普通高等学校非计算机专业计算机基础系列教材

FORTRAN 程序设计

主　编　谭世语　范幸义
编　者　(以姓氏笔画为序)
范幸义　周建丽
郑　昆　谭世语
主　审　孙　蓉

重庆大学出版社

内容提要

本书以标准 FORTRAN 77 为基础，系统介绍了 FORTRAN 语言的基本知识和程序设计方法，并以 MS-FORTRAN 5.1 为蓝本，深入介绍了 MS-FORTRAN 的高级功能及使用方法。为方便读者的使用和提高，书中适量地介绍了 FORTRAN 90 的一些基本扩展功能和用法。全书共分 10 章，包括概述、FORTRAN 语言基础、顺序结构设计、选择结构设计、循环结构程序设计、数组、程序设计的模块化、FORTRAN 的文件操作、MS-FORTRAN 的高级功能、MS-FORTRAN 的绘图程序设计。每章的后面都有大量的习题和上机练习题，供读者温习所学内容和上机实践。

本书是计算机基础教学第二层次教材之一，适合于各大专院校理工类学生，也可作为工程技术人员和科技人员、特别是从事涉及大量数值计算和工程设计人员的参考书。本书也适用于各类人员自学使用。

图书在版编目(CIP)数据

FORTRAN 程序设计/谭世语，范幸义主编. —重庆：重庆大学出版社，2002.8
普通高等学校非计算机专业计算机基础系列教材
ISBN 7-5624-2556-6

Ⅰ.F... Ⅱ.①谭... ②范... Ⅲ.FORTRAN 语言—程序设计—高等学校—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 055565 号

普通高等学校非计算机专业计算机基础系列教材
FORTRAN 程序设计
主　编　谭世语　范幸义
主　审　孙　蓉
责任编辑：肖顺杰　　版式设计：肖顺杰
责任校对：任卓惠　　责任印制：张永洋
*
重庆大学出版社出版发行
出版人：张鸽盛
社址：重庆市沙坪坝正街 174 号重庆大学(A 区)内
邮编：400044
电话：(023) 65102378　65105781
传真：(023) 65103686　65105565
网址：http://www.cqup.com.cn
邮箱：fxk@cqup.com.cn (市场营销部)
全国新华书店经销
自贡新华印刷厂印刷
*
开本：787×1 092　1/16　印张：23.5　字数：587 千
2002年8月第1版　　2002年8月第1次印刷
印数：1—5 000
ISBN 7-5624-2556-6/TP·346　　定价：30.00 元

本书如有印刷、装订等质量问题，本社负责调换
版权所有　翻印必究

顾　问　吴中福　邱玉辉

主　任　陈流汀

副主任　杨天怡　严欣平　张鸽盛

委　员　(以姓氏笔画为序)

王世迪　邓亚平　程小平

杨国才　范幸义　洪汝渝

郭松涛　黄　勤　曾　一

谭世语　熊　壮　莫　堃

此为试读,需要完整PDF请访问: www.ertongbook.com

序言

计算机技术的飞速发展，加快了人类进入信息社会的步伐，改变了世界，改变了人们的工作、学习和生活，对社会发展产生了广泛而深远的影响。计算机技术在其他各学科中的应用，极大地促进了各学科的发展。不掌握计算机技术，就无法掌握最先进、最有效的研究开发手段，将影响到其所从事学科的发展。因此，计算机技术基础是 21 世纪高校非计算机专业大学生必须掌握的、最重要的基础之一。

经过多年的探索和实践，按“计算机文化基础”、“计算机技术基础”、“计算机应用基础”三个层次组织教学已被公认为高校非计算机专业计算机基础教学的基本模式。第一层次开设“计算机文化基础”课程，教学的主要任务是使学生掌握计算机基础知识和基本操作能力；第二层次开设“计算机软件技术基础”和“计算机硬件技术基础”课程，教学的主要任务是使学生掌握计算机软、硬件技术的基本知识和基本开发技术；第三层次按专业群开设“计算机信息管理基础”、“计算机辅助设计基础”、“计算机网络技术基础”、“计算机控制技术基础”等不同课程，教学的主要任务是培养学生应用计算机技术分析解决本学科及相关领域问题的能力。

为了适应计算机技术的飞速发展和广泛应用对高校非计算机专业人才培养提出的新要求，我们组织一批

长期从事计算机技术教学和科研的教师,编写了这套计算机基础教学系列教材。本系列教材有如下特点:

1. 适合于层次教学模式。系列教材内容覆盖了高校各类非计算机专业三层次计算机基础教学要求,既有适合理工类专业使用的,也有适合文经类专业使用的,各类专业都可从中选择到相应的教材。

2. 内容新。系列教材较好地反映了计算机技术的新发展,如《计算机文化基础》介绍了图形窗口界面和网络、多媒体基础;《计算机软件技术基础》介绍了软件基本概念和基本工具、结构化及面向对象程序设计的概念与方法、软件工程的基本思想和最先进的开发环境及平台;《计算机信息管理基础》以大型关系数据库管理系统为背景,介绍了关系数据库的基本知识和数据的构造方法以及网络技术在系统中的应用等等。

3. 强调应用和实用。非计算机专业的计算机基础教学以应用为目的,因此,本系列教材在编写上特别注意应用需要,强调实用性。主要课程教材都配有实验教程,基本知识理论讲深讲透,使用技术主要通过学生上机实验来掌握。

4. 便于自学。为了充分调动学生的学习主动性和能动性,本系列教材在写法上,既注意概念的严谨与清晰,又特别注意用易读易懂的方法阐述问题,应用举例丰富,便于自学。

总而言之,本系列教材的编写指导思想是:内容要新,要体现计算机技术的新发展和适应教学改革的要求;概念要清晰、通俗易懂,便于学生自学;应用性、实用性要强,切实在培养学生应用能力上下功夫;层次配套,可选择性强,适用面宽,既是普通高校非计算机专业本专科学生教材,亦可作为高等教育自学教材和工程技术人员的参考书。

限于编者水平,系列教材的内容及体系难免有缺点错误,诚恳希望读者和专家给予指正。

编 委 会

2002 年 1 月

前言

程序设计是计算机应用技术中最重要的基本技术之一，无论是计算机专业还是非计算机专业的人员，要利用计算机处理专业技术方面的实际问题，程序设计技术将是必不可少的。对在校学习的大学生来说，学习程序设计主要是学习程序设计的思想和方法，至于使用什么样的语言则是无关紧要的。计算机语言只是实现程序的一个工具，掌握了用一种语言设计程序的方法，学习用其他的语言来设计程序将是十分容易的事情。很难想象一个人在一生的工作中仅会使用在校学习的一种计算机语言来设计程序，处理自己所涉及的专业问题。

FORTRAN 语言是世界上最早出现的高级语言。由于具有强大的数值计算功能，它一直是从事科学计算和工程计算的人员所首选的程序设计语言。从 1954 年第 1 个版本的 FORTRAN 语言问世到现在虽然已近半个世纪，但它并未因此而过时，而是随着时间的推移而不断更新和发展。从另一个角度看，正因为它长久的历史积累了大量成熟可靠的语言代码，为后人提供了丰富而实用的程序资源。

FORTRAN 77 是 ANSI 在 1978 年公布的一个标准 FORTRAN 版本，它使 FORTRAN 语言从非结构化语言向结构化语言迈进了一大步，它是 20 世纪 80 年代使用

最广泛的一种高级语言。随着计算机技术和程序设计技术的不断发展，FORTRAN 语言也不断地改进和发展，比如，Microsoft 公司在 FORTRAN 77 的基础上相继推出了扩展的 MS-FORTRAN 5.0 和 MS-FORTRAN 5.1。ANSI 在总结各种改进和扩展版本后也于 1991 年公布了新的 FORTRAN 90 版本。同时 FORTRAN 也为适应计算机操作系统的发展，一改传统 DOS 界面，由 Microsoft 公司推出了基于 Windows 平台的 FORTRAN Power-Station，并继而由 DEC 公司在 FORTRAN PowerStation 4.0 的基础上推出了 Visual FORTRAN 5.0。

本书基于 FORTRAN 77 的基本功能和用法，以大量的实例详细介绍了程序设计思想和方法，并以 MS-FORTRAN 5.1 为蓝本，适当介绍了一些扩展的高级功能，使读者能够熟练地掌握 FORTRAN 语言基础和具有程序设计基本能力，并具备进一步学习 FORTRAN 90 和其他语言的能力。

本书共分 10 章。第 1 章介绍 FORTRAN 语言的基本情况、算法的概念、程序设计思想和软件工程概念；第 2 章介绍 FORTRAN 的基础知识和基本规定；第 3 章介绍顺序结构程序的设计方法；第 4 章介绍选择结构程序设计方法；第 5 章介绍循环结构程序的设计方法；第 6 章介绍 FORTRAN 的数组；第 7 章介绍程序设计的模块化方法；第 8 章介绍 FORTRAN 的文件操作；第 9 章介绍 MS-FORTRAN 扩展的高级功能及其使用方法；第 10 章介绍 MS-FORTRAN 的绘图功能。每章的后面都附有大量的习题和上机练习题，供读者温习所学内容和上机实践。此外，为方便读者，本书在附录中还提供了 MS-FORTRAN 5.1 和 FORTRAN PowerStation 4.0 的使用方法。同时，还为读者提供了 FORTRAN 77 和 FORTRAN 90 的全部标准系统函数。

本书是重庆市非计算机专业计算机基础教育系列教材之一。重庆大学谭世语和范幸义任主编。第 1，2 章和附录由重庆大学谭世语编写，第 3～5 章由重庆交通学院周建丽编写，第 6，7 章由西南农业大学郑昆编写，第 8～10 章由重庆大学范幸义编写。全书由重庆大学孙蓉教授审稿。

由于编者水平有限，书中不免存在错误和不足，望读者批评指正。

编　者

2002 年 2 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目录

1 概 述

1.1 FORTRAN 语言的发展概况 …… (1)
1.2 FORTRAN 语言的特点和功能 …… (3)
1.3 程序设计中的算法及其表示方法 …… (4)
1.3.1 算法的基本概念 …… (4)
1.3.2 算法的特性 …… (5)
1.3.3 算法的表示方法 …… (6)
1.4 程序设计方法与软件工程 …… (12)
1.4.1 程序的设计方法 …… (12)
1.4.2 软件工程的概念 …… (16)
习题 1 …… (20)

2 FORTRAN语言基础

2.1 FORTRAN 的基本规定 …… (21)
2.1.1 FORTRAN 语言使用的字符集 …… (21)
2.1.2 FORTRAN 的标识符 …… (22)
2.1.3 FORTRAN 源程序的书写规则 …… (23)
2.2 FORTRAN 的程序和语句 …… (25)
2.2.1 FORTRAN 程序的构成 …… (25)
2.2.2 FORTRAN 的语句 …… (28)

2.3 FORTRAN 的数据类型 …… (31)
2.3.1 整型数据 …… (32)
2.3.2 实型数据 …… (33)
2.3.3 双精度型数据 …… (34)
2.3.4 复型数据 …… (35)
2.3.5 字符型数据 …… (35)
2.3.6 逻辑型数据 …… (37)
2.3.7 数据类型的扩展 …… (38)
2.4 FORTRAN 的函数 …… (41)
2.4.1 系统函数 …… (41)
2.4.2 自定义函数 …… (43)
2.5 FORTRAN 的表达式 …… (45)
2.5.1 算术表达式 …… (45)
2.5.2 字符表达式 …… (47)
2.5.3 关系表达式 …… (48)
2.5.4 逻辑表达式 …… (50)
2.5.5 运算的优先级 …… (51)
习题 2 …… (52)

3 顺序结构设计

3.1 变量的赋值 …… (54)
3.1.1 赋值语句 …… (54)
3.1.2 DATA 语句 …… (56)
3.2 数据的输入和输出 …… (57)
3.2.1 简单的输入和输出语句 …… (58)
3.2.2 格式输出 …… (61)
3.2.3 格式输入 …… (70)
3.2.4 自带格式的输入和输出 …… (73)
3.3 其他基本语句 …… (74)
3.3.1 END 语句 …… (74)
3.3.2 PAUSE 语句 …… (74)
3.3.3 STOP 语句 …… (74)
3.3.4 PARAMETER 语句 …… (75)
3.3.5 语句函数的使用 …… (76)
3.4 顺序结构程序设计 …… (78)
习题 3 …… (82)

4 选择结构设计

4.1 逻辑 IF 语句 …… (88)
4.2 块 IF 结构 …… (89)
4.2.1 单分支的块 IF 结构 …… (89)
4.2.2 双分支的块 IF 结构 …… (91)
4.2.3 块 IF 的嵌套 …… (92)
4.2.4 多分支的块 IF …… (94)
4.3 选择结构程序设计 …… (97)
习题 4 …… (101)

5 循环结构设计

5.1 循环结构 …… (107)
5.1.1 循环结构的构成 …… (107)
5.1.2 循环结构的两种类型 …… (113)
5.2 用 GOTO 语句设计的循环程序 …… (114)
5.2.1 块 IF 与 GOTO 语句实现的循环 …… (114)
5.2.2 逻辑 IF 语句与 GOTO 语句设计的循环程序 …… (116)
5.3 用循环语句设计循环程序 …… (117)
5.3.1 DO-WHILE 循环 …… (117)
5.3.2 DO 循环结构 …… (121)
5.4 结构化的循环结构 …… (129)
5.5 循环嵌套 …… (130)
5.6 循环结构程序设计 …… (133)
5.6.1 循环结构语句的比较 …… (133)
5.6.2 循环结构程序设计综合举例 …… (134)
习题 5 …… (145)

6 数 组

6.1 数组与数组元素 …… (158)
6.1.1 数组的说明 …… (159)
6.1.2 数组元素及其引用 …… (160)
6.2 数组的逻辑结构和存储结构 …… (161)
6.3 给数组赋初值 …… (162)
6.4 数组的输入与输出 …… (163)
6.4.1 数组的整体输入与输出 …… (164)

6.4.2 用 DO 循环实现数组的输入与输出 …… (165)
6.5 应用举例 …… (168)
习题 6 …… (183)

7 程序设计的模块化

7.1 程序的模块化思想 …… (189)
7.2 函数子程序 …… (191)
7.2.1 函数子程序的设计 …… (191)
7.2.2 函数子程序的引用 …… (193)
7.3 子例行程序 …… (194)
7.3.1 子例行程序的设计 …… (194)
7.3.2 子例行程序的调用 …… (195)
7.4 程序间的数据传递 …… (196)
7.4.1 实参和虚参之间的数据传递 …… (196)
7.4.2 程序间的过程名传递 …… (201)
7.4.3 程序间的数据传递 …… (203)
7.5 数据块子程序 …… (208)
7.6 模块化程序举例 …… (210)
习题 7 …… (216)

8 FORTRAN的文件操作

8.1 FORTRAN 的文件 …… (221)
8.1.1 记录与文件 …… (221)
8.1.2 文件的分类 …… (222)
8.2 文件相关操作语句 …… (223)
8.2.1 OPEN 语句 …… (223)
8.2.2 READ/WRITE 语句 …… (225)
8.2.3 CLOSE 语句 …… (226)
8.2.4 REWIND 语句 …… (227)
8.2.5 BACKSPACE 语句 …… (228)
8.2.6 ENDFILE 语句 …… (228)
8.2.7 INQUIRE 语句 …… (229)
8.3 文件的使用 …… (231)
8.3.1 文本文件的使用 …… (232)
8.3.2 有格式顺序文件的使用 …… (234)
8.3.3 无格式顺序文件的使用 …… (235)
8.3.4 有格式直接文件的使用 …… (237)

此为试读,需要完整PDF请访问: www.ertongbook.com

8.3.5 无格式直接文件的使用 …… (239)
8.3.6 二进码顺序文件的使用 …… (240)
8.3.7 二进码直接文件的使用 …… (242)
习题 8 …… (244)

9 MS-FORTRAN的高级功能

9.1 元命令及其使用方法 …… (246)
9.1.1 元命令 …… (247)
9.1.2 元命令的使用 …… (251)
9.2 结构、联合与内存映像 …… (257)
9.2.1 结构 …… (258)
9.2.2 联合与映像 …… (259)
9.3 符号属性和外部过程界面说明 …… (263)
9.3.1 符号属性说明 …… (263)
9.3.2 外部过程界面说明 …… (265)
9.4 标准外部函数过程 …… (267)
9.4.1 时间/日期函数 …… (267)
9.4.2 随机子过程 …… (270)
9.4.3 命令行参数过程 …… (271)
9.4.4 DOS 命令及外部命令调用函数 …… (272)
9.5 字形处理 …… (274)
9.5.1 文本字符与字形 …… (274)
9.5.2 字形子程序 …… (277)
习题 9 …… (281)

10 MS-FORTRAN的绘图程序设计

10.1 FORTRAN 的图形模式 …… (283)
10.1.1 屏幕坐标系与色彩 …… (284)
10.1.2 图形模式设置和取消 …… (284)
10.2 绘图程序设计 …… (285)
10.2.1 图形基本操作命令 …… (285)
10.2.2 基本绘图命令 …… (287)
10.2.3 基本绘图程序设计 …… (288)
10.2.4 综合绘图程序设计 …… (294)
10.2.5 三维图形程序设计 …… (303)
10.3 屏幕窗口和图像操作 …… (310)
10.3.1 页面设置 …… (310)

10.3.2 视区和窗口 …… (312)
10.3.3 屏幕图形的保存与重绘 …… (315)
10.4 Auto CAD 接口程序设计 …… (319)
10.4.1 与 Auto CAD 的接口技术 …… (320)
10.4.2 Auto CAD DXF 文件接口图形子程序 …… (320)
10.4.3 Auto CAD 接口程序设计 …… (322)
10.4.4 大型软件包介绍 …… (327)
习题 10 …… (330)

附 录

附录 1 MS-FORTRAN 5.1 的使用 …… (332)
附录 2 FORTRAN PowerStation 4.0 的使用 …… (341)
附录 3 常用字符的 ASCII 代码表 …… (353)
附录 4 FORTRAN 的系统函数 …… (354)

参考文献 …… (362)

概　　述

FORTRAN 语言是世界上最早出现的高级程序设计语言，也是世界上广泛流行的、最适合于数值计算的一种计算机语言。随着计算机技术的高速发展，新的程序设计语言不断出现，功能不断加强和改进，FORTRAN 语言也逐步改进，提高其功能，以适应计算机发展的要求。

1.1　FORTRAN 语言的发展概况

FORTRAN 是 Formula Translation 的缩写，意思是“公式翻译”。它最初是专为解决科学、工程及管理中能够用数学公式表达的问题而设计的一种高级程序设计语言，因而具有极强的数值计算功能。

John Backus 于 1953 年在纽约首先提出 FORTRAN 语言的思想，称为 FORTRAN Ⅰ。第一个 FORTRAN 程序于 1957 年 4 月投入运行后，FORTRAN 的应用迅速传播，逐渐广泛，并逐步充实而形成许多不同的版本。早期较为重要的流行版本有 1958 年出现的 FORTRAN Ⅱ和 1962 年出现的 FORTRAN Ⅳ。但这两个版本并不兼容，给 FORTRAN 的推广造成困难，从而使其标准化工作刻不容缓。

1962 年 5 月，美国国家标准学会(American National Standard Institute，简称 ANSI) 成立了相关的机构来进行 FORTRAN 的标准化工作。1966 年 ANSI 正式公布了 FORTRAN 的两个美国标准文本:

美国国家标准 FORTRAN(ANSI X3.9—1966) ，相当于 FORTRAN Ⅳ。

美国国家标准基本 FORTRAN(ANSI X3.10—1966) ，相当于 FORTRAN Ⅱ。

其中美国国家标准基本 FORTRAN 是美国国家标准 FORTRAN 的一个子集，从

而实现了 FORTRAN 语言的向下兼容。

1972 年,国际标准化组织(International Standard Organization,简称 ISO)在美国国家标准 FORTRAN 的基础上稍加修改后,公布了 ISO FORTRAN 标准,即《程序设计语言 FORTRAN ISO 1539—1972》(ISO R1539),它分为三级:

基本级,相当于 ANSI X3.10—1966。

中间级,介于基本级和完全级之间。

完全级,相当于 ANSI X3.9—1966。

FORTRAN Ⅳ(又称为 FORTRAN 66)为国际上广泛接受,几乎统治了所有的数值计算领域,在其公布后的 10 多年中,大量的应用程序和程序库都用 FORTRAN 语言编写。

然而,随着计算机软硬件技术的迅猛发展,特别是结构化程序设计方法(Structure Programming,简称 SP)提出后,FORTRAN Ⅳ明显不能满足其要求。这是因为 FORTRAN Ⅳ不是结构化的语言,没有直接实现三种基本结构的语句,而程序中需要用 GOTO 语句来实现某些特定的算法功能。为解决这一问题,许多计算机厂商对 FORTRAN Ⅳ进行了不同程度的扩充。针对这种情况,美国国家标准化学会于 1976 年对 ANSI FORTRAN(X3.9—1966)进行了修订,基本上把各厂商扩充的有效功能都吸收了进去,并增加了许多新的内容。1978 年 4 月 ANSI 正式公布了新的美国国家标准 ANSI X3.9—1978《程序设计语言 FORTRAN》,这就是通常所称的 FORTRAN 77。

1980 年,ISO 正式采纳 FORTRAN 77,成为国际标准 ISO 1539—1980,该标准同样分为子集和全集。FORTRAN 77 具有一定的结构化功能,并能向下兼容,因此在公布之后很快得到了广泛的应用。中国于 1983 年 5 月公布的中国国家标准 FORTRAN (GB3057—82)基本采用了 ISO 1539—1980 FORTRAN,即 FORTRAN 77。

由于 FORTRAN 77 不是一个完全结构化的程序设计语言,使其在现代应用中处于不利地位。为适应现代程序设计思想,国际上对 FORTRAN 语言的结构化和现代化进行了不懈的努力,一些公司相继推出了对 FORTRAN 77 的改进版本,较为典型的如 Microsoft 公司推出的MS-FORTRAN 5.1。1991 年 5 月 ANSI 通过了新的 FORTRAN 标准 ANSI X3.198—1991,相应的国际标准化组织编号为 ISO/IEC1539—1991,即 FORTRAN 90。FORTRAN 77 与其他 FORTRAN 版本一样作为 FORTRAN 90 的子集,仍可很方便地在 FORTRAN 90 系统中使用。

FORTRAN 90 不仅仅是将已有的 FORTRAN 语言进行了新的标准化,更重要的是吸收了其他现代计算机语言的优点,发展了 FORTRAN 语言,使之成为既保留传统优点,又具有完全结构化特点的现代程序设计语言。

在对语言功能本身进行升级提高的同时,FORTRAN 为适应计算机操作系统的发展,一改传统 DOS 界面,由 Microsoft 公司推出了基于 Windows 平台的 FORTRAN PowerStation,并继而由 DEC 公司在 FORTRAN PowerStation 4.0 的基础上推出了 Visual FORTRAN 5.0。使 FORTRAN 不仅能够开发基于 DOS 的应用程序,也能开发基于 Windows 的应用程序,并能够很容易地与其他语言开发的程序进行很好的连接,大大提高了 FORTRAN 的适用领域。

考虑到 FORTRAN 语言的基本特点及使用状况,本书仍以 FORTRAN 77 为基础

介绍 FORTRAN 语言的基本功能和用法,同时以 MS-FORTRAN 5.1 为蓝本,适当介绍一些扩展的高级功能,使读者能够熟练地掌握 FORTRAN 语言基础和具有程序设计基本能力,并具备进一步学习 FORTRAN 90 和其他语言的能力。

1.2 FORTRAN 语言的特点和功能

FORTRAN 语言的原始功能主要是数值运算,其文字和图形处理功能很弱。经过几代的发展,其功能早已突破这一界线,现代的 FORTRAN 已经具备很强的图形功能和文字处理功能,完全能与目前流行的其他语言媲美。

与同样常用于科学运算的 C/C ++ 语言和 Matlab 语言相比,FORTRAN 更具有以下独到的特点:

(1) FORTRAN 语言发展历史长,积累的财富多　FORTRAN 语言是最早使用的高级语言,长期以来一直是科技工作者和工程设计人员的首选语言。因而,在众多的领域中积累了大量实用程序,形成了一笔可观的财富。尽管 FORTRAN 语言进行了多次升级改版,但高级版本的向下兼容性保证了这一积累的继续可用性。这是其他语言所无法比拟的。

(2) FORTRAN 语言的书写和语法规则要求严格　FORTRAN 语言严格的书写和语法规则,特别适合严谨的科学计算和工程计算。与之对应,C 语言就没有这么严谨,比如,C 语言不提供数组的越界检查,虽然这使得数组的使用更加灵活,但如果允许访问到错误的内存地址,其计算结果将是不可预测的,这在科学和工程计算中将可能导致不可估量的损失。

(3) FORTRAN 语言可直接对数组和复数进行运算　FORTRAN 语言提供了对数组和复数的直接运算功能,大大简化了运算过程。而针对矩阵的运算和复数运算是科学计算和工程计算中使用很频繁的两种运算,采用其他语言处理这类运算,其程序设计则要复杂得多。

(4) FORTRAN 语言具有不可替代的并行计算功能　巨型计算机的主要功能,都是依靠并行处理来实现的。近年来 FORTRAN 的升级和扩展中包括了很多对数组的高水平操作,这些操作对并行优化十分有利。此外,还有对共用存储器并行系统的扩展(PCF)和对串并行系统的数据并行扩展(HPF),这些功能都使得 FORTRAN 在并行计算领域独领风骚。

(5) FORTRAN 语言是一种编译语言　编译语言生成的运行程序具有很高的运行速度。Matlab 尽管在矩阵运算上十分方便,但它是一种解释语言,运行速度慢,程序效率低,在大规模程序设计上,它无法与 FORTRAN 相比。

尽管 FORTRAN 语言书写呆板,结构化程度不如 C 语言,但它严谨的语法规则和丰富的底蕴,以及几经升级扩展的现代改进,使它仍然是科学与工程计算的首选语言。

1.3 程序设计中的算法及其表示方法

根据著名计算机科学家 Nikiklaus Wirth 的观点，一个程序应包括对数据的描述和对操作的描述两方面的内容。对数据的描述，就是在程序中指定数据的类型和数据的组织形式，即数据结构；对操作的描述，就是规定计算机的操作步骤，即算法。因此，可以将程序理解为：

程序 = 数据结构 + 算法

实际上，在一个具体的程序中，除以上两个要素外，还需要有程序的表现形式和对程序的描述语言，即程序的设计方法和计算机语言。所以，可以将程序进一步理解为：

程序 = 数据结构 + 算法 + 程序设计方法 + 语言工具和环境

在这 4 个方面中，数据是程序加工的对象，语言是工具，程序设计方法是手段，而算法则是实现程序功能的灵魂。因此，良好的算法设计，是程序实现的关键所在。

1.3.1 算法的基本概念

在数值运算中，要解决一个计算问题可以有多个方法和步骤来实现。例如，求自然数1 ～100的和，用以下 3 种方法中的任何一种都能实现：

(1) $\sum_{n=1}^{100} n = 1 + 2 + 3 + \cdots + 99 + 100 = 5\ 050$

(2) $\sum_{n=1}^{100} n = 100 + (1 + 99) + (2 + 98) + \cdots + (49 + 51) + 50 = 50 \times 100 + 50 = 5\ 050$

(3) $(1 + 2 + \cdots + 10) = 55$

$11 + 12 + \cdots + 20 = 10 \times 10 + (1 + 2 + \cdots + 10) = 100 + 55$

……

$91 + 92 + \cdots + 100 = 10 \times 90 + (1 + 2 + \cdots + 10) = 900 + 55$

$100 + 200 + \cdots + 900 = 100 \times (1 + 2 + \cdots + 9) = 100 \times 45$

$\sum_{n=1}^{100} n = 10 \times 55 + 100 \times 45 = 5\ 050$

这 3 种方法思路不同，步骤也不同，但解决的都是同一个问题，同样都能达到目的。虽然都能达到同样的目的，但方法有优劣之分，有的方法只需要很少的步骤就能得到结果，而有的方法则需要经过很多的步骤才能得到结果。

在程序设计中，仅有数学上的解题方法是远远不够的，必须将数学上的解题方法设计成计算机中的算法形式，即设计出计算机解决问题的步骤后，才能依此编写出相应的程序让计算机执行。例如，对求自然数 1 ～100 的和，可以将上述第(1)种方法

此为试读，需要完整PDF请访问：www.ertongbook.com