

实用科学与工程计算方法

角仕云 刘丽娅 编著

科学出版社

实用科学与工程计算方法

角仕云 刘丽娅 编著

科学出版社

2000

内 容 简 介

本书介绍了现代工程领域和科研过程中的常用计算方法及其结构化程序设计方法。全书共九章。第一至第七章系统地介绍了数值积分法、数值微分法、非线性方程或方程组的数值解法；第八章介绍计算方法的具体应用——数据处理技术；第九章简要总结了 BASIC 语言的用法与上机指南。书中将计算方法与程序设计有机结合，编写了十几个高质量的通用子程序，供读者调用和移植，其中每一个程序都按标准化、规范化、模块化、结构化的要求设计而成，其步骤明确、结构清晰、易学易懂、实用性强，而且详细介绍了编写程序的实用方法和技巧，并配有联系实际的例题与习题。本书提供的有关数值计算的通用子程序均以任何计算机共有的 BASIC 语句为基础，读者不做任何改动即可调用或移植。

本书可作为高等理工院校非计算数学专业的本科生、研究生及成人教育的教材，也可作为广大科技工作者的自学读本或参考资料。

图书在版编目(CIP)数据

实用科学与工程计算方法/角仕云、刘丽娅编著. -北京:科学出版社, 2000

ISBN 7-03-007465-3

I. 实… II. ①角… ②刘… III. ①计算方法-应用-科学研究 ②计算方法-应用-工程技术 IV. N32

中国版本图书馆 CIP 数据核字(1999)第 09792 号

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 1 月第 一 版 开本: 787×1092 1/16

2000 年 1 月第一次印刷 印张: 14 3/4

印数: 1—3 500

字数: 327 000

定价: 22.00 元

(如有印装质量问题,我社负责调换(环伟))

前 言

党的十五大提出,科技腾飞必须以科教为基础,人才培养是 21 世纪发展战略中的重中之重。而现代高等教育的原则是“厚基础、宽专业、少学时”与“素质、知识、能力”并重。也就是说,要在较少的学时内把学生培养成为基础扎实、专业面宽、应变能力强的现代工程型人才。这一要求不但给高等学校的一线教师提出了挑战(少学时),也为高等教育的专业调整与课程设置指明了方向(厚基础、宽专业)。

本书就是为解决上述问题而编写的,全书内容深入浅出,取材精练,以实用为目的。编写过程中我们始终坚持“加强基础、面向实际、便于自学、引导思考、启发创新”的原则,即遵循“符合面向 21 世纪教育的教材精神”:根据本课程和相邻学科的发展认识,对旧有课程体系进行了革新和发展;注重“加强基础和实践”环节,为使教材“便于自学、启发思维、开拓创新”方面做了大量的工作;与现代化教学手段(计算机)相结合,体现了采用现代化教学手段提高教学质量的原则。

本书实际上是“计算方法”与“计算机程序设计语言”两门课程交叉的产物,算法设计与程序设计相互辉映,步骤明确、结构清晰、易学易懂且实用性强。在本书编写过程中,编者在多年教学、科研和工程实践的基础上,参考了国内外较新的有关专著和文献。因此,本书是编者多年教学工作和科研实践经验的总结,书中的每一个程序都按标准化、规范化、模块化、结构化的要求设计而成,每一个程序中都包含着高超的程序设计艺术与技巧,希望读者在学习中认真体会。本书的阐述简明扼要、深入浅出,并安排有联系实际的例题与习题,实用性强,可读性好,便于自学,是一部内容丰富、范围广泛、实用性强、导引功能突出且涉及最新发展的教科书。

电子计算机的问世开始了“工业化之后的技术与信息革命”,在这场技术与信息革命之前,数值计算方法与数据处理技术受到老式计算工具的限制,那时,计算方法和数据处理主要依靠手册、计算尺、算盘、图形关联和列线图等传统工具。如今工程师的这些传统工具并没有作废,但逐渐被计算机所补充和代替。一个人可能被过于浩瀚的数据或过于复杂的计算问题弄得不知所措而不能将其消化。掌握数值计算方法和数据处理技术是推动和加速现代科学技术发展的关键所在,而使用计算机解决现代科学技术领域中日趋复杂的问题则是加速科学技术发展的重要手段。掌握程序设计语言和具备使用计算机解决实际问题的能力已成为现代化知识分子的重要标志之一。

工程技术的目的在于应用科学知识来提高现实生活水平。到 21 世纪,工程师和科学家将要解决哪些问题,并没有人确切知道,但可以列出一些可能的问题,如星际旅行的生命保障系统,在地球上将消费品进行循环,从而保存自然资源的未来化工厂,设计目前尚未发明的物质的加工过程等。将来的工程领域,无论它们将会怎样,都必须遵循自然界的物理或化学定律。目前的工程领域或科研工作中所遇到的许多实际问题,已复杂到人工无法解决的程度,将来还会更加复杂,幸好计算机的应用已渗透到各工程领域或科研过程。本书将试图以不带任何偏向的观点来介绍计算机的数值计算能力和它的局限性。

使用计算机进行科学计算时，我们最关心的是计算机的运算能力。当前的计算机就其本身的运算实质来说，它所能进行的仅仅是加、减、乘、除四则运算和简单的逻辑运算。一般数学理论中所讨论的解析方法，例如微分方程组的通解、积分的原函数，函数的导数等等，这些运算及其算法是计算机无能为力的。但是，如果所求的是过程的近似数值解，那么，计算机是完全可以胜任的，而且一般可以满足预先提出的精度要求。这正是本书所要完成的任务之一。

数值计算方法与数据处理技术是数学学科的两个古老的分支，同时又是两门新兴的学科。电子计算机的问世和发展，大大地促进了数值计算方法和数据处理技术的提高。现在，数值计算方法已深入到各科学技术领域。为了适应和满足 21 世纪高等教育“教学与科研”的需要，本书结合数值计算方法编写设计了几十个高质量的通用子程序，以供读者随时调用和移植。本书内容深入浅出，取材精练，以实用为目的。书中的每一种算法，都是简要地阐明数学原理，讨论方法与步骤，给出框图，再依据框图编写设计通用子程序，最后结合一个具体实例作为应用通用子程序的示范，希望对读者设计其它应用程序能起到举一反三的作用。本书中的全部通用子程序都是使用目前比较流行的 BASIC 语言编写而成的，而且所使用的语句是各种计算机的 BASIC 版本所共有的，书中全部程序都分别在 IBM 系列微机及其兼容机上调试通过，不作任何改动就可以在目前国内流行的大部分计算机上运行通过。

本书中的全部程序都采用标准化、模块化、结构化设计，思路清楚，结构清晰，易学易懂且通用性强。因此，本书不但是一本以实用为目的的推广并提高计算机应用能力的教科书，而且也是一本很好的提高程序设计艺术与技巧的参考资料，读者不但可以从中学到解决实际问题的通用程序，而且可以从中学到结构化程序设计的方法、手段、艺术与技巧，提高程序设计的质量，最终达到培养提出问题、分析问题、解决问题的综合动手能力的目的。

本书由云南工业大学青年教师角仕云和刘丽娅等编著，其中第一章、第四章至第八章由角仕云编写，第二章、第三章和第九章由刘丽娅编写，全书由角仕云统稿。另外，廖学品、赵之平、罗康碧、李沪萍、祝玲华、李国兵、贾愚、余建川等八位教师参加了本书部分章节的编写整理和程序调试工作。牛存镇教授和廖学品副教授在百忙中对本书全部书稿进行了认真的审阅并提出了一些宝贵的修改意见。在本书的编著过程中，编者查阅了大量的参考资料，在此对所有参考资料的作者和本书主审牛存镇教授及廖学品副教授表示衷心的感谢。

由于编者阅历不深，水平有限，书中难免有不妥之处，恳请读者来函批评指正。

编者

1999 年 3 月

于云南工业大学

关于本书的说明

本书是关于如何用微型计算机来解决数学问题的。假定读者已学过微积分和一些矩阵理论、微分方程,而且有机会接触中小规模的电子计算机,并能用手头版本的 BASIC 高级语言或者 FORTRAN 高级语言编写源程序。

BASIC 或 FORTRAN 子程序是本书的重要组成部分之一。实际上,本书也可以认为是一本广义的关于子程序的《用户指南》。这些子程序不像在许多教科书中所能见到的那种简单、示范性的程序,而是代表了现代科学计算的状况与水平。为了不仅使计算机能够接受这些子程序,而且读者也可以读懂它们,因而在书中列出了程序清单并作了详细的解释。而且,编者也确实希望这些子程序能够用来解决现代工程技术与科学研究中所遇到的实际问题。

至于书中的表示法,编者采用了一种完全与众不同的形式。因为多数工程师和科学家既没有时间,也没有兴趣经常翻阅当今有关计算方法的文献资料,而且编者注意到许多人在解决实际数学问题时,不是靠程序就是靠在以前课程中所用的方法,所以,编者打算提供一些高质量的通用子程序来帮助读者解决所遇到的数学问题。

在这些通用子程序中,有许多是很复杂的,要详细说明它们如何工作以及为什么这样工作的原因,需要的时间就会比多数读者所能够用在这种教材上的时间多得多。因此有必要把许多这样的子程序当作“黑盒”来对待,而且这也是常用的办法。大多数程序设计人员都把计算机硬件子程序,例如加法和乘法子程序以及各种标准函数子程序当作“黑盒”来对待,没有多少程序设计人员详细地了解它们,只是把它们作为有用的通用子程序接受下来。而这也正是编著者的意图,即现在需要读者把解决线性系统、常微分方程组系统和许多其它标准化数学问题的通用子程序当作通常是有效的基本子程序接受下来,而不必去详细了解它们是如何工作的。

然而,只是简单地解释如何调用每个通用子程序并指出它们“一般是有效的”是不够的,在计算方法中有许多易犯的错误,读者必须严防这些错误的发生。应当学会发现数值计算的错误发生的征兆以及正确地解决这些问题的方法。为此,就需要对在各种子程序中所使用的数值计算方法有一定程度的了解。程序设计人员有时也得去修改一个子程序或把一种算法稍加改动来解决一个有关的实际问题,而这往往需要作更进一步的了解。

编者尽量合理地安排了各个方面的内容和细节,以便读者能够逐步地把本书中的子程序看成是“透明的黑盒”乃至灰盒,在适当之处,编者还列出了参考文献,以便读者对某个特定的具体问题作更深入的探讨。

由于本书中只涉及到本书书名的一部分基本而实用的内容,因此每章后都列出与该章相关的参考资料,读者可以从这些参考资料中查阅到本书中没有列出的算法。

本书中共编入 37 个 BASIC 通用子程序和 5 个 FORTRAN 通用子程序。编者希望在学习本课程的同时和之后,这些子程序会得到广泛的应用。这些子程序都是仔细编写的,所以只要稍作修改或根本不用修改就可以方便地在目前国内流行的大多数计算机上运行。

近几年来,人们更加深刻地认识到编写、调试和推广高质量的科学计算软件的重要性。因此,尽管编者认为本书中的子程序是现阶段的佼佼者,但是编者希望,并且期待它们最终被淘汰掉。我们编写本书的目的之一就是帮助读者寻找更好的、质量更高的程序,并且在这些程序可资利用时去认识它们、评价并改进它们。

本书给出的算法是许多数值分析专家多年的研究成果和经验总结,解决每一类问题的数值计算方法,往往有几十种,在编写程序时,编者曾慎重地考虑哪些算法可列入本书,哪些算法通过参考资料给出。另外,在编写通用子程序和注释时,编者还稍微统一了格式。读者在弄清编者编写程序时的某些约定之后,就能更好地使用这些通用子程序,而这正是本书的目的之一。

今天,程序设计人员在应用程序时所遇到的一个主要问题就是计算机程序的移植性问题。的确,开发像 BASIC 和 FORTRAN 之类的计算机高级语言的目的之一就是要使程序能够在各种不同的计算机上运行。但是,尽管在这些语言的标准化方面作了很大的努力,程序的移植性问题仍然存在。制造计算机的各个厂家都有自己的 BASIC 版本。

本书在设计通用子程序时,考虑了目前国内流行的各种微型计算机的 BASIC 版本,使用的是各种计算机的 BASIC 版本所共有的语句。因此,本书中的全部通用程序,不作任何改动就可用于目前国内流行的大部分计算机,例如,IBM 系列微机及其兼容机、联想系列微机以及 AST 系列微机等,因此,本书中的全部程序都克服了移植性问题,这是本书的目的之二。

本书是编者多年从事教学、科研和工程实践工作和实践经验的总结,书中的每一个程序都按标准化、规范化、模块化、结构化的要求设计而成,每一个程序中都包含着高超的程序设计艺术与技巧,希望读者在学习中认真体会。要掌握程序设计的艺术与技巧,只学习程序设计语言的语法规则和算法是不够的,最好的办法是分析一些他人设计好了的高质量的通用程序中所用的方法与技巧。该书中的程序可用“麻雀虽小,五脏俱全”来概括其特点,即程序既短小精干,又能解决问题。

最后简单说明本书在程序设计时的一些约定:

1. 行号 1~999 留给读者编写主程序。
2. 行号 1000~1999 是本书约定为函数子程序所用的行号,虽然也是通用子程序的一部分,但也需要读者根据实际要解决的问题填上具体的内容(如函数或数据)。
3. 行号 2000~9999 为本书通用子程序所使用的行号,除个别有特殊需要之外,这些内容一般不需要读者改动。
4. 关于本书中框图的说明
本书中所涉及的每一种算法的程序框图,都由三部分组成,其中:
A: 通用子程序框图;
B: 函数子程序框图;
C: 调用通用子程序的主程序框图。

本书程序清单

一、BASIC 程序清单

PROG0:	常用递推结构使用说明程序	10
PROG1:	梯形积分通用程序	21
PROG2:	辛普森积分通用程序	29
PROG3:	龙贝格积分通用程序	34
PROG4:	四种积分比较(梯形, Simpson, Cotes, Romberg)	37
PROG5:	三种积分比较(梯形法, 中点法, Simpson 法)	39
PROG6:	利用差商求微分通用程序(三种)	45
PROG7:	利用插值公式求微分	49
PROG8:	求微分通用程序	52
PROG9:	直接迭代法解非线性方程程序	63
PROG10:	加速迭代法解非线性方程通用程序	67
PROG11:	对分区间法求非线性方程的单根通用程序	71
PROG12:	对分区间法求非线性方程在 $[A, B]$ 内的所有实根通用程序	75
PROG13:	牛顿迭代法解非线性方程通用程序	79
PROG14:	快速弦线法解非线性方程通用程序	84
PROG15:	简单高斯消去法解线性代数方程组通用程序	103
PROG16:	列主元高斯消去法解线性代数方程组通用程序	107
PROG17:	列主元高斯-约旦消去法解线性代数方程组通用程序	112
PROG18:	高斯-赛德尔迭代法解线性代数方程组通用程序	117
PROG19:	高斯-雅可比迭代法解非线性方程组通用程序	129
PROG20:	高斯-雅可比-赛德尔迭代法解非线性方程组通用程序	131
PROG21:	最速下降法解非线性方程组通用程序	133
PROG22:	欧拉折线法解常微分方程通用程序	145
PROG23:	欧拉二次逼近法解常微分方程通用程序	147
PROG24:	四阶龙格-库塔法解常微分方程通用程序	151
PROG25:	欧拉折线法解常微分方程组通用程序	155
PROG26:	改进欧拉法解常微分方程组通用程序	159
PROG27:	四阶龙格-库塔法解常微分方程组通用程序	161
PROG28:	四阶龙格-库塔法解高阶常微分方程通用程序	165
PROG29:	计算方法综合应用第一示范程序	173
PROG30:	计算方法综合应用第二示范程序	175

PROG31: 一元线性回归分析通用程序.....	181
PROG32: 多元线性回归分析通用程序.....	186
PROG33: 加权多元线性回归分析通用程序.....	191
PROG34: 非线性回归分析第一示范程序(指数回归).....	195
PROG35: 非线性回归分析第二示范程序(几何回归).....	197
PROG36: 非线性回归分析第三示范程序(多项式回归).....	198
PROG37: 非线性回归分析第四示范程序(特殊例子).....	201

二、 FORTRAN 77 通用程序清单

SIMP: 自动变步长辛普森积分通用程序.....	31
KSXX: 快速弦线法解非线性方程通用程序.....	86
LGAUSS: 列主元高斯消去法解线性代数方程组通用程序.....	118
ZSXJ: 最速下降法解非线性方程组通用程序.....	134
RGKT: 四阶龙格-库塔法解常微分方程组通用程序.....	165

目 录

前言

关于本书的说明

本书程序清单

第一章 绪 论	1
§ 1.1 本书的任务与性质	1
§ 1.2 程序设计语言概述	3
§ 1.3 算法概述	8
§ 1.4 程序设计中的常用工具	10
§ 1.5 用计算机进行科学计算时应注意的问题	13
习 题	15
参考资料	15
第二章 数值积分法	16
§ 2.1 概述	16
§ 2.2 数值积分的基本思想	17
§ 2.3 梯形积分法	19
§ 2.4 辛普森积分法	27
§ 2.5 龙贝格积分法	32
§ 2.6 FORTRAN 通用程序设计	37
§ 2.7 几种常用积分方法的比较	37
§ 2.8 关于本章例题的说明及数值积分方法评述	39
习 题	41
参考资料	41
第三章 数值微分法	43
§ 3.1 概述	43
§ 3.2 微分的差商数值解法	44
§ 3.3 利用插值公式求数值微分	48
§ 3.4 实用的通用微分法	51
§ 3.5 高阶导数的数值解法	55
§ 3.6 数值微分方法评述	56
习 题	56
参考资料	56
第四章 非线性方程的数值解法	58
§ 4.1 概述	58

§ 4.2 根的初值和存在范围	59
§ 4.3 简单迭代法 (直接迭代法)	60
§ 4.4 加速迭代法	66
§ 4.5 对分区间法 (二分法)	70
§ 4.6 牛顿切线法 (牛顿迭代法)	78
§ 4.7 弦线法	81
§ 4.8 方程求根方法评述	88
§ 4.9 方程求根的有关方法评述	90
习 题	91
参考资料	91
第五章 线性代数方程组的数值解法	93
§ 5.1 概述	93
§ 5.2 上三角形线性代数方程组的数值解法	95
§ 5.3 简单高斯消去法	96
§ 5.4 列主元高斯消去法	107
§ 5.5 列主元高斯-约旦消去法	111
§ 5.6 解线性代数方程组的迭代法	115
§ 5.7 FORTRAN 通用程序设计	120
§ 5.8 线性代数方程组的数值解法评述	122
习 题	123
参考资料	125
第六章 非线性代数方程组的数值解法	126
§ 6.1 概述	126
§ 6.2 高斯-雅可比迭代法	127
§ 6.3 高斯-雅可比-塞赛尔迭代法	130
§ 6.4 最速下降法	132
§ 6.5 非线性方程组的数值解法评述	138
习 题	139
参考资料	139
第七章 常微分方程及方程组的数值解法	141
§ 7.1 概述	141
§ 7.2 常微分方程的离散化方法	142
§ 7.3 一阶常微分方程的数值解法	144
§ 7.3.1 欧拉折线法	144
§ 7.3.2 欧拉二次逼近法 (梯形规则)	147
§ 7.3.3 龙格-库塔法	150
§ 7.3.4 前面三种数值解法的比较	153
§ 7.4 一阶常微分方程组的数值解法	154
§ 7.4.1 欧拉折线法	154
§ 7.4.2 欧拉二次逼近法	157

§ 7.4.3 四阶龙格-库塔法.....	160
§ 7.4.4 上述三种数值解法的精度分析.....	163
§ 7.5 高阶常微分方程(组)的数值解法	164
§ 7.6 FORTRAN 77 通用程序设计.....	167
§ 7.7 常微分方程组的数值解法评述.....	169
习 题	169
参考文献.....	170
第八章 计算方法的应用分析: 数据处理技术简介.....	171
§ 8.1 计算方法的应用综述	171
§ 8.2 数据处理技术综述	177
§ 8.3 一元线性最小二乘法	178
§ 8.4 多元线性最小二乘法	184
§ 8.5 加权多元最小二乘法	190
§ 8.6 非线性最小二乘法简介	193
§ 8.6.1 可化为线性拟合的非线性回归分析.....	194
§ 8.6.2 直接求解非线性回归分析问题.....	204
习 题	204
参考资料	205
第九章 BASIC 语言概要.....	207
§ 9.1 BASIC 语言中的运算量、运算符和表达式.....	207
§ 9.2 BASIC 语句功能分析.....	209
§ 9.3 BASIC 上机操作指南.....	217
附录: 相关系数 R 检验临界值表.....	221
参考资料.....	221

第一章 绪论

§1.1 本书的任务与性质

随着科学技术的发展,大量复杂的数学计算问题也随之摆到了人们面前。在计算机没有问世以前,为了解决某些复杂的数学计算问题,不少科学家为之献出了大半生甚至是毕生的精力。例如,18世纪法国天文学家 Dalmny 花了 20 多年的时间,为的是求解一个天体运动的摄动级数问题,以求解月球运行轨道的数值解。还有一些复杂的计算问题,本来就无法用人工计算去完成,例如,1948 年美国原子能研究中的一项课题,需要做 900 万道计算,大约相当于 1500 名工程师一年的计算工作量。不难看出,以上所列举的计算问题,都有这样一个特点,即都是人工计算所难于完成或者根本无法胜任的计算,在现代工程领域或科研过程中,像这样复杂的计算问题不胜枚举。

研究使用计算机解决科学计算的方法及其理论的科学,称为科学计算方法或数值计算方法。数值计算方法(Numerical calculus method)包括数值积分(Numerical integration)、数值微分(Numerical differentiation)、方程求根和方程组的数值解法等内容。即数值计算方法是以科学计算中的数学问题为研究对象的,因此,数值计算方法是数学的一个分支。但数值计算方法不像一般数学那样研究数学本身的理论,而是着重研究求解科学计算问题的数值解法及其稳定性(Stability)、收敛性(Divergence)与计算的误差分析(Error analysis),其最显著的特点就是由复杂的公式组合求过程的数值解(Numerical solution)。

现代工程领域或科研过程中遇到的另一类问题是如何由成批的已知数据求一些特定要求的数值解,或者如何把这些成批的数据转化为各变量间有内在联系的公式,这就是数据处理技术(Data processing technique)。数据处理技术包括离散点下的数值积分、离散点下的数值微分、方差分析(Analysis of variance)、曲线拟合或曲线回归(Curvilinear regression or fitting)、插值(Interpolation)等内容,因此,数据处理技术也是数学的一个分支。但数据处理技术也不像一般数学那样研究数学本身的理论,而是着重研究如何把成批的数据转化为真实反映各个变量之间的本质规律的“经验公式(Empirical formula)”,并研究这种转换的可信程度:方差分析或者称为显著性检验,当然,这种转换所得到的公式对我们是相当有用的,例如,用于其它过程的计算,求过程的导数、积分等等。

程序设计(Programming)是确定解题方式,设计程序框图,并用计算机高级语言编写源程序的过程,它是以程序框图和计算机源程序为研究对象的。

实用科学与工程计算方法及其结构化程序设计是计算方法与程序设计两门课程交融而成的课程,它把这两门课程有机地结合在一起,融计算方法与结构化程序设计于一体,成为现代科技工作者不得不掌握的一门实用性很强的技术基础课程。本课程的研究对象

是科学计算中常用的算法及其程序框图和结构化的高级语言源程序，而本课程的目的则是以源程序的方式给出这些常用算法的通用子程序，并教会读者使用这些通用子程序。

研究、学习实用科学与工程计算方法及其结构化程序设计，对于充分发挥计算机在科学与工程计算中的作用，对于提高程序设计的技巧和艺术以及正确使用计算机解决科学与工程计算问题，都有着十分重要的意义，其重要性主要表现在：

一、讲究方法，提高程序设计的质量

一般来说，计算机上通不过的程序当然是坏程序，但能通过的程序未必就是好程序，程序是有优劣之分的。那么，如何评价一个程序质量的优劣呢？一般而言，程序设计的质量可以从以下几个方面考虑：

1. 能够解决问题

这就是说，任何程序，都必须保证能够获得正确的结果，才算是合格的程序。这是对程序质量的最基本的要求。但进行程序设计时绝不能只满足于能得出正确的结果，还应该考虑如何提高程序的运行速度，如何节省计算机内存空间等问题。

2. 计算量要小

达到相同的精度要求，计算量越小，程序的质量就越高。例如，解线性代数方程组的经典方法是克莱姆（Cramer）法则，运用这个法则解线性代数方程组，解的每一个分量都可以表示成两个行列式之比。它们的分母行列式相同，而分子行列式不同。假如要用克莱姆法则解一个有 20 个未知数的线性代数方程组，就得计算 21 个 20 阶行列式。这样和就有 $20!$ 项之多，而且每一项都要作 19 次乘法运算。因此，要解这个线性代数方程组就得进行 $21 \times 20! \times 19 = 21! \times 19 = 2 \times 10^{20}$ 次乘法运算，还要进行同等数量级的加法运算。现代大型高速计算机每秒钟大约能进行 10^8 次运算，即使用这样的大型高速计算机求解上述 20 元方程组，也需要连续计算大约 5×10^{14} 年，而且在计算过程中还不能发生故障。可见，用克莱姆法则解线性方程组的计算方法虽然总可以求出方程组的解，但当自变量的个数较多时，这一方法的计算量是大得惊人的。一般地，用克莱姆法则解一个具有 n 个未知数的线性代数方程组，只乘法运算就要进行 $(n+1)n!(n-1)$ 次。

然而，有更好的解线性代数方程组的方法，运用上佳的计算方法解上述方程组，只需用很少的机时就能解出所有的未知数，所以选择计算量小的计算方法是十分重要的。

3. 存储量要小

用计算机进行计算时，需占用的内存单元数称为相应算法和程序的存储量要求。如何降低存储量和节省内存空间将在以后各章结合程序设计加以讨论。

4. 程序结构简单、思路清楚、便于阅读、学习和使用

5. 通用性强，易于移植

一个好的程序应该具有通用性强和易于移植的特点。例如不作任何改动即可在大部分流行的计算机上运行。这是对程序质量的最高要求。关于程序设计的质量要求，将在以后各章结合实际问题的逐点加以讨论。

二、把计算机上不能进行的计算转换成能进行的计算

我们知道，计算机只会做加、减、乘、除等四则运算，严格地说计算机只会做加减

法。但科学与工程计算中所遇到的实际计算问题，并非都是四则运算所能解决的，往往是种类繁多、过程极其复杂的计算。例如微分、积分、解微分方程组等，这些计算都是计算机无能为力的，然而，如果我们使用适当的计算方法，根据问题的性质与定义，按一定的规则处理后，就可以把计算机不会做的各种运算转换为等价的四则运算过程，从而用计算机间接求出计算结果。例如定积分是计算机不会做的运算，但是，如果我们根据定积分的几何意义把定积分计算转换成如下的等价形式：

$$I = \int_a^b f(x)dx = \sum_{i=1}^n f(x_i)\Delta x_i$$

当 n 取得足够大时，即可获得足够精确的积分近似值，这是计算方法的研究内容之一。

综上所述，我们可以清楚地看到，在解决实际问题时，选择或设计合适的数值计算方法对于提高程序的质量，对于充分发挥计算机在科学计算中的作用，都有着极其重要的意义。而在学完算法语言（计算机程序设计语言）之后，不失时机地学习数值计算方法及其程序设计课程，提高设计或选择算法的能力，掌握程序设计的艺术与技巧，具备使用计算机解决实际问题的能力，确实是十分必要的。

科技工作者在使用计算机时，如果不具备设计或选择合适的数值计算方法的能力，不会编写程序，那么，他处理问题的范围将是十分有限的，这样就限制了他自身潜在能力的发挥。反之，如果懂得如何选择或设计算法，并能用一种或多种高级语言把这种算法设计为高质量的计算机高级语言源程序，那么，他就可以对现成的各种算法或程序作出正确的选择、改进和移植，甚至设计新的算法和程序去分析和解决实际问题。这样就能充分发挥自身的潜力，在广泛的工程技术或科研领域内解决多种多样的复杂问题。

通过本课程的学习应达到以下几点要求：

1. 了解各种常用算法的优劣，并会选择、改进或设计算法；
2. 掌握求解数值积分、数值微分、非线性方程或方程组的数值解法、常微分方程或方程组的数值解法等常用数值计算方法；
3. 会设计常用算法范围内的源程序，并能借助计算机解决常用算法范围内的实际问题。

一个科技工作者，如果能集数学知识、外语知识、计算机知识与专业知识于一身，那他就能在广泛的科学与工程领域内解决复杂多变的实际问题。具有这四方面的知识是现代化知识分子的四大标志，无论缺乏其中的哪个方面，其解决问题的能力都会受到限制。作为本书的读者对象，虽然在数学知识与计算机知识方面有一定要求，但没有要求他们是数学分析专家与程序设计专家。

§1.2 程序设计语言概述

一、计算机是实现算法的高效工具

电子计算机通常简称为计算机（Computer），也有人称之为电脑。

说计算机是能进行计算的机器设备，这容易束缚人们对计算机应用潜力的理解与想象。比较合理的提法是：计算机是信息处理机（Information processing machine）。

其主要功能是信息处理和加工。信息的一种载体就是我们经常遇到的也是经常要处理的数据,数据可以分为数值数据和非数值数据两种。数值数据我们比较熟悉,例如 3.14159, 2.71828 等,计算机可以用于处理数值数据,如求方程的数值解,求数值积分等,这就是人们通常所理解的计算机可以用来作数值计算或科学计算。然而,一本书的书名或者一台设备的型号,如本书的书名《实用科学与工程计算方法》、计算机型号“IBM”等等,这些也是数据,称为非数值型数据。计算机也可以用来处理非数值数据,如图书情报检索、人事管理、工资管理、自动图形识别等,这些称为计算机的非数值应用。计算机的数值应用与非数值应用既有区别又有联系,它们共同构成了计算机应用的广阔领域。

概括地说,计算机是一个高速操作的、具有记忆和内部存储能力的、由程序自动控制操作的信息加工与处理装置。简单地说,计算机是实现算法的高效工具。

由于计算机的问世,使一些过去难以解决的问题得以有效地解决。例如,著名的世界难题“四色定理”(即在任何一张复杂的地图上,只需用四种颜色就能将各个国家区分开而不会出现相邻国家用同一颜色),过去用了一百多年的时间都没有解决,后来用计算机只花了不到 50 天的时间就证明了这一定理。

现在,从基本粒子研究到宇宙空间探索,从工业生产控制到生产管理,从服装设计到飞机制造,从电影卡通片的制作到家庭游戏机的制造,从程控电话到信息高速公路,无一不用到计算机。可以说,计算机的问世开始了“工业化之后的技术与信息革命”,而且这次技术革命与信息革命是当代科学技术发展中的一次划时代的革命。计算机的迅速发展对科学技术的发展、进步与突破,劳动方式、产业结构以至对人们的生活方式已经产生、正在产生、而且将继续产生重大的影响。现代社会中的任何科技人员、管理人员、乃至一般技术人员,都应当学会使用计算机去解决所面临的实际问题。具有计算机知识和掌握使用计算机的能力是现代化知识分子的一个重要标志。

二、程序设计语言

为了使计算机能够按照我们的要求对给定的“数据”进行预定的加工处理,我们必须给出一组特定的所谓指令(Command),即指示计算机进行操作的语言。这些指令序列就称为程序(Program)。从算法的角度说,编码了的算法就是程序,把预定的任务或设计好了的算法用程序实现的全过程称为程序设计(Programming),编写程序时所采用的一种事先约定用来与计算机交换信息的代码组合形式,就称为计算机程序设计语言。

程序设计语言有三大类:机器语言、汇编语言(或符号语言)和高级语言,下面分别简述它们的基本特点。

1. 机器语言(Machine programming language)

机器语言使用的是计算机自身的指令代码,是面向特定机器的,其优点是可直接执行、执行速度快、占内存空间少;缺点是难学、难记、难写、难检查、难调试、难推广,通用性差。因为各种计算机所用的机器语言是不相同的。

2. 汇编语言(Assembly language)

汇编语言也称为符号语言,它比机器语言前进了一大步,但仍然是一种依赖于具体机型并与数学计算公式的写法差别很大的程序设计语言。用汇编语言编写程序,也需要了解计算机的内部结构,对使用者要求也较高,因此也不易于推广和普及。但是汇编语

言程序与机器语言程序一样，执行速度快、占内存空间少。

3. 高级语言 (Senior algorithmic language)

由于机器语言和汇编语言是面向机器的，可移植性差，而且难学又不易推广和普及，因而被称为低级语言。人们希望能够找到一种既能保持人类的自然语言（英语）的原义，又接近数学公式的表达形式，既易学易懂又在各种计算机上都通用的高级语言。人类的这一设想在 20 世纪 50 年代得以实现，即 1956 年第一种高级语言 FORTRAN 的第一个版本问世了。FORTRAN 是 FORMula TRANslator 的缩写，意为公式翻译器。这种高级语言的表达方式比较接近人们的习惯，人们很容易接受，所以也就比较容易推广，而且不再是面向机器的了，而是面向过程的。例如，用“ $Y = \cos(X) - 2 * X$ ”表示使“Y 的值等于 $\cos x - 2x$ ”，而用“PRINT X, Y”表示“打印 X 和 Y 的值”，这对任何一种计算机都是如此。

从第一种高级语言问世到现在只不过 40 多年的时间，但目前高级语言已发展成为计算机科学的一个重要分支。就高级语言自身来说，至今已经设计出不下三四百种，其中较著名的也有二三十种，例如，仅仅是 BASIC 语言就有基本 BASIC、高级或者称为扩展 BASICA, GWBASIC, Quick BASIC, True BASIC, Turbo BASIC 以及 Visual BASIC 等。目前最常用的高级语言有以下几种：

FORTRAN 语言：世界上最早出版的高级语言，适用于数值计算。

BASICA 语言：是由 FORTRAN 语言改进和提炼出来的一种易学易懂、便于使用又有实用价值的高级语言。其最显著的特点是具有很强的人机对话功能，既适用于数值计算又适用于非数值运算。

Quick BASIC 语言：是在 BASICA 语言的基础上，增加了一些高级功能，例如过程和块 IF……THEN……ELSE 结构等，是结构化程序设计语言，这种高级语言容解释程序和编译程序于一体，并在程序输入的同时进行语法检查，减少了程序输入出错的机会，提高了机器效率，其最大的优点是可以利用计算机的全部内存，程序的执行速度快，无论是功能还是程序的运行速度都比 BASICA 要强和快得多，甚至比 FORTRAN 语言的运行速度还快。

True BASIC 语言：这种高级语言也是由 BASICA 语言改进而来的，它增加了矩阵语句，块 IF……THEN……ELSE 结构和多重选择 SELECT 结构与过程文件等，也是一种结构化程序设计语言。这种高级语言的执行速度和 Quick BASIC 差不多，但两者各有优点。

COBOL 语言：适用于管理领域的高级语言。

PASCAL 语言：最早出现的结构化语言。

dBASE 语言：也叫做大众数据库，主要用于管理领域。

FoxBASE 语言：与 dBASE 类似的数据库语言，主要用于管理领域。

计算机高级语言的出现，使人们不必深入了解计算机的内部结构和工作原理，而把计算机当作一个“黑盒”，只要设计出算法就能很容易地将它用高级语言表示，这样就有可能使一大批非计算机专业的各行各业人员得心应手地使用计算机来解决他们所遇到的实际问题。

世界著名的计算机科学家 With 提出了一个著名的公式来表示程序的实质，即：

$$\text{算法} + \text{数据结构} = \text{程序}$$