

原书第3版

数值方法 (MATLAB版)

[英] 乔治·林德菲尔德 (George Lindfield) 著 李君 任明明 译
约翰·彭尼 (John Penny)

Numerical Methods Using MATLAB
Third Edition

G. R. Lindfield // J.E.T. Penny

Third Edition
Numerical Methods
using MATLAB®



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

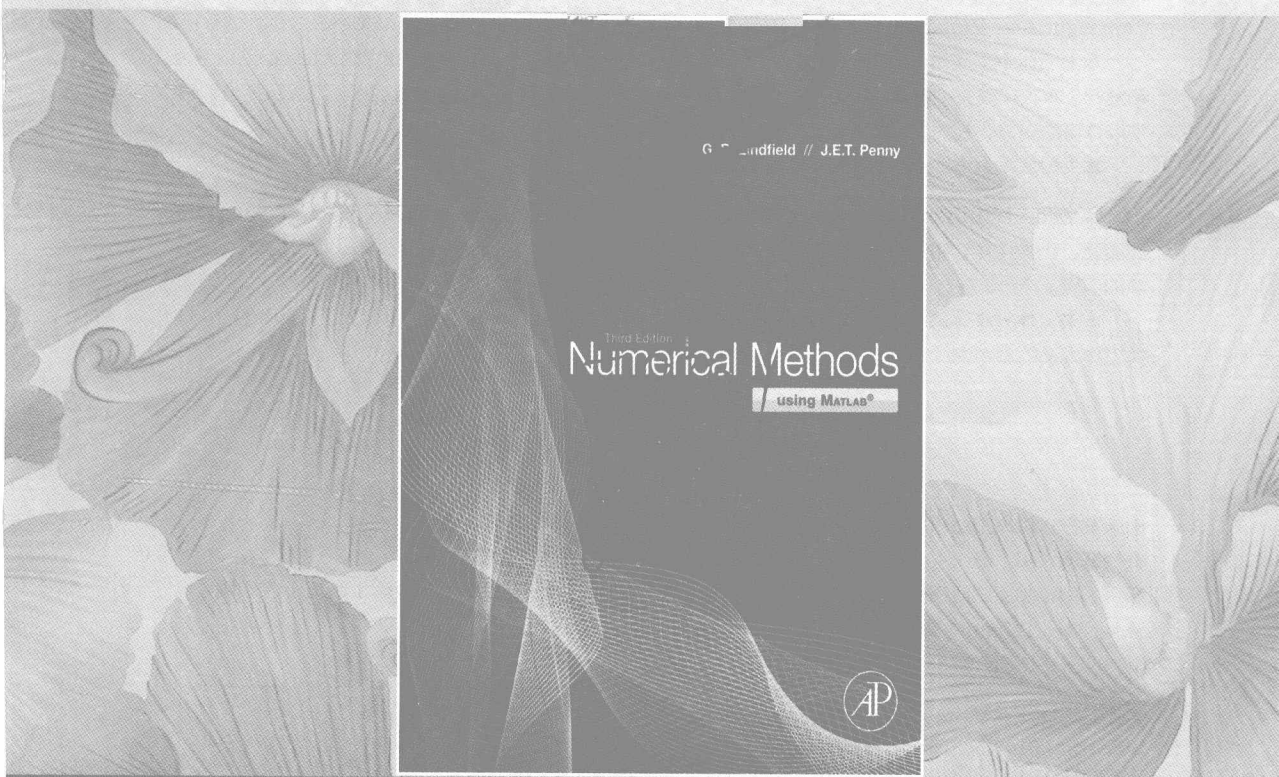
原书第3版

数值方法 (MATLAB版)

[英] 乔治·林德菲尔德 (George Lindfield) 著 李君 任明明 译
约翰·彭尼 (John Penny)

Numerical Methods Using MATLAB

Third Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

数值方法: MATLAB 版 (原书第 3 版) / (英) 林德菲尔德 (Lindfield, G.), (英) 彭尼 (Penny, J.) 著; 李君, 任明明译. —北京: 机械工业出版社, 2016.2
(计算机科学丛书)

书名原文: Numerical Methods Using MATLAB, Third Edition

ISBN 978-7-111-52429-8

I. 数… II. ①林… ②彭… ③李… ④任… III. ①电子计算机—数值计算—教材 ②计算机辅助计算—Matlab 软件—教材 IV. ① TP301.6 ② TP391.75

中国版本图书馆 CIP 数据核字 (2015) 第 313261 号

本书版权登记号: 图字: 01-2013-1802

Numerical Methods Using MATLAB, Third Edition

George Lindfield and John Penny

ISBN: 978-0-12-386942-5

Copyright © 2012 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

Copyright © 2016 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Printed in China by China Machine Press under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR, Macau SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权机械工业出版社在中国大陆境内独家出版和发行。本版仅限在中国境内 (不包括香港特别行政区、澳门特别行政区及台湾地区) 出版及标价销售。未经许可之出口, 视为违反著作权法, 将受法律之制裁。

本书封底贴有 Elsevier 防伪标签, 无标签者不得销售。

本书深入浅出地介绍数值分析, 除了以理论说明基本原理之外, 还辅以 MATLAB 程序让读者立即实验, 由做中学, 并提供习题解答以检验理解及应用是否正确, 此外还扩充方法以解决实际工程科学的问题。

本书可作为高等院校数学、计算机科学等专业本科生或研究生的教材, 也可作为工业和教育领域相关工作人员的参考书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 和 静

责任校对: 董纪丽

印 刷: 北京诚信伟业印刷有限公司

版 次: 2016 年 3 月第 1 版第 1 次印刷

开 本: 185mm × 260mm 1/16

印 张: 23

书 号: ISBN 978-7-111-52429-8

定 价: 99.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来,源远流长的科学精神和逐步形成的学术规范,使西方国家在自然科学的各个领域取得了垄断性的优势;也正是这样的优势,使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中,美国的产业界与教育界越来越紧密地结合,计算机学科中的许多泰山北斗同时身处科研和教学的最前线,由此而产生的经典科学著作,不仅擘划了研究的范畴,还揭示了学术的源变,既遵循学术规范,又自有学者个性,其价值并不会因年月的流逝而减退。

近年,在全球信息化大潮的推动下,我国的计算机产业发展迅猛,对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇,也是挑战;而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下,美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此,引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用,也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品,以“计算机科学丛书”为总称出版,供读者学习、研究及珍藏。大理石纹理的封面,也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助,国内的专家不仅提供了中肯的选题指导,还不辞劳苦地担任了翻译和审校的工作;而原书的作者也相当关注其作品在中国的传播,有的还专门为其书的中译本作序。迄今,“计算机科学丛书”已经出版了近两百个品种,这些书籍在读者中树立了良好的口碑,并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化,教育界对国外计算机教材的需求和应用都将步入一个新的阶段,我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: www.hzbook.com

电子邮件: hzjsj@hzbook.com

联系电话: (010) 88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

随着当今科技的日新月异，数值方法已经成为各个领域不可或缺的研究手段，而 MATLAB 作为世界上最先进的科学计算工具之一，以友好的界面和强大的分析、计算能力，越来越受到人们的青睐。

本书从 MATLAB 的使用入手，在简述 MATLAB 的特点和优势的基础上，系统地介绍了一些常用的数值方法，主要包括：线性方程组求解和特征值问题，非线性方程组的求解方法，数值积分和微分，微分方程的初、边值问题，数据拟合，优化以及符号计算等。其中，很多具体的实例来源于生物科学、混沌、神经网络、工程和科学等领域中的应用问题，所有的程序脚本均可在 MATLAB 7.13 中执行。

本书略去了繁琐深奥的理论证明，通过大量生动翔实的实例，演示了常用的数值方法在 MATLAB 中的实现。读者在了解掌握基本算法的同时，可以极大地激发学习研究的兴趣和热情，这无论对初学还是进一步提高都大有裨益。

本书的翻译工作由李君、任明明完成，受译者能力所限，错漏之处在所难免，敬请读者批评指正。

译者

2015 年 10 月于南开园

第3版在多方面对第1版和第2版进行了完善。我们检查和修正了所有的 MATLAB 脚本和函数，以确保它们在 MATLAB 7.13 中可以执行。

本版的主旨和之前的版本一样，即介绍大量的数值算法，解释它们的基本原理并举例说明其应用。这些算法用软件包 MATLAB 来实现，MATLAB 本身一直在不断改进，它为相关研究提供了一个有力的工具。

本书讨论了很多重要的理论结果，但目的并不是提供各领域里详细的、严谨的理论进展，而是希望展示数值方法是怎样解决诸多应用领域的问题的，同时给出了数值方法在解决具体问题时的预期理论功效。

若能谨慎使用，MATLAB 提供了一个自然简洁的方式来描述数值算法，并提供了试验这些算法的有力工具。然而，任何一个工具，不管它多么强大，都不可以轻率或不加鉴别地使用。

本书可以使得读者从数值分析的很多有趣问题中获得激励，通过系统的实验来学习数值方法。尽管 MATLAB 已经提供了许多有用的函数，但本书还要向读者介绍很多有用和重要的算法，并且开发 MATLAB 函数来实现它们。鼓励读者使用这些函数，并以数值和图形的形式产生结果。MATLAB 提供了强大多样的图形工具，这些工具可使我们对数值方法的结果有更清晰的认识。本书给出了具体的例子来说明如何利用数值方法研究生物科学、混沌、神经网络、工程和科学领域中的应用问题。

需要指出的是，我们对 MATLAB 的介绍相对简洁，只能作为读者学习的一种辅助。它无法取代标准的 MATLAB 手册或 MATLAB 软件的教科书。我们提供了宽泛的主题介绍，以 MATLAB 函数的形式开发算法，并鼓励读者去试验这些函数，为清晰起见它们已尽可能保持简单。这些函数仍可以改进，强烈建议读者开发自己特别感兴趣的部分。

除了 MATLAB 的一般介绍，本书涵盖了以下主题：线性方程组求解和特征值问题；求解非线性方程组的方法；数值积分和微分；初值和边值问题的求解；曲线拟合，包括样条函数、最小二乘法和傅里叶分析；优化课题中的内点法，非线性规划和遗传算法等。最后，我们展示了如何将符号计算与数值算法结合起来。具体到第3版，第1章中增加了一些新加到 MATLAB 中的函数的说明和示例，并举例讨论了图形操作。第4章现在包含一节洛巴托（Lobatto）积分方法和克龙罗德（Kronrod）展开的内容。第8章进行了全面修订，包括了连续遗传算法、莫勒（Moller）缩放共轭梯度法及求解约束优化问题的方法。

本书包含很多实用的例子、实践问题（其中很多是这一版中新加入的）和解决方法。我们希望书中提供了一些大家感兴趣的问题。

本书适合本科生、研究生以及工业和教育领域的人员。我们希望读者分享我们对这一研究领域的热情。对于那些还没有接触 MATLAB 的读者，本书给出了一般性介绍，其中包括大量的数值算法以及很多有用又有趣的例子和问题。

为方便本书读者，其他参考资料，包括所有 .m 文件脚本和书中所列函数，可从本书配套网站 www.elsevierdirect.com/9780123869425 获得。使用本书作为教材的老师，可以

到教科书网站 www.textbooks.elsevier.com 注册，获取本书的习题解答[⊖]。

感谢世界各地的很多读者为我们提供了有益的建议，完善了本版。也感谢同事 David Wilson 在调整 7.5、7.6 和 7.7 节时给予的大力协助。

欢迎读者指出错误或提出改进建议。此外，还要感谢 Elsevier 的主要工作人员，包括组稿编辑 Patricia Osborn、编辑项目经理 Kathryn Morrissey、出版商 Joe Hayton、编辑项目经理 Fiona Geraghty、设计师 Kristen Davis 和项目经理 Marilyn Rash。

George Lindfield 和 John Penny
伯明翰市阿斯顿大学

⊖ 关于本书教辅资源，使用教材的教师需通过爱思唯尔的教材网站（www.textbooks.elsevier.com）注册并通过审批后才能获取。具体方法如下：在 www.textbooks.elsevier.com 教材网站查找到该书后，点击“instructor manual”便可申请查看该教师手册。有任何问题，请致电 010-85208853。——编辑注

出版者的话

译者序

前 言

第 1 章 MATLAB 简介	1
1.1 MATLAB 软件包	1
1.2 MATLAB 中的矩阵和矩阵运算	2
1.3 操作矩阵的元素	4
1.4 转置矩阵	6
1.5 特殊矩阵	7
1.6 用给定元素值生成矩阵和向量	7
1.7 矩阵函数	9
1.8 用 MATLAB 运算符 “\” 做矩阵 除法	10
1.9 逐元素运算	10
1.10 标量运算及函数	11
1.11 字符串变量	14
1.12 MATLAB 中的输入/输出	17
1.13 MATLAB 中的图形操作	20
1.14 三维绘图	24
1.15 操作图形——Handle Graphics	25
1.16 MATLAB 脚本	29
1.17 MATLAB 中的用户自定义函数	34
1.18 MATLAB 中的数据结构	37
1.19 编辑 MATLAB 脚本	39
1.20 MATLAB 中的陷阱	41
1.21 MATLAB 中的快速计算	42
习题	42
第 2 章 线性方程组和特征系统	46
2.1 引言	46
2.2 线性方程组	48
2.3 求解 $Ax=b$ 的运算符 “\” 和 “/”	52
2.4 解的精度与病态性	55
2.5 初等行变换	57
2.6 用高斯消元法求解 $Ax=b$	58
2.7 LU 分解	59
2.8 楚列斯基分解	62

2.9 QR 分解	64
2.10 奇异值分解	66
2.11 伪逆	69
2.12 超定和欠定方程组	72
2.13 迭代法	78
2.14 稀疏矩阵	78
2.15 特征值问题	86
2.16 求解特征值问题的迭代法	89
2.17 MATLAB 函数 eig	92
2.18 小结	95
习题	95
第 3 章 非线性方程组的解	99
3.1 引言	99
3.2 非线性方程解的性质	100
3.3 二分法	101
3.4 迭代或不动点法	101
3.5 迭代法的收敛性	102
3.6 收敛和混沌的范围	103
3.7 牛顿法	104
3.8 施罗德法	107
3.9 数值问题	108
3.10 MATLAB 函数 fzero 和对比 研究	109
3.11 求多项式所有根的方法	110
3.12 求解非线性方程组	114
3.13 求解非线性方程组的 布罗伊登法	116
3.14 比较牛顿法和布罗伊登法	118
3.15 小结	119
习题	119
第 4 章 微分和积分	122
4.1 引言	122
4.2 数值微分	122
4.3 数值积分	125
4.4 辛普森公式	125
4.5 牛顿-科茨公式	128
4.6 龙贝格积分	129

4.7 高斯积分	131	习题	200
4.8 无穷限的积分	133	第7章 用函数拟合数据	202
4.9 高斯-切比雪夫公式	136	7.1 引言	202
4.10 高斯-洛巴托积分	137	7.2 多项式插值	202
4.11 菲隆正弦和余弦公式	139	7.3 样条函数内插	205
4.12 积分计算中的问题	143	7.4 离散数据的傅里叶分析	207
4.13 测试积分	144	7.5 多重回归: 最小二乘原则	217
4.14 累次积分	145	7.6 模型改进的诊断	219
4.15 MATLAB 函数做二重和三重 积分	148	7.7 残差分析	222
4.16 小结	149	7.8 多项式回归	225
习题	150	7.9 用一般函数拟合数据	230
第5章 微分方程的解	153	7.10 非线性最小二乘回归	231
5.1 引言	153	7.11 变换数据	233
5.2 欧拉法	154	7.12 小结	236
5.3 稳定性问题	155	习题	236
5.4 梯形法	156	第8章 优化方法	241
5.5 龙格-库塔法	158	8.1 引言	241
5.6 预测-校正法	161	8.2 线性规划问题	241
5.7 汉明法和误差估计的应用	163	8.3 单变量函数的优化	246
5.8 微分方程中误差的传播	165	8.4 共轭梯度法	248
5.9 特殊数值方法的稳定性	165	8.5 莫勒缩放共轭梯度法	252
5.10 联立的微分方程组	168	8.6 共轭梯度法解线性方程组	256
5.11 洛伦兹方程组	170	8.7 遗传算法	258
5.12 捕食者-猎物问题	171	8.8 连续遗传算法	269
5.13 微分方程应用于神经网络	172	8.9 模拟退火	273
5.14 高阶微分方程	174	8.10 带约束的非线性优化	276
5.15 刚性方程	175	8.11 顺序无约束极小化方法	279
5.16 特殊方法	177	8.12 小结	281
5.17 外插法	179	习题	281
5.18 小结	181	第9章 符号工具箱的应用	283
习题	181	9.1 符号工具箱的介绍	283
第6章 边值问题	184	9.2 符号变量和表达式	283
6.1 二阶偏微分方程的分类	184	9.3 符号计算中的变量精度计算	288
6.2 试射法	185	9.4 级数展开及求和	288
6.3 有限差分法	186	9.5 符号矩阵的操作	291
6.4 两点边值问题	187	9.6 符号法求解方程	294
6.5 抛物偏微分方程	191	9.7 特殊函数	295
6.6 双曲偏微分方程	194	9.8 符号微分	296
6.7 椭圆偏微分方程	196	9.9 符号偏微分	298
6.8 小结	200	9.10 符号积分	299
		9.11 常微分方程组的符号解	301
		9.12 拉普拉斯变换	304

9.13	Z-变换	306
9.14	傅里叶变换法	307
9.15	符号和数值处理的结合	310
9.16	小结	312
习题		312

附录 A	矩阵代数	315
附录 B	误差分析	322
部分习题解答		326
参考文献		342
索引		345

MATLAB 简介

MATLAB 是 MathWorks 公司 (www.mathworks.com) 生产的一个软件包, 它可应用于个人电脑和超级计算机系统, 包括并行计算. 本章的目标是提供 MATLAB 的有用介绍, 为将要讨论的数值方法提供充分的背景. 读者可以参照 MATLAB 手册了解该软件包的完整说明.

1.1 MATLAB 软件包

MATLAB 大概可以说是世界上最成功的商业数值分析软件包, 它的名字源于“矩阵实验室”. 它为科学、工程问题以及需要进行重要数值计算的广大领域提供了一个交互式的开发工具. 该软件包可直接执行单一语句或准备好的语句列表——称其为脚本 (script). 脚本一旦命名并保存, 即可作为一个整体被执行. 软件包最初是基于 LINPACK 和 EISPACK 项目编写的软件, 现在已经包括了 LAPACK 和 BLAS 库这些目前“最先进的”矩阵计算的数值软件. MATLAB 为用户提供了如下功能:

1. 简洁的矩阵结构运算.
2. 不断增长和发展的大量强大的内建程序.
3. 强大的二维、三维图形操作.
4. 脚本体系, 允许用户根据自己的需求开发和修改软件.
5. 称为工具箱的函数集, 它们可以添加到 MATLAB 核心中, 并被设计用于特殊应用, 如神经网络、最优化、数字信号处理及高阶谱分析.

虽然要以最高效率实现复杂的任务需要丰富的经验, 但这并不是说 MATLAB 难以使用. 一般来说 MATLAB 工作时需要使用矩形或方形数组的数据 (即矩阵), 它们的元素可能是实数也可能是复数. 一个标量也可以看成是包含单一元素的矩阵. 这是一个优雅而强大的想法, 但它也会造成用户初期使用时概念理解上的困难. 如果用户学过 C++ 或 Python 等语言, 自然会熟悉伪表达式 $A=B+C$ 这一形式, 而 MATLAB 可以立即把它解释为一个指令, 即 A 被指定为 B 和 C 中存储的数值的和. 在 MATLAB 中变量 B 和 C 可以表示数组, 所以数组 A 的每个元素就成为 B 和 C 的对应元素值的和.

有些语言或软件包与 MATLAB 有些相似之处, 包括:

APL. APL 是 A Programming Language 的首字母缩写, 即一种编程语言, 主要用于处理数组. 它包含了许多功能强大的工具, 但较罕见地使用了非标准的符号和语法, 并用这些特殊字符重新映射键盘. 这一语言对其他语言产生了重要影响, 现在已被 APL 2 取代, 至今仍在使用.

NAG 库. 这是一个非常广泛、优质的数值分析的子程序集合. MATLAB 中还有一个针对 NAG 库的工具箱.

Mathematica 和 Maple. 这两个软件包以执行复杂符号数学运算的能力而著称, 同时也能承担高精度的数值计算. 相较之下, MATLAB 更擅长强大的数值计算和矩阵操作. 当然, MATLAB 也提供了一个可选的符号工具箱, 将在第 9

章讨论。

其他软件包。如 Scilab[⊖]、Octave[⊖]（仅适用于 UNIX 平台）和 Freemate[⊖]，它们有点类似于 MATLAB，都实现了广泛的数值方法。商业上 O-Matrix[Ⓢ]常替代 MATLAB 使用。

本书撰写时 MATLAB 的最新发行版本是 7.13.0.564 (R2011b)，它可在多种平台上运行。一般来说 MathWorks 公司每半年发布 MATLAB 的一个升级版。调用 MATLAB 时，它会打开一个命令窗口；若需要也可以打开图形、编辑和帮助窗口。用户可以设计自己认为合适的 MATLAB 工作环境。MATLAB 脚本和函数通常是独立于平台的，它们可以轻易地从一个系统移植到另一个。关于安装和启动，读者可根据特定的工作环境参考 MATLAB 手册。

本书中给出的脚本和函数已在 MATLAB 7.13.0.564 (R2011b) 上做过测试。当然，大多数也可直接用于 MATLAB 的早期版本，不过可能需要修改。

本章余下部分将简要介绍 MATLAB 的一些语句和语法，而省略结构和语法的一些细节，读者可从 MATLAB 手册中获得。MATLAB 的详细描述由 Higham 和 Higham (2005) 给出。其他信息来源于 MathWorks 公司网站和维基百科。使用维基百科时需加以辨别。

2

1.2 MATLAB 中的矩阵和矩阵运算

矩阵是 MATLAB 的基础，附录 A 提供了一些宽泛简单的介绍。MATLAB 中矩阵的名称必须以字母开头，后面可以是字母或数字的任意组合，可以是大写或小写字母。请注意本书中用特殊字体来表示 MATLAB 的语句和输出，例如 disp。

在 MATLAB 中关于标量的加、减、乘、除的算术运算可用通常的方式进行，也可以直接用矩阵或数组数据。当然用户须先创建矩阵，才能使用这些矩阵的算术运算。在 MATLAB 中，有几种创建矩阵的方法，其中适合小型矩阵的最简单的方法如下所示。要指定一组值给矩阵 A，只需打开命令窗口，然后在提示符>> 后键入

```
>> A = [1 3 5; 1 0 1; 5 0 9]
```

方括号内是矩阵的元素，每行元素由至少一个空格或逗号分隔，分号(;)表示结束该行，开始另一行。按下回车键，将显示矩阵：

```
A =
     1     3     5
     1     0     1
     5     0     9
```

所有语句都将在按下回车键后执行。例如，在提示符>> 后输入 B = [1 3 51; 2 6 12; 10 7 28]并按回车，就给 B 赋了值。在命令窗口中，将矩阵相加的结果赋值给 C，只需键入 C = A+B，类似地，如果键入 C = A-B 即为矩阵相减的结果。上述两种情况的结果均在命令窗口逐行显示。需要注意的是，以分号结束的 MATLAB 语句无任何输出。

对于简单的问题，可以使用命令窗口。所谓简单，是指有限复杂度的 MATLAB 语句，当然，有限复杂度的 MATLAB 语句也可以提供某些强大的数值计算。但是，如果要按照顺序执行一个 MATLAB 语句（命令）序列，那么较合适的方法是，打开 MATLAB

⊖ www.scilab.org

⊖ www.gnu.org/software/octave

⊖ freemat.sourceforge.net

Ⓢ www.omatrix.com

编辑窗口，写入这些语句序列，创建脚本，再取一个合适的名称保存它，以备将来使用。除非在命令窗口中键入该脚本的名称并按回车键，否则程序不会执行或显示输出。

矩阵如果只有一行或者一列，则称为向量。一个行向量包含一行元素，而一个列向量包含一列元素。在数学、工程和科学领域，通常用黑体的大写字母来表示矩阵（例如 **A**），用黑体的小写字母来表示列向量（例如 **x**）。转置操作可以把一个行向量变成列向量，反之亦然，所以可以用列向量的转置来表示行向量。在数学上，通常用上标的 T 表示转置，这样一个行向量可以表示为 $\mathbf{x}^T$ 。在 MATLAB 中，用户可以直接定义向量的形式，既可以是行向量也可以是列向量。

在 MATLAB 中，对向量和矩阵进行乘法操作是非常方便的。对向量乘法，假设 **d** 和 **p** 是两个行向量，有相同的元素个数，对这两个向量做乘法可以写为 $\mathbf{x} = \mathbf{d} * \mathbf{p}'$ ，这里符号 “'” 表示把行向量 **p** 转置成一个列向量，这样才可以相乘。乘法的结果 **x** 是一个标量。也有很多人使用 “.’” 表示转置，原因将在 1.4 节中解释。

假设矩阵 **A** 和 **B** 都已赋值，那么矩阵乘法可简单地写为 $\mathbf{C} = \mathbf{A} * \mathbf{B}$ 。如果可以相乘，该语句把 **A** 乘以 **B** 的结果赋予 **C**，并打印输出 **C**。否则 MATLAB 将给出相应的错误提示。矩阵可以相乘的条件在附录 A 中给出。注意在这里符号 “*” 是必需的。

MATLAB 中一个很有用的函数是 `whos`（函数 `who` 提供相似的功能），它可以告诉用户当前工作区的内容。例如，假设前面提到的矩阵 **A**、**B** 和 **C** 都没有从内存中清除，那么将有

```
>> whos
  Name      Size      Bytes  Class
  A         3x3         72  double array
  B         3x3         72  double array
  C         3x3         72  double array
```

```
Grand total is 27 elements using 216 bytes
```

它说明 **A**、**B** 和 **C** 都是 3×3 的矩阵，以双精度数组存储。一个双精度数需要 8 个字节来存储，所以每个数组的 9 个元素共需要 72 个字节来存储。考虑下面的操作：

```
>> clear A
>> B = [ ];
>> C = zeros(4,4);

>> whos
  Name      Size      Bytes  Class
  B         0x0          0  double array
  C         4x4        128  double array
```

```
Grand total is 16 elements using 128 bytes
```

这里，从内存中清除（删掉）了矩阵 **A**，将一个空矩阵赋予 **B**，并将一个 4×4 的全 0 数组赋予 **C**。

矩阵的大小可以用函数 `size` 和 `length` 来获得：

```
>> A = zeros(4,8);
>> B = ones(7,3);
>> [p q] = size(A)
```

```
p =
```

```
4
```

```

q =
    8

>> length(A)

ans =
    8

>> L = length(B)

L =
    7

```

这里 `size` 给出了矩阵的大小，而 `length` 给出了最大维度上的元素个数。

1.3 操作矩阵的元素

在 MATLAB 中，可以单独访问矩阵的元素或者以块的方式访问矩阵的元素。例如：

```

>> X(1,3) = C(4,5)+V(9,1)
>> A(1) = B(1)+D(1)
>> C(i,j+1) = D(i,j+1)+E(i,j)

```

这些访问矩阵元素的语句都是合法的。矩阵的行和列作为完整的实体也是可以操作的，比如 `A(:,3)` 和 `B(5,:)` 分别表示 `A` 的第 3 列和 `B` 的第 5 行。如果 `B` 有 10 行 10 列，即 `B` 是一个 10×10 的矩阵，那么 `B(:,4:9)` 表示 `B` 的第 4 到 9 列，这里符号 “:” 表示所有的行。注意，在 MATLAB 中，缺省情况下矩阵下标从 1 开始，当实现某些算法时这可能会导致一些混乱。

下面的例子展示了 MATLAB 中使用下标的一些方式。首先，给一个矩阵赋值：

```

>> A = [2 3 4 5 6; -4 -5 -6 -7 -8; 3 5 7 9 1; ...
        4 6 8 10 12; -2 -3 -4 -5 -6]

```

```

A =
     2     3     4     5     6
    -4    -5    -6    -7    -8
     3     5     7     9     1
     4     6     8    10    12
    -2    -3    -4    -5    -6

```

这里使用“省略号”(...) 表示语句没有结束，下一行是这一行的继续。

执行下面的语句：

```

>> v = [1 3 5];
>> b = A(v,2)

```

将给出：

```

b =
     3
     5
    -3

```

即 `b` 由 `A` 的第 2 列的第 1、3、5 个元素组成。

执行语句：

```

>> C = A(v,:)

```

将给出：

```
C =
     2     3     4     5     6
     3     5     7     9     1
    -2    -3    -4    -5    -6
```

即 C 由 A 的第 1、3、5 行构成。

执行语句：

```
>> D = zeros(3);
>> D(:,1) = A(v,2)
```

将给出：

```
D =
     3     0     0
     5     0     0
    -3     0     0
```

这里，D 是 3×3 的零矩阵，而后将其第 1 列元素分别替换为 A 的第 2 列中第 1、3、5 个元素。

执行语句：

```
>> E = A(1:2,4:5)
```

将给出：

```
E =
     5     6
    -7    -8
```

如果用一个单独的下标来引用一个数组（方的或者长方的），对于该下标的解释如下：下标 1 对应该数组最左上的元素，下标增加的方向是对每一列从上往下，对不同列从左往右。例如，对于前面的数组 C：

```
C1 = C;
C1(1:4:15) = 10
```

```
C1 =
    10     3     4     5    10
     3    10     7     9     1
    -2    -3    10    -5    -6
```

注意这里下标增加的跨度是 4。

当操作非常大的矩阵时，常常不确定矩阵的大小。举例来说，如果要找矩阵 A 的倒数第 2 行和倒数第 1 列的元素，可以编写如下语句：

```
>> size(A)
```

```
ans =
     5     5
```

```
>> A(4,5)
```

```
ans =
    12
```

更简单的做法是使用 end：

```
>> A(end-1,end)
```

```
ans =
    12
```

函数 reshape 可对整个矩阵进行操作，如其名字所示，该函数将一个给定的矩阵重新调

整为指定大小的矩阵（要求两个矩阵有相同的元素个数）。例如，一个 3×4 的矩阵可以调整为一个 6×2 的矩阵，但是一个 3×3 的矩阵不能调整为一个 5×2 的矩阵。该函数按列取原始矩阵的元素，直到满足新矩阵的列大小，然后对下一列重复此过程。例如，对于矩阵 P：

```
>> P = C(:,1:4)

P =
     2     3     4     5
     3     5     7     9
    -2    -3    -4    -5

>> reshape(P,6,2)

ans =
     2     4
     3     7
    -2    -4
     3     5
     5     9
    -3    -5

>> s = reshape(P,1,12);
>> s(1:10)

ans =
     2     3    -2     3     5    -3     4     7    -4     5
```

1.4 转置矩阵

对矩阵可以做转置操作，以互换矩阵的行和列。在 1.2 节中简单介绍过向量的转置，在 MATLAB 中转置用符号 “'” 来表示。例如，对矩阵 A，

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

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

把 A 的转置赋予 B，语句如下：

```
>> B = A'
```

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

如果使用符号 “.’” 替代 “'”，将得到相同的结果。但是在 MATLAB 中，当 A 是复矩阵时，符号 “'” 表示复共轭转置。例如：

```
>> A = [1+2i 3+5i;4+2i 3+4i]
```

```
A =
 1.0000 + 2.0000i  3.0000 + 5.0000i
 4.0000 + 2.0000i  3.0000 + 4.0000i
```

```
>> B = A'
```

```
B =
```

```
1.0000 - 2.0000i    4.0000 - 2.0000i
3.0000 - 5.0000i    3.0000 - 4.0000i
```

为了得到转置，而非共轭转置，可以写为：

```
>> C = A.'
```

```
C =
```

```
1.0000 + 2.0000i    4.0000 + 2.0000i
3.0000 + 5.0000i    3.0000 + 4.0000i
```

1.5 特殊矩阵

有些矩阵在矩阵运算中经常出现，MATLAB 可以很容易地生成这些矩阵。几个最常用的命令，如 `ones(m,n)`、`zeros(m,n)`、`rand(m,n)`、`randn(m,n)` 和 `randi(p,m,n)`，分别生成全为 1 的矩阵、全为 0 的矩阵、由均匀分布随机数组成的矩阵、由正态分布随机数组成的矩阵以及由均匀分布随机整数组成的矩阵，矩阵维数均为 $m \times n$ 。`randi(p,m,n)` 中的 p 表示最大整数。如果只给了一个标量参数，那么这些函数将生成一个给定参数大小的方阵。函数 `eye(n)` 生成一个 $n \times n$ 的单位矩阵。函数 `eye(m,n)` 生成一个 m 行 n 列的矩阵，其对角线元素都是 1：

```
>> A = eye(3,4), B = eye(4,3)
```

```
A =
```

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

```
B =
```

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

如果需要生成一个随机矩阵 C ，而且要求 C 的大小和已存在的矩阵 A 的大小一致，则使用语句 `C = rand(size(A))`。类似地，`D = zeros(size(A))` 生成一个全 0 的矩阵，`E = ones(size(A))` 生成一个全 1 的矩阵， D 和 E 的大小都跟 A 相同。

第 2 章将介绍其他更多的特殊矩阵。

1.6 用给定元素值生成矩阵和向量

这里只讨论一些简单的例子：

`x = -8:1:8` (或 `x = -8:8`) 设置 x 为由元素 $-8, -7, \dots, 7, 8$ 构成的向量；

`y = -2:.2:2` 设置 y 为由元素 $-2, -1.8, -1.6, \dots, 1.8, 2$ 构成的向量；

`z = [1:3 4:2:8 10:0.5:11]` 设置 z 为向量 $[1 \ 2 \ 3 \ 4 \ 6 \ 8 \ 10 \ 10.5 \ 11]$ 。

函数 `linspace` 也可以用来生成向量，用户需要指定该向量的起始值、结束值以及元素个数，例如：