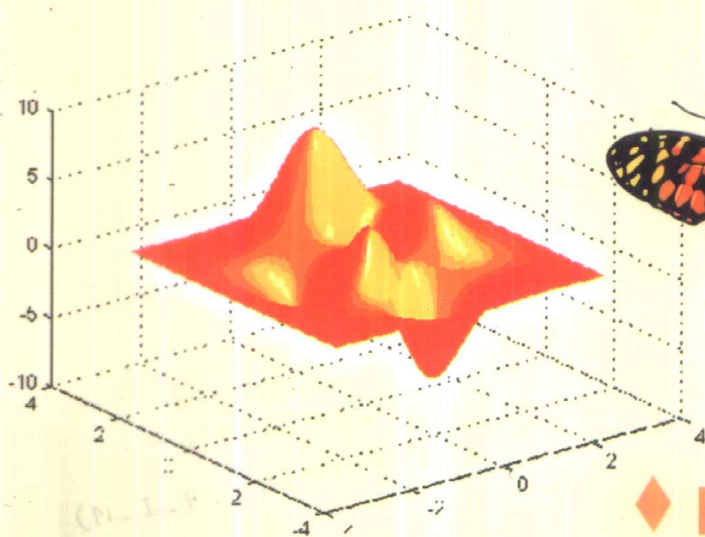


MATLAB6.0
STUDENT SOURCE



MATLAB6.0 数学手册

蒲俊 吉家锋 伊良忠 © 编著



讲解 MATLAB 在数学各分支中的应用

讲解 MATLAB 函数和命令

荟萃 MATLAB 应用实例及技巧

提供实例丰富的多媒体光盘



浦东电子出版社

PSP Pudong Epress

73/

013
p88

MATLAB6.0 数学手册

蒲 俊 吉家锋 伊良忠 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制



浦东电子出版社
PeP Pudong ePress

内 容 提 要

MATLAB 已成为多学科、多种工作平台的功能强大、界面友好、语言自然并且开放性强的应用软件,目前的最高版本是 6.0 版。本教程以 6.0 版为基础,从高等工科院校的数学课程出发,提供了使用 **MATLAB** 的实践性指导。本教程以教学的手段,系统详细地介绍了 **MATLAB** 在高等数学、数值分析、函数作图、线性代数、概率统计和优化理论中的应用,并配备了大量的例题,让读者能很快掌握 **MATLAB** 的运算技巧。

本教程按逻辑编排,自始至终用实例描述,既适用于初学者自学,也适用于高级 **MATLAB** 用户。可作为高等数学、数值分析、工程数学、数学建模、线性规划等课程的教学参考书,也可作为科技工作者学习和使用 **MATLAB** 的参考书,还可作为数学实验的教学用书,特别适合用作理工科大学学生学习数学课程的教学辅导书。

本教程的光盘内容详尽、实例丰富:包含了 **MATLAB** 实例的源文件,函数/命令及注解,程序实例。

书 名: MATLAB6.0 数学手册

文本著作者: 蒲 俊 吉家锋 伊良忠

C D 制 作 者: 海搏多媒体制作中心

责 任 编 辑: 舒红梅

出版、发行者: 浦东电子出版社

地 址: 上海浦东郭守敬路 498 号上海浦东软件园内 201203

电话: 021-38954510, 38953321, 38953323 (发行部)

经 销: 各地新华书店、软件连锁店

排 版: 四川中外科技文化交流中心排版制作中心

C D 生 产 者: 东方光盘制造有限公司

文本印刷者: 成都地图出版社印刷厂

开 本 / 规 格: 787×1092 毫米 16 开本 19.5 印张 240 千字

版 次 / 印 次: 2002 年 1 月第一版 2002 年 1 月第一次印刷

印 数: 0001——8000 册

本 版 号: ISBN 7-900346-16-3

定 价: 33.00 元 (1CD 配使用手册)

说明: 凡我社光盘配套图书有缺页、倒页、脱页、自然破损,本社发行部负责调换。

前 言

MATLAB 是美国 MathWorks 公司自 20 世纪 80 年代中期推出的数学软件，优秀的数值计算能力和卓越的数据可视化能力使其很快在数学软件中脱颖而出。到目前为止，其最高版本 6.0 版已经推出。随着版本的不断升级，它在数值计算及符号计算功能上得到了进一步完善。MATLAB 已经发展成为多学科、多种工作平台的功能强大的大型软件。在欧美等高校，MATLAB 已经成为线性代数、自动控制理论、概率论及数理统计、数字信号处理、时间序列分析、动态系统仿真等高级课程的基本教学工具，是攻读学位的大学生、硕士生、博士生必须掌握的基本技能。

MATLAB 的主要特点是：

- 有高性能数值计算的高级算法，特别适合矩阵代数领域；
- 有大量事先定义的数学函数，并且有很强的用户自定义函数的能力；
- 有强大的绘图功能以及具有教育、科学和艺术学的图解和可视化的二维、三维图；
- 基于 HTML 的完整的帮助功能；
- 适合个人应用的强有力的面向矩阵(向量)的高级程序设计语

言;

- 与其它语言编写的程序结合和输入输出格式化数据的能力;
- 有在多个应用领域解决难题的工具箱。

本教程提供了使用 **MATLAB** 的实践性指导,它基于 **MATLAB6.0** 版,内容由浅入深,特别是本书对每一条命令的使用格式都作了详细而又简单明了的说明,并配备了例题加以说明其用法,因此,对于初学者自学是很有帮助的;同时,又对数学中的一些深入问题如数值分析、稀疏矩阵、优化理论以及模糊数学等问题进行了较为详细的论述,因此,该书也可作为科技工作者的科学计算工具书。

本教程的具体特点是:

- 它是以简明方法写就的一本易于掌握的数学手册;
- 编写逻辑性强,内容由浅入深,对于初学者能很快掌握 **MATLAB** 的用法;
- 易于查找命令和问题,给读者灵感与启迪,以解决实际问题;
- 对每一条命令,都进行了详细论述;
- 对于每一条命令,几乎都有易懂的实例;
- 内容按数学分类进行描述。

本教程的光盘内容详尽、实例丰富:包含了 **MATLAB** 实例的源文件,函数/命令及注解,程序实例。

目 录

第 1 章 矩阵及其基本运算	1
1.1 矩阵的表示	1
1.1.1 数值矩阵的生成	1
1.1.2 符号矩阵的生成	2
1.1.3 大矩阵的生成	3
1.1.4 多维数组的创建	3
1.1.5 特殊矩阵的生成	4
1.2 矩阵运算	9
1.2.1 加、减运算	9
1.2.2 乘法	9
1.2.3 集合运算	12
1.2.4 除法运算	15
1.2.5 矩阵乘方	16
1.2.6 矩阵函数	16
1.2.7 矩阵转置	17
1.2.8 方阵的行列式	17
1.2.9 逆与伪逆	18
1.2.10 矩阵的迹	19
1.2.11 矩阵和向量的范数	19
1.2.12 条件数	20
1.2.13 矩阵的秩	20
1.2.14 特殊运算	21
1.2.15 符号矩阵运算	26
1.2.16 矩阵元素个数的确定	29
1.3 矩阵分解	29
1.3.1 Cholesky 分解	29
1.3.2 LU 分解	30
1.3.3 QR 分解	30
1.3.4 Schur 分解	32
1.3.5 实 Schur 分解转化成复 Schur 分解	32
1.3.6 特征值分解	33
1.3.7 奇异值分解	33
1.3.8 广义奇异值分解	34
1.3.9 特征值问题的 QZ 分解	35
1.3.10 海森伯格形式的分解	35
1.4 线性方程的组的求解	35
1.4.1 求线性方程组的唯一解或特解（第一类问题）	35
1.4.2 求线性齐次方程组的通解	38
1.4.3 求非齐次线性方程组的通解	39
1.4.4 线性方程组的 LQ 解法	41
1.4.5 双共轭梯度法解方程组	41
1.4.6 稳定双共轭梯度方法解方程组	42
1.4.7 复共轭梯度平方解方程组	43
1.4.8 共轭梯度的 LSQR 方法	44
1.4.9 广义最小残差法	44
1.4.10 最小残差法解方程组	45
1.4.11 预处理共轭梯度方法	46
1.4.12 准最小残差法解方程组	46

1.5	特征值与二次型	47
1.5.1	特征值与特征向量的求法	47
1.5.2	提高特征值的计算精度	48
1.5.3	复对角矩阵转化为实对角矩阵	48
1.5.4	正交基	49
1.5.5	二次型	49
1.6	秩与线性相关性	50
1.6.1	矩阵和向量组的秩以及向量组的线性相关性	50
1.6.2	求行阶梯矩阵及向量组的基	50
1.7	稀疏矩阵技术	51
1.7.1	稀疏矩阵的创建	51
1.7.2	将稀疏矩阵转化为满矩阵	52
1.7.3	稀疏矩阵非零元素的索引	53
1.7.4	外部数据转化为稀疏矩阵	53
1.7.5	基本稀疏矩阵	53
1.7.6	稀疏矩阵的运算	55
1.7.7	画稀疏矩阵非零元素的分布图形	56
1.7.8	矩阵变换	56
1.7.9	稀疏矩阵的近似欧几里得范数和条件数	59
1.7.10	稀疏矩阵的分解	59
1.7.11	稀疏矩阵的特征值分解	61
1.7.12	稀疏矩阵的线性方程组	61
第 2 章	数值计算与数据分析	62
2.1	基本数学函数	62
2.1.1	三角函数与双曲函数	62
2.1.2	其他常用函数	69
2.2	插值、拟合与查表	76
2.2.1	插值命令	77
2.2.2	查表命令	83
2.3	数值积分	84
2.3.1	一元函数的数值积分	84
2.3.2	二元函数重积分的数值计算	86
2.4	常微分方程数值解	87
2.5	偏微分方程的数值解	90
2.5.1	单的 Poisson 方程	91
2.5.2	双曲型偏微分方程	92
2.5.3	抛物型偏微分方程	93
第 3 章	符号运算	95
3.1	算术符号操作	95
3.2	基本运算	97
3.2.1	函数计算器	108
3.2.2	微积分	109
3.2.3	符号函数的作图	112
3.2.4	积分变换	118
3.2.5	Taylor 级数	123
3.2.6	其它	124
第 4 章	概率统计	134
4.1	随机数的产生	134
4.1.1	二项分布的随机数据的产生	134
4.1.2	正态分布的随机数据的产生	134

4.1.3	常见分布的随机数产生	135
4.1.4	通用函数求各分布的随机数据	135
4.2	随机变量的概率密度计算	136
4.2.1	通用函数计算概率密度函数值	136
4.2.2	专用函数计算概率密度函数值	137
4.2.3	常见分布的密度函数作图	138
4.3	随机变量的累积概率值(分布函数值)	141
4.3.1	通用函数计算累积概率值	141
4.3.2	专用函数计算累积概率值(随机变量 $X \leq K$ 的概率之和)	141
4.4	随机变量的逆累积分布函数	143
4.4.1	通用函数计算逆累积分布函数值	143
4.4.2	专用函数 inv 计算逆累积分布函数	143
4.5	随机变量的数字特征	145
4.5.1	平均值、中值	145
4.5.2	数据比较	147
4.5.3	期望	148
4.5.4	方差	149
4.5.5	常见分布的期望和方差	151
4.5.6	协方差与相关系数	152
4.6	统计作图	153
4.6.1	正整数的频率表	153
4.6.2	经验累积分布函数图形	154
4.6.3	最小二乘拟合直线	154
4.6.4	绘制正态分布概率图形	154
4.6.5	绘制威布尔(Weibull)概率图形	155
4.6.6	样本数据的盒图	155
4.6.7	给当前图形加一条参考线	156
4.6.8	在当前图形中加入一条多项式曲线	156
4.6.9	样本的概率图形	157
4.6.10	附加有正态密度曲线的直方图	157
4.6.11	在指定的界线之间画正态密度曲线	158
4.7	参数估计	158
4.7.1	常见分布的参数估计	158
4.7.2	非线性模型置信区间预测	160
4.7.3	对数似然函数	164
4.8	假设检验	165
4.8.1	σ^2 已知, 单个正态总体的均值 μ 的假设检验(U 检验法)	165
4.8.2	σ^2 未知, 单个正态总体的均值 μ 的假设检验(t 检验法)	166
4.8.3	两个正态总体均值差的检验(t 检验)	167
4.8.4	两个总体一致性的检验——秩和检验	168
4.8.5	两个总体中位数相等的假设检验——符号秩检验	168
4.8.6	两个总体中位数相等的假设检验——符号检验	169
4.8.7	正态分布的拟合优度测试	169
4.8.8	正态分布的拟合优度测试	170
4.8.9	单个样本分布的 Kolmogorov-Smirnov 测试	170
4.8.10	两个样本具有相同的连续分布的假设检验	171
4.9	方差分析	172
4.9.1	单因素方差分析	172
4.9.2	双因素方差分析	174
第 5 章	优化问题	176

5.1	线性规划问题.....	176
5.2	foptions 函数.....	177
5.3	非线性规划问题.....	178
5.3.1	有约束的一元函数的最小值.....	178
5.3.2	无约束多元函数最小值.....	179
5.3.3	有约束的多元函数最小值.....	181
5.3.4	二次规划问题.....	183
5.4	“半无限”有约束的多元函数最优解.....	185
5.5	极小化极大 (Minmax) 问题.....	189
5.6	多目标规划问题.....	191
5.7	最小二乘最优问题.....	194
5.7.1	约束线性最小二乘.....	194
5.7.2	非线性数据 (曲线) 拟合.....	195
5.7.3	非线性最小二乘.....	196
5.7.4	非负线性最小二乘.....	198
5.8	非线性方程(组)求解.....	198
5.8.1	非线性方程的解.....	198
5.8.2	非线性方程组的解.....	199
第 6 章	模糊逻辑.....	201
6.1	隶属函数.....	201
6.1.1	高斯隶属函数.....	201
6.1.2	两边型高斯隶属函数.....	201
6.1.3	建立一般钟型隶属函数.....	202
6.1.4	两个 sigmoid 型隶属函数之差组成的隶属函数.....	202
6.1.5	通用隶属函数计算.....	203
6.1.6	建立 Π 型隶属函数.....	203
6.1.7	通过两个 sigmoid 型隶属函数的乘积构造隶属函数.....	204
6.1.8	建立 Sigmoid 型隶属函数.....	204
6.1.9	建立 S 型隶属函数.....	205
6.1.10	建立梯形隶属函数.....	206
6.1.11	建立三角形隶属函数.....	207
6.1.12	建立 Z 型隶属函数.....	208
6.1.13	两个隶属函数之间转换参数.....	209
6.1.14	基本 FIS 编辑器.....	209
6.1.15	隶属函数编辑器.....	211
6.2	模糊推理结构 FIS.....	212
6.2.1	不使用数据聚类方法从数据生成 FIS 结构.....	212
6.2.2	使用减法聚类方法从数据生成 FIS 结构.....	213
6.2.3	生成一个 FIS 输出曲面.....	213
6.2.4	将 mamdan 型 FIS 转换为 Sugeno FIS.....	214
6.2.5	完成模糊推理计算.....	214
6.2.6	模糊 c 均值聚类.....	215
6.2.7	模糊均值和减法聚类.....	215
6.2.8	绘制一个 FIS.....	216
6.2.9	绘制给定变量的所有隶属的曲线.....	216
6.2.10	从磁盘装入一个 FIS.....	217
6.2.11	从 FIS 中删除某一隶属函数.....	218
6.2.12	从 FIS 中删除变量.....	218
6.2.13	设置模糊系统属性.....	219
6.2.14	以分行形式显示 FIS 结构的所有属性.....	220

6.2.15	完成模糊运算.....	221
6.2.16	解析模糊规则.....	222
6.2.17	规则编辑器和语法编辑器.....	223
6.2.18	规则观察器和模糊推理框图.....	224
6.2.19	保存 FIS 到磁盘上.....	224
6.2.20	显示 FIS 的规则.....	225
6.2.21	显示 FIS 结构的所有属性.....	226
第 7 章	绘图与图形处理.....	228
7.1	二维图形.....	228
7.1.1	基本平面图形命令.....	228
7.1.2	特殊平面图形命令.....	235
7.1.3	二维图形注释命令.....	241
7.2	三维图形.....	245
7.2.1	三维曲线、面填色命令.....	245
7.2.2	三维图形等高线.....	247
7.2.3	曲面与网格图命令.....	250
7.2.4	三维数据的其他表现形式命令.....	254
7.3	通用图形函数命令.....	260
7.3.1	图形对象句柄命令.....	260
7.3.2	轴的产生和控制命令.....	271
7.3.3	图形句柄操作命令.....	272
7.3.4	图形窗口的控制命令.....	274
7.4	颜色与光照模式命令.....	276
7.4.1	颜色控制命令.....	276
7.4.2	色图控制命令.....	278
第 8 章	MATLAB 编程.....	280
8.1	MATLAB 的注释和标点.....	280
8.2	MATLAB 的编程语言.....	280
8.2.1	M-文件编写的函数.....	280
8.2.2	交互式输入.....	288
8.2.3	程序控制流.....	289
8.2.3	逻辑函数.....	296
8.3	M-文件的出错信息与调试.....	297
8.3.1	M-文件执行的出错信息.....	297
8.3.2	函数的调试命令.....	298



第1章 矩阵及其基本运算

MATLAB, 即“矩阵实验室”, 它是以矩阵为基本运算单元。因此, 本书从最基本的运算单元出发, 介绍 MATLAB 的命令及其用法。

1.1 矩阵的表示

1.1.1 数值矩阵的生成

1. 实数值矩阵输入

MATLAB 的强大功能之一体现在能直接处理向量或矩阵。当然首要任务是输入待处理的向量或矩阵。

不管是任何矩阵(向量), 我们可以直接按行方式输入每个元素: 同一行中的元素用逗号(,) 或者用空格符来分隔, 且空格个数不限; 不同的行用分号(;) 分隔。所有元素处于一方括号([]) 内; 当矩阵是多维(三维以上), 且方括号内的元素是维数较低的矩阵时, 会有多重的方括号。如:

```
>> Time = [11' 12 1 2 3 4 5 6 7 8 9 10]
Time =
    11    12     1     2     3     4     5     6     7     8     9    10
>> X_Data = [2.32 3.43; 4.37 5.98]
X_Data =
    2.43    3.43
    4.37    5.98
>> vect_a = [1 2 3 4 5]
vect_a =
     1     2     3     4     5
>> Matrix_B = [1 2 3;
               2 3 4; 3 4 5]
Matrix_B = 1 2 3
            2 3 4
            3 4 5
>> Null_M = [] %生成一个空矩阵
```

2. 复数矩阵输入

复数矩阵有两种生成方式:

第一种方式

例 1-1

```
>> a=2.7;b=13/25;
>> C=[1,2*a+i*b,b*sqrt(a); sin(pi/4),a+5*b,3.5+1]
C=
    1.0000          5.4000 + 0.5200i    0.8544
    0.7071          5.3000          4.5000
```

第2种方式

例 1-2

```
>> R=[1 2 3;4 5 6], M=[11 12 13;14 15 16]
R =
     1     2     3
     4     5     6
M =
    11    12    13
    14    15    16
>> CN=R+i*M
CN =
 1.0000 +11.0000i  2.0000 +12.0000i  3.0000 +13.0000i
 4.0000 +14.0000i  5.0000 +15.0000i  6.0000 +16.0000i
```

1.1.2 符号矩阵的生成

在 MATLAB 中输入符号向量或者矩阵的方法和输入数值类型的向量或者矩阵在形式上很相像，只不过要用到符号矩阵定义函数 `sym`，或者是用到符号定义函数 `syms`，先定义一些必要的符号变量，再像定义普通矩阵一样输入符号矩阵。

1. 用命令 `sym` 定义矩阵：

这时的函数 `sym` 实际是在定义一个符号表达式，这时的符号矩阵中的元素可以是任何的符号或者是表达式，而且长度没有限制，只是将方括号置于用于创建符号表达式的单引号中。如下例：

例 1-3

```
>> sym_matrix = sym ('[a b c; Jack, Help Me!, NO WAY!]', 's')
sym_matrix =
     [a          b          c]
     [Jack   Help Me!  NO WAY!]
>> sym_digits = sym ('[1 2 3; a b c; sin (x) cos (y) tan (z)]')
sym_digits =
     [1          2          3]
     [a          b          c]
     [sin (x) cos (y) tan (z)]
```

2. 用命令 `syms` 定义矩阵

先定义矩阵中的每一个元素为一个符号变量，而后像普通矩阵一样输入符号矩阵。

例 1-4

```
>> syms a b c ;
>> M1 = sym ('Classical');
>> M2 = sym ('Jazz');
>> M3 = sym ('Blues')
>> syms_matrix = [a b c; M1, M2, M3; int2str ([2 3 5])]
syms_matrix =
     [ a          b          c]
     [Classical  Jazz   Blues]
     [ 2          3          5]
```

把数值矩阵转化成相应的符号矩阵。

数值型和符号型在 MATLAB 中是不相同的，它们之间不能直接进行转化。MATLAB 提供了一个将数值型转化成符号型的命令，即 `sym`。

例 1-5

```
>> Digit_Matrix = [1/3 sqrt(2) 3.4234; exp(0.23) log(29) 23^(-11.23)]
>> Syms_Matrix = sym(Digit_Matrix)
```

结果是:

```
Digit_Matrix =
    0.3333    1.4142    3.4234
    1.2586    3.3673    0.0000

Syms_Matrix =
[ 1/3, sqrt(2), 17117/5000]
[5668230535726899*2^(-52), 7582476122586655*2^(-51), 5174709270083729*2^(-103)]
```

注意: 矩阵是用分数形式还是浮点形式表示的, 将矩阵转化成符号矩阵后, 都将以最接近原值的有理数形式表示或者是函数形式表示。

1.1.3 大矩阵的生成

对于大型矩阵, 一般创建 M 文件, 以便于修改:

例 1-6 用 M 文件创建大矩阵, 文件名为 example.m

```
exm=[ 456    468    873     2   579    55
      21    687    54    488     8    13
      65   4567    88     98    21     5
      456    68   4589   654     5   987
      5488   10     9     6    33   77]
```

在 MATLAB 窗口输入:

```
>>example;
>>size(exm) %显示 exm 的大小
ans=
     5     6 %表示 exm 有 5 行 6 列。
```

1.1.4 多维数组的创建

函数 cat

格式 A=cat(n,A1,A2,...,Am)

说明 n=1 和 n=2 时分别构造[A1; A2]和[A1, A2], 都是二维数组, 而 n=3 时可以构造出三维数组。

例 1-7

```
>> A1=[1,2,3;4,5,6;7,8,9];A2=A1';A3=A1-A2;
>> A4=cat(3,A1,A2,A3)
A4(:,:,1) =
     1     2     3
     4     5     6
     7     8     9
A4(:,:,2) =
     1     4     7
     2     5     8
     3     6     9
A4(:,:,3) =
     0    -2    -4
     2     0    -2
     4     2     0
```

或用另一种原始方式可以定义:

例 1-8

```
>> A1=[1,2,3;4,5,6;7,8,9];A2=A1';A3=A1-A2;
```

```
>> A5(:,1)=A1, A5(:,2)=A2, A5(:,3)=A3
```

```
A5(:,1) =
```

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

```
A5(:,2) =
```

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

```
A5(:,3) =
```

```
0     -2    -4
2     0     -2
4     2     0
```

1.1.5 特殊矩阵的生成

命令 全零阵

函数 **zeros**

格式 **B = zeros(n)**

%生成 $n \times n$ 全零阵

B = zeros(m,n)

%生成 $m \times n$ 全零阵

B = zeros([m n])

%生成 $m \times n$ 全零阵

B = zeros(d1,d2,d3...)

%生成 $d1 \times d2 \times d3 \times \dots$ 全零阵或数组

B = zeros([d1 d2 d3...])

%生成 $d1 \times d2 \times d3 \times \dots$ 全零阵或数组

B = zeros(size(A))

%生成与矩阵 A 相同大小的全零阵

命令 单位阵

函数 **eye**

格式 **Y = eye(n)**

%生成 $n \times n$ 单位阵

Y = eye(m,n)

%生成 $m \times n$ 单位阵

Y = eye(size(A))

%生成与矩阵 A 相同大小的单位阵

命令 全 1 阵

函数 **ones**

格式 **Y = ones(n)**

%生成 $n \times n$ 全 1 阵

Y = ones(m,n)

%生成 $m \times n$ 全 1 阵

Y = ones([m n])

%生成 $m \times n$ 全 1 阵

Y = ones(d1,d2,d3...)

%生成 $d1 \times d2 \times d3 \times \dots$ 全 1 阵或数组

Y = ones([d1 d2 d3...])

%生成 $d1 \times d2 \times d3 \times \dots$ 全 1 阵或数组

Y = ones(size(A))

%生成与矩阵 A 相同大小的全 1 阵

命令 均匀分布随机矩阵

函数 **rand**

格式 **Y = rand(n)**

%生成 $n \times n$ 随机矩阵, 其元素在 (0, 1) 内

Y = rand(m,n)

%生成 $m \times n$ 随机矩阵

`Y = rand([m n])` %生成 $m \times n$ 随机矩阵
`Y = rand(m,n,p,...)` %生成 $m \times n \times p \times \dots$ 随机矩阵或数组
`Y = rand([m n p ...])` %生成 $m \times n \times p \times \dots$ 随机矩阵或数组
`Y = rand(size(A))` %生成与矩阵 A 相同大小的随机矩阵
`rand` %无变量输入时只产生一个随机数
`s = rand('state')` %产生包括均匀发生器当前状态的 35 个元素的向量
`rand('state', s)` %使状态重置为 s
`rand('state', 0)` %重置发生器到初始状态
`rand('state', j)` %对整数 j 重置发生器到第 j 个状态
`rand('state', sum(100*clock))` %每次重置到不同状态

例 1-9 产生一个 3×4 随机矩阵

```
>> R=rand(3,4)
R =
    0.9501    0.4860    0.4565    0.4447
    0.2311    0.8913    0.0185    0.6154
    0.6068    0.7621    0.8214    0.7919
```

例 1-10 产生一个在区间[10, 20]内均匀分布的 4 阶随机矩阵

```
>> a=10;b=20;
>> x=a+(b-a)*rand(4)
x =
    19.2181    19.3547    10.5789    11.3889
    17.3821    19.1690    13.5287    12.0277
    11.7627    14.1027    18.1317    11.9872
    14.0571    18.9365    10.0986    16.0379
```

命令 正态分布随机矩阵

函数 **randn**

格式

`Y = randn(n)` %生成 $n \times n$ 正态分布随机矩阵
`Y = randn(m,n)` %生成 $m \times n$ 正态分布随机矩阵
`Y = randn([m n])` %生成 $m \times n$ 正态分布随机矩阵
`Y = randn(m,n,p,...)` %生成 $m \times n \times p \times \dots$ 正态分布随机矩阵或数组
`Y = randn([m n p ...])` %生成 $m \times n \times p \times \dots$ 正态分布随机矩阵或数组
`Y = randn(size(A))` %生成与矩阵 A 相同大小的正态分布随机矩阵
`randn` %无变量输入时只产生一个正态分布随机数
`s = randn('state')` %产生包括正态发生器当前状态的 2 个元素的向量
`s = randn('state', s)` %重置状态为 s
`s = randn('state', 0)` %重置发生器为初始状态
`s = randn('state', j)` %对于整数 j 重置状态到第 j 状态
`s = randn('state', sum(100*clock))` %每次重置到不同状态

例 1-11 产生均值为 0.6, 方差为 0.1 的 4 阶矩阵

```
>> mu=0.6; sigma=0.1;
>> x=mu+sqrt(sigma)*randn(4)
x =
    0.8311    0.7799    0.1335    1.0565
    0.7827    0.5192    0.5260    0.4890
```

```
0.6127    0.4806    0.6375    0.7971
0.8141    0.5064    0.6996    0.8527
```

命令 产生随机排列

函数 **randperm**

格式 `p = randperm(n)` %产生 1~n 之间整数的随机排列

例 1-12

```
>> randperm(6)
```

```
ans =
     3     2     1     5     4     6
```

命令 产生线性等分向量

函数 **linspace**

格式 `y = linspace(a,b)` %在(a, b)上产生 100 个线性等分点

`y = linspace(a,b,n)` %在(a, b)上产生 n 个线性等分点

命令 产生对数等分向量

函数 **logspace**

格式 `y = logspace(a,b)` %在 (10^a , 10^b) 之间产生 50 个对数等分向量

`y = logspace(a,b,n)`

`y = logspace(a,pi)`

命令 计算矩阵中元素个数

`n = numel(a)` %返回矩阵 A 的元素的个数

命令 产生以输入元素为对角线元素的矩阵

函数 **blkdiag**

格式 `out = blkdiag(a,b,c,d,...)` %产生以 a,b,c,d,...为对角线元素的矩阵

例 1-13

```
>> out = blkdiag(1,2,3,4)
```

```
out =
     1     0     0     0
     0     2     0     0
     0     0     3     0
     0     0     0     4
```

命令 友矩阵

函数 **companion**

格式 `A = companion(u)` %u 为多项式系统向量, A 为友矩阵, A 的第 1 行元素为 -u(2:n)/u(1), 其中 u(2:n)为 u 的第 2 到第 n 个元素, A 为特征值就是多项式的特征根。

例 1-14 求多项式 $(x-1)(x-2)(x+3) = x^3 - 7x + 6$ 的友矩阵和根

```
>> u=[1 0 -7 6];
```

```
>> A=companion(u)      %求多项式的友矩阵
```

```
A =
     0     7    -6
     1     0     0
     0     1     0
```

```
>> eig(A)      %A 的特征值就是多项式的根
```

```
ans =
```

-3.0000
2.0000
1.0000

命令 **hadamard** 矩阵

函数 **hadamard**

格式 $H = \text{hadamard}(n)$ %返回 n 阶 hadamard 矩阵

例 1-15

```
>> h=hadamard(4)
h =
     1     1     1     1
     1    -1     1    -1
     1     1    -1    -1
     1    -1    -1     1
```

命令 **Hankel** 方阵

函数 **hankel**

格式 $H = \text{hankel}(c)$ %第 1 列元素为 c , 反三角以下元素为 0。

$H = \text{hankel}(c,r)$ %第 1 列元素为 c , 最后一行元素为 r , 如果 c 的最后一个元素与 r 的第一个元素不同, 交叉位置元素取为 c 的最后一个元素。

例 1-16

```
>> c=1:3,r=7:10
c =
     1     2     3
r =
     7     8     9    10
>> h=hankel(c,r)
h =
     1     2     3     8
     2     3     8     9
     3     8     9    10
```

命令 **Hilbert** 矩阵

函数 **hilb**

格式 $H = \text{hilb}(n)$ %返回 n 阶 Hilbert 矩阵, 其元素为 $H(i,j)=1/(i+j-1)$ 。

例 1-17 产生一个 3 阶 Hilbert 矩阵

```
>> format rat      %以有理形式输出
>> H=hilb(3)
H =
     1          1/2          1/3
    1/2          1/3          1/4
    1/3          1/4          1/5
```

命令 逆 Hilbert 矩阵

函数 **invhilb**

格式 $H = \text{invhilb}(n)$ %产生 n 阶逆 Hilbert 矩阵

命令 **Magic(魔方)** 矩阵

函数 **magic**

格式 $M = \text{magic}(n)$ %产生 n 阶魔方矩阵

例 1-18