第7章 绘图与图形处理 人们很难从一大堆原始的数据中发现它们的含义，而数据图形恰能使视觉感官直接感受到数据的许多内在本质，发现数据的内在联系。MATLAB可以表达出数据的二维，三维，甚至四维的图形。通过图形的线型，立面，色彩，光线，视角等属性的控制，可把数据的内在特征表现得淋漓尽致。下面我们分别介绍图形的命令。

**第7章  绘图与图形处理**

人们很难从一大堆原始的数据中发现它们的含义，而数据图形恰能使视觉感官直接感受到数据的许多内在本质，发现数据的内在联系。MATLAB可以表达出数据的二维，三维，甚至四维的图形。通过图形的线型，立面，色彩，光线，视角等属性的控制，可把数据的内在特征表现得淋漓尽致。下面我们分别介绍图形的命令。

**7.1  二维图形**

**7.1.1  基本平面图形命令**

命令1  plot

功能  线性二维图。在线条多于一条时，若用户没有指定使用颜色，则plot循环使用由当前坐标轴颜色顺序属性（current axes ColorOrder property）定义的颜色，以区别不同的线条。在用完上述属性值后，plot又循环使用由坐标轴线型顺序属性（axes LineStyleOrder property）定义的线型，以区别不同的线条。

用法  plot(X,Y) 当X,Y均为实数向量，且为同维向量（可以不是同型向量），X=[x(i)]，Y=[y(i)]，则plot(X,Y)先描出点(x(i)，y(i))，然后用直线依次相连；若X，Y为复数向量，则不考虑虚数部分。若X，Y均为同维同型实数矩阵，X = [X(i)]，Y = [Y(i)]，其中X(i),Y(i)为列向量，则plot(X,Y)依次画出plot(X(i),Y(i))，矩阵有几列就有几条线；若X，Y中一个为向量，另一个为矩阵，且向量的维数等于矩阵的行数或者列数，则矩阵按向量的方向分解成几个向量，再与向量配对分别画出，矩阵可分解成几个向量就有几条线；在上述的几种使用形式中，若有复数出现，则复数的虚数部分将不被考虑。

plot(Y) 若Y为实数向量，Y的维数为m，则plot(Y)等价于plot(X,Y)，其中x=1：m；若y为实数矩阵，则把y按列的方向分解成几个列向量，而y 的行数为n，则plot(Y)等价于plot(X,Y)其中x=[1;2;…;n]；在上述的几种使用形式中，若有复数出现，则复数的虚数部分将不被考虑。

plot(X1,Y1,X2,Y2,…)，其中Xi与Yi成对出现，plot(X1,Y1,X2,Y2,…)将分别按顺序取两数据Xi与Yi进行画图。若其中仅仅有Xi或Yi是矩阵，其余的为向量，向量维数与矩阵的维数匹配，则按匹配的方向来分解矩阵，再分别将配对的向量画出。

plot(X1,Y1,LineSpec1,X2,Y2,LineSpec2…) 将按顺序分别画出由三参数定义Xi,Yi,LineSpeci的线条。其中参数LineSpeci指明了线条的类型，标记符号，和画线用的颜色。在plot 命令中我们可以混合使用三参数和二参数的形式：

plot(X1,Y1,LineSpec1,X2,Y2,X3,Y3,LineSpec3)

plot(…,'PropertyName',PropertyValue,…) 对所有的用plot生成的line图形对象中指定的属性进行恰当的设置。

h = plot(…) 返回line图形对象句柄的一列向量，一线条对应一句柄值。

说明  参数LineSpec

功能  定义线的属性。Maltab允许用户对线条定义如下的特性：

1．线型

表7-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 定义符 | - | -- | ： | -. |
| 线型 | 实线（缺省值） | 划线 | 点线 | 点划线 |

2．线条宽度

指定线条的宽度，取值为整数（单位为像素点）

3．颜色

表7-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 定义符 | R（red） | G(green) | b(blue) | c(cyan) |
| 颜色 | 红色 | 绿色 | 兰色 | 青色 |
| 定义符 | M(magenta) | y(yellow) | k(black) | w(white) |
| 颜色 | 品红 | 黄色 | 黑色 | 白色 |

4．标记类型

表7-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 定义符 | + | o(字母) | \* | . | x |
| 标记类型 | 加号 | 小圆圈 | 星号 | 实点 | 交叉号 |
| 定义符 | d | ^ | v | > | < |
| 标记类型 | 棱形 | 向上三角形 | 向下三角形 | 向右三角形 | 向左三角形 |
| 定义符 | s | h | P |  |  |
| 标记类型 | 正方形 | 正六角星 | 正五角星 |  |  |

5．标记大小

指定标记符号的大小尺寸，取值为整数（单位为像素）

6．标记面填充颜色

指定用于填充标记符面的颜色。取值在上表。

7．标记周边颜色

指定标记符颜色或者是标记符（小圆圈、正方形、棱形、正五角星、正六角星和四个方向的三角形）周边线条的颜色。取值在上表。

在所有的能产生线条的命令中，参数LineSepc可以定义线条的下面三个属性：线型、标记符号、颜色进行设置。对线条的上述属性的定义可用字符串来定义，如：plot(x,y,'-.or')

结合x和y，画出点划线（-.），在数据点（x，y）处画出小圆圈（o），线和标记都用红色画出。其中定义符（即字符串）中的字母、符号可任意组合。若没有定义符，则画图命令plot自动用缺省值进行画图。若仅仅指定了标记符，而非线型，则plot只在数据点画出标记符。如：plot(x,y,’d’)

例7-1

   >>t = 0:pi/20:2\*pi;

   >>plot(t,t.\*cos(t),'-.r\*')

   >>hold on

   >>plot(exp(t/100).\*sin(t-pi/2),'--mo')

   >>plot(sin(t-pi),':bs')

   >>hold off

图形结果为图7-1。

例7-2

   >>plot(t,sin(2\*t),'-mo', 'LineWidth',2,'MarkerEdgeColor','k',…

         'MarkerFaceColor',[.49 1 .63],'MarkerSize',12)

图形结果为图7-2。

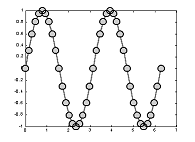
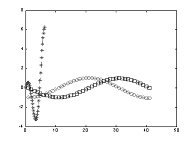


       图7-1  二维曲线图                  图7-2  二维图形的绘制

命令2  fplot

功能  在指定的范围limits内画出一元函数y=f（x）的图形。其中向量x的分量分布在指定的范围内，y是与x同型的向量，对应的分量有函数关系：y(i)=f(x(i))。若对应于x的值，y返回多个值，则y是一个矩阵，其中每列对应一个f（x）。例如，f（x）返回向量[f1(x),f2(x),f3(x)]，输入参量x=[x1;x2;x3]，则函数f（x）返回矩阵

                 f1(x1)   f2(x1)  f3(x1)

                 f1(x2)   f2(x2)  f3(x2)

                 f1(x3)   f2(x3)  f3(x3)

注意一点的是，函数function必须是一个m-文件函数或者是一个包含变量x，且能用函数eval计算的字符串。例如：’sin(x)\*exp(2\*x)’，’[sin(x),cos(x)]’，’hump(x)’。

用法  fplot('function',limits) 在指定的范围limits内画出函数名为function的一元函数图形。其中limits是一个指定x-轴范围的向量[xmin xmax]或者是x轴和y轴的范围的向量[xmin xmax ymin ymax]。

      fplot('function',limits,LineSpec) 用指定的线型LineSpec画出函数function。

      fplot('function',limits,tol) 用相对误差值为tol画出函数function。相对误差的缺省值为2e-3。

      fplot('function',limits,tol,LineSpec) 用指定的相对误差值tol和指定的线型LineSpec画出函数function的图形。

      fplot('function',limits,n) 当n>=1，则至少画出n+1个点（即至少把范围limits分成n个小区间），最大步长不超过(xmax-xmin)/n。

      fplot(‘function’,lims,…) 允许可选参数tol，n和LineSpec以任意组合方式输入。

      [X,Y] = fplot('function',limits,…) 返回横坐标与纵坐标的值给变量X和Y，此时fplot不画出图形。若想画出，可用命令plot(X,Y)。

      […] = plot('function',limits,tol,n,LineSpec,P1,P2,…) 允许用户直接给函数function输入参数P1，P2等，其中函数functiond的定义形式为

                y = function(x,P1,P2,…)

      若想用缺省的tol，n或LineSpec值，只需将空矩阵（[ ]）传递给函数即可。

注意：fplot采用自适应步长控制来画出函数function的示意图，在函数的变化激烈的区间，采用小的步长，否则采用大的步长。总之，使计算量与时间最小，图形尽可能精确。

例7-3

>>fplot('tanh',[-2 2])

图形结果为图7-3。

>>subplot(2,2,1);fplot('humps',[0 1])

>>subplot(2,2,2);fplot('abs(exp(-j\*x\*(0:9))\*ones(10,1))',[0 2\*pi])

>>subplot(2,1,2);fplot('[tan(x),sin(x),cos(x)]',2\*pi\*[-1 1 -1 1])

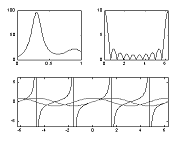
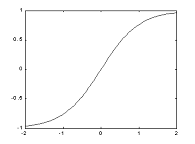


            图7-3  函数画图                      图7-4

命令3  loglog

功能  双对数图形。

用法  loglog(Y) 若y为实数向量或矩阵，则结合y列向量的下标与y的列向量画出。若y为复数向量或矩阵，则loglog(Y)等价于loglog(real(Y),imag(Y))，在loglog的其他使用形式中将忽略Y的虚数部分。

      loglog(X1,Y1,X2,Y2…) 结合Xn与Yn画出图形。若只有Xn或Yn为矩阵，另一个为向量，行向量维数等于矩阵的列数，列向量的维数等于矩阵的行数，则loglog把矩阵按向量的方向分解成向量，再与向量结合分别画出图形。

      loglog(X1,Y1,LineSpec1,X2,Y2,LineSpeec2…) 按顺序取三个参数Xn,Yn, LineSpecn画出线条，其中LineSpecn指定线条的线型，标记符号和颜色。用户可以混合使用二参数和三参数形式，如：

                 loglog(X1,Y1,X2,Y2,LineSpec2,X3,Y3)

      loglog(…,'PropertyName',PropertyValue,…) 对所有由loglog命令生成的图形对象句柄的属性进行设置。

      h = loglog(…) 返回line图形句柄向量，每条线对应一个句柄。

例7-4

>>x = logspace(-1,2);

>>loglog(x,10\*exp(x),'-s')

>>grid on

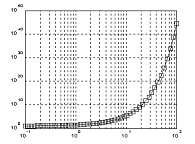


                图7-5

命令4  semilogx

功能  x轴对数图形。若没有指定使用的颜色，当所画线条较多时，semilogx将自动使用由当前轴的ColorOrder和LineStyleOrder属性指定的颜色顺序和线型顺序来画线。

用法  semilogx(Y)    %对x轴的刻度求常用对数（以10为底），而y轴为线性刻度。若y为实数向量或矩阵，则结合y列向量的下标与y的列向量画出线条；若y为复数向量或矩阵，则semilogx(Y)等价于semilogx(real(Y),imag(Y))。在semilogx的其他使用形式中，Y的虚数部分将被忽略。

      semilogx(X1,Y1,X2,Y2…)   %结合Xn和Yn画出线条，若其中只有xn或yn为矩阵，另外一个为向量，行向量的维数等于矩阵的列数，列向量的维数等于矩阵的行数，则按向量的方向分解矩阵，再与向量结合，分别画出线条。

      semilogx(X1,Y1,LineSpec1X2,Y2,LineSpec2,…)   %按顺序取三参数Xn,Yn,LineSpecn画线，参数LineSpecn指定使用的线型，标记符号和颜色。用户可以混合使用二参数和三参数形式，如：

                 semilogx(X1,Y1,X2,Y2,LineSpec2,X3,Y3)

      semilogx(…,'PropertyName',PropertyValue,…)   %对所有由semilogx命令生成的图形对象句柄的属性进行设置

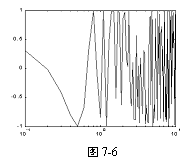
      h = semilogx(…)   %返回line图形句柄向量，每条线对应一个句柄。

例7-5

>>x = 0:.1:10;

>>semilogx(x,cos(10.^x))

图形结果为图7-6。



命令5  semilogy

用法：参见semilogx命令。

命令6  fill

功能  用颜色填充二维多边形。

用法  fill(X,Y,C) 用x和y中的数据生成多边形，用c指定的颜色填充它。其中c为色图向量或矩阵。若c是行向量，则要求c的维数等于x和y 的列数，若c为列向量，则要求c的维数等于x和y的行数。

      fill(X,Y,ColorSpec) 用ColorSpec指定的颜色填充由x和y定义的多边形

      fill(X1,Y1,C1,X2,Y2,C2,…) 指定多个要填充的二维区域

      fill(…,'PropertyName',PropertyValue) 允许用户对一个patch图形对象的某个属性设定属性值。

      h = fill(…) 返回patch图形对象句柄的向量，每一个patch对象对应一个句柄。

注意：

1. 若x或y是一矩阵，另一个是向量，向量应是维数与矩阵的行数相等的列向量或是维数等于矩阵列数的行向量时，函数fill将向量复制成与矩阵同型的矩阵。函数fill将矩阵x与y中列向量中的数据生成多边形的顶点。

2. 颜色阴影类型决定于用户在参数中列出的颜色，若用户用ColorSpec指定颜色，命令fill生成平坦阴影模式（flat-shaded）多边形，同时设置补片对象（patch）的FaceColor属性为相应的RGB颜色矩阵。

3. 若用户用参量c指定所用颜色，命令fill按坐标轴属性Clim的比例缩小c中的元素，之后，c成为引用当前色图的下标矩阵。

4. 若c为行向量，命令fill生成平面阴影的多边形，c的每一元素决定由矩阵x，y的每一列定义的多边形内的颜色，每一补片对象的FaceColor属性被设置为'flat'，x，y的每一行元素变成第n块补片对象的Cdata属性值，其中n为矩阵x或y中的相应的列。

5. 若c为一列向量或一矩阵，命令fill运用一线性插值法计算每一节点的颜色，以便用插值颜色填充多边形的内部。它设置补片对象的FaceColor属性为‘interp’，且在一列中的元素变成每一补片的Cdata属性值。若c为一列向量，命令fill用该向量复制成需要大小的尺寸。

例7-6

>>t = (1/16:1/8:1)'\*2\*pi;

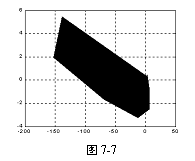
>>x = exp(t).\*sin(t);

>>y = t.\*cos(t);

>>fill(x,y,'k')

>>grid on

图形结果为图7-7。



命令7  zoom

功能  对二维图形进行放大或缩小。放大或缩小会改变坐标轴范围。

用法  zoom on 打开交互式的放大功能。当一个图形处于交互式的放大状态时，有两种方法来放大图形：

对于一键鼠标或二键，三键鼠标，单击坐标轴内的任意一点，可使图形放大一倍，这一操作可进行多次，直到matlab的最大显示为止；对于二键或三键的鼠标，在坐标轴内单击右键，可使图形缩小一倍，这一操作可进行多次，直到还原图形为止。对于一键鼠标，要想缩小图形，需要按住键盘上的Shift键，再单击鼠标键。

用鼠标拖出要放大的部分，系统将放大选定的区域。

      zoom off 关闭交互式放大功能。

      zoom out 将系统转回非放大状态，并将图形恢复原状。

      zoom reset 系统将记住当前图形的放大状态，作为放大状态的设置值。以后使用zoom out或者是双击鼠标时，交互式放大状态打开，且图形并不是返回到原状，而是返回reset时的放大状态。

      zoom 用于切换放大的状态：on和off。

      zoom xon 只对x轴进行放大。

      zoom yon 只对y轴进行放大。

      zoom(factor) 用放大系数factor进行放大或缩小，而不影响交互式放大的状态。若factor>1，系统将图形放大factor倍，若0<factor≤1，系统将图形放大1/factor倍。

      zoom(fig, option) 指定对窗口fig中（不一定为当前窗口）的二维图形进行放大，其中参数option为：on、off、xon、yon、reset、factor等。

命令8  meshgrid

功能  生成二元函数z = f(x,y)中x-y平面上的矩形定义域中数据点矩阵X和Y，或者是三元函数u =  f(x,y,z)中立方体定义域中的数据点矩阵X,Y和Z。

用法  a：[X,Y] = meshgrid(x,y)

      b：[X,Y] = meshgrid(x)

      c：[X,Y,Z] = meshgrid(x,y,z)

说明  对于形式a，输入向量x为x-y平面上矩形定义域的矩形分割线在x轴的值，向量y为x-y平面上矩形定义域的矩形分割线在y轴的值。输出向量X为x-y平面上矩形定义域的矩形分割点的横坐标值矩阵，输出向量Y为x-y平面上矩形定义域的矩形分割点的纵坐标值矩阵。

对于形式b，等价于形式a：[X,Y] = meshgrid(x) = meshgrid(x,x)。

对于形式c，输入向量x为立方体定义域的立方体分割平面在x轴上的值，输入向量y为立方体定义域的立方体分割平面在y轴上的值，输入向量z为立方体定义域的立方体分割平面在z轴上的值。输出向量X为立方体定义域中分割点的x轴坐标值,Y为立方体定义域中分割点的y轴坐标值,Z为立方体定义域中分割点的z轴坐标值。

例7-7

    >>x = [0.7 1.1 ]; y = [-2 3 1]; z = [2 5 3];  %分量不一定从小到大

    >>[X\_2d,Y\_2d] = meshgrid(x,y)

    >>[X\_3d,Y\_3d,Z\_3d] = meshgrid(x,y,z)

计算结果为：

X\_2d =

    0.7000    1.1000

    0.7000    1.1000

    0.7000    1.1000

Y\_2d =

    -2    -2

     3     3

     1     1

X\_3d(:,:,1) =

    0.7000    1.1000

    0.7000    1.1000

    0.7000    1.1000

X\_3d(:,:,2) =

    0.7000    1.1000

    0.7000    1.1000

    0.7000    1.1000

X\_3d(:,:,3) =

    0.7000    1.1000

    0.7000    1.1000

    0.7000    1.1000

Y\_3d(:,:,1) =

    -2    -2

     3     3

     1     1

Y\_3d(:,:,2) =

    -2    -2

     3     3

     1     1

Y\_3d(:,:,3) =

    -2    -2

     3     3

     1     1

Z\_3d(:,:,1) =

     2     2

     2     2

     2     2

Z\_3d(:,:,2) =

     5     5

     5     5

     5     5

Z\_3d(:,:,3) =

     3     3

     3     3

     3     3

**7.1.2  特殊平面图形命令**

命令1  polar

功能  画极坐标图。该命令接受极坐标形式的函数rho=f(θ)，在笛卡儿坐标系平面上画出该函数，且在平面上画出极坐标形式的格栅。

用法  polar(theta,rho) 用极角theta和极径rho画出极坐标图形。极角theta为从x轴到半径的单位为弧度的向量，极径rho为各数据点到极点的半径向量。

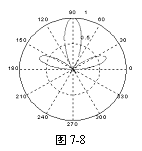
      polar(theta,rho,LineSpec) 参量LineSpec指定极坐标图中线条的线型、标记符号和颜色等。

例7-8

>>t = 0:.01:2\*pi;

>>polar(t,sin(3\*t).\*cos(2\*t),'--r')

图形结果为图7-8。



命令2  bar

功能  二维垂直条形图。用垂直条形显示向量或矩阵中的值。

用法  bar(Y) 若y为向量，则分别显示每个分量的高度，横坐标为1到length（y）；若y为矩阵，则bar把y分解成行向量，再分别画出，横坐标为1到size（y，1），即矩阵的行数。

      bar(x,Y) 在指定的横坐标x上画出y，其中x为严格单增的向量。若y为矩阵，则bar把矩阵分解成几个行向量，在指定的横坐标处分别画出。

      bar(…,width) 设置条形的相对宽度和控制在一组内条形的间距。缺省值为0.8，所以，如果用户没有指定x，则同一组内的条形有很小的间距，若设置width为1，则同一组内的条形相互接触。

      bar(…,'style') 指定条形的排列类型。类型有“group”和“stack”，其中“group”为缺省的显示模式。

      “group”：若y为n\*m阶的矩阵，则bar显示n组，每组有m个垂直条形的条形图。

      “stack”：对矩阵y的每一个行向量显示在一个条形中，条形的高度为该行向量中的分量和。其中同一条形中的每个分量用不同的颜色显示出来，从而可以显示每个分量在向量中的分布。

      bar(…,LineSpec) 用指定的颜色LineSpec显示所有的条形。

      [xb,yb] = bar(…) 返回用户可用命令plot或命令patch画出条形图的参量xb，yb。这对用户控制一个图形的显示是有用的，例如要在一个plot语句中加入装饰性的条形图等。

      h = bar(…) 返回一个patch图形对象句柄的向量。每一条形对应一个句柄。

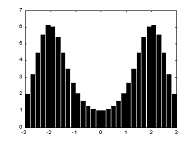
例7-9

x = -2.9:0.2:2.9;

bar(x,exp(x.\*sin(x)))

colormap gray

图形结果为图7-9。



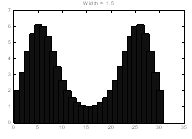
例7-10

subplot(2,2,4)

bar(Y,1.5)

title 'Width = 1.5'

图形结果为图7-10。



命令3  barh

功能  二维水平条形图。用水平条形显示向量或矩阵中的值。

用法  barh(Y) 若y为向量，则分别显示每个分量的高度，纵坐标为1到length(y)；若y为矩阵，则bar把y分解成行向量，再分别画出，纵坐标为1到size(y,1)，即矩阵的行数。

      barh(x,Y) 在指定的纵坐标x上以水平方向画出y，其中x为严格单增的向量。若y为矩阵，则barh把矩阵分解成几个行向量，在指定的纵坐标处分别画出。

      barh(…,width) 设置条形的相对宽度和控制在一组内条形的间距。缺省值为0.8，所以，如果用户没有指定x，则同一组内的条形有很小的间距，若设置width为1，则同一组内的条形相互接触。

      barh(…,'style') 指定条形的排列类型。类型有“group”和“stack”，其中“group”为缺省的显示模式。

             “group”：若y为n\*m阶的矩阵，则bar显示n组，每组有m个水平条形的条形图。

              “stack”：对矩阵y的每一个行向量显示在一个条形中，条形的高度为该行向量中的分量和。其中同一条形中的每个分量用不同的颜色显示出来，从而可以显示每个分量在向量中的分布。

      barh(…,LineSpec) 用指定的颜色LineSpec显示所有的条形。

      [xb,yb] = barh(…) 返回用户可用命令plot或命令patch画出条形图的参量xb，yb。这给用户控制一个图形的显示是有用的，例如要在一个plot语句中加入装饰性的条形图等。

      h = barh(…) 返回一个patch图形对象句柄的向量。每一条形对应一个句柄。

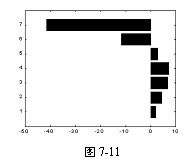
例7-11

>>X = 1:.5:5;

>>Y = exp(X).\*sin(X);

>>barh(Y,'stack')

图形结果为图7-11。



命令4  compass

功能  从原点画箭头图。箭头图为一显示起点为笛卡儿坐标系中的原点的二维或三维方向或向量的图形，同时在坐标系中显示圆形的分隔线。

用法  compass(X,Y) 参量x与y为同型的n维向量，则命令显示n个箭头，箭头的起点为原点，箭头的位置为[X(i),Y(i)]。

      compass(Z) 参量z为n维复数向量，则命令显示n个箭头，箭头起点为原点，箭头的位置为[real(Z),imag(Z)].

      compass(…,LineSpec) 用参量LineSpec指定箭头图的线型、标记符号、颜色等属性。

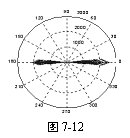
      h = compass(…) 返回line对象的句柄给h。

例7-12

  Z = magic(20).\*randn(20);

  compass(Z)

图形结果为图7-12。



命令5  comet

功能  二维彗星图。彗星图为彗星头（一个小圆圈）沿着数据点前进的动画，彗星体为跟在彗星头后面的痕迹，轨道为沿着整个函数的实线。我们要指出的是，由命令comet生成的轨迹是使用擦除模式（EraseMode）属性的值为none，该属性使用户不能打印该图形（只能得到彗星头），且当用户改变窗口的大小时，动画将消失。

用法  comet(y) 彗星图动画显示向量y确定的路线。

      comet(x,y) 彗星图动画显示向量x与y确定的路线。

      comet(x,y,p) 指定彗星体的长度p\*length(y)，缺省的p值为0.1。

例7-13

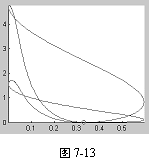
>>t = 0:.01:2\*pi;

>>x = exp(sin(2\*t)).\*(cos(t).^2/3);

>>y = t.\*(sin(t).^2);

>>comet(x,y);

图形结果为图7-13。



附：擦除模式(EraseMode)属性及属性值：{normal}|none|xor|background

    该属性控制系统用于显示与擦除线条对象的技术。不同的擦除模式对于生成动画系列，即控制个别对象的重新显示方式，对于改进外在显示和获得理想的效果是很必要的。

表7-4

|  |  |
| --- | --- |
| 属性值 | 含义 |
| Normal  (缺省值) | 重新显示受影响的区域，在必要的时候，进行三维分析计算，以保证所有的对象的显示都是正确的。该模式下的图形显示是最精确的，不过也是最缓慢的，以下其他三种模式显示速度较快，不过没有执行一个完全的重显过程，因而，图形显示也不是很精确的。 |
| none | 当线条移动或改动时，该模式没有擦除线条，而是仍然显示于屏幕上。该模式下不能打印图形，因为系统没有存储前一图形的任何信息。 |
| xor | 使用异或运算(xor)计算线条颜色与当前位置下的颜色，用所得结果显示与擦除线条。该模式对于线条下面对象的颜色没有任何破坏，只是影响到线条的当前显示颜色而已。 |
| Background | 用当前坐标轴颜色重新显示线条的方式来擦除线条，若当前坐标轴颜色设置为none，则用图形的背景色来代替坐标轴颜色。该模式对于处于擦除线条后面的对象来说是有损害的，不过当前线条的颜色总是最合适的。 |

命令6  errorbar

功能  沿着一曲线画误差棒形图。误差棒为数据的置信水平或者为沿着曲线的偏差。在下列参数中，若为矩阵，则按列画出误差棒。

用法  errorbar(Y,E) 画出向量y，同时显示在向量y的每一元素之上的误差棒。误差棒为E(i)在曲线y上面与下面的距离，所以误差棒的长度为2\*E(i)。

      errorbar(X,Y,E) X,Y,E必须为同型参量。若同为向量，则画出带长度为2\*E(i)、对称误差棒于曲线点(X(i),Y(i))之处；若同为矩阵，则画出带长度为E(i，j)、对称误差棒于曲面点(X(i,j),Y(i,j))之处，

      errorbar(X,Y,L,U) X，Y，L，U必须为同型参量。若同为向量，则在点(X(i),Y(i))处画出向下长为L(i)，向上长为U(i)的误差棒；若同为矩阵，则在点(X(i,j),Y(i.j))处画出向下长为L(i,j),向上长为U(i,j)的误差棒。

      errorbar(…,LineSpec) 用LineSpec指定的线型、标记符、颜色等画出误差棒。

      h = errorbar(…) 返回线图形对象的句柄向量给h。

例7-14

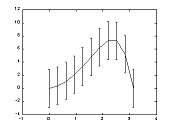
>>X = 0:pi/10:pi;

>>Y = exp(X).\*sin(X);

>>E = std(Y)\*ones(size(X));

>>errorbar(X,Y,E)

图形结果为图7-14。



命令7  feather

功能  画出速度向量图。一羽毛图在横坐标上等距地显示向量。用户要表示各个向量的、相对于原点的向量分量。

用法  feather(U,V) 显示由参量向量u与v确定的向量，其中u包含作为相对坐标系中的x成分，v包含作为相对坐标系中的y成分。

      feather(Z) 显示复数参量向量z确定的向量，等价于feather(real(Z),imag(Z))。

      feather(…,LineSpec) 用参量LineSpec指定的线型、标记符号、颜色等属性画出羽毛图。

例7-15

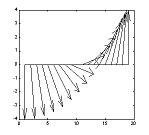
>>th = (-90:10:90)\*pi/180;

>>r = 4\*ones(size(th));

>>[u,v] = pol2cart(th,r);

>>feather(u,v);

图形结果为图7-15。



命令8  hist

功能  二维条形直方图，可以显示出数据的分配情形。所有向量y中的元素或者是矩阵y中的列向量中的元素是根据它们的数值范围来分组的，每一组作为一个条形进行显示。条形直方图中的x轴反映了数据y中元素数值的范围，直方图的y轴显示出参量y中的元素落入该组的数目。所以y轴的范围从0到任一条形中包含元素最多的数字。直方图为一patch图形对象，若想改变图形的颜色，可以对patch对象的属性进行设置。缺省时，图形颜色是由当前色图进行控制，当前色图的第一个颜色为直方图的颜色。

用法  n = hist(Y) 把向量y中的元素放入等距的10个条形中，且返回每一个条形中的元素个数。若y为矩阵，则该命令按列对y进行处理。

      n = hist(Y,x) 参量x为向量，把y中元素放到m（m=length(x)）个由x中元素指定的位置为中心的条形中。

      n = hist(Y,nbins) 参量nbins为标量，用于指定条形的数目。

      [n,xout] = hist(…) 返回向量n与包含频率计数与条形的位置向量xout，用户可以用命令bar(xout,n)画出条形直方图。

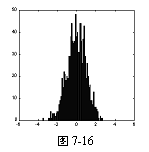
例7-16

>>x = -5:0.1:5;

>>y = randn(1000,1);

>>hist(y,x)

图形结果为图7-16。



命令9  histc

功能  直方图记数

用法  n = histc(x,edges) 统计向量x中、落入向量edges（元素必须为单调的非减的）各个元素之间的元素个数。输出参量n为一与向量edges同维的向量。其中若有edges(k)>=x(i)>=edges(k+1)，则n(k)增加1。X中超出向量edges规定的范围的元素将不被统计。参量edges中可使用-inf与inf，用于包括向量x中非NaN的元素。若x为一矩阵，则对x的每一列进行上述操作。

      n = histc(x,edges,dim) 对多维矩阵的第dim维进行统计。

      [n,bin] = histc(…) n结果同上，同时返回矩阵下标bin。若x为向量，n(k) = sum(bin == k)。对于超出范围的数值，bin为零值。

命令10  rose

功能  画角度直方图。该直方图是一个显示所给数据的变化范围内数据的分布情形的极坐标图，所给数据分成不同的组。每一组作为一小扇形进行显示。

用法  rose(theta)  画一角度直方图，显示参数theta的数据在20个区间或更少的区间内的分布。向量theta中的角度单位为弧度，用于确定每一区间与原点的角度。每一区间的长度反映出输入参量的元素落入一区间的个数。

      rose(theta,x) 用参量x指定每一区间内的元素与区间的位置，length(x)等于每一区间内元素的个数与每一区间位置角度的中间角度。例如，若x为一5维向量，rose命令分配参量theta中的元素为5部分，每一部分的角度中线由x指定。

      rose(theta,nbins) 于区间 [0,2\*pi] 内画出nbins个等距的小扇形。缺省值为20。

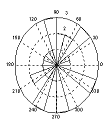
      [tout,rout] = rose(…) 返回向量tout与rout，可以用polar(tout,rout)画出图形。该命令没有画任何的图形。

例7-17

   >>theta = 3\*pi\*randn(1,30);

   >>rose(theta)

图形结果为图7-17。



命令11  stairs

功能  画二维阶梯图，这种图对与时间有关的数字样本系统的作图很有用处。

用法  stairs(Y) 用参量y的元素画一阶梯图。若y为向量，则横坐标x的范围从1到m=length(y)，若y为矩阵，则对y的每一行画一阶梯图，其中x的范围从1到y的列数m。

     stairs(X,Y) 结合x与y画阶梯图。其中要求x与y为同型的向量或矩阵。此外，x可以为行向量或为列向量，且y为有m=length（x）行的矩阵。

     stairs(…,LineSpec) 用参数LineSpec指定的线型、标记符号和颜色画阶梯图。

     [xb,yb] = stairs(Y) 该命令没有画图，而是返回可以用命令plot画出参量y的阶梯图的向量xb与yb。

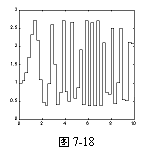
     [xb,yb] = stairs(X,Y) 该命令没有画图，而是返回可以用命令plot画出参量x，y的阶梯图的向量xb与yb。

例7-18

>>x = 0:.25:10;

>>stairs(x,exp(sin(x.^2)))

图形结果为图7-18。



命令12  stem

功能  画二维离散数据的柄形图。该图用线条显示数据点与x轴的距离，一小圆圈（缺省标记）或用指定的其他标记符号与线条相连，在y轴上标记数据点的值。

用法  stem(Y) 按y元素的顺序画出柄形图，在x轴上，柄与柄之间的距离相等；若y为矩阵，则把y分成几个行向量，在同一横坐标的位置上画出一个行向量的柄图。

      stem(X,Y) 在横坐标x上画出列向量y的柄形图 。其中x与y为同型的向量或矩阵，此外，x可以为行向量或列向量，而y为有m=length（x）行的矩阵。

      stem(…,'fill') 指定是否对柄形图末端的小圆圈填充颜色。

      stem(…,LineSpec) 用参数LineSpec指定线型，标记符号和柄图末端的小圆圈的颜色画柄图。

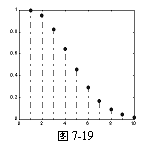
      h = stem(…) 返回柄形图的line图形对象句柄向量。

例7-19

>>x = linspace(0,2,10);

>>stem(exp(-x.^2),'fill','-.')

图形结果为图7-19。



命令13  stem3

功能  画三维离散数据的柄形图。该图用一线段显示数据离开xy平面的高度，在线段的末端用一小圆圈（缺省记号）或其他的标记符号表示数据的高度。

格式  stem3(Z) 用柄形图显示z中数据与xy平面的高度。若z为一行向量，则x与y将自动生成，stem3将在与x轴平行的方向上等距的位置上画出z的元素；若y为列向量，stem3将在与y轴平行的方向上等距的位置上画出z的元素。

      stem3(X,Y,Z) 在参数x与y指定的位置上画出z的元素，其中x，y，z必须为同型的向量或矩阵。

      stem3(…,'fill') 指定是否要填充柄形图末端小圆圈。

      stem3(…,LineSpec) 指定线型，标记符号和末端小圆圈的颜色。

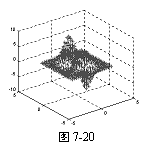
      h = stem3(…) 返回柄形图的line图形对象句柄。

例7-20

[X,Y,Z] = peaks(20);

stem3(X,Y,Z,’r\*')

图形结果为图7-20。



命令14  pie

功能  饼形图

格式  pie(X)  用x中的数据画一饼形图，x中的每一元素代表饼形图中的一部分。X中元素X(i)所代表的扇形大小通过X(i)/sum(X)的大小来决定。若有sum(X)=1，则x中元素就直接指定了所在部分的大小；若sum(X)<1，则画出一不完整的饼形图。

      pie(X,explode) 从饼形图中分离出一部分，explode为元素为零或非零的、与x相对应的向量或矩阵。与explode的非零值对应的部分将从饼形图中心分离出来。explode必须与x同型。

      h = pie(…) 返回一patch与text的图形对象句柄向量h。

例7-21

>>x = [1 3 0.5 2.5 2];

>>explode = [0 1 0 0 0];

>>pie(x,explode)

图形结果为图7-21。



**7.1.3  二维图形注释命令**

命令1  grid

功能  给二维或三维图形的坐标面增加分隔线。该命令会对当前坐标轴的Xgrid，Ygrid，Zgrid的属性有影响。

用法  grid on 给当前的坐标轴增加分隔线。

      grid off 从当前的坐标轴中去掉分隔线。

      grid 转换分隔线的显示与否的状态。

      grid(axes\_handle,on|off) 对指定的坐标轴axes\_handle是否显示分隔线。

命令2  gtext

功能  在当前二维图形中用鼠标放置文字。当光标进入图形窗口时，会变成一个大十字，表明系统正等待用户的动作。

用法  gtext('string') 当光标位于一个图形窗口内时，等待用户单击鼠标或键盘。若按下鼠标或键盘，则在光标的位置放置给定的文字“string”

      h = gtext('string') 当用户在鼠标指定的位置放置文字“string”后，返回一个text图形对象句柄给h。

命令3  legend

功能  在图形上添加图例。该命令对有多种图形对象类型（线条图，条形图，饼形图等）的窗口中显示一个图例。对于每一线条，图例会在用户给定的文字标签旁显示线条的线型，标记符号和颜色等。当所画的是区域（patch或surface对象）时，图例会在文字旁显示表面颜色。Matlab在一个坐标轴中仅仅显示一个图例。图例的位置有几个因素决定，像遮挡的对象等，用户可以用鼠标拖动图例到恰当的位置，双击标签可以进入标签编辑状态。

用法  legend('string1','string2',…) 用指定的文字string在当前坐标轴中对所给数据的每一部分显示一个图例。

      legend(h,'string1','string2',…) 用指定的文字string在一个包含于句柄向量h中的图形显示图例。用给定的数据对相应的图形对象加上图例。

      legend(string\_matrix) 用字符矩阵参量string­\_matrix的每一行字符串作为标签。

      legend(h,string\_matrix) 用字符矩阵参量string­\_matrix的每一行字符串作为标签给包含于句柄向量h中的相应的图形对象加标签。

      legend(axes\_handle,…) 给由句柄axes\_handle指定的坐标轴显示图例。

      legend('off') 从当前的坐标轴，或是由axes-handle指定的坐标轴中除掉图例。

      legend(axes\_handle,'off') 从由axes\_handle指定的坐标轴中除掉图例。

      legend\_handle = legend 返回当前坐标轴中的图例句柄，若坐标轴中没有图例存在，则返回空向量。

      legend 对当前图形中所有的图例进行刷新。

      legend(legend\_handle) 对由句柄legend\_handle指定的图例进行刷新。

      legend(…,pos) 在指定的位置pos放置图例见表7-5。

表7-5

|  |  |  |  |
| --- | --- | --- | --- |
| Pos取值 | pos=-1 | pos=0 | Pos=1 |
| 图例位置 | 坐标轴之外的右边 | 坐标轴之内，有可能遮挡部分图形 | 坐标轴的右上角（缺省位置） |
| Pos取值 | pos=2 | pos=3 | pos=4 |
| 图例位置 | 坐标轴的左上角 | 在坐标轴的左下角 | 坐标轴的右下角 |

      h = legend(…) 返回图例的句柄向量。

      [legend\_handle,object\_handles] = legend(…) 返回图例句柄，该句柄为坐标轴定义于图例中的图形对象、line对象、text对象的句柄。这些句柄允许用户对每个对象进行详细的操作。

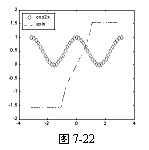
例7-22

     >>x = -pi:pi/20:pi;

     >>plot(x,(cos(x)).^2,'rd',x,asin(x),'-.b')

     >>h = legend('cos2x','asin',2);

图形结果为图7-22。



命令4  title

功能  给当前轴加上标题。每个axes图形对象可以有一个标题。标题定位于axes的上方正中央。

用法  title('string') 在当前坐标轴上方正中央放置字符串string作为标题

      title(fname) 先执行能返回字符串的函数fname，然后在当前轴上方正中央放置返回的字符串作为标题

      title(…,'PropertyName',PropertyValue,…) 对由命令title生成的text图形对象的属性进行设置

      h = title(…) 返回作为标题的text对象句柄。

命令5  text

功能  在当前轴中创建text对象。函数text是创建text图形句柄的低级函数。可用该函数在图形中指定的位置上显示字符串。

用法  text(x,y,'string')在图形中指定的位置(x,y)上显示字符串string

     text(x,y,z,'string') 在三维图形空间中的指定位置(x,y,z)上显示字符串string

     text(x,y,z,’string’.'PropertyName',PropertyValue…) 对引号中的文字string定位于用坐标轴指定的位置，且对指定的属性进行设置。表7-6给出文字属性名、含义及属性值。

表7-6

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性说明 | 属性值 |
| 定义字符串 | | |
| Editing | 能否对文字进行编辑 | 有效值：on、off  缺省值：off |
| Interpretation | TeX字符是否可用 | 有效值：tex、none  缺省值：tex |
| String | 字符串(包括TeX字符串) | 有效值：可见字符串 |
| 放置字符串 | | |
| Extent | text对象的范围（位置与大小） | 有效值：[left, bottom, width, height] |
| HorizontalAlignment | 文字水平方向的对齐方式 | 有效值：left(文本外框左边对齐，缺省对齐方式)、center(文本外框中间对齐)、right(文本外框右边对齐)  缺省值：left |
| Position | 文字范围的位置 | 有效值：[x,y,z]直角坐标系  缺省值：[]（空矩阵） |
| Rotation | 文字对象的方位角度 | 有效值：标量（单位为度）  缺省值：0 |
| Units | 文字范围与位置的单位 | 有效值：pixels (屏幕上的像素点)、normalized (把屏幕看成一个长、宽为1的矩形)、inches(英寸)、centimeters(厘米)、points (图象点)、data  缺省值：data |
| VerticalAlignment | 文字垂直方向的对齐方式 | 有效值：top (文本外框顶上对齐)、cap(文本字符顶上对齐)、middle(文本外框中间对齐)、baseline(文本字符底线齐)、bottom(文本外框底线对齐)  缺省值：middle |
| 指定文字字体 | | |
| FontAngle | 设置斜体文字模式 | 有效值：normal(正常字体)、italic(斜体字)、oblique(斜角字)  缺省值：normal |
| FontName | 设置文字字体名称 | 有效值：用户系统支持的字体名或者字符串FixedWidth。  缺省值为 Helvetica |
| FontSize | 文字字体大小 | 有效值：结合字体单位的数值  缺省值为：10 points |
| FontUnits | 设置属性FontSize的单位 | 有效值：points (1点=1/72英寸)、normalized(把父对象坐标轴作为一单位长的一个整体；当改变坐标轴的尺寸时，系统会自动改变字体的大小)、inches (英寸)、Centimeters(厘米)、Pixels(像素)  缺省值：points |
| FontWeight | 设置文字字体的粗细 | 有效值：light(细字体)、normal(正常字体)、demi(黑体字)、Bold(黑体字)  缺省值：normal |
| 控制文字外观 | | |
| Clipping | 设置坐标轴中矩形的剪辑模式 | 有效值：on、off  on：当文本超出坐标轴的矩形时，超出的部分不显示；  off：当文本超出坐标轴的矩形时，超出的部分显示。  缺省值：off |
| EraseMode | 设置显示与擦除文字的模式。这些模式对生成动画系列与改进文字的显示效果很有好处。 | 有效值：normal、none、 xor、   background  缺省值：normal |
| SelectionHighlight | 设置选中文字是否突出显示 | 有效值：on、off  缺省值：on |
| Visible | 设置文字是否可见 | 有效值：on、off  缺省值：on |
| Color | 设置文字颜色 | 有效的颜色值：ColorSpec |
| 控制对文字对象的访问 | | |
| HandleVisibility | 设置文字对象句柄对其他函数是否可见 | 有效值：on、callback、off  缺省值：on |
| HitTest | 设置文字对象能否成为当前对象（见图形CurrentObject属性） | 有效值：on、off  缺省值：on |
| 文字对象的一般信息 | | |
| Children | 文字对象的子对象（文字对象没有子对象） | 有效值：[]（即空矩阵） |
| Parent | 文字对象的父对象（通常为axes对象） | 有效值：axes的句柄 |
| Seleted | 设置文字是否显示出“选中”状态 | 有效值：on、off  缺省值：off |
| Tag | 设置用户指定的标签 | 有效值：任何字符串  缺省值：’’（即空字符串） |
| Type | 设置图形对象的类型（只读类型） | 有效值：字符串’text’ |
| UserData | 设置用户指定数据 | 有效值：任何矩阵  缺省值：[]（即空矩阵） |
| 控制回调例行执行程序 | | |
| BusyAction | 设置如何处理对文字回调过程中断的句柄 | 有效值：cancel、queue  缺省值：queue |
| ButtonDownFcn | 设置当鼠标在文字上单击时，程序做出的反应（即执行回调程序） | 有效值：字符串  缺省值：'  '（空字符串） |
| CreateFcn | 设置当文字被创建时，程序做出的反应（即执行的回调程序） | 有效值：字符串  缺省值：'  '（空字符串） |
| DeleteFcn | 设置当文字被删除（通过关闭或删除操作）时，程序做出的反应（即执行的回调程序） | 有效值：字符串  缺省值：'  '（空字符串） |
| Interruptible | 设置回调过程是否可中断 | 有效值：on、off  缺省值：on（能中断） |
| UIContextMenu | 设置与文字相关的菜单项 | 有效值：用户相关菜单句柄 |

     h = text(…) 返回文字对象句柄的列向量，每一对象对应一句柄。该命令的其他使用形式中，将随意地返回这个输出参量。

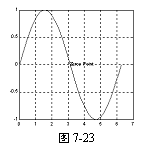
例7-23

   >>plot(0:pi/20:2\*pi,sin(0:pi/20:2\*pi))

   >>text(pi,0,’Zeros Point’)

   >>grid on

图形结果为图7-23。



命令6  xlabel、ylabel

功能  给x、y轴贴上标签

用法  xlabel('string')、 ylabel(‘string’)  给当前轴对象中的x、y轴贴标签；注意：若再次执行xlabel或ylabel命令，则新的标签会覆盖旧的标签。

      xlabel(fname)、ylabel(fname) 先执行函数fname，其返回一个字符串，然后在x、y轴旁边显示出来；

      xlabel(…,'PropertyName',PropertyValue,…)、ylabel(…,’PropertyName’, PropertyValue)指定轴对象中的要控制的属性名和要改变的属性值，这些都是由xlabel或ylabel创建的text图形对象的成对值；

      h = xlabel(…)、h = ylabel(…)返回作为标签的text对象的句柄。