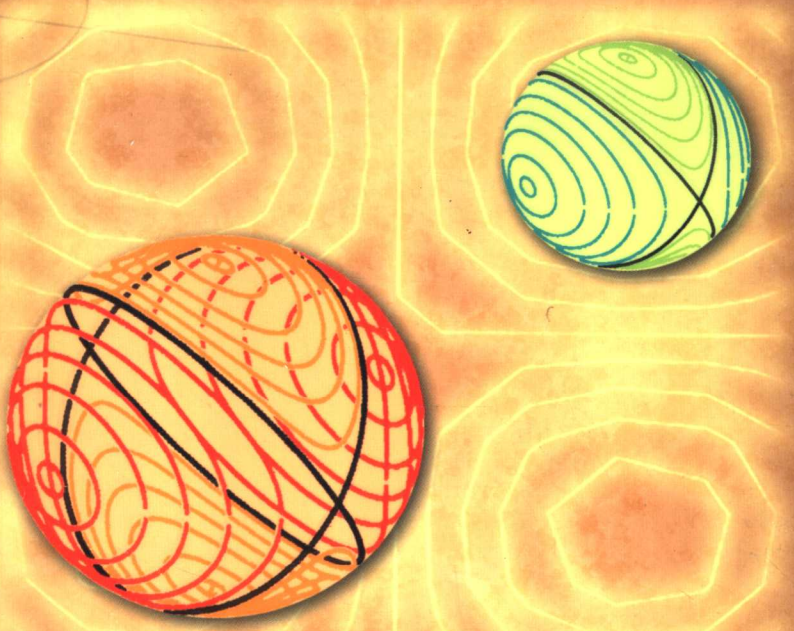
TURING 图灵数学·统计学丛书 10



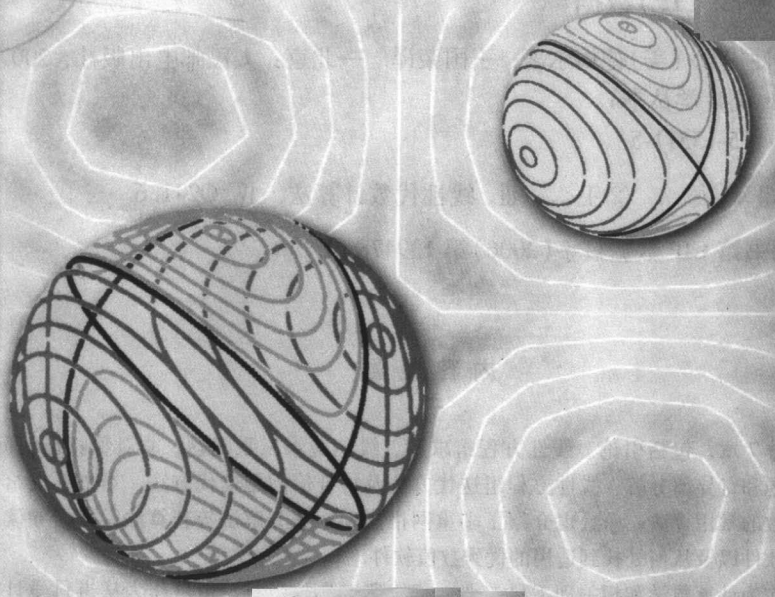
Applied Numerical Linear Algebra

应用数值线性代数

[美] James W. Demmel 著
王国荣 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵数学·统计学丛书 10



Applied Numerical Linear Algebra

应用数值线性代数

[美] James W. Demmel 著
王国荣 译

人民邮电出版社
北 京

图书在版编目(CIP)数据

应用数值线性代数/(美)德梅尔著；王国荣译. —北京：人民邮电出版社，2007.6
(图灵数学·统计学丛书)
ISBN 978-7-115-15511-5

Ⅰ. 应… Ⅱ. ①德… ②王… Ⅲ. 线性代数计算法 Ⅳ. O241.6

中国版本图书馆 CIP 数据核字(2006)第 139478 号

内 容 提 要

全书共分 7 章，包括引论、线性方程组求解、线性最小二乘问题、非对称特征值问题、对称特征问题和奇异值分解、线性方程组迭代方法及特征值问题迭代方法. 本书不仅给出了数值线性代数的常用算法，而且也介绍了多重网格法和区域分解法等新算法，并指导读者如何编写数值软件以及从何处找到适用的优秀数值软件.

本书可作为计算数学和相关理工科专业一年级研究生的教材，也可作为从事科学计算的广大科技工作者的参考书.

图灵数学·统计学丛书

应用数值线性代数

◆ 著 [美] James W Demmel
译 王国荣
责任编辑 明永玲 李 颖

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress com cn
网址 http //www ptpress com cn
北京铭成印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：700×1000 1/16
印张：22 彩插 4
字数：724 千字 2007 年 6 月第 1 版
印数：1-4 000 册 2007 年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2006-1279 号

ISBN 978-7-115-15511-5/O1

定价：49 00 元

读者服务热线：(010)88593802 印装质量热线：(010)67129223

译者介绍

王国荣 1960年上海师范学院数学系毕业后留校任教，历任上海师范大学数学科学学院院长和数学科学研究所所长，中国计算数学学会理事，中国工业与应用数学学会理事，现为上海师范大学教授，博士生导师，中国线性代数学会常务理事. 1988年和1996年两次公派赴美国Iowa大学及North Carolina州立大学作访问和合作研究. 曾3次获国家自然科学基金资助，从事广义逆理论、应用与并行算法研究. 曾获得国家教委科技进步三等奖(1994)，国务院特殊津贴(1997)，上海市优秀教育工作者(2004)，上海市科技进步三等奖(2005)，上海市级教学成果二等奖(2005).

在广义逆的扰动理论、条件数、递推算法、有限算法、嵌入算法、并行算法、Cramer法则的推广，广义逆的子式，广义逆的反序律以及算子广义逆的表示与逼近等方面取得了一系列研究成果，在国内外刊物上发表论文100篇，其中在SCI刊物LAA等上发表了25篇. 已独立或合作完成7本教材和专著的翻译及撰写工作：《矩阵计算引论》(G. W. Stewart著，王国荣等译)，《数值分析引论》(K. E. Atkinson著，匡蛟勋、王国荣等译)，《矩阵与算子广义逆》(王国荣著)，《大学数学》(一)、(二)(王国荣主编)，*Generalized Inverses: Theory and Computations*(《广义逆：理论与计算》，王国荣等)，《数值分析》(D. Kincard，W. Cheney著，王国荣等译).

译 者 序

数值线性代数是一门内容十分广泛的学科，它随着计算机和数值软件水平的不断提高发展越来越迅速. J. W. Demmel 教授所著《应用数值线性代数》(1997)是一本富有特色的、优秀的数值线性代数教材.

本书是根据作者制订的5个目标撰写的，它不仅适合计算科学和相关理工科专业一年级研究生使用，而且对广大从事科学计算的工程技术人员也有重要的参考价值；本书不仅给出线性方程组求解、线性最小二乘问题、特征值问题和奇异值分解常用的直接法和迭代法，而且介绍多重网格法、区域分解法等反映当前技术水平的新算法；此外，本书作者还强调学生不仅应该学会编写数学软件，而且能够从其他地方找出适用的优秀数值软件. 学习本书需要线性代数和程序设计方面的基础知识.

作为一本教材，本书配有大量课外作业题，涉及程序设计的问题均用“程序设计”标出，其余所有问题分为容易、中等和困难三档，便于读者深入理解和掌握有关的内容.

作者在前言中还列出本书的九大特色. 可以看出本书作者在新算法的设计和应用背景以及对各种算法的性能比较等方面都下了很大的工夫.

本人十分荣幸地应图灵公司之邀在半年时间内将本书译出，深感时间紧迫，译出后又来不及进行教学实践，因此译文中疏误和不妥之处在所难免. 敬请广大读者指正，以便再版时改正.

王国荣

2006年3月于上海师范大学

前　言

本教材包含线性方程组求解、最小二乘问题、特征问题和奇异值分解的直接法和迭代法. 本书较早的版本自 1990 年起作者在加州大学伯克利分校数学系研究生班中使用过，之前曾在柯朗研究所使用过.

在本书的写作过程中，我力求实现下列目标：

1. 教材应该吸引来自各种工科和理科的一年级研究生.

2. 教材应该是自成体系的，它仅仅假定一名优秀大学生具有线性代数方面的背景知识.

3. 学生应该学到本领域的数学基础知识以及学会如何编写数值软件或者找出优秀的数值软件.

4. 学生应该学到有效地解决实际问题的实用知识. 尤其是，即使我在本书中只作了较简单的描述，他们也应该能了解每个领域中当前最新的方法，或者何时去寻找它们以及在何处去找到它们.

5. 教材应该正好适合一个学期，因为大多数学生这门课程都是一个学期.

实际上，促使我写本书的原因是目前已有的各种教材虽然非常优秀，但是不能实现上述目标. Golub 和 Van Loan 的教材[121]是百科全书式的，然而仍旧省略了某些重要的论题，例如多重网格法、区域分解法以及特征值问题的最新算法. Watkins [252]和 Trefethen 与 Bau[243]的教材也省略了某些当前最新的算法.

虽然我相信上述 5 个目标已实现. 但是第 5 个目标最难以处理. 特别为了包含最新的研究成果以及来自同事们的要求，教材渐渐变厚而超过课时. 以本书为基础的适度的一门课程应该包含：

- 第 1 章，1.5.1 节除外；
- 第 2 章，2.2.1，2.4.3，2.5，2.6.3 和 2.6.4 节除外；
- 第 3 章，3.5 和 3.6 节除外；
- 第 4 章，直到 4.4.5 节，包括 4.4.5 节；
- 第 5 章，5.2.1，5.3.5，5.4 和 5.5 节除外；
- 第 6 章，6.3.3，6.5.5，6.5.6，6.6.6，6.7.2，6.7.3，6.7.4，6.8，6.9.2 和 6.10 节除外；
- 第 7 章，直到 7.3 节，包括 7.3 节.

本书包含下列显著特色：

- 一个课程主页，提供了教材中的例题和课外问题的 Matlab 源代码；
- 经常推荐和指出当前可利用的最佳软件(来自 LAPACK 和其他地方)；

- 讨论现代的基于高速缓冲存储器的计算机存储器是如何影响算法设计的；
- 对最小二乘问题和对称特征值问题的一些相互竞争的算法的性能进行比较；
- 讨论从雅可比法到多重网格法的各种迭代法求解正方形网格上泊松方程，并作详尽的性能比较；
- 对关于对称特征值问题的兰乔斯算法作详尽的讨论并给出数值例子；
- 从机械振动、计算几何等一些领域中提取数值例子；
- 含关于对称特征值问题和奇异值分解的“相对扰动理论”和相应的高精度算法等内容；
- 特征值算法的动力系统解释.

课程主页的 URL 为 http://www. siam. org/books/demmel/demmel_class，在本教材中简写为 HOMEPAGE. 也将使用其他两个简写的 URL. PARALLEL_HOMEPAGE 是http://www. siam. org/books/demmel/demmel_parallelclass 的简写，并指出作者在并行计算方面有关的在线的课程(on-line class). NETLIB 是 http://www. netlib. org 的简写.

课外问题按其难度用容易、中等或困难标出. 需要较多程序设计的问题用“程序设计”标出.

致谢

许多人都对本书作出了贡献. Zhaojun Bai 将本教材用于德州农机大学及肯塔基大学的教学，贡献了许多课外问题及有用的建议. Alan Edelman(用于麻省理工学院)，Martin Gutknecht(用于苏黎世理工学院)，Velvel Kahan(用于伯克利)，Richard Lehoucq，Beresford Parlett 及许多不留姓名的人给本书提供了许多建议. 表 2-2 取自我以前的学生 Xiaoye Li 的博士论文. Mark Adams，Tzu-Yi Chen，Inderjit Dhillon，Jian Xun He，Melody Ivory，Xiaoye Li，Bernd Pfrommer，Huan Ren 和 Ken Stanley 及其他许多在柯朗、伯克利、肯塔基和麻省理工的学生在这些年来，帮我改正了文中不少的错误. Bob Untiedt 和 Selene Victor 在制图和打字方面帮了很大的忙. Megan 提供了封面照片. 最后，感谢 Kathy Yelick 多年来始终不渝地支持本书的写作.

J. W. Demmel

1997 年 6 月于加利福尼亚伯克利

目　录

第 1 章　引论 …… 1
1.1　基本符号 …… 1
1.2　数值线性代数的标准问题 …… 1
1.3　一般的方法 …… 2
1.3.1　矩阵分解 …… 2
1.3.2　扰动理论和条件数 …… 3
1.3.3　舍入误差对算法的影响 …… 4
1.3.4　分析算法的速度 …… 4
1.3.5　数值计算软件 …… 5
1.4　例：多项式求值 …… 6
1.5　浮点算术运算 …… 8
1.6　再议多项式求值 …… 13
1.7　向量和矩阵范数 …… 17
1.8　第 1 章的参考书目和其他话题 …… 20
1.9　第 1 章问题 …… 21
第 2 章　线性方程组求解 …… 26
2.1　概述 …… 26
2.2　扰动理论 …… 26
2.3　高斯消元法 …… 32
2.4　误差分析 …… 38
2.4.1　选主元的必要性 …… 39
2.4.2　高斯消元法正式的误差分析 …… 40
2.4.3　估计条件数 …… 44
2.4.4　实际的误差界 …… 47
2.5　改进解的精度 …… 51
2.5.1　单精度迭代精化 …… 53
2.5.2　平衡 …… 53
2.6　高性能分块算法 …… 54
2.6.1　基本线性代数子程序（BLAS） …… 56
2.6.2　如何优化矩阵乘法 …… 57
2.6.3　使用 3 级 BLAS 改组高斯消元法 …… 62
2.6.4　更多的并行性和其他性能问题 …… 65
2.7　特殊的线性方程组 …… 66
2.7.1　实对称正定矩阵 …… 66
2.7.2　对称不定矩阵 …… 68
2.7.3　带状矩阵 …… 69
2.7.4　一般的稀疏阵 …… 72
2.7.5　不超过 $O(n^2)$ 个参数的稠密矩阵 …… 79
2.8　第 2 章的参考书目和其他的话题 …… 80
2.9　第 2 章问题 …… 80
第 3 章　线性最小二乘问题 …… 86
3.1　概述 …… 86
3.2　解线性最小二乘问题的矩阵分解 …… 89
3.2.1　正规方程 …… 89
3.2.2　QR 分解 …… 90
3.2.3　奇异值分解 …… 93
3.3　最小二乘问题的扰动理论 …… 98
3.4　正交矩阵 …… 100
3.4.1　豪斯霍尔德变换 …… 100
3.4.2　吉文斯旋转 …… 102
3.4.3　正交矩阵的舍入误差分析 …… 104
3.4.4　为什么用正交矩阵 …… 105
3.5　秩亏最小二乘问题 …… 105
3.5.1　用 SVD 解秩亏最小二乘问题 …… 107
3.5.2　用选主元的 QR 分解解秩亏最小二乘问题 …… 110

3.6 最小二乘问题解法的性能比较 …… 112
3.7 第3章的参考书目和其他话题 …… 113
3.8 第3章问题 …… 113
第4章 非对称特征值问题 …… 117
4.1 概述 …… 117
4.2 典范型 …… 117
4.3 扰动理论 …… 125
4.4 非对称特征问题的算法 …… 129
4.4.1 幂法 …… 129
4.4.2 逆迭代 …… 131
4.4.3 正交迭代 …… 132
4.4.4 QR 迭代 …… 135
4.4.5 使 QR 迭代有实效 …… 138
4.4.6 海森伯格约化 …… 139
4.4.7 三对角和双对角约化 …… 140
4.4.8 隐式位移的 QR 迭代 …… 141
4.5 其他的非对称特征值问题 …… 146
4.5.1 正则矩阵束和魏尔斯特拉斯典范型 …… 146
4.5.2 奇异矩阵束和克罗内克典范型 …… 151
4.5.3 非线性特征值问题 …… 154
4.6 小结 …… 155
4.7 第4章参考书目和其他话题 …… 157
4.8 第4章问题 …… 157
第5章 对称特征问题和奇异值分解 …… 164
5.1 概述 …… 164
5.2 扰动理论 …… 166
5.3 对称特征问题的算法 …… 177
5.3.1 三对角 QR 迭代 …… 178
5.3.2 瑞利商迭代 …… 180
5.3.3 分而治之 …… 182
5.3.4 对分法和逆迭代 …… 192
5.3.5 雅可比法 …… 195
5.3.6 性能比较 …… 199
5.4 奇异值分解算法 …… 202
5.4.1 双对角 SVD 的 QR 迭代及其变形 …… 204
5.4.2 计算双对角 SVD 达到高的相对精度 …… 207
5.4.3 SVD 的雅可比法 …… 210
5.5 微分方程和特征值问题 …… 215
5.5.1 Toda 格子 …… 216
5.5.2 与偏微分方程的关系 …… 220
5.6 第5章参考书目和其他话题 …… 221
5.7 第5章问题 …… 221
第6章 线性方程组迭代方法 …… 225
6.1 概述 …… 225
6.2 迭代法的在线(on-line)帮助 …… 225
6.3 泊松方程 …… 226
6.3.1 一维泊松方程 …… 226
6.3.2 二维泊松方程 …… 229
6.3.3 用克罗内克积表达泊松方程 …… 233
6.4 解泊松方程方法小结 …… 235
6.5 基本迭代法 …… 236
6.5.1 雅可比法 …… 238
6.5.2 高斯－塞德尔法 …… 239
6.5.3 逐次超松弛法 …… 241
6.5.4 模型问题的雅可比、高斯－塞德尔和 SOR(ω)法的收敛性 …… 242
6.5.5 雅可比、高斯－塞德尔和 SOR(ω)法明细的收敛准则 …… 243
6.5.6 切比雪夫加速和对称 SOR(SSOR) …… 250
6.6 克雷洛夫子空间方法 …… 255
6.6.1 通过矩阵－向量乘法得到关于 A 的信息 …… 256
6.6.2 利用克雷洛夫子空间 K_k 解 $Ax=b$ …… 260

6.6.3　共轭梯度法 …… 261
6.6.4　共轭梯度法的收敛性分析 …… 265
6.6.5　预条件 …… 269
6.6.6　解 $Ax=b$ 的其他克雷洛夫子空间算法 …… 271
6.7　快速傅里叶变换 …… 273
6.7.1　离散傅里叶变换 …… 275
6.7.2　用傅里叶级数解连续模型问题 …… 276
6.7.3　卷积 …… 277
6.7.4　计算快速傅里叶变换 …… 277
6.8　块循环约化 …… 279
6.9　多重网格法 …… 282
6.9.1　二维泊松方程多重网格法概述 …… 284
6.9.2　一维泊松方程的多重网格法详述 …… 287
6.10　区域分解法 …… 297
6.10.1　无交叠方法 …… 297
6.10.2　交叠方法 …… 300
6.11　第6章的参考书目和其他话题 …… 305
6.12　第6章问题 …… 305
第7章　特征值问题的迭代方法 …… 309
7.1　概述 …… 309
7.2　瑞利－里茨方法 …… 310
7.3　精确算术运算的兰乔斯算法 …… 313
7.4　浮点算术运算的兰乔斯算法 …… 318
7.5　选择正交化的兰乔斯算法 …… 323
7.6　选择正交化之外的方法 …… 324
7.7　非对称特征值问题的迭代算法 … 325
7.8　第7章的参考书目和其他话题 … 325
7.9　第7章问题 …… 325
参考文献(图灵网站下载)
索引 …… 327

第1章 引 论

1.1 基本符号

在本书中我们将经常提及矩阵、向量和标量. 矩阵用 $\boldsymbol{A}$ 这样的大写黑体字母表示，其(i,j)元素用 a_{ij}表示. 若矩阵用像 $\boldsymbol{A}+\boldsymbol{B}$ 这样的表达式给出，其(i,j)元素将记作 $(\boldsymbol{A}+\boldsymbol{B})_{ij}$. 在详细的算法描述中，有时将记作 $\boldsymbol{A}(i,j)$或利用 Matlab™ [1][184]符号 $\boldsymbol{A}(i:j,k:l)$表示 $\boldsymbol{A}$ 的位于第 i 行到第 j 行以及第 k 列到第 l 列的子阵. 像 $\boldsymbol{x}$ 这样的小写黑体字母表示一个向量，其第 i 个元素记作 x_i. 向量几乎总是列向量，它如同具有一列的矩阵. 小写的希腊字母(偶尔用小写字母)表示标量. $\mathbb{R}$ 表示实数集；$\mathbb{R}^n$ 表示 n 维实向量集；$\mathbb{R}^{m\times n}$表示 $m\times n$ 阶实阵集；$\mathbb{C}$、$\mathbb{C}^n$ 和$\mathbb{C}^{m\times n}$分别表示复数集、复向量集和复 $m\times n$ 阶阵集. 偶尔我们用简略的表达法 $\boldsymbol{A}^{m\times n}$表示 $\boldsymbol{A}$ 是 $m\times n$ 阶阵. $\boldsymbol{A}^{\mathrm{T}}$ 表示矩阵 $\boldsymbol{A}$ 的转置：$(\boldsymbol{A}^{\mathrm{T}})_{ij}=a_{ji}$. 对复矩阵也使用共轭转置 $\boldsymbol{A}^*$：$(\boldsymbol{A}^*)_{ij}=\overline{a}_{ji}$. $\Re z$ 和$\Im z$分别表示复数 z 的实部和虚部. 若 $\boldsymbol{A}$ 是 $m\times n$ 阶阵，则$|\boldsymbol{A}|$是 $\boldsymbol{A}$ 的元素的绝对值构成的 $m\times n$阶阵：$(|\boldsymbol{A}|)_{ij}=|a_{ij}|$. $|\boldsymbol{A}|\leqslant|\boldsymbol{B}|$的不等式表示：对一切 i 和 j，$|a_{ij}|\leqslant|b_{ij}|$. 我们也对向量用这个绝对值符号：$(|\boldsymbol{x}|)_i=|x_i|$. 证明的结束将用□来标记，例子的结束用◇来标记. 其他符号将在需要时介绍.

1.2 数值线性代数的标准问题

我们将考虑下列标准问题：

- 线性方程组：解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. 这里 $\boldsymbol{A}$ 是一个已知的 $n\times n$ 阶非奇异实的或复的矩阵，$\boldsymbol{b}$ 是一个已知的 n 维列向量，$\boldsymbol{x}$ 是我们要求解的 n 维列向量.
- 最小二乘问题：计算极小化 $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ 的 $\boldsymbol{x}$，这里 $\boldsymbol{A}$ 是 $m\times n$ 阶的，$\boldsymbol{b}$ 是 $m\times 1$ 阶的，$\boldsymbol{x}$ 是 $n\times 1$ 阶的，而 $\|\boldsymbol{y}\|_2\equiv\sqrt{\sum_i|y_i|^2}$称为向量 $\boldsymbol{y}$ 的 2-范数. 若 $m>n$，即方程数大于未知量的个数，这个方程组称为超定的. 此时，一般不能精确地求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. 若 $m<n$，这个方程组称为亚定的，其将有无穷多个解.
- 特征值问题：给定 $n\times n$ 阶矩阵 $\boldsymbol{A}$，求 $n\times 1$ 阶非零向量 $\boldsymbol{x}$ 和标量 λ 使得 $\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}$.

1. Matlab 是 MathWorks 股份有限公司的注册商标.

- 奇异值问题：给定 $m \times n$ 阶矩阵 $\boldsymbol{A}$，求 $n \times 1$ 阶非量向量 $\boldsymbol{x}$ 和标量 λ 使得 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$，我们将看到这个特殊类型的特征值问题是足够重要的，值得专门考虑这个问题及其算法.

之所以将上述这些标准问题选出来加以强调，是因为它们如此频繁地在工程和科学实践中出现. 本书将通过从工程、统计和其他领域中提炼出来的简单例子来说明它们. 还有很多这些标准问题的变形我们也将予以考虑，诸如广义特征值问题 $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{B}\boldsymbol{x}$（4.5 节）和“秩亏”最小二乘问题 $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$，因为 $\boldsymbol{A}$ 的列线性相关，所以其解不唯一（3.5 节）.

我们将认识到利用问题中可能有的任何特殊结构是极其重要的. 例如，如果利用最一般形式的高斯消元法求解 $n \times n$ 线性方程组所付出的代价是 $2/3n^3$ 次浮点运算. 若方程组是对称正定的，则可以用另一个称为楚列斯基的算法而节省一半工作量. 若进一步知道矩阵是带状的有半带宽 $\sqrt{n}$（即当 $|i-j| > \sqrt{n}$ 时，$a_{ij} = 0$），则可利用带状楚列斯基（Cholesky）算法进一步减少运算次数至 $O(n^2)$ 阶. 若我们试图用 5 点差分近似求解正方形区域上的泊松方程，因其方法几乎唯一地确定矩阵，则利用多重网格算法可使运算次数减至 $O(n)$ 阶，在每个解分量正好用一个固定的工作量的意义下，这个速度几乎是最快的（6.4 节）.

1.3　一般的方法

以下是几个我们将要反复使用的一般的概念和方法：

1. 矩阵分解；
2. 扰动理论和条件数；
3. 舍入误差对算法的影响，包括浮点运算的性质；
4. 分析算法的速度；
5. 数值计算软件.

下面将对这些方法逐一作简短的讨论.

1.3.1　矩阵分解

矩阵 $\boldsymbol{A}$ 的分解是把 $\boldsymbol{A}$ 表成几个“较简单”的矩阵之积，它使所讨论的问题容易求解. 我们给出两个例子.

例 1.1　假如要求解 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. 若 $\boldsymbol{A}$ 是下三角阵，利用向前回代：

for $i = 1$ to n

$$x_i = \left(b_i - \sum_{k=1}^{i-1} a_{ik}x_k\right)/a_{ii}$$

end for

容易求解

$$\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

若 $\boldsymbol{A}$ 是上三角阵，可利用类似的向后回代思想求解. 为利用这些思想求解一般的方程组 $\boldsymbol{Ax}=\boldsymbol{b}$，需要下面的矩阵分解，它正好是高斯消元法的另一种叙述方式. ◇

定理 1.1 *若 $n\times n$ 阶矩阵 $\boldsymbol{A}$ 非奇异，则存在一个置换阵 $\boldsymbol{P}$(对单位阵作行置换后得到的矩阵)、一个非奇异下三角阵 $\boldsymbol{L}$ 和一个非奇异上三角阵 $\boldsymbol{U}$，使得 $\boldsymbol{A}=\boldsymbol{P}\cdot\boldsymbol{L}\cdot\boldsymbol{U}$. 为求解 $\boldsymbol{Ax}=\boldsymbol{b}$，我们如下求解等价的方程组 $\boldsymbol{PLUx}=\boldsymbol{b}$:*

$$\boldsymbol{LUx}=\boldsymbol{P}^{-1}\boldsymbol{b}=\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b} \quad (\text{置换 } \boldsymbol{b} \text{ 的元素}),$$

$$\boldsymbol{Ux}=\boldsymbol{L}^{-1}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b}) \quad (\text{向前回代}),$$

$$\boldsymbol{x}=\boldsymbol{U}^{-1}(\boldsymbol{L}^{-1}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{b}) \quad (\text{向后回代}).$$

我们将在 2.3 节中证明此定理.

例 1.2 *若尔当(Jordan)典范分解 $\boldsymbol{A}=\boldsymbol{VJV}^{-1}$ 显示 $\boldsymbol{A}$ 的特征值和特征向量. 这里 $\boldsymbol{V}$ 是一个非奇异阵，其列包含特征向量，而 $\boldsymbol{J}$ 是 $\boldsymbol{A}$ 的若尔当典范型，这是一个特殊的三角阵，$\boldsymbol{A}$ 的特征值在它的对角线上. 应该认识到舒尔(Schur)分解 $\boldsymbol{A}=\boldsymbol{UTU}^{*}$ 数值计算上是出众的，其中 $\boldsymbol{U}$ 是酉阵(即 $\boldsymbol{U}$ 的列是规范正交的)，$\boldsymbol{T}$ 是上三角阵，$\boldsymbol{A}$ 的特征值在其对角线上. 舒尔型 $\boldsymbol{T}$ 可以比若尔当型 $\boldsymbol{J}$ 更快和更精确地算出. 我们在 4.2 节中讨论若尔当分解和舒尔分解.* ◇

3

1.3.2 扰动理论和条件数

由数值算法产生的结果很少是完全正确的. 存在两个误差源. 首先，输入数据到算法中可能产生的误差，它由先前的计算或者多半是测量误差引起的. 其次，由于在算法之内作近似，所以存在由算法本身引起的误差. 为了估计从这两个误差源在计算的结果中的误差，需要推断当输入数据出现小扰动时，问题的解有多少改变(或*扰动*).

例 1.3 *设 $f(x)$ 是实变量 x 的一个实值可微函数. 我们要计算 $f(x)$，但不知道确切的 x. 假定代之给定 $x+\delta x$ 和 δx 的一个界，我们所能做的(在没有更多的信息时)就是计算 $f(x+\delta x)$，并试图给出绝对误差 $|f(x+\delta x)-f(x)|$ 的界. 可以利用一个简单的对 f 的线性近似得到估计 $f(x+\delta x)\approx f(x)+\delta x f'(x)$，故误差是 $|f(x+\delta x)-f(x)|\approx|\delta x|\cdot|f'(x)|$. 称 $|f'(x)|$ 为 f 在 x 上的绝对条件数. 若 $|f'(x)|$ 足够大，则即使 δx 是小的，误差可能是大的，此时，我们称 f 于 x 处*

病态. ◇

我们称之为绝对条件数是因为给定输入值的绝对变化$|\delta x|$的一个界时，它提供绝对误差$|f(x+\delta x)-f(x)|$的一个界. 通常也使用下列本质上等价的表达式来界定误差：

$$\frac{|f(x+\delta x)-f(x)|}{|f(x)|} \approx \frac{|\delta x|}{|x|} \cdot \frac{|f'(x)| \cdot |x|}{|f(x)|}.$$

这个表达式界定相对误差$|f(x+\delta x)-f(x)|/|f(x)|$针对输入时相对改变$|\delta x|/|x|$的倍数关系，因子$|f'(x)| \cdot |x|/|f(x)|$称为相对条件数，或者简称为条件数.

当需要推断输入数据中的误差是如何影响计算结果时，条件数是头等重要的. 我们用输入误差的界乘以条件数简单地去界定计算解中的误差.

4　对每个考虑的问题，我们将导出其相应的条件数.

1.3.3　舍入误差对算法的影响

为继续分析由算法本身引起的误差，需要研究算术运算中舍入误差的影响，或简称为舍入影响. 我们将对最优良的算法所拥有的性质(向后稳定性)进行讨论. 它的定义如下：

> 设$\mathrm{alg}(x)$是含有舍入影响的$f(x)$的算法. 若对一切x存在一个“小的”δx使得$\mathrm{alg}(x)=f(x+\delta x)$，则称$\mathrm{alg}(x)$为$f(x)$的*向后稳定算法*，$\delta x$称为*向后误差*. 简略地说，我们对一个有一点错误的问题$(x+\delta x)$得到一个精确的解$(f(x+\delta x))$.

这蕴含可界定误差为绝对条件数$|f'(x)|$与向后误差$|\delta x|$的值的乘积：

$$\mathrm{error}=|\mathrm{alg}(x)-f(x)|=|f(x+\delta x)-f(x)| \approx |f'(x)| \cdot |\delta x|.$$

因此，若$\mathrm{alg}(\cdot)$是向后稳定的，因$|\delta x|$总是小的，所以，除非绝对条件数$|f'(x)|$大，误差将是小的. 因而，向后稳定性是算法的一个理想的性质，而我们提出的大多数算法将总是向后稳定的. 结合相应的条件数，我们所有的计算解将有误差界.

证明一个算法是向后稳定的，需要了解机器的基本浮点运算的舍入误差，以及这些误差如何经过算法传播. 这在1.5节中讨论.

1.3.4　分析算法的速度

在选择求解一个问题的算法中，人们当然必须考虑它的运算速度(也称为性能)以及向后稳定性. 有好几种方法估计速度. 给定一个特定的问题实例、一个算法的特定的执行以及一台特定的计算机，人们当然可以简单地运行算法并看看它需花费多长时间. 这样做可能是困难的或耗费时间的，故通常需要作一个比较简单的估计. 实际上，一个特定的算法执行之前，原则上我们要估计它要多少时间.

估计一个算法所花时间通常的方式是计算算法所执行的 flops 或称浮点运算量. 我们将对给出的所有算法做此项工作. 然而，在现代计算机体系结构上这通常是一个使人误解的时间估计，因为它把计算机内部的数据转移到做乘法的地方比它实际执行乘法所用的时间可能多得多，这在并行计算机上尤为正确，而在常规的机器如工作站和个人计算机(PC)上也是正确的. 例如，在 IBM RS6000/590 工作站上 5
通过仔细地重排标准算法的运算(并利用正确的编译优化)，矩阵乘法能从 65 Mflops(每秒 100 万次浮点运算)加速到 240 Mflops，几乎快 4 倍. 我们将在 2.6 节中进一步讨论.

若算法是迭代的，即产生一系列收敛到解的近似，而不是在某一个固定的步数后停止，则我们必须问，为把误差降到可忍受的水平需要多少步数. 为此，当它收敛时需要确定是否是线性的（即在每一步用一个常数因子 $0 < c < 1$ 来控制误差，使得 $|\text{error}_i| \leqslant c\,|\text{error}_{i-1}|$ 或更快些，例如二次的($|\text{error}_i| \leqslant c\,|\text{error}_{i-1}|^2$). 若两个算法都是线性的，则可以问哪个算法有较小的常数 c. 迭代线性方程解算器及它们的收敛性分析是本书第 6 章的主题.

1.3.5 数值计算软件

在设计或选择一个数值计算软件时有三个主要问题需要考虑：易操作性、可靠性和速度. 在本书中讨论的大多数算法设计时已经仔细地考虑了这三个问题. 如果已有软件能够解决你的问题，则它容易操作的好处可能胜过任何其他考虑，诸如速度. 实际上，如果你只是偶尔利用软件来解决你的问题，则使用由专家编写的用于一般用途的软件是比较方便的，用不着自己编写更特殊的程序.

利用其他专家的软件有三种方式. 第一种方式是传统的软件库，它由求解一组固定问题(如解线性方程组、求特征值等)的子程序汇集组成. 具体而言，我们将讨论 LAPACK 库[10]，这是一组体现当前技术水平的、以 Fortran 和 C 语言编写的程序，这个库以及其他许多像它一样的库不受版权限制，是可以免费使用的，见万维网上的 NETLIB[1]. LAPACK 可靠且速度快(例如，如上所述很好地使用矩阵乘法)，但是需要注意数据结构和用户的调用次序. 在本书中对这样的软件将不断提供指示性信息.

第二种方式能提供比像 LAPACK 这样的库更加轻松的使用环境，但也失去了某些性能. 例如商用系统 Matlab[184]或其他系统. Matlab 提供一个简单的交互式的程序设计环境，其中所有的变量用矩阵表示(标量是 1×1 矩阵)，大部分的线性代数运算都设计为固定的函数是可利用的. 例如，“$\boldsymbol{C} = \boldsymbol{A} * \boldsymbol{B}$” 把矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的积存放在 $\boldsymbol{C}$ 中，而“$\boldsymbol{A}$ = inv($\boldsymbol{B}$)”把矩阵 $\boldsymbol{B}$ 的逆存放在 $\boldsymbol{A}$ 中. 在 Matlab 中快速建立原型算法以

1. 在教材中把 URL 前缀 http://www.netlib.org 简写为 NETLIB.

6 及监控它们如何工作是容易的. 但是对用户来说，因为 Matlab 自动地作了较多算法上的决定，所以它可能不如用库程序快.

第三种方式是用简单的程序块组成较复杂算法方法，也称模板. 当有大量的方法构造算法但对一个特殊的输入问题没有简单的法则选择最佳构造时，可以使用模板. 所以，许多构造必须留给用户. 这方面的一个例子可以在线性方程组解的模板：迭代法的模块[24]中找到. 类似的一组关于特征问题的模板目前正在构造当中.

1.4 例：多项式求值

我们用多项式求值的例子

$$p(x) = \sum_{i=0}^{d} a_i x^i$$

来说明扰动理论、条件数、向后稳定性和舍入误差分析的思想.

如下是多项式求值的霍纳(Horner)法则：

$p = a_d$

for $i = d-1$ down to 0

$p = x * p + a_i$

end for

对 $p(x) = (x-2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$ 应用这个法则. 在图 1-1 的底部，我们看到接近于零点 $x = 2$ 时，由霍纳法则计算的 $p(x)$ 的值是相当难以预测的，并有理由称之为“噪声”. 图 1-1 上部展示了一条准确的曲线图.

为理解此图的含义，观察一下当我们试图用基于对分法求单根的定位程序去求解 $p(x)$ 的零点时会发生什么情况. 对分法将在下面算法 1.1 中指出.

下面简要介绍一下对分法. 首先寻找一个区间 $[x_{low}, x_{high}]$，在这个区间上 $p(x)$ 同时能取到正值和负值 $[p(x_{low}) \cdot p(x_{high}) < 0]$，所以 $p(x)$ 在该区间上必有一个零点. 然后对区间中点 $x_{mid} = (x_{low} + x_{high})/2$ 计算 $p(x_{mid})$，并观察在下半区间 $[x_{low}, x_{mid}]$ 中或者在上半区间 $[x_{mid}, x_{high}]$ 中 $p(x)$ 是否变号. 任何一种情形，我们都可找到一个包含 $p(x)$ 一个零点且长度为原区间一半的新区间. 如此继续对分直到区间如要求的那样短.

决定选择上半区间或下半区间与 $p(x_{mid})$ 的符号有关. 考察图 1-1 底部 $p(x)$ 的
7 图像，可以看到当 x 变化时，其符号迅速地从正变到负，故 x_{low} 和 x_{high} 稍稍的改变就可能完全改变决定符号的结论以及最后的区间. 实际上这与 x_{low} 和 x_{high} 的初始选择有关，从 1.95 到 2.05 “噪声区域”内部无论何处出发算法都能收敛(见问题 1.21).

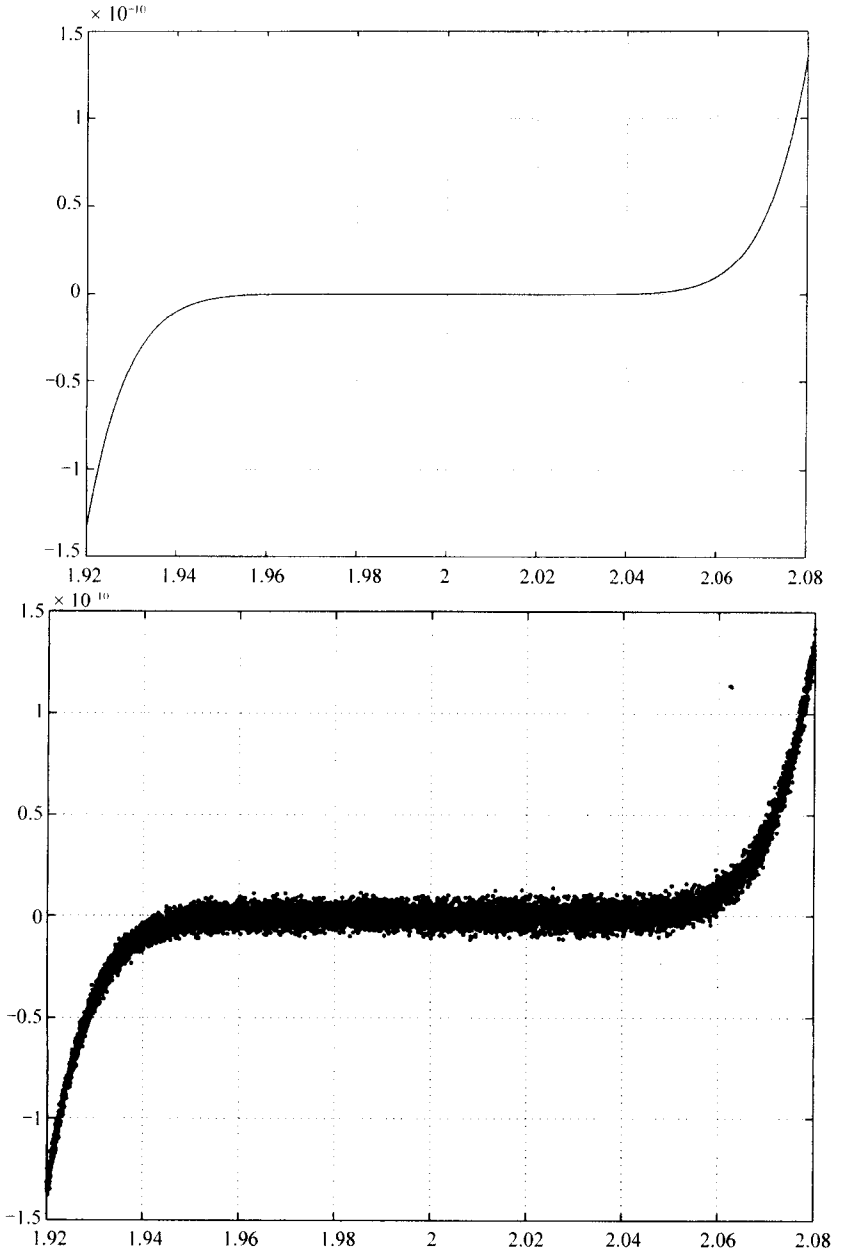


图 1-1 分别利用 $y=(x-2)^9$（上部）和霍纳法则（底部）在 8000 个等距点上求值作 $y=(x-2)^9=x^9-18x^8+144x^7-672x^6+2016x^5-4032x^4+5376x^3-4608x^2+2304x-512$ 的图形