

第 2 章 矩阵、数组、符号运算

在 MATLAB 中，矩阵是一方形的数组。一些特殊情形，比如只含有一个元素的矩阵称为标量，对于只含有一行或一列的矩阵称为矢量。MATLAB 为存储数或非数也提供了其他方式，但是最好在一开始就将所有数据定义为矩阵，因为这种操作在 MATLAB 中是最自然的。对于操作矩阵，MATLAB 赋予了我们能够同时操作一个矩阵所有元素的能力，不像其他程序语言一次只能对一个数（也就是一个矩阵元）进行操作。标量和矢量是特殊的矩阵，我们不仅可以通过操作矩阵来解决矩阵的问题，还可以解决所有和标量与矢量相关的问题。尤其是在操作含有多于一个元素的矩阵时，MATLAB 表现出远胜过其他编程软件的长处。

符号运算是本章介绍的另一重要内容。符号运算在数学中扮演着非常重要的角色，尤其在高等数学（微积分、高等代数等）中有较广泛的应用。在当前发达的社会，在处理许多事情时都引入了高等数学的方法，日常生活中的很多现象也满足高等数学基本定律。作为一门被广泛应用于科学与工程技术的计算机高级语言，引入符号计算是必需的。MATLAB 中符号计算是借助引入数学工具箱来完成的，数学工具箱出自于另一数学软件 MAPLE。MAPLE 拥有非常强的符号计算能力，自身创建并整合了大量有用而又有效的函数，可以对符号表达式进行各种数学操作。本章将对符号表达式的定义，以及一些常见基本操作函数进行阐述。

在 MATLAB 中，矩阵就是特殊的操作元素，在这章我们将首先介绍如何创建矩阵，然后介绍与矩阵相关的操作命令。由于数值计算的广泛应用，本章还将讲述数值计算的一些基本函数，数值计算方法将在第 5 章详细讲述。随着计算机的发展进步，数据处理软件也得到迅速发展。尽管 MATLAB 软件也提供了创建数据库的一些基本函数，但是限于篇幅本书不讲述这方面内容。另外，本章将对符号表达式的定义，求解等基本函数进行阐述，尤其是微积分和积分变换方面函数进行较详细讲解并给出示例。对符号方程、常微分方程求解函数也进行较详细阐述。

本章中所有例题求解中，开头“...”符号前为题目内容陈述，“%”后为注释，“>>”后为执行行语句，一般“=”后为输出结果。执行行命令应该在 Command Window 内输入运行，相对应本书附带的程序是以 M 文件模式存在的，没有“>>”符号，也没有结果显示行，这些需要注意。

2.1 创建矩阵方法

在学习数学过程中，整数是我们的第一位朋友，然后一步一步逐渐认识了各种各样的数。对各种各样数的理解和认识，与我们对整数的理解和认识是分不开的。矩阵在 MATLAB 语言相当于整数在数学中的地位，所以学习创建矩阵是学习 MATLAB 编程的基石。创建矩阵，有多种方法，下面将分别介绍。

2.1.1 直接输入法

对简单矩阵来讲，直接输入法创建矩阵是一种最直接、简单、有效的办法。并且基于已有矩阵，我们还可以通过直接输入法定义一些特定元素生成较大矩阵。

【例 2.1】 通过直接输入法，创建矩阵 $\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$ ，并通过定义特定元素，生成 3×5 矩阵如下：

$$A = \begin{bmatrix} 1 & 3 & 0 & 0 & 0 \\ 5 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

并将矩阵 A 的值赋予给另一个矩阵 B 。

EXAMP02001

创建矩阵 $A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$ ，我们可以通过在命令窗口一个接一个地输入所有矩阵元素来实现。

```
% By typing, one enters a matrix in Command Window. Need to note that
% ", " is used to separate elements in same row, and ";" is used to
% distinguish different rows.
% (通过键盘在命令窗口输入矩阵。需要注意：借助", "隔离同行内元素，借助"; "
% 分离矩阵的行)
>> A=[1,3;5,7]
A =
     1     3
     5     7
% Use "space" key to separate different elements in same row, and
% use "enter" key to distinguish different rows.
% (通过"space"键区分同行的不同元素，通过"enter"键去区分不同的行)
>> A=[1 3
5 7]
A =
     1     3
     5     7
% In terms of a matrix A, by defining a specific element to generate
% a 3-by-5 matrix as B.
% (基于矩阵A，通过定义特定元素产生 3×5 矩阵即为 B)
>> A(3,5)=0
A =
     1     3     0     0     0
     5     7     0     0     0
     0     0     0     0     0
>> B=A
B =
     1     3     0     0     0
     5     7     0     0     0
     0     0     0     0     0
```

【例 2.2】 创建 Dürer 矩阵，又称魔术阵，如下：

$$A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$$

EXAMP02002

```
% Type all elements of matrix A directly in Command window.
% (在命令窗口直接输入矩阵 A 的所有元素)
>>A=[16,2,3,13;5,11,10,8;9,7,6,12;4,14,15,1]
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

% Create a 4*4 magic matrix by "magic".
% (借助函数"magic"创建一个 4*4 魔术阵)
>> B=magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

2.1.2 矩阵生成命令

1. 全零矩阵

全零矩阵函数描述参见表 2.1。

表 2.1 zeros 函数

语 法	描 述	语 法	描 述
zeros	生成 1 个 0 标量	zeros(m, n, p, ...)	生成 1 个 $m \times n \times p \times \dots$ 的全零数组
zeros(n)	生成 1 个 $n \times n$ 的全零矩阵	zeros(size(A))	生成 1 个和矩阵 A 同尺寸的全零矩阵
zeros(m, n)	生成 1 个 $m \times n$ 的全零矩阵		

【例 2.3】 创建几个全零矩阵。

EXAMP02003

创建几个全零矩阵。

```
% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)
>> clear
% Create a 3-by-3 matrix containing random values drawn
% from a uniform distribution on the unit interval by
```

```
% "rand".
% (借助函数"rand"创建一个3×3的任意矩阵)
>> A=rand(3)
A =
    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
% Create a scalar 0 by "zeros".
% (通过函数"zeros"创建一个零标量)
>> zero1=zeros
zero1 =
    0
% Create a 2-by-2 matrix of zeros.
% (创建一个2×2的全零矩阵)
>> zero2=zeros(2)
zero2 =
    0    0
    0    0
% Create a 2-by-3 matrix of zeros.
% (创建一个2×3的全零矩阵)
>> zero23=zeros(2,3)
zero23 =
    0    0    0
    0    0    0
% Create a 3-by-5-by-2 array of zeros.
% (创建一个3×5×2层的全零数组)
>> zero352=zeros(3,5,2)
zero352(:,:,1) =
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
zero352(:,:,2) =
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
% Create a same size as A of zeros.
% (创建一个和矩阵A同行数同列数的全零矩阵)
>> zeroA=zeros(size(A))
zeroA =
    0    0    0
    0    0    0
    0    0    0
```

好习惯

在编写程序时，通常开始我们就将所有矩阵定义为全零矩阵，因为这样将会使得所编写程序运行节省一些时间。

2. 全1矩阵

全1矩阵函数描述参见表2.2。

表 2.2 ones 函数

语 法	描 述	语 法	描 述
ones	生成 1 个标量 1	ones(m, n, p, ...)	生成 1 个 $m \times n \times p \times \dots$ 的全 1 数组
ones(n)	生成 1 个 $n \times n$ 的全 1 矩阵	ones(size(A))	生成 1 个和矩阵 A 同大小的全 1 矩阵
ones(m, n)	生成 1 个 $m \times n$ 的全 1 矩阵		

【例 2.4】 创建几个全 1 矩阵。

EXAMP02004

创建几个全 1 矩阵。

```

% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)
>> clear
% Create a 3-by-3 matrix containing pseudo-random values
% drawn from a normal distribution with mean zero and
% standard deviation one by the function "randn".
% (借助函数"randn"创建 3×3 的任意矩阵)
>> A=randn(3)
A =
    2.7694    0.7254   -0.2050
   -1.3499   -0.0631   -0.1241
    3.0349    0.7147    1.4897
% Create a scalar 1 by "ones".
% (借助函数"ones"创建一个标量)
>> one1=ones
one1 = 1
% Create a 5-by-5 matrix of ones by "ones".
% (借助函数"ones"创建一个 5×5 列全 1 矩阵)
>> one5=ones(5)
one5 =
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
% Create a 2-by-7 matrix of ones by "ones".
% (借助函数"ones"创建 2 行 7 列的全 1 矩阵)
>> one27=ones(2,7)
one27 =
     1     1     1     1     1     1     1
     1     1     1     1     1     1     1
% Create a 3-by-6-by-3 array of ones by "ones".
% (借助函数"ones"创建 3 行 6 列 3 层的全 1 数组)
>> one363=ones(3,6,3)
one363(:,:,1) =

```

```

1     1     1     1     1     1
1     1     1     1     1     1
1     1     1     1     1     1
one363(:, :, 2) =
1     1     1     1     1     1
1     1     1     1     1     1
1     1     1     1     1     1
one363(:, :, 3) =
1     1     1     1     1     1
1     1     1     1     1     1
1     1     1     1     1     1
% Create a matrix of ones as the same size of A.
% (创建与矩阵 A 同行数同列数的全 1 矩阵)
>> oneA=ones(size(A))
oneA =
1     1     1
1     1     1
1     1     1

```

3. 单位矩阵

单位矩阵函数描述参见表 2.3。

表 2.3 eye 函数

语 法	描 述	语 法	描 述
eye	生成 1 个标量 1	eye(m, n)	生成 1 个 $m \times n$ 的单位矩阵
eye(n)	生成 1 个 $n \times n$ 的单位矩阵	eye(size(A))	生成 1 个和矩阵 A 一样大小的单位矩阵

【例 2.5】 创建几个单位矩阵。

EXAMP02005

创建几个单位矩阵。

```

% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)
>> clear
% Create a 3-by-3 matrix containing pseudo-random values drawn from
% a normal distribution with mean zero and standard deviation one.
% (创建 3×3 的任意矩阵, 矩阵元素符合正态分布)
>> A=randn(3)
A =
-0.4326    0.2877    1.1892
-1.6656   -1.1465   -0.0376
 0.1253    1.1909    0.3273
% Create a unit scalar by "eye".
% (借助"eye"函数创建单位标量)
>> eye1=eye

```

```

eye1 = 1
% Create a 2-by-2 identity matrix by "eye".
% (借助"eye"函数创建一个 2×2 的单位矩阵)
>> eye2=eye(2)
eye2 =
    1    0
    0    1
% Create a 3-by-5 identity matrix by "eye".
% (借助"eye"函数创建 3×5 的对角单位矩阵)
>> eye35=eye(3,5)
eye35 =
    1    0    0    0    0
    0    1    0    0    0
    0    0    1    0    0
% Create an identity matrix as the same size of A.
% (创建与矩阵 A 同规格的单位矩阵)
>> eyeA=eye(size(A))
eyeA =
    1    0    0
    0    1    0
    0    0    1

```

4. 随机矩阵生成函数

MATLAB 中 rand 函数是产生 0 到 1 的随机数，函数具体用法参见表 2.4。

表 2.4 rand 函数

语 法	描 述
rand	生成 1 个随机数标量
rand(n)	生成 1 个 $n \times n$ 随机数矩阵
rand(m, n)	生成 1 个 $m \times n$ 随机数矩阵
rand(m, n, p)	生成 1 个 $m \times n \times p$ 随机数数组
rand(size(A))	生成 1 个和矩阵 A 同尺寸的随机数矩阵
rand(method, s)	通过“method”来完成生成随机数的取值区间，“method”有以下三种选择，分别为“state”、“seed”和“twister”。而 s 是标量整数，取值范围为从 0 到 $2^{32}-1$ ，“state”对应取值范围为 $[2^{(-53)}, 1-2^{(-53)}]$ ，执行 2^{1492} 次才有可能重复一次；“seed”对应取值范围为 $[1/(2^{31}-1), 1-1/(2^{31}-1)]$ ，周期为 $2^{31}-2$ ；“twister”对应取值范围 $[2^{(-53)}, 1-2^{(-53)}]$ ，周期为 $(2^{19937}-1)/2$

【例 2.6】 创建几个随机矩阵。

EXAMP02006

通过 rand 函数创建几个随机矩阵。

```

% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)

```

```
>> clear
% Create a double precision pseudo-random number.
% (产生一个随机数标量)
>> rand
ans =
    0.2311
% Create a 3-by-3 double precision pseudo-random matrix.
% (创建一个3×3的随机数矩阵)
>> rand(3)
ans =
    0.6068    0.7621    0.8214
    0.4860    0.4565    0.4447
    0.8913    0.0185    0.6154
% Create a 3-by-5 double precision pseudo-random matrix.
% (创建一个3×5的随机数矩阵)
>> rand(3,5)
ans =
    0.7919    0.1763    0.9169    0.0579    0.0099
    0.9218    0.4057    0.4103    0.3529    0.1389
    0.7382    0.9355    0.8936    0.8132    0.2028
% Create a 3-by-5-by-2 double precision pseudo-random array.
% (创建一个3×5×2的随机数数组)
>> rand(3,5,2)
ans(:,:,1) =
    0.1987    0.1988    0.4451    0.4186    0.2026
    0.6038    0.0153    0.9318    0.8462    0.6721
    0.2722    0.7468    0.4660    0.5252    0.8381
ans(:,:,2) =
    0.0196    0.8318    0.4289    0.1934    0.5417
    0.6813    0.5028    0.3046    0.6822    0.1509
    0.3795    0.7095    0.1897    0.3028    0.6979
% Create a 3-by-3 magical matrix.
% (创建一个3×3的魔术矩阵)
>> A=magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
% Create a matrix with double precision pseudo-random numbers as
% the same size of A.
% (创建一个与矩阵A同尺寸的随机矩阵)
>> rand(size(A))
ans =
    0.3784    0.5936    0.8216
    0.8600    0.4966    0.6449
    0.8537    0.8998    0.8180
% Generate uniform values from the interval [5, 99] to construct
% a 5-by-5 matrix.
% (创建一个5×5的随机矩阵, 矩阵元素取值区间为[5, 99])
>> B=5+(99-5)*rand(5)
B =
```



```

33.1993  95.9007  91.1195  92.2779  65.3216
96.5908  84.4530  97.7124  14.3555  40.0986
27.7735   8.3009  77.9867  74.8394  57.2399
84.6939  23.3938  85.9498  17.0268  41.4749
14.6084  77.0194  65.6323  40.1215  65.1922
% Generate an integer array with 50 elements ranging from 1 to 100.
% (产生一个含 50 个元素的整数数组, 元素随机分布在 1 到 100 之间)
>> C=ceil(100*rand(1,50))
C =
Columns 1 through 14
8 31 68 93 5 43 54 59 41 57 97 50 61 39
Columns 15 through 28
73 4 52 99 9 96 69 44 50 89 35 12 94 81
Columns 29 through 42
45 18 58 81 71 1 97 80 52 7 60 41 92 45
Columns 43 through 50
83 47 65 82 95 69 32 78

```

注意:

创建随机矩阵时, 输入的 m 、 n 、和 p 应该是非负整数, 如果输入的是负整数, 将按 0 处理。

5. 正态分布的随机矩阵生成函数

MATLAB 提供 `randn` 函数用于产生正态分布的随机矩阵, 其具体用法参见表 2.5。

表 2.5 `randn` 函数

语 法	描 述
<code>randn</code>	生成正态分布的随机数标量
<code>randn(n)</code>	生成 1 个 $n \times n$ 正态分布的随机矩阵
<code>randn(m, n)</code>	生成 1 个 $m \times n$ 正态分布的随机矩阵
<code>randn(m, n, p)</code>	生成 1 个 $m \times n \times p$ 正态分布的随机数组
<code>randn(size(A))</code>	生成 1 个与矩阵 A 同尺寸的正态分布随机矩阵

【例 2.7】 创建几个正态分布随机矩阵。

EXAMP02007

通过 `randn` 函数创建几个正态分布随机矩阵。

```

% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)
>> clear
% Generate a double precision random number.
% (产生一双精度随机数)
>> randn

```

```
ans =
    0.7258
% Generate a 3-by-3 double precision random matrix.
% (创建一个3×3正态分布的随机矩阵)
>> randn(3)
ans =
   -0.5883    0.1139   -0.0956
    2.1832    1.0668   -0.8323
   -0.1364    0.0593    0.2944
% Generate a 2-by-3 double precision random matrix.
% (创建一个2×3正态分布的随机矩阵)
>> randn(2,3)
ans =
   -1.3362    1.6236    0.8580
    0.7143   -0.6918    1.2540
% Generate a 2-by-5-by-2 double precision random array.
% (创建一个2×5×2的正态分布随机数组)
>> randn(2,5,2)
ans(:,:,1) =
   -1.5937    0.5711    0.6900    0.7119    0.6686
   -1.4410   -0.3999    0.8156    1.2902    1.1908
ans(:,:,2) =
   -1.2025   -0.1567    0.2573    1.4151    0.5287
   -0.0198   -1.6041   -1.0565   -0.8051    0.2193
% Produce a 3-by-3 Hilbert matrix.
% (生成一个3×3希尔伯特矩阵)
>> A=hilb(3)
A =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
% Generate a double precision random matrix as
% the same size of A matrix.
% (生成一个与矩阵A同尺寸的正态分布随机矩阵)
>> randn(size(A))
ans =
   -0.9219   -1.0106    1.6924
   -2.1707    0.6145    0.5913
   -0.0592    0.5077   -0.6436
% Generate a double precision random matrix with mean 1 and
% standard deviation 2.
% (生成一个正态分布含有25个元素的行数组, 该数组平均值为1, 标准差为2)
>> 1+2*randn(1,25)
ans =
Columns 1 through 8
    1.7607   -1.0182    0.9610    0.9036    1.0001    0.3643    3.1900   -2.7480
Columns 9 through 16
    1.8564    2.7913    2.4619    2.1557    1.0806    2.3542    2.1378    0.4887
Columns 17 through 24
    0.2451    0.4082   -1.9503    0.5320    1.2369    1.6296    3.8870    0.2981
```

Column 25
2.2465

注意

该函数 `randn` 中输入的参数 m 、 n 、和 p 应该为非负整数, 如果输入的是负整数, 将按 0 处理。

如果读者对创建其他矩阵非常感兴趣, 可以借助 MATLAB 中“help”命令对如: “vander”、“hilb”、“invhilb”、“magic”和“compan”等函数进行学习。

【例 2.8】 创建几个特殊矩阵。

EXAMP02008

创建几个特殊矩阵。

```
% Generate a Vandermonde matrix following vector V=[1 2 3 4 5].
% (创建一个 5×5 范德蒙特矩阵)
>> vander([1 2 3 4 5])
ans =
     1     1     1     1     1
    16     8     4     2     1
    81    27     9     3     1
   256    64    16     4     1
   625   125    25     5     1
% Generate a 3-by-3 Hilbert matrix.
% (生成一个 3×3 希尔伯特矩阵)
>> hilb(3)
ans =
    1.0000    0.5000    0.3333
    0.5000    0.3333    0.2500
    0.3333    0.2500    0.2000
% Generate a 3-by-3 inverse Hilbert matrix.
% (生成一个 3×3 逆希尔伯特矩阵)
>> invhilb(3)
ans =
     9    -36     30
    -36    192   -180
     30   -180    180
% Generate a 5-by-5 magic matrix.
% (生成一个 5×5 魔术矩阵)
>> magic(5)
ans =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
% Generate a company matrix of the polynomial x^5+ 2*x^4+ 3*x^3+
% 4*x^2+ 5*x+6.
% (生成多项式 x^5+2*x^4+3*x^3+4*x^2+5*x+6 的伴随矩阵)
```

```
>> compan([1 2 3 4 5 6])
ans =
    -2    -3    -4    -5    -6
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
```

2.2 构建数组方法

2.2.1 数组生成命令

当矩阵离开线性代数就变成了二维数字数组。对矩阵进行四则运算是将整个矩阵看成一个操作对象，而数组操作是对单个元素进行的。仅仅从外表看，数组与矩阵具有完全相同的外在形式，所以我们可以借助所有生成矩阵的命令来创建数组。矩阵和数组有一个明显的不同之处是，矩阵是二维的，数组可以超过二维，比如在“EXAMP02005”和“EXAMP02006”中创建的所谓数组“Array”。在此我们将不再对这一内容进行深入讨论。如果读者对此仍有不解之处或仍有疑问，请参照 MATLAB 软件中的帮助命令“help”进行深入学习。

2.2.2 矢量生成命令

1. 冒号“:”符号

冒号符号在 MATLAB 中是最有用的符号之一，它不仅能用来引用、添加、删除数组的元素，而且可以用它来创建矢量数组。

【例 2.9】 通过冒号“:”创建几个数组。

EXAMP02009

借助冒号“:”符号创建几个数组。

```
% Generate a vector A=[1 2 3 4 5 6 7 8 9 10] by colon operator.
% (借助冒号创建一个矢量 A=[1 2 3 4 5 6 7 8 9 10])
>> A=1:10
A = 1     2     3     4     5     6     7     8     9    10
% Generate a vector ranging from 1 to 20 with step-length 2.
% (生成一个矢量，元素从 1 到 20，步长为 2)
>> B=1:2:20
B = 1     3     5     7     9    11    13    15    17    19
% Generate a vector ranging from 100 to 20 with step-length 9.
% (生成一个矢量，元素从 100 到 20，步长为-9)。
>> C=100:-9:20
C = 100    91    82    73    64    55    46    37    28
```

2. linspace 函数

`linspace` 函数是用来生成等差数列的命令，其语法与相应描述参见表 2.6。

表 2.6 linspace 函数

语 法	描 述
<code>linspace(x1, x2)</code>	在从 x_1 到 x_2 区间内创建含有 100 个元素的等差数列
<code>linspace(x1, x2, n)</code>	在从 x_1 到 x_2 区间内创建含有 n 个元素的等差数列

【例 2.10】 通过 `linspace` 函数创建几个等差数列。

EXAMP02010

通过 `linspace` 函数创建几个等差数列。

```
% Generate a vector of 100 linearly equally spaced points
% in the region [5,9].
% (生成一个含 100 个元素的矢量, 元素为 5 到 9 区间的等差数列)
>> linspace(5,9)
ans =
Columns 1 through 8 .....
5.0000 5.0404 5.0808 5.1212 5.1616 5.2020 5.2424 5.2828 .....
Columns 89 through 96
8.5556 8.5960 8.6364 8.6768 8.7172 8.7576 8.7980 8.8384
Columns 97 through 100
8.8788 8.9192 8.9596 9.0000
% Generate a vector of 9 linearly equally spaced points
% in closed interval [11,99].
% (生成一个含 9 个元素矢量, 元素为 11 到 99 区间的等差数列)
>> linspace(11,99,9)
ans = 11    22    33    44    55    66    77    88    99
```

3. logspace 函数

`logspace` 函数用来创建指数等差数列, 其语法与描述参见表 2.7。

表 2.7 logspace 函数

语 法	描 述
<code>logspace(x1, x2)</code>	在从 10^{x_1} 到 10^{x_2} 区间内生成 50 个指数为等差数列的元素
<code>logspace(x1, x2, n)</code>	在从 10^{x_1} 到 10^{x_2} 区间内生成 n 个指数为等差数列的元素

【例 2.11】 通过函数 `logspace` 创建几个数组。

EXAMP02011

通过函数 `logspace` 创建几个数组。

```
% Set all numbers with long format.
% (设定所有数为 long 格式)
>> format long
% Generate a vector of 50 logarithmically equally spaced points
% between  $10^2$  and  $10^9$ .
```

```

% (在 10^2 到 10^9 区间生成一个含 50 个元素的对数等差数组)
>> logspace(2,9)
ans =
    1.0e+09 *
    Columns 1 through 3.....
    0.000000100000000    0.000000138949549    0.000000193069773.....
    Columns 48 through 50
    0.517947467923122    0.719685673001153    1.000000000000000
% Generate a vector of 8 logarithmically equally spaced points
% between 10 and 10^9.
% (在 10 到 10^9 区间, 生成含 8 个元素的对数等差数组)
>> logspace(1,9,8)
ans =
    1.0e+009 *
    Columns 1 through 4
    0.00000001000000,0.00000013894955,0.00000193069773,0.00002682695795
    Columns 5 through 8
    0.00037275937203,0.00517947467923,0.07196856730012,1.00000000000000
% Generate a vector of 4 logarithmically equally spaced points
% between 10^0.01 and pi.
% (在 10^0.01 到 pi 区间, 生成含 4 个元素的对数等差数组)
>> logspace(0.01,pi,4)
ans =
    1.02329299228075, 1.48724764838776, 2.16155644993225, 3.14159265358979
% Reset the format of number as short.
% (重新将输入数设置为 short 格式)
>> format short

```

4. 基于已有矩阵或数组创建数组

在某些情况下, 已经定义一些矩阵或数组。如果待创建数组的所有元素或部分元素在已有数组或矩阵已经存在, 我们可以借助这些已有矩阵或数组来创建需要的新数组。下面提供了一些示例。

【例 2.12】 基于已有矩阵或数组构建新数组。

EXAMP02012

基于已有矩阵或数组构建新数组。

```

% Clear Command Window.
% (清除命令窗口)
>>clc
% Clear variables in workspace.
% (清除工作空间中的变量)
>> clear
% Generate a 5-by-5 Hilbert matrix.
% (生成一个 5×5 希尔伯特矩阵)
>> A=hilb(5)
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
% Define an array (or vector) B with all elements of
% the predefined matrix A.
% (通过已定义矩阵 A 的元素, 定义一个新数组或矢量, 记为 B)
>> B=A(:)
B =
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
% Define an array (or vector) C with elements in the predefined
% matrix A from the first to the last with step-length 4.
% (借助已定义矩阵 A, 定义一个新数组或矢量, 记为 C)
>> C=A([1:4:25])
C =
1.0000    0.2000    0.2000    0.2000    0.2000    0.2000    0.1111
% Define an array (or vector) D with all elements in the first
% column of predefined matrix A.
% (借助已定义矩阵 A, 定义一个新数组或矢量, 记为 D, 只含有矩阵第一列)
>> D=A(:,1)
D =
1.0000
0.5000
0.3333
0.2500
0.2000
```

```

% Define an array (or matrix) E with all elements in the first
% and forth columns in the predefined matrix A.
% (借助已定义矩阵 A, 定义一个新数组或矢量, 记为 E, 只含有矩阵的第一列与第二列)
>> E=A(:, [1:3:5])
E =
    1.0000    0.2500
    0.5000    0.2000
    0.3333    0.1667
    0.2500    0.1429
    0.2000    0.1250
% Define an array (or matrix) F with some special elements in
% the matrix A.
% (基于已定义矩阵 A, 定义一个新数组或矩阵, 记为 F, 只含有部分特定元素)
>> F=A([1,3,5],[2,4])
F =
    0.5000    0.2500
    0.2500    0.1667
    0.1667    0.1250
% Defined an array (or scalar) G with the special element in
% the matrix A.
% (选定已定义矩阵 A 中特定元素, 记为一个新数组或标量, 记为 G)
>> G=A(1,5)
G = 0.2000
% Create a matrix H with all elements of C and 99.
% (创建一个矩阵 H, 含有矩阵 C 和 99)
>>H=[C,99]
H=
    1.0000    0.2000    0.2000    0.2000    0.2000    0.2000    0.1111    99

```

2.3 矩阵数组的运算操作

本节将对矩阵、数组数学运算进行介绍, 有初等的四则运算, 也有相对复杂的高等运算。为了根据相应情况进行相应运算, 本节将对逻辑运算符号和关系运算符号进行阐述。在对运算函数与符号进行阐述前, 首先介绍一些常数变量参见表 2.8。

表 2.8 常数变量

变 量	描 述	变 量	描 述
pi	圆周率 π 的近似值	date	用来显示日期, 格式为日-月-年 (例如 17-sep-15)
realmin	最小正实数	eps	机器零阈值
realmax	最大正实数	ans	计算结果的默认数值变量
bitmax	最大正整数	nargin	函数输入参数个数
i, j	虚数单位	nargout	函数输出参数个数
inf, Inf	无穷大	lasterr	存放最新的错误信息
nan, NaN	非数	lastwarn	存放最新的警告信息
clock	一般显示 6 个元素, 分别为年、月、日、时、分、秒		

常用符号描述参见表 2.9。

表 2.9 常用符号

符 号	描 述
{ }	单元素数组构建符号
[]	数组或矩阵构建符号
()	标注角标参数
' '	标注字符串
,	用来区分下标或矩阵元素
:	(1) 可以压缩命令窗口的运算结果，让其不显示； (2) 断开矩阵行； (3) 断开 MATLAB 中的执行语句，表示上一指令到此为止
%	标注解内容
:	用于创建等差数列
+	加法
-	减法
.*	点乘，数组乘法
*	乘法
./	数组右除
.\	数组左除
/	右除
\	左除
.^	数组乘方
^	数乘方，矩阵乘方
'	矩阵和数组转置，复数取共轭
...	悬挂，代表上一行语句没有输入完

常见的操作命令描述参见表 2.10。

表 2.10 常见的操作命令

变 量	描 述	变 量	描 述
cd	显示或改变工作目录	pack	整理内存碎片
clc	清空命令窗口	path	显示搜索目录
clear	清空工作空间的变量	quit	退出 MATLAB
clf	清空图形窗口	save	保存内存变量
diary	日志文件命名	type	显示文件内容
dir	显示当前目录下的文件	what	列出所在目录 MATLAB 文件
disp	显示变量或文字的内容	which	定位 MATLAB 文件的路径
echo	命令窗口信息显示开关	who	列出工作空间的变量
hold	图形保持命令	whos	详细列出工作空间的变量
load	加载制订文件中的变量		

输出数据显示格式命令描述参见表 2.11。

表 2.11 输出数据显示格式命令

变 量	描 述
format short	小数点后包括 4 位有效数字，最多不超过 7 位有效数字；如果数值大于 1000，那么按照科学计数法表示
format long	用 15 位数字表示
format short e	5 位科学计数法表示
format long e	15 位科学计数法表示
format short g	从 format short 和 format short e 中自动选择最佳的数值表示方法
format long g	从 format long 和 format long e 中自动选择最佳的数值表示方法
format rat	采用近似有理数来表示
format hex	十六进制表示
format +	用+、-和空格来分别表示正数、负数和零，复数中的虚部不表示
format bank	金融表示方法，元、角、分等
format compact	显示变量之间没有空格
format loose	显示变量之间有空格

注意：

在 MATLAB 函数语法中，只有小括号，不能使用中括号和大括号，并且函数一般都是小写，不能出现大写字母，只有在函数所带的参数中有时有大写字母出现，比如后面绘图中将要讲述 plot 命令可带的参数“LineWidth”等。

所有带点的运算命令一般只能作用到数组，不能作用到矩阵，如果作用到矩阵时，已经把矩阵当作二维数组来处理了。带点的运算是针对矩阵或数组里的元素单个操作的，不能把矩阵看成一个整体进行运算操作。

2.3.1 四则运算

1. 矩阵四则运算

众所周知，标量、矢量和复数都可以进行四则运算。为简单起见，我们将对矩阵四则运算进行简单介绍，参见表 2.12。

表 2.12 矩阵四则运算符号

+	加	\	左除
-	减	^	乘方
*	乘	'	转置
/	右除	()	优先操作运算顺序

如果读者对矩阵四则运算不清楚，请阅读线性代数和矩阵相关参考书。

【例 2.13】 矩阵四则运算示例。

EXAMP02013

矩阵四则运算实例。