第 5 章

数据可视化

本章讨论数据可视化问题。数据可视化是 MATLAB 的一个特色，可以将计算结果直接以图形的形式直观地显示出来。例如，单变量函数 $y=f(x)$ 的图像是二维(2D)的，也就是平面的，而空间曲线与曲面显示的是三维(3D)的图形。

5.1 二 维 作 图

计算机作图实际上是逐点描绘的，需要给出一系列的点对 $(x_i,y_i)(i=1,2,\cdots,n)$。其中，数据 $x_i(i=1,2,\cdots,n)$ 作为一个 n 维的向量，需要我们录入或产生。而 y_i 则是根据函数表达式与 x_i 计算出来的值。作图需要的数据 x_i 有它的特殊性。其前后两个数值的差是一样的。也就是说，x_i 是一个等差数列，即线性等距(linear squally spaced)数据。2.2 节已经介绍，有两种方法产生线性等距数据：用带冒号的指令与用内置函数 linspace。

5.1.1 用内置函数 plot 作图

MATLAB 函数“plot”是一个功能非常强大的作“线状”图形(linear plot)，即平面图形的工具。它的形参可以是一个、两个或多个。它除了可以作通常的实线条以外，还可以作出多种样式的线条，如由小点、冒号、小短线形成的线条，或由小方块、“+”号、“*”号、小菱形和字母“x”形成的线条，等等，还可以使画出的线条带各种颜色。

plot 命令的基本用法如下。

(1) plot(X,Y)；创建 Y 中数据对 X 中对应值的二维线图。

- 如果 X 和 Y 都是向量，则它们的长度必须相同。plot 函数绘制 Y 对 X 的图。
- 如果 X 和 Y 均为矩阵，则它们的大小必须相同。plot 函数绘制 Y 的列对 X 的列的图。
- 如果 X 或 Y 中的一个是向量 $\boldsymbol{b}$ 而另一个是矩阵 $\boldsymbol{A}$，则矩阵 $\boldsymbol{A}$ 的行、列向量的长度中必须有一个与向量的长度相等。如果 $\boldsymbol{A}$ 的行向量的长度等于 $\boldsymbol{b}$ 的长度，则 plot 函数绘制 $\boldsymbol{A}$ 中的每一行对 $\boldsymbol{b}$ 的图。如果 $\boldsymbol{A}$ 的列数等于 $\boldsymbol{b}$ 的长度，则该函数绘制 $\boldsymbol{A}$ 的每一列对向量 $\boldsymbol{b}$ 的图。如果 $\boldsymbol{A}$ 为方阵，则该函数绘制 $\boldsymbol{A}$ 的每一列对 $\boldsymbol{b}$ 的图。
- 如果 X 或 Y 之一为标量，而另一个为标量或向量，则 plot 函数会绘制离散点。但是，要查看这些点，必须指定标记符号，例如，plot(X,Y,'o')。

(2) **plot**(X,Y,LineSpec)；LineSpec 设置线型、记号与颜色属性。

(3) **plot**(X1,Y1,…,Xn,Yn)；用同一个坐标轴绘制多条曲线。

(4) **plot**(X1,Y1,LineSpec1,…,Xn,Yn, LineSpec$n$)；LineSpec$n$ 用来设置不同的线型、记号与颜色以绘制多条曲线。

(5) **plot**(Y)；创建 Y 中数据的二维折线图。

(6) **plot**(Y,LineSpec)；LineSpec 指定 Y 中数据的二维折线图的线型、记号与颜色绘制。

如要详细了解指令 plot 的用法,可以使用 plot 的帮助命令“help plot”。

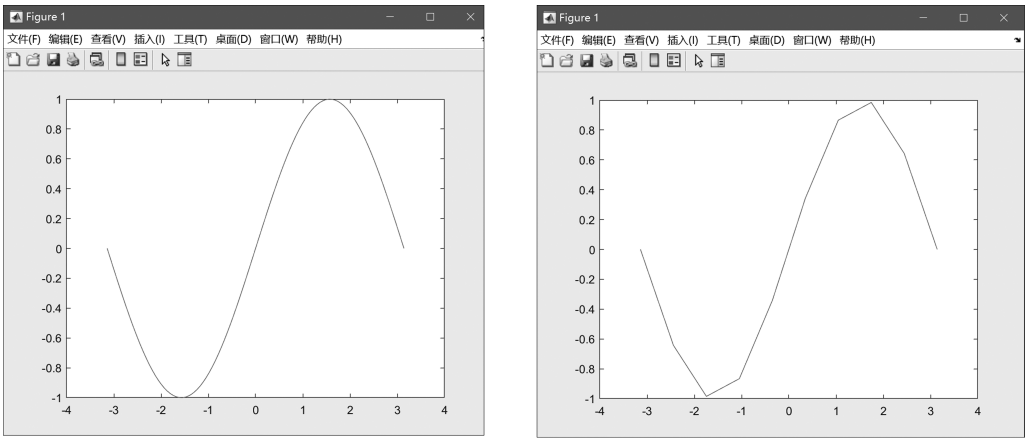
1. 绘制一般曲线

指令 plot(x, y) 绘制以首末相连直线段连接而成的(折线)图形,$\boldsymbol{x}$ 与 $\boldsymbol{y}$ 都是 n 个分量的向量。用此指令绘制由 n 个点对 $(x_i,y_i)(i=1,2,\cdots,n)$ 连接而成的曲线。

例 5.1 由默认值 100 个点对 linspace(−pi, pi) 和指定值 10 个点对 linspace(−pi, pi, 10)作出的不同光滑度的二维“曲线”图,如图 5.1 所示。

```
>>x = linspace(-pi, pi); y = sin(x);
  plot(x, y);
```

```
>>x = linspace(-pi, pi, 10); y = sin(x);
plot(x, y);
```



(a) 默认值100个点对绘制的正弦曲线图　　(b) 10个点对绘制的正弦曲线图

图 5.1　曲线图

对比图 5.1(a)和图 5.1(b),图 5.1(a)看上去是一条曲线,而图 5.1(b)明显是一条折线。这是因为,绘制图 5.1(b)的指令串 b 指定了(−pi, pi) 间绘制 10 段线,逼近 sin 曲线的精度较低。而绘制图 5.1(a)的指令串 a 没有指定(−pi, pi)间绘制线段的数目,此时采用默认逼近段数 100,逼近 sin 曲线的精度较高,看起来更像曲线。只有当点的个数足够多,即点足够密时,逐点描绘的图才能像一条曲线。

2. 图片窗口名字的命名

绘制指令产生的图形将在一个默认的图片窗口名“Figure 1”中显示。例如例 5.1 中绘制的 sin 曲线,输入(产生图 5.1(a) 的) 指令串 a 后,图 5.1(a) 左边顶上标有默认的图片窗口名“Figure 1”。如果紧接着输入(产生图 5.1(b) 的) 指令串 b 后,新产生的图形将仍在名为“Figure 1”原图片窗口中显示,前一个图就被指令串 b 产生的图形替代了。

如果想同时保留这两个图形，则可在指令串 a 与 b 的最前面分别加上图片窗口命名指令“figure(1) ;”与“figure(2) ;”。这样，两个指令产生的图形将在不同的图片窗口中显示。此时，两个指令就会分别变成：

- 指令串 a：“figure(1) ; x = linspace(－pi, pi); y = sin(x); plot(x, y);”(在 Figure 1 窗口中显示图形)。
- 指令串 b：“figure(2) ; x = linspace(－pi, pi, 10); y = sin(x); plot(x, y);”(在 Figure 2 窗口中显示图形)。

仅用指令“figure(1) ;”或“figure(2) ;”本身，则会创建一个新的图片窗口。如果在指令串 a 的最后加上指令“hold on ;”，则保持当前的图形，此后接着输入的绘图指令把新的内容加到现有的图形上，即所产生的图形都在 Figure 1 窗口中显示。

最好在最后一个绘图指令后加上一个指令“hold off ;”，表示在当前图片窗口的绘图指令已结束。

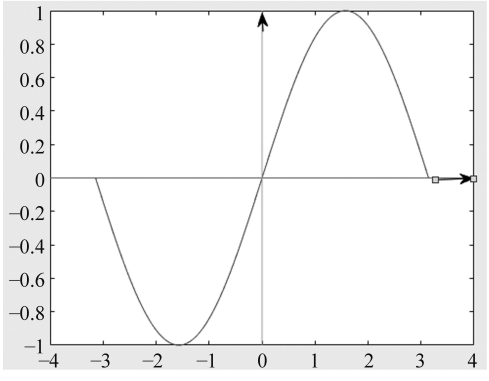
3. 绘制向量(直线)

指令 plot([x1, x2], [y1, y2]) 是 plot(x, y) 在向量 ***x*** 和 ***y*** 都只有两个分量时的特殊情况。它给出一条连接(x1, y1)与(x2, y2)的直线段。

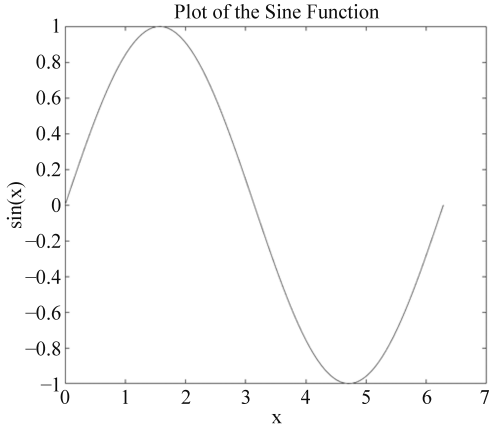
例 5.2　在例 5.1 的曲线图 5.1(*a*) 上再画两条直线段：从(－4,0) 到(4,0) 和从(0, －1)到(0,1)，这两条直线段相当于两条坐标轴，如图 5.2 所示。

```
>>x = linspace(-pi, pi);  y = sin(x);
plot(x, y);   hold on;
%绘制两条直线段(坐标轴)。
plot([-4,4],[0,0]);  plot([0,0],[-1,1]);
hold off;
```

```
>>x = linspace(0,2*pi,100);
y = sin(x) ;    plot(x,y);
%标记坐标区并添加标题。
xlabel('x');  ylabel('sin(x)');
title('Plot of the Sine Function');
```



(a) 加了坐标轴



(b) 标记坐标区并添加标题

图 5.2　绘制正弦曲线

这里，图 5.2(a) 用了“hold on;… hold off;”指令保证了 sin 曲线与后加的两条直线段显示在同一个窗口内。图 5.2(b)用了 xlabel 和 ylabel 指令分别作了横向和竖向标记，

用 title 指令加了标题。

图 5.2(a)中的向上与向右的箭头是这样画上去的：在 MATLAB 作图上方的菜单中，单击 Insert→Arrow，此时鼠标会变成“+”字。把鼠标移到要画箭头的直线段附近，顺着直线移动。如果所画的箭头不对，右键单击箭头，在打开的菜单，单击 Delete，即可删去箭头。图 5.2(b) 中的“xlable(x 轴标记)”“ylable(y 轴标记)”与“title (标题)”也可以在作出的图上单击 Insert 再选择有关项来完成。

4. 设置绘制属性

指令 plot(x, y, s) 中 *s* 是字符串，它用来指定线条的颜色和/或类型。代码的格式见后面的例子。

(1) *s* 指定线条的颜色。

MATLAB 可供选择的线条颜色有以下几种，字符所表示的颜色写在括号中。除 k 表示黑色(black 的字尾)以外，其他字符都是所表示的颜色的英文首字母。

b (blue，蓝)； g (green，绿)； r (red，红)； y (yellow，黄)； c(cyan，青)；m(magenta，洋红)；k(black，黑)。

(2) *s* 指定线条的类型。

线条的主要类型有以下几种(其他类型可用指令“help plot”来查找)。

- (solid，实线)。

: (dotted，小点组成的虚线)。

O (大小写均可，circle，小圆圈)。

-- (dashed，小短线组成的虚线)。

. (point，点)。

-. (dashdot，小短线与小点组成的虚线)。

* (star，星号)。

s (square，小方块)。

d (diamond，菱形)。

(3) *s* 指定字符串的写法。

线条的类型与颜色要写在单引号(')内的字符串 *s* 中，表示类型与颜色的字符，其先后次序是任意的。例如，绿色的菱形线条，可以写为'gd'或'dg'。

例 5.3 以下指令串画出了图 5.3，程序中有简单说明，作完图后有详细说明。

```
>>x=linspace(0,2*pi,200);                          %[0,2*pi] 中 200 个分点
y=sin(x); z=cos(x); v=sin(x-pi/2);                 %3 个函数在 200 个分点上的值
plot(x,y,'r:');                                    %作 sin(x) 曲线
hold on;                                           %要继续作图
ix=[1,40,80,120,160,200];                          %要描的 6 个点的 x 的下标向量
plot(x,z,'b.'); plot(x,v,'gd');                    %作另外 2 条曲线
plot(x(ix),z(ix),'sk');                            %黑色小方块描出 6 个离散点
legend('sin(x)', 'cos(x)', 'sin(x-pi/2)','6 points');      %图例
hold off;                                          %作图结束
```

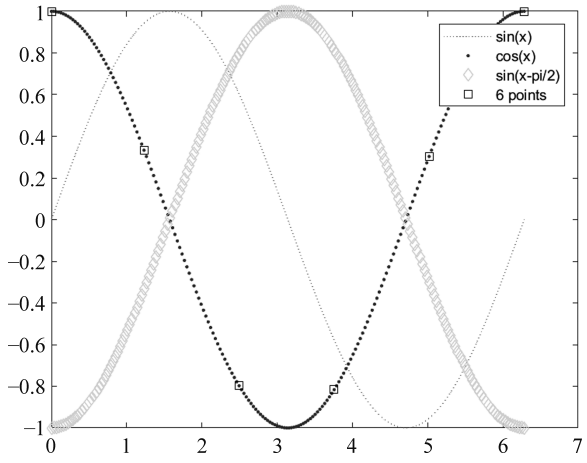


图 5.3　三种类型与颜色不同的曲线及 6 个离散点

(1) 上面最后一个 plot 指令描出的 6 个离散点中，横坐标是下标为 **ix** 分量的 ***x*** 值，而纵坐标为对应的余弦值。在数据拟合的图示中，常常需要这样来描出观察(实验)数据的离散值。

(2) 指令 legend() 是给出图例，括号内的字符串的个数应与曲线的个数相同，而且字符串的次序与 plot 曲线的次序一致。在图上显示在小方框内。把鼠标移入小方框内，按住鼠标，这时可将小方框拖移到任何地方(例如不挡住曲线的地方)。

例 5.4　以下的指令串画出了图 5.4。它与图 5.3 是同样的三条曲线，只是线条的类型与 ***s*** 中的字符次序不同。这里还把绘图 5.3 的三个 plot 指令合并成一个。而且，那些使用实参 y、z、v 的地方，这里直接用它们的表达式来代替。所以，两者画出了三条同样的曲线。

```
>> x=linspace(0,2*pi,200);
plot(x, sin(x),'-ro', x, cos(x),'-.b',x, sin(x-pi/2),'--g');
legend('sin(x)', 'cos(x)', 'sin(x-pi/2)');
```

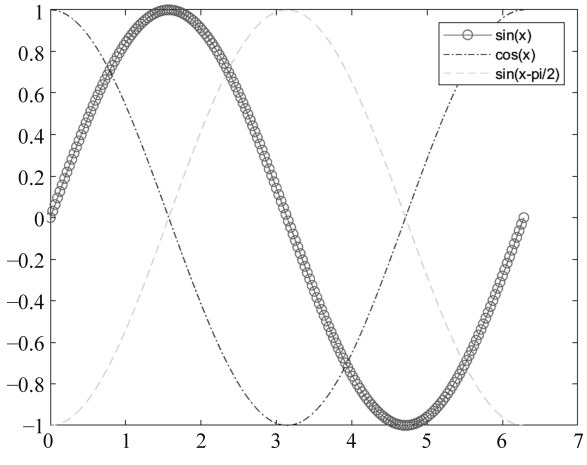


图 5.4　三种线条类型与图 5.3 不同的曲线

5. 绘制数列

指令 plot(y)用于创建 Y 中数据的二维折线图。一个数列 $y_n(n=1,2,\cdots)$可以看作 y 是它的下标 n 的函数。此时,plot(y) 是 plot(n,yn)的特例。另外,当 y 是一个复向量时,plot(y)等价于 plot(real(y),imag(y))。其中,指令 real(y)与 imag(y)分别取 y 的实部与虚部。

(1) 实数列 y。

例 5.5 绘制实数列 $y_n=\ln(n)$ $(n=1,2,\cdots,10)$,如图 5.5 所示。

```
>>n=1:10;    y=log(n);    plot(y);
```

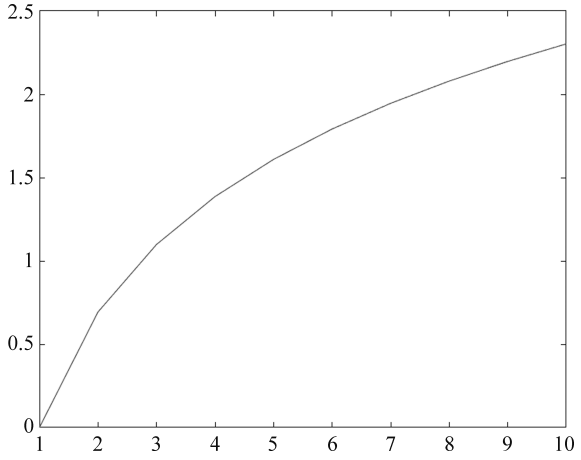


图 5.5 数列 $y_n=\ln(n)$ 相对于下标 n 的图形

(2) 复向量 $\boldsymbol{y}$。

例 5.6 指令串"x = linspace(−pi, pi); y = x+i * sin(x); plot(y);"也可以得到图 5.1(a)。

这里可以用"sum(abs(real(y)−x))"与"sum(abs(imag(y)−sin(x)))"的答案是否都等于 0 来检查 $\boldsymbol{y}$ 的实部 real(y)是否等于 x,以及 $\boldsymbol{y}$ 的虚部 imag(y) 是否等于 sin(x)。回忆一下:其中内置函数 abs(z)是求 z 的绝对值;而 sum(a)当 a 是向量时,是求它的各分量的和。

例 5.7 以下三行指令串都得到图 5.6,只是在 Figure1、Figure2 和 Figure3 这 3 个窗口里。

```
>>figure(1);  x=linspace(0, pi/2);  y=cos(x)+i * sin(x);    plot(y);
figure(2);  x=linspace(0, pi/2);  y=exp(i * x);          plot(y);
figure(3);  x=linspace(0, pi/2);  x1=cos(x); y1=sin(x);  plot(x1,y1);
```

以上第 1、2 行指令串的等价性说明了欧拉公式($e^{ix}=\cos(x)+i\sin(x)$)的正确性;

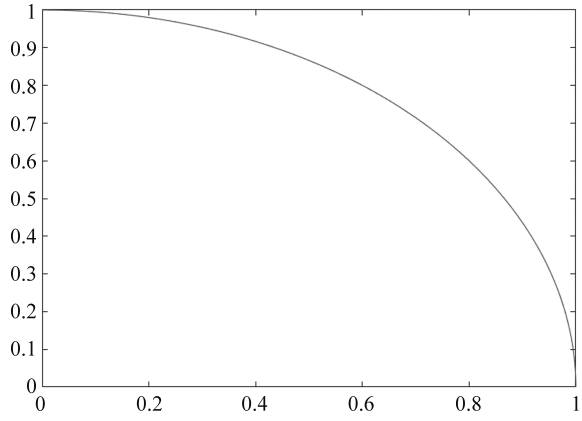


图 5.6　复向量 e^{ix} 的实部与虚部点对所描绘的图形

而第 1、3 行指令串的等价性说明了 plot(y)在 y 为复向量时，图形是实数点对（cos(x_i)，sin(x_i))描绘出的曲线。

5.1.2　辅助作图的内置函数与参数

1. 绘制网格

内置函数 grid 在图上加上网格，见例 5.8；而 grid minor 加上细网格，见例 5.9。

2. 加图标题

内置函数 title('') 在图顶上加上标题，见例 5.8。标题内容写在单引号内，可以是中文。写中文内容时要注意，先在英文状态下写好单引号，再切换到中文状态后，在单引号内打印中文。

3. 水平或垂直标记

内置函数 xlabel('') 与 ylabel('') 分别在 X 轴与 Y 轴上写上标记，见例 5.2 与例 5.8。其内容写在单引号内，可以是中文。

4. 绘制坐标轴

内置函数 axis 是画坐标轴的，可用 help 指令来查看详情。

例 5.8　以下是添加网格、水平与垂直标记以及标题的作图例子（见图 5.7）。

```
>> x = linspace(-pi, pi);   y = sin(x);      plot(x, y);
grid; xlabel('弧度');    ylabel('正弦值');   title('Graph of y = sin(x)');
```

例 5.9　以下是添加细网格的作图例子（见图 5.8）。

```
>>x = linspace(-pi, pi);   y = sin(x);   plot(x, y);   grid minor;
```

5. 曲线“宽度”的设置

参数 LineWidth（一个词）控制曲线的“宽度”，其后必须写上宽度值，用“,”号分开。

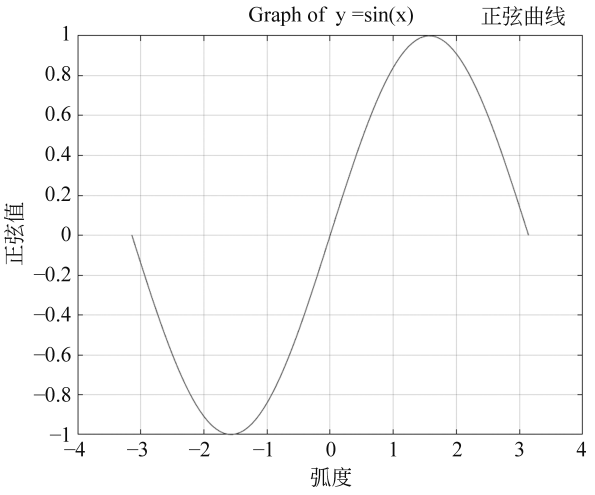


图 5.7　加了网格，x、y 坐标轴标记等的正弦曲线图

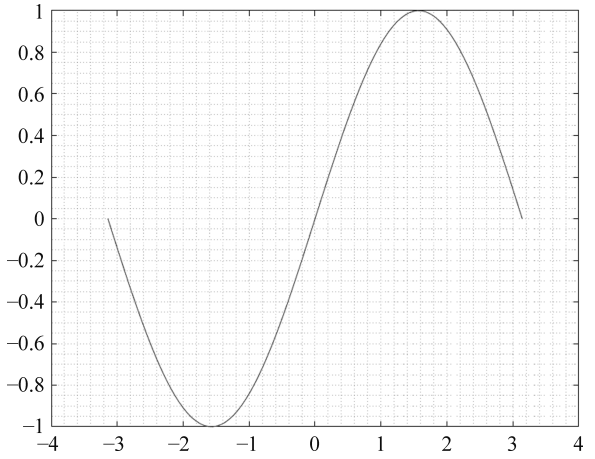


图 5.8　加上细网格的正弦曲线图

下例是用不同的宽度绘制两条曲线。

例 5.10　以下指令串画出了例 5.3 中的两条曲线，线条的类型与颜色与前例相同。这里只是加了参数，并给出不同的宽度值。从图 5.9 可以看出，宽度值为 4 的曲线比值为 3 的粗曲线。

```
>> x=linspace(0,2*pi,200);
plot(x, cos(x),' -.b','LineWidth',3);          %颜色 blue,宽度值为 3
hold on;
plot(x, sin(x-pi/2),'--g','LineWidth',4);      %颜色 green,宽度值为 4
hold off;
```

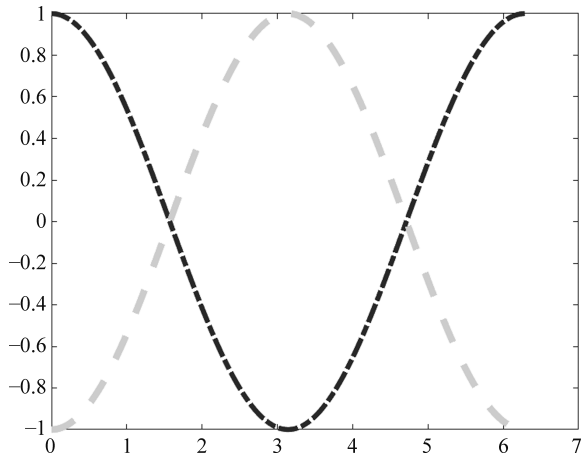


图 5.9　宽度值为 3 与 4 的两条曲线

5.1.3　用矩阵作为 plot 的参数作图

内置函数 plot(x,y)当 x、y 中 x 为向量，y 为矩阵时，且 x 的长度等于 y 的行(列)数，则 x 相对于 y 的每一行(列)作出一条曲线。

例 5.11　我们把例 5.3 中生成的三个向量合成一个矩阵，然后用来画出三条曲线。即用如下的程序画出图 5.10。注意，在程序中，并不设置曲线的颜色与类型。这里用的是默认值，曲线类型都用实线。

```
>> x=linspace(0,2*pi,200);      %产生 200 个 x 点
y=sin(x);     z=cos(x);          %计算 3 个向量的值
v=sin(x-pi/2);
Y=[y',z',v'];                    %用此指令,把这 3 个向量作为列,合并成矩阵 Y
%Y=[y;z;v];                      %或用此指令,把这 3 个向量作为行,合并成矩阵 Y。结果相同
plot(x,Y) ;                      %同时画出 3 条曲线
legend('sin(x)', 'cos(x)', 'sin(x-pi/2)') %图例
```

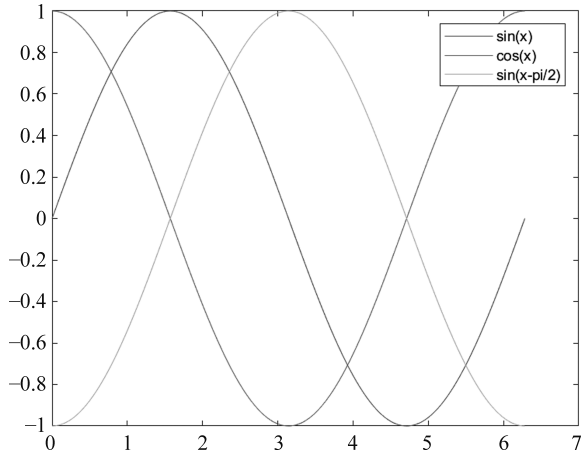


图 5.10　用矩阵表达式同时画出 3 条曲线

5.2 三维作图

三维(3D)作图包括空间曲线与曲面作图。3D作图也可以用单变量函数作图时用的内置函数grid、title、xlabel、ylabel(加上zlabel)、legend等。

5.2.1 空间曲线作图

1. 曲线方程

空间曲线作图时,所用的曲线方程应为参数方程:$x=f(t)$,$y=g(t)$,$z=h(t)$。与单变量函数作图时一样,数据$t_i(i=1,2,\cdots,n)$作为一个n维的向量,需要我们录入或产生。三列向量$(x_i|y_i|z_i)(i=1,2,\cdots,n)$则是根据参数方程与$t_i$计算出来的值。

2. 曲线作图用的内置函数

空间曲线作图时,所用的内置函数与单变量函数作图的内置函数类似,有plot3(x,y,z)与plot3(x,y,z,s),其中,s是用来设置曲线的类型、所用的画线符号与颜色的符号串。

3. 用内置函数plot3(x,y,z,s)作曲线

例5.12 以下指令串作出在参数区间$[0,10\pi]$内的一条螺旋线,其参数方程为$x=\sin(t)$,$y=\cos(t)$,$z=t$。以及过$a(-1,1,0)$与$b(1,-1,20)$两点的直线,其参数方程为$x=t/(5\pi)-1$,$y=-t/(5\pi)+1$,与$z=2t/\pi$。

得到图5.11。指令串中的legend是两个分量都是字符串的向量。另外,这里的两个plot3指令可以合并为一个指令(可去掉hold on;与hold off;)。

```
>> t = 0:pi/50:10*pi;                                  %产生数据 t,步长 pi/50
plot3(sin(t),cos(t),t, 'r* ');                         %用红色 * 作螺旋线
grid; xlabel('x=sin(t)');                              %加网格与 X 轴的标记
ylabel('y=cos(t)'); zlabel('z=t');                     %加 Y、Z 轴的标记
hold on;                                               %准备加上另一个图
%以下用蓝实线画过 a(-1,1,0), b(1,-1,20) 两点的直线
plot3([-1,1],[1,-1],[0,20], 'b- ');                    %也可以用下一参数方程
%plot3(t/(5*pi)-1,-t/(5*pi)+1,2*t/pi,'b-');
title('Helix & straight line ');                       %加上标题
legend('螺旋线','直线');                               %加上图例
hold off;
```

5.2.2 曲面作图

给出两个变量的曲面方程$z=f(x,y)$,如何作出在区域$a<x<b$,$c<y<d$内的曲面图呢?

首先,用冒号":"或"linspace"生成向量x与y的数据,再用"[X,Y] = meshgrid(x,y)"(注意,左边是大写的X、Y)生成用来计算函数值z的矩阵$\boldsymbol{X}$与$\boldsymbol{Y}$。其中,$\boldsymbol{X}$的行向量