

MATLAB 7 及工程问题解决方案

Introduction to MATLAB 7

(美) Delores M. Etter David C. Kuncicky Holly Moore 著
邱李华 译

MATLAB 7 及工程问题解决方案

Introduction to MATLAB 7

(美) Delores M. Etter David C. Kuncicky Holly Moore 著

邱李华 译



机械工业出版社
China Machine Press

本书介绍MATLAB基本功能以及如何用MATLAB来解决数学中的典型问题。书中涉及的主要内容有: MATLAB的基本使用方法, MATLAB的预定义函数, 如何用MATLAB绘图, 如何在MATLAB中编程, 基于MATLAB的矩阵运算、符号运算、数值分析等。另外, 每一章结束时都进行了小结, 还有大量具有实际意义的思考题, 供读者上机练习, 巩固所学的知识。

本书既可作为高等院校相应课程的教材使用, 也可作为广大读者学习MATLAB的自学用书。

Simplified Chinese edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Introduction to MATLAB 7* (ISBN 0-13-147492-8) by Delores M. Etter and David C. Kuncicky with Holly Moore, Copyright © 2005, 2004, 2002.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

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

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2005-1310

图书在版编目(CIP)数据

MATLAB 7及工程问题解决方案/ (美) 埃特 (Etter, D. M.) 等著; 邱李华译. -北京: 机械工业出版社, 2006. 3

书名原文: Introduction to MATLAB 7

ISBN 7-111-18372-X

I. M… II. ①埃… ②邱… III. 计算机辅助计算-软件包, MATLAB 7-教材 IV. TP391.75

中国版本图书馆CIP数据核字 (2006) 第004681号

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: 迟振春

北京京北制版印刷厂印刷 · 新华书店北京发行所发行

2006年3月第1版第1次印刷

787mm × 1092mm 1/16 · 15.5印张

印数: 0 001-4 000册

定价: 30.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010) 68326294

译者序

MATLAB是美国Math Works公司于1984年推出的一种可视化科学计算软件。它以矩阵计算为基础，将计算功能、可视化功能和程序设计功能集成在一个易学易用的环境之下，可以实现数学计算、算法研究、数据获取、建模和仿真、数据分析及数据可视化、科学和工程绘图、应用程序开发（包括图形用户界面的设计）等功能。

本书由美国科罗拉多大学教授Delores M. Etter、新墨西哥州Sante Fe市Bioreason公司的工程主任David C. Kuncicky和盐湖社区学院的工程教授Holly Moore编写，是一本介绍MATLAB 7的入门书籍。书中主要介绍了MATLAB的环境以及如何用MATLAB来解决数学中的许多典型问题。该书首先提出了科学和工程中存在的一系列巨大挑战，并提出了工程问题的一种解题步骤。随后把相对枯燥的数学问题与解决这些挑战有关的大量实际工程问题结合起来，将该解题步骤应用于书中的所有示例，贯穿全书。

书中每一章的开始都提出了一个或几个实际工程问题，并在示例中结合这些问题逐渐展开进行讨论，高度体现了运用MATLAB解决工程问题的优越性。例题按规范的工程问题解题步骤给出解答，既做到结合实际，又规范了解决问题的方法，使解决复杂问题也变得井井有条，通俗易懂。在每一章结束时都进行了全面的小结。各章之后附有大量具有实际意义的思考题，供读者上机练习，巩固所学知识。

在内容编排上，本书采用了由浅入深、从简到繁的安排，给阅读带来了方便。读者通过阅读本书并自己上机实践，可以建立良好的MATLAB基础，并规范解题的方法。

在翻译本书的过程中，译者同时也领略了作者丰富的工作经验和严谨的科学态度，相信读者在阅读本书的过程中也会有同样的体会。

本书既可作为科学研究和工程技术工作者的自学用书或参考书，也可作为相应课程的教材使用。

全书由邱李华翻译，郭全、龙月元、张金枝协助工作。由于译者水平所限，虽然我们已尽了最大努力，译文中还是难免会有不当和疏漏之处，恳请读者提出宝贵意见。

邱李华

关于作者



Delores M. Etter是科罗拉多大学电机与计算机工程教授。Etter博士是新墨西哥大学的职员，同时也是斯坦福大学的客座教授。她在新墨西哥大学负责新生工程计划，并且在科罗拉多大学的综合教学实验室工作。由于对教育事业的贡献，以及在数字信号处理领域的技术领先作用，她被选为IEEE会员。



David C. Kuncicky是佛罗里达州人，获得了佛罗里达州立大学的心理学学士学位、计算机科学硕士学位以及计算机科学博士学位。他是FAMUFSU工程学院电机工程系和佛罗里达州立大学计算机科学系的教员。他讲授计算机科学和计算机工程课程超过15年，发表了在智能混合系统和神经网络领域的研究成果。目前，他是新墨西哥州Sante Fe市Bioreason公司的工程主任。

Holly Moore是盐湖社区学院的工程教授，在该学院讲授热力学、材料学工程和工程计算课程。她获得了南达科他矿业与技术学校的化学工程学士学位、化学工程硕士学位，并获得了犹他大学化学工程博士学位。她致力于航空工业长达十年之久，为国防和空间计划设计并分析固体燃料火箭助推器。她还积极参加犹他州的“hands-on elementary science material”项目开发。

不可否认，Holly受到了她的父亲George Moore教授的巨大影响，Moore教授在南达科他矿业与技术学校电机工程系任教约20年。Moore教授在美国空军事业的成功生涯之后从事大学教育工作，是一个“活到老学到老”的典范。

目 录

译者序	
关于作者	
第1章 工程问题解决方案	1
1.1 巨大的挑战	1
1.2 计算系统	2
1.2.1 计算机硬件	2
1.2.2 计算机软件	4
1.3 一种工程问题的解题步骤	9
1.3.1 陈述问题	9
1.3.2 描述输入/输出	10
1.3.3 手算示例	10
1.3.4 MATLAB解决方案	10
1.3.5 测试	11
本章小结	11
关键词	12
思考题	12
第2章 MATLAB环境	14
2.1 开始	14
2.1.1 MATLAB学习版	15
2.1.2 MATLAB窗口	15
2.2 用MATLAB解决问题	22
2.2.1 变量	22
2.2.2 使用矩阵	22
2.2.3 数字显示	36
2.3 保存工作	38
2.3.1 保存变量	38
2.3.2 脚本M-文件	41
本章小结	44
MATLAB摘要	45
关键词	46
思考题	46
第3章 MATLAB预定义函数	50
3.1 使用预定义函数	50
3.1.1 使用帮助功能	51
3.1.2 基本数学函数	53
3.1.3 三角函数	57
3.1.4 数据分析函数	60
3.1.5 随机数	70
3.2 处理矩阵	72
3.2.1 定义矩阵	72
3.2.2 使用冒号运算符	74
3.3 计算的限制	84
3.4 特殊值和函数	85
本章小结	85
MATLAB摘要	85
关键词	86
思考题	87
第4章 绘图	90
4.0 引言	90
4.1 二维图形	90
4.1.1 基本绘图方法	90
4.1.2 线、颜色和标记风格	94
4.1.3 坐标轴的缩放	97
4.1.4 给图形添加注释	97
4.1.5 其他类型的二维图形	103
4.1.6 子图形	105
4.2 三维图形的绘制	112
4.2.1 三维线图	112
4.2.2 曲面图	113
4.3 用菜单栏编辑图形	117
4.4 从Workspace窗口创建图形	119
本章小结	120
MATLAB摘要	120
思考题	122
第5章 MATLAB编程	125
5.0 引言	125
5.1 两个变量的问题	125
5.2 输入/输出	131
5.2.1 用户定义的输入	131
5.2.2 输出选项	134
5.3 函数	138
5.3.1 语法	138
5.3.2 局部变量	140

5.3.3 给函数M-文件命名	140	7.1.2 符号绘图	188
5.3.4 编写和使用函数M-文件的规则	141	7.1.3 数学表达式的化简	189
5.4 语句级的控制结构	144	7.1.4 符号表达式的操作	190
5.4.1 关系运算符和逻辑运算符	144	7.2 解方程	191
5.4.2 选择结构	146	7.3 微分和积分	197
5.4.3 循环	153	7.3.1 微分	197
本章小结	159	7.3.2 积分	198
MATLAB摘要	159	本章小结	200
关键词	160	MATLAB摘要	200
思考题	160	关键词	201
第6章 矩阵运算	164	思考题	201
6.1 矩阵运算和函数	165	第8章 数值技术	203
6.1.1 转置	165	8.1 插值	203
6.1.2 点乘	165	8.1.1 线性插值	204
6.1.3 矩阵乘法	167	8.1.2 三次样条插值	204
6.1.4 矩阵乘方	171	8.1.3 <code>interp1</code> 函数	205
6.1.5 矩阵求逆	172	8.2 曲线拟合: 线性回归和多项式回归	211
6.1.6 行列式	172	8.2.1 线性回归	211
6.2 求线性方程组的解	173	8.2.2 多项式回归	212
6.2.1 使用逆矩阵解方程组	173	8.2.3 <code>polyfit</code> 和 <code>polyval</code> 函数	213
6.2.2 使用矩阵左除法解方程组	176	8.3 使用交互式拟合工具	217
6.3 特殊矩阵	176	8.3.1 基本拟合工具	217
6.3.1 零矩阵	176	8.3.2 曲线拟合工具箱	220
6.3.2 一矩阵	177	8.4 数值积分	223
6.3.3 单位矩阵	177	8.4.1 梯形法则和辛普森法则	224
6.3.4 对角矩阵	179	8.4.2 MATLAB求积分函数	225
6.3.5 幻方矩阵	179	8.5 数值微分	227
本章小结	181	8.5.1 差分表达式	227
MATLAB摘要	181	8.5.2 <code>diff</code> 函数	228
关键词	181	本章小结	231
思考题	181	MATLAB摘要	231
第7章 符号数学	185	关键词	231
7.1 符号代数	185	思考题	232
7.1.1 符号表达式	185	附录 特殊字符、命令和函数	235

第1章 工程问题解决方案

学习目标

阅读完本章之后，你将学会以下内容：

- 熟悉当今科学与工程学中的基本问题
- 了解MATLAB与计算机软硬件之间的关系
- 了解一种工程问题的解决步骤

1.1 巨大的挑战

工程师们使用来自不同学科的科学原理来解决现实世界中的问题，这些学科包括计算机科学、数学、物理和化学。正是这些学科以及用其解决现实问题的挑战性，激发了工程师们极大的兴趣和工作热情。本节将向读者展现一组**巨大的挑战**——在科学和工程学中具有广泛的潜在影响的基本问题。这些巨大的挑战被美国首都华盛顿的科学与技术策略办公室（Office of Science and Technology Policy）确定为高性能计算的研究与开发策略的一部分。在随后的段落中将简要介绍这些巨大的挑战，以及解决这些挑战所带来的各种好处。有关具体挑战内容的进一步讨论将在每一章的开始处介绍。正如计算机在过去35年来在工程成就中所起的重要作用一样，在解决与这些巨大的挑战相关的问题中，计算机将发挥更为重大的作用。

气象、气候以及全球环境变化的预测要求我们了解大气和海洋生物圈这两个相互联系的系统。其中包括了解大气中CO₂的动态变化、海洋、臭氧层损耗以及由于化学药品或能源的释放所带来的气候变化。这些复杂的过程还包括太阳光的交互作用。一次来自“冕洞”（太阳风的一个泄漏点）附近的太阳风暴的大爆发将以每小时超过一百万英里^①的速度从太阳表面向地球表面喷射大量的热气。这种热气的喷射将向地球轰击X射线，影响通信并导致输电线的波动。学习气象、气候和全球环境变化的预测涉及搜集大量的研究数据、开发出能够代表许多变化的事物之间相互依赖关系的新的数学模型。

计算机化的语音识别可以使我们的通信系统得以改革。但是，它的开发和实施涉及许多问题。目前可以做到“教会”一台计算机识别来自某一个人的少量词汇，但要开发一个独立于说话者的、能理解大量词汇和不同语言的系统就要困难得多。另外，一个人声音的细微变化，如因感冒或重音引起变化，也会影响语音识别系统的性能。即便假设计算机能够识别词语，要让计算机能够确定词语的含义也不是一件简单的事情。许多词语是上下文相关的，因此不能把它们独立出来进行分析。声调的变化（如一个人声音的提高）可能将一个陈述句变成疑问句。虽然在自动语音识别和理解领域中还可以列举出许多难题，但令人振奋的应用还是随处可见的。例如，可以想象一下在电话系统中对电话线上传输的语言进行识别，并对语音信号进行翻译，使每一个人所听到的对话是用他自己本国的语言说出来的。

人类基因工程的目标是对人类主要的遗传物质（DNA，或称脱氧核糖核酸）中所包含的

① 英里的单位符号为mile，1 mile = 1 609.344 m。——编辑注

30 000多种人类基因进行定位、识别并确定其功能。到2000年,大多数的人类基因已经得到了排序。2001年2月公布了基因的30亿个碱基对中的90%的基因顺序。

既然大多数人类基因已经形成图谱,那么,研究者的研究目标是要弄清楚人类基因在创造基因产品中是如何起作用的,特别是由这些基因编码的许多蛋白。

破译人类的基因密码并确定每一个基因的功能将引发许多技术的发展,包括对4000多种已知的人类基因疾病中的许多疾病的检测、治疗和预防,例如镰状细胞贫血症和囊肿性纤维化。

运载工具性能的重大改进要求为三维流场、涡轮机组及流体输送管道内部的流动建立更加复杂的流体动力性能物理模型。流体流动中的湍流影响太空飞行器的稳定性、可控性、热力性能和燃料消耗性能;为了对新的结构进行分析,有必要对这种流动进行建模。分析飞行器的空气弹性变形性能还影响到新设计的开发。燃烧系统的效率也与飞行器的性能有关,因为要在燃烧效率方面获得重大的改进,就需要了解各种不同物质的流动以及引起物质起反应的化学原理之间的关系。人们也正通过在运载工具上使用计算机和微处理器来确定其性能。例如,目前人们正通过在汽车的仪表板上安装小型的视频屏幕来研究运输系统的性能。驾驶员将目的地输入到计算机中,视频屏幕将显示从当前位置到达预定目标的有关路线,包括街道的名称。通信网络还会及时提醒汽车驾驶员沿途有关的交通堵塞情况,并在必要时自动为汽车重选路线。其他有关运输工具的研究目标是如何使用计算机和网络来处理所有的控制和信息交换,以实现全自动驾驶。

提高石油和天然气开采技术将使我们得以在美国找到估计3 000亿桶的石油储藏量。目前用来鉴别地下结构是否含有石油和天然气的震动技术可以探测到地表之下20 000英尺^①深的位置。这些技术在被探测的区域附近使用了一组传感器,称为传感器组。一种地面震动信号被发送到地下,在不同的地质层分界处信号将产生反射,由传感器接收反射回来的信号。使用复杂的信号处理方法,则可以描绘出地质层的分界面,因此可以对不同地质层中的物质做出某种预测,例如沙岩、泥板岩和水。地面震动信号可以通过多种方法来产生:可以钻一个洞,然后在洞中制造一个猛烈的撞击;这种猛烈的撞击也可以在地表上进行;或者可以使用一种带有液压锤的特殊车在地上每秒猛烈撞击若干次。进一步的研究是如何提高对信息的分析技术,并且寻找经济可行的生产和开采方法。

这些巨大的挑战仅仅是等待工程师和科学家们解决的许多有趣问题的一小部分。将想法与技术系统地结合起来,形成规范的方法,将使这些重大问题得以解决。计算机的使用和工程问题解决技术的相结合将是解决问题的关键。

1.2 计算系统

在开始讨论MATLAB之前,我们先简要介绍一下有关计算的问题,这对于没有计算机经验的读者尤其有用。计算机被设计用来执行一组操作,这组操作用专门的指令写出来,称为一个程序。计算机硬件是指计算机设备,例如键盘、鼠标、终端、硬盘和打印机。计算机软件是指描述我们希望计算机执行的操作步骤的程序。

1.2.1 计算机硬件

所有的计算机都具有共同的内部结构,如图1-1所示。处理器用于控制计算机的所有其他

^① 英尺的单位符号为ft, 1 ft = 0.304 8 m。——编辑注

部分。它用于接收输入信息（来自某种设备，如键盘）并将它们保存到计算机的存储器中。它还对计算机程序中的指令进行解释。如果我们要求两个数相加，处理器将从存储器中检索这两个数，并将它们传送到**算术逻辑单元**（或称ALU）中，ALU执行加法操作，然后由处理器将相加的结果保存到存储器中。处理单元和ALU在工作过程中使用由只读存储器（ROM）和随机存储器（RAM）组成的内部存储器，大多数的数据存储在外部存储器或辅助存储器中，它们使用硬盘驱动器或软盘驱动器连接到处理器。处理器和ALU一起统称为**中央处理器**（或称为CPU）。**微处理器**是一种CPU，它由一块集成电路芯片组成，在这种芯片的一个比邮票还小的区域中就包含了数百万的部件。

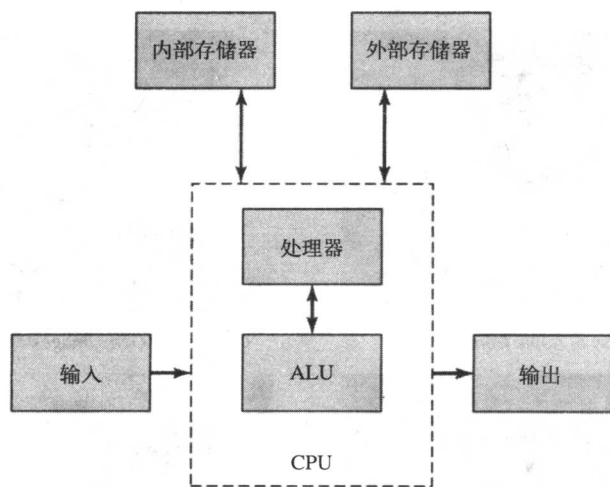
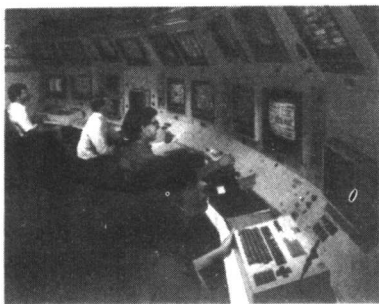


图1-1 计算机的内部结构

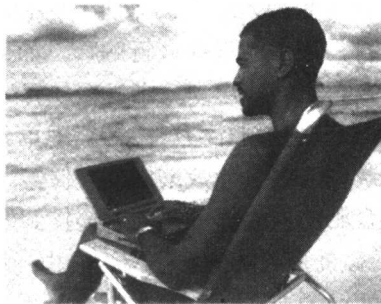
我们通常命令计算机将计算结果打印到终端或通过打印机打印到纸张上。点阵打印机使用一个打印针阵（或称为网格）在纸上产生一个字符的形状，而激光打印机使用光束将图像转移到纸上。计算机还可以将信息写到磁盘上，磁盘用磁化的方法存储信息。信息的打印拷贝称为**硬拷贝**，而信息的磁拷贝称为**电子拷贝**或**软拷贝**。

计算机有各种大小、形状和组成（见图1-2）。**个人计算机（PC）**较小，价格便宜，常用于办公、家庭和实验室。PC也称为**微型计算机**。它们围绕微处理器进行设计，例如奔腾微处理器，它每秒可以处理数百万条指令（mips）。小型计算机比微型计算机功能要强。大型计算机具有更强大的功能，常用于商业和研究实验室。**工作站**是一种小到足以放到桌面上的小型机或大型机。**超级计算机**是一种速度最快的计算机，每秒可以处理几十亿条指令。由于超级计算机处理速度快，因此能够解决其他计算机所不容易完成的极其复杂的问题。大型计算机和超级计算机需要使用专门的配备（facility），并由专业人员来运行和维护。

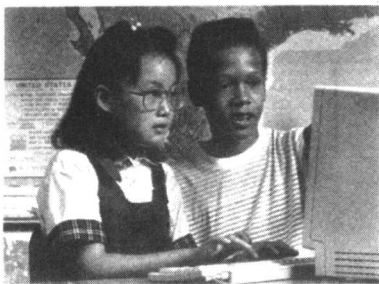
解决某种特殊问题所需要的计算机类型取决于问题的需要，如果用于家用安全系统的一部分，使用一个微处理器就足够了；如果使用计算机来控制军用飞行模拟器，可能就需要使用大型计算机。**计算机网络**使计算机之间可以相互通信，以实现资源和信息的共享。例如，以太网就是一种常用的局域网（Local Area Network, LAN）。



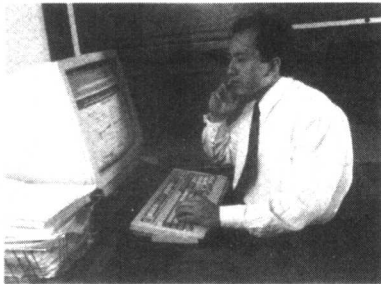
Courtesy of Johnson Space Center



Courtesy of Getty Images, Inc.



Courtesy of Apple Computer, Inc.



Courtesy of Getty Images/EyeWire, Inc.



Courtesy of Boeing Commercial Airplane Group



Courtesy of PhotoEdit

图1-2 各种不同用途的计算机

1.2.2 计算机软件

计算机软件包含了我们希望计算机执行的指令或命令。软件分为几种主要类型，包括操作系统、工具软件和语言编译器。图1-3表明了这些类型的软件及其与计算机硬件之间的相互关系。以下将详细介绍这些类型的软件。

1. 操作系统

用于控制或“操作”计算机的程序称为操作系统。操作系统负责管理计算机硬件，如磁盘驱动器、终端、键盘和调制解调器。其他程序如果要访问计算机硬件，必须向操作系统提出请求。

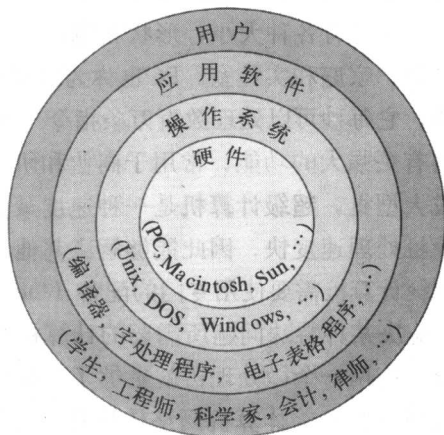


图1-3 软件和硬件之间的相互关系

操作系统还负责管理运行于计算机上的程序。现代计算机可以在同一时刻同时运行几十个或几百个程序。操作系统就像交通警察一样,负责安排在某一时刻允许某个程序访问CPU。

当打开计算机时,操作系统中称为内核的部分被加载。该内核会一直运行,直到关闭计算机为止。目前流行的操作系统有Microsoft Windows 2000、Microsoft Windows XP、Linux、UNIX和Apple Mac OS。

操作系统中还包含了一组**实用程序**,用于执行如打印文件、将文件从一个磁盘复制到另一个磁盘、列出保存在磁盘上的文件清单等功能。虽然大多数操作系统都有这些共同的实用程序,但是不同操作系统执行这些功能所使用的命令是不同的。例如,使用DOS操作系统(主要用在PC环境下的一种磁盘操作系统)列出文件清单的命令是`dir`;使用UNIX操作系统(常用在 workstation 环境下的一种功能强大的操作系统)或Linux操作系统(常移植到各种不同硬件平台下的一种功能强大的免费操作系统)列出文件清单的命令是`ls`;有些操作系统被称为是用户友好的,因为它们简化了和用户的接口,例如Macintosh环境和Windows环境都是用户友好的操作系统。

由于MATLAB程序可以运行在许多不同的平台或硬件系统之下,而且一台个人计算机可以使用不同的操作系统,因此要讨论该课程内容在各种不同操作系统下的使用是不可能的。我们假设你的导师将向你提供使用大学里的计算机上的特定操作系统所需要的信息。这些信息也可以从操作系统的用户手册上获得。

2. 工具软件

工具软件是编写出来的程序,用于执行一些常用的功能。例如,**字处理程序**,Microsoft Word和Corel Word-Perfect都是用于录入和格式化文本的程序。使用字处理程序可以移动句子、段落,并且通常可以录入数学公式、检查拼写和语法。字处理程序也常用于录入计算机程序并将它们保存为文件。复杂的字处理程序还可以用来产生精美的页面,包括含有文本和标题的精美的图表和图形。这些字处理程序使用了一种称为**桌面出版**(desktop publishing)的技术,将功能强大的字处理程序和高质量的打印机结合起来,从而产生外观上很专业的文档。

电子表格程序是一种用于将数据很容易地显示在由行和列组成的网格上的工具软件。电子表格程序最初被用在财务和会计领域中,但是,许多科学和工程问题也可以使用电子表格程序来解决。多数电子表格程序包包含绘图功能,因此在分析和显示信息时非常有用。LOTUS 1-2-3和Microsoft Excel都是流行的电子表格程序。

另外一组流行的工具软件是数据库管理程序,如Microsoft Access、Microsoft SQL Server和ORACLE。使用这些程序可以存储大量的数据,然后可以很容易地检索这些数据,并将它们格式化为报表。数据库用于一些大型的机构中,如用于银行、医院、大学、饭店以及航空业中存储和组织重要信息。数据库还用于分析大量的科学数据,如气象学和海洋学就是常需要进行大量数据存储和分析的两个科学领域。

计算机辅助设计(CAD)软件包,如AutoCAD、ProE和Unigraphics,可以用来定义对象并按图形方式来操纵它们。例如,可以定义一个对象,然后从不同的角度进行观察,或者观察一个对象从一个位置旋转到另一个位置。CAD软件包常用于工程应用领域。

MATLAB、Mathematica、Mathcad和Maple都是功能非常强大的**数学计算工具**。这些工具不仅提供了功能强大的数学操作命令,而且还提供了丰富的图形生成功能。这种计算功能和可视化功能的结合使它们成为对工程师们非常有用的工具。

如果一个工程问题可以使用某种工具软件来解决,则应使用工具软件,这通常比使用计算机语言编写程序来解决该问题效率更高。但是,有许多问题不能用工具软件解决,或者所用的计算机上没有解决该问题所需的工具软件。因此,我们还要知道如何使用计算机语言编写程序。工具软件和计算机语言之间的区别越来越不明显,因为一些功能强大的工具软件,如MATLAB和Mathematica,除了拥有一些专门的操作之外,自身也提供了编程语言。

3. 计算机语言

计算机程序设计语言是一种符号表示形式,用于将指令与计算机相关联。计算机语言可以划分为不同的级别。低级语言(或称为机器语言)是最原始的语言。机器语言与计算机硬件的设计紧密相关。由于计算机的设计是基于两种状态的技术(也就是说,计算机是具有两种状态的设备,如线路的打开或关闭,开关的连接或断开,电荷的正或负),因此机器语言使用两种符号编写,这两种符号通常表示为数字0和1。于是,机器语言是一种二进制语言,而指令是由若干个0和1组成的序列,称为二进制串。由于机器语言与计算机硬件的设计紧密相关,因此,用于Sun Microsystems公司的计算机上的机器语言和用于Silicon Graphics公司的计算机上的机器语言是不一样的。

汇编语言是机器语言的符号形式。每一行代码通常产生一条机器指令。汇编语言与某种特定的处理器结构紧密相关,如Intel公司的80×86系列或Sun Microsystems公司的SPARC系列。使用汇编语言编写程序比使用二进制语言编写程序要容易得多,但仍然是一个繁琐的过程。

【例1.1】以下列出的汇编语言程序代码说明了典型的汇编语言语法。每一条指令单独占一行,指定一种操作,也就是操作码(opcode),其后跟随着相应的操作数。

```
mov    cx,bx
shl    cx,8
shl    bx,6
add    bx,cx
add    ax,bx
mov    cx,es:[ax]
```

高级语言是一种计算机语言,具有类似于英语的命令和指令,例如C、Fortran、Ada、Pascal、COBOL和BASIC。使用高级语言编写程序要比使用机器语言或汇编语言编写程序容易得多。但是,高级语言包含大量的命令以及大量的使用命令的句法(或语法)规则。为了说明工具软件和高级语言对语法和标点符号的要求,表1-1列出了使用几种不同的语言 and 工具计算给定直径时圆的面积。注意这一简单计算的相似点和不同点。虽然我们已经把C归入到高级语言,但也有许多人把C描述为一种中级语言,因为在C语言中允许访问低级程序,且常用于定义一些被转换成汇编语言的程序。

表1-1 不同软件的语句比较

软 件	语 句 示 例
MATLAB	area = pi*((diameter/2)^2);
C	area = 3.141593*(diameter/2)*(diameter/2);
Fortran	area = 3.141593*(diameter/2.0)**2
Ada	area:= 3.141593*(diameter/2)**2;
Pascal	area:= 3.141593*(diameter/2)*(diameter/2)
BASIC	let a = 3.141593*(d/2)*(d/2)
COBOL	compute area = 3.141593*(diameter/2)*(diameter/2)

计算机语言也常按代来划分。第一代计算机语言是机器语言，第二代是汇编语言，第三代是高级语言，第四代语言（也称为4GL）还没研制出来，仅根据其性能和程序的生产率来定义。第五代语言称为自然语言。使用第五代语言编写程序时，应该使用自然语言的语法。显然，自然语言的实现需要使用以上巨大挑战之一的成就：计算机化的语音识别。

Fortran (FORmula TRANslation) 在20世纪50年代中期被研制出来，用于解决工程问题和科学问题。随后该语言不断更新其标准，当前的标准是Fortran 90，它具有强大的数值计算能力，同时具备如C语言中的许多新特性和结构。COBOL (COmmon Business-Oriented Language) 出现于20世纪50年代后期，用于解决商业问题。许多用COBOL编写的程序遗留至今，成为2000年(Y2K)程序设计缺陷的一个常见根源。BASIC (Beginner's All-purpose Symbolic Instruction Code) 于20世纪60年代中期被研制出来，并作为教育工具使用；20世纪80年代，PC机的系统软件中常含有BASIC解释器。Pascal于20世纪70年代初期被研制出来，在20世纪80年代被广泛用于向学生介绍用于计算的计算机科学程序中。Ada最早在美国国防部被开发出来，其主要目的是开发出一种可以嵌入到使用微处理器的计算机系统的高级语言。1979年，该语言的最终设计得到了认可，为了表示对Ada Lovelace的敬意，该语言命名为Ada。Ada Lovelace在19世纪初期开发了用于在一种分析机上执行计算的指令集。C是一种通用的语言，源自BCPL和B两种语言，这两种语言于20世纪60年代后期在贝尔实验室被研制出来。1972年，Dennis Ritchie在贝尔实验室研制并实现了DEC PDP-11计算机上的第一个C编译器。研制该语言的初衷是用来编写UNIX操作系统。在此之前，大多数操作系统都是用汇编语言编写的。C成为了非常流行的系统开发工具，因为它独立于硬件（不同于汇编程序）。由于它在工业和学术届的普及，因此就需要对其标准进行定义。1983年创建了美国国家标准化组织(American National Standards Institute, ANSI)下的一个协会，用于提供一种独立于机器且无二义性的C定义标准。1989年，提出了C的ANSI标准。C++是一种面向对象的程序设计语言，是C语言的一种扩展。早期C++的多数功能在20世纪80年代中期由贝尔实验室的Bjarne Stroustrup开发出来，C++比C新增的主要功能是继承、抽象类、重载运算符和一种动态类型绑定（虚函数）。20世纪90年代期间，C++成为了一种占据主要地位的程序设计语言，广泛用于工程、金融、电信、嵌入式系统以及计算机辅助设计领域。1997年，国际标准化组织(International Standards Organization, ISO)通过了C++标准。

4. 执行计算机程序

使用高级语言（如C）所编写的程序中的指令要在计算机上执行，必须首先被翻译成机器语言。执行这种翻译功能的程序是一种称为编译器的特殊程序。因此，要在一台计算机上编写并执行C程序，该计算机的软件中必须包含一个C编译器。

如果编译器在编译期间检测到任何错误（通常称为bug），则打印出相应的错误消息。我们必须修改程序中的语句，然后再次执行编译步骤。编译器在编译期间检测到的错误称为编译错误，或编译期错误。例如，如果我们要将存储在变量`sum`中的值除以3，在C中的正确表达式是`sum/3`，如果我们错误地使用反斜杠“\”书写该表达式，如写成`sum\3`，则将得到一个编译错误。编译、修改语句（或称调试）、再重新编译的过程通常要重复多次，直到程序中没有编译错误。当没有编译错误时，编译器产生一个机器语言程序，该机器语言程序具有原来的C程序指定的功能，原来的C程序称为源程序，而相应的机器语言程序称为目标程序。源程序和目标程序指定的操作步骤相同，只是源程序用高级语言编写，而目标程序用机器语言表示。

对程序进行了正确的编译之后，还需要执行附加的操作步骤，才能使目标程序成为可执行程序。该步骤包括将其他的机器语言形式的语句链接到现有的目标程序中，然后将程序装载到内存。完成了链接和装载操作步骤之后，计算机接着就执行程序中的命令。在该阶段会产生新的错误，称为**执行错误**、**运行期错误**、**逻辑错误**或**程序缺陷**。执行错误通常会导致程序的中止。例如，程序中的语句可能会试图执行一个除以0的操作，这将产生一个执行错误。有一些执行错误不会引起程序的执行中止，而是产生错误的计算结果，这种类型的错误可能是编程人员在确定正确的解题步骤时出错以及程序所处理的数据有错误造成的。如果执行错误是由于程序中的语句错误造成的，则需要修改源程序中的错误，然后再次重新开始编译步骤。即使程序似乎能够正常执行，我们也必须仔细检查执行结果，确保它们是正确的。计算机将严格按照我们指定的操作步骤执行。如果我们指定了错误的操作步骤，计算机将会执行这些错误的（但语法上是合法的）操作步骤，并提供错误的执行结果。

图1-4表明了编译、链接/装载和执行的过程。将汇编语言程序转换成二进制程序的过程由**汇编程序**实现，相应的转换过程称为**汇编**、**链接/装载**和**执行**。

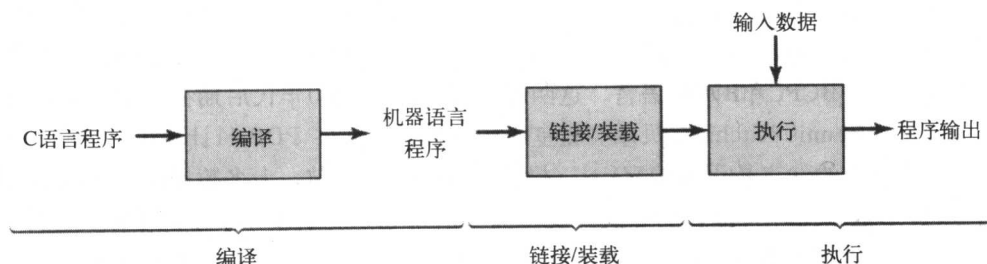


图1-4 程序的编译、链接/装载和执行

5. 执行MATLAB程序

在MATLAB环境中，我们可以编制并执行包含MATLAB命令的程序或脚本。我们还可以执行一个MATLAB命令，观察其执行结果，然后执行另外一个MATLAB命令，与内存中的信息进行交互，观察其执行结果，等等。这种**交互环境**不需要像高级计算机语言那样的正式的编译、链接/装载和执行过程。但是，当MATLAB环境试图对命令进行翻译的时候，MATLAB命令中的语法错误会被检测出来；而当MATLAB环境试图执行命令的时候，逻辑错误将导致执行错误。

6. 软件生命周期

1955年，典型的计算机解决方案的开发成本被划分成以下几部分进行估算：15%的软件开发和85%的相关计算机硬件。随后，硬件的成本急剧下降，而软件成本开始上升。在1985年，这些估算结果发生了根本上的变化，85%的成本用于软件，而15%用于硬件。一个计算机解决方案的主要成本在于软件的开发，人们对了解一个软件开发方案给予了极大的关注。

一个软件项目的开发通常遵循一定的步骤，或分为若干周期，总称为**软件生命周期**。一般情况下，这些步骤包括项目定义、详细说明、编码和模块测试、集成测试以及维护。表1-2中的数据表明了每一个阶段工作量的百分比估算值。从这些估算值中可以清楚地看出，软件维护占据了一个软件系统成本的大部分。这种维护包括软件功能的增强，修复软件使用过程中发现的错误，使软件与新的软硬件环境相适应。维护的难易程度直接和解决方案开始时的

定义及说明有关,因为这些步骤为项目的其他步骤奠定了基础。在下一节中所介绍的解题步骤强调了开始编码或测试之前对解决方案的定义和说明的重要性。

表1-2 软件生命周期各阶段

生命周期	工作量百分比	生命周期	工作量百分比
定义	3%	集成测试	8%
说明	15%	维护	60%
编码和模块测试	14%		

在减少软件开发时间和降低软件开发费用等成本方面的成功技术之一是软件原型的开发。在软件生命周期的早期开发一种系统的原型来让用户使用,而不是直到整个软件系统开发出来后再让用户使用。这种原型并没有最终软件所要求的全部功能,却允许用户在软件生命周期的早期使用它,并对软件说明进行必要的修改。在生命周期的早期进行修改,将有利于降低费用和时间成本。由于MATLAB具有强大的命令和图形功能,因此在开发软件原型方面尤其有效。一旦MATLAB原型正确地执行了所希望的操作,而且用户对“用户-软件”的交互比较满意,那么最终的解决方案可能就是MATLAB程序,或者最终的解决方案可能被转换成另一种语言,以便在某种特殊的计算机或仪器上执行。

作为一名工程师,你很有可能需要对现有软件进行修改或添加功能。如果现有软件结构良好,可读性强,与软件配套的文档编写清楚而且是最新的,那么这些修改就非常简单。即使使用像MATLAB这样功能强大的工具,编写结构良好且可读性强的代码也是非常重要的。由于这些原因,因此我们特别强调养成良好的习惯,使软件更具可读性并保证配套文档齐全的重要性。

1.3 一种工程问题的解题步骤

解题不仅仅是工程课程中的一个关键部分,而且也是计算机科学、数学、物理和化学课程中的一个重要部分。因此使用统一的方法解题是非常重要的。如果这种方法对所有这些不同的领域足够通用,也是很有帮助的,这样我们就不需要为解决数学问题去学习一种技术,而为解决物理问题学习另一种技术,等等。我们为工程问题所提供的解题技术也适用于解决其他领域的问题。但是,它确实假设我们正在使用计算机帮助解题。

本书中所使用的解题过程或步骤有以下五步:

- 1) 清楚地陈述问题。
- 2) 描述输入/输出信息。
- 3) 使用一组简单的数据对问题进行手工计算(或使用计算器)。
- 4) 开发一个MATLAB解决方案。
- 5) 使用大量数据测试该解决方案。

下面以从一个物理实验室的实验中收集来的数据为例讨论这些步骤。假设我们已经通过一台正在工作的仪器上的传感器收集了一些温度数据,在5分钟的实验期间,每隔30秒测量一次温度。我们要计算平均温度,并通过图形表示温度值。

1.3.1 陈述问题

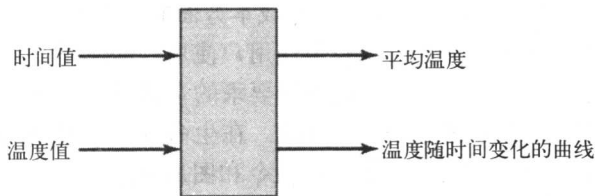
第一步是清楚地陈述问题。给出一个有关问题的清楚而简明的陈述,以免引起误解,这

是非常重要的。对于本例，问题的陈述如下：

计算一系列温度的平均值。然后画出温度随时间变化的曲线。

1.3.2 描述输入/输出

第二步是详细描述给定的用来解题的信息，然后明确要计算的值。这些项目表示了问题的输入和输出，总称为**输入/输出**，或称**I/O**。对于许多问题来说，画一张表示输入和输出的图是很有用的。在这里，程序被称为**抽象**，因为我们并没有定义确定输出的步骤，而只表明了用来计算输出结果的信息。本例的I/O图如下：



1.3.3 手算示例

第三步是使用一组简单的数据通过手算或计算器解决问题。这一步骤非常重要，不能省略，即使对于简单问题也是如此。在该步骤中制定出解决问题的详细步骤。如果你不能采用一组简单的数据来计算输出（手算或使用计算器），则说明还没有为下一步工作做好准备。你应该重新审阅问题，或者参考相关资料。对于本题，惟一的计算是求一系列温度的平均值。假设我们使用下表中的数据作为手算示例：

时间（分）	温度（°F）
0.0	105
0.5	126
1.0	119

通过手工计算，平均值为 $(105 + 126 + 119) / 3$ ，也就是 $116.666\ 7\ ^\circ\text{F}$ 。

1.3.4 MATLAB解决方案

在使用一组简单的数据解决问题之后，就可以开始设计**算法**了，算法就是解决问题的具体步骤。对于如本例这样简单的问题，算法可以使用MATLAB命令立即写出。对于更复杂的问题，有必要首先编写一个操作步骤的大纲，然后将这些操作步骤分解，直到可以用MATLAB命令写出。MATLAB的一个强大功能就是它的命令同我们用来解决工程问题的步骤非常接近。因此，确定解题步骤的过程也就确定了MATLAB命令。可以看出，以下的MATLAB操作步骤和手算示例的解决方案的步骤非常接近。

```
% Compute average temperature and
% plot the temperature data.
%
time = [0.0, 0.5, 1.0];
temps = [105, 126, 119];
average = mean(temps)
plot (time,temps),title('Temperature Measurements'),
xlabel ('Time, minutes'),
ylabel ('Temperature, degrees F'),grid
```