

# 第3章

# MATLAB 基本操作

数值计算是 MATLAB 中最重要、最有特色的功能之一。MATLAB 强大的数值计算功能使其成为诸多数学计算软件中的佼佼者,同时其也是 MATLAB 软件的基础。而数组和矩阵是数值计算的最基本运算单元,在 MATLAB 中,向量可看作一维数组,而矩阵则可看作二维数组。数组和矩阵在形式上没有区别,但二者的运算性质却有很大的不同,数组运算强调的是元素对元素的运算,而矩阵运算则采用线性代数的运算方式。

## 3.1 MATLAB 数据类型

MATLAB 的基本数据单位是矩阵,而 MATLAB 数据类型的最大特点是每一种类型都以数组为基础。

数据类型是掌握任何一门编程语言都必须首先了解的内容。MATLAB R2014a 的数据类型主要有:逻辑、数值、字符串、矩阵、元胞、Java、函数句柄、稀疏以及结构等类型。数值型又分为单精度型、双精度型以及整数型。而整数类型里又分为无符号型(uint8、uint16、uint32、uint64)和符号型(int8、int16、int32、int64)两种,它们间的层次关系如图 3-1 所示。在 MATLAB 中,所有的数据不管是属于什么类型,都是以数组或矩阵的形式保存的。

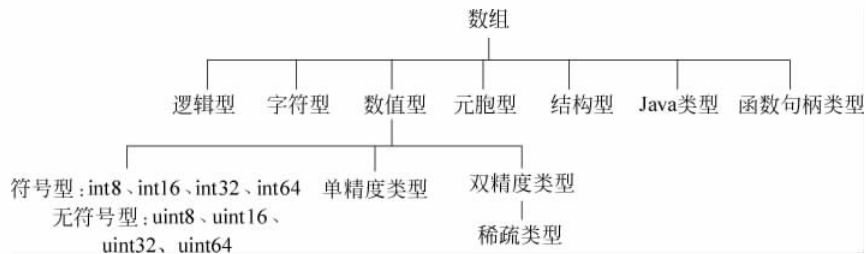


图 3-1 数据类型的层次结构图

## 3.2 MATLAB 基本元素

本节介绍常量、变量和矩阵这三种最基本且常用的数据类型以及赋值语句的基本形式。

### 3.2.1 常量

常量,在 MATLAB 中习惯称为特殊变量,即系统自定义的变量,它们在 MATLAB 启动以后驻留在内存中。在 MATLAB 中常用的特殊变量如表 3-1 所示。

表 3-1 MATLAB 常用特殊变量表

特殊变量	取值
ans	MATLAB 中运行结果的默认变量名
pi	圆周率 $\pi$
eps	计算机中的最小数
flops	浮点运算数
inf	无穷大,如 $1/0$
NaN	不定值,如 $0/0, \infty/\infty, 0 * \infty$
i 或 j	复数中的虚数单位, $i=j=\sqrt{-1}$
nargin	函数输入变量数目
narout	函数输出变量数目
realmax	最大的可用正实数
realmin	最小的可用正实数

在 MATLAB R2014a 的命令窗口中输入一个表达式或者一组数据,系统将会自动把计算的结果赋值给 ans 变量。

**注意:** A 和 a 表示的是不同的变量,读者编程时必须注意。

### 3.2.2 变量

变量是任何程序设计语言的基本元素之一, MATLAB 语言当然也不例外。与常规的程序设计语言不同的是, MATLAB 并不要求事先对所使用的变量进行声明,也不需要指定变量类型, MATLAB 语言会自动依据所赋予变量的值或对变量进行的操作来识别变量的类型。在赋值过程中,如果赋值变量已存在,则 MATLAB 将使用新值代替旧值,并以新值类型代替旧值类型。在 MATLAB 中变量的命名应遵循以下规则:

- 变量名必须以字母开头,之后可以是任意的字母、数字或下划线。
- 变量名区分字母的大小写。
- 变量名不超过 31 个字符,第 31 个字符以后的字符将被忽略。

与其他的设计语言相同,在 MATLAB 语言中也存在变量作用域的问题。在未加特殊说明的情况下, MATLAB 语言将所识别的一切变量视为局部变量,即仅在其使用

的 M 文件内有效。如果要变量定义为全局变量,则应当对变量进行说明,即在该变量前加关键字 global。一般来说,全局变量均用大写的英文字符表示。

### 3.2.3 赋值语句

MATLAB 采用命令行形式的表达式语言,每一个命令行就是一条语句,其格式与书写的数学表达式十分相近,非常容易掌握。读者在命令窗口中输入语句并回车确认后,该语句就由 MATLAB 系统解析运行,并给出运行结果。MATLAB 赋值语句有以下两种结构。

#### 1. 直接赋值语句

直接赋值语句的基本结构为:

$$\text{赋值变量} = \text{赋值表达式}$$

其中,等号右边的表达式由变量名、常数、函数和运算符构成。直接赋值语句把右边表达式的值直接赋给了左边的赋值变量,并将返回值显示在 MATLAB 命令窗口中。

**【例 3-1】** 对 A 赋值,实现  $A=3 \times 27$ 。

在 MATLAB 命令窗口中输入语句并回车。

```
>> A = 3 * 27
A =
    81
```

**注意:**

(1) 如果赋值语句后面没有分号“;”,MATLAB 命令窗口将显示表达式的运算结果;如果不想显示运算结果,则应该在赋值语句末尾加上分号“;”。

(2) 如果省略赋值语句左边的赋值变量和等号,则表达式运算结果将默认赋值给系统保留变量 ans。

(3) 如果等式右边的赋值表达式不是数值,而是字符串,则字符串两边应加单引号。

#### 2. 函数调用语句

函数调用语句的基本结构为:

$$[\text{返回变量列表}] = \text{函数名}(\text{输入变量列表})$$

其中,等号右边的函数名对应于一个存放在合适路径中的 MATLAB 文本文件。函数可以分为两大类:一类是 MATLAB 内核中已经存在的内置函数;另一类是读者根据需要自定义的函数。

返回变量列表和输入变量列表均可由若干变量名组成。

**注意:** 如果返回变量个数大于 1,则它们之间应该用逗号或空格分隔;如果输入变量个数大于 1,则它们之间只能用逗号分隔。

**【例 3-2】** 调用 cos 函数求  $a = \cos\left(\frac{\pi}{2}\right)$  的值。

在 MATLAB 命令窗口中输入语句并回车：

```
>> a = cos(pi/2)
a =
    6.1232e-17
```

**注意：**

(1) 函数名的命名规则与变量命名规则一致，读者在命名自定义函数时也必须避免与 MATLAB 已有的内置函数重名；

(2) 对于内置函数，读者可直接调用；对于自定义函数，该函数所对应的 M 文件应当存在并且保存在 MATLAB 可搜索到的目录中。

### 3.2.4 矩阵及元素

在 MATLAB 中，最基本的数据结构为元素，这是一个二维的矩形数据结构。它能够存储多个数据元素，具有非常易存取的特点。数据元素可以是数字、字符、逻辑真或假，甚至另外类型的结构。

#### 1. 矩阵的表示

用 MATLAB 语言表示一个矩阵非常容易。在 MATLAB 命令窗口中输入以下代码并回车：

```
>> A = [1 3;4 7]
A =
     1     3
     4     7
>> B = [1 7;4 6]
B =
     1     7
     4     6
>> C = [1;2;3]
C =
     1
     2
     3
```

可见矩阵变量 A、B、C 被成功赋值，可以在后续命令和函数中任意调用这几个矩阵。在输入矩阵过程中必须遵循以下规则：

- (1) 必须使用方括号“[]”包括矩阵的所有元素；
- (2) 矩阵不同的行之间必须用分号或回车键隔开；
- (3) 矩阵同一行的各元素之间必须用逗号或空格隔开。

为了方便用户使用，提高编程效率，除了最基本的直接输入方法外，MATLAB 还提供给用户一些可以直接调用的内置基本矩阵函数，有时可以成为创建矩阵的捷径。

表 3-2 为 MATLAB 提供的主要内置矩阵函数。

表 3-2 MATLAB 主要内置基本矩阵函数

函数	功 能	函数	功 能
compan	创建伴随矩阵	magic	创建魔方矩阵
diag	创建对角矩阵	ones	创建全 1 矩阵
eye	创建单位矩阵,即主对角线元素为 1,其余元素全为 0	rand	创建均匀分布随机矩阵
gallery	创建测试矩阵	randn	创建正态分布随机矩阵
hadamard	创建 Hadamard 矩阵	rosser	创建经典对称特征值测试矩阵
hilb	创建 Hilbert 矩阵	wilkinson	创建 Wilkinson 特征值测试矩阵
invhilb	创建 Hilbert 矩阵转置	zeros	创建全 0 矩阵

**【例 3-3】** 调用 MATLAB 内置函数创建矩阵。

```
>> A = rand(3,4)           % 创建一个 3 行 4 列的随机矩阵
A =
    0.8147    0.9134    0.2785    0.9649
    0.9058    0.6324    0.5469    0.1576
    0.1270    0.0975    0.9575    0.9706
>> B = ones(2,4)          % 创建一个 2 行 4 列的全 1 矩阵
B =
     1     1     1     1
     1     1     1     1
>> C = zeros(2,2)         % 创建一个 2 维全 0 矩阵
C =
     0     0
     0     0
>> D = magic(3)           % 创建一个三维魔方矩阵
D =
     8     1     6
     3     5     7
     4     9     2
>> E = eye(3,4)           % 创建一个 3 行 4 列的单位矩阵
E =
     1     0     0     0
     0     1     0     0
     0     0     1     0
```

**注意：**向量是矩阵的一种特例,前面介绍的有关矩阵的表示方法完全适用于向量,只是表示矩阵的行列数,有一个为 1。

**【例 3-4】** 创建一个行向量  $a$  和一个列向量  $b$ 。

```
>> a = [1 4 7]           % 行向量
a =
     1     4     7
>> b = [1;4;7]           % 列向量
b =
     1
     4
     7
```

MATLAB 还提供了一个便利且高效的表达式来给等步长的行向量赋值,即冒号表达式。冒号表达式的格式为:

$$X = N_1 : \text{step} : N_2$$

用于创建一维行向量  $X$ , 第一个元素为  $N_1$ , 然后每次递增 ( $\text{step} > 0$ ) 或递减 ( $\text{step} < 0$ )  $\text{step}$ , 直到最后一个元素与  $N_2$  的差的绝对值小于等于  $\text{step}$  的绝对值为止。当不指定  $\text{step}$  时, 系统默认  $\text{step} = 1$ 。

**【例 3-5】** 利用冒号法创建向量。

```
>> A = 1:5
A =
    1    2    3    4    5
>> B = 2.6:2:11.2          % 通过冒号创建数组
B =
    2.6    4.6    6.6    8.6   10.6
>> C = 2.4:1.5:10
C =
    2.4    3.9    5.4    6.9    8.4    9.9
```

在程序中,通过冒号创建一组矩阵,如果不指定  $\text{step}$ ,则系统默认为 1; 如果  $\text{step} > 0$ ,则每次递增  $\text{step}$ ,但是如果  $N_1 > N_2$  则返回空数组; 如果  $\text{step} < 0$ ,则每次递减  $\text{step}$ ,但是如果  $N_1 < N_2$ ,则返回空数组。

在 MATLAB 中还提供两个内置函数用于创建向量,分别为 `linspace` 及 `logspace` 函数。`linspace` 创建一维矩阵,和冒号的功能类似; `logspace` 建立一维矩阵,和函数 `linspace` 的功能类似。

**【例 3-6】** 利用 `logspace` 及 `linspace` 函数创建向量。

```
>> A = linspace(1,5,8)          % 创建数组
A =
    1.0000    1.5714    2.1429    2.7143    3.2857    3.8571    4.4286    5.0000
>> A1 = logspace(1,7,9)
A1 =
    1.0e+07 *
    0.0000    0.0000    0.0000    0.0002    0.0010    0.0056    0.0316    0.1778    1.0000
```

## 2. 矩阵元素表示与赋值

矩阵元素的行号和列号称为该元素的下标,是通过“( )”中的数字(行、列的标号)来标识的。矩阵元素可以通过其下标来引用,如  $A(i,j)$  表示矩阵  $A$  第  $i$  行第  $j$  列的元素。

**【例 3-7】** 获取矩阵  $A = [1 \ 5 \ 6; 3 \ 9 \ 7]$  第 2 行全部元素。

```
>> A = [1 5 6; 3 9 7]
A =
    1    5    6
    3    9    7
>> B = [A(2,1), A(2,2), A(2,3)]
B =
    3    9    7
```

注意：冒号“:”在此也能发挥很大作用。 $A(2,:)$ 表示矩阵  $A$  第 2 行全部元素, $A(:,2)$ 表示矩阵  $A$  第 2 列全部元素, $A(1,1:2)$ 表示矩阵  $A$  第 1 行第 1~2 列的全部元素。如：

```
>> B1 = A(2, :)
B1 =
     3     9     7
>> B2 = A(:, 3)
B2 =
     6
     7
>> B3 = A(1, 1:2)
B3 =
     1     5
```

### 3.3 矩阵运算

矩阵运算是 MATLAB 最重要的运算,因为 MATLAB 的运算大部分都建立在矩阵运算的基础之上。MATLAB 有三种矩阵运算类型:矩阵的代数运算、矩阵的关系运算和矩阵的逻辑运算。其中,矩阵的代数运算应用最广泛。

根据不同的应用目的,矩阵的代数运算又包括两种重要的运算形式:按矩阵整体进行运算、按矩阵单个元素的元素群运算。

#### 3.3.1 矩阵代数运算

##### 1. 矩阵的算术运算

矩阵算术运算的书写格式与普通算术运算相同,包括优先顺序规则,但其乘法和除法的定义和方法与标量截然不同。

表 3-3 为 MATLAB 矩阵的算术运算符及其说明。

表 3-3 MATLAB 矩阵的算术运算符及说明

运算符	名称	实例	说 明
+	加	$A+B$	如果 $A$ 、 $B$ 为同维数矩阵,则表示 $A$ 与 $B$ 对应元素相加;如果其中一个矩阵为标量,则表示另一个矩阵的所有元素加上该标量
-	减	$A-B$	如果 $A$ 、 $B$ 为同维数矩阵,则表示 $A$ 与 $B$ 对应元素相减;如果其中一个矩阵为标量,则表示另一矩阵的所有元素减去该标量
*	乘	$A * B$	矩阵 $A$ 与 $B$ 相乘, $A$ 和 $B$ 均可向量为向量或标量,但 $A$ 和 $B$ 的维数必须符合矩阵乘法的定义
\	左除	$A \setminus B$	方程 $A * X = B$ 的解 $X$
/	右除	$A / B$	方程 $X * A = B$ 的解 $X$
^	乘方	$A ^ B$	当 $A$ 、 $B$ 均为标量时,表示 $A$ 的 $B$ 次方幂;当 $A$ 为方阵, $B$ 为正整数时,表示矩阵 $A$ 的 $B$ 次乘积;当 $A$ 、 $B$ 均矩阵时,无定义

注意：当运算失败时 MATLAB 会提示出错。

**【例 3-8】** 矩阵的代数运算。

```

>> A = [1,2,4;3,9,7;5,4,6]
A =
     1     2     4
     3     9     7
     5     4     6
>> B = [1:3;0,11,2;6:8]
B =
     1     2     3
     0    11     2
     6     7     8
>> A + B                                % 矩阵的加运算
ans =
     2     4     7
     3    20     9
    11    11    14
>> A - B                                % 矩阵的减法
ans =
     0     0     1
     3    -2     5
    -1    -3    -2
>> A * B                                % 矩阵乘法
ans =
    25    52    39
    45   154    83
    41    96    71
>> A \ B                                % 矩阵左除
ans =
    1.4722    0.8056    1.2500
   -0.6528    1.1806   -0.8750
    0.2083   -0.2917    0.8750
>> C = [1,3;5,6];
>> A \ C                                % 矩阵左除

```

错误使用“\”，矩阵维度必须一致。

```

>> A/B                                % 矩阵右除
ans =
    1.6600   -0.0500   -0.1100
    1.3200    0.4000    0.2800
   -0.2200   -0.1500    0.8700
>> A/C

```

错误使用“/”，矩阵维度必须一致。

```
>> A ^ B
```

错误使用“^”，输入必须为标量和方阵。要按元素进行 POWER 计算，请改用 POWER (.^).

```

>> A ^ 3
ans =
    345    546    612
    995   1633   1767
    677   1060   1182

```

**注意:**

- (1) 如果  $A$ 、 $B$  两矩阵进行加、减运算, 则  $A$ 、 $B$  必须维数相同, 否则系统提示出错;
- (2) 如果  $A$ 、 $B$  两矩阵进行运算, 则前一矩阵的列数必须等于后一矩阵的行数(内维数相等);
- (3) 如果  $A$ 、 $B$  两矩阵进行右除运算, 则两矩阵的列数必须相等(实际上,  $X=B/A=A \times B^{-1}$ );
- (4) 如果  $A$ 、 $B$  两矩阵进行左除运算, 则两矩阵的行数必须相等(实际上,  $X=A \setminus B=A^{-1} \times B$ )。

**2. 矩阵的运算函数**

在 MATLAB 中除了提供运算符实现运算外, 还专门提供一些常用的矩阵运算函数, 熟悉这些函数将对读者非常有用。

表 3-4 列出了部分常用的矩阵运算函数。

表 3-4 常用的矩阵运算函数

函数	说 明
size(A)	获得矩阵 A 的行数和列数
A'	计算矩阵 A 的转置矩阵
inv(A)	计算矩阵 A 的逆矩阵
length(A)	计算矩阵 A 的长度(列数)
sum(A)	如果 A 为向量, 则计算 A 所有元素之和; 如果 A 为矩阵, 则产生一行向量, 其元素分别为矩阵 A 各列元素之和
max(A)	如果 A 为向量, 则求出 A 所有元素的最大值; 如果 A 为矩阵, 则产生一行向量, 其元素分别为矩阵 A 各列元素的最大值
min(A)	如果 A 为向量, 则求出 A 所有元素的最小值; 如果 A 为矩阵, 则产生一行向量, 其元素分别为矩阵 A 各列元素的最小值

**【例 3-9】 常用矩阵运算函数实例。**

```
>> clear all; % 清除工作空间中的所有变量
>> X = [5, 3.4, 72, 28/4, 3.61, 17 94 89];
length(X)
ans =
     8
>> size(X)
ans =
     1     8
>> inv(X)
```

错误使用 inv, 矩阵必须为方阵。

```
>> A = magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
```

```

>> inv(A)
ans =
    0.1472    -0.1444    0.0639
   -0.0611     0.0222     0.1056
   -0.0194     0.1889    -0.1028

>> A'
ans =
     8     3     4
     1     5     9
     6     7     2

>> max(X)
ans =
    94

```

### 3. 矩阵元素群运算

元素群运算,是指矩阵中的所有元素按单个元素进行运算。为了与矩阵作为整体的运算符相区别,元素群运算约定:在矩阵运算符“\*”、“/”、“\”、“^”前加一个点符号“.”,以表示在做元素群运算,而非矩阵运算。元素群加、减运算的效果与矩阵加、减运算是一致的,运算符也相同。

表 3-5 为矩阵元素群运算符及说明。

表 3-5 矩阵元素群运算符及说明

运算符	名称	实例	说 明
. *	元素群乘	$A.*B$	矩阵 $A$ 与 $B$ 对应元素相乘, $A$ 和 $B$ 必须为同维矩阵或其中之一为标量
.\	元素群左除	$A.\backslash B$	矩阵 $B$ 除以矩阵 $A$ 的对应元素, $A$ 和 $B$ 必须为同维矩阵或其中之一为标量
./	元素群右除	$A./B$	矩阵 $A$ 除以矩阵 $B$ 的对应元素, $A$ 和 $B$ 必须为同维矩阵或其中之一为标量
.^	元素群乘方	$A.^B$	矩阵 $A$ 的各元素与矩阵 $B$ 的对应元素的乘方运算, 运算结果 $C=A.^B$ , 其中 $C(i,j)=A(i,j)^B(i,j)$ , $A$ 和 $B$ 必须为同维矩阵

**【例 3-10】** 矩阵元素群运算实例。

```

>> A=[3 8;2 7];
>> B=[3 9;11,2];
>> A.*B
ans =
     9    72
    22    14

>> A.\B
ans =
    1.0000    1.1250
    5.5000    0.2857

>> A./B
ans =
    1.0000    0.8889
    0.1818    3.5000

```

```
>> A.^3
ans =
    27    512
     8    343
```

#### 4. 元素群函数

MATLAB 提供了几乎所有初等函数,包括三角函数、对数函数、指数函数和复数运算函数等。大部分的 MATLAB 函数运算都是分别作用于函数变量(矩阵)的每一个元素,这意味着这些函数的自变量可以是任意阶的矩阵。

表 3-6 列出了 MATLAB 常用初等函数名及其对应的说明。

表 3-6 MATLAB 常用初等函数名及说明

函 数 名	说 明
sin	正弦函数(角度单位为弧)
cos	余弦函数(角度单位为弧)
tan	正切函数(角度单位为弧)
abs	求实数绝对值或复数的模
sqrt	平方根函数
angle	求复数的复角
real	求复数的实部
imag	求复数的虚部
conj	求复数的共轭
exp	自然指数函数(以 e 为底)
log	自然对数函数(以 e 为底)
log10	以 10 为底的对数函数

#### 【例 3-11】 元素群的函数实例。

```
>> x = [0, pi/6, pi/4, pi/3];
>> y = tan(x)
y =
     0     0.5774     1.0000     1.7321
>> y1 = cos(x)
y1 =
     1.0000     0.8660     0.7071     0.5000
>> log10(x)
ans =
    -Inf    -0.2810    -0.1049     0.0200
>> Z = [ 1 - 1i  2 + 1i  3 - 1i  4 + 1i
        1 + 2i  2 - 2i  3 + 2i  4 - 2i
        1 - 3i  2 + 3i  3 - 3i  4 + 3i
        1 + 4i  2 - 4i  3 + 4i  4 - 4i ]; % 复数矩阵
>> angle(Z)
ans =
    -0.7854     0.4636    -0.3218     0.2450
     1.1071    -0.7854     0.5880    -0.4636
    -1.2490     0.9828    -0.7854     0.6435
     1.3258    -1.1071     0.9273    -0.7854
```

```

>> imag(Z)
ans =
    -1     1    -1     1
     2    -2     2    -2
    -3     3    -3     3
     4    -4     4    -4
>> abs(Z)
ans =
    1.4142    2.2361    3.1623    4.1231
    2.2361    2.8284    3.6056    4.4721
    3.1623    3.6056    4.2426    5.0000
    4.1231    4.4721    5.0000    5.6569

```

### 3.3.2 矩阵的关系运算

关系运算符主要用于比较数、字符串、矩阵之间的大小或不等式关系，其返回值为 0 或为 1。常用的关系运算符如表 3-7 所示。

表 3-7 MATLAB 语言的关系运算符

关系操作符	说 明	对应的函数
==	等于	eq(A,B)
~=	不等于	ne(A,B)
<	小于	lt(A,B)
>	大于	gt(A,B)
<=	小于等于	le(A,B)
>=	大于等于	ge(A,B)

**注意：**表 3-7 中的比较运算符都是双操作数运算符，两个操作数是大小相同的数组，或者其中一个为标量。例如， $A > a$ 、 $a > A$  ( $a$  为标量) 都是有效的，其意义为  $A$  中所有元素分别与  $a$  作比较。

**【例 3-12】** 比较矩阵  $A$ 、 $B$  与标量  $a$  的大小关系。

```

>> clear all;
>> A=[1 2 3;4 5 8;9 7 6];
>> B=[1 4 7;2 5 8;3 6 9];
>> a=2;
>> A==B           % 比较矩阵大小
ans =
     1     0     0
     0     1     1
     0     0     0
>> A==a           % 比较矩阵 A 与标量 a 关系
ans =
     0     1     0
     0     0     0
     0     0     0
>> B==a           % 比较 B 与标量 a 关系
ans =

```

```

0     0     0
1     0     0
0     0     0
>> A>B           % 比较 A、B 大小关系
ans =
0     0     0
1     0     0
1     1     0

```

### 3.3.3 矩阵的逻辑运算

逻辑运算符主要用于逻辑表达式和进行逻辑运算,参与运算的逻辑量以“0”代表“假”,以任意非“0”元素代表“真”。逻辑表达式和逻辑函数的值以“0”表示“假”,以“1”表示“真”。常用的逻辑运算符如表 3-8 所示。

表 3-8 MATLAB 中的逻辑操作符

逻辑操作符	说 明	对应的函数
&	逻辑与	and(A,B)
	逻辑或	or(A,B)
~	逻辑非	nor(A,B)
	先决或	—
&&	先决与	—

**注意:** &、|、&&、||、xor、~的操作数也可以是非逻辑矩阵的数值矩阵,但是, MATLAB 会首先将其转换为逻辑矩阵,非 0 元素转换为逻辑 1,0 转换为逻辑 0,然后按照逻辑运算法则进行运算。逻辑运算符同样支持矩阵与标量的逻辑运算,其意义为各元素与标量分别作逻辑运算。

&、&& 执行相同的运算,都是逻辑与,其结果相同,但两者运算方式不同。 $A \& B$  首先分别计算出  $A$ 、 $B$ , 然后进行逻辑与;  $A \&\& B$  首先计算  $A$ , 如  $A$  的某一元素为 0, 则结果的对应元素为 0, 而不用计算  $B$  的对应元素。当  $A$  计算比较简单、 $B$  很复杂时, 采用  $\&\&$  会提高运算效率。|、|| 的区别与之类似。

**【例 3-13】** 矩阵的逻辑运算。

```

>> clear all;
A=[5 4 3;0 8 9;3 6 7];B=[1 5 7;3 9 7;0 2 4];
A|B           % 同维矩阵的逻辑或运算
ans =
1     1     1
1     1     1
1     1     1
>> a=4;A=[3 4;5 9];a&A           % 标量与矩阵的逻辑与运算
ans =
1     1
1     1
>> ~A
ans =

```

```
0 0 0
1 0 0
0 0 0
```

## 3.4 MATLAB 流程控件

作为计算机语言,编程是必需的。编程靠的是程序控制语句。计算机语言程序控制模式主要有三大类:顺序结构、循环结构和选择结构。这一点 MATLAB 与其他编程语言完全一致。

### 3.4.1 顺序结构

MATLAB 程序结构中最基本的结构即是顺序结构,这种结构不需要任何流程控制语句,完全是依照从前到后的自然顺序执行代码。顺序结构符合一般的逻辑思维顺序习惯,简单易读、容易理解。所有的实际程序代码中都会出现顺序结构。

**【例 3-14】** 使用 MATLAB 顺序结构,计算两数的和、差。

```
>> clear all;
% 输入第一个数值
num1 = 9;
% 输入第二个数值
num2 = 12;
% 计算两个数的和
disp('两个数的和为: ')
s = num1 + num2
% 计算两个数的差
disp('两个数的差为: ')
d = num1 - num2
```

运行程序,输出如下:

```
两个数的和为:
s =
    21
两个数的差为:
d =
    -3
```

### 3.4.2 循环结构

循环结构是指重复执行某一段相同的语句,用循环控制结构。如果已知循环次数,用 for 语句;如果未知循环次数,但有循环条件,则用 while 语句。

#### 1. for 语句

for 循环语句的结构如下:

```
for index = values
```

```

    program statements
        :
end

```

其中, index 表示循环变量, values 一般为使用冒号进行步进的等差数列 [start: increment: end], statements 为循环体, 最后是关键字 end。通过 for 循环语句的基本结构, 可以看出, 使用 for 语句控制循环结构, 其循环次数是一定的, 由 values 列数决定, 即  $(\text{end}-\text{start})/\text{increment}$ 。

**【例 3-15】** 从自然数 1 开始累加, 加数为自然数的质数因子最小数, 直到累加和达到 99 时停止累加, 返回累加和于停止的位置。

```

>> clear all;
for m = 1:k
    for n = 1:k
        if m == n
            a(m, n) = 2;
        elseif abs(m - n) == 2
            a(m, n) = 1;
        else
            a(m, n) = 0;
        end
    end
end

```

当  $k=6$  时, 得到一个矩阵:

```

a =
    2     0     1     0     0     0
    0     2     0     1     0     0
    1     0     2     0     1     0
    0     1     0     2     0     1
    0     0     1     0     2     0
    0     0     0     1     0     2

```

## 2. while 循环

与 for 循环以固定次数求一组命令的值相反, while 循环以不定的次数求一组语句的值。while 循环的一般调用格式为:

```

while expression
    statements
end

```

当表达式 expression 的结果为真时, 就执行循环语句, 直到表达式 expression 的结果为假, 才退出循环。

如果表达式 expression 是一个数组 A, 则相当于判断  $\text{all}(A)$ 。注意空数组被当做逻辑假, 循环不执行。

**【例 3-16】** 利用 while 循环结构求方程  $x^3 - 2x - 5$  的解。

```

>> clear all;

```

```

a = 0;
fa = -Inf;
b = 3;
fb = Inf;
while b - a > eps * b
    x = (a + b) / 2;
    fx = x^3 - 2 * x - 5;
    if fx == 0
        break
    elseif sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
disp('方程的解为:')
disp(x)

```

运行程序,输出如下:

```

方程的解为:
    2.0946

```

### 3. break 语句和 continue 语句

与循环结构相关的语句还有 break 语句和 continue 语句。它们一般与 if 语句配合使用。

break 语句用于终止循环的执行。当在循环体内执行到该语句时,程序将跳出循环,继续执行循环语句的下一语句。

continue 语句控制跳过循环体中的某些语句。当在循环体内执行到该语句时,程序将跳过循环体中所有剩下的语句,继续下一次循环。

**【例 3-17】** 编写求 0~50 范围内 3 与 5 的公倍数的程序。

```

>> clear all;
% 输出 0~50 范围内不能被 3 和 5 的公倍数
disp('输出 0~50 范围内能同时被 3 和 5 整除的数')
for n = 0:50
    if mod(n,3) == 0;           % 当 n 不能整除时,跳出 if 语句
        if mod(n,5) ~= 0
            continue          % 当 n 可以被 3 整除时,但不能被 5 整除,跳出此行 if 语句
        end
        disp(n)
    end
end
end

```

运行程序,输出如下:

```

输出 0~50 范围内能同时被 3 和 5 整除的数
    0
    15
    30
    45

```

**【例 3-18】** 求解经典的鸡兔同笼问题,在笼子中有头 36 个,脚有 100 只,求鸡兔各几只。其实现的 MATLAB 代码如下:

```
>> clear all;
i = 1;
while i > 0
    if rem(100 - i * 2, 4) == 0 & (i + (100 - i * 2) / 4) == 36;
        break;
    end
    i = i + 1;
    n1 = i;
    n2 = (100 - 2 * i) / 4;
end
fprintf('The number of chicken is %d.\n', n1);
fprintf('The number of rabbit is %d.\n', n2);
```

运行程序,输出如下:

```
The number of chicken is 22.
The number of rabbit is 14.
```

### 3.4.3 选择结构

在 MATLAB 中选择结构程序有两种形式,分别为 if 形式和 Switch 形式。

#### 1. if 条件选择结构

在编写程序时,往往需要根据一定的条件进行一定的选择来执行不同的语句,此时,需要使用分支语句来控制程序的进程。在 MATLAB 中,使用 if-else-end 结构来实现这种控制。

if-else-end 结构的使用形式有以下三种:

##### 1) 只有一种选择情况

此时的 if 程序结构如下:

```
if 表达式
    执行语句
end
```

这是 if 结构最简单的一种应用形式,其只有一个判断语句,当表达式为真时,即执行 if 和 end 间的执行语句;否则不予执行。

##### 2) 有两种选择情况

假如有两种选择,if-else-end 的结构如下:

```
if 表达式
    执行语句 1
else
    执行语句 2
end
```

## 3) 有三种或三种以上选择

当有三种或三种以上选择时,if-else-end 结构采用形式如下:

```
if 表达式 1
    表达式为真时的执行语句 1
else if 表达式 2
    表达式 2 为真时的执行语句 2
else if 表达式 3
    表达式 3 为真时的执行语句 3
else if ...
    ...

else
    所有表达式都为假时的执行语句
end
```

**注意:**

- (1) else 子句不能单独使用,必须与 if 配对使用;
- (2) if 条件选择结构可以嵌套使用。

**【例 3-19】** 利用分支语句 if-else 语句实现输入一个百分制成绩,要求输出成绩的等级为 A、B、C、D、E。其中 90~100 分为 A,80~89 分为 B,70~79 分为 C,60~69 分为 D,60 分以下为 E。

```
>> clear;
disp('if_else 语句!')
x = input('请输入分数:');
if (x <= 100 & x >= 90)
    disp('A')
elseif (x >= 80 & x <= 89)
    disp('B')
elseif (x >= 70 & x <= 79)
    disp('C')
elseif (x >= 60 & x <= 69)
    disp('D')
elseif (x < 60)
    disp('E')
end
```

运行程序,输出如下:

```
if_else 语句!
请输入分数:55
E
```

## 2. switch 条件选择结构

switch...case 语句适用于条件多而且比较单一的情况,类似于一个数控的多个开关。它的基本组成结构的语法格式为:

```
switch 条件表达式
    case 常量 1
```

```

    语句组 1
    case{常量 1, 常量 2}
    语句组 2
    ...
    otherwise
    语句组 n+1
end

```

执行过程为,首先计算表达式的值,并与各 case 语句中的常量比较,然后选择第一个与之匹配的 case 语句组执行,完成后立即跳出语句;若没有找到与条件表达式值相匹配的 case 语句,则执行 otherwise 后的语句组,并退出 switch 语句。

**【例 3-20】** 用 switch...case 实现输入一个百分制成绩,要求输出成绩的等级为 A、B、C、D、E。其中 90~100 分为 A,80~89 分为 B,70~79 分为 C,60~69 分为 D,60 分以下为 E。

```

>> clear all;
disp(' switch 语句!')
c = input('请输入成绩:');
switch c
    case num2cell(90:100),
        disp('A');
    case num2cell(80:89),
        disp('B');
    case num2cell(70:79),
        disp('C');
    case num2cell(60:69),
        disp('D');
    otherwise
disp('E');
end

```

运行程序,输出如下:

```

switch 语句!
请输入成绩:100
A

```

**注意:** MATLAB 中,switch 条件选择结构只执行第一个匹配的 case 对应的语句组,因此不需要 break。

## 3.5 M 文件

M 文件可分为脚本 M 文件(简称脚本文件)和函数 M 文件(简称函数文件)两大类,其特点和适用领域均不同。

### 3.5.1 脚本文件

MATLAB 命令类似于 DOS 命令,而脚本文件类似于 DOS 系统中的 .bat 批处理文件。脚本文件是一连串 MATLAB 命令,可以将烦琐的计算或操作放在一个 M 文件里

面,每当调用这一连串命令时,只需输入 M 文件名即可,从而简化了操作。运行脚本文件后,所产生的变量都保存在 MATLAB 的工作空间中,除非用户使用 clear 函数清除或关闭 MATLAB,否则这些变量将一直保存在工作空间中。

命令脚本文件包括两部分:注释部分与程序部分。其中注释部分必须在符号“%”之后,MATLAB 不对其进行计算,只帮助程序设计人员和读者理解程序。程序部分即为程序中一般的命令行和程序段,MATLAB 要对其进行编译和计算。

**【例 3-21】** 编写脚本文件,实现图像的绘制,效果如图 3-2 所示。

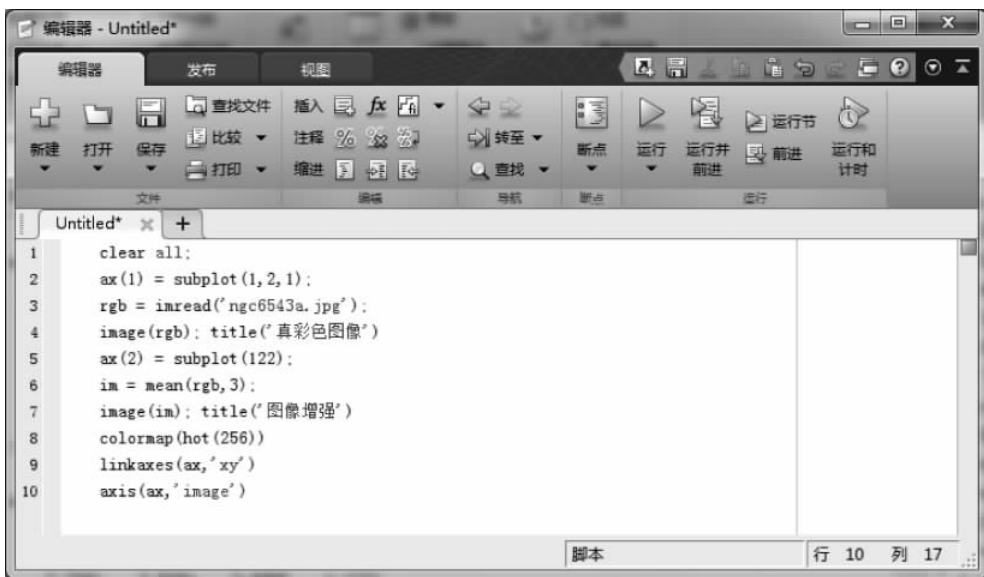


图 3-2 新建脚本文件并输入语句

选中所编写的文件右击,在弹出的快捷菜单中选择“执行所选内容”选项,即可运行程序,效果如图 3-3 所示。

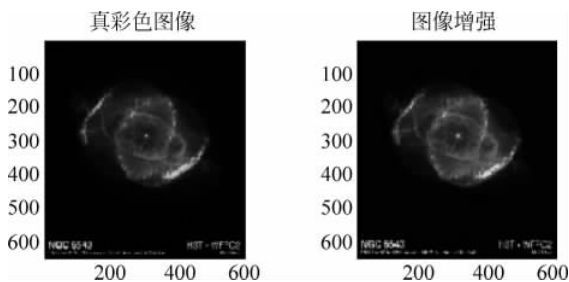


图 3-3 图像绘制

### 3.5.2 函数文件

MATLAB 用户可以根据编辑需要,编写所需要的 M 文件,它可以像 MATLAB 提供的库函数一样方便调用。这种用 MATLAB 语言创建与定义新函数的功能,体现了

MATLAB 语言强大的扩展性。

用户自定义的 M 函数有输入与输出变量,其一般格式为:

```
function 返回变量 = 函数名(输入变量)
% 注释说明语句
程序段
```

**注意:** M 函数文件第一行必须以关键字 function 作为引导,文件名必须为 \*.m。程序中的变量不保存在工作空间中,只在函数运行期间有效。

**【例 3-22】** 自定义函数用于判断读入图像的格式。

```
function imageData = readImage(filename)
try
    imageData = imread(filename);
catch exception
    % 无法找到该文件
    if ~exist(filename,'file')
        % 检查扩展中常见的拼写错误
        [~,~,extension] = fileparts(filename);
        switch extension
            case '.jpg'
                altFilename = strrep(filename, '.jpg', '.jpeg');
            case '.jpeg'
                altFilename = strrep(filename, '.jpeg', '.jpg');
            case '.tif'
                altFilename = strrep(filename, '.tif', '.tiff');
            case '.tiff'
                altFilename = strrep(filename, '.tiff', '.tif');
            otherwise
                rethrow(exception);
        end
        % 与修改过的文件名再试一次
        try
            imageData = imread(altFilename);
        catch exception2
            % 重新抛出原来的错误
            rethrow(exception)
        end
    else
        rethrow(exception)
    end
end
```

函数文件具有如下特点:

#### 1) 函数声明行

函数声明行定义了函数的名称。函数首行以关键字 function 开头,并在首行中列出全部输入、输出参量以及函数名。函数名应置于等号右侧,虽没作特殊要求,但一般函数名与对应的 M 文件名相同。输出参量紧跟在 function 之后,常用方括号括起来(若仅有一个输出参量则无须方括号);输入参量紧跟在函数名之后,用圆括号括起来。如果函数有多个输入或输出参数,输入变量之间用“,”分隔,返回变量用“,”或空格分隔。与输入或输出参数相关的两个特殊变量是 varargin 和 varargout,它们都是单元数组,分别获取

输入和输出的各元素内容。这两个参数对可变输入或输出参数特别有用。

#### 2) H1 行

H1 行是函数帮助文本的第一行,以“%”开头,用来概要说明该函数的功能。在 MATLAB 中用命令 lookfor 查找某个函数时,查找到的就是函数 H1 行及其相关信息。

#### 3) 函数帮助文本

在 H1 行之后而在函数体之前的说明文本就是函数的帮助文本。它可以有多行,每行均以“%”开头,用于比较详细地对该函数进行注释,说明函数的功能与用法、函数开发与修改的日期等。在 MATLAB 中用命令“help+函数名”查询帮助时,就会显示函数 H1 行与帮助文本的内容。

#### 4) 函数体

函数体是函数的主要部分,是实现该函数功能、进行运算所有程序代码的执行语句。

#### 5) 函数注释

函数体中除了进行运算外,还包括函数调用与程序调用的必要注释。注释语句段每行用“%”引导,“%”后的内容不执行,只起注释作用。

此外,函数结构中一般都应有变量检测部分。如果输入或返回变量格式不正确,则应该给出相应的提示。输入和返回变量的实际个数分别用 nargin 和 nargout 两个 MATLAB 保留变量给出,只要进入函数, MATLAB 就将自动生成这两个变量。nargin 和 nargout 可以实现变量检测。

如其他程序语言一样, MATLAB 也有子函数(subfunction)的概念。一个 M 文件中的第一个函数为主函数,其函数名就是调用 M 文件的文件名,而同一个文件中的其他函数则为子函数,这些子函数只对同一个文件中的主函数和其他子函数有效。

## 3.6 MATLAB 图形绘制

MATLAB 除了强大的数值分析功能外,还具有方便的绘图功能。利用 MATLAB 丰富的二维、三维图形函数和多种修饰方法,只要指定绘图方式并提供绘图数据,就可以绘制出理想的图形。由于 MATLAB 的图形系统是建立在诸如线、面等图形对象的集合基础之上,因此用户可以对任何一个图形元素进行单独的修改,而不影响图形的其他部分。

### 3.6.1 二维图形绘制

在 MATLAB 中,对于一般绘图及特殊绘图都提供了相应的内置函数,并为图形的修饰提供了函数,下面分别给予介绍。

#### 1. 基本绘图函数

MATLAB 中最常用的绘图函数为 plot,它用于绘制二维曲线,根据函数输入参数不同,其调用格式也不相同,其调用格式主要有:

plot(Y): 其中输入参数 Y 就是 Y 轴的数据,一般习惯输入向量,则 plot(Y) 可以用

以绘制索引值所对应的行向量  $Y$ , 若  $Y$  为复数, 则  $\text{plot}(Y)$  等于  $\text{plot}(\text{real}(Y), \text{image}(Y))$ 。在其他几种使用方式中, 如果有复数出现, 则复数的虚数部分将不被考虑。

$\text{plot}(X_1, Y_1, \dots, X_n, Y_n)$ : 当  $X_i, Y_i$  均为实数向量, 且为同维向量(可以不是同型向量), 则  $\text{plot}$  先描出点  $(X(i), Y(i))$ , 然后用直线依次相连; 若  $X_i, Y_i$  为复数向量, 则不考虑虚数部分。若  $X_i, Y_i$  均为同型实数矩阵, 则  $\text{plot}(X_i, Y_i)$  依次画出矩阵的几条线段; 若  $X_i, Y_i$  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。

$\text{plot}(X_1, Y_1, \text{LineStyle}, \dots, X_n, Y_n, \text{LineStyle})$ :  $\text{LineStyle}$  为选项(开关量)字符串, 用于设置曲线颜色、线型、数据点等;  $\text{LineStyle}$  的标准设定值见表 3-9, 前 7 种颜色依序(蓝、绿、红、青、品红、黄、黑)自动着色。

表 3-9 常用的绘图选项

选 项	含 义	选 项	含 义
—	实线	.	用点号标出数据点
--	虚线	O	用圆圈标出数据点
:	点线	x	用叉号标出数据点
-.	点划线	+	用加号标出数据点
r	红色	s	用小正方形标出数据点
g	绿色	D	用菱形标出数据点
b	蓝色	V	用下三角标出数据点
y	黄色	^	用上三角标出数据点
m	洋红	<	用左三角标出数据点
c	青色	>	用右三角标出数据点
w	白色	H	用六角形标出数据点
k	黑色	P	用五角形标出数据点
*	用星号标出数据点	—	—

$\text{plot}(X_1, Y_1, \text{LineStyle}, \text{'PropertyName'}, \text{PropertyValue})$ : 对所有用  $\text{plot}$  函数创建的图形进行属性值设置, 常用属性如图 3-10 所示。

表 3-10 常用属性

属 性 名	含 义	属 性 名	含 义
LineWidth	设置线的宽度	MarkerEdgeColor	设置标记点的边缘颜色
MarkerSize	设置标记点的大小	MarkerFaceColor	设置标记点的填充颜色

$h = \text{plot}(X_1, Y_1, \text{LineStyle}, \text{'PropertyName'}, \text{PropertyValue})$ : 返回绘制函数的句柄值  $h$ 。

$\text{loglog}$  函数、 $\text{semilogx}$  函数与  $\text{semilogy}$  函数的用法与  $\text{plot}$  函数的用法类似。

**【例 3-23】** 用三种不同的线型、标记符号和颜色分别绘制正余弦曲线。

```
>> clear all; %清除工作空间中变量
x = linspace(-2 * pi, 2 * pi);
```

```

y1 = sin(x);
y2 = cos(x);
figure
h = plot(x,y1,x,y2);

```

运行程序,效果如图 3-4 所示。

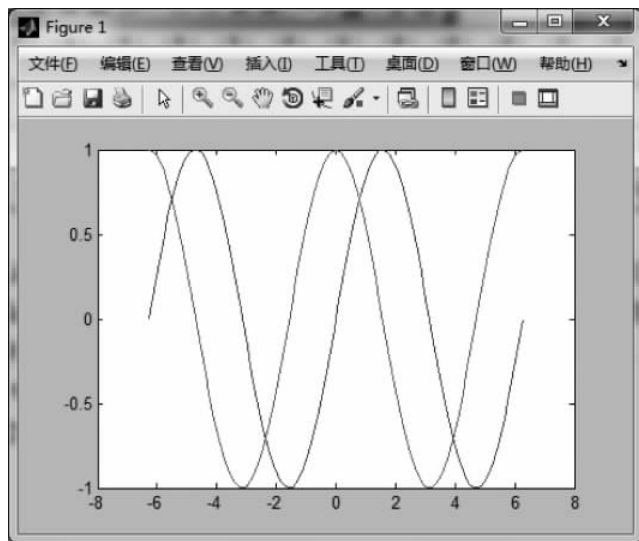


图 3-4 正余弦曲线

#### 注意:

(1) 用来绘制图形的数据必须已经存储在工作空间中,也就是说在执行 plot 函数前,当前工作空间中必须有可用来绘制图形的数据;

(2) 对应的 x 轴和 y 轴的数据长度必须相同;

(3) 如果省略选项 option,系统将按默认的格式绘制曲线;

(4) option 中的属性可以多个连用,如选项“-r”表示红色的虚线;

(5) 执行 Figure 命令时,绘图结果将出现在一个新的窗口中。如图 3-4 所示,Figure 1 显示的为例 3-23 绘制的结果,如同一般的窗口,它有自己的菜单栏和工具栏;

(6) 如果对已绘制的图形不满意,提出更具体的要求,如坐标轴范围、绘制网格等,可通过以下代码实现:

```

>> axis([0 10 -2 2]);
>> grid on;

```

运行程序,效果如图 3-5 所示。

#### 2. 修饰图形

如果对图形还不太满意,可对图形进行一些修改,在 MATLAB 中,提供了多种图形函数,用于图形的修饰。常用的图形修饰函数名称及其说明如表 3-11 所示。

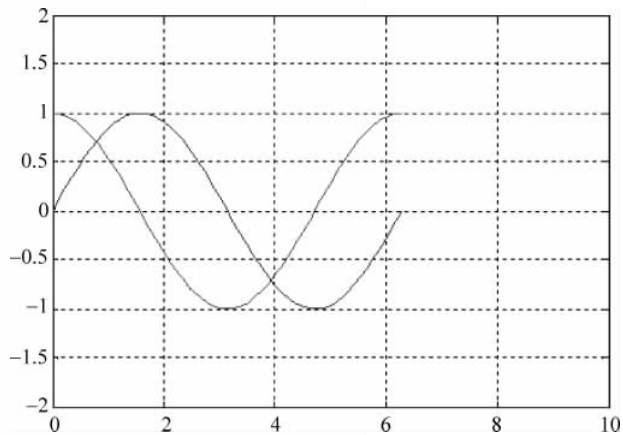


图 3-5 修改后的图形

表 3-11 常用图形修改函数及说明

函 数	说 明
axis([Xmin, Xmax, Ymin, Ymax])	x、y 坐标轴范围的调整
xlabel('string')	标注 x 轴名称
ylabel('string')	标注 y 轴名称
title('string')	标注图形标题
legend('string1', 'string2', ...)	标注图形标注
grid on	给图形增加网格
grid off	给图形取消网格
gtext('string')	在图形中加入普通文本标注

**【例 3-24】** 对例 3-23 的图形进行修饰,实现以下要求:

- (1) 将图形的 x 轴大小范围限定在  $[0, 2\pi]$ , y 轴的大小范围限定在  $[-1, 1]$ ;
- (2) x、y 轴分别标注为“弧度值”、“函数值”;
- (3) 图形标题标注为“正余弦曲线”;
- (4) 添加图例标注,标注字符分别为 y1, y2;
- (5) 属性两条曲线的线型、线条大小;
- (6) 在两条曲线上分别标注文本  $y1 = \sin(x)$ ,  $y2 = \cos(x)$ ;
- (7) 给图形添加网格。

其实现的 MATLAB 代码为:

```
>> clear all; %清除工作空间中变量
x = linspace(-2 * pi, 2 * pi);
y1 = sin(x);
y2 = cos(x);
figure
h = plot(x, y1, x, y2);
set(h(1), 'LineWidth', 2);
set(h(2), 'Marker', '* ');
axis([0, 2 * pi, -1, 1]);
xlabel('弧度值');
```

```

ylabel('函数值');
title('正余弦曲线');
legend('y1','y2');
grid
gtext('y1 = sin(x)');
gtext('y2 = cos(x)');

```

运行程序,效果如图 3-6 所示。

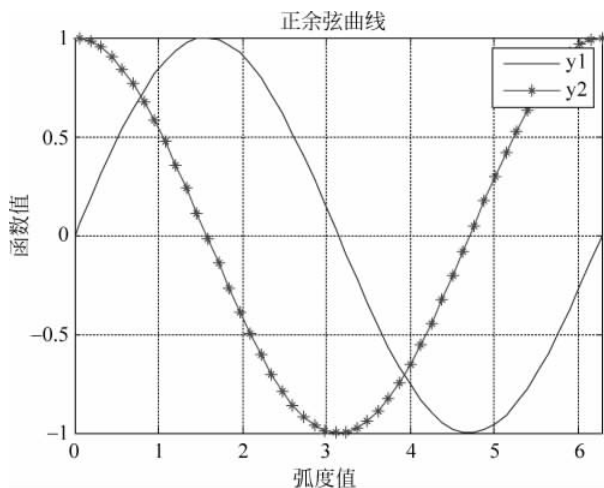


图 3-6 输出图形(未添加文本说明)

如图 3-6 所示,脚本在执行第一个 `gtext` 命令时,需要在图形窗口 Figure 1 中确定该文本的位置。

Figure 1 上可以看到一个跟随鼠标移动的十字形指针,将鼠标拖动到对应曲线附近,然后单击鼠标,字符串 `y1 = sin(x)` 即可添加到此处。

同理,在执行第二个 `gtext` 命令时,仍需要进行类似的操作,将字符串 `y2 = cos(x)` 添加到图形中,最终效果如图 3-7 所示。

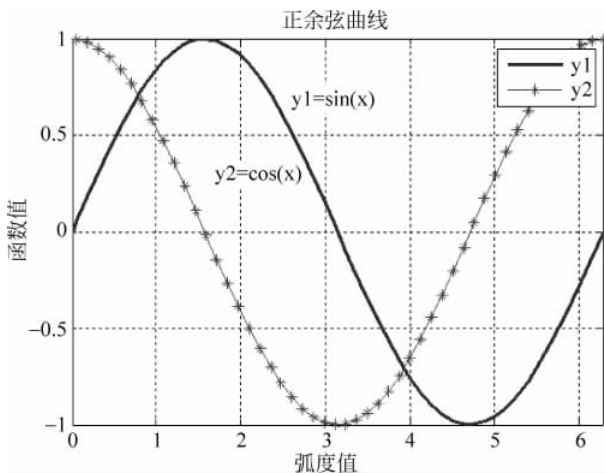


图 3-7 最终输出图形

### 3. 特殊二维曲线绘图

在 MATLAB 中,除了可以通过函数 plot 等绘制图形外,还有一些函数可以绘制特殊的图形,例如条形图、直方图等。

表 3-12 列出了 MATLAB 自带的常用的特殊二维图形函数。

表 3-12 特殊二维图形函数

函 数	说 明
bar/barh	bar 函数用于绘制垂直条形图,barh 用于绘制水平二维条形图
hist	用于绘制直方图
area	用于绘制面积图
pie	用于绘制二维饼图
scatter	用于绘制散点图
pareto	用于绘制排列图(累托图)
compass	用于绘制罗盘图
feather	用于绘制羽毛图
quiver	用于绘制二维矢量图
stem	用于绘制火柴杆图
stairs	用于绘制阶梯图
polar	用于绘制极坐标图
contour	用于绘制二维等高线图
contourf	用于绘制带填充的二维等高线图
clabel	为指定的等高线添加数据标签
errorbar	用于绘制曲线误差形图

**【例 3-25】** 利用 MATLAB 提供的特殊函数绘制特殊二维图形。

其实现的 MATLAB 代码为:

```
>> clear all;
y = [75.995,91.972,105.711,123.203,131.669,...
     150.697,179.323,203.212,226.505,249.633,281.422];
subplot(231); bar(y);
title('垂直等高线图');axis square;
subplot(232); barh(y);
title('水平等高线图');axis square;
rng(0,'twister');
theta = linspace(0,2 * pi,300);
x = sin(theta) + 0.75 * rand(1,300);
y1 = cos(theta) + 0.75 * rand(1,300);
s = 40;subplot(233);
scatter(x,y1,s,'MarkerEdgeColor','b','MarkerFaceColor','c','LineWidth',1.5);
title('散点图');axis square;
theta = (-90:10:90) * pi/180;
r = 2 * ones(size(theta));
[u,v] = pol2cart(theta,r);
subplot(234);feather(u,v);
title('羽毛图');axis square;
[X1,Y1] = meshgrid(-2:.2:2);
```

```

Z = X1.*exp(-X1.^2 - Y1.^2);
[DX,DY] = gradient(Z,.2,.2);
subplot(235);contour(X1,Y1,Z) %等高线图
hold on
quiver(X1,Y1,DX,DY) % 矢量图
colormap hsv;
title('带等高线的矢量图');axis square;
X2 = linspace(0,2*pi,25)';
Y2 = (cos(2*X2));subplot(236);
stem(X2,Y2,'LineStyle','- .','MarkerFaceColor','red','MarkerEdgeColor','green');
title('火柴杆图');axis square;

```

运行程序,效果如图 3-8 所示。

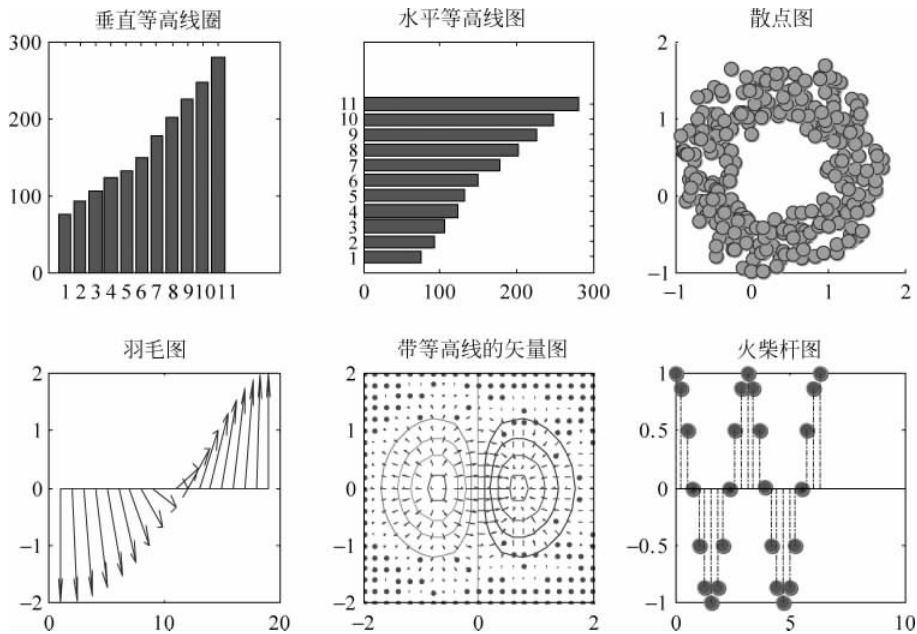


图 3-8 二维特殊图形

#### 4. 图形窗口控制

MATLAB 提供了一系列专门的图形窗口控制函数,通过这些函数,可以创建或者关闭图形窗口,可以同时打开几个窗口,也可以在一个窗口内绘制若干子图。这些函数及其功能说明如表 3-13 所示。

表 3-13 MATLAB 图形窗口控制函数及说明

函 数	说 明
figure	每调用一次就打开一个新的图形窗口
figure(n)	创建或打开第 n 个图形窗口,使之成为当前窗口
clf	清除当前图形窗口
hold on	保留当前窗口的图形不被后续图形覆盖,可实现在同一坐标系中多幅图形的重叠
hold off	解除 hold on 命令,一般与 hold on 成对使用

函 数	说 明
subplot(m,n,p)	将当前绘图窗口分割成 m 行、n 列,并在第 p 个区域绘图
close	关闭当前图形窗口
close all	关闭所有图形窗口

**注意:**

(1) 第一个绘图命令(如 plot)运行后,将自动创建一个名为 Figure 1 的图形窗口。这个窗口将被当作当前窗口,接着的所有绘图命令(包括绘图修饰和再一次的 plot 等命令)均在该图形窗口中执行,后续绘图指令会覆盖原图形或叠加在原图形上;

(2) 使用 subplot 命令时,各个绘图区域以“从左到右、先上后下”的原则来编号。MATLAB 允许每个绘图区域以不同的坐标系单独绘制图形。

**【例 3-26】** 取三个不同的 x 值,  $x_1=0:\pi/20:\pi$ ,  $x_2=\pi/2:\pi/20:3*\pi/2$ ,  $x_3=\pi:\pi/20:2*\pi$ , 在同一坐标系下绘制  $y_1=\sin(x)$ ,  $y_2=\sin(x-0.25)$ ,  $y_3=\sin(x-0.5)$  的图形,并利用 hold on 绘图。

其实现的 MATLAB 代码为:

```
>> clear all; %清除工作空间中变量
x1 = 0:pi/20:pi;
x2 = pi/2:pi/20:3 * pi/2;
x3 = pi:pi/20:2 * pi;
y1 = sin(x1);
y2 = sin(x2 - 0.25);
y3 = sin(x3 - 0.5);
figure
hold on;
plot(x1, y1, '-.r * ');
plot(x2, y2, '--mo');
plot(x3, y3, ':bs');
hold off;
%图形修饰
axis([0, 2.2 * pi, -1, 1]);
xlabel('弧度值');
ylabel('函数值');
title('三条不同相位的正弦曲线');
legend('y1', 'y2');
grid
gtext('y1 = sin(x)');
gtext('y2 = sin(x - 0.25)');
gtext('y3 = sin(x - 0.5)');
```

运行程序,效果如图 3-9 所示。

**注意:** 在程序中,绘制三条曲线的命令不同之处在于使用了配对的 hold on 和 hold off,然后分别使用了三次 plot 函数。这与直接使用 plot 绘制三条曲线效果一致,只需给出 plot(x1,y1,x2,y2,x3,y3,'option')即可。如果去掉 hold on 会得到如图 3-10 所示的结果,只显示最后一个 plot 绘制结果,也即 y3。

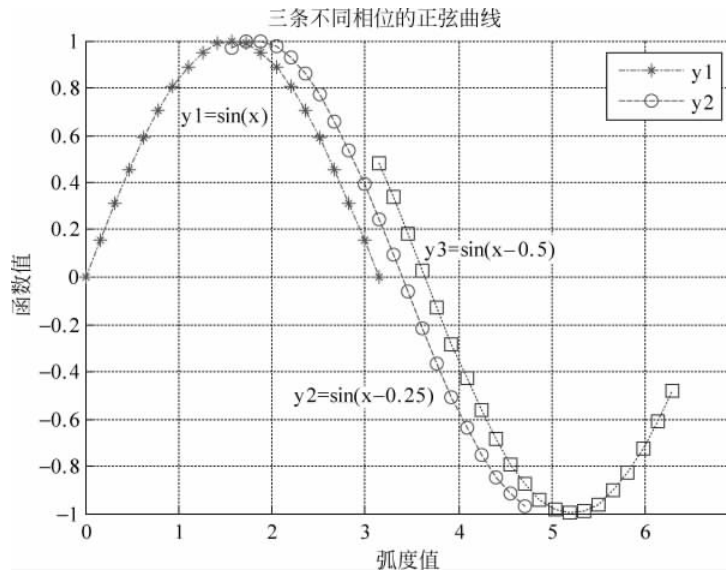


图 3-9 绘图结果

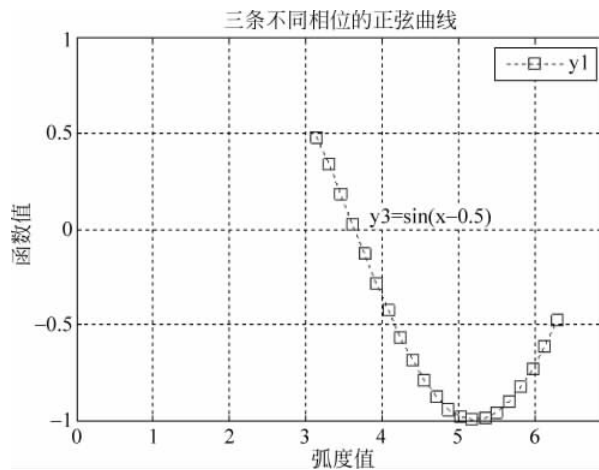


图 3-10 去掉 hold on 的绘图效果

### 3.6.2 三维图形绘制

除了常用的二维图形外, MATLAB 还提供了三维数据的绘制函数, 可以在三维空间绘制曲线或曲面。

#### 1. 三维曲线的绘制

在 MATLAB 中, 提供了 `plot3` 函数用于绘制三维曲线图, 其函数用法与二维曲线绘制函数 `plot` 类似。函数 `plot3` 的调用格式为:

(1) `plot3(X1, Y1, Z1, ...)`: 以默认线型属性绘制三维点集  $(X_i, Y_i, Z_i)$  确定的曲线。

$X_i, Y_i, Z_i$  为相同大小的向量或矩阵。

(2) `plot3(X1,Y1,Z1,LineStyle,...`): 以参数 `LineStyle` 确定的线型属性绘制三维点集( $X_i, Y_i, Z_i$ )确定的曲线。 $X_i, Y_i, Z_i$  为相同大小的向量或矩阵。

(3) `plot3(...,'PropertyName',PropertyValue,...`): 绘制三维曲线,根据指定的属性值设定曲线的属性。

(4) `h = plot3(...)`: 返回绘制的曲线图的句柄值向量 `h`。

**【例 3-27】** 利用 `plot3` 函数绘制以下参数方程的三维曲线。

$$\begin{cases} x = t \\ y = \cos t \\ z = \sin 2t \end{cases}$$

其实现的 MATLAB 代码如下:

```
>> clear all;
x = 0:0.01:50;
y = cos(x);
z = sin(2 * x);
plot3(x,y,z,'r-.');
grid on;
title('三维曲线');
```

运行程序,效果如图 3-11 所示。

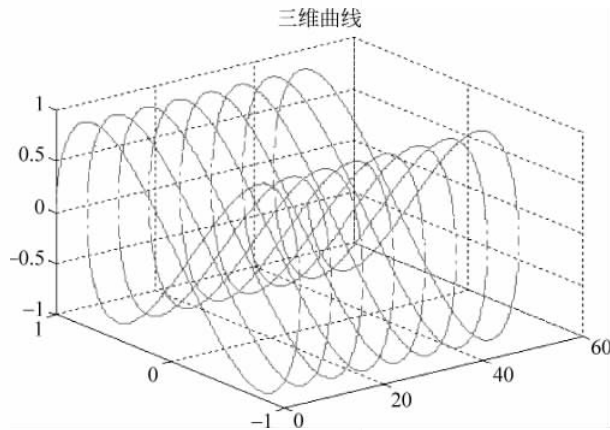


图 3-11 三维曲线的绘制

三维曲线修饰与二维图形的图形修饰函数类似,但比二维图形的修饰函数多了一个  $z$  轴方向,如 `axis([Xmin,Xmax,Ymin,Ymax,Zmin,Zmax])`、`zlabel('String')`。

例如,为图 3-11 添加标注,代码为:

```
>> clear all;                                % 清除工作空间中变量
x = 0:0.01:50;
y = cos(x);
z = sin(2 * x);
plot3(x,y,z,'r-.');
grid on;
title('三维曲线');
```

```

xlabel('x 轴');
ylabel('y 轴');
zlabel('z 轴');
axis([0,60,-1.5,1.5,-1,1]);

```

运行程序,效果如图 3-12 所示。

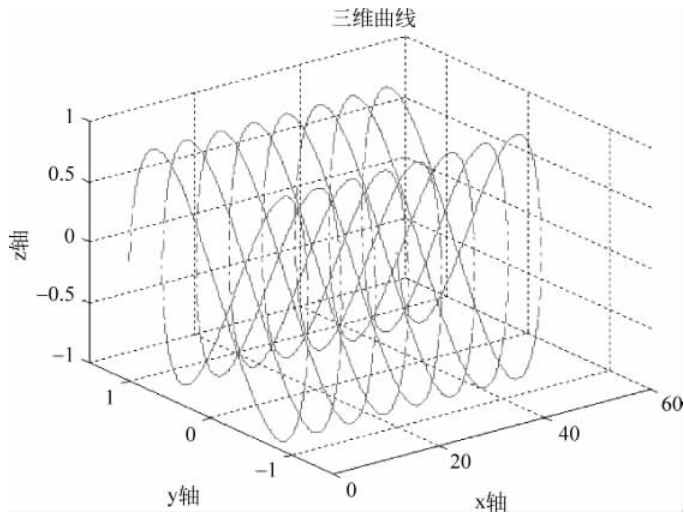


图 3-12 添加标注

## 2. 三维曲面的绘制

三维曲面方程存在两个自变量  $x, y$  和一个因变量  $z$ 。因此,绘制三维曲面图形必须先  $xy$  平面上建立网络坐标,每一个网络坐标点,和它对应的  $z$  坐标所确定的一组三维数据就定义了曲面上的一个点。三维曲面绘制中,常用的 3 个函数及功能如表 3-14 所示。

表 3-14 三维曲面绘制函数

函 数	说 明
$[X, Y] = \text{meshgrid}(x, y)$	根据 $(x, y)$ 二维坐标数据生成 $xy$ 网格点坐标数据,其中, $x, y$ 为向量, $X, Y$ 为矩阵
$\text{mesh}(X, Y, Z)$	绘制三维网络曲面,通过直接连接相邻的点构成三维曲面
$\text{surf}(X, Y, Z)$	绘制三维阴影曲面,通过平面连接相邻的点构成三维曲面

### 【例 3-28】 绘制三维网格图实例。

```

>> clear all; % 清除工作空间中的所有变量
[X, Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
subplot(231); mesh(Z);
title('绘制数据 Z 的网格图');
subplot(232); mesh(X, Y, Z);
axis([-8 8 -8 8 -0.5 1]);

```

```

title('绘制三维网格图')
C = gradient(Z);
subplot(233);mesh(X,Y,Z,C);
title('颜色由C指定')
C = del2(Z);
subplot(234);mesh(Z,C,'FaceLighting','gouraud','LineWidth',0.3);
title('设置网格图属性');
subplot(235);meshz(Z);
title('meshz 绘制网格图');
subplot(236);meshc(Z);
title('meshc 绘制网格图');

```

运行程序,效果如图 3-13 所示。

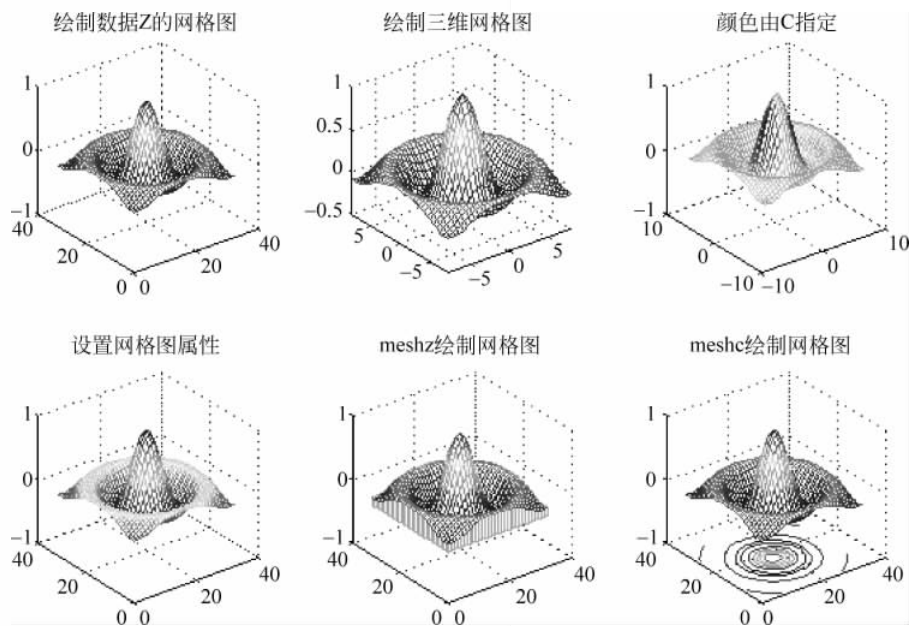


图 3-13 三维网格图

**【例 3-29】** 利用 surf 绘制三维表面图。

```

>> clear all; % 清除工作空间中的所有变量
[x,y] = meshgrid(-3:1/8:3);
z = peaks(x,y);
subplot(221);surf(z);
title('surf(z)绘图形式');
subplot(222);surf(x,y,z);
title('surf(x,y,z)绘图形式');
subplot(223);surfl(x,y,z);
title('surfl(x,y,z)绘图形式');
subplot(224);surfc(x,y,z);
title('surfc(x,y,z)绘图形式');

```

运行程序,效果如图 3-14 所示。

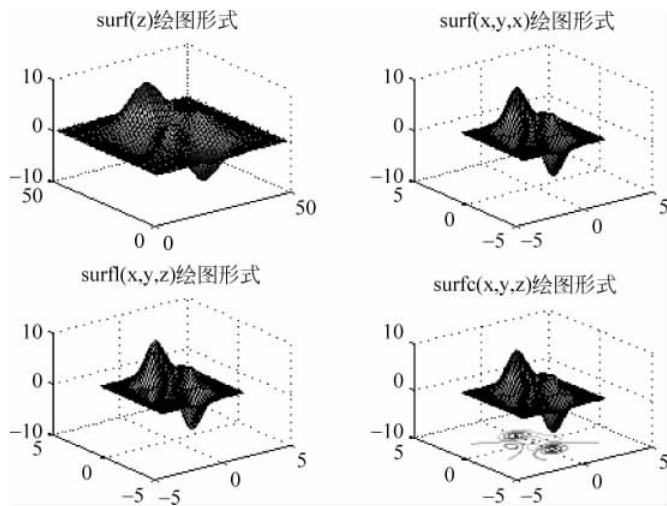


图 3-14 三维表面图

### 3. 三维特殊绘图

在科学研究中,有时也需要绘制一些特殊的三维图形,如统计学中三维直方图、圆柱体图、饼形图等。MATLAB 中提供了用于绘制这些特殊三维图形的函数,表 3-15 列出了 MATLAB 常用的三维特殊图形绘图函数。

表 3-15 三维特殊图形函数

函 数	说 明
bar3、barh3	绘制三维垂直(水平)柱状图
cylinder	绘制三维柱面图
sphere	用于绘制球面图
contour3	用于绘制三维等高线图
pie3	用于绘制三维饼图
scatter3	用于绘制三维散点图
stem3	用于绘制三维火柴杆图
quiver3	用于绘制三维向量图
comet3	用于绘制三维彗星图
fill3	用于绘制三维填充图
ribbon	用于绘制三维彩带图
patch	绘制三维片块图

**【例 3-30】** 绘制特殊三维图形。

其实现的 MATLAB 程序代码为:

```
>> clear all;
t = 0:pi/10:2 * pi;
[X1,Y1,Z1] = cylinder(2+cos(t));
subplot(231);surf(X1,Y1,Z1)
axis square;title('三维柱面图');
```

```

subplot(232);sphere
axis equal;title('三维球体');
x1 = [1 3 0.5 2.5 2];
explode = [0 1 0 0 0];
subplot(233);pie3(x1,explode)
title('三维饼图');axis equal;
X2 = [0 1 1 2;1 1 2 2;0 0 1 1];
Y2 = [1 1 1 1;1 0 1 0;0 0 0 0];
Z2 = [1 1 1 1;1 0 1 0;0 0 0 0];
C = [0.5000 1.0000 1.0000 0.5000;
     1.0000 0.5000 0.5000 0.1667;
     0.3330 0.3330 0.5000 0.5000];
subplot(234);fill3(X2,Y2,Z2,C);
colormap hsv
title('三维填充图');axis equal;
[x2,y2] = meshgrid(-3:.5:3,-3:.1:3);
z2 = peaks(x2,y2);
subplot(235);ribbon(y2,z2)
colormap hsv
title('三维彩带图');axis equal;
[X3,Y3] = meshgrid(-2:0.25:2,-1:0.2:1);
Z3 = X3.*exp(-X3.^2 - Y3.^2);
[U,V,W] = surfnorm(X3,Y3,Z3);
subplot(236);quiver3(X3,Y3,Z3,U,V,W,0.5);
hold on
surf(X3,Y3,Z3);
colormap hsv
view(-35,45);
title('三维向量场图');axis equal;
set(gcf,'color','w');

```

运行程序,效果如图 3-15 所示。

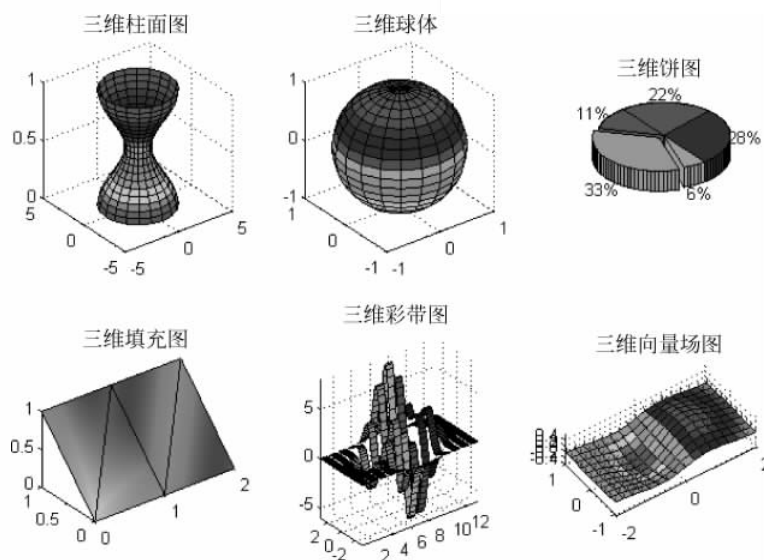


图 3-15 特殊三维效果图

## 3.7 图形对象属性

### 3.7.1 图形对象及属性

前面提到,任何一个图形元素,读者都可以对其进行单独修饰,而不影响图形的其他部分,这种独立的图形元素称为图形对象。图形对象的修改通过调整其属性来完成。

表 3-16 列出了 MATLAB 中常用的图形对象及主要属性。

表 3-16 MATLAB 中常用的图形对象及属性

图形对象	说明	属性
root(根对象)	一切对象的根对象	无须设置属性
figure(图形窗口对象)	root 对象的下级,子对象	figurename(图形窗口的名称) figurecolor(图形窗口的颜色)
axis(坐标轴对象)	figure 对象的下级,子对象	title(图形标注),label(坐标轴标注),limit(坐标轴范围),color(坐标轴颜色),grid(是否加网格线)
line(线对象)	axis 对象的下级,子对象	linetype(曲线线型,如 line、bar、stem、stairs),color(曲线颜色),linewidth(曲线线宽),data(数据),Marker(曲线上的标记类型)
text(字符对象)	axis 对象的下级,子对象	string(字符串内容),fontname(字体名称),fontsize(字体大小),color(字符颜色)

当调用 plot 命令绘制二维曲线时,MATLAB 的执行过程大致为:

- (1) 使用 figure 命令,在 root 根对象上生成一个 figure 图形窗口对象;
- (2) 使用 axis 命令,在图形窗口内生成一个绘图区域(axis 对象);
- (3) 最后用 line 命令在 axis 指定的区域内绘制线条(line 对象)。

因此,MATLAB 所绘制的图形是由基本的图形对象组合而成,可以通过改变图形对象的属性来设置所绘制的图形,以满足不同的绘图需求。

### 3.7.2 图形属性的设置

在 MATLAB 中,提供了 set 函数用于设置图形的属性;提供了 get 函数用于获取图形的属性。

**【例 3-31】** 绘制 peaks 函数的三角图形,并通过 set 函数查看各种属性。

```
>> clear all; % 清除工作空间所有变量
H1 = surf(peaks(45)); % peaks 为 MATLAB 内置的绘制山峰图
```

以上命令行可得到 peak 的曲面图,同时将图形句柄保存在变量 H1 中。接着通过 set 函数查看设置的图形属性。

```
set(H1)
ans =
AlphaData: {}
```

```
AlphaDataMapping: {3x1 cell}
    CData: {}
CDataMapping: {2x1 cell}
DisplayName: {}
EdgeAlpha: {2x1 cell}
EdgeColor: {3x1 cell}
FaceAlpha: {3x1 cell}
FaceColor: {4x1 cell}
LineStyle: {5x1 cell}
LineWidth: {}
    Marker: {14x1 cell}
MarkerEdgeColor: {3x1 cell}
MarkerFaceColor: {3x1 cell}
MarkerSize: {}
MeshStyle: {3x1 cell}
    XData: {}
    YData: {}
    ZData: {}
FaceLighting: {4x1 cell}
EdgeLighting: {4x1 cell}
BackFaceLighting: {3x1 cell}
AmbientStrength: {}
DiffuseStrength: {}
SpecularStrength: {}
SpecularExponent: {}
SpecularColorReflectance: {}
VertexNormals: {}
    NormalMode: {2x1 cell}
ButtonDownFcn: {}
    Children: {}
Clipping: {2x1 cell}
CreateFcn: {}
DeleteFcn: {}
BusyAction: {2x1 cell}
HandleVisibility: {3x1 cell}
    HitTest: {2x1 cell}
Interruptible: {2x1 cell}
    Selected: {2x1 cell}
SelectionHighlight: {2x1 cell}
    Tag: {}
UIContextMenu: {}
    UserData: {}
Visible: {2x1 cell}
Parent: {}
XDataMode: {2x1 cell}
XDataSource: {}
YDataMode: {2x1 cell}
YDataSource: {}
CDataMode: {2x1 cell}
CDataSource: {}
ZDataSource: {}
```

以上结果中,如果某项属性值为空,例如“LineWidth: {}”表示用户不能对该图形对象设置 LineWidth 属性;如果对应的属性值不为空,用户则可以在对应的属性值列表中

设置相应的属性。

```
>> set(H1, 'Marker')
[ + | o | * | . | x | square | diamond | v | ^ | > | < | pentagram | hexagram | {none} ]
```

从以上结果可看出,可以从 Marker 属性的 14 个选项中选择任何一个来设置图形标记的属性。

**【例 3-32】** 利用 get 函数查看创建对象的属性值。

```
>> clear all;           % 清除工作空间中的变量
patch;                 % 绘制色块图形
surface;               % 绘制表面图
text;                  % 标注文字
line;                  % 绘制线条
```

运行程序,效果如图 3-16 所示。

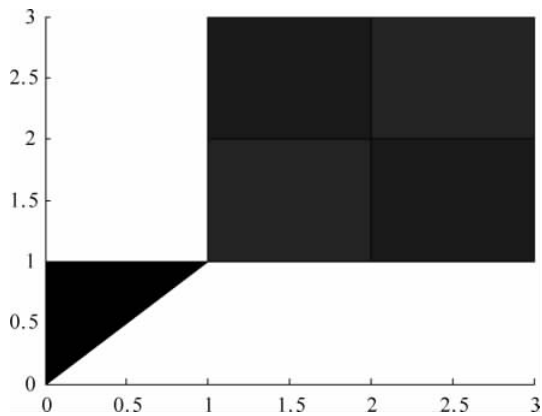


图 3-16 创建的图形对象

利用 get 函数查看所创建对象的属性值。

```
>> get(gca)
ActivePositionProperty = outerposition
ALim = [0.1 10]
ALimMode = auto
AmbientLightColor = [1 1 1]
Box = off
CameraPosition = [1.5 1.5 9.16025]
CameraPositionMode = auto
CameraTarget = [1.5 1.5 0.5]
CameraTargetMode = auto
CameraUpVector = [0 1 0]
CameraUpVectorMode = auto
CameraViewAngle = [6.60861]
CameraViewAngleMode = auto
...
BeingDeleted = off
ButtonDownFcn =
Children = [ (4 by 1) double array]
Clipping = on
```

```

CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [1]
Selected = off
SelectionHighlight = on
Tag =
Type = axes
UIContextMenu = []
UserData = []
Visible = on

```

### 3.7.3 图形可视编辑工具

MATLAB执行绘图函数后,将出现一个图形窗口。该窗口除了简单的显示图形功能外,其本身还是一个功能强大的图形可视编辑工具,可实现的功能主要有:

- (1) 通用的图形文件管理功能,如保存、打开、新建图形文件等;
- (2) 通用的图形效果编辑功能,如图形放大、缩小、旋转、对齐等;
- (3) 图形对象插入功能,如插入坐标轴名称、图形标题、图例标注、线段、文字等;
- (4) 独立展示窗口中各图形对象属性功能,如线段的类型、颜色、粗细等。

图形对象插入功能可通过选择菜单命令 Insert 后,再选择相应的对象选项来完成,如图 3-17 所示。

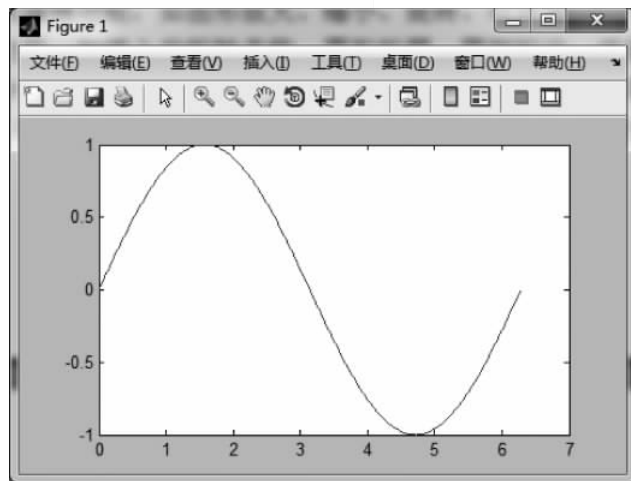


图 3-17 绘制的正弦曲线

在图 3-16 中,选择菜单命令“插入|标题”,效果如图 3-18 所示。该菜单项与 title('string')功能一致。菜单命令“插入|图例”与 legend('string1','string2',...)功能一致。

显然,使用菜单命令比在 MATLAB 命令窗口输入函数命令要简便得多,且更具可视性。

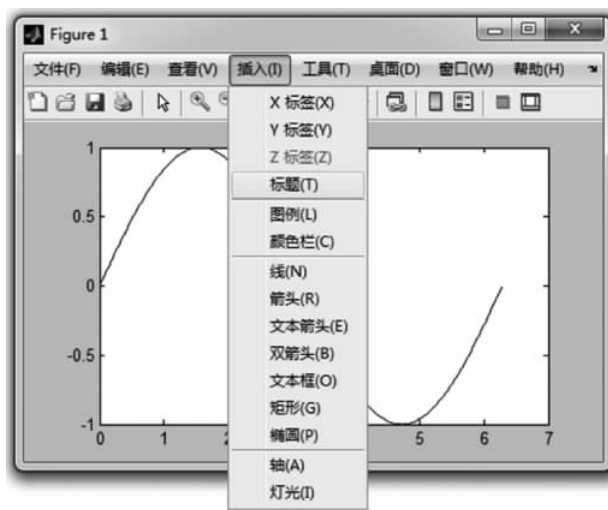


图 3-18 图形窗口及菜单功能

图形对象属性的设置可以通过以下两种方法实现：

(1) 选择菜单命令“查看|属性编辑器”，如图 3-19 所示，在图形正文出现了一个“属性编辑器”窗口。

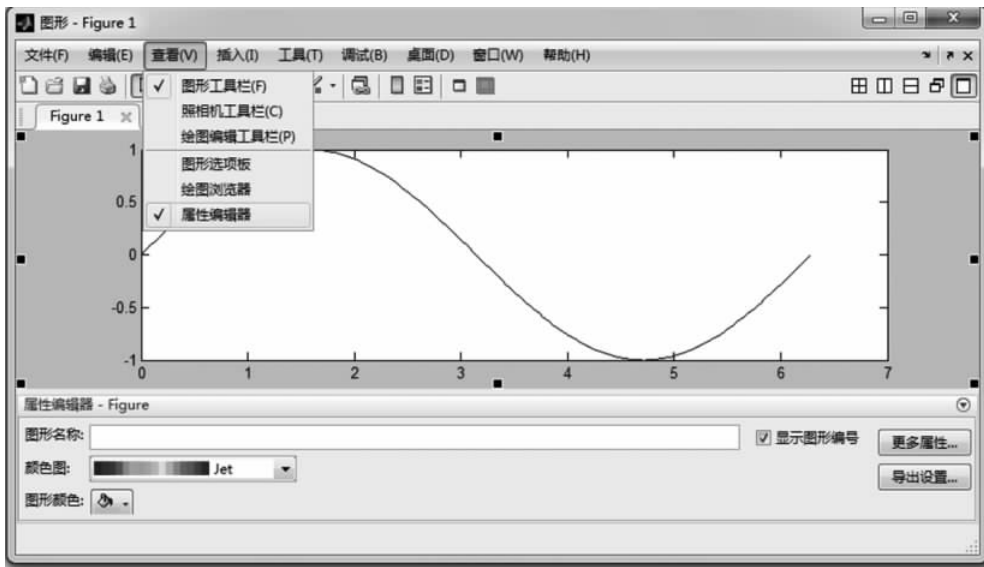


图 3-19 第一种方法打开属性编辑窗口

(2) 选择菜单命令“工具|编辑图形”，如图 3-20 所示，鼠标移到需要修改的对象上双击，或者选中后再右击，在弹出的菜单中选择“显示属性编辑器”项，将显示属性编辑窗口。

这两种方法都可以使“编辑图形”按钮高亮化，同时图形窗口下方会出现属性编辑窗口。在此属性编辑窗口下，可以进行坐标、线段、标题等项目的颜色、字体、网格、范围的设置。

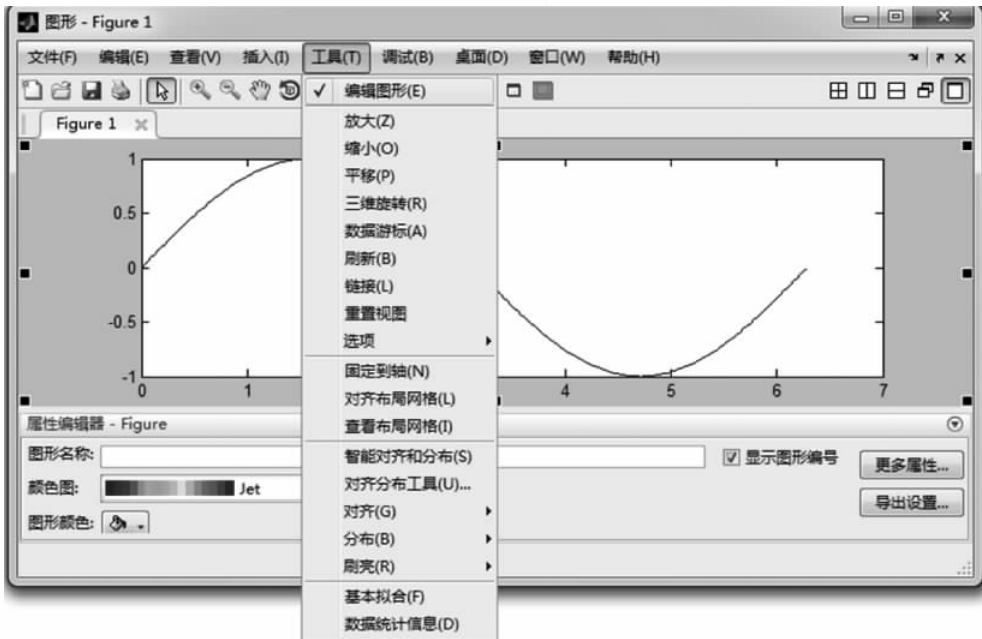


图 3-20 第二种方法打开属性编辑窗口

选中一个对象后,再单击右下角的“更多属性”按钮,将出现该对象更加详细的属性编辑窗口,如图 3-21 所示。

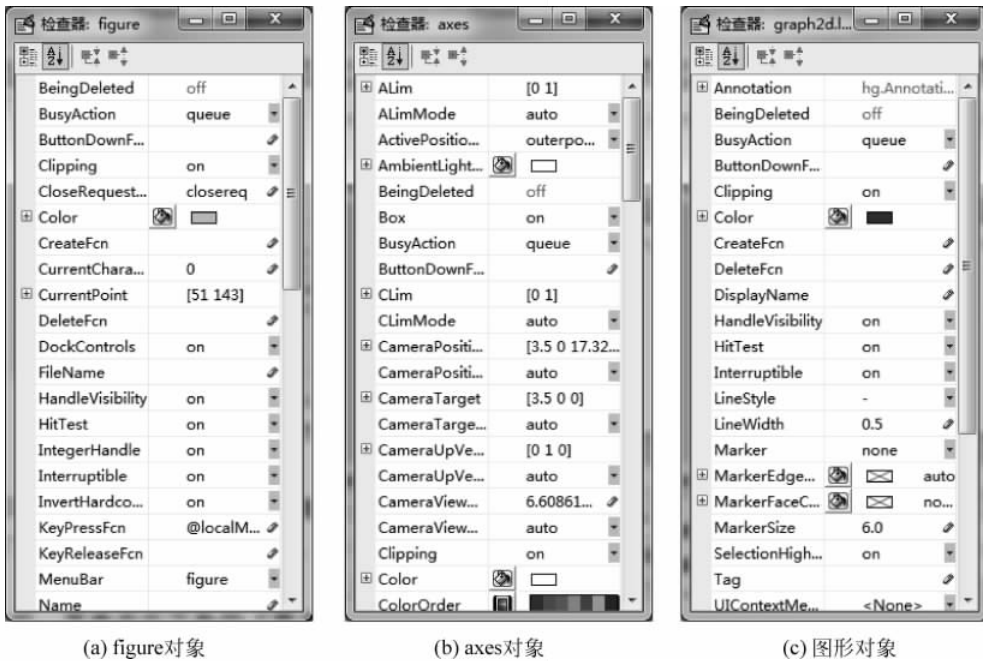


图 3-21 更多属性编辑窗口