第3章 MATLAB简单操作

数值计算是MATLAB中最重要、最有特色的功能之一，也是MATLAB软件的基础。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。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基本元素

本节介绍常量、变量和矩阵这三种最常用的MATLAB基本元素以及赋值语句的基本形式。

3.1.1 常量

常量，在MATLAB中习惯称之为特殊变量，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π
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，∞/∞，0*∞
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的命令行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。

3.1.2 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、数字或下画线。
- 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。

与其他的程序设计语言相同，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。如果要将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 global。一般来说，全局变量均用大写的英文字符表示。

3.1.3 赋值语句

MATLAB 采用命令行形式的表达式语言，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。读者在命令行窗口中输入语句并按 Enter 键确认后，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如下：

```
赋值变量 = 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 命令行窗口中。

【例 3-1】 对 A 赋值，实现 $A=4\times28$。

在 MATLAB 命令行窗口中输入语句并按 Enter 键。

```
>> A = 4 * 28
A =
  112
```

注意：

(1) 如果赋值语句后面没有分号“;”，MATLAB 命令行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如下：

```
[返回变量列表] = 函数名(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；另一类是用户根据需要自定

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 tan 函数求 $a=\tan\left(\frac{\pi}{2}\right)$ 的值。

在 MATLAB 命令行窗口中输入语句并按 Enter 键：

```
>> a = tan(pi/2)
a =
  1.6331e + 16
```

注意：

(1) 函数名的命名规则与变量名命名规则一致，用户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，用户可直接调用；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。

3.1.4 矩阵及数组

最基本的 MATLAB 数据结构体是矩阵。矩阵是按行和列排列的数据元素的二维矩形数组。元素可以是数字、逻辑值(true 或 false)、日期和时间、字符串或者其他 MATLAB 数据类型。

即使一个数字也能以矩阵的形式存储。例如，包含值 100 的变量存储为 double 类型的 1×1 矩阵。

```
>> A = 99
A =
    99
>> whos
  Name        Size          Bytes    Class      Attributes
  A           1x1           8        double
```

1. 构建数据矩阵

如果有一组具体的数据，可以使用方括号将这些元素排列成矩阵。一行数据的元素之间用空格或逗号分隔，行与行之间用分号分隔。例如，创建只有一行的矩阵，其中包含四个数字元素。得到的矩阵大小为 1×4，因为它有一行和四列。这种形状的矩阵通常称为行向量。

```
>> A = [12 52 91 -3]
A =
    12    52    91    -3
>> sz = size(A)   % 显示矩阵的大小
sz =
    1     4
```

现在再用这些数字创建一个矩阵，但排成两行，此矩阵有两行和两列。

```
>> A = [12 52;91 -3]
```

```
A =
    12    52
    91   -3
>> sz = size(A)
sz =
    2     2
```

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 Enter 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。

2. 专用矩阵函数

MATLAB 中有许多函数可以帮助我们创建具有特定值或特定结构的矩阵。例如，zeros 和 ones 函数可以创建元素全部为零或全部为一的矩阵。这些函数的第一个和第二个参数分别是矩阵的行数和列数。

```
>> A = zeros(3,2)   % 创建 3 行 2 列的零矩阵
A =
    0     0
    0     0
    0     0
>> B = ones(2,4)   % 创建 2 行 4 列的全 1 矩阵
B =
    1     1     1     1
    1     1     1     1
```

diag 函数将输入元素放在矩阵的对角线上。例如，创建一个行向量 $\boldsymbol{A}$，其中包含四个元素。然后创建一个 4×4 矩阵，其对角元素是 $\boldsymbol{A}$ 的元素。

```
>> A = [12 52;91 -3];
B = diag(A)     % 对角矩阵
B =
    12
    -3
```

3. 串联矩阵

还可以使用方括号将现有矩阵连接在一起。这种创建矩阵的方法称为串联。例如，将两个行向量串联起来，形成一个更长的行向量。

```
>> A = ones(1,4);
B = zeros(1,4);
C = [A B]
C =
    1     1     1     1     0     0     0     0
```

要将 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 排列为一个矩阵的两行，请使用分号。

```
>> D = [A;B]
D =
    1     1     1     1
    0     0     0     0
```

要串联两个矩阵，它们的大小必须兼容。也就是说，水平串联矩阵时，它们的行数必须相同。垂直串联矩阵时，它们的列数必须相同。例如，水平串联两个各自包含两行的矩阵。

```
>> A = ones(2,3)
A =
     1     1     1
     1     1     1
>> B = zeros(2,2)
B =
     0     0
     0     0
>> C = [A B]
C =
     1     1     1     0     0
     1     1     1     0     0
```

串联矩阵的另一种方法是使用串联函数，如 horzcat，它可以水平串联两个兼容的输入矩阵。

```
>> D = horzcat(A,B)
D =
     1     1     1     0     0
     1     1     1     0     0
```

4. 生成数值序列

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。冒号表达式的格式如下：

```
X = N1:step:N2
```

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

【例 3-3】 利用冒号法创建向量。

```
>> A = 1:8
A =
     1     2     3     4     5     6     7     8
>> % 可以使用冒号运算符创建在任何范围内以 1 为增量的数字序列
A = -2.5:2.5
A =
   -2.5000   -1.5000   -0.5000    0.5000    1.5000    2.5000
>> % 要更改序列的增量值,请在范围起始值和结束值之间指定增量值,以冒号分隔
>> A = 0:2:10
A =
     0     2     4     6     8    10
>> % 要递减,请使用负数
>> A = 6:-1:0
A =
```

```
    6     5     4     3     2     1     0
>> %还可以按非整数值递增.如果增量值不能均分指定的范围,MATLAB会在超出范围之前在可以达
 %到的最后一个值处自动结束序列
>> A = 1:0.2:2.1
A =
    1.0000    1.2000    1.4000    1.6000    1.8000    2.0000
```

5. 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$\boldsymbol{A}(i,j)$表示矩阵$\boldsymbol{A}$第i行第j列的元素。

【例 3-4】 获取矩阵$\boldsymbol{A}$=[1 4 6;3 9 7]第2行全部元素。

```
>> A=[1 4 6;3 9 7]
A =
    1     4     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。$\boldsymbol{A}(2,:)$表示矩阵$\boldsymbol{A}$第2行全部元素,$\boldsymbol{A}(:,2)$表示矩阵$\boldsymbol{A}$第2列全部元素,$\boldsymbol{A}(1,1:2)$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     4
```

3.2 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。

3.2.1 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3-2为MATLAB矩阵的算术运算符及其说明。

表 3-2 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\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 均为矩阵时，无定义

注意：当运算失败时 MATLAB 会提示出错。

【例 3-5】 矩阵的代数运算。

```
>> A = [8 1 6; 3 5 7; 4 9 2]
A =
     8     1     6
     3     5     7
     4     9     2
>> 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 =
     9     3     9
     3    16     9
    10    16    10
>> A-B              %矩阵的减法
ans =
     7    -1     3
     3    -6     5
    -2     2    -6
>> A*B              %矩阵乘法
ans =
    44    69    74
    45   110    75
    16   121    46
>> A\B              %矩阵左除
ans =
    0.5306   -0.8472    0.6639
    0.5722    0.8611    0.7056
   -0.6361    1.3194   -0.5028
>> 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 =
   -2.0800    -0.6000    1.6800
    1.8000    -0.0000    0.2000
   -2.7200     0.6000    1.1200
>> A/C
```

错误使用“/”,矩阵维度必须一致。

```
>> A^B
```

错误使用“^”,用于对矩阵求幂的维度不正确。请检查并确保矩阵为方阵并且幂为标量。要执行按元素矩阵求幂,请使用“.^”。

```
>> A^3
ans =
        1197        1029        1149
        1077        1125        1173
        1101        1221        1053
```

注意:

(1) 如果 $\boldsymbol{A}$、$\boldsymbol{B}$ 两矩阵进行加、减运算,则 $\boldsymbol{A}$、$\boldsymbol{B}$ 必须维数相同,否则系统提示出错。

(2) 如果 $\boldsymbol{A}$、$\boldsymbol{B}$ 两矩阵进行乘运算,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 $\boldsymbol{A}$、$\boldsymbol{B}$ 两矩阵进行右除运算,则两矩阵的列数必须相等(实际上,$\boldsymbol{X}=\boldsymbol{B}/\boldsymbol{A}=\boldsymbol{A}\times\boldsymbol{B}-1$)。

(4) 如果 $\boldsymbol{A}$、$\boldsymbol{B}$ 两矩阵进行左除运算,则两矩阵的行数必须相等(实际上,$\boldsymbol{X}=\boldsymbol{A}\backslash\boldsymbol{B}=\boldsymbol{A}-1\times\boldsymbol{B}$)。

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 3-3 列出了部分常用的矩阵运算函数。

表 3-3 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 3-6】 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
```

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号相区别,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 3-4 为矩阵元素群运算符及说明。

表 3-4 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.\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 3-7】 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3-5列出了MATLAB常用初等函数名及其对应的说明。

表3-5 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rad)
cos	余弦函数(角度单位为rad)
tan	正切函数(角度单位为rad)
abs	求实数绝对值或复数的模
sqrt	平方根函数
angle	求复数的辐角
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3-8】 元素群函数实例。

```
>> x = [0,pi/5,pi/6,pi/3];
>> y = tan(x)
y =
         0    0.7265    0.5774    1.7321
>> y1 = cos(x)
y1 =
    1.0000    0.8090    0.8660    0.5000
>> log10(x)
ans =
     -Inf    -0.2018    -0.2810    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
```

3.2.2 矩阵的关系运算

关系运算符使用“小于”“大于”和“不等于”等运算符对操作数进行定量比较。关系比较的结果是一个逻辑数组，指示关系为 true 的位置。常用的关系运算符如表 3-6 所示。

表 3-6 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)
<=	小于或等于	le(A, B)
>=	大于或等于	ge(A,B)

注意：表 3-6 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 做比较。

【例 3-9】 如果比较两个大小相同的矩阵，则结果将是大小相同且其元素指示关系为 true 的位置的逻辑矩阵。

```
>> A = [2 4 6; 8 10 12]
A =
    2     4     6
    8    10    12
>> B = [5 5 5; 9 9 9]
B =
    5     5     5
    9     9     9
```

```
>> A < B
ans =
  2×3 logical 数组
  1    1    0
  1    0    0
>> % 同样,也可以将某一个数组与标量进行比较
>> A > 7
ans =
  2×3 logical 数组
  0    0    0
  1    1    1
>> % 如果将一个 1×N 行向量与一个 M×1 列向量进行比较,则 MATLAB 会在执行比较之前将每个
% 向量都扩展为一个 M×N 矩阵,生成的矩阵包含这些向量中元素的每个组合的比较结果
>> A = 1:3
A =
    1     2     3
>> B = [2; 3]
B =
    2
    3
>> A >= B
ans =
  2×3 logical 数组
  0    1    1
  0    0    1
```

3.2.3 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 3-7 所示。

表 3-7 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:&、|、&&、||、~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做逻辑运算。

&、&& 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 计算很复杂时,采用 && 会提高运算效率。|、||也有相同的区别。

【例 3-10】 矩阵的逻辑运算。

```
>> %将 A 中大于 10 的所有值替换为数值 10
>> A(A>10) = 10
A =
     1     2     3
>> %然后,将 A 中不等于 10 的所有值替换为 NaN 值
>> A(A~=10) = NaN
A =
   NaN   NaN   NaN
>> %最后,将 A 中的所有 NaN 值替换为 0,并应用逻辑 NOT 运算符 ~A
>> A(isnan(A)) = 0;
C = ~A
C =
  1×3 logical 数组
   1   1   1
```

生成的矩阵用逻辑值 1(true)取代 NaN 值,用逻辑值 0(false)取代 10。逻辑 not 运算~A 将数值数组转换为逻辑数组,因此 A&C 返回逻辑值 0(false)的矩阵,A|C 返回逻辑值 1(true)的矩阵。

3.3 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、选择结构和循环结构。这一点 MATLAB 与其他编程语言完全一致。

3.3.1 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 3-11】 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
```

3.3.2 循环结构

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 循环语句

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。由 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(end-start)/increment。

【例 3-12】 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=7 时，得到一个矩阵：

```
>> a
a =
     2     0     1     0     0     0     0
     0     2     0     1     0     0     0
     1     0     2     0     1     0     0
     0     1     0     2     0     1     0
     0     0     1     0     2     0     1
     0     0     0     1     0     2     0
     0     0     0     0     1     0     2
```

2. while 循环

与 for 循环以固定次数求一组语句的值相反，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如下：

```
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 all(A)。特别地，空数组则被当作逻辑假，循环不执行。

【例 3-13】 利用 while 循环结构求方程 $x^3-2x-5=0$ 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 3-14】 编写求 0～50 之间 3 与 5 的公倍数的程序。

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

运行程序，输出如下：

```
输出 0～50 之间能同时被 3 和 5 整除的数
     0
    15
    30
    45
```

【例 3-15】 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('鸡的数量为 % d.\n',n1);
fprintf('兔子的数量为 % d.\n',n2);
```

运行程序，输出如下：

```
鸡的数量为 22.
兔子的数量为 14.
```

3.3.3 选择结构

在 MATLAB 中选择结构有两种形式，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 i f 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情况

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 1 为真时的执行语句 1
elseif 表达式 2
    表达式 2 为真时的执行语句 2
elseif 表达式 3
    表达式 3 为真时的执行语句 3
elseif …
    …
    …
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 3-16】 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:89
B
```

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如下：

```
switch 条件表达式
    case 常量 1
    语句组 1
    case 常量 2
    语句组 2
    …
    otherwise
    语句组 n + 1
end
```

执行过程：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 3-17】 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:75
C
```

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。

3.4 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。

3.4.1 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 3-18】 编写脚本文件，实现图像的绘制，效果如图 3-1 所示。

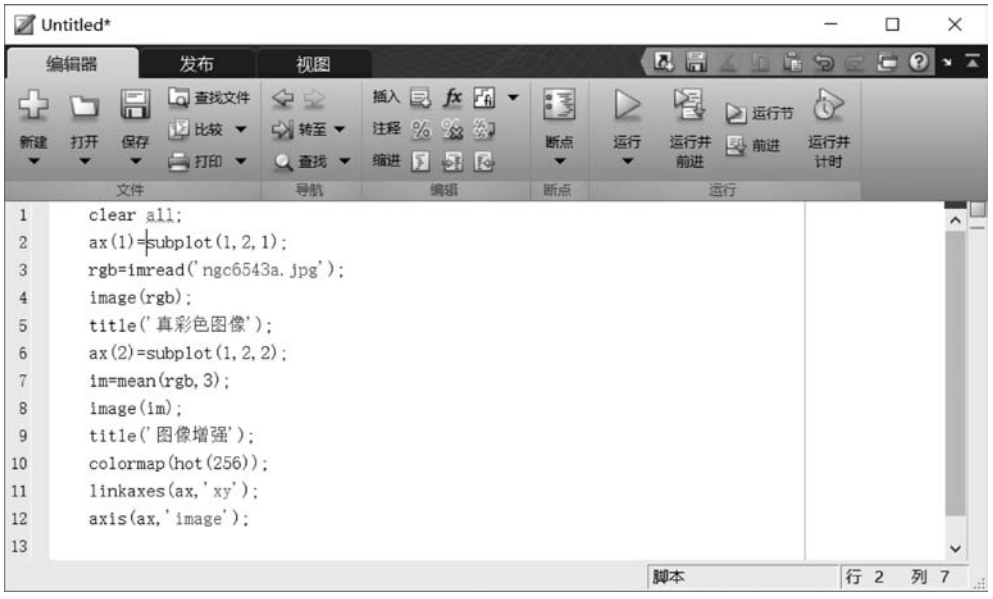

图 3-1 新建脚本文件并输入语句

选中所编写的文件并右击，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 3-2 所示。

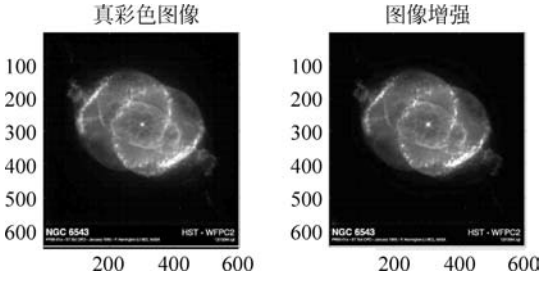

图 3-2 图像绘制

3.4.2 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,其一般格式如下:

```
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3-19】 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
```

函数文件包括以下几部分。

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,虽没做特殊要求,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 这两个 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。

3.5 MATLAB 图形绘制

使用绘图可以用可视化形式呈现数据。例如，可以比较多组数据、跟踪数据随时间所发生的更改或显示数据分布。使用图形函数或交互使用 MATLAB 桌面顶部的绘图选项卡，以编程方式创建绘图。

3.5.1 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，其调用格式主要如下：

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、Y 中一个是向量而另一个是矩阵，则矩阵的各维中必须有一维与向量的长

度相等。如果矩阵的行数等于向量长度，则 plot 函数绘制矩阵中的每一列对向量的图。如果矩阵的列数等于向量长度，则该函数绘制矩阵中的每一行对向量的图。如果矩阵为方阵，则该函数绘制每一列对向量的图。

- 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')。

plot(X,Y,LineSpec)：LineSpec 用于设置线型、标记符号和颜色。LineSpec 的标准设定值如表 3-8 所示，前七种颜色依序(蓝、绿、红、青、品红、黄、黑)自动着色。

表 3-8　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
-.	点画线	+	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	品红	<	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	—	—

plot(X1,Y1,…,Xn,Yn)：绘制多个 X、Y 对组的图，所有线条都使用相同的坐标区。

plot(X1,Y1,LineSpec1,…,Xn,Yn,LineSpecn)：设置每个线条的线型、标记符号和颜色。可以混用 X、Y、LineSpec 三元组和 X、Y 对组：例如，plot(X1,Y1,X2,Y2,LineSpec2,X3,Y3)。

plot(Y)：创建 Y 中数据对每个值索引的二维线图。

- 如果 Y 是向量，x 轴的刻度范围是从 1 至 length(Y)。
- 如果 Y 是矩阵，则 plot 函数绘制 Y 中各列对其行号的图。x 轴的刻度范围是从 1 到 Y 的行数。
- 如果 Y 是复数，则 plot 函数绘制 Y 的虚部对 Y 的实部的图，使得 plot(Y)等效于 plot(real(Y),imag(Y))。

plot(Y,LineSpec)：设置线型、标记符号和颜色。

plot(___,Name,Value)：使用一个或多个 Name,Value 对组(名称-值对组)参数指定线条属性。有关属性列表如表 3-9 所示。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。名称-值对组设置将应用于绘制的所有线条。

表 3-9 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

plot(ax,___)：将在由 ax 指定的坐标区中，而不是在当前坐标区(gca)中创建线条。选项 ax 可以位于前面的语法中的任何输入参数组合之前。

h = plot(___)：返回由图形线条对象组成的列向量。在创建特定的图形线条后，可以使用 h 修改其属性。

【例 3-20】 绘制三条正弦曲线，每条曲线之间存在较小的相移。第一条正弦曲线使用绿色线条，不带标记。第二条正弦曲线使用蓝色虚线，带圆形标记。第三条正弦曲线只使用青色星号标记。

```
>> x = 0:pi/10:2 * pi;
y1 = sin(x);
y2 = sin(x - 0.25);
y3 = sin(x - 0.5);
figure
plot(x,y1,'g',x,y2,'b -- o',x,y3,'c * ')
```

运行程序，效果如图 3-3 所示。

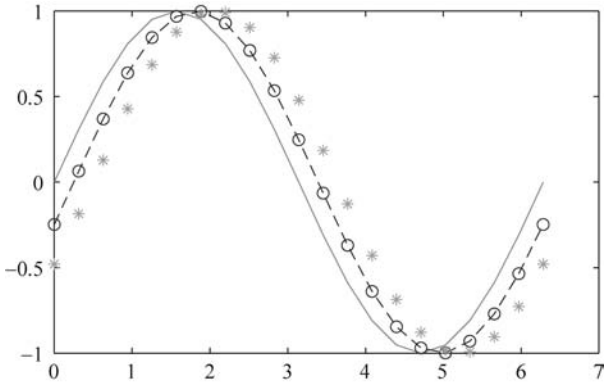

图 3-3 三条正弦曲线

彩色图片

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) 如果对已绘制的图形不满意，可提出更具体的要求，如坐标轴范围、绘制网格等。

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，MATLAB 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 3-10 所示。

表 3-10　常用图形修饰函数及说明

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 3-21】 对绘制的正弦曲线进行修饰，实现以下要求：

(1) 将图形的 x 轴大小范围限定在[0,2π]，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) 设置两条曲线的线型、线条大小。

(6) 在两条曲线上分别标注文本 y1=sin(x),y2=cos(x-0.25)。

(7) 给图形添加网格。

其实现的 MATLAB 代码如下：

```
>> clear all;   % 清除工作空间变量
x = 0:pi/10:2 * pi;
y1 = sin(x);
y2 = cos(x - 0.25);
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 - 0.25)');
```

运行程序，效果如图 3-4 所示。

如图 3-4 所示，脚本在执行第一个 gtext 命令时，需要在图形窗口 Figure1 中确定该文本的位置。

Figure1 上可以看到一个跟随鼠标移动的十字形指针，将鼠标指针拖动到对应曲线附近，然后单击，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−0.25)添加到图形中，最终效果如图 3-5 所示。

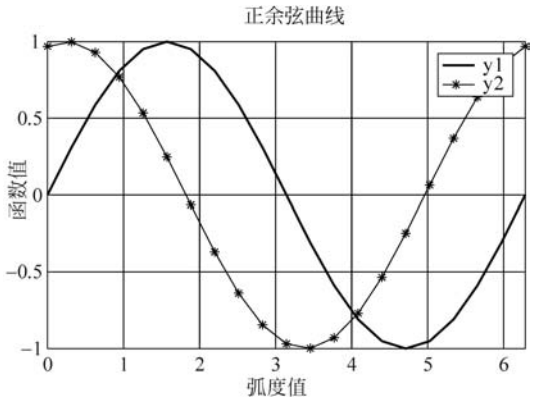

图 3-4　输出图形(未添加文本说明)

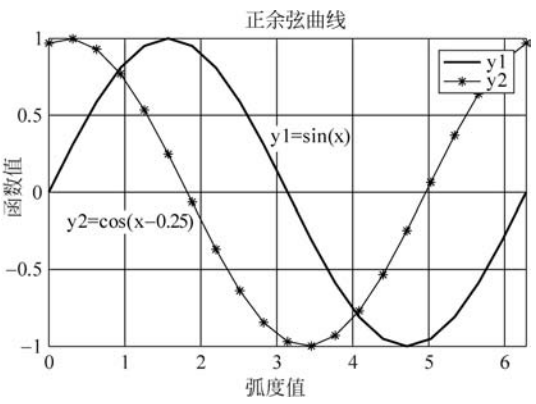

图 3-5　最终输出图形

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 3-11 列出了 MATLAB 自带的常用的特殊二维图形函数。

表 3-11　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 函数用于绘制水平二维条形图
hist	用于绘制直方图
are	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	用于绘制二维向量图
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 3-22】 利用 MATLAB 提供的特殊函数绘制特殊二维图形。

其实现的 MATLAB 代码如下:

```
>> 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)                    % 向量图
colormap hsv;
title('带等高线的向量图');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 3-6 所示。

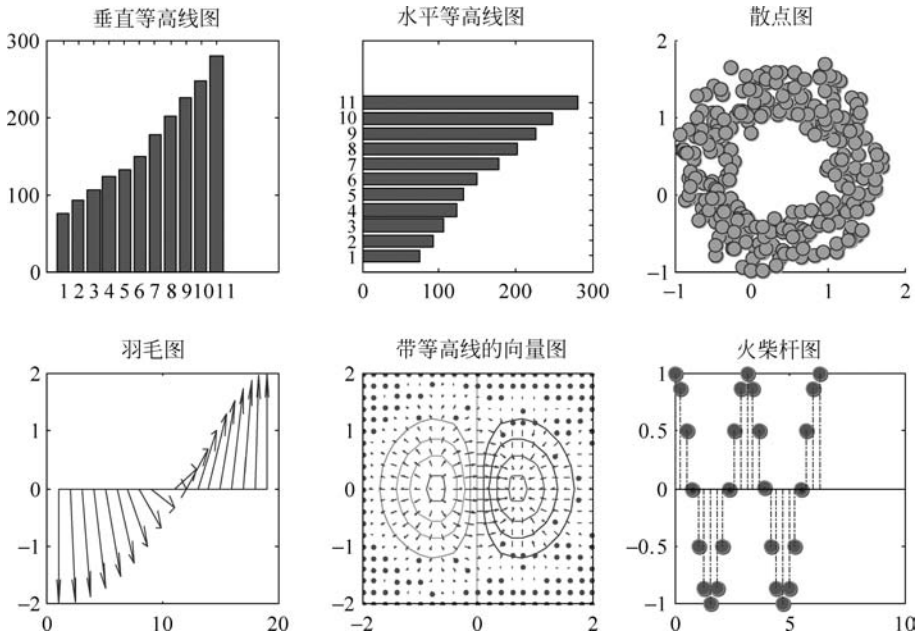

图 3-6　二维特殊图形

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，也可以同时打开几个窗口，还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3-12所示。

表3-12 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
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Figure1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3-23】 取三个不同的x值，x1=0:pi/20:pi，x2= pi/2:pi/20:3*pi/2，x3=pi:pi/20:2*pi，在同一坐标系下绘制y1=sin(x)，y2=sin(x-0.25)，y3=sin(x-0.5)的图形，并利用hold on绘图。

其实现的MATLAB代码如下：

```
>> 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 3-7 所示。

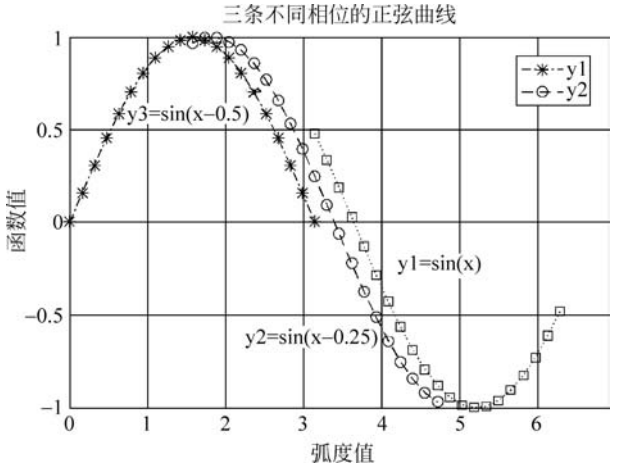

图 3-7　绘图结果

注意:在程序中,绘制三条曲线的命令的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 3-8 所示的结果,只显示最后一个 plot 绘制结果,也即是 y3。

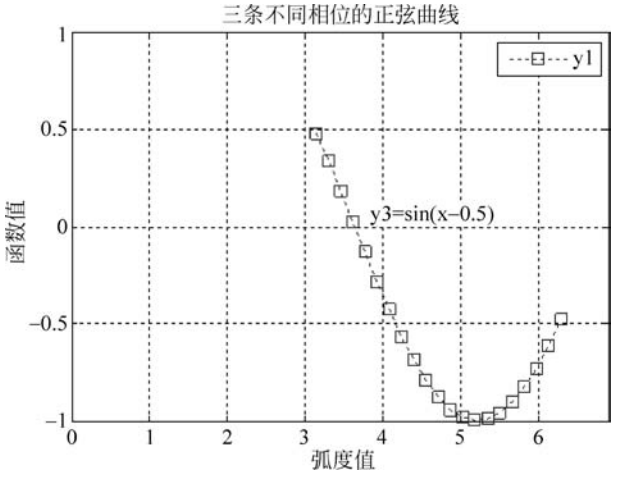

图 3-8　去掉 hold on 的绘图效果

3.5.2　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 的调用格式如下所述。

plot3(X,Y,Z)：绘制三维空间中的坐标。

- 要绘制由线段连接的一组坐标，请将 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，请将 X、Y 或 Z 中的至少一个指定为矩阵，其他指定为向量。

plot3(X,Y,Z,LineSpec)：使用指定的线型、标记和颜色创建绘图。

plot3(X1,Y1,Z1,…,Xn,Yn,Zn)：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。

plot3(X1,Y1,Z1,LineSpec1,…,Xn,Yn,Zn,LineSpecn)：可为每个 XYZ 三元组指定特定的线型、标记和颜色。可以对某些三元组指定 LineSpec，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标记，但没有对第二个三元组指定标记。

plot3(___,Name,Value)：使用一个或多个名称-值对组参数指定 Line 属性。在所有其他输入参数后指定属性。

plot3(ax,___)：在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。

p = plot3(___)：返回一个 Line 对象或 Line 对象数组。创建绘图后，使用 p 修改该绘图的属性。

【例 3-24】 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 3-9 所示。

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 3-9 添加标注，代码如下：

```
>> 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 3-10 所示。

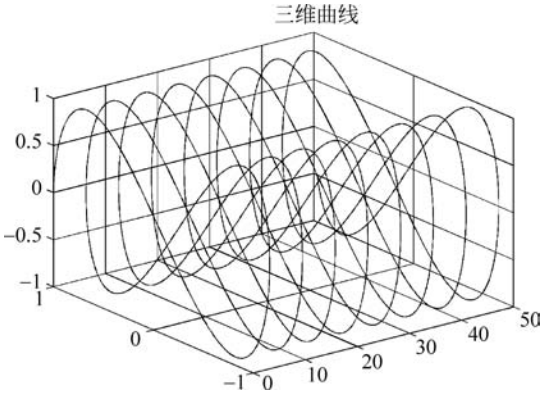

图 3-9　三维曲线的绘制

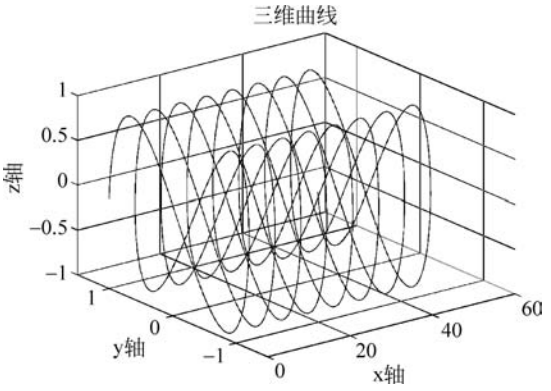

图 3-10　添加标注

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在 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 3-13 所示。

表 3-13　三维曲面绘制函数

函　　数	说　　明
[X,Y]=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
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
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 3-25】 绘制三维网格实例。

```
>> 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 3-11 所示。

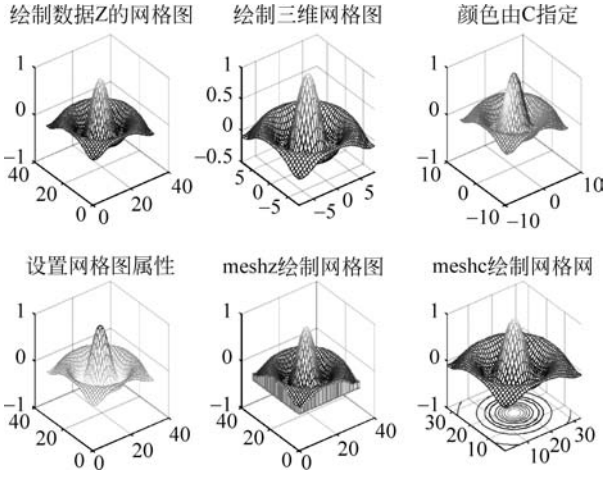

图 3-11　三维网格图

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 3-14 列出了 MATLAB 常用的三维特殊图形绘图函数。

表 3-14　三维特殊图形绘图函数

函　　数	说　　明
bar3、barh3	用于绘制三维垂直(水平)柱状图
cylinder	用于绘制三维柱面图
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	用于绘制三维片块图

【例 3-26】 绘制特殊三维图形。

其实现的 MATLAB 程序代码如下:

```
>> 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 3-12 所示。

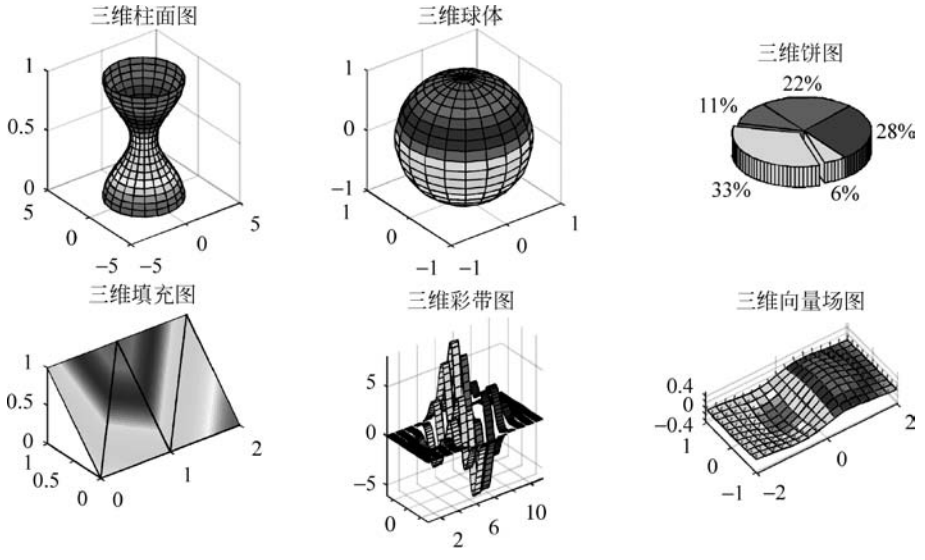

图 3-12　特殊三维效果图