi,\pi]$区间内的曲线。

解 解决这种问题的最简捷方法是采用下面的语句直接绘制函数曲线。

```
>> x=-pi:0.05:pi;                          %以0.05为步距构造自变量向量
   y=sin(tan(x))-tan(sin(x)); plot(x,y)    %求出并绘制各个点上的函数值
```

这些语句可以绘制出该函数的曲线，如图5-1所示。不过由这里给出的曲线看，得出的曲线似乎有问题。

值得指出的是，由MATLAB的`plot()`函数绘制出的“曲线”不是真正的曲线，只是给

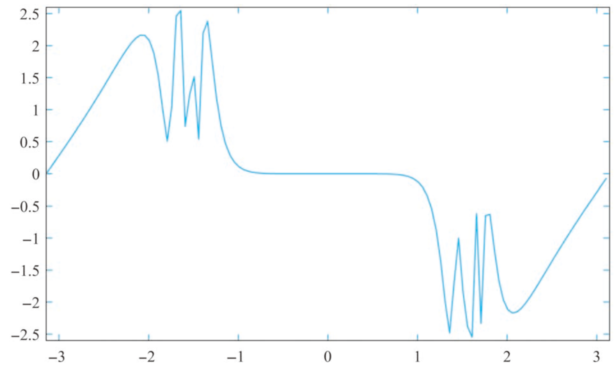


图5-1 给定函数的曲线表示

出各个数值点间的折线。如果给出的数据点足够密,或突变少些,则看起来就是曲线了,故以后将其称为曲线。

例5-2 试重新生成密集些的数据点,得出例5-1函数正确的曲线表示。

解 仔细观察图5-1中给出的曲线可以看出,在$\pm\pi/2$附近图形好像有问题,其他位置还是比较平滑的。为什么会出现这样的现象呢?观察$\sin(\tan x)$项,由于括号内的部分在$\pm\pi/2$附近将趋近于无穷大,因而,其正弦值变化很不规则,会出现强振荡。

可以考虑全程采用小步距,或在比较粗糙的$x\in(-1.8,-1.2)$及$x\in(1.2,1.8)$两个子区间内选择小步距,其他区域保持现有的步距,这样可以将上述的语句修改为

```
>> x=[-pi:0.05:-1.8,-1.799:0.0001:-1.2, -1.2:0.05:1.2,...
       1.201:0.0001:1.8, 1.81:0.05:pi];  %以变步距方式构造自变量向量
   y=sin(tan(x))-tan(sin(x)); plot(x,y)  %求出并绘制各个点上的函数值
```

这样将得出如图5-2所示的曲线。可见,这样得出的曲线在剧烈变化区域内表现良好。前面解释过,在$\pm\pi/2$附近出现强振荡是正常现象。

如果全程都选择0.0001这样的小步距,也能得出看起来完全一致的效果。

```
>> x=-pi:0.0001:pi;                      %以0.0001为步距构造自变量向量
   y=sin(tan(x))-tan(sin(x)); plot(x,y)  %求出并绘制各个点上的函数值
```

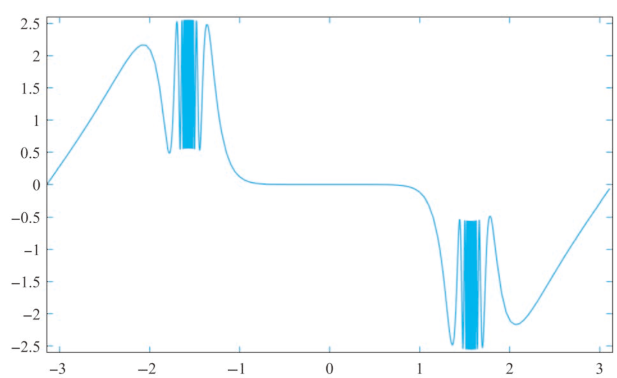


图5-2 改变步距后的曲线

从这个例子可以看出,不能过分地依赖MATLAB绘制的曲线,需要对曲线的正确性做检验。比较有效的方法是选择不同的步距,观察得出的曲线是不是吻合,如果吻合则可以认

为曲线正确，否则需要选择更小的步距绘制曲线后再进行检验，直至得出吻合的结果。

在实际应用中，`plot()`函数的调用格式还可以进一步扩展。

（1）$\boldsymbol{t}$仍为向量，而$\boldsymbol{y}$为如下的矩阵，则将在同一坐标系下绘制m条曲线，每一行和$\boldsymbol{t}$之间的关系将绘制出一条曲线。注意，这时要求矩阵$\boldsymbol{y}$的列数应该等于$\boldsymbol{t}$的长度。

$$\boldsymbol{y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix}$$

（2）$\boldsymbol{t}$和$\boldsymbol{y}$均为矩阵，且假设矩阵$\boldsymbol{t}$和$\boldsymbol{y}$的行数和列数均相同，则可绘制出矩阵$\boldsymbol{t}$每行和矩阵$\boldsymbol{y}$对应行之间关系的曲线。

（3）假设有多对这样的向量或矩阵$(\boldsymbol{t}_1, \boldsymbol{y}_1), (\boldsymbol{t}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{t}_m, \boldsymbol{y}_m)$，则可以用下面的语句直接绘制出各自对应的曲线：

plot($\boldsymbol{t}_1$,$\boldsymbol{y}_1$,$\boldsymbol{t}_2$,$\boldsymbol{y}_2$,$\cdots$,$\boldsymbol{t}_m$,$\boldsymbol{y}_m$)

（4）曲线的性质，如线型、粗细、颜色等，还可以使用下面的命令进行指定：

plot($\boldsymbol{t}_1$,$\boldsymbol{y}_1$,选项1,$\boldsymbol{t}_2$,$\boldsymbol{y}_2$,选项2,$\cdots$,$\boldsymbol{t}_m$,$\boldsymbol{y}_m$,选项m)

其中“选项”可以按表5-1中说明的形式给出，不同的选项可以进行组合。例如，若想绘制红色的点画线，且每个转折点上用五角星表示，则可以使用组合字符串`'r-.pentagram'`。

表5-1 MATLAB绘图命令的各种选项

曲线线型		曲线颜色				标记符号			
选 项	意 义	选 项	意 义	选 项	意 义	选 项	意 义	选 项	意 义
'-'	实线	'b'	蓝色	'c'	蓝绿色	'*'	星号	'pentagram'	☆
'--'	虚线	'g'	绿色	'k'	黑色	'.'	点号	'o'	圆圈
':'	点线	'm'	红紫色	'r'	红色	'x'	叉号	'square'	□
'-.'	点画线	'w'	白色	'y'	黄色	'v'	▽	'diamond'	◇
'none'	无线					'^'	△	'hexagram'	✡
						'>'	▷	'<'	◁

（5）除了表5-1给出的简洁绘图参数方法，新版本中的`plot()`函数还允许以

plot(~,参数名1,参数值1,参数名2,参数值2,$\cdots$)

的形式指定绘图参数，其中，~表示上述的正常调用格式。常用的参数名与参数值在表5-2中列出。

（6）还可以由h=plot(~)格式调用`plot()`函数，绘图的同时返回曲线的句柄h，以后可以通过该句柄读取或修改该曲线的属性。

（7）如果想在句柄为h的坐标系下绘制曲线，还可以给出plot(h,$\boldsymbol{t}$,$\boldsymbol{y}$,$\cdots$)命令。

例5-3 分形树的数学模型（习题3.21）：任意选定一个二维平面上的初始点坐标(x_0, y_0)，假设可以生成一个在$[0,1]$区间上均匀分布的随机数γ_i，那么根据其取值的大小，可以按下面的公式

表5-2 常用的绘图控制参数

参数名	参数值
`LineSpec`	曲线线型控制字符串,如`'r-.pentagram'`字符串,可以参考表5-1
`LineWidth`	曲线的线宽,默认的宽度是0.5pt,其中1pt=0.3527mm
`MeshDensity`	自动步距计算点的密度,默认值为23,增大该值可以提高曲线精度,主要用于后面将介绍的`fimplicit()`和`fplot()`等基于函数表达式的曲线绘制函数
`Color`	曲线颜色,除了表5-1指定的8种颜色,还可以设定为RGB分量$[r,g,b]$
`MarkerEdgeColor`	标记的边缘颜色,事实上就是标记自身的颜色
`MarkerSize`	标记的大小,默认值6pt

生成一个新的坐标点(x_1,y_1)。

$$(x_1,y_1)\Leftarrow\begin{cases} x_1=0, & y_1=y_0/2, & \gamma_i<0.05\\ x_1=0.42(x_0-y_0), & y_1=0.2+0.42(x_0+y_0), & 0.05\leqslant\gamma_i<0.45\\ x_1=0.42(x_0+y_0), & y_1=0.2-0.42(x_0-y_0), & 0.45\leqslant\gamma_i<0.85\\ x_1=0.1x_0, & y_1=0.2+0.1y_0, & \text{其他}\end{cases}$$

试递推生成10000个坐标点,并用圆点绘制出分形树的结果。

解 分形树的数据可以由下面给出的语句直接计算。有了这些语句就可以得到向量$\boldsymbol{x}$与$\boldsymbol{y}$,这样,由下面的命令可以直接绘制出如图5-3所示的分形树图。注意,在这里给出的迭代公式中,当前点x、y取决于上一步的x、y值,所以这里只能采用循环结构,不能采用向量化运算。

```
>> v=rand(10000,1); N=length(v); x=0; y=0;
   for k=2:N, gam=v(k);              %针对每个γi值,更新x、y
      if gam<0.05, x(k)=0; y(k)=0.5*y(k-1);
      elseif gam<0.45
         x(k)=0.42*(x(k-1)-y(k-1)); y(k)=0.2+0.42*(x(k-1)+y(k-1));
      elseif gam<0.85
         x(k)=0.42*(x(k-1)+y(k-1)); y(k)=0.2-0.42*(x(k-1)-y(k-1));
      else, x(k)=0.1*x(k-1); y(k)=0.1*y(k-1)+0.2;
   end, end
   plot(x,y,'.','MarkerSize',5)  %注意,不能忽略最后的圆点标记
```

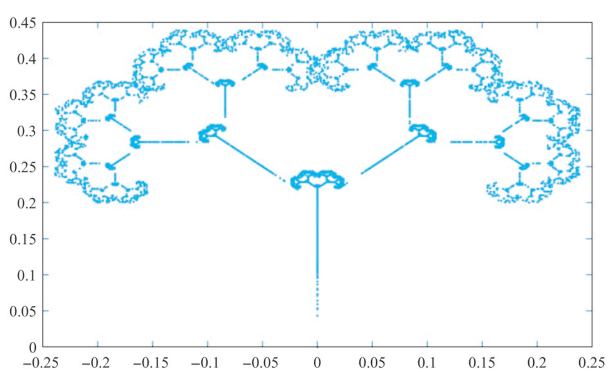


图5-3 分形树的显示

5.1.2　基于函数表达式的绘图

如果已知数学函数，还可以利用 fplot() 函数绘制函数曲线。其调用格式为 fplot(f)，其中，f 可以是匿名函数描述的函数句柄，也可以是描述函数的符号表达式或符号函数，默认的绘图区间为 $[-5,5]$。如果想指定绘图区域，还可以给出 fplot(f,[$x_{\rm m}$,$x_{\rm M}$])。注意，在匿名函数中需要使用点运算描述数学函数，因为传入的自变量 $\boldsymbol{x}$ 可以为向量。符号运算无需点运算，因为符号变量 x 是标量。

MATLAB 早期版本提供的 ezplot() 函数也可以用于绘制类似的曲线，默认的绘图区间为 $[-2\pi,2\pi]$。该函数中的数学公式描述方式与 fplot() 函数是不同的，这里不过多讨论该函数的使用方法。

例 5-4　试用 fplot() 函数重新绘制例 5-1 的函数曲线。

解　可以考虑用符号函数描述原来的数学函数，给出下面的语句就可以绘制出函数的二维曲线，如图 5-4 所示。从效果上看，与自选数据点得出的曲线基本一致。该曲线还自动绘制了 $x=-\pi/2$ 处的虚线。

```
>> syms x; f(x)=sin(tan(x))-tan(sin(x)); fplot(f,[-pi,pi])
```

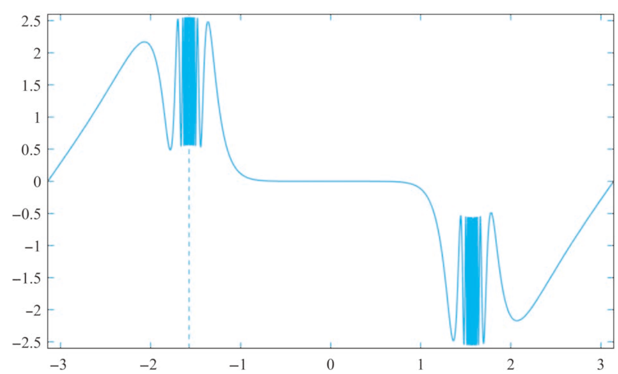


图 5-4　由 fplot() 函数绘制的函数曲线

还可以用匿名函数的形式描述原函数，得出一致的结果。用匿名函数描述已知数学函数时，应该采用点运算。对本例而言，sin() 与 tan() 函数的作用与点运算一致，无须特殊处理。

```
>> f=@(x)sin(tan(x))-tan(sin(x)); fplot(f,[-pi,pi])
```

类似于 plot() 函数，fplot() 函数也可以支持不同的调用格式。例如，若想在同一坐标系内绘制多个函数，则可以给出命令 fplot([f_1,f_2,$\cdots$,f_m])，其中，f_i 为第 i 个数学函数的函数句柄或符号表达式。此外，该函数允许带有不同的选项，可以返回图形句柄等。如果这些函数句柄数据结构不同，或含有匿名函数的句柄，则在调用 fplot() 时不能使用方括号，只能使用花括号，否则将给出错误信息。当然，不管哪种场合，都可以统一使用花括号。

例 5-5　考虑正弦函数 $f(t)=\sin x$。由高等数学课程中学习的 Taylor 级数展开可以得出函数的有限项逼近表达式为 $y(t)=x^9/362880-x^7/5040+x^5/120-x^3/6+x$。试在同一坐标系下绘制在 $x\in[-4,4]$ 区间内两个函数的曲线，并评价函数逼近的效果。

解　可以用符号表达式直接表示原函数与 Taylor 级数表达式，然后调用 fplot() 函数，同时绘

制两条曲线，如图5-5所示。为方便起见，这里用手工的方法将Taylor级数曲线设置为虚线，具体的设置方法将在后面介绍。

```
>> syms x; f=sin(x); y=x^9/362880-x^7/5040+x^5/120-x^3/6+x;
   fplot([f,y],[-4 4])  %同时绘制两个函数的曲线
```

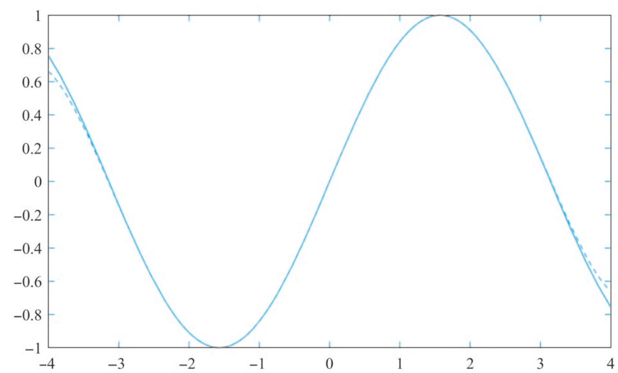


图5-5 函数的Taylor级数逼近对比研究

如果想用匿名函数表示数学函数，应该使用点运算。

```
>> f=@(x)sin(x); y=@(x)x.^9/362880-x.^7/5040+x.^5/120-x.^3/6+x;
   fplot({f,y})    %注意，这里只能使用花括号
```

5.1.3 参数方程曲线绘制

如果某函数由参数方程（parametric equation）给出

$$x = x(t),\ \ y = y(t),\ \ t_{\mathrm{m}} \leqslant t \leqslant t_{\mathrm{M}} \tag{5-1}$$

且$x(t)$由函数句柄h_x表示，$y(t)$由句柄h_y表示，其中，函数句柄可以同时为匿名函数，或同时为符号表达式，则可以由`fplot(`h_x`,`h_y`,[`t_{m}`,`t_{M}`])`命令绘制其轨迹曲线。

例5-6 Lissajous曲线是由两个不同频率正弦函数构成的参数方程，试绘制出$x(t) = \sin t$，$y = \sin 1.25t, t \in [0,30]$的函数曲线。

解 由符号表达式描述参数方程，并绘制出Lissajous函数曲线，如图5-6所示。

```
>> syms t; x=sin(t); y=sin(1.25*t); fplot(x,y,[0,30])
```

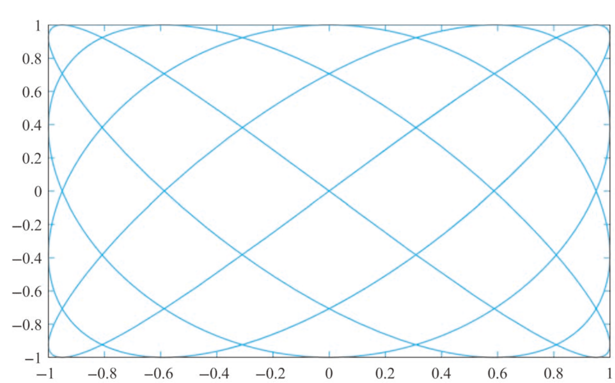


图5-6 Lissajous曲线

默认状态下的`fplot()`函数自动选择曲线绘制的参数，例如，给出 $[t_{\rm m}, t_{\rm M}]$ 时自动选择时间步距，有时可能得出完全错误的结果。这时，用户可以自行选择`Meshdensity`等控制参数，增大其值，直至得出正确的结果。

例5-7 重新考虑例5-6给出的Lissajous曲线。如果 $t \in [0, 1000]$ 可能会得出错误的曲线，如何绘制正确的曲线？

解 直接给出下面的命令，则可能绘制出如图5-7所示的错误曲线。

```
>> syms t; x=sin(t); y=sin(1.25*t); fplot(x,y,[0,1000])
```

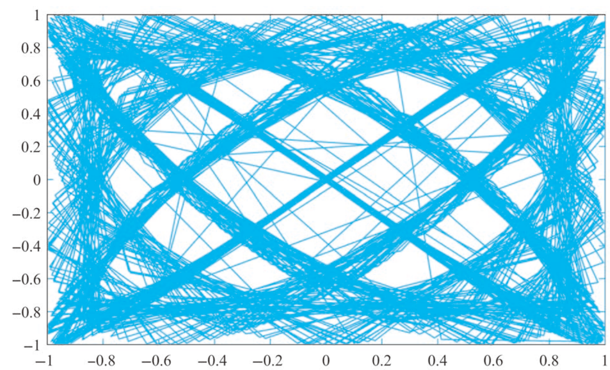


图5-7 错误的Lissajous曲线

这时，需要人为指定`MeshDensity`选项的值，例如将其设置成300或更大的值，这样得出的结果与图5-6中给出的完全一致，因为该函数是周期函数。

```
>> fplot(x,y,[0,1000],'MeshDensity',300)
```

和`plot()`函数类似，`fplot()`函数曲线自动绘制的结果同样也应该被检验。设置不同的`MeshDensity`参数是一种有效的检测方法。一般说来，较大的值对应着更精确的结果。选择不同的`MeshDensity`参数值，如果得出一致的曲线，则说明选择合理。

5.1.4 双 y 轴曲线

前面介绍的`plot()`函数可以在同一坐标系下同时绘制多条曲线，不过在某些特定的应用中，如果两条曲线的幅值差异过大，则可以为图形设置双 y 轴，分别表示不同的曲线。具体做法为：调用`yyaxis left`和`yyaxis right`命令设置坐标系，绘制所需的曲线。早期版本中的`plotyy()`函数在当前版本下仍可以使用，不过这里不推荐使用该函数。

例5-8 考虑两个函数 $y_1 = \sin x$ 与 $y_2 = 0.01\cos x$，试绘制它们的曲线。

解 当然可以考虑使用`plot()`函数直接绘制曲线，不过由于它们的幅值相差过于悬殊，y_2 曲线看起来就像一条直线，分辨率很差。所以，可以考虑给曲线设置两个纵轴，这样得出的曲线如图5-8所示。从图中可见，实线的标度在左侧坐标轴给出，虚线的标度在右侧坐标轴给出，用这样的方法可以很好地显示两条幅值相差悬殊的曲线。

```
>> x=0:0.01:2*pi; y1=sin(x); y2=0.01*cos(x);
   yyaxis left; plot(x,y1), yyaxis right; plot(x,y2,'--')
```

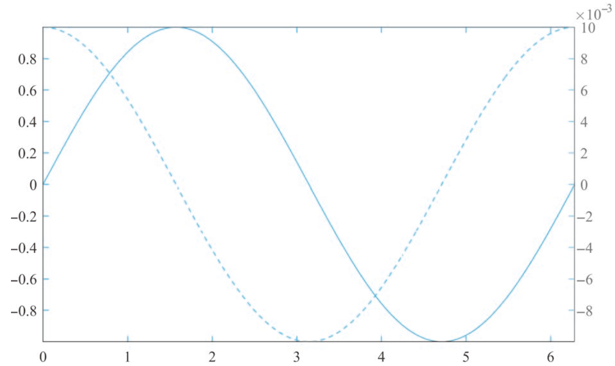


图 5-8　双纵轴的函数曲线

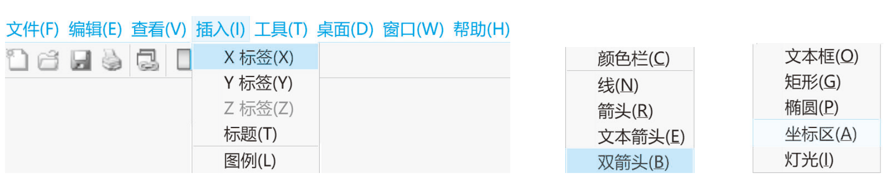
5.1.5　图形修饰与编辑

曲线绘制之后，还可以对绘制的图形进行进一步修饰。表 5-3 中列出了常用的图形修饰命令。用户可以利用这些命令在绘制的图形上做适当的修饰，也可以利用图形窗口提供的工具对图形进行处理。

表 5-3　常用的图形修饰命令

修饰命令	调用格式	命令的解释
`title()`	`title(str)`	给图形加标题，标题的内容由字符串 `str` 表示
`xlabel()`	`xlabel(str)`	给 x 轴加标签，而 `ylabel(str)` 命令给 y 轴加标签，并将标签旋转 90°
`text()`	text(x,y,str)	在图形的 (x,y) 坐标处添加文字说明
`gtext()`	`gtext(str)`	允许用户用光标选择添加文字说明的位置
`legend()`	legend(s_1,s_2,$\cdots$)	与图形对应的线型介绍图例，字符串 s_k 为第 k 条曲线的说明文字
`annotation`	annotation(s,x,y)	曲线加标注，其中，s 为标注类型，可以选择 `'arrow'`(箭头)、`'line'`(线段)、`'doublearrow'`(双向箭头)等，由向量 $[x_1,x_2]$ 和 $[y_1,y_2]$ 表示起点 (x_1,y_1) 和终点坐标 (x_2,y_2)
`hold`	`hold on`	可以用 `hold on` 或 `hold off` 锁定或释放坐标系。如果坐标系被锁定，再使用 `plot()` 这类命令将在现有的图形上叠印新的曲线；`hold off` 命令解除锁定状态。还可以由 `key=ishold` 命令查询坐标系的锁定状态，返回的值为 0 或 1
`zoom`	`zoom on`	局部放大功能，可以用光标选择想放大的区域；`zoom off` 命令可以取消局部放大功能；还可以使用 `zoom xon` 和 `zoom yon` 单独放大某坐标轴

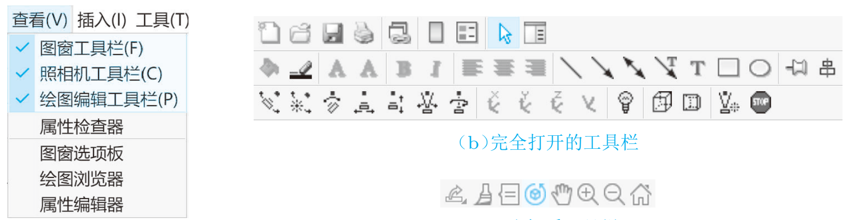
MATLAB 图形窗口的“插入”菜单的内容如图 5-9 所示。这里为排版方便将整个菜单截成三段给出。其中，图 5-9(a)还给出了图形窗口的主菜单。可见，表 5-3 中的大多数函数或命令可以由图形窗口的“插入”菜单直接实现。



(a) 菜单系统及“插入”菜单上段　　(b) 中段　　(c) 下段

图 5-9　MATLAB 图形窗口的“插入”菜单

MATLAB图形窗口的“查看”菜单在图5-10(a)中给出，其中，前面三项控制工具栏的样式。如果前三项都被选中，则图形窗口的工具栏如图5-10(b)所示。默认状态下，只显示第一行工具。该工具栏的按钮控制图形的编辑状态。如果选中，则可以用光标选择曲线单独编辑；如果反选，则取消编辑状态。取消编辑状态后，如果把光标移入某个坐标系内，则该坐标系的右上角出现如图5-10(c)所示的坐标系工具栏。



(a)“查看”菜单　(b)完全打开的工具栏　(c)坐标系工具栏

图5-10　MATLAB图形窗口的“查看”菜单与工具栏

5.1.6　图形数据的提取

MATLAB的菜单系统允许用户将当前的图形窗口直接存入文件，文件的后缀名为`fig`。如果想恢复图形，直接用菜单从`.fig`文件读入即可。如果想获得当前图形窗口的数据，或从打开的`.fig`文件获取数据，则需要进入图形编辑状态，用光标选中感兴趣的曲线。这时，由`gco`(get current object，获得当前对象句柄)命令即可获得选中曲线的句柄，再使用`get()`函数即可提取曲线数据。

```
x=get(gco,'xData');  y=get(gco,'yData');
```

MATLAB提供了一系列类似于`gco`的命令，如`gcf`(获得当前窗口句柄)、`gca`(获得当前坐标系句柄)。此外可以用`clf`命令清空当前的图形窗口。

5.2　特殊二维图形

除了前面介绍的`plot()`函数与`fplot()`函数，还可以采用`line()`函数绘制曲线，其作用与`plot()`函数相仿。当前版本的`line()`函数支持的格式为`line(x,y,属性名,属性值)`，调用格式与`plot()`函数类似。不同的是，`line()`函数在当前坐标系下叠印曲线。

此外，MATLAB还支持其他的二维图形绘制命令，常用的特殊曲线绘制命令与调用格式在表5-4中给出。本节将介绍一些常用二维图形的绘制方法。

5.2.1　极坐标

极坐标是以一个给定的点为原点，以从原点出发的某条线为极轴构造的坐标系。空间上某一点到原点的距离为ρ，该点与原点的连线和极轴的夹角为θ，该角度以从极轴出发逆时针方向的转角为正方向。这样的有序对(ρ,θ)称为极坐标。极坐标下的曲线一般可以表示为显式函数$\rho=\rho(\theta)$，称为极坐标方程。传统意义下，极坐标方程中的$\rho\geqslant 0$，如果将其拓展到实数空间，再通过直角坐标系变换得出曲线所对应的方程，则该方程可以被理解为广义极坐标方程。

表5-4　MATLAB提供的特殊二维曲线绘制函数

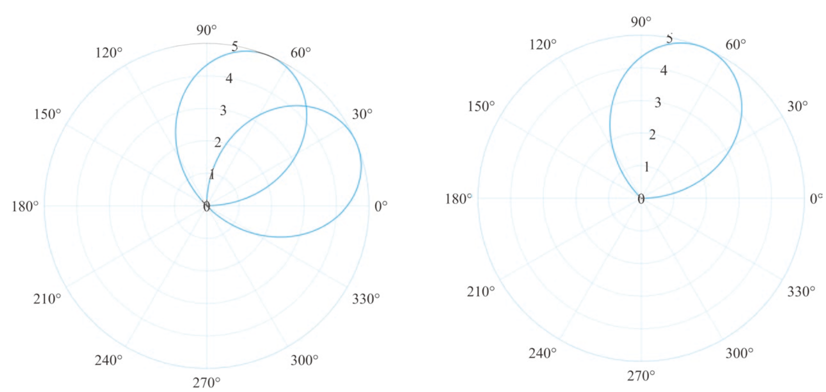
函数名	意　义	常用调用格式	函数名	意　义	常用调用格式
comet()	彗星轨迹图	comet(x,y)	bar()	二维条形图	bar(x,y)
compass()	罗盘图	compass(x,y)	errorbar()	误差限图形	errorbar(x,y,$y_{\rm m}$,$y_{\rm M}$)
feather()	羽毛状图	feather(x,y)	fill()	二维填充图	fill(x,y,c)
hist()	直方图	hist(y,n)	loglog()	对数图	loglog(x,y)
quiver()	矢量图	quiver(x,y)	polarplot()	极坐标图	polarplot(x,y)
stairs()	阶梯图形	stairs(x,y)	semilogx()	x-半对数图	semilogx(x,y)
stem()	火柴杆图	stem(x,y)	semilogy()	y-半对数图	semilogy(x,y)

MATLAB提供了polarplot()函数，其调用格式为polarplot($\boldsymbol{\theta}$,$\boldsymbol{\rho}$)，其中，$\boldsymbol{\theta}$和$\boldsymbol{\rho}$为给定数据构成的向量。该函数与早期版本polar()函数的调用格式是一样的，但在新版本中不建议使用该函数。在当前的版本中，即使$\rho<0$，也可以通过坐标变换绘制极坐标曲线。如果不希望绘制$\rho<0$部分的曲线，则可以将$\rho<0$时的函数值设置为NaN，以便绘图时自动排除这些点。

例5-9　试用极坐标绘制函数polarplot()绘制出$\rho=5\sin(4\theta/3)$的极坐标曲线。

解　由初中数学课程介绍可以立即得出结论，该函数的周期为$3\pi/2$。所以若想绘制极坐标曲线，则应该先构造向量$\boldsymbol{\theta}$，然后求出向量$\boldsymbol{\rho}$，最后调用polarplot()函数就可以绘制出所需的极坐标曲线，如图5-11(a)所示。

```
>> theta=0:0.01:3*pi/2; rho=5*sin(4*theta/3); %生成极坐标向量
   polarplot(theta,rho)                        %极坐标图曲线
```



(a) $\theta\in(0,3\pi/2)$　　(b) $\rho>0$的新极坐标曲线

图5-11　极坐标曲线

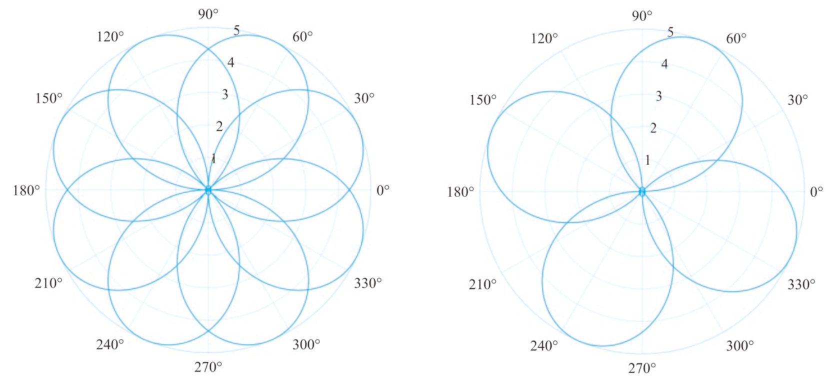
如果想排除$\rho<0$的部分，则可以将满足$\rho<0$的点强行置为NaN，这样，绘图时自动排除这些点，绘制的新极坐标曲线如图5-11(b)所示。

```
>> rho(rho<0)=NaN; polarplot(theta,rho)   %屏蔽掉NaN点
```

观察得出的曲线，绘制的极坐标图好像不完整。如何绘制完整的极坐标曲线呢？很多极坐标函数是周期函数，如果能确定函数的周期则可以绘制完整的曲线，这样就引出了新的问题——如何确定周期？对MATLAB这样的工具而言，其实不必考虑周期问题。若想绘制完整的极坐标曲线，选一个较大的θ范围，如$0\leqslant\theta\leqslant 20\pi$或更大范围，其完整的极坐标曲线如图5-12(a)所示。通过试凑

方法可见，该函数的实际周期是6π。如果只考虑$\rho \geqslant 0$部分，则得出的结果如图5-12(b)所示。

```
>> theta=0:0.01:20*pi; rho=5*sin(4*theta/3); %重新生成数据
   polarplot(theta,rho)                        %极坐标图曲线
   figure; rho(rho<0)=NaN; polarplot(theta,rho)
```



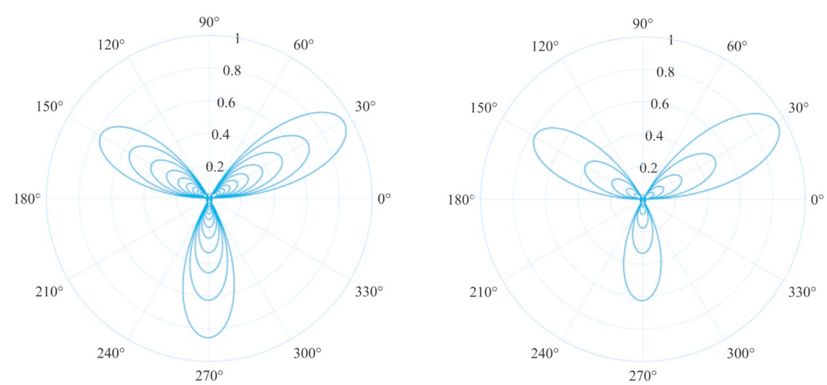
（a）$\theta \in (0, 20\pi)$　　（b）$\rho \geqslant 0$部分

图5-12　更大范围的极坐标曲线

例5-10　试绘制非周期极坐标函数$\rho = \mathrm{e}^{-0.1\theta}\sin 3\theta$的函数曲线。

解　选择自变量θ的变化范围为$\theta \in (0, 10\pi)$，则可以生成函数的极坐标数据，并绘制出极坐标曲线，如图5-13所示。

```
>> theta=0:0.001:10*pi;
   rho=exp(-0.1*theta).*sin(3*theta); polarplot(theta,rho)
   figure; rho(rho<0)=NaN; polarplot(theta,rho)
```



（a）广义极坐标曲线　　（b）$\rho \geqslant 0$部分

图5-13　非周期函数的极坐标曲线

很多极坐标函数是周期函数，选择一个周期内的θ值就可以把极坐标曲线绘制出来。这里的函数不是周期性的，所以不论取多大范围都不能绘制完整的极坐标曲线。

5.2.2　离散数据的图形表示

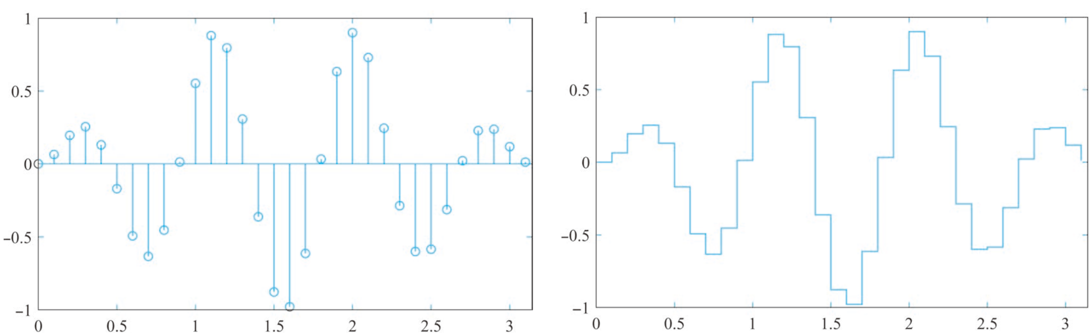
本节首先给出离散信号（discrete signal）的定义，然后介绍基于MATLAB的离散信号表示方法，以及经过零阶保持器后的输出信号。

离散信号可以表示为时间序列$y_1, y_2, \cdots, y_n$。离散信号当然可以用`plot()`函数直接绘制，不过更恰当的是用`stem()`绘图语句绘制，绘成火柴杆图。其调用格式为`stem(t,y)`，其中，t为时间点构成的向量。如果离散信号后面跟一个零阶保持器（zero-order hold，ZOH），则该信号将变成连续信号且在每一个采样周期（sample period）内都保持常值，那么该信号可以用`stairs()`函数绘制出来的阶梯信号表示，其调用格式为`stairs(t,y)`。

例 5-11 假设已知离散信号的数学表示为$f(t)=\sin t\sin 7t$，且$t=kT$，$k=0,1,2,\cdots,31$，$T=0.1\,\mathrm{s}$称为采样周期，表示每间隔0.1 s采集一次函数信号。试用图形方式表示该序列信号。

解 根据给出的采样周期，可以由下面命令生成时间向量t，然后计算出离散信号的函数值，并直接绘制出其示意图，如图5-14(a)所示。

```
>> T=0.1; t=(0:31)*T;              %生成时间采样点向量
   f=sin(t).*sin(7*t); stem(t,f)   %计算函数值并绘图
```



（a）火柴杆图　　（b）阶梯信号图

图5-14　离散信号示意图

如果用`stairs()`函数取代`stem()`，则可以绘制出如图5-14(b)所示的阶梯图。

```
>> stairs(t,f)   %绘制阶梯信号图，使其在一个采样周期内保持不变
```

5.2.3　统计图形绘制

直方图（histogram）与饼图（pie chart）是统计学领域经常使用的绘图工具。本节将先给出直方图与频度的定义，然后通过例子介绍直方图与饼图的绘制方法与技巧。

假设已知一组离散的检测数据$x_1, x_2, \cdots, x_n$，并且这组数据都位于(a,b)区间内，则可以将这个区间分成等间距的m个子区间（bins），使得$b_1=a$，$b_{m+1}=b$。将每个随机量x_i依其大小投入相应的子区间，并记子区间(b_j, b_{j+1})落入的数据个数为k_j，$j=1,2,\cdots,m$，则可以得出$f_j=k_j/n$，f_j为频度（frequency）。

如果想由数据绘制其概率密度函数直方图，可以直接调用`histogram()`函数：

```
h=histogram(x,b,'Normalization','pdf'); %直接绘制概率密度直方图
```

其中，h为结构体，其`Values`成员返回每个子区间的频度。下面将通过例子演示直方图的表示方法。

例 5-12　生成满足参数为$b=1$的Rayleigh分布的30000×1伪随机数向量，并用直方图验证生成的数据是否满足期望的分布。

解 可以由raylrnd()函数生成30000×1的伪随机数向量，选择子区间刻度向量$\boldsymbol{x}$，这样可以通过histogram()函数计算每个子区间落入的数据个数，其结果的Values属性为落入每个子区间的点数。可以用函数bar()绘制近似概率密度函数，如图5-15所示。该图还叠印了Rayleigh分布的概率密度函数理论值，可以看出，二者的吻合度比较高。

```
>> b=1; p=raylrnd(1,30000,1); x=0:0.1:4;        %子区间划分
   h=histogram(p,x,'Normalization','pdf');       %直方图数据
   y=raylpdf(x,1); line(x,y)                     %概率密度函数理论值
```

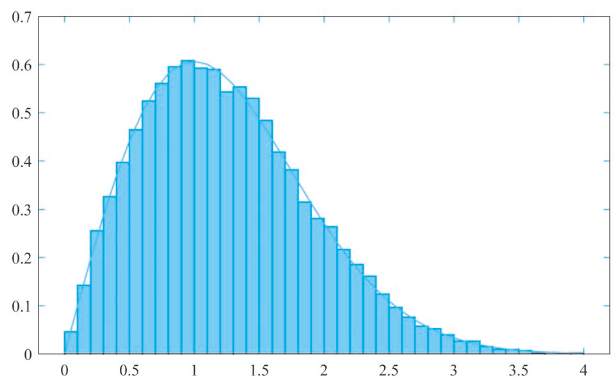


图5-15 Rayleigh分布的概率密度函数及其近似

由前面介绍的频度向量h.Values还可以绘制出饼图，调用格式为pie(h.Values)。用饼图可以大致显示落入每个子区间点数的占比。

例5-13 仍考虑例5-12中的数据。绘制饼图不能像前面例子中分那么多子区间，否则绘制的饼图意义不大，可将子区间分为$[0,0.5]$，$(0,5,1]$，$\cdots$，$(3.5,4]$，试绘制饼图表示数据的区间分布。

解 可以仿照上述方法先将频度向量计算出来，然后根据频度向量绘制出饼图，如图5-16所示。

```
>> b=1; p=raylrnd(1,30000,1); x=0:0.5:4;
   h=histogram(p,x,'Normalization','pdf');
   h1=h.Values; pie(h1), f1=h1*50
```

饼图显示虽然直观，但如果不对照频度数据很难看出哪个饼图分区对应于哪个子区间，所以还应该同时显示占比向量$\boldsymbol{f}_1=[11.5,28.2,27.8,\ 19.1,8.9,3.4,0.8,0.2]$%。

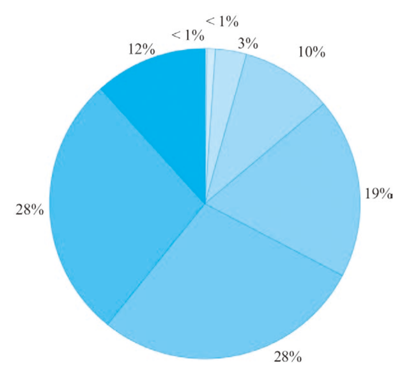


图5-16 Rayleigh分布的饼图表示

5.2.4 填充图

如果有一组坐标点$A_1(x_1,y_1)$，$A_2(x_2,y_2)$，$\cdots$，$A_n(x_n,y_n)$，由$A_1\sim A_n$作折线，再由线段连接A_n和A_1点，构成一个封闭的形状，MATLAB提供的fill()函数可以对封闭形状的内部进行填充，得出填充图(filled plot)。该函数的调用格式为fill($\boldsymbol{x}$,$\boldsymbol{y}$,c)，其中c是颜色标识，例如，可以用'g'表示绿色，参考表5-1，此外，c还可以由三原色向量表示。

如果想让A_i这些坐标点与x轴围成封闭图形，且向量$\boldsymbol{x}$是从小到大或从大到小排列的向量，则可以在左右各补充一个点，分别为$(x_1,0)$和$(x_n,0)$，使得$\boldsymbol{x}=[x_1,\boldsymbol{x},x_n]$，$\boldsymbol{y}=[0,\boldsymbol{y},0]$，这样就可以由fill()函数获得填充图形了。

例5-14 考虑例5-12中的Rayleigh分布，试用填充颜色的方法表示出面积达到95%的概率密度函数曲线。

解 生成一个$x \in (0,4)$的行向量，得出相应的Rayleigh概率密度函数值。本例一个关键的步骤是得出95%面积的关键点，该点可以由逆概率密度函数`raylinv()`直接求出，记为x_0，由下面的语句可以得出$x_0 = 2.4477$。由于左端$x = 0$时概率密度的值为0，所以左侧不必补充点。现在看填充向量$\boldsymbol{x}_1$的右侧。先从向量$\boldsymbol{x}$中提取出$\boldsymbol{x} < x_0$的点，再在横坐标补上两次x_0的值。这两个x_0对应的$\boldsymbol{y}$值，一个是概率密度函数计算出来的函数值；另一个是0，以确保围成区域是期望的区域。这样得出的曲线如图5-17所示。

```
>> x=0:0.1:4; b=1; y=raylpdf(x,b); x0=raylinv(0.95,b)
   ii=x<=x0; x1=[x(ii) x0, x0]; y1=[y(ii),raylpdf(x0,b),0];
   plot(x,y), hold on; fill(x1,y1,'c'), hold off
```

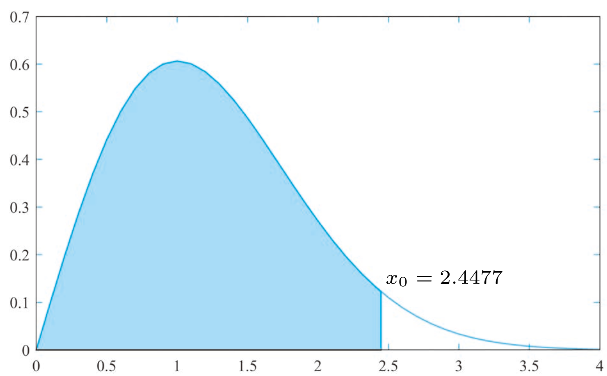


图5-17 Rayleigh分布的概率密度与95%区域

5.2.5 对数图绘制

在一些特定的领域，如数字信号处理与自动控制等领域，经常需要对信号与系统做频域分析，而Bode图分析方法就是一种常用的频域分析方法。

Bode图是对系统$G(s)$在$s = \mathrm{j}\omega_1, \mathrm{j}\omega_2, \cdots, \mathrm{j}\omega_m$处的增益的描述，其中$\omega_k$称为频率点。增益$\boldsymbol{G}(\mathrm{j}\boldsymbol{\omega})$为复数向量，而复数采用的是幅值$|\boldsymbol{G}(\mathrm{j}\boldsymbol{\omega})|$与相位$\angle\boldsymbol{G}(\mathrm{j}\boldsymbol{\omega})$形式描述的。正常情况下，Bode图采用上下两幅图表示，分别表示幅值与频率的关系（幅频特性）、相位与频率的关系（相频特性）。频率坐标轴采用对数型坐标轴，幅值通过$20\lg|\boldsymbol{G}(\mathrm{j}\boldsymbol{\omega})|$变换，单位为分贝（dB），相位采用角度为单位。

如果绘制横坐标为对数坐标、纵坐标为线性坐标，则可以使用`semilogx()`函数直接绘制，而绘制纵坐标为对数坐标、横坐标为线性坐标的函数为`semilogy()`，两个坐标轴都是对数坐标的图形可以利用`loglog()`函数绘制。

例5-15 假设系统的传递函数如下，选择频率范围为$\omega \in (0.01, 1000)$，试绘制幅值与频率之间的Bode图。

$$G(s) = \frac{2(s^{0.4}-2)^{0.3}}{\sqrt{s}(s^{0.3}+3)^{0.8}(s^{0.4}-1)^{0.5}}$$

解 一般情况下，频率点按照对数等间距的方式直接生成。这样，由给定的传递函数模型可以直接计算出以分贝为单位的幅值数据，Bode幅频特性曲线如图5-18所示。

```
>> G=@(s)2*(s.^0.4-2).^0.3./sqrt(s)./(s.^0.3+3).^0.8./(s.^0.4-1).^0.5;
   w=logspace(-2,3,100); M=20*log10(abs(G(1i*w))); %变换为分贝
   semilogx(w,M)       %绘制半对数图,横轴为对数坐标,纵轴为线性坐标
```

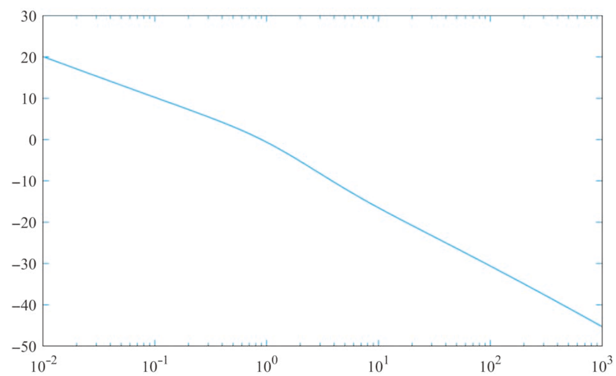


图5-18 Bode幅频特性曲线

5.2.6 动态轨迹绘制与动画制作

前面所介绍的都是静态曲线的绘制方法。如果将一条曲线看作一个粒子的运动轨迹，则用前面介绍的方法只能显示运动的最终结果，并不能看出粒子是如何运动的。如果将普通的曲线绘制函数 `plot()` 替换成 `comet()`，则可以动态地显示粒子的运动轨迹。

例5-16 试动态显示例5-1中粒子的运动轨迹。

解 选择步距为0.001，则可以由下面语句直接动态地显示粒子的运动轨迹。

```
>> x=-pi:0.001:pi; y=sin(tan(x))-tan(sin(x)); comet(x,y)
```

前面所给出的绘图命令似乎可以直接绘制出所需的图形。不过可以考虑这样一种场景：如果一个绘图命令后面紧接一组耗时计算命令，则由MATLAB现有的执行机制看，图形绘制往往不能立即进行，需要在计算完成后才能将图形绘制出来，这样的执行机制不利于动画的处理。MATLAB提供了 `drawnow` 命令，强行暂缓后面的计算命令，直接完成图形绘制任务后再开始后续命令，利用这样的方法可以实现动画的处理。

动画处理的一个关键任务是更新图形的数据点位置，让其动起来。如果调用 `plot()` 函数返回句柄，则曲线对象的数据存储在 `XData` 和 `YData` 属性中，可以更新图形数据点的位置信息，实现动画的效果。下面通过例子演示动画处理方法。

例5-17 考虑Brown运动的一群粒子，粒子个数 $n=30$，观察的区域为 $[-30,30]$，每个粒子的位置满足 $x_{i+1,k}=x_{i,k}+\sigma\Delta x_{i,k}$，$y_{i+1,k}=y_{i,k}+\sigma\Delta y_{i,k}$，$k=1,2,\cdots,n$，其中，$\sigma$ 为比例因子，增量 $\Delta x_{i,k}$ 和 $\Delta y_{i,k}$ 满足标准正态分布。试用动画的方法模拟粒子的Brown运动。

解 标准正态分布的随机数可以由 `randn()` 函数直接生成，取比例因子 $\sigma=0.3$，则可以在死循环下由向量化方法模拟。由于这里用到了死循环，所以可按Ctrl+C组合键强行终止程序的运行。

```
>> n=30; x=randn(1,n); y=randn(1,n); s=0.3;  %生成伪随机数
   figure(gcf), hold off; %当前的窗口提前,若没有当前窗口则打开新窗口
   h=plot(x,y,'o'); axis([-30,30,-30,30])     %固定坐标系范围
```

```
while (1)                                      %死循环结构模拟动画
   x=x+s*randn(1,n); y=y+s*randn(1,n);         %计算粒子新位置
   h.XData=x; h.YData=y; drawnow               %改写粒子位置并立即更新
end
```

5.2.7 图形窗口的分割

在实际应用中，可以根据需要将MATLAB的图形窗口划分为若干区域，在每个区域内绘制出不同的图形。本节将介绍规范的分区方法与不规则的分区方法，并通过例子演示这样的方法及其应用。

规范分区就是将整个图形窗口分割为$m \times n$个均匀的分区，以便在每个分区绘制出不同的图形。规范分割方法在实际应用中是很常用的。MATLAB提供的`subplot()`函数可以直接用于图形窗口的分割，其调用格式为`subplot(`m`,`n`,`k`)`，其中k是需要绘图的分区编号，该编号是按行计算的。该函数还可以带一个返回变元h`=subplot(`m`,`n`,`k`)`，其中h为该分区坐标系的句柄。如果m、n和k都是个位数，则逗号可以略去。

例5-18 试绘制例5-15中分数阶系统$G(s)$的Bode图。

解 Bode图将图形窗口分为上下两个部分，所以比较适合使用`subplot()`函数分割——分割成2×1的区域，上面的区域编号是1，下面的是2。分割完成之后就可以在两个部分分别绘制幅频特性与相频特性曲线了。幅频特性可以完全使用例5-15的代码，相频特性需要重新计算，得出的结果如图5-19所示。

```
>> G=@(s)2*(s.^0.4-2).^0.3./sqrt(s)./(s.^0.3+3).^0.8./(s.^0.4-1).^0.5;
   w=logspace(-2,3,100); subplot(211)              %或 subplot(2,1,1)
   G0=G(1i*w); M=20*log10(abs(G0)); semilogx(w,M)  %幅频特性
   subplot(212), P=angle(G0)*180/pi; semilogx(w,P) %相频特性
```

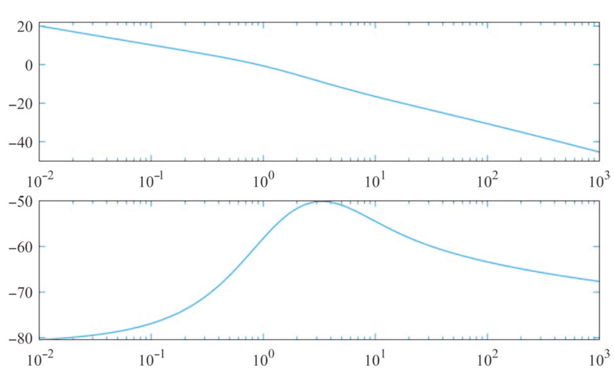


图5-19 分数阶系统的Bode图

不规则分区是指利用图形窗口的“插入▸坐标区”菜单项（图5-9(c)），实现用户用光标在图形窗口中画出任意的坐标系的功能。这样，用户就可以在新坐标系绘制图形。

5.3　MATLAB三维绘图

有些数学函数和数据的三维图可以由三维坐标系下的曲线表示，有些需要用三维曲面表示，这完全取决于数学函数与数据的形式和意义。本节将侧重介绍三维图形的表示方法。

5.3.1　三维曲线绘制

考虑一个在三维空间运动的质点，如果这个质点在t时刻的空间位置由参数方程$x(t)$、$y(t)$、$z(t)$表示，则这个质点的运动轨迹就可以看成一条三维曲线。

MATLAB中的二维曲线绘制函数`plot()`可以扩展到三维曲线的绘制中。这时可以用`plot3()`函数绘制三维曲线。该函数的调用格式为

plot3($\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$)

plot3($\boldsymbol{x}_1$,$\boldsymbol{y}_1$,$\boldsymbol{z}_1$,选项1,$\boldsymbol{x}_2$,$\boldsymbol{y}_2$,$\boldsymbol{z}_2$,选项2,$\cdots$,$\boldsymbol{x}_m$,$\boldsymbol{y}_m$,$\boldsymbol{z}_m$,选项m)

其中“选项”和二维曲线绘制的完全一致，如表5-1所示。$\boldsymbol{x}$、$\boldsymbol{y}$、$\boldsymbol{z}$为时刻$\boldsymbol{t}$的空间质点的坐标构成的向量。还可以按照表5-2中的控制参数绘制三维曲线。

相应地，类似于二维曲线绘制函数，MATLAB还提供了其他的三维曲线绘制函数，如`stem3()`可以绘制三维火柴杆型曲线，`fill3()`可以绘制三维的填充图形，`bar3()`可以绘制三维的直方图等。如果采用`comet3()`函数将得出动态的轨迹显示。这些函数的调用格式可以参见其二维曲线绘制函数原型。

例5-19　试绘制参数方程$x(t)=t^3\mathrm{e}^{-t}\sin 3t$, $y(t)=t^3\mathrm{e}^{-t}\cos 3t$, $z=t^2$的三维曲线。

解　若想绘制该参数方程的曲线，可以先定义一个时间向量$\boldsymbol{t}$，由其计算出向量$\boldsymbol{x}$、$\boldsymbol{y}$、$\boldsymbol{z}$，并用函数`plot3()`绘制出三维曲线，如图5-20所示。注意，这里应该采用点运算。

```
>> t=0:0.01:2*pi;        %构造时间向量,注意下面的点运算
   x=t.^3.*exp(-t).*sin(3*t); y=t.^3.*exp(-t).*cos(3*t); z=t.^2;
   plot3(x,y,z), grid   %三维曲线绘制,并绘制坐标系网格
```

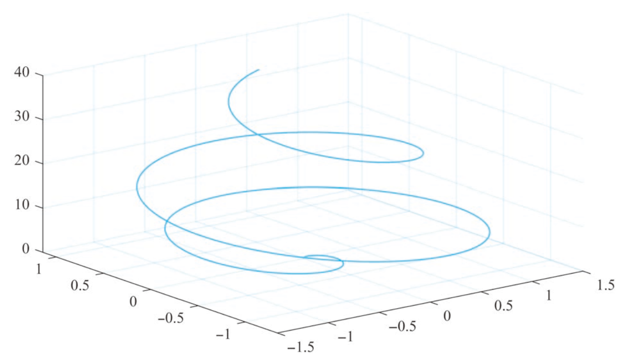


图5-20　三维曲线的绘制

5.3.2　三维参数方程的曲线绘制

如果已知三维函数的参数方程$x(t)$、$y(t)$、$z(t)$，还可以使用`fplot3()`函数直接绘制三维函数的曲线，该函数的调用格式为

fplot3(f_x,f_y,f_z), fplot3(f_x,f_y,f_z,[$t_{\rm m}$,$t_{\rm M}$])

其中，f_x、f_y 和 f_z 为参数方程的数学表示形式，可以是符号表达式，也可以是匿名函数表达式。参数 t 的默认区间为 $[0,5]$。

例5-20 重新考虑例5-19的空间质点函数，试由数学公式直接绘制三维曲线。

解 将参数方程用符号表达式表示，则可以给出如下的命令，这样，由fplot3()函数可以得出与例5-19完全一致的结果。

```
>> syms t; x=t^3*exp(-t)*sin(3*t);
   y=t^3*exp(-t)*cos(3*t); z=t^2; fplot3(x,y,z,[0,2*pi])
```

参数方程还可以由匿名函数表示，由下面的语句可以绘制出同样的三维曲线。

```
>> x=@(t)t.^3.*exp(-t).*sin(3*t); y=@(t)t.^3.*exp(-t).*cos(3*t);
   z=@(t)t.^2; fplot3(x,y,z,[0,2*pi]);   %注意使用点运算
```

5.3.3 三维曲面绘制

如果已知二元函数 $z=f(x,y)$，则可以考虑先在 xy 平面生成一些网格点，然后求出每个点处的函数值 z，这样就可以由这些信息绘制三维图了。

例3-13演示了MATLAB提供的meshgrid()函数生成网格的方式与矩阵的生成格式，由该函数可以生成两个矩阵 $\boldsymbol{x}$ 与 $\boldsymbol{y}$，将两个矩阵摆在一起正好形成了每个网格点的 x 与 y 坐标值，这时如果函数 $z=f(x,y)$ 已知，则可以直接用点运算的方式计算出每个网格点的函数值矩阵 $\boldsymbol{z}$。有了这三个矩阵，就可以调用MATLAB提供的mesh()与surf()函数直接绘制三维数据的网格图与表面图。这两个函数的调用格式为mesh($\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$)或surf($\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$)。surf()函数还可以返回曲面的句柄，这样就可以对得出的曲面进行进一步的操作处理。

例5-21 给出二元函数 $z=f(x,y)=(x^2-2x){\rm e}^{-x^2-y^2-xy}$，其中，$-3\leqslant x\leqslant 2$，$-2\leqslant y\leqslant 2$，试绘制该函数的三维表面图形。

解 选择步距为0.1，可以调用meshgrid()函数生成 xy 平面的网格矩阵 $\boldsymbol{x}$、$\boldsymbol{y}$。然后由给出的公式计算出曲面的矩阵 $\boldsymbol{z}$。最后调用surf()函数绘制曲面的三维表面图，如图5-21所示。

```
>> [x,y]=meshgrid(-3:0.1:2,-2:0.1:2);   %生成xy平面的网格矩阵x、y
   z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);  %计算高度矩阵z
   surf(x,y,z)                          %绘制三维表面图
```

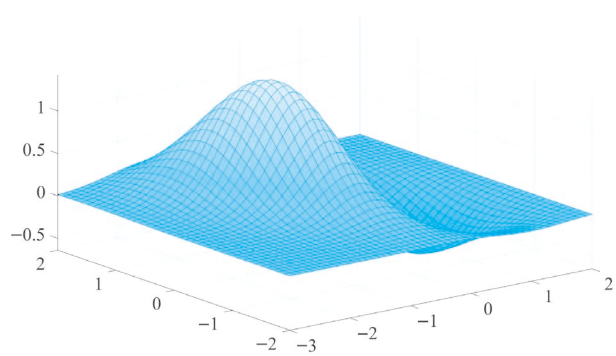


图5-21 三维函数的表面图

如果二元函数的解析表达式已知，还可以采用fsurf()直接绘制函数的曲面。该函数的调用格式为fsurf(f)或fsurf(f,[$x_{\rm m}$,$x_{\rm M}$,$y_{\rm m}$,$y_{\rm M}$])，其中，f为二元函数的符号表达式或匿名函数句柄，函数绘制的默认区域为$-5 \leqslant x, y \leqslant 5$。

例5-22 试用fsurf()函数绘制例5-21函数的三维曲面。

解 可以用符号表达式描述给定的函数，然后调用fsurf()函数绘制函数的表面图，其结果与图5-21中给出的结果完全一致。

```
>> syms x y; f=(x^2-2*x)*exp(-x^2-y^2-x*y); %给出符号表达式
   fsurf(f,[-3,2,-2,2])                      %在指定区域内绘制曲面图
```

当然，还可以由下面的匿名函数表示给定的函数，替代上面语句的f。由该函数调用上述的fsurf()函数，也能得出完全一致的结果。注意，这里函数描述需要采用点运算。

```
>> f=@(x,y)(x.^2-2*x).*exp(-x.^2-y.^2-x.*y); fsurf(f,[-3,2,-2,2])
```

5.3.4 视角设置

单击MATLAB图形窗口的坐标系工具栏（图5-10(c)）中的图标，则可以用拖动鼠标的方法直接修改三维图的视角（viewpoint），用可视化方法直接调整为期望的视角。也可以调用view()函数进行设置。

MATLAB三维图形视角的定义如图5-22所示，视角是由两个角度唯一描述的，这两个角度分别为方位角（azimuth）与仰角（elevation），其中，方位角α定义为视点与原点连线在xy平面投影线与y轴负方向之间的夹角，默认值为$\alpha=-37.5°$，仰角β定义为视点与原点连线和xy平面的夹角，默认值为$\beta=30°$。MATLAB中的当前的视角可以由[α,β]=view(3)语句读出，如果想改变视角来观察曲面，则可以给出view(α,β)命令。

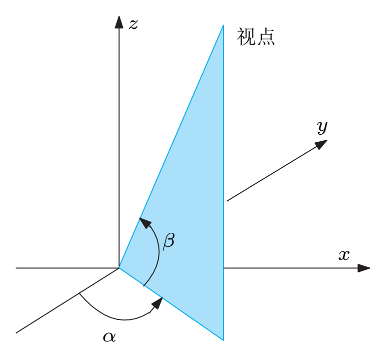


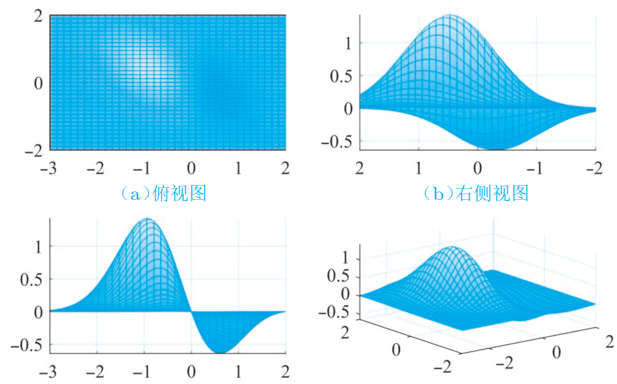
图5-22 视角定义示意图

在工程制图领域经常需要绘制物体的三视图。其中，俯视图是从上往下看的，显然，这时仰角为90°，方位角为0°，所以可以由命令view(0,90)直接设置视角。类似地，还可以分别由view(0,0)和view(90,0)命令定义主视图和右视图的视角。这样由surf()等命令绘制函数曲面后，给出修改视角的命令，就可以得出其三视图。下面通过例子演示三视图的绘制方法。

例5-23 仍考虑例5-21中的函数表达式，试绘制其曲面的三视图。

解 可以将整个图形窗口分割成2×2的分区，这样就可以在不同的命令在相应的分区绘制三视图，如图5-23所示。

```
>> syms x y; f=(x^2-2*x)*exp(-x^2-y^2-x*y);
   subplot(221), fsurf(f,[-3,2,-2,2]), view(0,90)  %俯视图
   subplot(222), fsurf(f,[-3,2,-2,2]), view(-90,0) %右视图
   subplot(223), fsurf(f,[-3,2,-2,2]), view(0,0)   %主视图
   subplot(224), fsurf(f,[-3,2,-2,2])              %三维表面图
```



（a）俯视图　（b）右侧视图

（c）主视图　（d）三维表面图

图5-23　三维曲面的三视图

5.3.5　二元参数方程的曲面绘制

前面介绍了带有单个自变量的参数方程，该参数方程对应的是三维曲线。如果参数方程带有两个自变量u、v，三维参数方程的数学形式为

$$x=f_x(u,v),\ y=f_y(u,v),\ z=f_z(u,v) \tag{5-2}$$

且$u_\mathrm{m}\leqslant u\leqslant u_\mathrm{M}$，$v_\mathrm{m}\leqslant v\leqslant v_\mathrm{M}$，则由`fsurf(`$f_x$`,`$f_y$`,`$f_z$`,[`$u_\mathrm{m}$`,`$u_\mathrm{M}$`,`$v_\mathrm{m}$`,`$v_\mathrm{M}$`])`函数可以直接绘制三维表面图，其中，变量$u$、$v$的默认区间为$(-5,5)$。使用早期版本的MATLAB还可以尝试使用`ezsurf()`函数绘制表面图，不过本书不建议使用该函数。

例5-24　著名的Möbius带可以由数学模型$x=\cos u+v\cos u\cos u/2$，$y=\sin u+v\sin u\cos u/2$，$z=v\sin u/2$描述。如果$0\leqslant u\leqslant 2\pi$，$-0.5\leqslant v\leqslant 0.5$，试绘制Möbius带的三维表面图。

解　声明两个符号变量u、v，并将参数方程输入MATLAB环境中，这样就可以由下面的语句直接绘制Möbius带，得出如图5-24所示的表面图。

```
>> syms u v; x=cos(u)+v*cos(u)*cos(u/2); y=sin(u)+v*sin(u)*cos(u/2);
   z=v*sin(u/2); fsurf(x,y,z,[0,2*pi,-0.5,0.5])  % Möbius 带的绘制
```

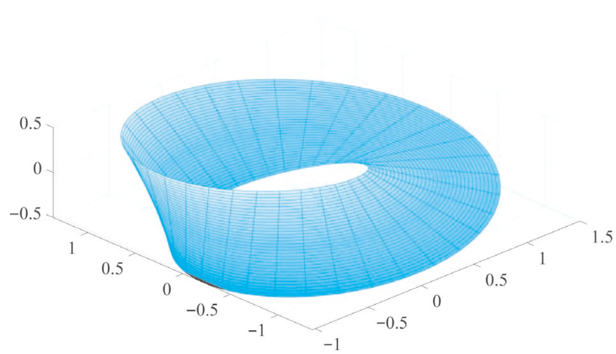


图5-24　Möbius带的表面图(图形经过了旋转处理)

5.3.6　三维动画的制作与播放

若某个三维曲面图是时间的函数，则由`getframe`命令提取每个时间样本点绘制的三维曲面句柄，这样就可以提取一系列句柄。有了句柄就可以调用`movie()`函数制作三维动画

(3D animation)的视频。本节将通过例子演示三维动画的制作与播放方法。

例5-25 考虑一个时变的函数 $z(x,y,t)=\sin(x^2t+y^2)$，其中，$0\leqslant t\leqslant 1, -2\leqslant x,y\leqslant 2$，试用动画的方式表示函数表面图随时间 t 的变化效果。

解 三维动画的处理分为两部分。第一部分是动画的制作，需要计算出各个时刻的曲面图数据，由每个时刻的数据绘制三维表面图，然后用 getframe() 函数提取出一帧图像的句柄，通过这样的方法可以获得一系列句柄。为使得动画变化平稳，可以考虑用 axis() 函数将每帧动画固定在相同的坐标系范围内。第二部分是动画播放，有了动画的一系列句柄，则可以调用 movie() 函数播放动画。前面的叙述可以由下面的语句直接实现，获得期望的三维动画的结果。

```
>> t=linspace(0,1); [x,y]=meshgrid(-2:0.1:2);
   for i=1:length(t)                        %用循环的方式对每个时刻单独处理
      z=sin(x.^2*t(i)+y.^2); surf(x,y,z);   %绘制三维表面图
      axis([-2,2,-2,2,-1,1]); h(i)=getframe; %提取一帧图像的句柄
   end
   figure, movie(h)                         %三维动画的直接播放
```

值得指出的是，getframe() 函数不仅可以提取三维动画的帧，如果绘制二维曲线图也可以用该函数提取一帧，所以可以用该函数将二维图形制作成动画的形式显示出来。此外，VideoWriter() 函数可以打开一个视频文件，writeVideo() 函数可以往视频文件中写入一帧视频，下面将通过例子演示动画视频的制作过程。

例5-26 试将例5-17中的Brown二维运动动画制作成视频文件。

解 假设运动200步，则由下面的命令生成动画数据，并制作动画的视频文件。这些语句调用结束后将在当前文件夹下生成 brown.avi 文件，该文件可以用任意媒体播放器播放。

```
>> n=30; x=randn(1,n); y=randn(1,n); s=0.3;    %生成初始位置
   figure(gcf), hold off; %当前的窗口提前，若没有当前窗口则打开新窗口
   h=plot(x,y,'o'); axis([-30,30,-30,30])
   vid=VideoWriter('brown.avi'); open(vid);    %打开空白的视频文件
   for k=1:200                                 %运动200步的动画
      x=x+s*randn(1,n); y=y+s*randn(1,n);      %更新粒子位置
      h.XData=x; h.YData=y; drawnow            %生成并绘制一帧图片
      hVid=getframe; writeVideo(vid,hVid);     %获得并写入一帧视频
   end, close(vid)                             %关闭完成视频文件
```

5.4 隐函数绘制

前面介绍的是显函数的曲线与曲面绘制，隐函数用前面介绍的方法是不能直接绘制的，必须使用专门的隐函数图形绘制方法。本节将介绍二元与三元隐函数的图形绘制方法。

5.4.1 二维隐函数曲线绘制

隐函数(implicit function)即满足 $f(x,y)=0$ 方程的 x 和 y 之间的关系式。用前面介绍的曲线绘制方法绘制显然会有问题。例如，很多隐函数无法求出 x 和 y 之间的显式关系，所以无法先定义向量 $\boldsymbol{x}$ 再求出相应的向量 $\boldsymbol{y}$，从而不能采用 plot() 函数绘制曲线。另外，即使

能求出x、y之间的显式关系，但可能不是单值函数，因此绘制也是很麻烦的。使用前面介绍的显函数绘制命令`fplot()`也不能绘制隐函数曲线。

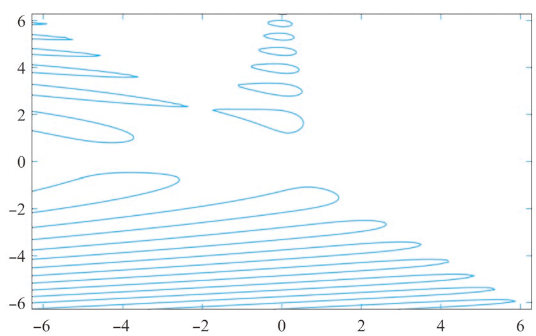
使用MATLAB提供的`fimplicit()`函数可以直接绘制隐函数曲线，该函数的调用格式为`fimplicit(隐函数表达式)`，其中，“隐函数表达式”既可以是符号表达式，也可以是点运算描述的匿名函数。用户还可以指定绘图范围`fimplicit(隐函数表达式,[`$x_{\rm m}$`,`$x_{\rm M}$`])`，得出可读性更好的曲线。坐标轴范围的默认区间为$[-5,5]$。

早期版本的MATLAB还提供了绘制二元隐函数曲线的实用函数`ezplot()`，其调用格式与`fimplicit()`函数接近，可以用字符串描述隐函数方程。不过，`ezplot()`函数不能处理由`piecewise()`语句描述的分段函数模型。下面将通过例子演示`fimplicit()`函数的使用方法。

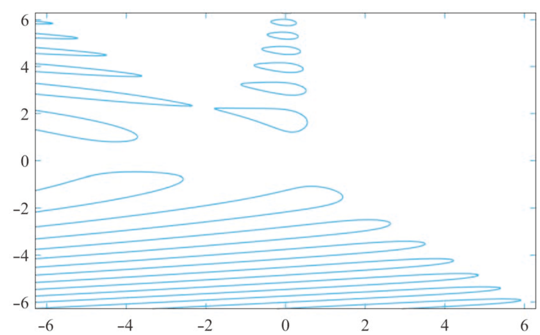
例5-27　试绘制隐函数$y^2\cos(x+y^2)+x^2{\rm e}^{x+y}=0$在$(-2\pi,2\pi)$的曲线。

解　从给出的函数可见，无法用解析的方法写出该函数，所以不能用前面给出的`plot()`函数绘制出该函数的曲线。可以给出如下的MATLAB命令，绘制出如图5-25(a)所示的隐函数曲线。可见，隐函数绘制是很简单的，只需将隐函数原原本本地表示出来，就能直接得出相应的曲线。

```
>> syms x y; f=y^2*cos(x+y^2)+x^2*exp(x+y); %符号表达式描述
   fimplicit(f,[-2*pi,2*pi])                %在指定的范围内绘制隐函数曲线
```



（a）默认参数直接绘制



（b）光滑的隐函数曲线

图5-25　隐函数曲线绘制

从得出的曲线看，默认设置下得到的曲线不平滑，有些地方有毛刺，故可以修改`MeshDensity`参数，将其设置为1000，绘制出如图5-25(b)所示的光滑隐函数曲线。

```
>> fimplicit(f,[-2*pi,2*pi],'MeshDensity',1000) %绘制光滑的隐函数曲线
```

还可以使用下面的匿名函数形式描述隐函数，注意，应该使用点运算描述向量运算。描述了函数之后，绘制命令`fimplicit()`与前面介绍的是完全一致的。

```
>> f=@(x,y)y.^2.*cos(x+y.^2)+x.^2.*exp(x+y); %匿名函数,点运算
```

例5-28　试用隐函数绘制的方法求解联立方程组在$-2\pi\leqslant x,y\leqslant 2\pi$的范围内的解。

$$\begin{cases} x^2{\rm e}^{-xy^2/2}+{\rm e}^{-x/2}\sin(xy)=0 \\ y^2\cos(y+x^2)+x^2{\rm e}^{x+y}=0 \end{cases}$$

解　上面的每个方程都可以看作一个隐函数，这样就可以用`fimplicit()`函数同时绘制这两个隐函数，得出如图5-26所示的隐函数曲线。这样，两组曲线的每个交点都是联立方程的解（这是

习题1.8的解答。）

```
>> syms x y; f1=x^2*exp(-x*y^2/2)+exp(-x/2)*sin(x*y);
   f2=y^2*cos(y+x^2)+x^2*exp(x+y);
   fimplicit([f1,f2],[-2*pi,2*pi],'MeshDensity',1000)
```

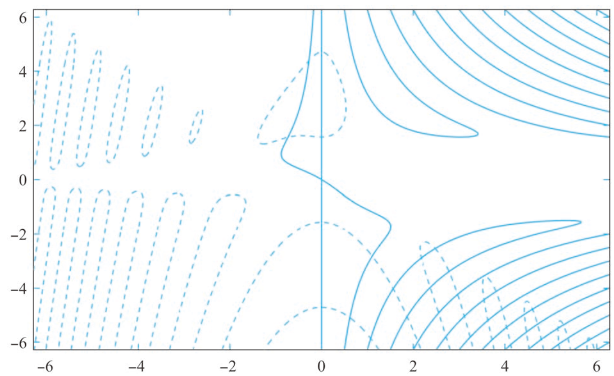


图5-26　联立方程图解法示意图

如果对某个交点处的解感兴趣，可以考虑采用局部放大的方法读出解的x，y值，这种方法称为图解法。不过，若想利用图解法逐一求出图中所有交点处的解是极其麻烦的。第9章将介绍多解方程的数值求解方法，一次性得出方程在指定区域内所有的解。

5.4.2　三维隐函数曲面绘制

如果某三维曲面由隐函数$f(x,y,z)=0$表示，则可以利用MATLAB的`fimplicit3()`绘制其曲面图形，该函数的调用格式为`fimplicit3(fun,[`$x_{\rm m}$,$x_{\rm M}$,$y_{\rm m}$,$y_{\rm M}$,$z_{\rm m}$,$z_{\rm M}$`])`，其中`fun`可以为匿名函数，也可以是符号表达式；坐标轴范围向量$x_{\rm m}$、$x_{\rm M}$、$y_{\rm m}$、$y_{\rm M}$、$z_{\rm m}$、$z_{\rm M}$的默认值为±5。如果只给出一对上下限$x_{\rm m}$、$x_{\rm M}$，则表示三个坐标轴均为同样的设置。该函数的核心部分是等高面绘制函数。

例5-29　假设某三维隐函数的数学表达式为

$$f(x,y,z)=x\sin\left(y+z^2\right)+y^2\cos\left(x+z\right)+zx\cos\left(z+y^2\right)=0$$

且感兴趣的区域为$x,y,z\in(-1,1)$，试绘制其三维曲面。

解　用符号表达式或匿名函数的方式都可以描述原始的隐函数，二者作用相同。用下面语句就可以直接绘制出该隐函数的三维曲面图，如图5-27(a)所示。

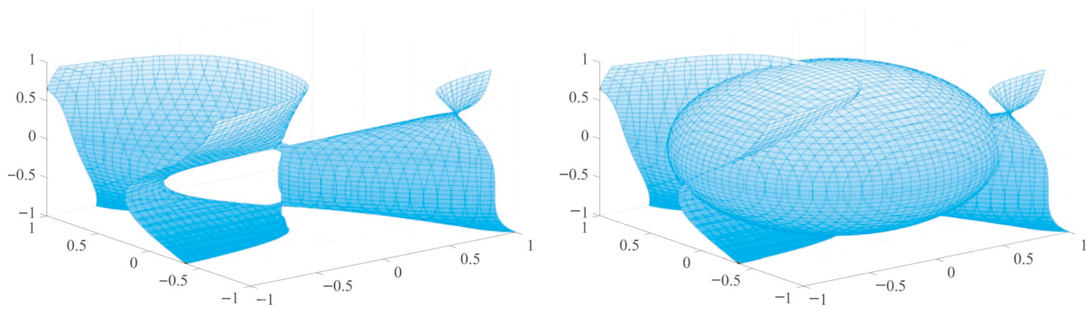
```
>> syms x y z; f=x*sin(y+z^2)+y^2*cos(x+z)+z*x*cos(z+y^2);
   fimplicit3(f,[-1 1]) %三维隐函数曲面绘制
```

其实，三元隐函数还可以用匿名函数描述，得出的结果是一致的。

```
>> f=@(x,y,z)x.*sin(y+z.^2)+y.^2.*cos(x+z)+z.*x.*cos(z+y.^2);
   fimplicit3(f,[-1,1])
```

还可以在原曲面上叠印单位球面$x^2+y^2+z^2=1$，如图5-27(b)所示。

```
>> f1=x^2+y^2+z^2-1; fimplicit3([f f1],[-1 1]); %同时绘制两个曲面
```



（a）直接绘制的三维曲面　　（b）叠印球面的三维曲面

图5-27　隐函数三维曲面绘制

5.5　小项目

5.5.1　Mandelbrot图

美国数学家Benoît Mandelbrot造了fractal（分形）这个词，并创建了一个名为分形几何的领域。在该领域定义了 Mandelbrot集，可以由简单数学公式生成漂亮的计算机图像。

假设复平面的一个点z，不妨假设$z=0$。考虑复平面的一个点c，如果反复使用迭代公式$z=z^2+c$，且z仍为有界的，则它为Mandelbrot集内的一个点。如果经过一些迭代次数之后z的值发散，则该点不属于Mandelbrot集。如果给出一个收敛条件$|z|<2$，且运行d次迭代之后，该条件仍然满足，则该点的指标设置为d，其中d为事先指定的数值，称为深度。如果z点映射k次，该条件不再满足，则将z点的指标设置为$k-1$。

文献[1]给出了一段简单的代码。利用上述的变量名将其修改如下：

```
>> x=0:0.05:0.80; y=0:0.05:0.80; d=32;
   n=length(x); z=zeros(n); M=z; [X,Y]=meshgrid(x,y); c=X+Y*1i;
   for k=1:d
      z=z.^2+c; M(abs(z)<2)=k;
   end
   image(M), axis image, colormap(flipud(jet(d)))
```

用户可以修改第一行代码，其中，向量$\boldsymbol{x}$和$\boldsymbol{y}$描述感兴趣的复平面区域，d为深度，可以设置为256。若该区域的中心为c_0，宽度为w，且网格格式为m，则向量$\boldsymbol{x}$和$\boldsymbol{y}$可以由下面命令计算。

```
>> hw=0.5*w; rc=real(c0); ic=imag(c0);
   x=linspace(rc-hw,rc+hw,m); y=linspace(ic-hw,ic+hw,m);
```

将上面的代码修改为可重用的M-函数，并绘制出$c_0=-0.5$，$w=3$，$m=512$且$d=256$时的图像。修改感兴趣的区域，例如，选择$c_0=-1.6735-0.0003318\mathrm{j}$，$w=1.5\times10^{-4}$，$n=160$且$m=1026$，使用不同的颜色映射方案，如`flag`方案，绘制Mandelbrot集的图像。用户还可以自行选择中心点、宽度、网格数与深度等值，观察得出的美丽图像。

5.5.2 曲面的交线

考虑图5-27(b)，其数学解释为：如果有两个由三自变量隐函数描述的曲面，则可以同时绘制出来。我们可能很自然地思考，这两个曲面的交线是什么？

MATLAB并未提供直接求取、绘制曲面交线的函数。文献[2]给出了一种解决这类问题的比较好的思路与代码实现。阅读相应的材料，弄明白其中的绘制原理是什么，每条MATLAB语句的含义是什么。适当修改对应的代码，绘制例5-29中隐函数曲面的交线。选择合适的步距，观察其与交线宽度之间的关系，并测出运行时间。将交线叠印到图5-27(b)的曲面上，观察得出的结果是否正确。更进一步地，编写绘制交线的通用M-函数。

本章习题

5.1 试绘制函数曲线 $y(x)=\sin \pi x/(\pi x)$，其中 $x\in(-4,4)$。

5.2 选择合适的步距绘制出图形 $\sin 1/t$，其中 $t\in(-1,1)$。

5.3 选择合适的步距，绘制 $\tan t$ 曲线，$t\in(-\pi,\pi)$，并观察不连续点的处理方法。

5.4 试绘制下面的函数曲线。

(1) $f(x)=x\sin x, x\in(-50,50)$ (2) $f(x)=x\sin 1/x, x\in(-1,1)$

5.5 试选择合适的 t 范围，绘制 $x=\sin t, y=\sin 2t$ 的曲线。如果有一个质点在该曲线上运动，试绘制其运动的动态显示。

5.6 试在 $t\in(0,2\pi)$ 区间内在同一坐标系下绘制三条曲线 $\sin x$、$\sin 2x$、$\sin 3x$。

5.7 用MATLAB语言的基本语句显然可以立即绘制一个正三角形。试结合循环结构，编写一个小程序，在同一个坐标系下绘制出该正三角形绕其中心旋转后得出的一系列三角形，并且可以通过调整旋转步距观察不同效果。

5.8 试在区间 $-50\leqslant x,y\leqslant 50$ 内绘制 $x\sin x+y\sin y=0$ 的曲线。

5.9 试绘制分段函数的曲线

$$y(t)=\begin{cases}\sin t+\cos t, & t\leqslant 0\\ \tan t, & t>0\end{cases}$$

5.10 已知正态分布的概率密度函数为 $p(x)=\dfrac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-(x-\mu)^2/(2\sigma^2)}$，其中 μ 为均值，σ 为方差，试绘制不同参数 μ、σ 下的概率密度函数曲线。

5.11 试按照下面的规则构造某序列的前40项，再用 `stem()` 函数显示序列变化趋势。

$$x_k=1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{k}-\ln k$$

5.12 已知迭代模型如下，且迭代初值为 $x_0=0$，后续各点可以递推求出 $y_0=0$。

$$\begin{cases}x_{k+1}=1+y_k-1.4x_k^2\\ y_{k+1}=0.3x_k\end{cases}$$

如果取迭代初值为 $x_0=0, y_0=0$，那么进行30000次迭代求出一组向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，然后在所有的 x_k 和 y_k 坐标处点亮一个点(注意不要连线)，最后绘制出所需的图形。提示：这样绘制出的图形又称为Hénon引力线图，它将迭代出来的随机点吸引到一起，最后得出貌似连贯的引力线图。

5.13 假设某幂级数展开表达式为

$$f(x)=\lim_{N\to\infty}\sum_{n=1}^{N}(-1)^n\frac{x^{2n}}{(2n)!}$$

若N足够大，则幂级数$f(x)$收敛为某个函数$\hat{f}(x)$。试写出一个MATLAB程序，绘制出$x\in(0,\pi)$区间的$\hat{f}(x)$的函数曲线，观察并验证$\hat{f}(x)$是什么样的函数。

5.14 试在$t\in(0,\pi)$区间内绘制$\sin t^2$的函数曲线，若某个区域曲线关系不清晰，则可以考虑采用局部放大的方法显示细节。

5.15 分别选取合适的θ范围，绘制出下列极坐标图形。

（1）$\rho=1.0013\theta^2$　（2）$\rho=\cos 7\theta/2$　（3）$\rho=\sin\theta/\theta$　（4）$\rho=1-\cos^3 7\theta$

5.16 试绘制参数方程曲线$x=(1+\sin 5t/5)\cos t, y=(1+\sin 5t/5)\sin t, t\in(0,2\pi)$。如果将参数方程中的5替换成其他数值会得出什么结果？

5.17 用图解的方式求解下面联立方程的近似解。

（1）$\begin{cases}x^2+y^2=3xy^2\\ x^3-x^2=y^2-y\end{cases}$　（2）$\begin{cases}\mathrm{e}^{-(x+y)^2+\pi/2}\sin(5x+2y)=0\\ (x^2-y^2+xy)\mathrm{e}^{-x^2-y^2-xy}=0\end{cases}$

5.18 已知正弦函数$y=\sin(\omega t+20^\circ), t\in(0,2\pi), \omega\in(0.01,10)$，试绘制当$\omega$变化时正弦函数曲线的动画。

5.19 已知某空间质点的运动方程为$x(t)=\cos t+t\sin t$，$y(t)=\sin t-t\cos t$，$z(t)=t^2$，且$t\in(0,2\pi)$，试绘制该空间质点的运动轨迹，并绘制出该质点随时间变化的动态轨迹。

5.20 试分别绘制出xy、$\sin xy$和$\mathrm{e}^{2x/(x^2+y^2)}$的三维表面图。

5.21 试绘制函数的三维表面图$f(x,y)=\sin\sqrt{x^2+y^2}/\sqrt{x^2+y^2}, -8\leqslant x,y\leqslant 8$。

5.22 试绘制下列参数方程的三维表面图[3]。

（1）$x=2\sin^2 u\cos^2 v, y=2\sin u\sin^2 v, z=2\cos u\sin^2 v, -\pi/2\leqslant u,v\leqslant\pi/2$

（2）$x=u-u^3/3+uv^2, y=v-v^3/3+vu^2, z=u^2-v^2, -2\leqslant u,v\leqslant 2$

5.23 试绘制下面函数的表面图，还可以使用`waterfall()`、`surfc()`、`surfl()`等函数并观察效果。

（1）$z=xy$　（2）$z=\sin x^2y^3$　（3）$z=\dfrac{(x-1)^2y^2}{(x-1)^2+y^2}$　（4）$z=-xy\,\mathrm{e}^{-2(x^2+y^2)}$

5.24 已知$x=u\sin t, y=u\cos t, z=t/3, t\in(0,15), u\in(-1,1)$，试绘制参数方程的曲面。

5.25 在图形绘制语句中，若函数值为不定式`NaN`，则相应的部分不绘制出来。试利用该规律绘制$z=\sin xy$的表面图，并剪切$x^2+y^2\leqslant 0.5^2$的部分。

5.26 试绘制$x(z,y)=(z^2-2z)\mathrm{e}^{-z^2-y^2-zy}$的三维曲面。

5.27 若已知$x=\cos t(3+\cos u), y=\sin t(3+\cos u), z=\sin u$，且$t\in(0,2\pi), u\in(0,2\pi)$，试绘制表面图。

5.28 已知二元函数$f(x,y)=x\sin(1/y)+y\sin(1/x)$，试用图形方法研究$(x,y)$在$(0,0)$点附近的行为。

5.29 试绘制复杂隐函数曲线$(r-3)\sqrt{r}+0.75+\sin 8\sqrt{r}\cos 6\theta-0.75\sin 5\theta=0$，其中$r=x^2+y^2$，$\theta=\arctan(y/|x|)$。

5.30 试绘制出三维隐函数$(x^2+xy+xz)\mathrm{e}^{-z}+z^2yx+\sin(x+y+z^2)=0$的曲面。

5.31 试绘制两个曲面$x^2+y^2+z^2=64, y+z=0$并观察其交线。

5.32 MATLAB提供的`treeplot()`函数可以绘制树图。例如，图4-7给出的二叉树图可以由下面语句直接绘制。

```
>> nodes=[0,1,1,2,2,3,3,4,4,5,5,6,6,8,8]; treeplot(nodes)
```

试理解上面语句的实际含义，并扩展该图形，绘制Fibonacci序列的7级二叉树结构，并编写函数绘制k级二叉树图。

参考文献

[1] 克利夫•莫勒. MATLAB之父：编程实践（修订版）[M]. 薛定宇，译. 北京：北京航空航天大学出版社，2018.

[2] Garrity M. Implicit surface intersections[EB]. https://blogs.mathworks.com/graphics/2015/07/22/implicit-surface-intersections, 2015.

[3] Majewski M. MuPAD pro computing essentials[M]. 2nd ed. Berlin: Springer, 2004.