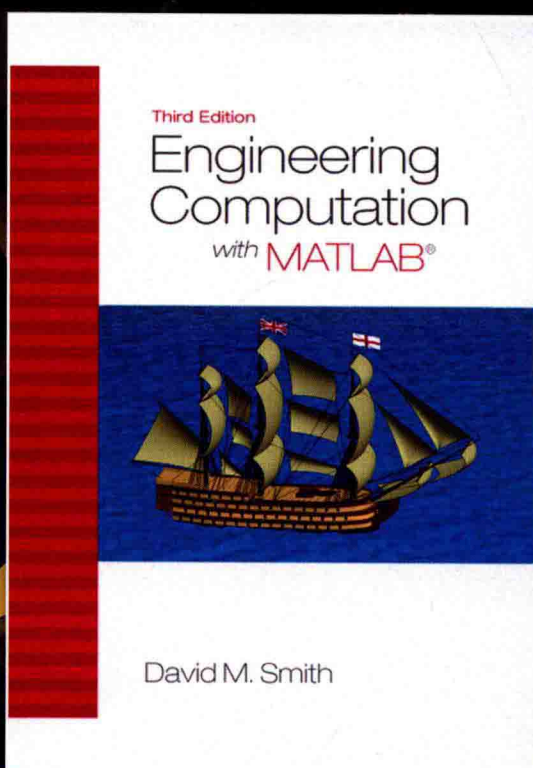


MATLAB工程计算

[美] 大卫·M. 史密斯 (David M. Smith) 著

卓金武 周英 译

Engineering Computation with MATLAB
Third Edition



算 机 科 学 丛 书

原书第3版

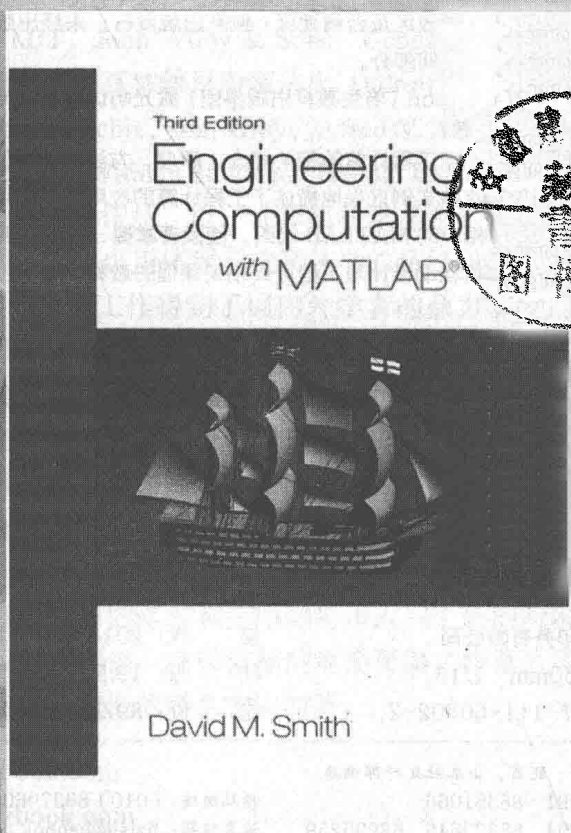
MATLAB工程计算

[美] 大卫·M. 史密斯 (David M. Smith) 著

卓金武 周英 译

Engineering Computation with MATLAB

Third Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

MATLAB 工程计算 (原书第 3 版) / (美) 大卫·M. 史密斯 (David M. Smith) 著; 卓金武, 周英译. —北京: 机械工业出版社, 2018.9

(计算机科学丛书)

书名原文: Engineering Computation with MATLAB, Third Edition

ISBN 978-7-111-60792-2

I. M… II. ①大… ②卓… ③周… III. Matlab 软件 - 高等学校 - 教材 IV. TP317

中国版本图书馆 CIP 数据核字 (2018) 第 204953 号

本书版权登记号: 图字 01-2016-3784

Authorized translation from the English language edition, entitled *Engineering Computation with MATLAB, Third Edition*, ISBN: 9780132568708 by David M. Smith, published by Pearson Education, Inc., Copyright © 2013, 2010, 2008 by Pearson Education, Inc., publishing as Addison-Wesley.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press, Copyright © 2018.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

本书基于 MATLAB 平台介绍了工程计算的相关概念、原理、方法和计算的实现过程。书中重要的知识点后都有相关的工程实例, 这些实例直观地描述了工程计算的应用场景和用法, 以加深读者对基础概念的理解; 每个实例都给出了相应的 MATLAB 代码, 使读者掌握一般的实现方法。

本书适合作为高等院校计算机辅助设计与工程计算相关课程的教材。

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: 唐晓琳

责任校对: 李秋荣

印刷: 三河市宏图印务有限公司

版次: 2018 年 9 月第 1 版第 1 次印刷

开本: 185mm×260mm 1/16

印张: 19.5

书号: ISBN 978-7-111-60792-2

定价: 89.00 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中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

理工科的知识大体可以分为三个方面：一是科学，通过理论和实验发掘新的知识；二是技术，以成熟科学理论为基础，为科学付诸实践提供辅助；三是工程，综合科学和技术理论，汇集各种资源，实现某一具体目标。在国内，科学技术常被人作为一个词语提起，却很少有人提起工程。工程往往目标简单而明确，具体实现起来却具有极大的复杂性和艰巨性。比如核聚变反应堆、引力波探测、量子通信，实现这些工程需要无数人才，要花费无数金钱和时间。可以说，科学的前沿往往离不开工程的身影，希格斯玻色子的发现就是欧洲的大型对撞机这一超级科学工程推动的。可见，工程与科学同等重要，甚至工程更重要，因为工程的推进会引导科学的发展方向，近代计算数学的发展明显强于理论数学的发展就是一个很好的证明。

工程的一个重要基础是计算，相比科学计算，工程计算更侧重应用，而科学计算更侧重计算背后的理论。另一方面，现在的工程（比如生产一辆汽车或制作一个机器人）都涉及复杂的工程计算，这些计算是设计和量产的基础。对于未来的工程师，学习工程计算是非常必要的，而且非常重要，这也是国外高校重视工程计算的原因，如本书的作者很早就在美国佐治亚理工学院开设这样的课程，而且大受欢迎，因为这样的课程比理论更生动、更具体、更实用。

本书的作者也是一位有着多年实际工程计算经验的工程师，所以本书的内容整体上比较接地气。具体说来，本书具有以下三方面的特色：

- 1) 知识体系全面。本书涵盖了工程计算中所需的基础知识，包括基本的概念、原理、方法和具体实现计算的过程。
- 2) 工程实例丰富。书中重要的知识点后基本都有相关的工程实例，比如航天器发射中的工程计算、土壤体积的计算等，这些实例直观地描述了工程计算的应用场景和用法，可以加深读者对基础概念的理解。
- 3) 详细的 MATLAB 代码。MATLAB 是工程计算中功能最强大也是应用最广的工具，全书的实例都用 MATLAB 代码作为实现过程，代码可以直接借鉴，对实际的工程计算会有很大的帮助。

此外，本书还包含了测试习题及答案等内容，对读者的自我检测很有帮助。

我们大学毕业后，从事的工作基本都涉及工程计算，深知工程计算对实际工程的重要性，尤其在 MathWorks 工作后，经常接触不同行业的工程计算问题。作为本书的译者，我们倍感荣幸，翻译的过程让我们有机会重新梳理工程中用到的计算方法，受益匪浅。所以希望读者能从本书中学习到一些实际的工程计算方法，相信你也一定会因曾经读过这样的一本书而感到幸运。一个小建议是要重视实践环节，本书中包含了很多生动的代码实例，不妨自己尝试运行一下，或者按照自己的理解重新修改一下，体验不同方法、不同代码的运行结果，相信这些尝试式的实践过程会让你加深对工程计算的理解，同时提升你的 MATLAB 功力。

本书的翻译过程中得到了机械工业出版社朱劭老师的指导和帮助，在此致以诚挚的感谢！

限于我们的知识储备，译文中难免有不当之处，烦请读者予以批评指正。

卓金武 周英

2018年5月

本书向之前没有或有很少计算经验的工科专业新生介绍计算的强大和乐趣。它是佐治亚理工学院向工程师介绍计算的入门读本。然而，它的内容已被大大地扩充了，以满足想了解编程的广大学生和教育者的需要。在本书中，为了理解计算，我们使用了 MATLAB 的基本语法和功能，这是一种用户友好的语言，正在成为工程学中最流行的计算语言之一。

第 3 版的新增内容

很多工程学科使用图的概念来表达一些具体含义。所以，这版我们增加了一章，从工程的角度讨论图处理的内容（第 17 章），着重介绍如何寻找最小生成树和寻找两点之间的最佳路径（分别介绍了精确方法和近似方法）。我们还介绍了 MATLAB 的一些新功能，学习编程入门课程的学生应该会比较感兴趣。例如，用户可以通过调整和保存值来操作绘图数据。虽然这些功能很有趣，但是通过编辑原始数据和重复绘图可以获得相同的结果，并且可跟踪性和可重复性更好。

在更新第 16 章中的排序算法时，出现了一个有趣的现象。使用老版 MATLAB（R2008 之前）时，归并排序和快速排序的递归实现没有达到预期的效果。我们推断的原因是：数据在输入和输出函数时，必须在工作空间和调用函数之间进行复制。但是，使用 R2011 时，同样的代码却运行得非常出色，这表明早期低效的参数传递机制已经得到了显著的改进。

教学方法

计算不是一门“旁观”的课程。学习计算的方法是使用计算系统来解决问题。本书不仅介绍了计算的概念及其 MATLAB 实现，还为学生提供了很多练习。本书使用工程领域的实例阐明了各种知识，提出了很多要点，还展示了学生可能遇到的一些案例。每一章都包含一些超出入门级基本内容的知识点。教师可以在两个学期慢慢地、充分地讲解本书的知识。同时，本书也为高年级学生提供了丰富的自学材料。

本书讲解各个知识点的方式如下：

- 1) 全面解释一个计算的概念。
- 2) 讨论其 MATLAB 实现。
- 3) 提供一些让学生理解这个概念的练习。

为了帮助学生理解概念及其实现，本书使用了两种方法：通用模板和 MATLAB 清单。通用模板能让学生从整体上理解概念，可以应用于任何编程语言。MATLAB 清单显示了如何在 MATLAB 中实现概念，清单后面还有对这些代码的详细解释。

本书的特点

- **练习：**让学生“自己动手”，通过使用刚刚学到的知识来熟悉概念。每个知识点都有相应的练习。
- **样式要点：**建议学生编写易于理解、调试和重用的高质量代码。

- **提示：**加深学生对知识点的理解。在学生可能从中受益的内容间穿插了提示。
- **工程实例：**提供稳健的模型，并将其应用于现实世界中的问题。每一章的最后都给出了不同工程学科的实例。
- **特殊字符、保留字和函数：**为各章讨论的关键的 MATLAB 原理提供快速参考。
- **自我测验：**帮助学生检查对各章内容的理解程度。
- **编程：**提供各种大型项目，学生可以通过这些项目来巩固自己的技能。

各章概述

第 1 章讨论计算机架构的历史及其与当今计算系统的关系。该章概述了计算机的硬件和软件，以及程序如何运行。

第 2 章讨论计算的一些基本概念，然后介绍 MATLAB 用户界面的基本操作。该章还描述了如何通过脚本的形式来获得简单的 MATLAB 程序。

第 3 章介绍 MATLAB 区别于其他编程语言的基本机制——它能对同质集合执行数学和逻辑运算。

第 4 章描述控制代码块执行的常用技术——条件运算和迭代。

第 5 章描述如何通过定义可重用代码块来实现过程抽象。

第 6 章讨论 MATLAB 如何处理包含文本的变量。

第 7 章讨论按索引和按名称访问的两种异构数据集合。

第 8 章描述 MATLAB 读写数据文件的三个层次的能力——保存工作空间，读写特定数据文件的专用工具，处理任何类型文件的通用工具。

第 9 章讨论并演示一种重复执行代码的替代方法。

第 10 章介绍帮助学生解决新问题的思路，避免“一片空白”——如何开始一个程序。

第 11 章将学生从二维基本绘图引入绘制三维物体的高级工具中，利用这些工具能绘制出平滑的阴影效果和多种发光效果。

第 12 章描述实现矩阵计算的 MATLAB 能力。

第 13 章讨论如何使用向量和数组计算来处理彩色图像。

第 14 章演示如何对声音文件进行分析、合成和操作。

第 15 章介绍工程中几种常见的数值技术：插值、曲线拟合、积分和微分。

第 16 章给出排序数据的五种算法（插入排序、冒泡排序、快速排序、归并排序和基数排序），同时介绍这些算法特定的适用环境，然后比较它们应用于大量数据时的性能。

第 17 章讨论如何表示一般的图形，然后讨论如何解决两个重要的工程问题——寻找最小生成树、寻找图上两个节点之间的最优路径。

附录总结本书用到的 MATLAB 的特殊字符、保留字、函数，介绍 ASCII 字符集和计算机内部的数字表示，给出判断题和填空题的答案。

本书的阅读顺序

不是所有涵盖编程和 MATLAB 的课程都使用相同的教学大纲。本书旨在提供适用于不同风格、不同进度课程的教学内容。例如，第 3~5 章讨论 MATLAB 的数组操作、迭代和函数编写。引入这些概念的方法有三种。第一种方法是首先介绍数组的结构，然后引入更“传统”的迭代概念；第二种方法是先讲解迭代，然后介绍 MATLAB 特有的数组操作；第三

种方法是先探讨函数。为了适应特定的教学风格，我们选择按照第一种方法来编写本书。但是，如果你更喜欢迭代或函数，可以按照自己的顺序来使用第 3~5 章。多年来，我们的课程在实践中已经变成了第三种方法，这样我们就可以使用函数接口来分隔学生的代码，以便自动进行代码分级。第 6~9 章应该按顺序讲解——各章之间存在依赖关系，改变顺序会变得很难理解。第 10 章是一个很难列入课程体系的重要章节，它在书中的位置还算比较合理。但是，在每个学期的那个时候，学生还没有准备好去思考更大的问题，所以我通常会在每学期结束时以复习的方式来讨论这些材料。第 11 章介绍了基本的绘图能力，是其余章节所需的。第 12~17 章实际上是独立的，可以按任何顺序进行教学，但是应该放在第 2~9 章和第 11 章之后。

补充材料^①

本书的各种补充材料可以从配套网站 www.pearsonhighered.com/smith 获取。所有读者均可查阅以下内容：

- 一些编程项目的解决方案。
- 一些彩色图。
- 所有 MATLAB 清单的源代码。
- 更多章节，包括面向对象编程、链表、N 叉树和图、计算成本。

此外，通过访问 www.pearsonhighered.com/irc 或发送电子邮件到 computing@aw.com，Addison-Wesley 教师资源中心的教师可以得到更多补充材料：

- 所有编程项目的解决方案。
- PowerPoint 幻灯片。

致谢

本书的基本理念和形成其框架的材料来源于 Russell Shackelford 教授 1996 年左右的工作。Melody Moore 博士是佐治亚理工学院交互式计算系的副教授，他帮忙创建了许多教学材料（并被制作成幻灯片），这门课第一次使用的就是这些教学材料。非常感谢佐治亚理工学院航空航天工程系的 James Craig 教授，他和我一起教了 CS1 的第一个工程版本，还告诉我很多关于 MATLAB 的知识，我们一起开创了这门课，从最初的 35 名学生发展到现在的每学期超过 1000 名学生。这门课成了一个向学生介绍 MATLAB 语言的平台。

我要感谢以下审稿人在本书编写期间给予的帮助和意见：

Kenneth Rouse，奥本大学

Suparna Datta，东北大学

Gerardine G. Botte，俄亥俄大学

Mica Grujicic，克莱姆森大学

Kuldip S. Rattan，莱特州立大学

Y. J. Lin，阿克伦大学

Mark Nagurka，马凯特大学

① 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师请联系机械工业出版社华章公司，电话 136 0115 6823，邮箱 wanguang@hzbook.com。——编辑注

Michael Peshkin, 西北大学

Howard Silver, 菲尔莱狄更斯大学

Steve Swinnea, 德克萨斯大学奥斯汀分校

本书材料来自佐治亚理工学院每一位教 CS1 的助教、研究生、讲师和教授（人员太多，无法一一列举）的努力。特别是那些富有创造力的助教，他们为书中的例子提出了很多想法，极大地丰富了它的内容。Aaron Bobick 教授说的一段话对本书做出了重要贡献。这段话将这门课从 MATLAB 编程课的边缘拉回到以 CS 概念为基础的课程。2004 年秋，Bobick 教授和我一起教 CS1。学期初，他提出了一个非常简单的要求：如果每一堂课中明确地表达出计算概念，而不是让他和学生从教材中梳理出这些概念，那么会更容易讲授这门课程。

我无法充分地表达我对 Addison-Wesley 团队的感激之情，他们为本书的出版做了很多“幕后”工作，他们是：Michael Hirsch、Emma Snider、Yez Alayan、Kayla Smith-Tarbox、Pat Brown 和 Jeff Holcomb。我也真的很感谢 Kailash Jadli 和他的团队在设计和维护第 3 版时所做的工作。

最重要的是，我要感谢其他两个人的贡献，没有他们就没有这本书的存在。在写这本书的过程中，我的妻子和我最好的朋友 Julie 一直给我力量和鼓励。Bill Leahy 是我 1998 年春天教的第一个 CS1 班的学生，他还继续攻读了计算机科学硕士学位，现在是计算机学院的讲师。除了要感谢他对本书无数的技术贡献之外，我还要感谢他的友谊、鼓励和睿智的判断，在我编写本书的过程中，这些都给了我很大的启发。

出版者的话
译者序
前 言

第 1 章 计算机和程序开发导论.....1

1.1 背景.....	1
1.2 计算机架构的历史.....	1
1.2.1 巴贝奇差分机.....	2
1.2.2 巨人计算机.....	2
1.2.3 冯·诺依曼架构.....	2
1.3 现代计算系统.....	3
1.3.1 计算机硬件.....	3
1.3.2 计算机存储器.....	3
1.3.3 计算机软件.....	5
1.3.4 运行计算机程序.....	7
1.4 解释型程序的运行.....	7
1.5 预期结果.....	8
1.5.1 MATLAB 简介.....	8
1.5.2 学习编程概念.....	8
1.5.3 解题技巧.....	8
本章小结.....	9
自我测验.....	9

第 2 章 入门.....11

2.1 编程语言的背景.....	11
2.1.1 抽象.....	12
2.1.2 算法.....	12
2.1.3 编程范式.....	12
2.2 基本的数据处理.....	13
2.2.1 启动和退出 MATLAB.....	13
2.2.2 变量赋值.....	13
2.2.3 数据类型.....	14
2.2.4 类和对象.....	15
2.3 MATLAB 用户界面.....	16
2.3.1 命令行窗口.....	16

2.3.2 命令历史窗口.....	17
2.3.3 工作空间窗口.....	18
2.3.4 当前目录窗口.....	21
2.3.5 变量编辑器.....	21
2.3.6 图形窗口.....	21
2.3.7 编辑器窗口.....	22

2.4 脚本.....	23
2.4.1 文本文件.....	23
2.4.2 创建脚本.....	23
2.4.3 当前目录.....	24
2.4.4 运行脚本.....	24
2.4.5 脚本中的标点.....	24
2.4.6 调试脚本.....	25
2.5 工程实例——航天器发射.....	25
本章小结.....	27
特殊字符、保留字和函数.....	28
自我测验.....	28
编程.....	29

第 3 章 向量和数组.....31

3.1 概念：使用内置函数.....	31
3.2 概念：数据集合.....	31
3.2.1 数据抽象.....	31
3.2.2 同质集合.....	31
3.3 向量.....	32
3.3.1 创建向量.....	32
3.3.2 向量大小.....	33
3.3.3 索引向量.....	33
3.3.4 缩短向量.....	34
3.3.5 向量运算.....	35
3.4 工程实例——力和力矩.....	41
3.5 数组.....	41
3.5.1 数组的属性.....	42
3.5.2 创建数组.....	42
3.5.3 访问数组的元素.....	43

3.5.4 删除数组的元素	44	5.3.3 函数的保存和使用	76
3.5.5 数组运算	44	5.3.4 函数的调用	76
3.6 工程实例——计算土壤体积	50	5.3.5 可变数量的参数	77
本章小结	52	5.3.6 返回多个结果	77
特殊字符、保留字和函数	52	5.3.7 辅助函数	78
自我测验	54	5.3.8 MATLAB 函数中的封装	78
编程	54	5.3.9 全局变量	79
第 4 章 执行控制	57	5.4 工程实例——测量固体	79
4.1 概念：代码块	57	本章小结	81
4.2 条件语句的一般形式	57	特殊字符、保留字和函数	81
4.3 if 语句	58	自我测验	81
4.3.1 通用模板	58	编程	81
4.3.2 MATLAB 实现	59	第 6 章 字符串	85
4.3.3 重要的概念	60	6.1 概念：映射、类型转换、分词、 分隔	85
4.4 switch 语句	61	6.2 MATLAB 实现	86
4.4.1 通用模板	61	6.2.1 字符串的切片与连接	87
4.4.2 MATLAB 实现	62	6.2.2 算术运算和逻辑运算	87
4.5 迭代的一般形式	63	6.2.3 函数	87
4.6 for 循环	63	6.3 类型转换函数	88
4.6.1 通用模板	63	6.3.1 将数字转换为字符串	88
4.6.2 MATLAB 实现	64	6.3.2 将字符串转换为数字	88
4.6.3 索引实现方式	65	6.4 字符串的运算	90
4.6.4 退出 for 循环	65	6.4.1 简单的数据输出： disp(...) 函数	90
4.7 while 循环	66	6.4.2 复杂的数据输出	90
4.7.1 通用模板	66	6.4.3 字符串的比较	91
4.7.2 MATLAB 实现	66	6.5 字符串数组	92
4.7.3 loop-and-a-half 的实现	67	6.6 工程实例——加密	93
4.7.4 退出 while 循环	68	本章小结	95
4.8 工程实例——计算液体的体积	68	特殊字符、保留字和函数	95
本章小结	69	自我测验	96
特殊字符、保留字和函数	70	编程	96
自我测验	70	第 7 章 元胞数组和结构体	99
编程	71	7.1 概念：异构集合	99
第 5 章 函数	74	7.2 元胞数组	99
5.1 概念：抽象和封装	74	7.2.1 元胞数组的创建	99
5.2 函数的黑盒视图	74	7.2.2 元胞数组的访问	100
5.3 MATLAB 实现	75	7.2.3 元胞数组的使用	101
5.3.1 通用模板	75		
5.3.2 定义函数	75		

7.2.4 元胞数组的处理	102	9.4.2 异常处理的通用方法	134
7.3 结构体	103	9.4.3 MATLAB 实现	134
7.3.1 结构体的构造和访问	103	9.5 封装函数	136
7.3.2 构造函数	104	9.6 递归的例子	137
7.4 结构体数组	106	9.6.1 检测回文	138
7.4.1 构建结构体数组	106	9.6.2 斐波那契数列	138
7.4.2 访问结构体的元素	107	9.6.3 函数零点	140
7.4.3 操作结构体	109	9.7 工程实例——机器人手臂运动	141
7.5 工程实例——建造一个建筑物	111	本章小结	144
本章小结	114	特殊字符、保留字和函数	144
特殊字符、保留字和函数	114	自我测验	144
自我测验	114	编程	145
编程	115		
第 8 章 文件的输入和输出	118	第 10 章 解决问题的原则	147
8.1 概念：串行 I/O	118	10.1 解决简单问题	147
8.2 工作空间 I/O	119	10.2 解题步骤	148
8.3 高级 I/O 函数	119	10.3 运算概述	148
8.3.1 探索	120	10.3.1 基本算术运算	148
8.3.2 电子表格	120	10.3.2 集合的插入	148
8.3.3 带分隔符的文本文件	121	10.3.3 集合的遍历	149
8.4 低级文件 I/O	122	10.3.4 集合的创建	150
8.4.1 文件的打开与关闭	123	10.3.5 集合的映射	150
8.4.2 文本文件的读取	123	10.3.6 集合的过滤	151
8.4.3 读取文本文件的例子	123	10.3.7 集合的汇总	151
8.4.4 文本文件的写入	124	10.3.8 集合的搜索	151
8.5 工程实例——电子表格数据	125	10.3.9 集合的排序	152
本章小结	126	10.4 解决更大的问题	152
特殊字符、保留字和函数	127	10.5 工程实例——处理地缘 政治数据	152
自我测验	127	本章小结	156
编程	127	自我测验	157
		编程	157
第 9 章 递归	130	第 11 章 绘图	159
9.1 概念：活动栈	130	11.1 基本绘图	159
9.1.1 栈	130	11.1.1 窗口——图形的“容器”	159
9.1.2 活动栈	130	11.1.2 调整图形的简单函数	159
9.1.3 函数实例	131	11.1.3 一个窗口中的多个图形—— 子图	160
9.2 递归的定义	131	11.1.4 手动编辑图形	162
9.3 递归函数的实现	132	11.2 二维绘图	162
9.4 异常	133		
9.4.1 早期的方法	133		

11.2.1 简单图形	162	本章小结	196
11.2.2 绘图选项	163	特殊字符、保留字和函数	196
11.2.3 参数图	163	自我测验	196
11.2.4 其他二维绘图功能	164	编程	196
11.3 三维绘图	164	第 13 章 图像	198
11.3.1 线性三维图	165	13.1 图像的性质	198
11.3.2 三维线性参数图	166	13.2 图像类型	199
11.3.3 其他三维绘图功能	167	13.2.1 真彩色图像	199
11.4 曲面图	167	13.2.2 灰度图像	199
11.4.1 基本功能	167	13.2.3 颜色映射图像	199
11.4.2 简单的练习	168	13.2.4 更好的图像格式	200
11.4.3 三维参数曲面	171	13.3 读取、显示、编写图像	200
11.4.4 旋转体	172	13.4 图像处理	200
11.4.5 其他三维曲面绘图功能	175	13.4.1 缩放图像	201
11.4.6 组合曲面	176	13.4.2 图像掩模	201
11.5 操作绘图数据	176	13.4.3 创建万花筒	205
11.6 工程实例——地理数据的 可视化	176	13.4.4 曲面图片	206
11.6.1 数据的分析	176	13.5 工程实例——边缘检测	208
11.6.2 数据的显示	177	本章小结	210
本章小结	178	特殊字符、保留字和函数	210
特殊字符、保留字和函数	179	自我测验	211
自我测验	180	编程	211
编程	181	第 14 章 声音的处理	213
第 12 章 矩阵	183	14.1 声音的物理原理	213
12.1 概念：行为抽象	183	14.2 记录和回放	213
12.2 矩阵运算	183	14.3 实现	214
12.2.1 矩阵乘法	183	14.4 时域操作	214
12.2.2 矩阵除法	185	14.4.1 切割和连接声音	214
12.2.3 矩阵求幂	185	14.4.2 音乐的背景	216
12.3 实现	185	14.4.3 音频的修改	217
12.3.1 矩阵乘法	185	14.5 快速傅里叶变换	219
12.3.2 矩阵除法	186	14.5.1 背景	219
12.4 旋转坐标系	188	14.5.2 实现	220
12.4.1 二维旋转	188	14.5.3 简单的频谱分析	221
12.4.2 三维旋转	190	14.6 频域操作	223
12.5 联立线性方程组的求解	192	14.7 工程实例——音乐合成器	225
12.6 工程实例	194	本章小结	226
12.6.1 陶瓷的成分	194	特殊字符、保留字和函数	227
12.6.2 电路分析	195	自我测验	227

编程	227	16.4.4 快速排序	258
第 15 章 数值方法	229	16.4.5 归并排序	259
15.1 插值	229	16.4.6 基数排序	259
15.1.1 线性插值	229	本章小结	259
15.1.2 三次样条插值	231	自我测验	259
15.1.3 外推	232	第 17 章 图论的处理	260
15.2 曲线拟合	233	17.1 队列	260
15.2.1 线性回归	233	17.1.1 队列的性质	260
15.2.2 多项式回归	235	17.1.2 队列的实现	260
15.2.3 应用	236	17.1.3 优先队列	261
15.3 数值积分	237	17.1.4 队列的测试	263
15.3.1 完全积分的计算	238	17.2 图	265
15.3.2 连续函数的积分	239	17.2.1 图的例子	265
15.4 数值微分	240	17.2.2 图的处理	265
15.5 解析式的运算	242	17.2.3 图的创建	266
15.5.1 解析式的积分	242	17.2.4 图的遍历	268
15.5.2 解析式的微分	242	17.2.5 图的搜索	270
15.6 实现	242	17.3 最小生成树	270
15.7 工程实例——调整合成器的 音符	243	17.4 图的路径搜索	272
本章小结	244	17.4.1 精确算法	272
特殊字符、保留字和函数	245	17.4.2 广度优先搜索算法	272
自我测验	245	17.4.3 Dijkstra 算法	273
编程	245	17.4.4 近似算法	275
第 16 章 排序	248	17.4.5 图搜索算法的测试	276
16.1 衡量算法成本	248	17.5 工程应用	277
16.1.1 具体的大 O 的例子	248	17.5.1 简单的应用	277
16.1.2 分析复杂的算法	250	17.5.2 复杂的应用	278
16.2 数据排序算法	250	本章小结	278
16.2.1 插入排序	250	编程	278
16.2.2 冒泡排序	251	附录 A MATLAB 的特殊字符、 保留字和函数	279
16.2.3 快速排序	253	附录 B ASCII 字符集	287
16.2.4 归并排序	254	附录 C 内部数字表示	288
16.2.5 基数排序	256	附录 D 判断题和填空题答案	290
16.3 性能分析	256		
16.4 排序算法的应用	258		
16.4.1 使用 <code>sort(...)</code>	258		
16.4.2 插入排序	258		
16.4.3 冒泡排序	258		

计算机和程序开发导论

本章介绍关于计算、计算机硬件和软件概念的历史概况，包括：

- 硬件架构。
- 软件分类。
- 编程语言。
- 预期结果。

1.1 背景

取得技术上的进步需要经过如下两步：

- 一个梦想促成一个从来没有尝试过的想法。
- 工程师发现或发明一些工具让这个想法变成现实。

寻找新的软件工具因此成了工程师生活中必不可少的部分。开发这些工具的过程经常造成一些问题，而这些问题需要创造性的解决方案。世界改变的节奏在加快，并且这种现象在计算机科学中比任何其他领域更明显。仅仅几代间，计算机已经渗透到我们生活中可能想到的方方面面了，并且还没有任何迹象表明这一趋势在放慢。

本书将帮助你熟悉一个具体的程序开发工具 MATLAB，并希望指引你达到熟练应用的水平，以便你能够自信地按照自己的方式学习对你有帮助的其他编程工具。

注意：学习一种编程语言很像学习一门外语。为了在慕尼黑找到吃的，你必须能够清楚地表达你的想法以让一个德国人能理解。这里涉及的不仅仅是知道一些词汇，还要知道一些语法规则，以让这些词汇组合之后是可理解的——比如，在德语里，动词要放在句尾。

如果语言是一种完美的个性化工具，你就可以构建你自己的词汇和语法，并且无疑是对当前语言的改进——特别是英语，因为英语有太多复杂的拼写和发音规则，改进后可以更简单。然而，语言并不是个性化的工具，它是一种实际的交流工具，所以我们不能构建自己的规则，而应该遵从我们想与之交流的那些人所期望的词汇和语法规则。

同样，本书也不是关于计算机语言本质的概要，而是一本关于创造性解决问题的实用手册。要想实现创造性地解决问题，就需要按照计算机能够理解的格式表达你的解决方案。因此，你必须使用特定的词汇（即关键词）和语法。

想熟练运用计算机，对于任何编程语言，仅仅知道语法和词汇是不够的，你还必须通过“交流”来多加练习。对于外语，这意味着去这个国家旅游，让自己浸润于他们的文化中，并且要与他们交谈。对于计算机语言，这意味着要实实在在地写程序，看看程序是怎样工作的，并且要决定如何使用计算机语言的功能解决你自己的工程问题。

1.2 计算机架构的历史

本节将回溯计算架构发展的轨迹，回顾计算机硬件的基本结构，并通过关注计算机发展历程中的三个里程碑（巴贝奇差分机、巨人计算机和冯·诺依曼架构）来重点介绍计算机对

数据存储和处理的执行过程。

1.2.1 巴贝奇差分机

查尔斯·巴贝奇(1791—1871)通常被认为是现代计算机的先驱。巴贝奇差分机(一个可以在一系列数中减去相邻值的简单装置)是一个可以提高数学运算速度和可重复性的很好的计算装置。巴贝奇关注的是工程师用于开发对数和三角函数的表。在那个时代,开发这些表的唯一方法是数学家手动计算。而算法很简单,只需在相邻的数值间组合差分表,然而人类犯错误的概率却非常高。1854年,巴贝奇设计了一个可以自动制作数学函数表的装置。因为目标是产生数值表,所以装置的输出是一系列为印刷机准备的铜板。这个装置用于存储数据的内存是一组组排列在垂直柱子中的轮子,而数学运算是通过手工转动曲柄带动的棘轮完成的。

遗憾的是,当时的制造工具和可用的材料限制了他去实际建造一台这样的装置。然而,在1991年,伦敦的科学博物馆按照他当时的设计规格建造了一台这样的装置,如图1-1所示。仅仅有几个很小的改变,他们就可以使这台装置正常工作。尽管受限于它的灵活性,但这台装置依然可以计算高达13位有效数字的七次差分方程。

1.2.2 巨人计算机

巨人计算机是设计用来快速求解计算量大的复杂问题的机器。在第二次世界大战早期,英国在大西洋战役中节节失利,因为德国的潜艇不断击沉同盟军的多艘补给船。

于是,英国在布莱奇利建立了国家密码学院,目的就是破译德国在北大西洋用来与潜艇通信的密码。当时,英国人用恩尼格玛密码机——一种非常简单的设备,通过移动字母表中的字符来解密信息。然而,为了破解密码需要遍历任意的文本组合。尽管算法是已知的,但是手动求解需要太长的时间,通常破解后已经太迟了。后来,马克斯·纽曼专门为此设计了巨人计算机(如图1-2所示)。尽管巨人计算机不是通用的处理器,但已经足够快了,除了特别复杂的密码,绝大多数密码都是可以破译的。遗憾的是,为了安全考虑,这台机器在战争结束后就被破坏了。然而普适计算的曙光已经出现,更通用的计算机不久就可以用了。

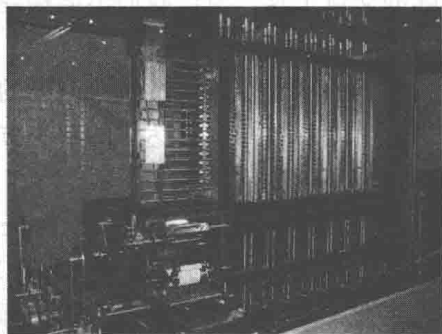


图 1-1 巴贝奇差分机

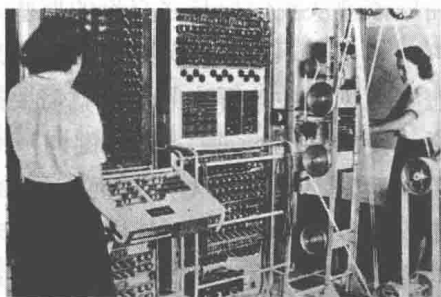


图 1-2 巨人计算机

1.2.3 冯·诺依曼架构

像巨人计算机这样的机器和那个时期其他的类似成就显示了特殊计算机在求解特定问题方面的能力。然而,冯·诺依曼发明的计算机才开辟了现代通用计算机的先河,因为它足够