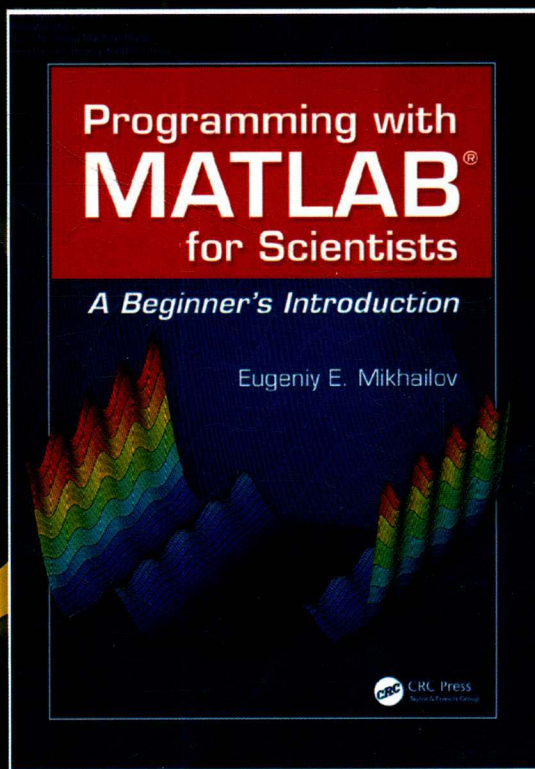


# MATLAB

# 程序设计导论

[美] 尤金尼·E. 米哈伊洛夫 (Eugeniy E. Mikhailov) 著  
于俊伟 刘楠 译

Programming with MATLAB for Scientists  
A Beginner's Introduction



计 算 机 科 学 丛 书

# MATLAB 程序设计导论

[美] 尤金尼·E. 米哈伊洛夫 (Eugeniy E. Mikhailov) 著

于俊伟 刘楠 译


Programming with MATLAB for Scientists  
A Beginner's Introduction

Programming with  
**MATLAB**<sup>®</sup>  
for Scientists  
*A Beginner's Introduction*

Eugeniy E. Mikhailov

CRC Press  
Taylor & Francis Group



 机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

MATLAB 程序设计导论 / (美) 尤金尼·E. 米哈伊洛夫 (Eugeniy E. Mikhailov) 著; 于俊伟, 刘楠译. —北京: 机械工业出版社, 2019.6

(计算机科学丛书)

书名原文: Programming with MATLAB for Scientists: A Beginner's Introduction

ISBN 978-7-111-62598-8

I. M… II. ①尤… ②于… ③刘… III. Matlab 软件—程序设计 IV. TP317

中国版本图书馆 CIP 数据核字 (2019) 第 080036 号

本书版权登记号: 图字 01-2018-6839

Programming with MATLAB for Scientists: A Beginner's Introduction by Eugeny E. Mikhailov (ISBN 978-1-4987-3828-6)

Copyright © 2017 by Taylor & Francis Group, LLC

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版, 并经授权翻译出版。版权所有, 侵权必究。

本书中文简体字翻译版授权由机械工业出版社独家出版并仅限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) 销售。未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何内容。

本书封面贴有 Taylor & Francis 公司防伪标签, 无标签者不得销售。

本书意在教你学会使用 MATLAB 语言及应知应会的数学和物理知识快速解决科学研究和工程技术中的实际问题。全书共分为三部分: 第一部分介绍了 MATLAB 基础、计算历史、编程语言及良好的编程实践; 第二部分系统地讲解了如何应用 MATLAB 解决日常问题, 主要涉及解线性代数方程组、拟合和数据约简、数值导数、数值积分方法等; 第三部分则是关于 MATLAB 的深入研究和扩展, 介绍了一些实用的工具箱, 如随机数生成器和随机过程、蒙特卡罗仿真、优化问题和常微分方程。

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: 赵亮宇

责任校对: 殷虹

印刷: 北京市荣盛彩色印刷有限公司

版次: 2019 年 6 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 12.75

书号: ISBN 978-7-111-62598-8

定价: 69.00 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 88379833

投稿热线: (010) 88379604

购书热线: (010) 68326294

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，不仅擘画了研究的范畴，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](http://www.hzbook.com)

电子邮件：[hzsj@hzbook.com](mailto:hzsj@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



## 译者序

Programming with MATLAB for Scientists: A Beginner's Introduction

MATLAB 是一种以矩阵和数组运算为基础的高级编程语言，为用户提供了简洁、高效的计算环境，深受广大科学家和工程师的信赖。MATLAB 简单易用、功能强大，针对多种科学和工程应用提供了经过专业开发和严格测试的工具箱，比如信号处理、统计分析、控制系统、最优化等工具箱。

译者多年以前就开始从事 MATLAB 学习、应用和教学，具体研究涉及车辆系统仿真、计算机视觉和深度学习等领域。正是由于对 MATLAB 编程和系统仿真的熟悉，使我们怀疑是否有必要将这本入门级的图书介绍给大家。直到了解到本书资料来源及作者简洁的写作风格，我们就迫不及待地想将它介绍给中国读者了。本书是根据美国威廉与玛丽学院(College of William & Mary)“科学家的实用计算”课程材料编写的，授课对象包括计算机科学、物理学、应用数学、化学等专业的大学生。本书内容涉及广泛，示例简洁生动，既适合没有系统编程知识的初学者学习，也适合有一定科学研究和工程技术基础的人员阅读，还可以作为高等院校本科生的学习教材。

本书分三个层次组织内容：计算基础、使用 MATLAB 求解日常生活问题、深入研究并扩展科学家的工具箱。第一部分除了介绍 MATLAB 的基础知识，还包含对计算历史和编程语言以及良好的编程实践的简要介绍。这部分内容凝聚了大学中常开设的“计算机基础”和“计算机科学导论”等课程的精华。第二部分主要涉及线性代数方程求解、数值求导、求根算法等内容，利用高等数学和线性代数等课程的基本知识，简单、快速地解决日常生活中的常见问题。最后一部分是关于 MATLAB 的深入研究和扩展，介绍了随机过程、蒙特卡罗仿真、优化问题和离散傅里叶变换等内容，当你深入到实际科学研究和工程项目中时会用到这方面的内容。

本书没有详细介绍 MATLAB 命令和工具箱，而是针对科学计算问题进行相关概念介绍，然后结合完整、简洁的代码进行讲解。本书不是教你成为优秀的计算机专家，而是利用应知应会的数学和物理知识快速解决科学研究和工程技术中的实际问题。在从事科学研究和工程技术相关工作时，导师或项目负责人往往比较看中你的“数学素养”，系统地阅读本书可以有效帮助你提高这方面的水平。千里之行，始于足下。当你掌握本书内容之后，再去尝试使用 MATLAB 的专业工具箱，甚至研究当前热门的人工智能和深度学习等内容，就会更得心应手了。

本书第 11 章由解放军信息工程大学刘楠副教授翻译，其余章节由河南工业大学信息科学与工程学院于俊伟博士翻译。由于译者水平有限，错误和疏漏在所难免，欢迎广大专家和读者提出宝贵意见。

本书的翻译工作得到国家自然科学基金项目(61300123)和河南工业大学第二批青年骨干教师培育计划项目的资助。感谢机械工业出版社华章公司对本书出版的重视,让我们有机会将本书呈现给大家,特别感谢王春华等编辑的支持和帮助,她们的工作使本书能够顺利推进。最后还要感谢家人对我们工作的支持,他们的爱和包容使我们有更多的时间投入本书的翻译工作,也使我们对正在从事的教育和科技工作更加坚定。

最后,希望我们的工作能为读者提供有益的帮助,那是我们最大的心愿!

译者

2019年1月

## 目标读者

本书适合任何想学习 MATLAB 编程的读者。如果你正在寻求关于编程、MATLAB 和数值方法的简明易懂的教程，那本书正好适合你。我们希望读者能在这里找到处理日常计算和程序设计问题的必要知识。即使是经验丰富的读者也能从书中得到对常见方法的有用见解，找到可能遇到的困惑问题的解释。

我们从简单的概念开始，帮你逐步掌握建模、模拟和分析真实系统的技能。此外，还概述了成功的科学或工程工作所必需的数值方法。本书帮你熟悉计算的“学问”，这样当你决定学习高级技术时，会知道该学习什么。

本书是根据“科学家的实用计算”(Practical Computing for Scientists)课程材料编写的，该课程为威廉与玛丽学院开设的一学期课程，教学对象为尚未确定主修学科的学生，以及物理、神经科学、生物学、计算机科学、应用数学和统计学或化学等专业的学生。成功学习这门课的学生水平也不相同，有的是大一新生，有的是大四毕业生，有的介于两者之间。

## 为什么选择 MATLAB

我们选择 MATLAB 作为编程语言，是因为 MATLAB 对一些已实现的功能有很好的平衡，这些功能对科学家来说非常重要又易于学习。MATLAB 为用户隐藏了许多底层细节，你不需要考虑变量类型、编译过程等。MATLAB 使用起来就是这么便利，它可以在不跟踪每个元素的情况下对整个数组进行计算，这正是 MATLAB 的核心。

从教师的角度来说，你不必为学生安装 MATLAB 而担心。软件安装过程很简单，学生都能独立完成。更重要的是，MATLAB 在 Windows、Mac 和 Linux 等操作系统中的界面和工作方式都一样，在不同计算机上产生的结果完全相同。

从学生的角度来说，MATLAB 可能是从事工程或科学研究工作最常用的编程语言。因此，如果你现在学习 MATLAB，或许就不需要再强迫自己学习其他行业标准编程语言了。

MATLAB 的主要缺点是价格昂贵，如果学校或单位不能提供，就得高价购买了。但这不是一个大问题，你可以选择免费的替代软件 GNU Octave。本书所有章节的练习，除了数据拟合外都可以通过 Octave 完成。Octave 中的数据拟合使用了一套不同的命令，其他部分和 MATLAB 的工作方式相同(对于一些高级选项可能需要稍作调整)。

## 本书不包含哪些内容

本书没有广泛介绍 MATLAB 命令，因为 MATLAB 已经有一个很好的手册，我们没有必要再编写一本，也无须重做一个在线教程。

本书也不能代替讲解数值方法的来龙去脉的书。我们尽可能讨论可以用数值方法完成的有趣的事情，而不必关心最有效的实现方法。然而，这本书的开头是个例外——通过数值算法的实现解释了一些编程基础知识，这些算法大多是 MATLAB 的内置函数。

## 如何阅读本书

如果你不是编程新手，可以跳过第一部分的大部分内容，但是要确保熟悉其中的元素操作、数组操作与数组元素操作之间的区别以及数组切片等内容。

如果你是科学家，那么绘图和数据拟合是必须掌握的。请务必阅读第 6 章。如果需要学习关于数据分析的重要内容，也需要学习这一章。

第三部分在某种程度上可作为自选内容，尽管我们十分推荐优化问题一章（即第 13 章）。令人惊讶的是有很多问题本质上都属于优化问题，可以用第 13 章提出的方法来解决。可能在本科高年级的课堂上才会用到这部分内容。

随着编程水平的提高，请重新阅读 4.4 节，并尝试从中学习更多技术。

## 数据文件和代码链接<sup>⊖</sup>

本书所有 MATLAB 代码及数据文件可在网站 [http://physics.wm.edu/programming\\_with\\_MATLAB\\_book](http://physics.wm.edu/programming_with_MATLAB_book) 下载。本书英文电子版中直接给出了相关文件的链接地址。

---

⊖ 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，需要的教师可到原出版社网站注册下载，若有问题，请与泰勒·弗朗西斯集团北京代表处联系，电话 010-58452806，电子邮件 [janet.zheng@tanfchina.com](mailto:janet.zheng@tanfchina.com)。——编辑注



# 目 录

Programming with MATLAB for Scientists: A Beginner's Introduction

出版者的话  
译者序  
前言

## 第一部分 计算基础

第 1 章 计算机与编程语言简介	2
1.1 早期计算史	2
1.2 现代计算机	3
1.3 什么是编程	3
1.4 编程语言概述	4
1.5 计算机中的数字表示及其 潜在问题	5
1.5.1 离散化——计算机的 主要弱点	5
1.5.2 二进制表示	6
1.5.3 浮点数表示	6
1.5.4 结论	7
1.6 自学	7
第 2 章 MATLAB 基础	9
2.1 MATLAB 的图形用户界面	9
2.2 功能强大的 MATLAB 计算器	11
2.2.1 MATLAB 的变量类型	11
2.2.2 内置函数和运算符	12
2.2.3 运算符的优先级	13
2.2.4 注释	14
2.3 高效编辑	14
2.4 使用帮助文档	15
2.5 矩阵	16
2.5.1 创建和访问矩阵元素	16

2.5.2 基本矩阵运算	17
2.5.3 字符串矩阵	20
2.6 冒号运算符	20
2.7 绘图	21
2.8 自学	23

## 第 3 章 布尔代数、条件语句和 循环

3.1 布尔代数	24
3.1.1 MATLAB 中布尔运算符 的优先级	25
3.1.2 MATLAB 布尔逻辑运算 举例	25
3.2 比较运算符	26
3.2.1 向量比较	26
3.2.2 矩阵比较	27
3.3 条件语句	27
3.3.1 if-else-end 语句	27
3.3.2 if 语句的简短形式	28
3.4 等于语句的常见错误	28
3.5 循环	28
3.5.1 while 循环	28
3.5.2 特殊命令——break 和 continue	29
3.5.3 for 循环	30
3.6 自学	31

## 第 4 章 函数、脚本和良好的编程 实践

4.1 动机引例	32
4.1.1 银行利率问题	32
4.1.2 飞行时间问题	32

4.2 脚本 .....	33	6.6.3 拟合参数的初始猜测 .....	61
4.3 函数 .....	35	6.6.4 基于初始猜测的数据和 模型绘制 .....	61
4.4 良好的编程实践 .....	37	6.6.5 拟合数据 .....	62
4.4.1 简化代码 .....	37	6.6.6 拟合参数的不确定性 评估 .....	63
4.4.2 试着预见非预期行为 .....	37	6.7 自学 .....	65
4.4.3 运行测试用例 .....	38	<b>第7章 数值导数</b> .....	67
4.4.4 检查并清理输入参数 .....	39	7.1 通过前向差分估计导数 .....	67
4.4.5 判断解是否符合实际 .....	40	7.2 数值导数的算法误差估计 .....	68
4.4.6 良好的编程实践总结 .....	40	7.3 通过中心差分估计导数 .....	69
4.5 递归函数和匿名函数 .....	40	7.4 自学 .....	70
4.5.1 递归函数 .....	40	<b>第8章 求根算法</b> .....	71
4.5.2 匿名函数 .....	41	8.1 求根问题 .....	71
4.6 自学 .....	42	8.2 试错法 .....	71
<b>第二部分 使用 MATLAB 求解 日常问题</b>		8.3 二分法 .....	72
<b>第5章 线性代数方程组求解</b> .....	46	8.3.1 二分法示例和测试用例 .....	74
5.1 风铃问题 .....	46	8.3.2 二分法代码的可能改进 .....	76
5.2 MATLAB 内置求解器 .....	48	8.4 算法收敛 .....	76
5.2.1 逆矩阵法 .....	48	8.5 试位法 .....	77
5.2.2 无逆矩阵计算的方法 .....	48	8.6 割线法 .....	78
5.2.3 选用哪种方法 .....	48	8.7 牛顿-拉弗森法 .....	79
5.3 用 MATLAB 求解风铃问题 .....	49	8.7.1 使用牛顿-拉弗森法进行 解析求导 .....	80
5.4 示例: 惠斯通电桥问题 .....	50	8.7.2 使用牛顿-拉弗森法进行 数值求导 .....	81
5.5 自学 .....	52	8.8 Ridders 法 .....	81
<b>第6章 数据约简与拟合</b> .....	53	8.9 求根算法的陷阱 .....	82
6.1 数据约简与拟合的必要性 .....	53	8.10 求根算法总结 .....	83
6.2 拟合的正式定义 .....	53	8.11 MATLAB 内置求根命令 .....	84
6.3 数据拟合示例 .....	54	8.12 自学 .....	84
6.4 参数不确定性估计 .....	56	<b>第9章 数值积分方法</b> .....	86
6.5 拟合结果评估 .....	56	9.1 积分问题描述 .....	86
6.6 如何得到最优拟合 .....	58	9.2 矩形法 .....	86
6.6.1 数据绘图 .....	60		
6.6.2 选择拟合模型 .....	60		

9.3	梯形法	89	11.1.2	概率密度函数	108
9.4	辛普森法	90	11.2	均匀随机分布	109
9.5	广义积分公式	90	11.3	随机数生成器和计算机	110
9.6	蒙特卡罗积分	91	11.3.1	线性同余生成器	110
9.6.1	示例: 计算池塘面积	91	11.3.2	随机数生成器周期	111
9.6.2	朴素蒙特卡罗积分	91	11.4	如何检验随机数生成器	111
9.6.3	蒙特卡罗积分推导	91	11.5	MATLAB 的内置随机数 生成器	113
9.6.4	蒙特卡罗方法的算法 误差	92	11.6	自学	114
9.7	多维积分	92	<b>第 12 章</b>	<b>蒙特卡罗仿真</b>	115
9.8	蒙特卡罗多维积分	94	12.1	钉板实验	115
9.9	数值积分陷阱	94	12.2	抛硬币游戏	117
9.9.1	使用大量的数据点	94	12.3	传染病传播	118
9.9.2	使用过少的数据点	95	12.4	自学	123
9.10	MATLAB 的积分函数	95	<b>第 13 章</b>	<b>优化问题</b>	125
9.11	自学	96	13.1	优化问题简介	125
<b>第 10 章</b>	<b>数据插值</b>	98	13.2	一维优化	126
10.1	最近邻插值	98	13.2.1	黄金分割最优搜索 算法	126
10.2	线性插值	99	13.2.2	一维最优 MATLAB 内置 函数	128
10.3	多项式插值	101	13.2.3	一维优化示例	128
10.4	好的插值程序的准则	102	13.3	多维优化	130
10.5	三次样条插值	102	13.4	组合优化	135
10.6	MATLAB 内置的插值 方法	104	13.4.1	背包问题	135
10.7	外推法	104	13.4.2	旅行商问题	138
10.8	插值的非常规应用	104	13.5	模拟退火算法	143
10.9	自学	105	13.6	遗传算法	150
<b>第三部分</b>	<b>深入研究并扩展科学家 的工具箱</b>		13.7	自学	151
<b>第 11 章</b>	<b>随机数生成器和随机 过程</b>	108	<b>第 14 章</b>	<b>常微分方程</b>	153
11.1	统计和概率简介	108	14.1	常微分方程简介	153
11.1.1	离散事件的概率	108	14.2	边界条件	154
			14.3	求解常微分方程的数值 方法	155

14.3.1 欧拉方法 .....	155	15.1.3 复数的傅里叶级数表示 ...	171
14.3.2 二阶 Runge-Kutta 方法 (RK2) .....	156	15.1.4 非周期函数 .....	172
14.3.3 四阶 Runge-Kutta 法 (RK4) .....	157	15.2 离散傅里叶变换 .....	172
14.3.4 其他数值求解器 .....	157	15.3 MATLAB 的 DFT 实现及快速 傅里叶变换 .....	174
14.4 刚性常微分方程及数值解的 稳定性问题 .....	157	15.4 傅里叶变换的简化符号 .....	175
14.5 MATLAB 的内置常微分方程 求解器 .....	159	15.5 DFT 示例 .....	175
14.6 常微分方程示例 .....	159	15.6 自学 .....	178
14.6.1 自由落体 .....	159	<b>第 16 章 数字滤波器</b> .....	179
14.6.2 空气阻力运动 .....	161	16.1 奈奎斯特频率和最小 采样率 .....	179
14.7 自学 .....	166	16.2 DFT 滤波器 .....	182
<b>第 15 章 离散傅里叶变换</b> .....	167	16.2.1 低通滤波器 .....	182
15.1 傅里叶级数 .....	167	16.2.2 高通滤波器 .....	183
15.1.1 示例: $ t $ 的傅里叶 级数 .....	169	16.2.3 带通和带阻滤波器 .....	184
15.1.2 示例: 阶跃函数的傅里叶 级数 .....	170	16.3 滤波器的伪影 .....	186
		16.4 窗口效应 .....	188
		16.5 自学 .....	189
		<b>参考文献</b> .....	191

| 第一部分 |

Programming with MATLAB for Scientists: A Beginner's Introduction

# 计算基础

- 第 1 章 计算机与编程语言简介
- 第 2 章 MATLAB 基础
- 第 3 章 布尔代数、条件语句和循环
- 第 4 章 函数、脚本和良好的编程实践

# 计算机与编程语言简介

本章以现代计算的概述为背景，定义了什么是编程和编程语言，并进一步解释了数字在计算机中的作用和潜在的使用问题。

计算机的特点是难以置信的快速、准确和愚蠢，人的特点却是缓慢、不准确和聪明。计算机和人类的结合会产生超乎想象的能量。

——Leo Cherne<sup>⊖</sup> (1969)

## 1.1 早期计算史

我们很难相信，仅仅 100 年之前，世界上还没有计算机。过去“计算机”是指那些训练有素的人，他们只使用大脑和一些辅助工具就能完成相当复杂的计算。如图 1.1 所示，这幅画<sup>⊖</sup>描绘了俄罗斯乡村的一所普通学校中的场景。尽管黑板上的算式很难看清，但学生们却在努力求解如下表达式的结果：

$$\frac{10^2 + 11^2 + 12^2 + 13^2 + 14^2}{365} \quad (1.1)$$

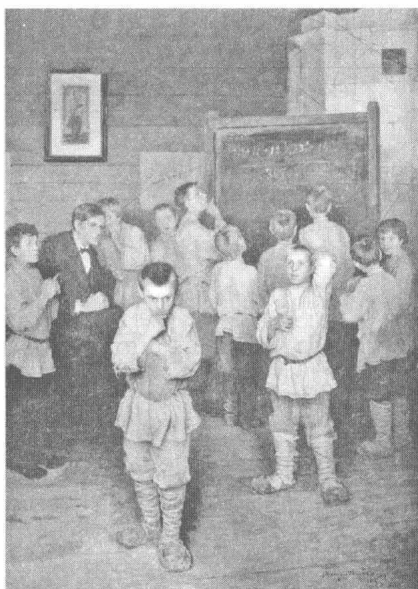


图 1.1 左图是 Nikolay Bogdanov-Belsky 1895 年创作的画《Mental Arithmetic. In the Public School of S. Rachinsky》，右图是黑板的局部放大图

<sup>⊖</sup> 这句话的出处并不是十分确定。

<sup>⊖</sup> 这幅画不受版权限制，可以从以下地址获得：[www.wikiart.org/en/nikolay-bogdanov-belsky/mental-arithmetic-in-the-public-school-of-s-rachinsky](http://www.wikiart.org/en/nikolay-bogdanov-belsky/mental-arithmetic-in-the-public-school-of-s-rachinsky)。

常用的计算辅助工具有：算盘、滑动尺、预先计算的函数表(对数表、三角函数表、指数表……)和19世纪后期开始出现的机械计算器。这些辅助工具是不可编程的，它们只用于完成一些基本计算。相反，人类却会“编程”，也就是说，人类可以解决更复杂的问题<sup>①</sup>。

## 1.2 现代计算机

世界上第一台通用计算机是诞生于1946年的ENIAC(电子数字积分计算机)，ENIAC也是现代计算机的鼻祖。其规格如下：

- 重量：30t
- 成本：50万美元(考虑到通货膨胀，相当于现在的600万美元)
- 能耗：150kW(相当于500个家庭的平均用电量)

ENIAC最快可以在1s内完成以下运算：5000次加法运算，357次乘法运算，或者38次除法运算。现代计算机速度是用FLOPS(每秒浮点运算次数)来度量的。ENIAC的性能相当于100 FLOPS，而几年前笔者的台式机的性能大约是50M FLOPS。现在普通手机的性能都比ENIAC高很多数量级。

### 现代计算机的共同特征

现代计算机通常具有以下特征：具有一个或多个中央处理单元(CPU)、存储数据与程序的存储器以及输入和输出接口(键盘、硬盘、显示器、打印机等)。典型的计算机内部使用二进制系统<sup>②</sup>。尽管存在一些差异，但是能够将现代计算机与早期计算机区分的主要特征是：计算机可以在不改变硬件的情况下为任何通用任务编程。

## 1.3 什么是编程

如果计算机是可编程的，那么我们应该能够进行编码，即由计算机生成适合执行的指令列表。“生成指令列表”听起来并不可怕。那么，问题是什么呢？编程可不是一件小事，事实上，甚至可以把它称为“编程艺术”。

想象一下你想吃东西的情景。对你来说最简单的程序就是：买比萨饼，然后吃掉它。这听起来很容易。现在开始把它分解成适合身体执行的指令列表：拿起电话(注意，首先要找到电话，这可不能忽略)，拨打号码(想象一下活动你的手臂需要多少条指令，仅仅弯曲手指、指向数字、按下号码就需要很多指令)，和电话另一端的售货人员交谈，确认订单(如果尝试将这分解成一些基本操作，也不是那么简单的)，等待快递，付款(涉及开门、交谈、找钱包、数钱、付钱、找零一系列动作)，把比萨饼盒拿到厨房，打开盒子，将比

① 可以说这是可编程性的顶峰。只需要通过正常的人类语言指定要做什么，然后任务就能完成。现在，我们必须把任务分解成非常小的基本子任务，以便计算机能理解这些任务。

② 有人尝试使用三进制系统的计算机，即基于数字3的计算机。三进制有一些优势，但是硬件成本更高。

萨饼切片，最后才是把它吃掉。这些操作中的每一步都能分解成一组更基本的动作。

在计算机上编程时，不仅需要实现所有细节，而且必须以一门“外语”（对你来说是这样的）来完成，这种外语甚至不是为人类设计的。

所以，我们把这些都放进了“编码”的范畴。这相对容易；更难的问题是需要提前考虑，并为每步操作设计一个安全保障（比如比萨饼很凉或者你拨错号码该怎么办）。但这还算相对容易。从那些看起来正确的程序流中发现错误并进行修复（即调试）才是比较难掌握的技能。科学上的编程甚至更难，因为你常常是第一个进行特定计算的人，没有参考样例来进行验证。例如，如果你是第一个精确计算  $\pi$  的数值的人，那么如何知道第一万亿位的小数是否正确呢？

本书中将尽可能地展示检查或测试程序的方法。在编程过程中你应该有意识地去寻找测试用例。任何人都可以编码，但是只有真正的高手才可以肯定地说某个程序是正确的，实现了它的设计目标<sup>⊖</sup>。

## 1.4 编程语言概述

毫不夸张地说，编程语言有成百上千种。不幸的是，没有一种语言是能适应任何情况的“银弹”（silver bullet）语言<sup>⊖</sup>。编程语言的特点各不相同，有的以速度见长，有的注重简洁，有的更易于硬件控制，有的侧重于减少错误，有的擅长图形和声音输出，有的则适合数值计算，等等。值得高兴的是大多数编程语言都有些相似。这与人类语言一样，一旦知道词语分为名词、动词和形容词等，在所有语言中都能这样划分词语，然后只需要掌握特定语言的正确语法就可以了。

编程语言的分类方法有很多，其中一种就是根据语言的层级高低来划分。在比萨饼订单的示例中，高级语言命令集可能只需要“订购”和“吃”两个指令；低级语言程序将非常详细，甚至包括控制神经脉冲来确定弯曲胳膊和手指的时机。

最低级的编程语言无疑是二进制代码。这是计算机唯一能理解的语言，当程序由计算机执行时，其他编程语言都需要转换成二进制语言。二进制代码不适合人类阅读，除了使用内存不足的微处理器的情况，现在基本不使用它。低级语言的代表有汇编语言、C、C++、Forth 和 LISP。高级语言的代表有 Fortran、Tcl、Java、JavaScript、PHP、Perl、Python 和 MATLAB。低级语言和高级语言之间的区别取决于使用者的具体应用。例如，使用 MATLAB 进行数值计算时，代码非常简洁，而在执行文件查找时，代码就显得非常冗长，比如在所有文件中查找名为 victory 的文件。

编程语言分类的另一种依据是内部实现。有些语言在执行之前需要进行编译（compile），即将程序编译成整体的二进制代码。而其他的语言，当命令到达执行队列时，将逐

⊖ 计算机科学家有一个定理：不可能证明一个通用（足够复杂）程序的正确性。这就是停机问题（halting problem）。

⊖ 所有编程语言中笔者只懂 10 种，日常工作中常用的也只有 4 种左右。



一解释(interpret)这些命令。

下面这个例子很好地突出了这两种方法之间的差异。想象一下，有人给你一篇外语论文，让你在众人面前阅读。你可以提前翻译它，然后直接阅读完整的翻译(编译)版本。或者，也可以等到演示的时候再逐行翻译(即解释它)。通过这个例子，我们可以看到每种方法的优点和缺点。如果你在演讲过程中需要速度，那就先进行完整的翻译或编译；然后，你在演示期间才有能力做其他的事情。但是这需要花时间来做准备。你也可以动态地解释论文，但是你会被这个任务压得喘不过气来，无法进行别的活动。然而，如果你正在处理一篇论文，需要在目标读者和作者之间来回切换，每次都重新开始翻译论文将很痛苦(计算机与人类不同，它没有记忆能力，一切都要从零开始)。如上所述，解释方式更方便进行交互式调试：找到问题点，修复它，然后继续。没有必要重新开始，因为它是一样的。在这种情况下，解释式的编程语言非常出色。

还有介于以上两种类别之间的第三种编程语言，将程序预先转换为一种更简单、更容易解释的语言，而不是二进制代码。但是从我们的观点来看，这与需要编译的语言类似，因为我们不能用这种方法交互式地调试程序。

在这个意义上来说，MATLAB 是一种交互式语言<sup>⊖</sup>，所以使用 MATLAB 还是很有趣的。想出命令，直接运行，查看结果。如果出现错误(bug)可以直接修复，而不需要花几个小时来重新进行初步计算。

### 智慧之言

计算机做你要求的工作，而不是你想做的。如果计算机输出了不满意的结果，很可能是因为你没有正确地传递你的愿望。

## 1.5 计算机中的数字表示及其潜在问题

### 1.5.1 离散化——计算机的主要弱点

让我们看下面的表达式：

$$1/6 = 0.1666666666666666\dots$$

因为小数部分的 6 是无限循环的，所以要把这个数字无限精确地保存在计算机里是不可能的。毕竟，计算机具有有限的内存大小。为了避免对无限内存的需求，计算机将每个数字截断到指定的有效位数。

例如，假设计算机只保存四位有效数字，则有：

$$1/6 = 0.1667_c$$

这里，下标“c”代表计算机表示方法。注意，最后一个数字不再是 6，计算机把它四

<sup>⊖</sup> 如果你确实需要速度，可以编译代码的关键任务部分，尽管这超出了本书讨论的范围。