第 2 章 基 本 运 算

CHAPTER 2

本章主要介绍 MATLAB 的基本运算，包括 MATLAB 的数据类型、矩阵运算、关系运算和逻辑运算、多项式运算、符号运算、复数运算等。通过本章的学习，读者能够熟练掌握 MATLAB 的基本运算，从而能对 MATLAB 的运算功能有一个全面的认识，并开始利用 MATLAB 解决科学工程问题。

学习目标：

- 掌握 MATLAB 的数据类型；
- 掌握数组及矩阵的表示方法及运算；
- 熟练掌握符号运算方法；
- 掌握复数及其运算；
- 掌握多项式的求解。

2.1 MATLAB 的数据类型

MATLAB 数据类型的最大特点是每种类型的数据都是以矩阵的形式存在的。MATLAB 常用的数据类型包括数值型、字符型、元胞数组、结构体和函数句柄等，其中数值型又包括双精度类型、单精度类型和整型。MATLAB 支持不同数据类型间的转换，增加了数据处理的灵活性。

2.1.1 变量和常量

1. 变量

变量是任何程序设计语言的基本元素之一。相比 C、Fortran 语言等其他高级语言，MATLAB 在变量声明方面要求并不严格，其变量无须事先声明，也不需要指定变量类型，MATLAB 会自动根据变量的赋值与其相关操作来确定变量的类型。

MATLAB 变量的命名规则有以下 3 种：

（1）变量名区分大小写。

（2）变量名必须以字母开头，可包含字母（大小写）、数字和下画线。

（3）变量名长度不超过 63 个字符。

2. 常量

MATLAB 中存在一些预定义的特殊变量，称为常量，MATLAB 的常用常量如表 2-1 所示。

表 2-1　MATLAB的常用常量

常　量	说　明	常　量	说　明
i,j	虚数单位，定义为$\sqrt{-1}$	eps	浮点运算的相对精度
pi	圆周率	realmax	最大的正实数
Inf	无穷大	realmin	最小的正实数
NaN	不定值（0/0）	ans	默认变量名

在 MATLAB 中，定义变量时应避免与常量名相同，以免改变常数的值，给计算带来不便。

【例 2-1】在 MATLAB 中查看预定义常量。

解：在命令行窗口中输入命令及输出结果如下：

```
>> eps
ans =
   2.2204e-16
>> pi
ans =
    3.1416
```

2.1.2　数值型数据

MATLAB 数值型数据包括整数（有符号和无符号）和浮点数（单精度和双精度），表 2–2 列出了数值型的不同格式。需要注意的是，在默认状态下，数据类型默认为双精度浮点数。

表 2-2　数值型

数　值　型		说　明	表 示 范 围
浮点型	single	单精度浮点数	$-2^{128}\sim-2^{-126},2^{-126}\sim2^{128}$
	double	双精度浮点数	$-2^{1024}\sim-2^{-1022},2^{-1022}\sim2^{1024}$
整型	int8	8位有符号整数	$-2^{7}\sim2^{7}-1$
	int16	16位有符号整数	$-2^{15}\sim2^{15}-1$
	int32	32位有符号整数	$-2^{31}\sim2^{31}-1$
	int64	64位有符号整数	$-2^{63}\sim2^{63}-1$
	uint8	8位无符号整数	$0\sim2^{8}-1$
	uint16	16位无符号整数	$0\sim2^{16}-1$
	uint32	32位无符号整数	$0\sim2^{32}-1$
	uint64	64位无符号整数	$0\sim2^{64}-1$

1. MATLAB的数值精度

MATLAB 所能表示的最小实数称为 MATLAB 的数值精度，MATLAB 的数值精度为 2^{-1074}，任何绝对值小于 2^{-1074} 的实数，MATLAB 都将其视为 0。

【例 2-2】MATLAB 的数值精度。

解：在命令行窗口中输入命令及输出结果如下：

```
>> x=2^(-1074)
x =
    4.9407e-324
>> x=2^(-1075)
x =
     0
```

由此可知，$2^{-1075}<2^{-1074}$，MATLAB 可以表示 2^{-1074}，但将 2^{-1075} 视为 0。因此 MATLAB 程序在判断某个实数是否为 0 时，最可靠的方法是判断其绝对值是否小于或等于 2^{-1075}，但实际上并不一定要与 2^{-1075} 比较，针对不同精度需求的问题，可以使用不同的比较对象，例如 MATLAB 使用内置常量 eps 作为浮点运算的相对精度，其值为 2.2204×10^{-16}。

2. MATLAB的显示精度

MATLAB 所能显示的有效位数称为 MATLAB 的显示精度。默认状态下，若数据为整数，则以整型显示；若为实数，则以保留小数点后 4 位的浮点数显示。

MATLAB 的显示格式可由 format()函数控制，需要注意的是，format()函数并不改变原数据的类型，只影响其在命令行窗口中的显示格式。此外还可以使用 digits()和 vpa()函数来控制数据的显示精度。

【例 2-3】使用 format()、short()、rat()、digits()和 vpa()函数控制显示精度。

解：在命令行窗口中输入命令及输出结果如下：

```
>> eps
ans =
   2.2204e-16
>> format long
>> eps
ans =
   2.220446049250313e-16
>> format short
>> eps
ans =
   2.2204e-16
>> format rat
>> eps
ans =
    1/4503599627370496
>> digits(10);
>> vpa(pi)
ans =
    3.141592654
>> vpa(pi,20)
ans =
    3.1415926535897932385
```

2.1.3 字符型数据

类似于其他高级语言，MATLAB 的字符和字符串运算也相当强大。在 MATLAB 中，字符串可以用单引号“'”进行赋值，字符串的每个字符（含空格）都是字符数组的一个元素。MATLAB 中还包含很多字符串操作函数，具体见表 2–3。

表 2-3　字符串操作函数

函　数　名	说　　明	函　数　名	说　　明
char	生成字符数组	strsplit	在指定的分隔符处拆分字符串
strcat	水平连接字符串	strtok	寻找字符串中记号
strvcat	垂直连接字符串	upper	转换字符串为大写
strcmp	比较字符串	lower	转换字符串为小写
strncmp	比较字符串的前*n*个字符	blanks	生成空字符串
strfind	在其他字符串中寻找此字符串	deblank	移去字符串内空格
strrep	以其他字符串代替此字符串		

【例 2-4】字符型数据操作。

解：在命令行窗口中输入命令及输出结果如下：

```
>> syms a b
>> y=2*a+1
y =
    2*a + 1
>> y1=a+2;                          % 表示不在命令行窗口显示
>> y2=y-y1                          % 字符串的相减运算操作
y2 =
    a - 1
>> y3=y+y1                          % 字符串的相加运算操作
y3 =
    3*a + 3
>> y4=y*y1                          % 字符串的相乘运算操作
y4 =
    (2*a + 1)*(a + 2)
>> y5=y/y1                          % 字符串的相除运算操作
y5 =
    (2*a + 1)/(a + 2)
```

2.1.4　元胞数组

元胞数组是 MATLAB 语言中一种特殊的数据类型。元胞数组的基本组成单位是元胞，元胞可以存放任意类型、任意大小的数组，而且同一个元胞数组中每个元胞的内容可以不同。

MATLAB 中元胞数组可通过赋值语句直接定义，也可由 cell()函数预先分配存储空间再对元胞元素逐个赋值。元胞数组可以使用花括号{}直接定义，而使用 cell()函数创建空元胞数组可节约内存占用，提高执行效率。

MATLAB 中元胞数组的相关操作函数如表 2–4 所示。

表 2-4　元胞数组的相关操作函数

函　数　名	说　　明	函　数　名	说　　明
cell	生成元胞数组	cellfun	对元胞数组中元素指定不同的函数
cellstr	生成字符型元胞数组	iscell	判断是否为元胞数组
celldisp	显示元胞数组的内容	reshape	改变元胞数组的结构
cellplot	图形显示元胞数组的内容		

【例 2-5】元胞数组的创建。

解：在命令行窗口中输入语句及输结果如下：

```
>> A={[1,2,3],ones(3),'matlab'}                                  % 直接定义元胞数组
A =
  1×3 cell 数组
    {1×3 double}    {3×3 double}    {' matlab '}
>> B=cell(1,3);                                                   % 创建空的元胞数组
>> B{1,1}=[1,2,3];B{1,2}=ones(2);B{1,3}=' matlab ';               % 为元胞中元素赋值
>> celldisp(B)                                                    % 显示元胞数组 B
B{1} =
     1     2     3
B{2} =
     1     1
     1     1
B{3} =
   matlab
```

2.1.5 结构体

结构体是 MATLAB 语言中另一种能够存放不同类型数据的数据类型，它与元胞数组的区别在于结构体是以指针的方式传递数据的，而元胞数组则是通过值传递的。结构体与元胞数组在程序中的合理使用，能够让程序简洁易懂，操作方便。

MATLAB 中结构体的定义也有两种方式：一种是直接赋值，另一种是通过 struct()函数来定义。

直接赋值需要指出结构体的属性名称，以指针操作符“.”连接结构体变量名与属性名。对某属性进行赋值时，MATLAB 会自动生成包含此属性的结构体变量，而且在同一结构体变量中，属性的数据类型不要求完全一致，这也是 MATLAB 语言灵活性的体现。

结构体变量也可以构成数组，即结构体数组，对结构体数组进行赋值时，可以只对部分元素赋值，此时未被赋值的元素将赋以空矩阵，可以随时对其进行赋值。

使用 struct()函数定义结构体时，需采用如下调用方式：

```
结构体变量名 = struct (属性名 1，属性值 1，属性名 2，属性值 2，…)
```

MATLAB 中结构体的相关操作函数如表 2–5 所示。

表 2-5 结构体的相关操作函数

函 数 名	说 明	函 数 名	说 明
struct	生成结构体变量	isfield	判断是否为结构体变量的属性
fieldname	得到结构体变量的属性名	isstruct	判断是否为结构体变量
getfield	得到结构体变量的属性值	rmfield	删除结构体变量中的属性
setfield	设定结构体变量的属性值		

【例 2-6】结构体的创建。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A.b1=1;                                                        % 直接赋值
>> A.b2=ones(2);
>> A.b3='matlab 2016a';
>> B=struct('b1',1,'b2',ones(2),'b3','matlab 2016a');             % 使用 struct()函数赋值
```

```
>> A
A =
  包含以下字段的 struct:
    b1: 1
    b2: [2×2 double]
    b3: 'matlab 2016a'
>> B
B =
  包含以下字段的 struct:
    b1: 1
    b2: [2×2 double]
    b3: 'matlab 2016a'
```

2.1.6　函数句柄

函数句柄用于间接调用一个函数的值或数据类型。在调用其他函数时可以传递函数句柄，也可在数据结构中保存函数句柄备用。

引入函数句柄是为了使 feval 及借助于它的泛函指令工作更可靠，特别是在反复调用的情况下更显效率：使“函数调用”像“变量调用”一样方便灵活，提高函数调用速度、提高软件重用性、扩大子函数和私有函数的可调用范围，并迅速获得同名重载函数的位置和类型信息。

函数句柄可以通过命令 fhandle=@functionname 来创建，例如 trig_f=@sin 或 sqr=@(x)x.^2。

使用句柄调用函数的形式是 fhandle(arg1,arg2,...,argn)或 fhandle()（无参数）。

【例 2-7】函数句柄的创建和调用。

解：在命令行窗口中输入语句及输出结果如下：

```
>> sin_f=@sin
sin_f =
  包含以下值的 function_handle:
    @sin
>> a=sin_f(pi)
a =
      1/8165619676597685
>> myadd = @(x,y) x+y^2
myadd =
  包含以下值的 function_handle:
    @(x,y)x+y^2
>> b1=myadd(1,2)
b1 =
      5
```

2.1.7　数据类型间的转换

MATLAB 支持不同数据类型间的转换，这给数据处理带来极大方便，常用的数据类型转换函数如表 2-6 所示。

【例 2-8】数据类型之间的转换，特别对于图像而言，较多的应用、图像读入的多位 uint8 型数据，需要转换成 double 型数据进行处理。

表 2-6　常用的数据类型转换函数

函　数　名	说　　明	函　数　名	说　　明
int2str	整数→字符串	dec2hex	十进制数→十六进制数
mat2str	矩阵→字符串	hex2dec	十六进制数→十进制数
num2str	数字→字符串	hex2num	十六进制数→双精度浮点数
str2num	字符串→数字	num2hex	浮点数→十六进制数
base2dec	任意进制字符串→十进制数	cell2mat	元胞数组→数值数组
bin2dec	二进制数→十进制数	cell2struct	元胞数组→结构体数组
dec2base	十进制数→任意进制字符串	mat2cell	数值数组→元胞数组
dec2bin	十进制数→二进制数	struct2cell	结构体数组→元胞数组

解：在命令行窗口中输入如下语句：

```
>> im = imread('cameraman.tif');
>> imshow(im)
>> im1=im2double(im);
>> imshow(im)
```

图 2-1　数据类型转换

运行结果如图 2-1 所示。

继续字符型变量转换，在命令行窗口中输入语句及输出结果如下：

```
>> a = '2'
a =
    '2'
>> b=double(a)
b =
    50
>> b1=str2num(a)
b1 =
    2
>> c=2*a
c =
    100
>> d=2*b
d =
    100
>> d=2*b1
d =
    4
```

2.2　数组运算

数组运算是 MATLAB 运算的基础。由于 MATLAB 面向对象的特性，使得数值数组成为 MATLAB 最重要的一种内置数据类型，而数组运算就是定义这种数据结构的方法。本节将系统地列出具备数组运算能力的函数名称，为兼顾一般性，以二维数组的运算为例，读者可推广至多维数组和多维矩阵的运算。

下面将介绍在 MATLAB 中如何建立数组，以及数组的常用操作等，包括数组的算术运算、关系运算和逻辑运算。

2.2.1　数组的创建和操作

在 MATLAB 中一般使用方括号“[ ]”、逗号“,”、空格和分号“;”来创建数组，数组中同一行的元素使用逗号或空格进行分隔，不同行之间用分号进行分隔。

【例 2-9】创建空数组、行向量和列向量。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[]
A =
    []
>> B=[5 4 3 2 1]
B =
    5    4    3    2    1
>> C=[5,4,3,2,1]
C =
    5    4    3    2    1
>> D=[5; 4; 3; 2; 1]
D =
     5
     4
     3
     2
     1
>> E=B'                                  % 转置
E =
     5
     4
     3
     2
     1
```

【例 2-10】访问数组。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[5 4 3 2 1]
A =
    5    4    3    2    1
>> a1=A(1)                               % 访问数组第 1 个元素；
a1 =
    5
>> a2=A(1:3)                             % 访问数组第 1、2、3 个元素
a2 =
    5    4    3
>> a3=A(3:end)                           % 访问数组第 3 个到最后一个元素
a3 =
    3    2    1
>> a4=A(end:-1:1)                        % 数组元素反序输出
a4 =
    1    2    3    4    5
>> a5=A([1 5])                           % 访问数组第 1 及第 5 个元素
a5 =
    5    1
```

【例 2-11】子数组的赋值（assign）。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[5 4 3 2 1];
>> A(3) = 0
A =
    5    4    0    2    1
>> A([1 4])=[1 1]
A =
    1    4    0    1    1
```

在 MATLAB 中还可以通过其他各种方式创建数组，下面进行具体介绍。

1. 通过冒号创建一维数组

在 MATLAB 中，通过冒号创建一维数组的代码如下：

```
X=A:step:B
```

其中，A 是创建一维数组的第一个变量，step 是每次递增或递减的数值，直到最后一个元素和 B 的差的绝对值小于或等于 step 的绝对值为止。

【例 2-12】通过冒号创建一维数组。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=2:6
A =
    2    3    4    5    6
>> B=2.1:1.5:6
B =
   2.1000    3.6000    5.1000
>> C=2.1:-1.5:-6
C =
   2.1000    0.6000   -0.9000   -2.4000   -3.9000   -5.4000
>> D=2.1:-1.5:6
D =
  空的 1×0 double 行向量
```

2. 通过logspace()函数创建一维数组

MATLAB 常用 logspace()函数创建一维数组，该函数的调用方式如下。

```
y = logspace(a,b)          % 创建行向量 y，第一个元素为 10^a，最后一个元素为 10^b，形成总数为 50
                           % 个元素的等比数列
y = logspace(a,b,n)        % 创建行向量 y，第一个元素为 10^a，最后一个元素为 10^b，形成总数为 n
                           % 个元素的等比数列
```

【例 2-13】通过 logspace()函数创建一维数组。

解：在命令行窗口中输入语句及输出结果如下：

```
>> format short;
>> A=logspace(1,2,8)
A =
   10.0000   13.8950   19.3070   26.8270   37.2759   51.7947   71.9686  100.0000
```

3. 通过linspace()函数创建一维数组

MATLAB 常用 linspace()函数创建一维数组，该函数的调用方式如下。

```
y = linspace (a,b)              % 创建行向量 y，第一个元素为 a，最后一个元素为 b，形成总数为
                                % 100 个元素的等差数列
y = linspace (a,b,n)            % 创建行向量 y，第一个元素为 a，最后一个元素为 b，形成总数为 n
                                % 个元素的等差数列
```

【例 2-14】通过 linspace()函数创建一维数组。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A = linspace(1,10)
A =
  列 1 至 20
    1.0000   1.0909   1.1818   1.2727   1.3636   1.4545   1.5455   1.6364   1.7273   1.8182
    1.9091   2.0000   2.0909   2.1818   2.2727   2.3636   2.4545   2.5455   2.6364   2.7273
  列 21 至 40
    2.8182   2.9091   3.0000   3.0909   3.1818   3.2727   3.3636   3.4545   3.5455   3.6364
    3.7273   3.8182   3.9091   4.0000   4.0909   4.1818   4.2727   4.3636   4.4545   4.5455
  列 41 至 60
    4.6364   4.7273   4.8182   4.9091   5.0000   5.0909   5.1818   5.2727   5.3636   5.4545
    5.5455   5.6364   5.7273   5.8182   5.9091   6.0000   6.0909   6.1818   6.2727   6.3636
  列 61 至 80
    6.4545   6.5455   6.6364   6.7273   6.8182   6.9091   7.0000   7.0909   7.1818   7.2727
    7.3636   7.4545   7.5455   7.6364   7.7273   7.8182   7.9091   8.0000   8.0909   8.1818
  列 81 至 100
    8.2727   8.3636   8.4545   8.5455   8.6364   8.7273   8.8182   8.9091   9.0000   9.0909
    9.1818   9.2727   9.3636   9.4545   9.5455   9.6364   9.7273   9.8182   9.9091  10.0000
>> B = linspace(1,36,8)
B =
     1     6    11    16    21    26    31    36
>> C = linspace(1,36,1)
C =
    36
```

2.2.2　数组的常见运算

1. 数组的算术运算

数组的运算是从数组的单个元素出发，针对每个元素进行的运算。在 MATLAB 中，一维数组的基本运算包括加、减、乘、左除、右除和乘方。

（1）数组的加减运算：通过格式 $A+B$ 或 $A-B$ 可实现数组的加减运算。但是运算规则要求数组 A 和 B 的维数相同。

提示：如果两个数组的维数不相同，则将给出错误的信息。

【例 2-15】数组的加减运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 85 6 2 4 0];
>> C=[1 1 1 1 1];
>> D=A+B                                  % 加法
D =
    10    90    12    10    13     6
>> E=A-B                                  % 减法
E =
    -8   -80     0     6     5     6
>> F=A*2
F =
     2    10    12    16    18    12
>> G=A+3                                  % 数组与常数的加法
G =
     4     8     9    11    12     9
>> H=A-C
矩阵维度必须一致。                          % 错误信息
```

（2）数组的乘除运算：通过点乘格式“.*”或点除“./”可实现数组的乘除运算。但是运算规则要求数组 *A* 和 *B* 的维数相同。

乘法：数组 *A* 和 *B* 的维数相同，运算为数组对应元素相乘，计算结果与 *A* 和 *B* 是相同维数的数组。

除法：数组 *A* 和 *B* 的维数相同，运算为数组对应元素相除，计算结果与 *A* 和 *B* 是相同维数的数组。

右除和左除的关系：*A./B=B.\A*，其中 *A* 是被除数，*B* 是除数。

提示：如果两个数组的维数不相同，则将给出错误的信息。

【例 2-16】数组的乘法。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
>> C=A.* B                                % 数组的点乘
C =
    9    25    36    16    36     0
>> D=A * 3                                % 数组与常数的乘法
D =
    3    15    18    24    27    18
```

【例 2-17】数组的除法。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
>> C=A.\B                              % 数组和数组的左除，B 为被除数
>> D=A./B                              % 数组和数组的右除，A 为被除数
>> E=A./3                              % 数组与常数的除法
>> F=A/3
C =
    9.0000    1.0000    1.0000    0.2500    0.4444         0
D =
    0.1111    1.0000    1.0000    4.0000    2.2500       Inf
```

```
E =
    0.3333    1.6667    2.0000    2.6667    3.0000    2.0000
F =
    0.3333    1.6667    2.0000    2.6667    3.0000    2.0000
```

（3）通过乘方格式“.^”可实现数组的乘方运算。数组的乘方运算包括：数组间的乘方运算、数组与某个具体数值的乘方运算，以及常数与数组的乘方运算。

【例 2-18】数组的乘方。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
>> C=A.^B                                                    % 数组的乘方
C =
    1       3125      46656         64       6561          1
>> D=A.^3                                                    % 数组与数值的乘方
D =
    1   125   216   512   729   216
>> E=3.^A                                                    % 常数与数组的乘方
E =
    3        243        729       6561      19683        729
```

（4）通过函数 dot()可实现数组的点积运算，但是运算规则要求数组 *A* 和 *B* 的维数相同，其调用格式如下：

```
C= dot(A,B)
C = dot(A,B,dim)
```

【例 2-19】数组的点积。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
>> C=dot(A,B)  % 数组的点积
C =
   122
>> D=sum(A.*B) % 数组元素的乘积之和
D =
   122
```

2. 数组的关系运算

在 MATLAB 中提供了 6 种数组关系运算符，即<（小于）、<=（小于或等于）、>（大于）、>=（大于或等于）、==（恒等于）、~=（不等于）。

关系运算的运算法则如下：

（1）当两个比较量是标量时，直接比较两个数的大小。若关系成立，则返回的结果为 1，否则为 0。

（2）当两个比较量是维数相等的数组时，逐一比较两个数组相同位置的元素，并给出比较结果。最终的关系运算结果是一个与参与比较的数组维数相同的数组，其组成元素为 0 或 1。

【例 2-20】数组的关系运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
```

```
>> C=A<6                                % 数组与常数比较，小于
C =
  1×6 logical 数组
    1    1    0    0    0    0
>> D=A>=6                               % 数组与常数比较，大于或等于
D =
  1×6 logical 数组
    0    0    1    1    1    1
>> E=A<B                                % 数组与数组比较，小于
E =
  1×6 logical 数组
    1    0    0    0    0    0
>> F=A==B                               % 数组与数组比较，恒等于
F =
  1×6 logical 数组
    0    1    1    0    0    0
```

3. 数组的逻辑运算

在 MATLAB 中提供了 3 种数组逻辑运算符，即&（与）、|（或）和~（非）。逻辑运算的运算法则如下：

（1）如果是非零元素则为真，用 1 表示；反之是零元素则为假，用 0 表示。

（2）当两个比较量是维数相等的数组时，逐一比较两个数组相同位置的元素，并给出比较结果。最终的关系运算结果是一个与参与比较的数组维数相同的数组，其组成元素为 0 或 1。

（3）与运算（*a*&*b*）时，*a*、*b* 全为非零，则为真，运算结果为 1；或运算（*a*|*b*）时，只要 *a*、*b* 有一个为非零，则运算结果为 1；非运算（*~a*）时，若 *a* 为 0，运算结果为 1，*a* 为非零，运算结果为 0。

【例 2-21】数组的逻辑运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[1 5 6 8 9 6];
>> B=[9 5 6 2 4 0];
>> C=A&B                                % 与
C =
  1×6 logical 数组
    1    1    1    1    1    0
>> D=A|B                                % 或
D =
  1×6 logical 数组
    1    1    1    1    1    1
>> E=~B                                 % 非
E =
  1×6 logical 数组
    0    0    0    0    0    1
```

2.3 矩阵的运算

MATLAB 简称矩阵实验室，对于矩阵的运算，MATLAB 软件有着得天独厚的优势。

生成矩阵的方法有很多种：直接输入矩阵元素；对已知矩阵进行矩阵组合、矩阵转向、矩阵移位操作；读取数据文件；使用函数直接生成特殊矩阵。表 2–7 列出了常用的特殊矩阵生成函数。

表 2-7　常用的特殊矩阵生成函数

函　数　名	说　　明	函　数　名	说　　明
zeros	全0矩阵	eye	单位矩阵
ones	全1矩阵	company	伴随矩阵
rand	均匀分布随机矩阵	hilb	Hilbert矩阵
randn	正态分布随机矩阵	invhilb	Hilbert逆矩阵
magic	魔方矩阵	vander	Vander矩阵
diag	对角矩阵	pascal	Pascal矩阵
triu	上三角矩阵	hadamard	Hadamard矩阵
tril	下三角矩阵	hankel	Hankel矩阵

2.3.1　矩阵的生成

【例 2-22】随机矩阵输入。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=rand(5)
A =
    0.0512    0.       0.0594    0.0557    0.5681
    0.8698    0.1400    0.3752    0.6590    0.0432
    0.0422    0.2867    0.8687    0.9065    0.4148
    0.0897    0.0919    0.5760    0.1293    0.3793
    0.0541    0.1763    0.8402    0.7751    0.7090
>> A(:,1)                                          % A 中第 1 列
ans =
    0.7577
    0.7431
    0.3922
    0.6555
    0.1712
>> A(:,2)                                          % A 中第 2 列
ans =
    0.7060
    0.0318
    0.2769
    0.0462
    0.0971
>> A(:,3:5)                                        % A 中第 3~5 列
ans =
    0.8235    0.4387    0.4898
    0.6948    0.3816    0.4456
    0.3171    0.7655    0.6463
    0.9502    0.7952    0.7094
    0.0344    0.1869    0.7547
>> A(1,:)                                          % A 中第 1 行
ans =
    0.7577    0.7060    0.8235    0.4387    0.4898
>> A(2,:)                                          % A 中第 2 行
ans =
    0.7431    0.0318    0.6948    0.3816    0.4456
```

```
>> A(3:5,:)                                          % A 中第 3~5 行
ans =
    0.3922    0.2769    0.3171    0.7655    0.6463
    0.6555    0.0462    0.9502    0.7952    0.7094
    0.1712    0.0971    0.0344    0.1869    0.7547
```

【例 2-23】矩阵的运算（乘法、点乘、除法、减法和加法）。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[0.0512    0.4141    0.0594    0.0557    0.5681;
     0.8698    0.1400    0.3752    0.6590    0.0432;
     0.0422    0.2867    0.8687    0.9065    0.4148;
     0.0897    0.0919    0.5760    0.1293    0.3793;
     0.0541    0.1763    0.8402    0.7751    0.7090];
>> A^2                                               % 矩阵的乘法运算
ans =
    0.4011    0.2015    0.7194    0.7772    0.4955
    0.2436    0.5555    0.8460    0.5994    0.9364
    0.3919    0.4631    1.7354    1.4175    1.0347
    0.1410    0.2939    0.9334    0.8985    0.6118
    0.2995    0.4842    1.8414    1.5305    1.1836
>> A.^2                                              % 矩阵的点乘运算
ans =
    0.0026    0.1715    0.0035    0.0031    0.3227
    0.7565    0.0196    0.1408    0.4343    0.0019
    0.0018    0.0822    0.7547    0.8217    0.1721
    0.0080    0.0085    0.3318    0.0167    0.1439
    0.0029    0.0311    0.7059    0.6008    0.5026
>> A^2\A.^2                                          % 矩阵的除法运算
ans =
    0.2088    0.5308   -0.4762    0.8505   -0.0382
    1.3631   -0.1769    1.1661    0.8143   -4.2741
   -0.3247   -0.0898    1.5800    2.7892   -1.0326
   -0.5223    0.0537   -0.5715   -2.4802    0.4729
    0.5725    0.0345   -1.4792   -1.1727    3.1778
>> A^2-A.^2                                          % 矩阵的减法运算
ans =
    0.3984    0.0300    0.7159    0.7741    0.1728
   -0.5129    0.5359    0.7052    0.1652    0.9345
    0.3901    0.3810    0.9807    0.5958    0.8626
    0.1330    0.2854    0.6016    0.8818    0.4679
    0.2965    0.4531    1.1355    0.9297    0.6809
>> A^2+A.^2                                          % 矩阵的加法运算
ans =
    0.4037    0.3730    0.7229    0.7803    0.8182
    1.0001    0.5751    0.9868    1.0337    0.9383
    0.3937    0.5453    2.4901    2.2392    1.2068
    0.1491    0.3023    1.2652    0.9152    0.7558
    0.3024    0.5153    2.5473    2.1314    1.6862
```

【例 2-24】生成 Hankel 矩阵、Hibert 矩阵及 Hibert 逆矩阵。

解：在命令行窗口中输入语句及输出结果如下：

```
>> c=[1:3];
>> r=[3:9];
>> H=hankel(c,r)                                     % 生成 Hankel 矩阵
```

```
H =
     1     2     3     4     5     6     7
     2     3     4     5     6     7     8
     3     4     5     6     7     8     9
>> A=hilb(5)                                    % Hilbert 矩阵
A =
    1.0000    0.5000    0.3333    0.2500    0.2000
    0.5000    0.3333    0.2500    0.2000    0.1667
    0.3333    0.2500    0.2000    0.1667    0.1429
    0.2500    0.2000    0.1667    0.1429    0.1250
    0.2000    0.1667    0.1429    0.1250    0.1111
>> format rat
>> A
A =
      1            1/2          1/3          1/4          1/5
      1/2          1/3          1/4          1/5          1/6
      1/3          1/4          1/5          1/6          1/7
      1/4          1/5          1/6          1/7          1/8
      1/5          1/6          1/7          1/8          1/9
>> format short
>> B=invhilb(5)                                 % Hilbert 逆矩阵
B =
     25         -300         1050        -1400          630
   -300         4800       -18900        26880       -12600
   1050       -18900        79380      -117600        56700
  -1400        26880      -117600       179200       -88200
    630       -12600        56700       -88200        44100
```

2.3.2　向量的生成

向量是指单行或单列的矩阵，是组成矩阵的基本元素之一。在求某些函数值或曲线时，常常要设定自变量的一系列值，因此除了直接使用“[]”生成向量，MATLAB 还提供了两种为等间隔向量赋值的简单方法。

1. 使用冒号表达式生成向量

冒号表达式的格式为 x=[初值 x_0:增量:终值 x_n]。这里需要注意以下几点：

（1）生成的向量尾元素并不一定是终值 x_n，当 x_n-x_0 恰好为增量的整数倍时，x_n 才为尾元素。

（2）当 $x_n>x_0$ 时，增量必须为正值；当 $x_n<x_0$ 时，增量必须为负值；当 $x_n=x_0$ 时，向量只有一个元素。

（3）当增量为 1 时，增量值可以略去，直接写成 x=[初值 x_0:终值 x_n]。

（4）方括号“[]”可以删去。

2. 使用linspace()函数生成向量

linspace()函数的调用格式如下：

```
y =linspace(x1,xn,n)     % 生成包含 x1 和 xn 之间的 n 个等间距点的行向量 y，省略 n 默认为 100
```

有时需要生成对数等比向量，可以使用 logspace()函数，其调用格式：

```
y =logspace(a,b,n)       % 从 10 的 a 次幂到 b 次幂等比生成 n 个点组成的行向量 y，省略 n 默认为 50
```

【例 2-25】生成等间隔向量及对数等比向量。

解：在命令行窗口中输入语句及输出结果如下：

```
>> t=1:3:20
t =
```

```
     1     4     7    10    13    16    19
>> t=10:-3:-20
t =
    10     7     4     1    -2    -5    -8   -11   -14   -17   -20
>> t=1:2:1
t =
     1
>> t=1:5
t =
     1     2     3     4     5
>> t=linspace(1,10,5)                    % 生成等间隔向量
t =
    1.0000    3.2500    5.5000    7.7500   10.0000
>> t=logspace(0,1,8)                     % 生成对数等比向量
t =
    1.0000    1.3895    1.9307    2.6827    3.7276    5.1795    7.1969   10.0000
```

矩阵的加、减、乘、除、比较运算和逻辑运算等代数运算是 MATLAB 数值计算最基础的部分。下面重点介绍这些运算。

2.3.3 矩阵的加减运算

进行矩阵加法、减法运算的前提是参与运算的两个矩阵或多个矩阵必须具有相同的行数和列数，即 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 等多个矩阵均为 $m \times n$ 矩阵，在 MATLAB 中允许有一个或多个操作数为标量。

在上述前提下，对于同型的两个矩阵，其加减法定义如下：

$\boldsymbol{C}=\boldsymbol{A} \pm \boldsymbol{B}$，矩阵 $\boldsymbol{C}$ 的各元素 $C_{mn}=A_{mn} \pm B_{mn}$。

当其中含有标量 x 时，$\boldsymbol{C}=\boldsymbol{A} \pm x$ 表示矩阵 $\boldsymbol{C}$ 的各元素 $C_{mn}=A_{mn} \pm x$。

由于矩阵的加法运算归结为其元素的加法运算，因此容易验证矩阵的加法运算满足下列运算律：

（1）交换律：$\boldsymbol{A}+\boldsymbol{B}=\boldsymbol{B}+\boldsymbol{A}$。

（2）结合律：$\boldsymbol{A}+(\boldsymbol{B}+\boldsymbol{C})=(\boldsymbol{A}+\boldsymbol{B})+\boldsymbol{C}$。

（3）存在零元：$\boldsymbol{A}+\boldsymbol{0}=\boldsymbol{0}+\boldsymbol{A}=\boldsymbol{A}$。

（4）存在负元：$\boldsymbol{A}+(-\boldsymbol{A})=(-\boldsymbol{A})+\boldsymbol{A}$。

【例 2-26】矩阵加减法运算。已知矩阵 $\boldsymbol{A}$= [10 5 79 4 2;1 0 66 8 2;4 6 1 1 1]，矩阵 $\boldsymbol{B}$= [9 5 3 4 2;1 0 4 –23 2;4 6 –1 1 0]，行向量 $\boldsymbol{C}$= [2 1]，x=20，试求 $\boldsymbol{A}+\boldsymbol{B}$、$\boldsymbol{A}-\boldsymbol{B}$、$\boldsymbol{A}+\boldsymbol{B}+x$、$\boldsymbol{A}-x$、$\boldsymbol{A}-\boldsymbol{C}$。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A = [10 5 79 4 2;1 0 66 8 2;4 6 1 1 1];
>> B = [9 5 3 4 2;1 0 4 -23 2;4 6 -1 1 0];
>> C = [2 1];
>> x = 20;
>> ApB= A+B
ApB =
    19    10    82     8     4
     2     0    70   -15     4
     8    12     0     2     1
>> AmB= A-B
AmB =
     1     0    76     0     0
     0     0    62    31     0
     0     0     2     0     1
```

```
>> ApBpX= A+B+x
ApBpX =
    39    30   102    28    24
    22    20    90     5    24
    28    32    20    22    21
>> AmX= A-x
AmX =
   -10   -15    59   -16   -18
   -19   -20    46   -12   -18
   -16   -14   -19   -19   -19
>> AmC= A-C
矩阵维度必须一致。
```

在 $\boldsymbol{A}-\boldsymbol{C}$ 的运算中，MATLAB 返回错误信息，并提示矩阵维数必须一致。这也证明了矩阵进行加减法运算必须满足一定的前提条件。

2.3.4　矩阵的乘法运算

MATLAB 中矩阵的乘法运算包括两种：数与矩阵的乘法、矩阵与矩阵的乘法。

1. 数与矩阵的乘法

由于单个数在 MATLAB 中是以标量存储的，因此数与矩阵的乘法也可以称为标量与矩阵的乘法。

设 x 为一个数，$\boldsymbol{A}$ 为矩阵，则定义 x 与 $\boldsymbol{A}$ 的乘积 $\boldsymbol{C}=x\boldsymbol{A}$ 仍为一个矩阵，$\boldsymbol{C}$ 的元素就是用数 x 乘矩阵 $\boldsymbol{A}$ 中对应的元素而得到，即 $C_{mnx}=xA_{mn}$。数与矩阵的乘法满足下列运算律：

（1）$1\boldsymbol{A}=\boldsymbol{A}$。

（2）$x(\boldsymbol{A}+\boldsymbol{B})=x\boldsymbol{A}+x\boldsymbol{B}$。

（3）$(x+y)\boldsymbol{A}=x\boldsymbol{A}+y\boldsymbol{A}$。

（4）$(xy)\boldsymbol{A}=x(y\boldsymbol{A})=y(x\boldsymbol{A})$。

【例 2-27】矩阵数乘。已知矩阵 $\boldsymbol{A}$= [0 3 3;1 1 0;−1 2 3] ，$\boldsymbol{E}$ 是三阶单位矩阵，$\boldsymbol{E}$= [1 0 0;0 1 0;0 0 1]，试求表达式 $2\boldsymbol{A}+3\boldsymbol{E}$。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A = [0 3 3;1 1 0;-1 2 3];
>> E = eye(3);
>> R=2*A+3*E
R =
     3     6     6
     2     5     0
    -2     4     9
```

2. 矩阵与矩阵的乘法

两个矩阵的乘法必须满足被乘矩阵的列数与乘矩阵的行数相等。设矩阵 $\boldsymbol{A}$ 为 $m\times h$ 矩阵，$\boldsymbol{B}$ 为 $h\times n$ 矩阵，则两矩阵的乘积 $\boldsymbol{C}=\boldsymbol{A}\times\boldsymbol{B}$ 为一个矩阵，且 $C_{mn}=\sum_{i=1}^{h}A_{mi}\times B_{in}$ 。

矩阵之间的乘法不遵循交换律，即 $\boldsymbol{A}\times\boldsymbol{B}\neq\boldsymbol{B}\times\boldsymbol{A}$。但矩阵乘法遵循下列运算律。

（1）结合律：$(\boldsymbol{A}\times\boldsymbol{B})\times\boldsymbol{C}=\boldsymbol{A}\times(\boldsymbol{B}\times\boldsymbol{C})$。

（2）左分配律：$\boldsymbol{A}\times(\boldsymbol{B}+\boldsymbol{C})=\boldsymbol{A}\times\boldsymbol{B}+\boldsymbol{A}\times\boldsymbol{C}$。

（3）右分配律：$(\boldsymbol{B}+\boldsymbol{C})\times\boldsymbol{A}=\boldsymbol{B}\times\boldsymbol{A}+\boldsymbol{C}\times\boldsymbol{A}$。

（4）单位矩阵的存在性：$E \times A=A$，$A \times E=A$。

【例 2-28】矩阵乘法。已知矩阵 A=[2 1 4 0;1 −1 3 4]，矩阵 B= [1 3 1;0 −1 2;1 −3 1;4 0 −2]，试求矩阵乘积 AB 及 BA。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A = [2 1 4 0;1 -1 3 4];
>> B = [1 3 1;0 -1 2;1 -3 1;4 0 -2];
>> R1= A*B
R1 =
     6    -7     8
    20    -5    -6
>> R2= B*A
错误使用  *
用于矩阵乘法的维度不正确。请检查并确保第一个矩阵中的列数与第二个矩阵中的行数匹配。要执行按元素相乘，
请使用 '.*'。
```

2.3.5 矩阵的除法运算

矩阵的除法是乘法的逆运算，分为左除和右除两种，分别用运算符号“\”和“/”表示。如果 A 和 B 是标量，那么 A/B 和 $A\backslash B$ 是等价的。对于一般的二维矩阵 A 和 B，当进行 $A\backslash B$ 运算时，要求 A 的行数与 B 的行数相等；当进行 A/B 运算时，要求 A 的列数与 B 的列数相等。

【例 2-29】矩阵除法。设矩阵 A= [1 2;1 3]，矩阵 B= [1 0;1 2]，试求 $A\backslash B$ 和 A/B。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A = [1 2;1 3];
>> B = [1 0;1 2];
>> R1=A\B
R1 =
     1    -4
     0     2
>> R2=A/B
R2 =
         0    1.0000
   -0.5000    1.5000
```

2.4 矩阵的基本函数运算

MATLAB 支持多种矩阵的函数，常用的矩阵函数如表 2–8 所示。

表 2-8 MATLAB常用的矩阵函数

函 数 名	说 明	函 数 名	说 明
det	求矩阵的行列式	fliplr	矩阵左右翻转
inv	求矩阵的逆	flipud	矩阵上下翻转
eig	求矩阵的特征值和特征向量	resharp	矩阵阶数重组
rank	求矩阵的秩	rot90	矩阵逆时针旋转90°
trace	求矩阵的迹	diag	提取或建立对角阵
norm	求矩阵的范数	tril	取矩阵的左下三角部分
poly	求矩阵特征方程的根	triu	取矩阵的右上三角部分

【例 2-30】计算矩阵的特征值、特征向量及矩阵的逆、矩阵的范数。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[8,1,6; 3,5,7; 4,9,2];
>> [x,y]=eig(A)                          % x 为特征向量矩阵，y 为特征值矩阵
x =
   -0.5774   -0.8131   -0.3416
   -0.5774    0.4714   -0.4714
   -0.5774    0.3416    0.8131
y =
   15.0000         0         0
         0    4.8990         0
         0         0   -4.8990
>> B=inv(A)                              % 矩阵的逆
B =
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
>> C=norm(A)                             % 矩阵的范数
C =
    15.0000
```

2.4.1　矩阵的分解运算

矩阵的分解常用于求解线性方程组，常用的矩阵分解运算的函数如表 2–9 所示。

表 2-9　MATLAB矩阵分解运算的函数

函 数 名	说　明	函 数 名	说　明
eig	特征值分解	chol	Cholesky分解
svd	奇异值分解	qr	QR分解
lu	LU分解	schur	Schur分解

【例 2-31】矩阵分解运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[8,1,6;3,5,7;4,9,2];
>> [U,S,V]=svd(A)                        % 矩阵的奇异值分解，A=U*S*V'
U =
   -0.5774    0.7071    0.4082
   -0.5774    0.0000   -0.8165
   -0.5774   -0.7071    0.4082
S =
   15.0000         0         0
         0    6.9282         0
         0         0    3.4641
V =
   -0.5774    0.4082    0.7071
   -0.5774   -0.8165   -0.0000
   -0.5774    0.4082   -0.7071
```

奇异值分解在某些方面与对称矩阵或 Hermite 矩阵基于特征向量的对角化类似。

【例 2-32】对已知矩阵奇异值分解。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=rand(4)
A =
    0.8723    0.0655    0.2104    0.4736
    0.4546    0.6575    0.2797    0.9138
    0.3999    0.4190    0.9981    0.1929
    0.6204    0.4577    0.1198    0.9293
>> [V,D,U]=svd(A)
V =
   -0.4257    0.1086   -0.8529   -0.2821
   -0.5766    0.1895    0.5127   -0.6072
   -0.4161   -0.8954    0.0433    0.1522
   -0.5596    0.3879    0.0882    0.7271
D =
    2.0785         0         0         0
         0    0.8795         0         0
         0         0    0.5765         0
         0         0         0    0.0304
U =
   -0.5518    0.0721   -0.7613   -0.3328
   -0.4029   -0.0749    0.5893   -0.6962
   -0.3527   -0.8771    0.0307    0.3245
   -0.6393    0.4689    0.2688    0.5470
```

2.4.2 关系运算和逻辑运算

除了传统的数学运算外，MATLAB 还支持关系运算和逻辑运算。通常这些运算符和函数的目的是给出命题的真假，从而控制基于真假命题的 MATLAB 命令的流程或执行次序。

当关系表达式和逻辑表达式做输入时，MATLAB 将任何非 0 数值都当作真，而将 0 当作假；而关系表达式和逻辑表达式的输出，真输出 1，假输出 0。

表 2–10 给出了基本的关系运算符和逻辑运算符。

表 2-10　基本的关系运算符和逻辑运算符

运算符	符　　号	功　　能	对 应 函 数
关系运算符	==	等于	eq()
	~=	不等于	ne()
	<	小于	lt()
	>	大于	gt()
	<=	小于或等于	le()
	>=	大于或等于	ge()
逻辑运算符	&	逻辑与	and()
	|	逻辑或	or()
	~	逻辑非	not()

【例 2-33】关系运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=reshape(1:9,3,3)
A =
     1     4     7
     2     5     8
     3     6     9
>> B=magic(3)
B =
     8     1     6
     3     5     7
     4     9     2
>> A>B
ans =
  3×3 logical 数组
   0   1   1
   0   0   1
   0   0   1
>> A==B
ans =
  3×3 logical 数组
   0   0   0
   0   1   0
   0   0   0
```

【例 2-34】逻辑运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A= 1; B = 0; C = -10;
>> D1=~A
D1 =
  logical
   0
>> D2= A | B
D2 =
  logical
   1
>> D3= A & B
D3 =
  logical
   0
>> D4= A & B | C
D4 =
  logical
   1
>> D5= A & (B | C)
D5 =
  logical
   1
>> D6= ~(A & C)
D6=
  logical
   0
```

此外，MATLAB 还提供相关的关系运算函数和逻辑运算函数，如表 2–11 所示。

表 2-11　关系运算函数和逻辑运算函数

函　数　名	说　　明	函　数　名	说　　明
any	任意元素不为0时为真	all	所有元素均不为0时为真
xor	逻辑异或运算	find	寻找非0元素坐标
bitand	位方式的逻辑与运算	bitor	位方式的逻辑或运算
bitxor	位方式的逻辑异或运算	bitcmp	位比较运算
bitshift	二进制移位运算		

【例 2-35】关系运算函数和逻辑运算函数的应用。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A= [1 2 0 4; 2 2 0 1]; B = [2 0 0 4; 0 0 2 4];
>> C1= xor(A, B)
C1 =
  2×4 logical 数组
   0   1   0   0
   1   1   1   0
>> C2= any(A)
C2 =
  1×4 logical 数组
   1   1   0   1
>> C3= all(B)
C3 =
  1×4 logical 数组
   0   0   0   1
```

2.5　符号运算

MATLAB 不仅在数值计算功能方面相当出色，在符号运算方面也提供了专门的符号数学工具箱(Symbolic Math Toolbox)——MuPAD Notebook。

符号数学工具箱是操作和求解符号表达式的符号函数的集合，其功能主要包括符号表达式与符号矩阵的基本操作、符号微积分运算以及求解代数方程和微分方程。

符号运算与数值运算的主要区别在于：数值运算必须先对变量赋值，才能进行运算；符号运算无须事先对变量进行赋值，运算结果直接以符号的形式输出。

2.5.1　符号表达式的生成

在符号运算中，数字、函数、算子和变量都是以字符的形式保存并进行运算的。符号表达式包括符号函数和符号方程，两者的区别在于前者不包括等号，后者必须带等号，但它们的创建方式是相同的。

MATLAB 中创建符号表达式的方法有两种：一种是直接使用字符串变量的生成方法对其进行赋值；另一种是根据 MATLAB 提供的符号变量定义函数 sym()和 syms()。

sym()函数用来定义单个符号变量，调用格式为：

```
符号变量名=sym('符号字符串')                % 其中符号字符串可以是常量、变量、函数或表达式
```

syms()函数用来建立多个符号变量，调用格式为：

```
syms 符号变量名 1 符号变量名 2 … 符号变量名 n          % 符号变量名不需要加字符串分界符(‘’)，各
                                                        % 符号变量名之间用空格分隔
```

【例 2-36】符号表达式的生成。

解：在命令行窗口中输入语句及输出结果如下：

```
>> y1='exp(x)'                    % 直接创建符号函数
y1 =
    'exp(x)'
>> equ='a*x^2+b*x+c=0'            % 直接创建符号方程
equ =
    'a*x^2+b*x+c=0'
>> syms x y                       % 建立符号变量 x、y
>> y2=sym(exp(x))                 % 使用 sym()函数生成符号表达式
y2 =
   exp(x)
>> y3=x^2+y^2                     % 生成符号表达式
y3 =
   x^2 + y^2
```

2.5.2　符号矩阵

符号矩阵也是一种特殊的符号表达式。MATLAB 中的符号矩阵也可以通过 sym()函数建立，矩阵的元素可以是任何不带等号的符号表达式，其调用格式为

```
符号矩阵名=sym('符号字符串矩阵')    % 符号字符串矩阵的各元素之间可用空格或逗号分隔
```

【例 2-37】符号矩阵的生成。

解：在命令行窗口中输入语句及输出结果如下：

```
>> syms aa bb a b ;
>> A=sym([aa,bb;1,a+2*b])
A =
    [ aa,    bb]
    [  1, a + 2*b]
>> A=sym([a,b,0,0; 1,a+2*b,1,2; 4,5,0,0])
A =
    [ a,     b, 0, 0]
    [ 1, a + 2*b, 1, 2]
    [ 4,     5, 0, 0]
```

从输出结果可以看出，与数值矩阵的输出形式不同，符号矩阵的每一行两端都有方括号。

在 MATLAB 中，数值矩阵不能直接参与符号运算，必须先转换为符号矩阵，同样也是通过 sym()函数转换的。

符号矩阵也是一种矩阵，因此之前介绍的矩阵的相关运算也适用于符号矩阵。很多应用于数值矩阵运算的函数，如 det()、inv()、rank()、eig()、diag()、triu()、tril()等，也能应用于符号矩阵。

【例 2-38】符号矩阵的运算。

解：在命令行窗口中输入语句及输出结果如下：

```
>> syms aa bb a b;
>> A=[aa,bb; 1,a+2*b]
A =
    [ aa,    bb]
    [  1, a + 2*b]
```

```
>> inv(A)                          % 符号矩阵的逆
ans =
    [ (a + 2*b)/(a*aa - bb + 2*aa*b),  -bb/(a*aa - bb + 2*aa*b)]
    [     -1/(a*aa - bb + 2*aa*b),  aa/(a*aa - bb + 2*aa*b)]
>> rank(A)                         % 符号矩阵的秩
ans =
   2
>> triu(A)                         % 符号矩阵的上三角阵
ans =
    [ aa,     bb]
    [  0,  a + 2*b]
>> tril(A)                         % 符号矩阵的下三角阵
ans =
    [ aa,      0]
    [  1,  a + 2*b]
```

2.5.3 常用符号运算

符号数学工具箱中提供了符号矩阵因式分解、展开、合并、简化和通分等符号运算函数，如表 2–12 所示。

表 2-12 常用符号运算函数

函 数 名	说 明	函 数 名	说 明
factor	符号矩阵因式分解	expand	符号矩阵展开
collect	符号矩阵合并同类项	simplify	应用函数规则对符号矩阵进行化简
simple	调用MATLAB其他函数对符号矩阵进行综合化简，并显示化简过程	numden	分式通分
compose	复合函数运算	finverse	反函数运算
limit	计算符号表达式极限	int	符号积分（定积分或不定积分）
diff	微分和差分函数	gradient	近似梯度函数
jacobian	计算多元函数的Jacobi矩阵		

由于微积分是大学教学、科研及工程应用中最重要的基础内容之一，这里只对符号微积分运算举例说明，读者可以通过查阅 MATLAB 的帮助文档学习其余的符号函数运算。

【例 2-39】符号微积分运算。

解：在命令行窗口输入语句及输出结果如下。

```
>> syms t x y                      % 定义符号变量
>> f1=sin(2*x);
>> df1=diff(f1)                    % 对函数 f1 中变量 x 求导
df1 =
   2*cos(2*x)
>> f2=x^2+y^2;
>> df2=diff(f2,x)                  % 对函数 f2 中变量 x 求偏导
df2 =
   2*x
>> f3=x*sin(x*t);
>> int1=int(f3,x)                  % 求函数 f3 的不定积分
int1 =
```

```
    (sin(t*x) - t*x*cos(t*x))/t^2
>> int2=int(f3,x,0,pi/2)                % 求 f3 在[0,pi/2]区间上的定积分
int2 =
    (sin((pi*t)/2) - (pi*t*cos((pi*t)/2))/2)/t^2
```

2.6　复数及其运算

复数运算根本上是对实数运算的拓展，在自动控制、电路学科等自然科学与工程技术中应用非常广泛。

2.6.1　复数和复矩阵的生成

复数有两种表示方式：一般形式和复指数形式。

一般形式为 $x=a+b\mathrm{i}$，其中 a 为实部，b 为虚部，i 为虚数单位。在 MATLAB 中，使用以下赋值语句即可生成复数 x（其中 a，b 为任意实数）。

```
>> syms a b
>> x=a+b*i
x =
   a + b*i
```

复指数形式为 $x=r\cdot \mathrm{e}^{\mathrm{i}\theta}$，其中 r 为复数的模，θ 为复数的幅角，i 为虚数单位。在 MATLAB 中，使用以下赋值语句即可生成复数 x。

```
>> syms r theta                      % r, theta 为任意实数
>> x=r*exp(theta*i)
x =
   r*exp(theta*i)
```

选取合适的表示方式能够便于复数运算，一般形式适合处理复数的代数运算，复指数形式适合处理复数旋转等涉及幅角改变的问题。

复数的生成有两种方法：一种是直接赋值，如上所述；另一种是通过符号函数 syms()构造，将复数的实部和虚部看作自变量，用 subs()函数对实部和虚部进行赋值。

【例 2-40】复数的生成。

解：在命令行窗口中输入语句及输出结果如下：

```
>> x1=-1+2i                          % 直接赋值
x1 =
  -1.0000 + 2.0000i
>> x2=sqrt(2)*exp(i*pi/4)
x2 =
   1.0000 + 1.0000i
>> syms a b real
>> x3=a+b*i                          % 构造符号函数
x3 =
   a + b*i
>> subs(x3,{a,b},{-1,2})             % 使用 subs()函数对实部和虚部赋值
ans =
    - 1 + 2*i
>> syms r theta real
>> x4=r*exp(theta*i);
>> subs(x4,{r,theta},{sqrt(20),pi/8})
```

```
ans =
    2*5^(1/2)*((2^(1/2) + 2)^(1/2)/2 + ((2 - 2^(1/2))^(1/2)*i)/2)
```

复数矩阵的生成也有两种方法：一种是直接输入复数元素生成；另一种是将实部和虚部矩阵分开建立，再写成和的形式，此时实部矩阵和虚部矩阵的维度必须相同。

【例 2-41】复数矩阵的生成。

解：在命令行窗口中输入语句及输出结果如下：

```
>> A=[-1+20i,-3+40i;1-20i,30-4i]                % 复数元素
A =
  -1.0000 +20.0000i  -3.0000 +40.0000i
   1.0000 -20.0000i  30.0000 - 4.0000i
>> B=real(A)                                    % 矩阵 A 的实部矩阵
B =
    -1    -3
     1    30
>> C=imag(A)                                    % 矩阵 A 的虚部矩阵
C =
    20    40
   -20    -4
>> D=B+C*i                                      % 由矩阵 A 的实部和虚部构造复向量矩阵
D =
  -1.0000 +20.0000i  -3.0000 +40.0000i
   1.0000 -20.0000i  30.0000 - 4.0000i
```

2.6.2 复数的运算

复数的基本运算与实数相同，都是使用相同的运算符或函数。此外，MATLAB 还提供了一些专门用于复数运算的函数，如表 2-13 所示。限于篇幅这里不再赘述具体使用方法。

表 2-13 复数运算的函数

函 数 名	说 明	函 数 名	说 明
abs	求复数或复数矩阵的模	angle	求复数或复数矩阵的幅角，单位为弧度
real	求复数或复数矩阵的实部	imag	求复数或复数矩阵的虚部
conj	求复数或复数矩阵的共轭	isreal	判断是否为实数
unwrap	去掉幅角突变	cplxpair	按复数共轭对排序元素

2.7 多项式求解初步

多项式 $p(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ 可以用 $\boldsymbol{p}=[a_0,a_1,a_2,\cdots,a_{n-1},a_n]$ 系数行向量表示。多项式行向量的创建方法有：

（1）多项式系数向量的直接输入法。

（2）利用 poly()函数产生多项式系数向量。

函数 poly()的调用格式如下：

```
p = poly(r)              % r 是向量，返回多项式的系数，其中多项式的根是 r 的元素
p = poly(A)              % A 是 n 阶方阵，返回矩阵 A 的特征多项式的 n+1 个系数
```

【例 2-42】求三阶方阵 A 的特征多项式。

解：在命令行窗口中输入语句及输出结果如下：

```
>> a=[11:13;14:16;20:22];
>> a1=poly(a)
a1 =
    1.0000  -48.0000  -27.0000    0.0000
>> a2=poly2str(a1,'s')
a2 =
    's^3 - 48 s^2 - 27 s + 5.0869e-14'
```

注意：①n 阶方阵的特征多项式系数向量一定是 n+1 维的；②特征多项式向量的第一个元素必是 1。

【例 2-43】给定根向量求解多项式系数向量。

解：在命令行窗口中输入语句及输出结果如下：

```
>> r=[-0,5,3,6+i];
>> r1=poly(r)
r1 =
   1.0000 + 0.0000i  -14.0000 - 1.0000i  63.0000 + 8.0000i  -90.0000 -15.0000i  0.0000
+ 0.0000i
>> r2=real(r1)
r2 =
     1   -14    63   -90     0
>> r3=poly2str(r2,'x')
r3 =
    'x^4 - 14 x^3 + 63 x^2 - 90 x'
```

由结果可知：

（1）要形成实数多项式，根向量中的复数必须共轭成对。

（2）可采用取实部的函数 real()把虚部值去掉。

（3）poly2str()是一个函数文件，它存在于 MATLAB 控制工具箱中。

常用的多项式运算函数有：

```
R=roots(p);              % 求多项式向量 p 的根
PS=polyval(P;S);         % 按照数组运算规则计算多项式值，其中 P 为多项式，S 为矩阵
PS=polyvalm(P;S);        % 按照数组运算规则计算多项式值，其中 P 为多项式，S 为矩阵
[r,p,k]=residue(b,a);    % 部分分式展开，其中 b、a 分别是分子、分母多项式系数向量；r、p、k 分别
                         % 为留数、极点、直项(多项式)向量
P=polyfit(x,y,n);        % 用 n 阶多项式拟合 x、y 向量给定的数据
```

【例 2-44】求多项式 $x^3-6x^2-72x-27$ 的根。

解：在命令行窗口中输入语句及输出结果如下：

```
>> r=roots([1,-6,-72,-27])
r =
   12.1229
   -5.7345
   -0.3884
```

说明：MATLAB 约定多项式系数用行向量表示，一组根用列向量表示。

【例 2-45】求解两种多项式求值指令的差别。

解：在命令行窗口中输入语句及输出结果如下：

```
>> s=pascal(4);
>> p=poly(s)
p =
    1.0000  -29.0000   72.0000  -29.0000    1.0000
>> p1=poly2str(p,'x')
p1 =
    'x^4 - 29 x^3 + 72 x^2 - 29 x + 1'
>> p2=polyval(p,s)
p2 =
   1.0e+04 *
    0.0016    0.0016    0.0016    0.0016
    0.0016    0.0015   -0.0140   -0.0563
    0.0016   -0.0140   -0.2549   -1.2089
    0.0016   -0.0563   -1.2089   -4.3779
>> p3=polyvalm(p,s)
p3 =
   1.0e-10 *
    -0.0003   -0.0036   -0.0052   -0.0143
    -0.0021   -0.0136   -0.0179   -0.0464
    -0.0059   -0.0330   -0.0400   -0.1047
    -0.0130   -0.0639   -0.0750   -0.1962
```

由输出结果可知，p3 中的元素都很小，它是运算误差造成的，理论上它应该是 0。这就是著名的 Caylay–Hamilton 定理：任何一个矩阵满足它本身的特征多项式。

【例 2-46】六阶多项式对区间[0, 2.5]上的误差函数 $y(x)=\frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-\mu^2}\mathrm{d}\mu$ 进行最小二乘拟合。

解：在命令行窗口中输入语句及输出结果如下：

```
>> x=0:0.1:2.5;
>> y=erf(x) ;                  % 计算误差函数在[0,2.5]的数据点
>> p=polyfit(x,y,6)
p =
    0.0084   -0.0983    0.4217   -0.7435    0.1471    1.1064    0.0004
>> p1=poly2str(p,'s')
p1 =
   ' 0.0084194 s^6 - 0.0983 s^5 + 0.42174 s^4 - 0.74346 s^3 + 0.1471 s^2
     + 1.1064 s + 0.00044117'
```

【例 2-47】用[0, 2.5]区间数据拟合曲线拟合[0, 5]区间的数据。

解：在命令行窗口中输入语句及输出结果如下：

```
>> x=0:0.1:5;
>> x1=0:0.1:2.5;
>> y=erf(x);
>> y1=erf(x1);
>> p1=polyfit(x1,y1,6)
p1 =
    0.0084   -0.0983    0.4217   -0.7435    0.1471    1.1064    0.0004
>> f=polyval(p1,x)
f =
  列 1 至 8
    0.0004    0.1119    0.2223    0.3287    0.4288    0.5209    0.6041    0.6778
  列 9 至 16
    0.7418    0.7965    0.8424    0.8800    0.9104    0.9342    0.9526    0.9664
```

```
  列 17 至 24
    0.9765    0.9838    0.9889    0.9925    0.9951    0.9969    0.9982    0.9991
  列 25 至 32
    0.9995    0.9994    0.9984    0.9964    0.9931    0.9882    0.9818    0.9737
  列 33 至 40
    0.9642    0.9539    0.9434    0.9341    0.9277    0.9267    0.9339    0.9533
  列 41 至 48
    0.9897    1.0488    1.1375    1.2640    1.4380    1.6706    1.9745    2.3646
  列 49 至 51
    2.8574    3.4718    4.2290
>> plot(x,y,'bo')
>> hold on
>> plot(x,f,'r--')
>> grid on
>> axis([0,5,0,2])
>> legend('拟合曲线','原始数据')
```

输出图形如图 2–2 所示。

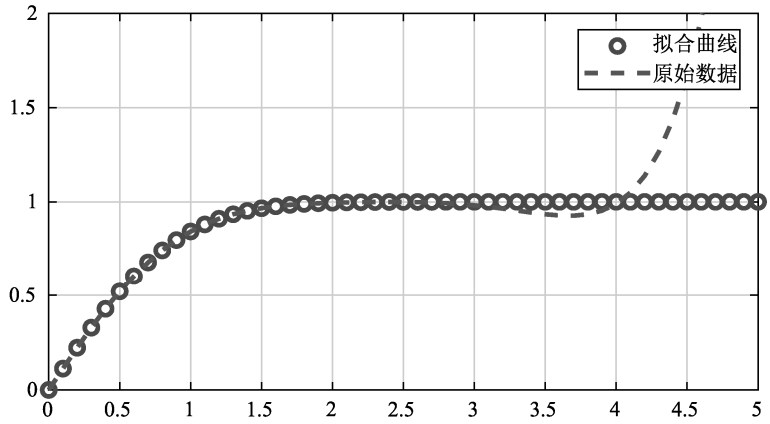

图 2–2　曲线拟合

2.8　本章小结

MATLAB 的科学计算包括多个方面，涵盖面极广，它对于矩阵表现出了极好的运算能力。本章首先介绍了 MATLAB 基本数据类型，之后主要围绕数组运算、矩阵运算、符号运算、复数运算等进行讲解。通过本章的学习，读者可以深入了解 MATLAB 的基本运算操作。结合本章的知识，读者完全可以独立解决基础数学问题。