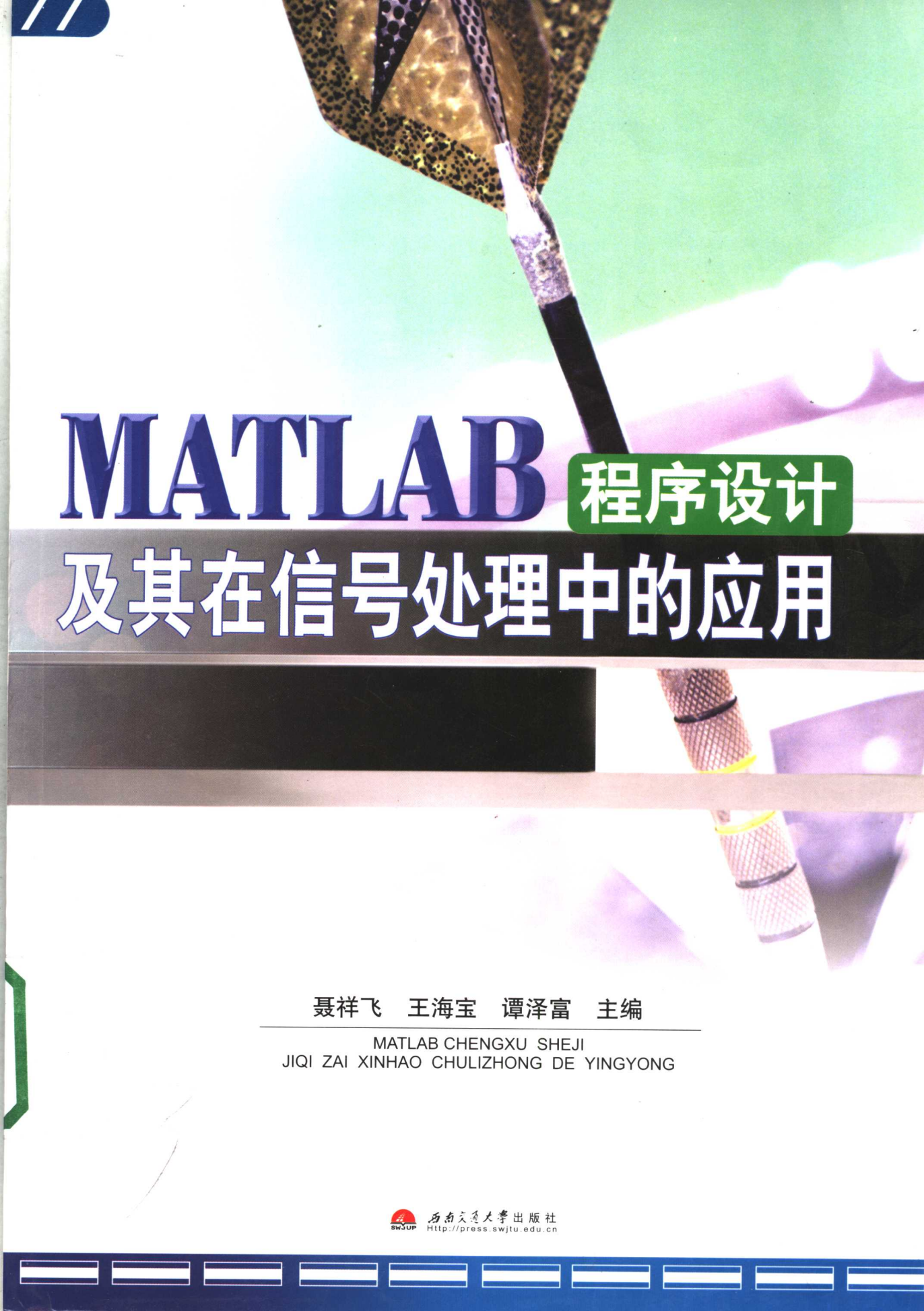




MATLAB 程序设计

及其在信号处理中的应用

聂祥飞 王海宝 谭泽富 主编

MATLAB CHENGXU SHEJI
JIQI ZAI XINHAO CHULIZHONG DE YINGYONG



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)

MATLAB 程序设计 及其在信号处理中的应用

聂祥飞 王海宝 谭泽富 主编

西南交通大学出版社

· 成 都 ·

内 容 简 介

MATLAB 是一种功能强大的数据分析和工程计算语言, 在工业、电子、信号处理、通信工程、医学、建筑及航空等领域有着广泛的应用。本书以目前较为流行的 MATLAB6.5 和 MATLAB7.0 版本为对象, 对 MATLAB 的程序设计方法及其在信号处理中的应用做了层次清晰、浅显易懂的介绍。

本书内容可以分为基础和应用两个部分。第 1~6 章主要介绍 MATLAB 的入门基础和编程的基本技巧, 第七章介绍 MATLAB 在信号处理中的具体应用。

本书的主要读者为电子信息类专业的科技工作者和理工科大学相关专业的本科生、研究生; 对其他领域的科研人员也有一定的参考作用。

图书在版编目 (C I P) 数据

MATLAB 程序设计及其在信号处理中的应用 / 聂祥飞, 王海宝, 谭泽富主编. —成都: 西南交通大学出版社, 2005.4

ISBN 7-81104-066-2

I. M... II. ①聂... ②王... ③谭... III. 计算机辅助计算 - 软件包, MATLAB IV. TP391.75

中国版本图书馆 CIP 数据核字 (2005) 第 039177 号

MATLAB 程序设计及其在信号处理中的应用

聂祥飞 王海宝 谭泽富 主编

*

责任编辑 刘永淑 唐晴

封面设计 何东琳设计工作室

西南交通大学出版社出版发行

(成都二环路北一段 111 号 邮政编码: 610031 发行部电话: 028-87600564)

<http://press.swjtu.edu.cn>

E-mail: cbsxx@swjtu.edu.cn

四川森林印务有限责任公司印刷

*

开本: 787 mm × 1 092 mm 1/16 印张: 13.125

字数: 325 千字

2005 年 4 月第 1 版 2005 年 4 月第 1 次印刷

ISBN 7-81104-066-2/TP · 010

定价: 22.00 元

图书如有印装问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

MATLAB 是 MathWorks 公司于 1982 年推出的一种功能强大、效率高、交互性好的高级程序语言。MATLAB 将数值分析、矩阵运算、信号处理和图形显示有机地融合为一体，形成了一个方便、界面友好的操作环境。随着版本的不断提高，MATLAB 的功能也越来越强大，应用范围也越来越广，如今已经广泛应用于信号处理、图像处理、系统仿真、自动控制、小波分析和神经网络等领域。

本书主要介绍 MATLAB 程序设计方法及其在信号处理中的具体应用。随着信息通信技术的飞速发展，信号处理已经逐渐发展成为一门独立的学科并成为信息科学的重要组成部分，在语音处理、图像处理、雷达、航空航天、地质勘探、通信、生物医学工程等领域得到了广泛的应用。然而，由于信号处理的数学推导复杂、理论抽象，往往令初学者望而生畏。本书将 MATLAB 的程序设计和信号处理有机地结合起来，给出了详细的函数使用说明，再通过具体的实例，加深了读者对函数应用方法和技巧的认识，使读者在学习信号处理的同时，能够轻松地利用 MATLAB 对信号处理中基本理论和基本方法进行仿真实验。

本书结构合理，层次清晰，说理透彻，浅显易懂。相信读者通过本书的学习，一定能够有所收获。

本书在编写过程中，得到了胡继明博士的热心帮助与鼓励，在此表示衷心的感谢！

由于作者学识水平有限，再加上 MATLAB 和信号处理学问的博大精深，本书的疏漏之处在所难免，望广大读者批评指正。

编 者

2004 年 10 月

目 录

第 1 章 MATLAB 简介	1
1.1 运行界面	1
1.2 矩阵操作	4
1.2.1 矩阵的输入	4
1.2.2 矩阵的索引	6
1.2.3 矩阵的基本运算	8
1.3 建立 M 文件	14
1.3.1 脚本 M 文件	15
1.3.2 函数 M 文件	15
第 2 章 图形的绘制	18
2.1 二维图形的绘制	18
2.1.1 plot 函数	18
2.1.2 stem 函数	20
2.1.3 figure 函数和 subplot 函数	21
2.1.4 hold 函数和 grid 函数	22
2.1.5 图形的标注与坐标轴的操作	24
2.1.6 其他二维图形的绘制	25
2.2 三维图形的绘制	29
2.2.1 plot3 函数	29
2.2.2 stem3 函数	29
2.2.3 mesh 函数和 surf 函数	30
2.2.4 三维图形的透视与视角设置	33
2.3 动画的绘制	34
2.3.1 电影动画	34
2.3.2 程序动画	35
2.4 图形 I/O	37
第 3 章 MATLAB 程序设计	53
3.1 变量和常量	53
3.1.1 MATLAB 中的常量	53
3.1.2 MATLAB 中的变量	54
3.2 数据类型	55

3.3 程序结构	58
3.3.1 顺序结构	58
3.3.2 循环结构	58
3.3.3 分支结构	60
3.4 程序流控制语句	62
3.4.1 echo 语句	62
3.4.2 input 和 yesinput 语句	63
3.4.3 pause 和 keyboard 语句	64
3.4.4 break 和 continue 语句	64
3.5 函数调用及变量传递	66
3.5.1 函数调用	66
3.5.2 参数传递	66
第 4 章 可视化界面编程	71
4.1 可视化编程简介	71
4.1.1 布局编辑器	71
4.1.2 属性检查器	74
4.1.3 菜单编辑器	74
4.1.4 回调函数	75
4.2 实例演示	75
4.2.1 程序界面的设计	76
4.2.2 程序代码	83
4.2.3 程序运行结果	91
第 5 章 M 文件转换成独立运行的程序	96
5.1 编译器的选择	96
5.1.1 命令 mex 的使用	96
5.1.2 命令 mbuild 的使用	102
5.2 生成独立的应用程序	108
5.2.1 命令 mcc 的使用	108
5.2.2 生成 C 源代码	117
5.2.3 生成 C++ 源代码	120
5.3 MATLAB 与 VC++ 的连接	122
5.3.1 从 VC++ 中调用 M 函数	122
5.3.2 从 VC++ 中调用 MATLAB C Math Library	125
第 6 章 信号与系统仿真常用函数介绍	130
6.1 信号的产生	130
6.1.1 产生简单信号	131

6.1.2 产生周期信号	132
6.1.3 产生随机序列	134
6.2 信号的输入与输出	135
6.2.1 文件的输入和输出	135
6.2.2 声音的输入和输出	145
6.3 信号的变换和处理	146
6.3.1 符号变换	147
6.3.2 数值计算	149
6.3.3 滤波	153
6.3.4 功率谱估计	159
6.4 系统分析	166
6.4.1 系统的时域分析	166
6.4.2 系统的频域分析	171
6.4.3 系统的模型变换	180
6.4.4 系统的零极点分析	183
第 7 章 数字滤波器设计	186
7.1 IIR 数字滤波器设计	186
7.1.1 IIR 滤波器阶数的选择	186
7.1.2 滤波器的频率变换	189
7.1.3 IIR 数字滤波器设计	190
7.2 FIR 数字滤波器设计	194
7.2.1 常用窗函数	194
7.2.2 窗函数法设计 FIR 数字滤波器	195
7.2.3 其他方法	197
参考文献	201

第 1 章 MATLAB 简介

MATLAB 是美国 MathWorks 公司于 1982 年推出的用于教育、工程与科学计算的软件产品，它向用户提供从概念设计、算法开发、建模仿真到实时实现的理想集成环境。无论是进行科学研究、产品开发，还是从事教育事业，MATLAB 软件都是非常有效的工具。利用该软件，用户可以用类似于数学公式的方式来编写算法，提高了编程效率，节省了时间，让用户把主要精力集中在算法的具体构思上。

1.1 运行界面

运行 MATLAB6.5 后的主界面如图 1.1 所示。从图中可以看出，主界面主要由三个部分组成，左上部分为当前目录显示窗口，列出了当前目录中的文件和文件夹内容；左下部分显示了用户使用 MATLAB 的过程中，输入命令的历史记录，内容包括输入的时间和命令的名称；右边部分为命令窗口，命令提示符为 `>>`，可以在命令提示符的后面输入可执行命令直接执行，执行结果会立即显示在命令行的后面。下面就是一些简单的 MATLAB 命令和执行结果。

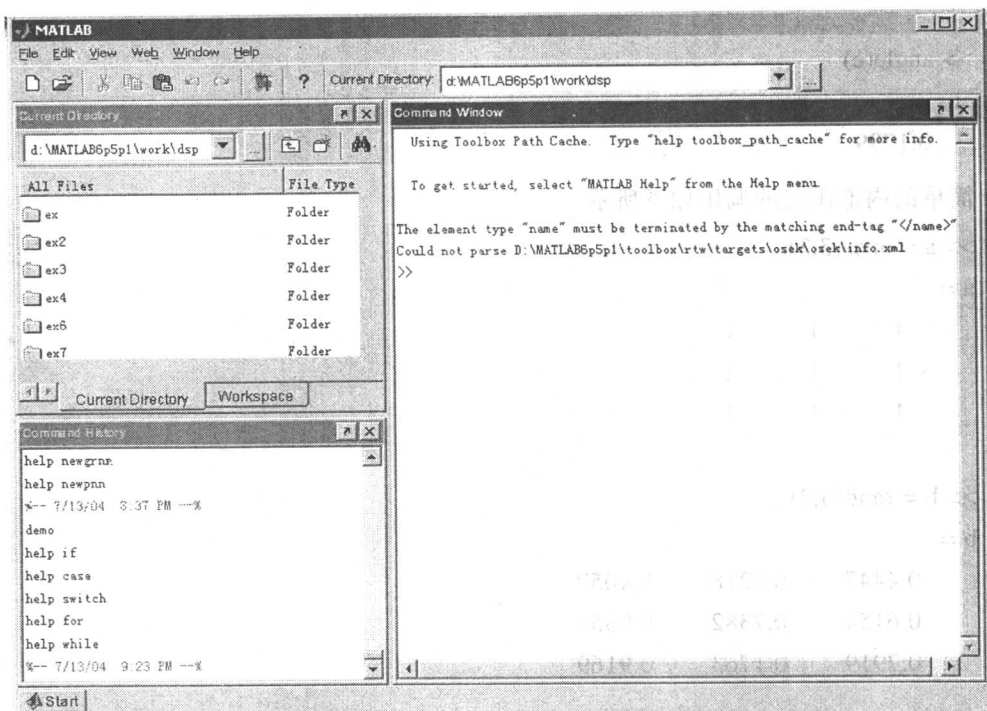


图 1.1 运行 MATLAB6.5 后的主界面

复数的简单运算如下所示:

```
>> a = 1 + 2i  
a =  
1.0000 + 2.0000i
```

```
>> b = 3 + 4i  
b =  
3.0000 + 4.0000i
```

```
>> c = a + b  
c =  
4.0000 + 6.0000i
```

```
>> d = a * b  
d =  
-5.0000 + 10.0000i
```

```
>> e = a / b  
e =  
0.4400 + 0.0800i
```

```
>> angle(e)  
ans =  
0.1799
```

简单的内部函数的调用如下所示:

```
>> a = ones(3,3)  
a =  
1 1 1  
1 1 1  
1 1 1
```

```
>> b = rand(3,3)  
b =  
0.4447 0.9218 0.4057  
0.6154 0.7382 0.9355  
0.7919 0.1763 0.9169
```

```
>> c = dct2(a + b)
```

```
c =
    4.9821   -0.1658    0.1031
   -0.0461    0.0820   -0.6782
   -0.2171    0.1600    0.0357
```

```
>> d = abs(fft2(a + b))
```

```
d =
   14.9464    0.4141    0.4141
    0.4707    0.7805    1.2696
    0.4707    1.2696    0.7805
```

```
>> inv(c)
```

```
ans =
    0.2096    0.0422    0.1956
    0.2801    0.3766    6.3489
    0.0196   -1.4317    0.7541
```

可见，在命令窗口输入命令后，可以立即得到计算的结果，因此，也有人给 MATLAB 取了个别名——草稿纸。

另外，在命令窗口中，使用 help 命令，可以查询函数的帮助文件。例如要查询一维快速傅里叶变换函数 FFT 的帮助文件，则如下所示：

```
>> help fft
```

FFT Discrete Fourier transform.

FFT(X) is the discrete Fourier transform (DFT) of vector X. For matrices, the FFT operation is applied to each column. For N-D arrays, the FFT operation operates on the first non-singleton dimension.

FFT(X,N) is the N-point FFT, padded with zeros if X has less than N points and truncated if it has more.

FFT(X,[],DIM) or FFT(X,N,DIM) applies the FFT operation across the dimension DIM.

For length N input vector x, the DFT is a length N vector X, with elements

N

$$X(k) = \sum_{n=1}^N x(n) \exp(-j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1) / N), 1 \leq k \leq N.$$

The inverse DFT (computed by IFFT) is given by

$$x(n) = (1/N) \sum_{k=1}^N X(k) \exp(j \cdot 2 \cdot \pi \cdot (k-1) \cdot (n-1) / N), 1 \leq n \leq N.$$

See also IFFT, FFT2, IFFT2, FFTSHIFT.

Overloaded methods

help gf/fft.m

help qfft/fft.m

help iddata/fft.m

1.2 矩阵操作

MATLAB 是 Matrix Laboratory (矩阵实验室) 的简写, 该软件最强大的功能之一就是可以把矩阵当作运算单元。和我们常用的高级语言 (如 C 语言等) 相比, 在编程时, 免去了对矩阵操作 (如矩阵的乘、除, 矩阵求逆等) 的代码的编写, 提高了编程效率。

1.2.1 矩阵的输入

在 MATLAB 的命令窗口或者 M 文件中, 矩阵的输入有多种形式。

➤ 用中括号 ([]) 作为矩阵的输入符, 分号 (;) 作为矩阵的行分割符, 逗号 (,) 或者是空格作为矩阵的列分隔符。

```
>> A = [1,2,3;4,5,6;7,8,9]
```

```
A =
```

```
1     2     3
4     5     6
7     8     9
```

➤ 用冒号 (:) 可以生成数列, 一般形式是 A: Step: B, 其中, A 为初始值, B 为终值, Step 为步长, Step 的默认值为 1。

```
>> B = 0:0.1:0.5
```

```
B =
```

```
0    0.1000    0.2000    0.3000    0.4000    0.5000
```

```
>> C = -2:6
```

```
C =
```

```
-2 -1 0 1 2 3 4 5 6
```

► 使用某些 MATLAB 中的内部函数生成矩阵。

```
>> A = ones(3,3)
```

```
A =
```

```
1 1 1
1 1 1
1 1 1
```

```
>> B = eye(3,3)
```

```
B =
```

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

```
>> C = zeros(3,3)
```

```
C =
```

```
0 0 0
0 0 0
0 0 0
```

```
>> D = randn(3,3)
```

```
D =
```

```
-0.4326 0.2877 1.1892
-1.6656 -1.1465 -0.0376
0.1253 1.1909 0.3273
```

```
>> E = magic(6)
```

```
E =
```

```
35 1 6 26 19 24
3 32 7 21 23 25
31 9 2 22 27 20
8 28 33 17 10 15
30 5 34 12 14 16
4 36 29 13 18 11
```

```
>> F = hadamard(4)
```

F =

```

1     1     1     1
1    -1     1    -1
1     1    -1    -1
1    -1    -1     1

```

另外, MATLAB 中的矩阵必须是矩形的, 没有定义的元素默认为 0。例如若只告诉某矩阵 A 的第 3 行第 4 列元素为 99, MATLAB 自动将 A 扩展为一个 3 行 4 列的矩阵, 没有定义的元素默认为 0。

```
>> A(3,4) = 99
```

A =

```

0     0     0     0
0     0     0     0
0     0     0    99

```

1.2.2 矩阵的索引

在 MATLAB 中对矩阵元素进行索引与 C 语言类似。常用的方法有:

➤ 使用行、列的形式进行索引。MATLAB 中矩阵的索引从矩阵的左上角开始, 初始值为 1。

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
```

A =

```

1     2     3     4
5     6     7     8
9    10    11    12
13   14   15   16

```

```
>> A(3,3)
```

ans =

```
11
```

```
>> A(2,4)
```

ans =

```
8
```

```
>> A(1:3,2:4)
```

ans =

```

2     3     4
6     7     8
10    11    12

```

```
>> A(2:4,:)
```

```
ans =
```

```
    5    6    7    8
    9   10   11   12
   13   14   15   16
```

```
>> A(:,1:3)
```

```
ans =
```

```
    1    2    3
    5    6    7
    9   10   11
   13   14   15
```

► 使用绝对下标索引。MATLAB 最初是用 FORTRAN 语言编写的, FORTRAN 语言中对数组数据的存储是按列依次进行的。所以, 在 MATLAB 中, 矩阵的元素可以用绝对下标来引用, 绝对下标索引是按从上到下、从左到右的顺序进行的, 初始值是 1。

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
```

```
A =
```

```
    1    2    3    4
    5    6    7    8
    9   10   11   12
   13   14   15   16
```

```
>> A(2)
```

```
ans =
```

```
5
```

```
>> A(10)
```

```
ans =
```

```
7
```

```
>> A(1:6)
```

```
ans =
```

```
    1    5    9   13    2    6
```

```
>> A(7:16)
```

```
ans =
```

```
   10   14    3    7   11   15    4    8   12   16
```

1.2.3 矩阵的基本运算

➤ 矩阵的加法(减法)运算。矩阵间的加法(减法)为矩阵对应元素的加(减)。矩阵与常数的加(减),等于矩阵的每一个元素同时加(减)该常数。

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

```
A =
```

```
1     2     3
4     5     6
7     8     9
```

```
>> B = rand(3,3)
```

```
B =
```

```
0.9501    0.4860    0.4565
0.2311    0.8913    0.0185
0.6068    0.7621    0.8214
```

```
>> C = A + B
```

```
C =
```

```
1.9501    2.4860    3.4565
4.2311    5.8913    6.0185
7.6068    8.7621    9.8214
```

```
>> D = B - 0.5
```

```
D =
```

```
0.4501   -0.0140   -0.0435
-0.2689    0.3913   -0.4815
0.1068    0.2621    0.3214
```

```
>> E = A + 2 - B
```

```
E =
```

```
2.0499    3.5140    4.5435
5.7689    6.1087    7.9815
8.3932    9.2379   10.1786
```

➤ 矩阵的求逆运算。矩阵求逆可以用函数 `inv`。

```
>> A = [1 2 3; 6 4 5; 7 8 9];
```

```
>> B = inv(A)
```

```
B =
```

```
-0.2222    0.3333   -0.1111
-1.0556   -0.6667    0.7222
 1.1111    0.3333   -0.4444
```

```
>> D = A * B
```

```
D =
```

```
1.0000         0         0
         0    1.0000         0
         0         0    1.0000
```

► 矩阵的共轭转置运算和转置运算。

```
>> A = [1+2i,3+4i;5+6i,7+8i]
```

```
A =
```

```
1.0000 + 2.0000i    3.0000 + 4.0000i
5.0000 + 6.0000i    7.0000 + 8.0000i
```

```
>> A'
```

```
ans =
```

```
1.0000 - 2.0000i    5.0000 - 6.0000i
3.0000 - 4.0000i    7.0000 - 8.0000i
```

```
>> A.'
```

```
ans =
```

```
1.0000 + 2.0000i    5.0000 + 6.0000i
3.0000 + 4.0000i    7.0000 + 8.0000i
```

► 矩阵的乘法运算、左除运算和右除运算。

```
>> A = [1 2 3; 6 4 5; 7 8 9]; B = [1 4 7; 2 5 8; 3 3 9];
```

```
>> C = A * B
```

```
C =
```

```
14    23    50
29    59   119
50    95   194
```

```
>> D = A / B
```

```
D =
```

```
-0.3333    0.6667         0
-5.8333    5.1667    0.5000
-6.3333    6.6667    0.0000
```

```
>> E = A * inv(B)
```

```
E =
    -0.3333    0.6667    0
    -5.8333    5.1667    0.5000
    -6.3333    6.6667    0
```

```
>> F = A \ B
```

```
F =
    0.1111    0.4444    0.1111
   -0.2222   -5.3889   -6.2222
    0.4444    4.7778    6.4444
```

```
>> G = inv(A) * B
```

```
G =
    0.1111    0.4444    0.1111
   -0.2222   -5.3889   -6.2222
    0.4444    4.7778    6.4444
```

从上面的例子可以看出，左除以某个矩阵相当于左乘以该矩阵的逆矩阵，同样，右除以某个矩阵等价于右乘以该矩阵的逆矩阵。

➤ 矩阵的点乘运算和点除运算。

```
>> A = 2 * ones(3,3)
```

```
A =
     2     2     2
     2     2     2
     2     2     2
```

```
>> B = [1 1 1; 2 2 2; 3 3 3]
```

```
B =
     1     1     1
     2     2     2
     3     3     3
```

```
>> C = A .* B
```

```
C =
     2     2     2
     4     4     4
     6     6     6
```

```
>> D = A ./ B
```