

工程硕士数学主干课程系列教材

最 优 化 计 算 方 法 与 实 现

蒋金山 编

ZUIYOUHUA JISUAN FANGFA YU SHIXIAN



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

工程硕士数学主干课程系列教材

最优化计算方法与实现

蒋金山 编

华南理工大学出版社

· 广州 ·

图书在版编目(CIP)数据

最优化计算方法与实现/蒋金山编. —广州: 华南理工大学出版社, 2012. 2
(工程硕士数学主干课程系列教材)
ISBN 978-7-5623-3595-5

I. ①最… II. ①蒋… III. ①最优化算法 - Matlab 软件 - 高等学校 - 教材
IV. O242.23-39

中国版本图书馆 CIP 数据核字(2011)第 282423 号

总发行: 华南理工大学出版社(广州五山华南理工大学 17 号楼, 邮编 510640)
营销部电话: 020-87113487 87110964 87111048(传真)
E-mail: scutcl3@scut.edu.cn <http://www.scutpress.com.cn>

责任编辑: 张颖

印刷者: 广州市穗彩彩印厂

开本: 787mm×1092mm 1/16 印张: 10 字数: 250 千

版次: 2012 年 2 月第 1 版 2012 年 2 月第 1 次印刷

印数: 1~2000 册

定 价: 18.00 元

版权所有 盗版必究

前 言

自20世纪90年代开始,教育部陆续批准在一些较高层次的工科院校设立“工程硕士”学位授权点,此举的目的是为在高新企业第一线工作的科技人员提供一个在职培训和提高的机会,从而为企业的转型升级注入活力。多年来,教育部工程硕士课程教学指导小组希望工程硕士课程的教学内容、教学方法以及考核方式等方面与工学硕士课程有所区别,特别是强调学以致用,然而,目前在大多数院校中,工程硕士的数学课程基本上都是工学硕士课程的“缩水版”,既不能较深入地介绍基本理论和方法,又不能讲解如何应用,更谈不上介绍如何借助计算机软件工具解决实际问题。针对这种现状,我们在华南理工大学研究生院的资助下,筹划编写“工程硕士数学主干课程系列教材”,力争使每门课程都能着重介绍模型的应用背景和应用实例,并介绍一种数学软件工具,力争每个实例都能以算法实现,以期全面改善教学效果,使学生能够自觉地、得心应手地运用该课程的基本理论、方法解决实际问题,大幅度提高学生运用数学工具和技术的能力与素质。

经过多年的工程硕士最优化计算课程的教学实践,我们认为,最优化计算作为工程硕士的学位课程,应该既包含最优化问题的基本理论介绍,又包含丰富的应用案例(着重数学建模和求解);在符合教学大纲要求的前提下,既可以降低工程硕士生学习这门课程的难度,又可以提高学生的学习兴趣,并且所选取的案例还具备工程和管理实践背景。对工程硕士生来说,完全掌握最优化计算方法所涉及的必要的基础理论有一定的难度,考虑到最优化问题都来自工程和管理实践,要能做到学以致用,必须注重对最优化计算方法最基本知识的掌握,并经过编程实践环节实现最优化计算方法。因此,本教材突出了最优化计算方法原理的介绍并配备了应用实例的分析、建模与用计算机求解。

本书的基本内容包括:MATLAB系统概述,最优化理论概述, MATLAB优化工具箱简介,无约束优化问题,约束优化问题,多目标规划,整数规划,最小二乘问题等。书中所介绍的最优化计算方法都给出了MATLAB的实现方案和许多优化计算案例,侧重于算法的实现。本书可作为工程硕士的最优化计算课程的教材,也可作为工学硕士同名课程的教学参考书,还可作为科技和管理工作者的参考资料。

本书的编写工作是在何春雄教授的组织 and 指导下展开的;在编写过程中参阅了许多参考文献,引用了参考文献中的部分案例和程序,在此谨向相关参

文献的作者致以诚挚的谢意。

限于编者的知识和能力，书中难免存在疏漏和失误之处，有待在教学实践中加以更正和补充，也恳请读者不吝赐教。编者的邮箱地址为 `jshhjiang@scut.edu.cn`。

编者

2011年9月于华南理工大学理学院

目 录

1	MATLAB 系统概述	1
1.1	MATLAB 简介	1
1.2	MATLAB 的基本功能	3
1.3	MATLAB 的文件操作	38
	习题 1	39
2	最优化理论概述	40
2.1	最优化问题的一般形式	40
2.2	无约束最优化算法	41
2.3	约束最优化算法	45
2.4	最优化问题建模举例	48
	习题 2	53
3	MATLAB 优化工具箱	54
3.1	MATLAB 工具箱	54
3.2	MATLAB 优化工具箱简介	55
4	无约束优化问题	66
4.1	一维优化问题	66
4.2	无约束非线性规划问题	68
4.3	应用案例分析、建模与求解	74
	习题 4	78
5	约束优化问题	79
5.1	线性规划问题	79
5.2	二次规划问题	89
5.3	一般的约束非线性最优化问题	92
5.4	“半无限”有约束的多元函数最优化	97
5.5	应用案例分析、建模与求解	102
	习题 5	107
6	多目标规划	111
6.1	数学原理与模型	111
6.2	最大最小问题	113
6.3	应用案例分析与求解	116
	习题 6	122
7	整数规划	124
7.1	整数规划的定义与特点	124

7.2	整数规划的求解	125
7.3	指派问题	131
7.4	整数规划的 MATLAB 求解	132
7.5	应用案例分析、建模与求解	137
	习题 7	140
8	最小二乘问题	142
8.1	线性约束最小二乘问题	142
8.2	非线性数据(曲线)拟合	144
8.3	非线性最小二乘	146
8.4	非负线性最小二乘	148
	习题 8	149
	参考文献	151

1 MATLAB 系统概述

1.1 MATLAB 简介

MATLAB 源于 MATrix LABoratory 一词,原意为矩阵实验室, MATLAB 的产生是与科学计算联系在一起的。20 世纪 70 年代后期,时任美国新墨西哥大学计算机科学系主任的 Cleve Moler 教授出于减轻学生编程负担的动机,为学生设计了一组调用 LINPACK 和 EISPACK 库程序的“通俗易懂”的接口,此即用 FORTRAN 语言编写的萌芽状态的 MATLAB。

经几年的校际流传,在 Little 的推动下,由 Little、Moler、Steve Bangert 合作,于 1984 年成立了 MathWorks 公司,并把 MATLAB 正式推向市场。从那时起, MATLAB 的内核采用 C 语言编写,而且除原有的数值计算功能外,还新增了数据图视功能。

MATLAB 以商品形式出现后,仅短短几年,就以其良好的开放性和运行的可靠性,使原先控制领域里的封闭式软件包(如英国的 UMIST、瑞典的 LUND 和 SIMNON、德国的 KEDDC)纷纷遭到淘汰,而以 MATLAB 为平台加以重建。MathWorks 公司正式推出 MATLAB 后,于 1993 年推出了具有划时代意义的 MATLAB 4.0 版本,4.x 版在继承和发展其原有的数值计算和图形可视能力的同时,出现了以下几个重要变化:①推出了 SIMULINK——一个交互式操作的动态系统建模、仿真、分析集成环境。②推出了 Symbolic Math Toolbox 1.0——一个以 Maple 为“引擎”的符号计算工具包;此举结束了国际上数值计算、符号计算孰优孰劣的长期争论,促成了两种计算互补发展新时代。③构作了 Notebook。之后陆续推出了几个改进和提高的版本,2004 年 9 月正式推出 MATLAB Release 14,即 MATLAB 7.0,其功能在原有的基础上又有了进一步的改进。自 2006 年开始,MathWorks 公司宣布每年更新两次版本,已经有了 MATLAB 2006a, MATLAB 2006b, MATLAB 2007a, MATLAB 2007b, …, MATLAB 2009a 等。本书的所有程序都经过 MATLAB 2009a 测试计算。

自 20 世纪 90 年代以来,美国和欧洲的各大学陆续将 MATLAB 正式列入研究生和本科生的教学计划,成为学生必须掌握的基本软件之一。在欧美大学,诸如应用代数、数理统计、自动控制、数字信号处理、模拟与数字通信、时间序列分析、动态系统仿真等课程的教材都把 MATLAB 作为内容。这几乎成了 20 世纪 90 年代的教材与旧版书籍的标志性区别。

在国际学术界, MATLAB 已经被确认为准确、可靠的科学计算标准软件。在许多国际一流学术刊物上(尤其是信息科学刊物),都可以看到 MATLAB 的应用。

1.1.1 MATLAB 语言的主要特点

1. 功能强大

MATLAB 4.0 以上(不包括 4.0 版本)的各版本,不仅在数值计算上继续保持着相对其

他同类软件的绝对优势,而且还开发了自己的符号运算功能。特别是 MATLAB 6. x 以后各版本在符号运算功能上丝毫不逊于其他各类软件,如 MathCAD、Mathematica 等软件。用户只要学会了 MATLAB 6. x,就可以很方便地处理线性代数中的矩阵计算、方程组的求解、微积分运算、多项式运算、偏微分方程求解、统计与优化等问题。

在数值计算过程中, MATLAB 中许多功能函数都带有算法的自适应能力,且算法先进,解决了用户的后顾之忧,弥补了 MATLAB 程序因非可执行文件而影响其速度的缺陷,因为在很多实际问题中,计算速度对算法的依赖程度大大高于对算法本身的依赖程度。另外, MATLAB 还提供了一套完善的图形可视化功能,为用户向别人展示自己的计算结果提供了广阔的空间。图 1-1 所示为利用 MATLAB 绘制的三维图形。

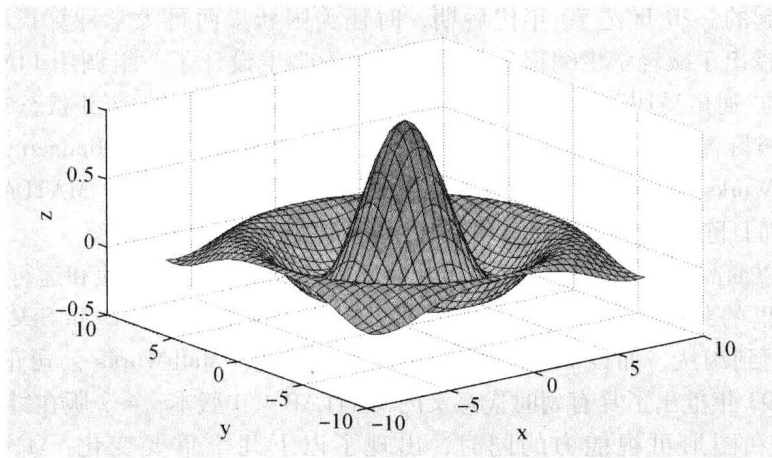


图 1-1 MATLAB 三维图形示例

2. 语言简单

如果一种语言的功能非常强大,但语言本身却晦涩难懂,那么它绝不是一种成功的语言。MATLAB 在此方面是成功的,它是解释型语言,允许用户以数学形式的语言编写程序,比 BASIC、FORTRAN、UNIX 等语言更接近于书写计算公式的思维方式,其操作和功能函数指令就是以计算机和数学书上的一些简单英文单词表达的。由于它在很长一段时间内是用 C 语言开发的,它的为数不多的几个程序流控制语句同 C 语言差别甚微,初学者很容易掌握。

MATLAB 语言的联机帮助系统也很完备,用户几乎可以不使用参考书也能方便地查询到想要的各种信息。

3. 扩充能力强,可开发性强

MATLAB 能发展到今天这个程度,它的可扩充性和可开发性起着关键的作用。MATLAB 本身就像一个解释系统,对其中的函数程序的执行以一种解释执行的方式进行。这样处理最大的好处是 MATLAB 完全成了一个开放的系统,用户可以方便地看见函数的源程序,也可以很方便地开发自己的程序,甚至可以创建自己的工具箱。

另外, MATLAB 提供了与 FORTRAN、C 等语言的接口,可以充分利用各种资源。用户只需将已有的 EXE 文件转换成 MEX 文件,就可以很方便地调用有关程序和子程序。

MATLAB 和 Maple 也有很好的接口,这大大扩充了 MATLAB 的符号运算功能。

4. 编程容易,效率高

从形式上看, MATLAB 程序文件是一个纯文本文件,扩展名为 .m。用任意字处理软件都可以对它进行编写和修改,因此程序易调试,人机交互性强。

另外, MATLAB 6. x 以后各版本还具有比较健全的调试系统,调试方便、简单,还向用户提供了旧版本所未有的、成系列的交互式工作界面。

5. 用户图形界面

与以前版本相比, MATLAB 6. x 以后各版本的图形用户界面(GUIs)的设计更加灵活,对 GUIs 编辑工具也做了相应的改进。MATLAB 通过命令窗口进行人机对话,使用方便。MATLAB 的人机界面如图 1-2 所示。

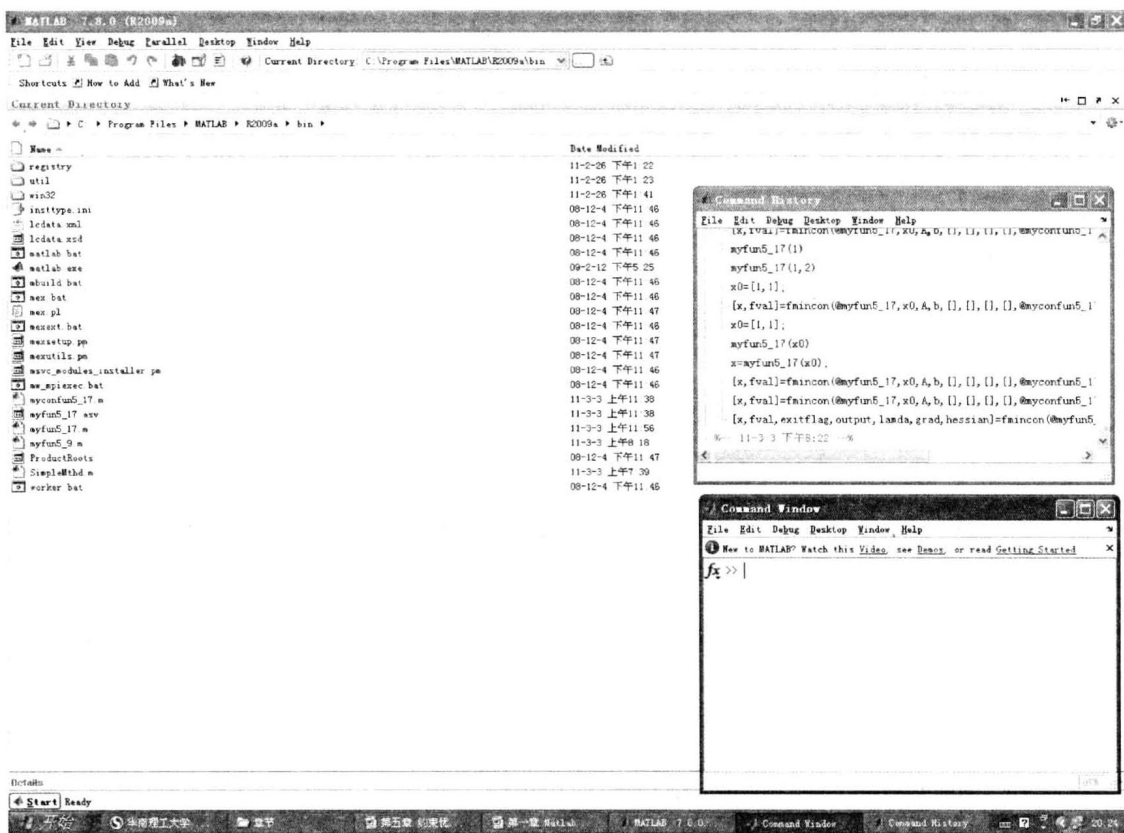


图 1-2 MATLAB 的人机界面

1.2 MATLAB 的基本功能

MATLAB 的基本功能包括数值计算功能、符号计算功能和图形处理功能。

1.2.1 数值计算功能

MATLAB 作为一种计算机语言，与其他计算机语言一样有自己的语法和约定，使用者必须遵守这些语法和约定，才能正确地使用 MATLAB。

1.2.1.1 变量与函数

变量、函数与编程所形成的 M 文件是 MATLAB 操作的基本，在介绍它们的具体使用方法之前，先介绍一些必须了解的基本规则。

1. 变量

MATLAB 和其他编程工具一样，变量是必需的基本元素，变量名是基本的标识符，以字母开头，后接字母、数字或下划线的字符序列，用法与其他语言基本一样。其具体的命名规则是：

- ①变量名必须是不含空格的单个词；
- ②变量名区分大小写；
- ③变量名最多不超过 19 个字符；
- ④变量名必须以字母开头，之后可以是任意字母、数字或下划线，变量名中不允许使用标点符号。

MATLAB 的特殊变量如表 1-1 所示。

表 1-1 MATLAB 的特殊变量

特殊变量	取 值
ans	用于结果的缺省变量名
pi	圆周率
eps	计算机的最小数，当和 1 相加就产生一个比 1 大的数
flops	浮点运算数
inf	无穷大，如 1/0
NaN	不定量，如 0/0
i, j	$i = j = \sqrt{-1}$
nargin	所有函数的输入变量数目
nargout	所有函数的输出变量数目
realmin	最小可用正实数
realmax	最大可用正实数

注意，自定义的变量名不能和表 1-1 中变量名相同。

2. 数学运算符号及标点符号

MATLAB 的主要运算符号和意义如表 1-2 所示。

表 1-2 MATLAB 的运算符

符 号	意 义
+	加法运算, 适用于两个数或两个同阶段矩阵相加
-	减法运算
*	乘法运算
.*	点乘运算
/	除法运算
./	点除运算
^	乘幂运算
.^	点乘幂运算
\	反斜杠表示左除

(1) MATLAB 的每条命令后; 若为逗号或无标点符号, 则显示命令的结果; 若命令后为分号, 则禁止显示结果;

(2) “%”后面所有文字为注释;

(3) “...”表示续行。

3. 数学函数

基本上, 常用的数学函数在 MATLAB 中都有相应的命令, 部分如表 1-3 所列。

表 1-3 MATLAB 部分数值函数

函 数	功 能	函 数	功 能
sin(x)	正弦函数	asin(x)	反正弦函数
cos(x)	余弦函数	acos(x)	反余弦函数
tan(x)	正切函数	atan(x)	反正切函数
abs(x)	绝对值或复数的模	sqrt(x)	开平方
real(x)	复数的实部	imag(x)	复数的虚部
angle(x)	复数的相角	conj(x)	复数的共轭
exp(x)	e 的指数函数	pow2(x)	2 的指数函数
log(x)	自然对数函数	log2(x)	以 2 为底的对数函数
log10(x)	以 10 为底的对数函数	sign(x)	符号函数
round(x)	四舍五入函数	fix(x)	沿原点方向舍入
floor(x)	沿负无穷大方向舍入	ceil(x)	沿正无穷大方向舍入
rat(x)	将实数 x 展开为多项分数	rats(x)	将实数 x 化为分数
rem(x, y)	x 除以 y 的余数	mod(x, y)	求模数(带符号的余数)
gcd(x, y)	整数 x 、 y 的最大公因数	lcm(x, y)	整数 x 、 y 的最小公倍数

4. M 文件

MATLAB 的内部函数是有限的, 有时为了研究某一个函数的各种性态, 需要为 MATLAB 定义新函数, 为此必须编写函数文件。函数文件是文件扩展名为 .m 的文件, 称为 M 文件, 这类文件的第一行必须是以特殊字符 function 开始, 格式为

```
function 因变量名 = 函数名(自变量名)
```

函数值的获得必须通过具体的运算实现, 并赋给因变量。

M 文件建立方法:

- ①在 MATLAB 中, 点击 File - > New - > M - file;
- ②在编辑窗口中输入程序内容;
- ③选择菜单 File - > Save, 存盘, M 文件名必须与函数名一致。

MATLAB 的应用程序也以 M 文件保存。

例如, 定义函数 $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ 。

建立 M 文件: fun.m

```
function f = fun(x)
```

```
f = 100 * (x(2) - x(1)^2)^2 + (1 - x(1))^2
```

可以直接使用函数 fun.m

例如, 计算 $f(1, 2)$, 只需在 MATLAB 命令窗口键入命令:

```
>> x = [1 2];
```

```
>> fun(x)
```

1.2.1.2 数组与矩阵

MATLAB 之所以成名, 是由于它具备了比其他软件更全面、更强大的矩阵运算功能, 因此掌握其矩阵运算功能成为学习 MATLAB 最基本的要求。MATLAB 所有的数值功能都是以矩阵为基本单元进行的, 所有的标量(整数、实数和复数)都可以看作是 1×1 矩阵, 行向量和列向量可分别看作为 $1 \times n$ 和 $n \times 1$ 矩阵。

先对向量的运算作一简单介绍。要对向量进行运算, 首先要生成向量, 生成向量最直接的方法就是在命令窗口中直接输入各分量。所有的分量用空格、逗号或分号分隔, 按次序写在中括号“[]”中, 用空格和逗号分隔生成行向量, 用分号分隔生成列向量。

矩阵是 MATLAB 最基本的数据对象, MATLAB 的大部分运算或命令都是在矩阵运算的意义下执行的。在 MATLAB 中, 不需对矩阵的维数和类型进行说明, MATLAB 会根据用户所输入的内容自动进行配置。

1. 数组

数组可以看作是只有一行或一列的简单矩阵, 但作为常用的计算单元, MATLAB 也专门为其设计了一系列命令。

(1) 创建简单的数组

$x = [1\ 3\ 5\ 7\ 9\ 11]$ 创建包含指定元素的行向量;

$x = \text{first} : \text{last}$ 创建从 first 开始, 加 1 计数, 到 last 结束的行向量;

$x = \text{first} : \text{increment} : \text{last}$ 创建从 first 开始, 加 increment 计数, 到 last 结束的行向量;

$x = \text{linspace}(\text{first}, \text{last}, n)$ 创建从 first 开始, 到 last 结束, 有 n 个元素的行向量;

$x = \text{logspace}(\text{first}, \text{last}, n)$ 创建从 first 开始, 到 last 结束, 有 n 个元素的对数分隔行

向量。

(2) 数组元素的访问

(i) 访问一个元素

$x(i)$ 表示访问数组 x 的第 i 个元素。

(ii) 访问一块元素

$x(a:b:c)$ 表示访问数组 x 的从第 a 个元素开始, 以步长为 b 到第 c 个元素(但不超过 c), b 可以为负数, b 缺省时为 1。

(iii) 直接使用元素编址序号

$x([a \ b \ c \ d])$ 表示提取数组 x 的第 a 、 b 、 c 、 d 个元素构成一个新的数组 $[x(a) \ x(b) \ x(c) \ x(d)]$ 。

(3) 数组的方向

前面例子中的数组都是一行数列, 是行方向分布的, 称之为行向量。数组也可以是列向量, 它的数组操作和运算与行向量是一样的, 唯一的区别是结果以列形式显示。

产生列向量有两种方法:

直接产生 如 $c = [1; 2; 3; 4]$

转置产生 如 $b = [1 \ 2 \ 3 \ 4]; c = b'$

说明: 以空格或逗号分隔的元素指定的是不同列的元素, 而以分号分隔的元素指定了不同行的元素。

(4) 数组的运算

(i) 标量 - 数组运算

数组对标量的加、减、乘、除、乘方是数组的每个元素对该标量施加相应的加、减、乘、除、乘方运算。

设
则

$$a = [a_1, a_2, \dots, a_n], c = \text{标量}$$

$$a + c = [a_1 + c, a_2 + c, \dots, a_n + c]$$

$$a * c = [a_1 * c, a_2 * c, \dots, a_n * c]$$

$$a ./ c = [a_1/c, a_2/c, \dots, a_n/c] \text{ (右除)}$$

$$a \setminus c = [c/a_1, c/a_2, \dots, c/a_n] \text{ (左除)}$$

$$a.^c = [a_1^c, a_2^c, \dots, a_n^c]$$

$$c.^a = [c^{a_1}, c^{a_2}, \dots, c^{a_n}]$$

(ii) 数组 - 数组运算

当两个数组有相同维数时, 加、减、乘、除、幂运算可按元素对元素方式进行, 不同大小或维数的数组是不能进行运算的。

设
则

$$a = [a_1, a_2, \dots, a_n], b = [b_1, b_2, \dots, b_n]$$

$$a + b = [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n]$$

$$a * b = [a_1 * b_1, a_2 * b_2, \dots, a_n * b_n]$$

$$a ./ b = [a_1/b_1, a_2/b_2, \dots, a_n/b_n]$$

$$a \setminus b = [b_1/a_1, b_2/a_2, \dots, b_n/a_n]$$

$$a.^b = [a_1^{b_1}, a_2^{b_2}, \dots, a_n^{b_n}]$$

2. 矩阵

矩阵是 MATLAB 的主要数据类型。通过下面的介绍，可见矩阵的相关命令和数组基本类似。

(1) 矩阵的建立

逗号或空格用于分隔某一行的元素，分号用于区分不同的行。除了分号，在输入矩阵时，按 Enter 键也表示开始新一行。输入矩阵时，严格要求所有行有相同的列。

```
例如          m = [1 2 3 4 ; 5 6 7 8; 9 10 11 12]
                p = [1 1 1 1
                    2 2 2 2
                    3 3 3 3]
```

特殊矩阵的建立：

$a = []$ 产生一个空矩阵，当对一项操作无结果时，返回空矩阵，空矩阵的大小为零；

$b = \text{zeros}(m, n)$ 产生一个 m 行、 n 列的零矩阵；

$c = \text{ones}(m, n)$ 产生一个 m 行、 n 列的元素全为 1 的矩阵；

$d = \text{eye}(m, n)$ 产生一个 m 行、 n 列的单位矩阵。

(2) 矩阵中元素的操作

(i) 矩阵 A 的第 r 行： $A(r, :)$

(ii) 矩阵 A 的第 r 列： $A(:, r)$

(iii) 依次提取矩阵 A 的每一列，将 A 拉伸为一个列向量： $A(:)$

(iv) 取矩阵 A 的第 $i_1 \sim i_2$ 行、第 $j_1 \sim j_2$ 列构成新矩阵： $A(i1:i2, j1:j2)$

(v) 以逆序提取矩阵 A 的第 $i_1 \sim i_2$ 行，构成新矩阵： $A(i2:-1:i1, :)$

(vi) 以逆序提取矩阵 A 的第 $j_1 \sim j_2$ 列，构成新矩阵： $A(:, j2:-1:j1)$

(vii) 删除 A 的第 $i_1 \sim i_2$ 行，构成新矩阵： $A(i1:i2, :) = []$

(viii) 删除 A 的第 $j_1 \sim j_2$ 列，构成新矩阵： $A(:, j1:j2) = []$

(ix) 将矩阵 A 和 B 拼接成新矩阵： $[A \ B]$ ； $[A; B]$

(3) 矩阵的运算

(i) 标量 - 矩阵运算

与标量 - 数组运算相同；

(ii) 矩阵 - 矩阵运算

① 元素对元素的运算，与数组 - 数组运算相同；

② 矩阵运算：

矩阵加法： $A + B$

矩阵乘法： $A * B$

方阵的行列式： $\text{det}(A)$

方阵的逆： $\text{inv}(A)$

方阵的特征值与特征向量： $[V, D] = \text{eig}[A]$

1.2.1.3 MATLAB 编程

利用 MATLAB 的命令窗口，可以完成较为简单的运算，但遇到较为复杂的问题时，仅

靠命令窗口来解决可能会非常繁琐,这时就要考虑使用 MATLAB 的程序设计。MATLAB 提供了一个完善的程序设计语言环境,使我们能方便地编写复杂的程序,完成各种复杂的计算。MATLAB 语言在形式上与 C 语言相似,但它的编程效率比 C 语言高得多,因为用 MATLAB 语言编程,不需要事先定义变量,不需要考虑数据类型,系统会自动把所有的数据(包括标量、向量、字符、字符串等)统一处理成矩阵,矩阵的大小是根据需要动态变化的。下面主要介绍 MATLAB 类似于其他高级语言的关系运算、逻辑运算、程序的控制结构,以及 MATLAB 特有的 M 文件。

MATLAB 虽然提供了大量现成的函数供计算时使用,但当要真正解决一些实际问题时,还远远不够,编程仍然是不可或缺的一步。MATLAB 也可以像 C、FORTRAN 等高级计算机语言一样,进行程序设计。下面简单介绍 MATLAB 中一些重要的编程手段。

1. 关系与逻辑运算

除了传统的数学运算, MATLAB 支持关系和逻辑运算。一个重要的应用是控制基于真/假命题的一系列 MATLAB 命令的流程,或执行次序。

(1) 关系操作符

MATLAB 关系操作符包括所有常用的比较,如表 1-4 所示。能用来比较两个具有相同行、列的矩阵,或用来比较一个矩阵和一个标量(矩阵中每一个元素都与标量相比,结果与矩阵大小一样)。

表 1-4 MATLAB 关系操作符

关系操作符	说明
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	等于
~=	不等于

表 1-5 逻辑操作符

逻辑操作符	说明
&	与
	或
~	非

(2) 逻辑操作符

逻辑操作符提供了一种组合或否定关系表达式,具体如表 1-5 所示。

(3) 控制流

MATLAB 提供三种决策或控制流结构,即 for 循环、while 循环、if-else-end 结构。

这些结构常包含大量的 MATLAB 命令,故常出现在 MATLAB 程序中,而不是直接加在 MATLAB 提示符下。

(i) for 循环

允许一组命令以固定的和预定的次数重复,如

```
for x = array
    {commands}
end
```

在 for 和 end 语句之间的命令串 {commands} 按数组 (array) 中的每一列执行一次。在每

一次迭代中, x 被指定为数组的下一列, 即在第 n 次循环中, $x = \text{array}(:, n)$ 。

(ii) while 循环

与 for 循环以固定次数求一组命令相反, while 循环以不定的次数求一组语句的值。

```
while expression
    {commands}
end
```

只要在表达式(expression)里的所有元素为真, 就执行 while 和 end 语句之间的命令串 {commands}。

(iii) if - else - end 结构

(a) 有一个选择的一般形式

```
if expression
    {commands}
end
```

如果在表达式(expression)里的所有元素为真, 就执行 if 和 end 语句之间的命令串 {commands}。

例 1-1 设 $f(x) = \begin{cases} x^2 + 1 & \text{当 } x > 1 \\ 2x & \text{当 } x \leq 1 \end{cases}$, 求 $f(2), f(-1)$ 。

解 先建立 M 文件 fun1.m 定义函数 $f(x)$, 再在 MATLAB 命令窗口输入 fun1(2), fun1(-1) 即可。

(b) 有三个或更多选择的一般形式

```
if (expression1)
    {commands1}
else if (expression2)
    {commands2}
else if (expression3)
    {commands3}
else if .....
.....
else
    {commands}
end
end
end
.....
end
```

例 1-2 设 $f(x) = \begin{cases} x^2 + 1 & \text{当 } x > 1 \\ 2x & \text{当 } 0 < x \leq 1 \\ x^3 & \text{当 } x \leq 0 \end{cases}$, 求 $f(2), f(0.5), f(-1)$

解 先建立 M 文件 fun2.m 定义函数 $f(x)$, 再在 MATLAB 命令窗口输入 fun2(2),