

数据可视化

数据可视化是研究数据的视觉表现形式的技术。其中，数据的视觉表现形式被定义为一种以某种概要形式提炼的信息，包括相应信息单位的各种属性和变量。将数学公式与数据表现在图表中，是展示符号的具体物理含义及大量数据的内在联系和规律的科学有效的方法。

MATLAB 中可以绘制二维、三维的数据图形，并且通过对图形的线型、颜色、标记、观察角度、坐标轴范围等属性的设置，将大量数据的内在联系及规律表现得更加细腻、完善。

绘制图形的基本步骤如下：

(1) 准备绘图的数据。对于二维图形，需要准备横纵坐标数据；对于三维图形，需要准备矩阵参数变量和对应的 Z 轴数据。在 MATLAB 中，可以通过下面几种方法获得绘图数据。

第一种，把数据保存为文本文件，用 load 函数调入数据。

第二种，由用户自己编写命令文件得到绘图数据。

第三种，在命令窗口直接输入数据。

第四种，在 MATLAB 主窗口的“主页”选项卡中，通过“导入数据”按钮导入可以识别的数据文件。

(2) 选定绘图窗口和绘图区域。MATLAB 使用 figure 函数指定绘图窗口，默认时打开标题为 Figure 1 的绘图窗口。绘图区域如果位于当前绘图窗口中，则可以省略这一步。可以使用 subplot 函数指定当前绘图窗口的绘图子区域。

(3) 绘制图形。根据数据，使用绘图函数绘制图形。

(4) 设置图形的格式。图形的格式设置主要包括以下几方面：

- 线型、颜色和数据点标记设置。
- 坐标轴范围、标识及网格线设置。
- 坐标轴标签、图题、图例和文本修饰等设置。

(5) 输出绘制好的图形。MATLAB 可以将绘制的图形窗口保存为 .fig 文件，或者转换为别的图形文件，也可以复制图片或者打印图片等。

3.1 二维图形绘制

二维图形的绘制是 MATLAB 图形处理的基础。MATLAB 提供了丰富的绘图函数，既可以绘制基本的二维图形，又可以绘制特殊的二维图形。

MATLAB 中绘制二维图形的常用函数如表 3-1 所示。

表 3-1 二维图形绘制的常用函数

函 数	功 能	函 数	功 能
line	绘制直线或折线	plot	绘制二维平面图形
compass	绘制向量图或罗盘图	fplot	对函数自适应采样
plotyy	绘制双纵坐标二维平面图形	subplot	分区绘制子图
polar	绘制极坐标图形	figure	创建图形窗口
semilogx	半对数图 (x 轴对数刻度)	errorbar	为图形加上误差范围
semilogy	半对数图 (y 轴对数刻度)	hist	绘制累计图
loglog	全对数坐标	rose	绘制极坐标累计图
bar	绘制条形图或直方图	feather	绘制羽毛图
stairs	绘制阶梯图	quiver	绘制向量场图
stem	绘制杆图或针状图	pie	绘制饼图
fill	绘制填充图或实心图		

本节将对常用函数分类进行介绍。

3.1.1 基本绘图命令

MATLAB 提出了句柄图形学(Handle Graphics)的概念,同时为面向对象的图形处理提供了十分丰富的工具软件支持。MATLAB 在绘制图形时,其中的每个图形元素(如坐标轴或图形上的曲线、文字等)都是一个独立的对象。

用户可以对其中任何一个图形元素单独进行修改,而不影响图形的其他部分,具有这样特点的图形绘制方法称为矢量化绘图。矢量化绘图要求给每个图形元素分配一个句柄,以后再对该图形元素做进一步操作时,则只需对该句柄进行操作即可。

1. line 函数

MATLAB 允许用户在图形窗口的任意位置用绘图命令 line 画直线或折线。

line 创建一个直线对象,可以定义颜色、宽度、直线类型、标记类型以及其他的一些特征。line 函数的常用语法格式如下:

```
line(x,y)
line(x,y,z)
line
line(__,Name,Value)
line(ax,__)
pl=line(__)
```

其中 x、y 都是一维数组, line(x,y)能够把(x(i),y(i))代表的各点用线段顺次连接起来,从而绘制出一条折线。不带参数的 line 命令使用默认属性设置绘制一条从点(0,0)到(1,1)的线条。

2. plot 函数

plot 函数是 MATLAB 中最核心的二维绘图函数。它有以下 3 种格式：

```
plot(y)                % 格式 1
plot(x,y)              % 格式 2
plot(x1,y1,x2,y2,...) % 格式 3
```

格式 1：当 y 是一维数组时，`plot(y)` 把 $(i,y(i))$ 各点顺次连接起来，其中 i 的取值范围为 $1 \sim \text{length}(y)$ 。当 y 是普通的二维数组时，相当于对 y 的每一列用 `plot(y(:,i))` 画线，并把所有的折线累叠绘制在当前坐标系中。

格式 2：是 `plot` 最常用的语法格式。当 x 和 y 都是一维数组时，功能和 `line(x,y)` 类似；但 `plot` 函数中的 x 和 y 也可以是一般的二维数组，这时候就是对 x 和 y 的对应列画线。

特别地，当 x 是一个向量， y 是一个在某一方向和 x 具有相同长度的二维数组时，`plot(x,y)` 则是对 x 和 y 的每一行（或列）画线。

格式 3：对多组变量同时进行绘图。对于每一组变量，其意义同前所述。

在 `plot` 函数的以上用法中，坐标轴均为线性刻度。

【例 3-1】 绘制函数曲线 $y = 2e^{-\frac{x}{2}} \cos 2\pi x$ ，其中 $0 \leq x \leq 2\pi$ 。

```
>> x=0:pi/200:2*pi;
>> y=2*exp(-0.5*x).*cos(2*pi*x);
>> plot(x,y)
```

程序执行后，输出如图 3-1 所示的图形。

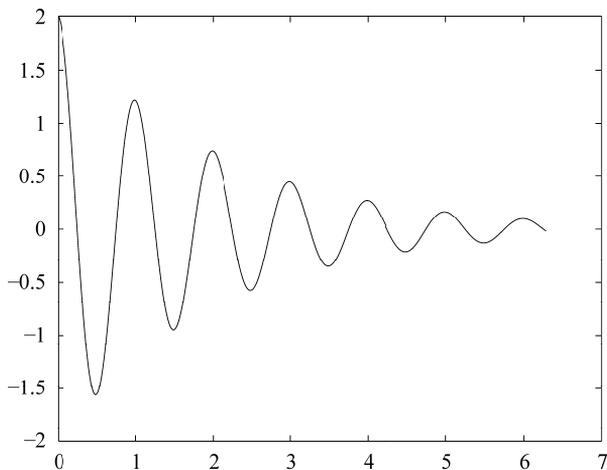


图 3-1 例 3-1 输出图形

3. fplot 函数

为了提高精度，绘制出比较真实的函数曲线，就不能等间隔采样，而必须在变化率大的区间密集采样，以充分反映函数的变化规律，进而提高图形的真实度。`fplot` 函数可自适应地对函数进行采样，能更好地反映函数的变化规律。

fplot 函数调用格式如下:

```
fplot(f) % 格式 1
fplot(f,xinterval) % 格式 2
fplot(funx,funy) % 格式 3
fplot(funx,funy,tinterval) % 格式 4
fplot(ax,____) % 格式 5
fp=fplot(____) % 格式 6
fplot(filename,lims,tol,operation) % 格式 7
```

格式 1: 在 x 的默认区间[-5, 5]绘制由函数 $y=f(x)$ 定义的曲线。

格式 2: 在 x 的指定区间绘图, 将区间指定为[xmin, xmax]形式的二元向量。

格式 3: 在 t 的默认区间[-5, 5]绘制由 $x=funx(t)$ 和 $y=funy(t)$ 定义的曲线。

格式 4: 在 t 的指定区间绘制由 $x=funx(t)$ 和 $y=funy(t)$ 定义的曲线, 将区间指定为 [tmin, tmax]形式的二元向量。

格式 5: 将图形绘制到 ax 指定的坐标区中, 而不是当前坐标区中。指定的坐标区必须作为第一个输入参数。

格式 6: 返回 FunctionLine 对象或 ParameterizedFunctionLine 对象, 具体情况取决于输入。使用 fp 查询和修改特定线条的属性。

格式 7: 在 fplot 函数的这种调用方法中, 支持用于指定误差容限或计算点数量的输入参数。但自 MATLAB R2019b 开始, 已不再支持此调用方法, 仅在旧版本中可用。其中, filename 为函数名, 以字符串形式出现。它可以是由多个分量函数构成的行向量, 分量函数可以是函数的直接字符串, 也可以是内部函数名或函数文件名, 但自变量都必须是 x。lims 为 x、y 的取值范围, 以向量形式给出。取二元向量[xmin,xmax]时, x 轴的范围被人为确定; 取四元向量[xmin,xmax, ymin,ymax]时, x、y 轴的范围被人为确定。tol 为相对允许误差, 默认值为 $2e-3$ 。

【例 3-2】 绘制函数曲线 $y=\sin(\tan\pi x)$ 。

建立函数文件 li3_2.m:

```
function y=li3_2(x)
    y=sin(tan(pi*x));
end
```

在命令行窗口中调用 li3_2 函数:

```
>> fplot(@li3_2,[0.7,1.7])
```

程序执行后, 输出如图 3-2 所示的图形。

4. bar 函数

在 MATLAB 中, 使用 bar 函数绘制柱状图。

柱状图又称条形图, 是一种以长方形的条表示数据的统计图表。柱状图用来比较两个或多个数据, 只有一个变量, 通常用于小规模数据分析。柱状图亦可横向排列, 或用多维方式表达。

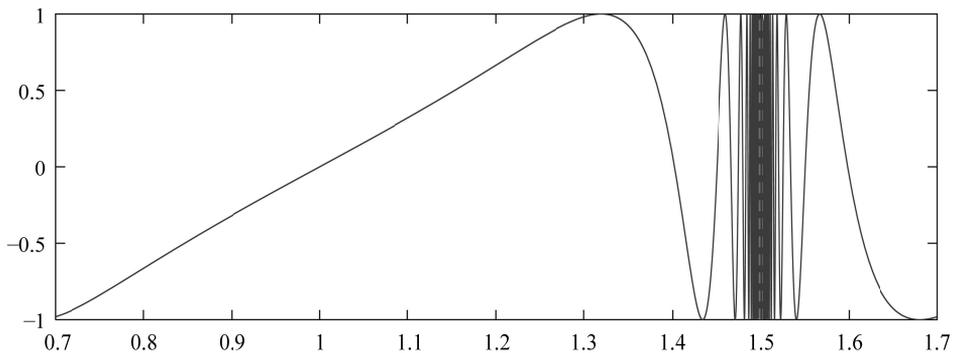


图 3-2 例 3-2 输出图形

`bar` 函数调用格式如下：

<code>bar(y)</code>	%格式 1
<code>bar(x,y)</code>	%格式 2
<code>bar(____,width)</code>	%格式 3
<code>bar(____,style)</code>	%格式 4
<code>bar(____,Name,Value)</code>	%格式 5
<code>bar(ax,____)</code>	%格式 6
<code>b = bar(____)</code>	%格式 7

格式 1：创建一个柱状图，`y` 中的每个元素对应一个条形。要绘制单个条形序列，需将 `y` 指定为长度为 `m` 的向量。这些条形沿 `x` 轴从 1 到 `m` 依次放置；要绘制多个条形序列，需将 `y` 指定为矩阵，每个条形序列对应一列。

格式 2：在 `x` 指定的位置绘制条形。

格式 3：设置条形的相对宽度以控制组中各个条形的间隔。将 `width` 指定为标量值。可将此选项与上述语法中的任何输入参数组合一起使用。

格式 4：指定条形组的样式。例如，使用 `'stacked'` 可以将每个条形组显示为一个多种颜色的条形叠合组成的。

格式 5：使用一个或多个名称-值对作为参数，指定柱状图的属性。只有采用默认的 `'grouped'` 或 `'stacked'` 样式的条形图支持设置条形属性。

格式 6：将图形绘制到 `ax` 指定的坐标区中，而不是当前坐标区中。选项 `ax` 可以位于上述语法中的任何输入参数组合之前。

格式 7：根据 `y` 的类型，返回一个或多个 `Bar` 对象。如果 `y` 是向量，则 `bar` 函数将创建一个 `Bar` 对象；如果 `y` 是矩阵，则 `bar` 函数为每个条形序列返回一个 `Bar` 对象。显示柱状图后，使用 `b` 设置条形的属性。

【例 3-3】 将矩阵 $y = \begin{bmatrix} 2 & 2 & 3 \\ 2 & 5 & 6 \\ 2 & 8 & 9 \\ 2 & 11 & 12 \end{bmatrix}$ 中的每一行显示为一个条形，每个条形的高度是行中

各元素之和。

```
>> y = [2 2 3; 2 5 6; 2 8 9; 2 11 12];
```

```
>> bar(y,'stacked')
```

执行以上操作后，输出如图 3-3 所示的图形。

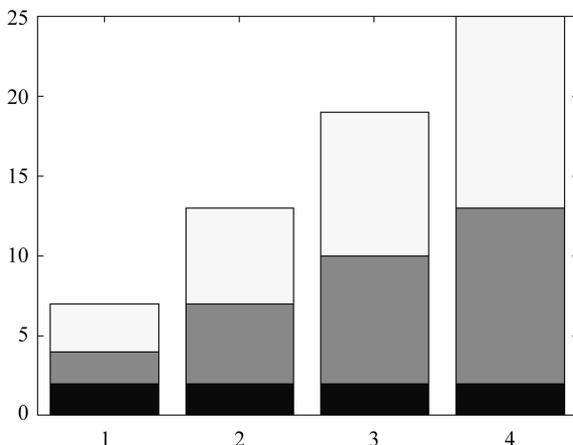


图 3-3 例 3-3 输出图形

5. stairs 函数

在 MATLAB 中，使用 stairs 函数绘制阶梯图。

阶梯图通常用于 y 值发生离散的改变，且在某个特定的 x 值位置突然变化的情况。

stairs 函数调用格式如下：

```
stairs(y) %格式 1  
stairs(x,y) %格式 2  
stairs(____,name,value) %格式 3  
stairs(ax,____) %格式 4  
h = stairs(____) %格式 5  
[xb,yb] = stairs(____) %格式 6
```

格式 1：绘制 y 中元素的阶梯图。如果 y 为向量，则 stairs 绘制一个线条；如果 y 为矩阵，则 stairs 为矩阵每一列绘制一个线条。

格式 2：在 y 中由 x 指定的位置绘制元素。x 和 y 必须是相同维数的向量或矩阵。另外，x 可以是行或列向量，y 必须是包含 length(x) 行的矩阵。

格式 3：使用一个或多个名称-值对参数修改阶梯图。例如，“'Marker','o','MarkerSize',8”指定大小为 8 磅的圆形标记。

格式 4：将图形绘制到 ax 指定的坐标区中，而不是当前坐标区中。选项 ax 可以位于前面的语法中的任何输入参数组合之前。

格式 5：返回一到多个 Stair 对象。创建特定 Stair 对象后，使用 h 更改该对象属性。

格式 6：不直接绘制图形，但返回维数相等的矩阵 xb 和 yb，以使用 plot(xb,yb) 绘制阶梯图。

3.1.2 线型、数据点型和颜色

部分二维绘图函数，如 plot 函数、fplot 函数、stairs 函数等，可以对图形的颜色、数据点型和线型进行设置，其调用格式如下：

```
plot(____,LineStyle)
fplot(____,LineStyle)
stairs(____,LineStyle)
```

若要改变线型、数据点型和颜色，需要在坐标对后面（即 LineSpec）加上相关符号，如表 3-2 所示。

表 3-2 线型、数据点型和颜色符号

线型符号	含 义	数据点型符号	含 义	颜色符号	含 义
-	实线	.	点	b	蓝色
:	点线	x	×符号	g	绿色
-.	点画线	+	加号	r	红色
--	虚线	h	六角星形	c	蓝绿色
(空白)	不画线	*	星号	m	紫红色
		s	方形	y	黄色
		d	菱形	k	黑色
		o	圆形		

【例 3-4】 绘制具有不同相位的 3 个正弦波形。对于第一个正弦波形，使用 2 磅的线宽。对于第二个正弦波形，指定带有圆形标记的红色虚线线型。对于第三个正弦波形，指定带有星号标记的蓝绿色点画线线型。

```
>> fplot(@(x) sin(x+pi/5),'Linewidth',2);
>> hold on
>> fplot(@(x) sin(x-pi/5),'--or');
>> fplot(@(x) sin(x),'-.*c')
>> hold off
```

执行以上操作后，输出如图 3-4 所示的图形。

【例 3-5】 创建一个阶梯图，将线型设置为点画线，将数据点型设置为圆形，将颜色设置为红色。

```
>> x = linspace(0,4*pi,20);
>>y = sin(x);
>> figure
>> stairs(y, '-.or')
```

执行以上操作后，输出如图 3-5 所示的图形。

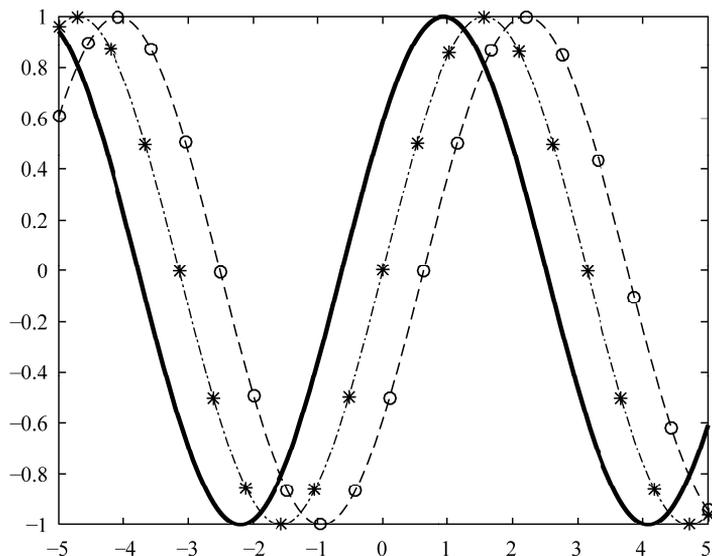


图 3-4 例 3-4 输出图形

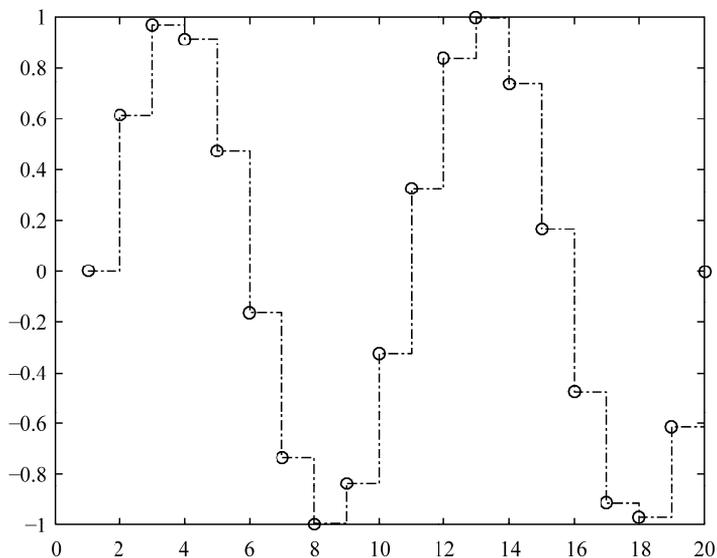


图 3-5 例 3-5 输出图形

3.1.3 多条曲线的绘制

在 MATLAB 中，有多种方法可以支持多条曲线的绘制，最简单的就是以 `plot(x1,y1,x2,y2,...`) 这样的形式直接在一个坐标系中绘制多条直线或曲线。同时，MATLAB 也提供了一些工具函数，如 `plotyy` 函数、`yyaxis` 函数等，以便更便捷地绘制多条曲线。

1. `plotyy` 函数

在 MATLAB 中利用双纵坐标 `plotyy` 函数，能把具有不同量纲、不同数量级的函数值的两个函数绘制在同一坐标系中。其调用格式如下：

```

plotyy(x1,y1,x2,y2) %格式 1
plotyy(x1,y1,x2,y2,function) %格式 2
plotyy(x1,y1,x2,y2,'function1','function2') %格式 3
plotyy(ax1,____) %格式 4
[ax,h1,h2]=plotyy(____) %格式 5

```

格式 1: 绘制 y_1 对 x_1 的图形, 在左侧显示 y_1 的标签; 同时绘制 y_2 对 x_2 的图形, 在右侧显示 y_2 的标签。

格式 2: 使用指定的绘图函数生成图形。`function` 可以是指定 `plot`、`semilogx`、`semilogy`、`loglog` 函数的句柄或字符向量 (后 3 个函数见 3.1.4 节), 也可以是符合语法 “@函数” 的任意 MATLAB 函数。函数的句柄用于访问用户定义的局部函数。

格式 3: 用 `function1(x1,y1)` 绘制左纵轴的数据, 用 `function2(x2,y2)` 绘制右纵轴的数据。

格式 4: 使用第一组数据的 `ax1` 指定的坐标区 (而不是使用当前坐标区) 绘制数据。将 `ax1` 指定为单个坐标区对象或由以前调用 `plotyy` 所返回的两个坐标区对象的向量。如果指定了向量, 则 `plotyy` 使用向量中的第一个坐标区对象。可以将此选项与前面语法中的任何输入参数组合一起使用。

格式 5: 返回 `ax` 中创建的两个坐标区的句柄以及 `h1` 和 `h2` 中每个绘图的图形对象的句柄。`ax(1)` 是左边的坐标区, `ax(2)` 是右边的坐标区。

【例 3-6】 使用两个纵轴在一个图上绘制两个数据集。对与左纵轴关联的数据使用线图, 对与右纵轴关联的数据使用针状图。

```

>>clc;
>>clear all;
>>close all;
>>x = 0:0.1:10;
>>y1 = 200*exp(-0.05*x).*sin(x);
>>y2 = 0.8*exp(-0.5*x).*sin(10*x);
>>figure
>>[hAx,hLine1,hLine2] = plotyy(x,y1,x,y2,'plot','stem')
>>hLine1.LineStyle = '--';
>>hLine2.LineStyle = ':';
>>title('Multiple Decay Rates')
>>xlabel('Time (\musec)')
>>ylabel(hAx(1),'Slow Decay'); % left y-axis
>>ylabel(hAx(2),'Fast Decay'); % right y-axis

```

执行以上操作后, 输出如图 3-6 所示的图形。

2. yyaxis 函数

`yyaxis` 函数可以创建具有两个纵轴的图形。其调用格式如下:

```

yyaxis left
yyaxis right
yyaxis(ax,____)

```

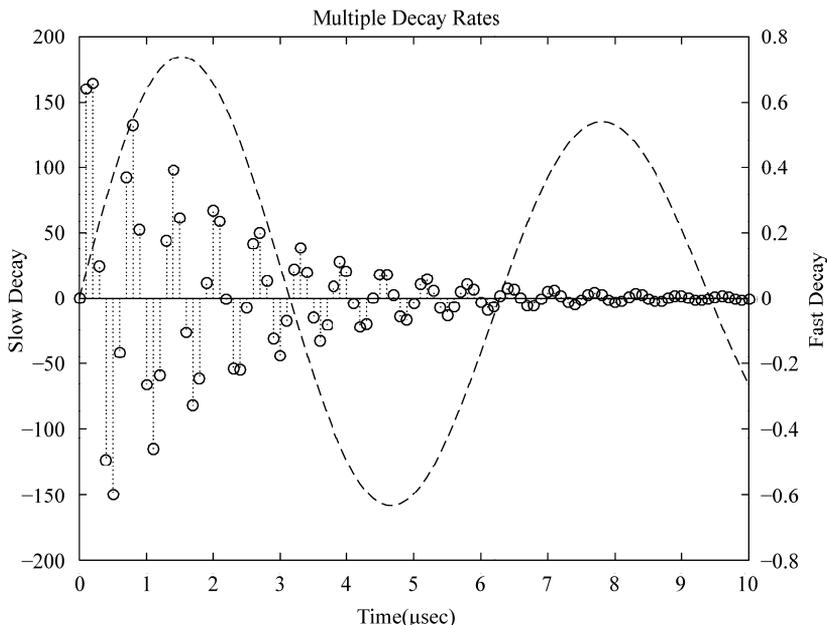


图 3-6 例 3-6 输出图形

`yyaxis left` 激活当前坐标区中与左侧纵轴关联的一侧。后续图形命令的目标为左侧。如果当前坐标区中没有两个纵轴，此命令将添加第二个纵轴。如果没有坐标区，此命令将首先创建坐标区。

`yyaxis right` 激活当前坐标区中与右侧纵轴关联的一侧。后续图形命令的目标为右侧。

`yyaxis(ax,___)` 指定 `ax` 坐标区（而不是当前坐标区）的活动侧。如果坐标区中没有两个纵轴，此命令将添加第二个纵轴。指定坐标区作为第一个输入参数。使用单引号将 `left` 和 `right` 引起来。

3.1.4 图形标注与坐标控制

1. 文字标注

文字标注是图形修饰中的重要因素，它可以是用户添加的字符说明，还可以是坐标轴上的刻度标志等。字符对象的常用属性如下：

- **Color** 属性：字符的颜色。该属性的属性值是一个 1×3 的颜色向量。
- **FontAngle** 属性：字体倾斜形式，如 `'normal'`（正常）和 `'italic'`（斜体）等。
- **FontName** 属性：字体的名称，如 `'Times New Roman'` 与 `'Courier'` 等。
- **FontSize** 属性：字号。默认以 `pt` 为单位，属性值为实数。
- **FontWeight** 属性：字体是否加粗。可以选择 `'light'`、`'normal'`（默认值）、`'demi'` 和 `'bold'` 4 个选项，其笔画逐渐变粗。
- **HorizontalAlignment** 属性：表示文字的水平对齐方式。可以有 `'left'`（按左边对齐）、`'center'`（居中对齐）、`'right'`（按右边对齐）3 种选择。
- **FontUnits** 属性：字号的单位。`'points'`（磅）为默认值，此外，还可以使用 `'inches'`

(英寸)、'centimeters'(厘米)、'normalized'(归一值)与'pixels'(像素)等。

2. 图例注释

可以通过对每一条曲线标注不同颜色和应用不同的线条来区分一张图中绘制的多条曲线。用户可以通过插入菜单的图例项(Legend)为曲线添加图例,也可以使用 legend 函数为曲线添加图例。

当在一个坐标系中绘制多个图形时,为区分各个图形, MATLAB 提供了图例注释函数。其格式如下:

```
legend(字符串 1, 字符串 2, 字符串 3, ..., 参数)
```

这里参数的意义如下:

- 0: 尽量不与数据冲突,自动放置在最佳位置。
- 1: 放置在图形的右上角。
- 2: 放置在图形的左上角。
- 3: 放置在图形的左下角。
- 4: 放置在图形的右下角。
- -1: 放置在图形视窗外的右边。

legend 函数在图形中开启一个注释视窗,按照绘图的先后顺序,依次输出字符串对各个图形进行注释。例如,字符串 1 表示第一个出现的线条,字符串 2 表示第二个出现的线条,参数用于确定注释视窗在图形中的位置。同时,注释视窗也可以用鼠标拖动,以便将其放在一个合适的位置。

3. 颜色条标注

颜色条主要用于显示图形中颜色和数值的对应关系,常用于三维图形和二维等高线图形中。

用户可以通过“插入”菜单的“颜色条”(Colorbar)命令为图形添加颜色条,也可以使用 colorbar 函数为图形添加颜色条。其常见的调用格式如下:

(1) colorbar: 在当前坐标轴的右侧添加新的垂直方向的颜色条。如果在那个位置已经存在颜色条,colorbar 函数将使用新的颜色条替代它;如果在非默认的位置存在颜色条,则保留该颜色条。

(2) colorbar('off')、colorbar('hide')和 colorbar('delete'): 删除所有与当前坐标轴相关联的颜色条。

(3) colorbar(...,'peer',axes_handle): 创建与 axes_handle 所代表的坐标轴相关联的颜色条。

(4) colorbar(...,'location'): 在相对于坐标轴的指定方位添加颜色条。如果在指定的方位存在颜色条,则它将被新的颜色条取代。location 可以是如下的值: North, 图形边框内部靠近上方的位置; South, 图形边框内部靠近下方的位置; East, 图形边框内部靠近右方的位置; West, 图形边框内部靠近左方的位置; NorthOutside, 图形边框外部靠近上方的位置; SouthOutside, 图形边框外部靠近下方的位置; EastOutside, 图形边框外部靠近右方的位置;

WestOutside, 图形边框外部靠近左方的位置。使用 4 个 Outside 值设置 location 能确保颜色条不会覆盖坐标系中的图形。

(5) `colorbar(...,'PropertyName',propertyvalue)`: 指定用来创建颜色条的坐标轴的属性名称和属性值。location 属性值仅适用于颜色条和图例, 不适用于坐标轴。

(6) `cbar_axes = colorbar(...)`: 返回新的颜色条对象的句柄。颜色条对象是当前窗口的子对象。如果颜色条已经存在, 将创建一个新的颜色条。

(7) `colorbar(cbar_handle, 'PropertyName',propertyvalue,...)`: 为 cbar_handle 所代表的颜色条对象设置属性值。要得到已存在的颜色条的句柄, 使用的命令为

```
cbar_handle = findobj(figure_handle,'tag','Colorbar')
```

其中, figure_handle 是包含颜色条的图形窗口的句柄。如果图形窗口包含多个颜色条, 返回的 cbar_handle 是一个向量, 用户需要选择指向要修改的颜色条的句柄。

4. semilogx 函数、semilogy 函数和 loglog 函数

在很多工程问题中, 通过对数据进行对数转换可以更清晰地看出数据的某些特征, 在对数坐标系中描绘数据点的曲线可以直接表现对数坐标转换, 对数转换有单轴对数坐标转换和双轴对数坐标转换两种。用 semilogx 和 semilogy 函数可以实现单轴对数坐标转换, 用 loglog 函数可以实现双轴对数坐标转换。

- loglog: 横轴和纵轴均为对数刻度 (Logarithmic scale)。
- semilogx: 横轴为对数刻度, 纵轴为线性刻度。
- semilogy: 横轴为线性刻度, 纵轴为对数刻度。

其中, 最常用的是 semilogy 函数。

5. axis 函数

图形完成后, 可以用 `axis([xmin,xmax,ymin,ymax])` 函数调整坐标轴的范围。

控制坐标轴的 axis 函数的多种调用格式如下:

- `axis([xmin,xmax,ymin,ymax])`: 指定二维图形坐标轴的刻度范围。
- `axis auto`: 设置坐标轴的自动刻度(默认值)。
- `axis manual` (或 `axis(asix)`): 保持刻度不随数据的大小而变化。
- `axis tight`: 以数据的大小为坐标轴的范围。
- `axis ij`: 设置坐标轴为矩阵轴模式, 原点在左上角, i 轴是纵轴 (从上往下标数), j 轴是横轴 (从左往右标数)。
- `axis xy`: 设置坐标轴回到直角坐标系。
- `axis equal`: 设置坐标轴刻度增量相同。
- `axis square`: 设置坐标轴长度相同, 但刻度增量未必相同。
- `axis normal`: 自动调节轴与数据的外表比例, 使其他设置生效。
- `axis off`: 使坐标轴消隐。
- `axis on`: 显示坐标轴。

3.1.5 子图绘制

为了便于多个图形的比较，MATLAB 提供了 `subplot` 函数，实现一个图形窗口绘制多个图形的功能。`subplot` 函数可以将同一个窗口分割成多个子图，能在不同坐标系绘制不同的图形，这样便于对比多个图形，也可以节省绘图空间。将窗口划分成 $m \times n$ 个子图，`subplot` 函数的调用格式如下：

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

其中， m 表示行， n 表示列， p 表示将子图画在第几行、第几列。

【例 3-7】 画出两个频率不同的正弦曲线。

```
>>close all
>>figure
>>grid on
>>t=0:0.001:1;
>>y1=sin(10*t);
>>y2=sin(15*t);
>>subplot(2,1,1)
>>plot(t,y1)
>>subplot(2,1,2)
>>plot(t,y2)
```

执行以上操作后，输出如图 3-7 所示的图形。

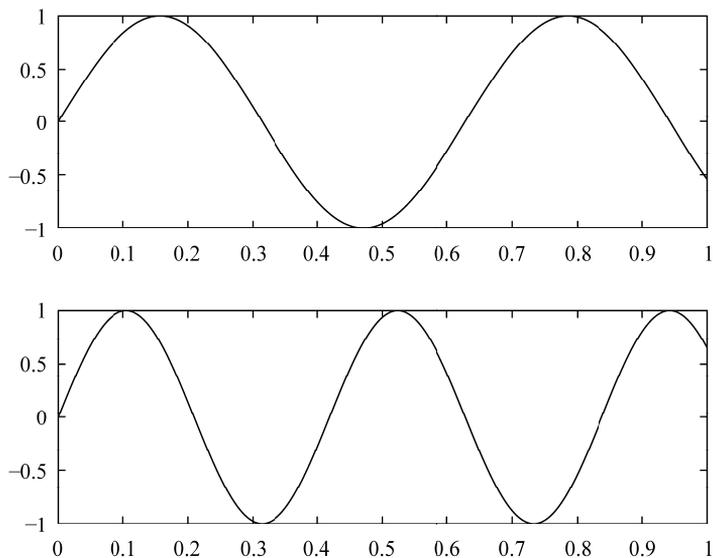


图 3-7 例 3-7 输出图形

3.2 三维图形绘制

MATLAB 具有强大的三维绘图能力，如绘制三维曲线、三维网格图和三维曲面图，并提供了大量的三维绘图函数。

3.2.1 三维曲线的绘制

三维曲线由与一组(x,y,z)坐标相对应的点连接而成。绘制三维曲线最常使用的是 `plot3` 函数，其调用格式如下：

```
plot3(x,y,z) %格式 1
plot3(x,y,z,linespec) %格式 2
plot3(x1,y1,z1,x2,y2,z2,...) %格式 3
plot3(x1,y1,z1,linespec1,x2,y2,z2,linespec2,...) %格式 4
plot3(tbl,xvar,yvar,zvar) %格式 5
plot3(ax,___) %格式 6
plot3(____,name,value) %格式 7
p=plot3(____) %格式 8
```

格式 1：绘制三维空间中的坐标。要绘制由线段连接的一组坐标，需将 `x`、`y`、`z` 指定为相同长度的向量。要在同一组坐标轴上绘制多组坐标，需将 `x`、`y` 或 `z` 中的至少一个指定为矩阵，其他指定为向量。

格式 2：使用指定的线型、数据点型和颜色创建绘图。

格式 3：在同一组坐标轴上绘制多组坐标。使用此语法作为将多组坐标指定为矩阵的替代方法。

格式 4：可为每个(x,y,z)三元组指定特定的线型、数据点型和颜色。这里可以对某些三元组指定 `LineStyle`，而对其他三元组省略它。例如，`plot3(x1,y1,z1,'o',x2,y2,z2)`对第一个三元组指定数据点型，但没有对第二个三元组指定数据点型。

格式 5：这种调用方法仅在 MATLAB R2022a 以后的版本中提供，用于绘制表 `tbl` 中的变量 `xvar`、`yvar` 和 `zvar`。要绘制一个数据集，需为 `xvar`、`yvar` 和 `zvar` 各指定一个变量；要绘制多个数据集，需为其中至少一个参数指定多个变量。对于指定多个变量的参数，指定的变量数目必须相同。

格式 6：在 `ax` 指定的目标坐标区上显示绘图。将坐标区指定为上述任一语法中的第一个参数。

格式 7：使用一个或多个名称-值对参数指定 `Line` 属性。需要在所有其他输入参数后指定 `Line` 属性。

格式 8：返回一个 `Line` 对象或 `Line` 对象数组。创建图形后，使用返回值 `p` 修改该图形的属性。

【例 3-8】 将 `t` 定义为介于 0 和 10π 之间的值组成的向量。将 `st` 和 `ct` 定义为正弦和余弦值向量。然后绘制 `st`、`ct` 和 `t` 形成的三维曲线。

```
>>t = 0:pi/50:10*pi;  
>>st = sin(t);  
>>ct = cos(t);  
>>plot3(st,ct,t)
```

执行以上操作后，输出如图 3-8 所示的图形。

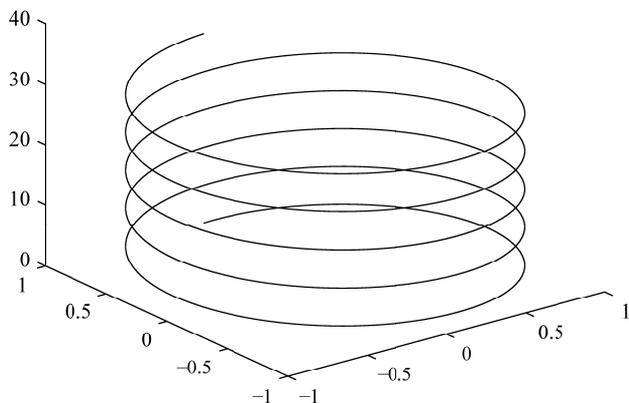


图 3-8 例 3-8 输出图形

3.2.2 三维曲面的绘制

1. mesh 函数

mesh 函数绘制由 x 、 y 和 z 指定网线面且由 c 指定颜色的三维网格图，其调用格式如下：

```
mesh(x,y,z,c)
```

x 、 y 、 z 必须均为向量或矩阵， x 、 y 是网格坐标矩阵， z 是网格点上的高度矩阵。 c 用于指定在不同高度下的颜色范围。 c 省略时，默认 $c=z$ ，即颜色的设定是正比于图形的高度的。当 x 、 y 为向量时， x 、 y 的长度分别为 m 和 n ，则 z 必须是 $m \times n$ 的矩阵。若参数中不提供 x 、 y ，则将 (i,j) 作为 z 矩阵元素 $z(i,j)$ 的 x 、 y 坐标值。

另外，mesh 函数还派生出另外两个函数——meshc 和 meshz。meshc 用来绘制带有等高线的三维网格图，meshz 用来绘制带基准平面的三维网格图，它们的用法和 mesh 类似。

2. surf 函数

surf 函数绘制三维表面图，surf 和其派生函数 surfc 是通过矩形区域观测数学函数的函数。surf 和 surfc 能够产生由 x 、 y 、 z 指定的有色曲面，即三维有色图。surf 函数调用格式如下：

```
surf(x,y,z,c)
```

其参数的定义和 mesh 函数相同。另外，surf 函数还派生出两个函数——surfc 和 surfl。surfc 用来绘制带有等高线的三维表面图，surfl 用来绘制带光照效果的三维表面图，它们的

用法和 surf 类似。

【例 3-9】 用 surf(x,y,z,c)的调用形式，绘制 peaks 函数的图形。

```
>>[x,y]=meshgrid([-3:0.1:3]);  
>>z=3*(1-x).^2.*exp(-(x.^2)-(y+1).^2)-10*(x/5-x.^3-y.^5).*exp(-x.^2-y.^2)  
-1/3*exp(-(x+1).^2-y.^2);  
>>surf(x,y,z)
```

执行以上操作后，输出如图 3-9 所示的图形。

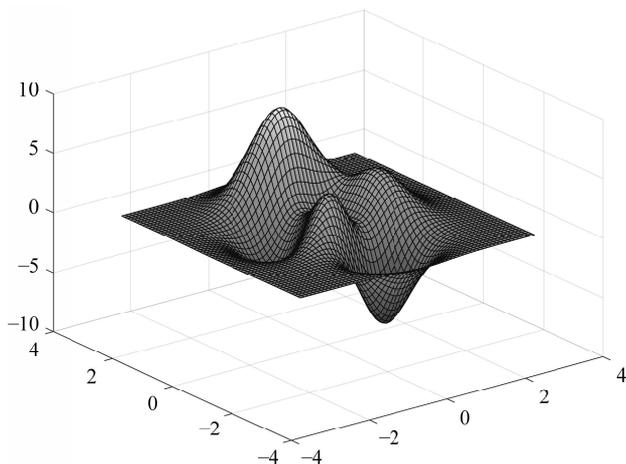


图 3-9 例 3-9 输出图形

3.3 交互式绘图工具

MATLAB 操作界面提供了丰富的工具，可以交互式探查和编辑绘图数据，以改善数据的视觉效果或显示有关数据的其他信息。

可用的交互操作取决于坐标区的内容，通常包括缩放、平移、旋转、数据提示、数据刷亮以及还原原始视图。有些类型的交互可通过坐标区工具栏进行。将鼠标指针悬停在图形区域时，工具栏会显示在坐标区的右上角。对于这些界面工具，本节不做详述。这里主要介绍常用的、与交互绘图有关的函数。

1. ginput 函数

ginput 函数的主要功能是接收来自鼠标或其他输入设备的图形输入，即手动输出点，形成曲线。其调用格式如下：

```
[x,y]=ginput(n)           %格式 1  
[x,y]=ginput             %格式 2  
[x,y,button]=ginput(...) %格式 3
```

格式 1：能够从当前轴标识 n 个点并在 x 和 y 列向量中返回这些点的 x 和 y 坐标。按 Enter 键可在输入 n 个点之前结束输入。将 n 指定为正整数。

格式 2: 可收集无限多个点, 直到按 Enter 键为止。

格式 3: 返回 x 坐标、y 坐标以及按钮或键名称。button 是一个向量, 代表鼠标按键对应的整数 (1 表示左键, 2 表示中键, 3 表示右键) 或键盘上的键的 ASCII 码值。

ginput 提高当前坐标区中的交叉线以便标识图形窗口中的点, 从而使用鼠标定位光标。图形窗口必须获得界面焦点, ginput 才能接收输入。如果图形窗口没有坐标区, 将在第一次单击或按键时创建一个坐标区。

单击某个坐标区会使该坐标区成为当前坐标区。即使已经在调用 ginput 之前设置了当前坐标区, 再单击其他坐标区, 也会使其成为当前坐标区, 并且 ginput 返回相对于该坐标区的点。如果在多个坐标区上选择了点, 则会相对于各坐标区原本所处的坐标系返回相应的结果。

2. gtext 函数

gtext 函数的主要功能是用鼠标把字符串或者字符串元胞数组放置到图形中作为文字说明。其调用格式如下:

```
gtext(arg)
```

用户用鼠标确定其放置的位置, 右击, 字符串将被放置在光标十字的“第一象限”位置上。

如果 arg 是一行字符串, 单击一次即可; 如果 arg 是多行字符串, 单击一次只能将一行放置在图形中。

3. zoom 函数

zoom 函数的主要功能是启动一个类似放大镜的光标, 在当前图形上, 可直接单击进行放大, 也可按住鼠标左键框选需要放大的区域。右击即可进行图片的缩小。

3.4 实验三: MATLAB 绘图设计

3.4.1 实验目的

1. 掌握绘制三维曲面的方法。
2. 掌握交互式绘制二维曲线的方法。

3.4.2 实验内容

1. 根据 $z = e^{-\frac{R}{10}} \cos R$ 和 $R = \sqrt{x^2 + y^2}$, 编程计算 z 与 x 、 y 的关系。
2. 绘制上面的 x 、 y 、 z 形成的三维曲面。
3. 调用 ginput 函数接收手动输入的点。

4. 利用上面接收的手动输入的点绘制二维曲线。

3.4.3 参考程序

1. 编写脚本文件，代码如下。

```
clear
a=-0.1;
rm=20;
r=-rm:0.5:rm;
[x,y]=meshgrid(r);
R=sqrt(x.^2+y.^2);
z=exp(a*R).*cos(R);
```

2. 在上面的脚本中，添加如下代码。

```
z(x<0&y<0)=nan;
figure
surf(x,y,z)
```

运行程序，输出如图 3-10 所示的图形。

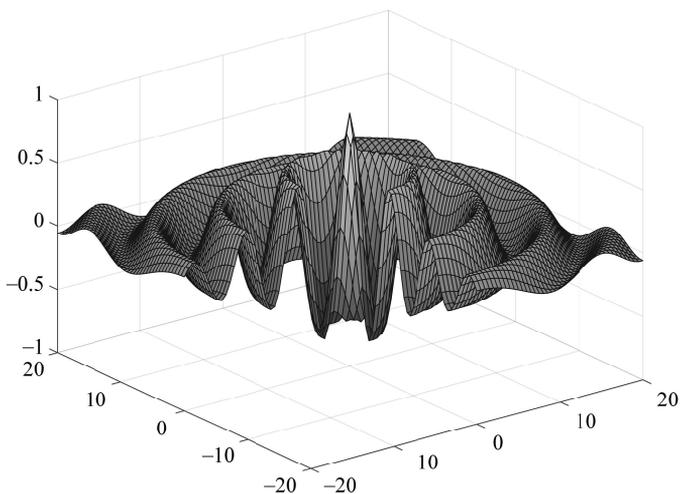


图 3-10 实验内容 2 输出图形

3. 编写脚本文件，代码如下。

```
clc;
clear all;
close all;
figure;
axis([0 10 0 10]);
hold on
x=[];
y=[];
n=0;
```

```

disp('单击鼠标左键点取需要的点');
disp('单击鼠标右键点取最后一个点');
but=1;
while but==1
    [xi,yi,but]=ginput(1);
    plot(xi,yi,'bo')
    n=n+1;
    disp('单击鼠标左键点取下一个点');
    x(n,1)=xi;
    y(n,1)=yi;
end

```

4. 在上面的脚本中，添加如下代码。

```

t=1:n;
ts=1:0.1:n;
xs=spline(t,x,ts);
ys=spline(t,y,ts);
plot(xs,ys,'r-');
hold off

```

在绘图区域内，用鼠标左键点取多个点，最后用右键点取一个点结束输入，输出如图 3-11 所示的图形。

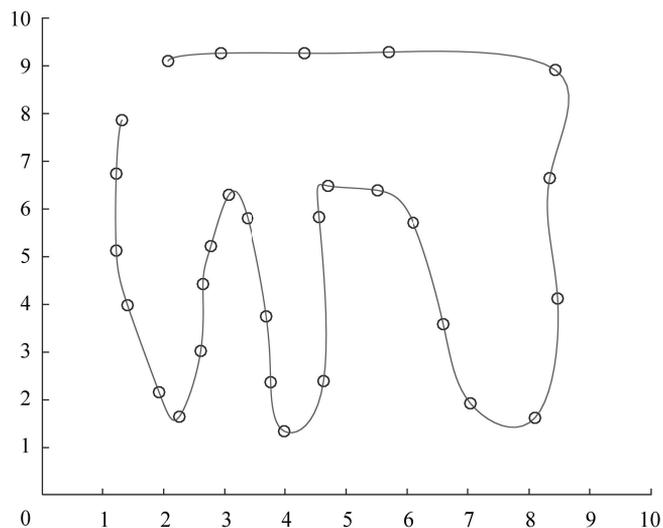


图 3-11 实验内容 4 输出图形

3.5 课程思政

本章部分内容的思政元素融入点如表 3-3 所示。

表 3-3 本章部分内容的思政元素融入点

节	思政元素融入点
3.1 二维图形绘制	在函数调用方法的讲解中,使学生了解复杂输入参数的运用及常见错误,体会到注意细节在程序设计中的重要性,培养良好的职业精神和职业素养
3.2 三维图形绘制	以近年来国产三维数据可视化软件在技术和市场上的不断突破为例,激发学生积极投身于国家发展的热情
3.4 实验三: MATLAB 绘图设计	通过编写、调试程序的操作,引导学生形成理论联系实际意识,不断增强动手解决问题的能力

练 习 三

一、选择题

1. 用 MATLAB 建立数据可视化图形,按照时间(x)记录支票余额(y)的变化曲线。当每一次发生收支时,支票余额都立即变更,而不是按照时间的推移缓缓地改变。此时最适用的绘图函数是()。

A. plot 函数 B. bar 函数 C. stairs 函数 D. pie3 函数

2. 在调用 plot 函数绘图时,希望数据点型为六角星,则要使用的符号为()。

A. s B. h C. * D. 6

3. 以下关于 plot 与 fplot 两个指令叙述错误的是()。

A. plot 只能绘制函数曲线 B. 两者都可以绘制函数曲线
C. fplot 只能绘制函数曲线 D. fplot 采用自适应算法控制点的选取

二、填空题

1. 在 MATLAB 中,使用 bar 函数可以绘制_____,使用 stairs 函数可以绘制_____,使用 pie 函数可以绘制_____。

2. 在使用 plotyy(x1,y1,x2,y2,function)这种调用方式时,function 不仅可以是函数,而且可以是_____向量。

3. 在使用 legend 函数为曲线添加图例时,调用方式为 legend(字符串 1,字符串 2,1),则图例会放置在图形的_____。

三、操作题

1. 绘制参数方程
$$\begin{cases} x = t \sin^2 2t \\ y = t \cos \frac{1}{3}t \end{cases} \quad (-\pi \leq t \leq \pi)$$
 的曲线。

2. 绘制单位矩阵的网格图。