

FORTRAN 语言

与

计算机数学

(供工程师和科学工作者用)

[美] R.E.卡里勒 B.E.基勒特

石油工业出版社

F O R T R A N 语 言 与 计 算 机 数 学

(供工程师和科学工作者用)

R.E.卡里勒 B.E.基勒特

蔡陞健 译

石 油 工 业 出 版 社

内 容 提 要

本书系统地介绍了FORTRAN语言和常用的计算机数学问题。全书分两大部分。第一部分详细介绍了目前国际上常用的计算机FORTRAN语言，第二部分介绍了科技方面经常遇到的应用数学问题及其数值解法。其特点是把FORTRAN语言程序与计算机数学结合在一起，并给出了常用的程序实例。它着重于实用，读者可学会用FORTRAN对一些数学问题编制程序，进行上机运算。本书可供从事数据处理工作的技术人员和高等院校有关专业师生参考。

全书由王宏琳同志校订。

R.E.CARLILE B.E.GILLETT
FORTRAN and Computer Mathematics
for the Engineer and Scientist
The Petroleum Publishing Company Tulsa, 1973.

FORTRAN语言与计算机数学

(供工程师和科学工作者用)

R.E.卡里勒 B.E.基勒特

蔡陆健 译

石油工业出版社出版
(北京安定门外外馆东后街甲36号)
人民交通出版社印刷厂排版
京辉印刷厂印刷
新华书店北京发行所发行

850×1168毫米 32开本 17¹/₂。印张449千字 印1—11,500

1984年3月北京第1版 1984年3月北京第1次印刷

书号: 15037·2079 定价: 2.10元

前 言

要在技术工作中学习如何使用现代计算机的科技方面的大学生，面临着三个基本问题。这些问题涉及：所要解决问题的类型，[†]在计算机上解决问题所需要的技巧，以及（由计算机灵活性所涉及的数学和所使用的语言决定的）限制。

依我们的观点，这三个问题是“它（即问题和计算机）是什么？”“我如何使用它（即计算机）？”和“它有哪些限制（即FORTRAN语言，数字计算机和在计算机上应用数学的限制）？”。

在这本分为两部分的书里，我们努力向技术方面的大学生一步一步地介绍计算机语言，随之介绍应用技术和他所需的数学，以提供给他一些解决这三个问题的主要方法。

这本书的第一部分是对FORTRAN语言的介绍，它适用于所有的技术方面的学生和所有当代的计算机。这部分共十一章，专门介绍这个语言的基础和对已经懂得基本知识的学生来说非常有助于记忆的英语关系。介绍程序员可以使用的每一条命令时，同时要介绍此命令或语句所执行的动作和语句的一般形式，并用一些实例来说明它的用途。在介绍这些内容时总是力求适用于回答这三个基本问题，即“它是什么？”，“我如何使用它？”和“它有哪些限制？”。

学习完了前两章，读者已能胜任地进行计算机上的基本程序设计，后面的几章是用辅助语句和灵活性来提高读者的技巧和能力。

从第十二章开始，所有教材都是些可在数字计算机上编制程序的数值计算技术，每项计算技术后面都有说明其使用方法的应用例题。在大部分情况下都给出了计算机程序，读者只需将程序

穿成卡片，提供正确的数据，并将此完整的程序在计算机上来解决他的问题。

在这部著作中，讲到了好几种技术，即：线性规划，最小二乘法分析，等待线分析等。总之，我们试图介绍多种技术的基础，并有意将重点放在“如何解决问题”上，而不在于对这些技术进行详细的理论分析。好在，在用循序渐进的方法，足够详细地介绍每一项技术的过程中，已使读者能够获得很好使用这项技术的知识，而避免陷入繁杂的理论分析之中。当出现可以使用几种技术解决同一问题时，书中给出了对于这些技术的分析方法，可以决定在什么情况下使用那种技术。

这本书的后一部分，完全可以做为一个学期的教材，用来重点讲授在计算机上解题的数值方法。特别是，当要教给学生们研究或在数字计算机上实践“如何”求出他们有关工程和科学问题的数值解时，这是一本很适用的教科书。当然要把有关如何应用的训练和在本书中介绍的数值技术这两个重点分开。

在所有这类著作中，都存在“如何进行探讨”这样一个基本问题。而目前在如何介绍这类教材的问题上，存在着两种基本不同的探讨方法。第一种探讨方法是那些认为三种内容必须同时介绍的作者们所采用的。第一项内容是：FORTRAN 语言和它的每一条命令功能的介绍；第二：同时要介绍由于应用每一条命令所引起计算机内部发生的情况；第三：完整地介绍在工程和科学方面的应用，以阐明每一命令语句不同的灵活性和它的变化。

当不需要减少这三个方面中的任一方面仍可以有效地进行讲授的情况下，我们并不完全不同意这种“包罗万象”的探讨方法。然而根据我们在进修教育学校中对四百多个工程师和在大学中对几百个工程和科学方面大学生，经过四年和在81个专业讲课的经验，我们已经确信最好的探讨方法就是第二种，即不同时介绍两种以上的功能，并在很多情况下只介绍一种。前面几章首先集中介绍 FORTRAN 语言，而后面一部分则集中介绍数值技术和技术学生所需的数学。只有极少一部分才牵扯到机器内部运行

和数字运算情况，除非这些运算是严重影响到程序设计和运算结果。

在每一章的末尾都有一些基本例题，用它来检查对前面教材的理解程度并加深对它们的理解。再者，为了和我们前述观点一致起见，我们尽量不把教材弄混，或者说不把工程问题和 FORTRAN 语言混在一起。每章末尾的例题要么都是 FORTRAN 语言，要么就都是数值技术。只有在这些情况下，即我们认为详细的工程或科学问题，可以大大有助于说明我们的意图时我们才给出一个复杂的例题。我们确信使用这本教科书的技术人员和大学里的学生们会同意我们所采用的探讨方法和它的效果的。

对 FORTRAN 和对计算机数学这两部分的探讨，是为工程师和科学家准备的。它所包含的教材，一般在高等院校中，要分为两门课程来讲授。此外，对那些希望通过自学 FORTRAN 和数值技术，来求得问题的数字解法的工程师或科学家们，我们希望，这本书能很好的满足他们的需要。

R.E. 卡里勒 B.E. 基勒特
罗拉，密苏里 1973

目 录

前 言

第一章 数字计算机	1
1.1 数字计算机是什么?	1
1.2 计算机做些什么?	9
1.3 技术人员的计算机语言——FORTRAN N.....	11
1.4 如何建立程序.....	18
1.5 流程图.....	23
第二章 数和计算机	31
2.1 FORTRAN 语言	31
2.2 算术语句.....	31
2.3 算术运算分级.....	33
2.4 括号	35
2.5 变量和常数的命名.....	38
2.6 整数和实数.....	42
2.7 单精度和双精度数.....	46
2.8 科学记数法——指数 E	47
2.9 二进制和十六进制.....	50
第三章 库函数	62
3.1 FORTRAN 函数	62
3.2 库函数.....	62
3.3 算术语句函数.....	70
第四章 输入/输出语句	77
4.1 输入/输出 (INPUT/OUTPUT) 语句.....	77
4.2 自由 INPUT/OUTPUT	77
4.3 INPUT/OUTPUT——格式控制.....	82
4.4 FØRMAT字段说明.....	98

4.5	整数字段说明——Iw.....	99
4.6	浮点字段——不带指数——Fw.d.....	102
4.7	浮点值——带指数——Ew.d.....	104
4.8	其它格式字段说明.....	106
4.9	X格式说明.....	107
4.10	霍勒斯 (Hollerith) 字段说明——wH.....	108
4.11	撇号.....	111
4.12	走纸控制符.....	112
4.13	比例因子 P.....	115
4.14	重复格式说明.....	117
第五章	数组和矩阵的下标.....	122
5.1	数组.....	122
5.2	下标变量的表示法.....	124
5.3	下标存储——维数.....	128
5.4	实型和整型语句的内存分配.....	130
5.5	数组的 INPUT/OUTPUT.....	131
第六章	循环——IF 和 DO 语句 (条件语句和循环语句).....	133
6.1	算术 IF 语句.....	133
6.2	逻辑 IF 语句.....	136
6.3	DO 语句.....	139
6.4	关于 DO 循环的一些限制.....	149
6.5	CONTINUE 语句.....	150
第七章	数组的隐 DO 循环.....	153
7.1	数组 I/O 的标准 DO 循环.....	153
7.2	隐 DO 循环——一维.....	154
7.3	隐 DO 循环——多维.....	155
7.4	摘要.....	158
7.5	说明 DO 循环的例题.....	159
第八章	附加命令.....	166
8.1	计算 GO TO 语句.....	166

8.2	ASSIGN语句(语句标号赋值语句)	167
8.3	赋值GOTO语句	167
8.4	DATA语句	168
8.5	EQUIVALENCE语句(等价语句)	170
8.6	COMPLEX语句(复数语句)	172
8.7	附加逻辑表示式	173
8.8	逻辑变量	176
8.9	DOUBLEPRECISION语句(双精度语句)	179
8.10	IMPLICIT语句(隐语句)	182
第九章	子例程序和函数子程序	188
9.1	函数子程序	188
9.2	子例程序	194
9.3	COMMON语句(公共语句)	197
9.4	数据块	200
9.5	EXTERNAL语句(外部语句)	201
9.6	ENTRY语句(入口语句)	202
第十章	附加输入/输出命令	208
10.1	Tw说明	208
10.2	A——说明	211
10.3	G——说明	215
10.4	E——说明	216
第十一章	附加语句命令和程序设计的改进	220
11.1	磁带命令	220
11.2	磁盘(直接存取)控制	221
11.3	PUNCH语句(穿孔语句)	223
11.4	相加和其它计算机操作技巧	224
11.5	循环	227
11.6	编码中应注意的一些问题	232
11.7	节省机时的求平方根法	238
第十二章	矩阵	242

12.1	矩阵理论基础	242
12.2	什么是矩阵	242
12.3	减	244
12.4	乘	245
12.5	矩阵的转置	248
12.6	单位矩阵	249
12.7	向量	250
12.8	行列式	250
12.9	矩阵的秩	252
12.10	余子式和伴随方阵	254
12.11	逆方阵	256
第十三章	线性方程组	261
13.1	线性方程组	261
13.2	线性方程组的矩阵表示法	262
13.3	克拉姆 (CRAMER) 法则	268
13.4	解线性方程组的方法	272
13.5	高斯-约当 (Gauss-Jordan) 消去法	272
13.6	推广的高斯-约当法	278
13.7	解线性方程组的高斯消去法	284
13.8	乔理斯基 (CHOLSKY) 法解线性方程组	292
13.9	解病态线性方程组	298
13.10	解线性方程组的高斯-赛德尔 (GAUSS-SEIDEL 迭代法	306
第十四章	最小二乘法的原理和应用	312
14.1	引言	312
14.2	最小平方原理	313
14.3	最小二乘法的计算机程序	318
14.4	Z 因子的最小平方解	323
第十五章	微分方程的解	329
15.1	简介	329

15.2	解微分方程的皮卡 (Picard) 方法	329
15.3	微分方程的台劳级数解	332
15.4	解微分方程的欧拉方法	334
15.5	解微分方程的迭代欧拉方法	338
✓ 15.6	龙格-库塔 (Runge-Kutta) 方法	340
15.7	微分方程的亚当斯(ADAMS)预测-校正解法	343
15.8	常微分方程的汉明解法	345
15.9	解高阶微分方程的汉明数值方法	351
15.10	汉明方法的应用	358
15.11	亚当斯-摩尔顿(Adams-Mouton)和亚当斯- 巴士费德(Adams-Bashford)方法	362
第十六章	三对角线和五对角线线性方程组的直接解	369
16.1	三对角线线性方程组的直接解	369
16.2	五对角线方程组的直接解	375
第十七章	具有边界条件微分方程的解法	377
17.1	边值问题的有限差分解法	377
17.2	四阶微分方程的有限差分解	384
17.3	打靶法近似	386
17.4	具有边界条件的非线性微分方程	390
第十八章	偏微分方程	394
18.1	偏微分方程的数值解	394
18.2	抛物型偏微分方程	400
第十九章	相关随机数据的基本技术	406
19.1	简介	406
19.2	选点法 (MSP)	407
19.3	平均法 (MOA)	407
19.4	最小二乘法 (MLS)	408
19.5	随机数据的线性拟合	409
19.6	半对数数据相关	416
19.7	具有曲率的半对数数据相关	419

19.8	对数数据相关	421
19.9	log-log 数据相关	424
19.10	抛物线型数据的相关	427
19.11	双曲线型数据的相关	430
19.12	“S”型数据相关	432
参考文献		436
第二十章 数值积分		437
20.1	简介	437
20.2	梯形法则	437
20.3	尤贝积分	443
20.4	数值积分的辛卜生 1/3 法则	448
20.5	高斯求积公式	454
20.6	勒让德-高斯求积公式	458
20.7	拉盖尔-高斯(Laguerre-Gauss)求积公式	462
20.8	埃尔米特-高斯求积公式	466
第二十一章 运筹学技术		470
21.1	简介	470
21.2	线性规划	470
21.3	线性规划的进一步探讨	479
21.4	线性规划解汽油混成问题	485
21.5	排队论(等待线)	491
21.6	排队论:多服务台情况	499
21.7	泊松到达分布和厄兰服务时间	503
21.8	分配规划	506
21.9	N项工作,2台机器的序列问题	511
21.10	N项工作,3台机器的序列问题	514
21.11	更换模型	514
21.12	随时间磨损的资产设备	514
21.13	彻底损坏的物品	518
21.14	动态规划	520

第一章 数字计算机

1.1 数字计算机是什么？

高速数字计算机是被设计用来进行加，减，乘和除等一般算术运算的。这和台式计算器，算盘和计算尺所进行的运算是完全相同的。

计算机执行这些运算在方式上有点像手摇计算机，在运算顺序方面则是模仿工程师在书桌上每天进行的计算。

使计算机成为无与伦比的原因，是因为它具有三个主要功能：（1）计算机在运算时的高速；（2）存储先前计算或输入的数据，供以后再使用的能力；（3）执行逻辑运算的能力。所有计算机执行的算术和非算术运算，都是由加，减，乘和除等基本运算组成，它们是在极高速度下完成的。

今天，计算机已被用在工业，科研和教育的各个方面。由于这些不同而又唯一的用途，就出现了一些不同而又唯一的语言，和日益增多的附属语句，修饰语句和输入/输出方面的新语句等等。我们将集中介绍在工程和科学方面使用的计算机，特别是工程师和科学家所使用的语言 FORTRAN。

数字和模拟的比较 计算机分为数字计算机和模拟计算机。数字计算机和模拟计算机的区别在于数字计算机在显示、运算、或存储数据和信息时使用的方法是“有限”和“离散”的方法，和模拟计算机的“连续”、“等价”、或平滑变化的方式完全不同，数字计算机的每一步运算都是用有限的和离散的单元来表示。

图 1.1 表示数值的连续表示法和数字表示法概念的比较。图的上部为一时钟，它的秒针，分针和时钟做正常的圆周扫描。这种扫描就是连续的，它的测量量——时间是在它连续运动的过程

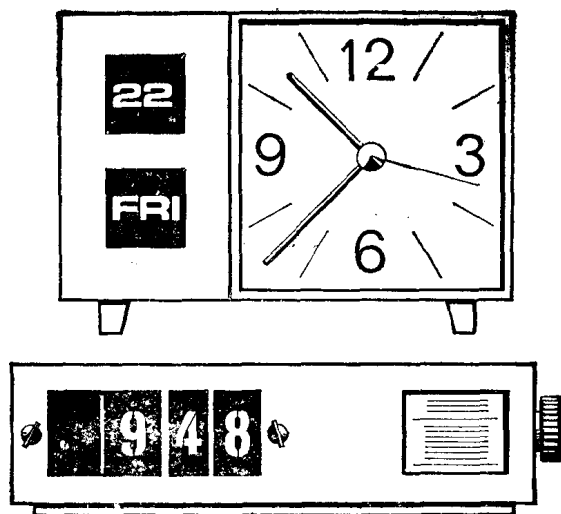


图1.1 数值的数字表示法和连续表示法的比较

中读出的。这就类似于模拟计算机。

在图的下部，为一时钟收音机，在它的左面为四个转动鼓，在鼓上可以显示出数字。这四个鼓从右至左，后两位为分的读数，从左边两位读出从1到12的小时读数。这种设备就和数字计算机一样是用离散和非连续的方式来显示数值。两种都是时钟，但后者是“冻结”了每一步的值或总合，而用离散数将它们显示出来。

计算机部件

数字计算机和台式计算器相似，但有三个主要差别：

- 计算机具有大的存储容量
- 它要快的多
- 它可以做逻辑（已编程序）判别。

计算机的主要部分示于图 1.2 中，这里的五个基本部件是互相分开而又按一定的方式连接在一起，这样每一个部件可以用很高的速度执行它的任务，然后把它的结果通过连线送到下一部件。

，这就使全部运算只用很少的时间。

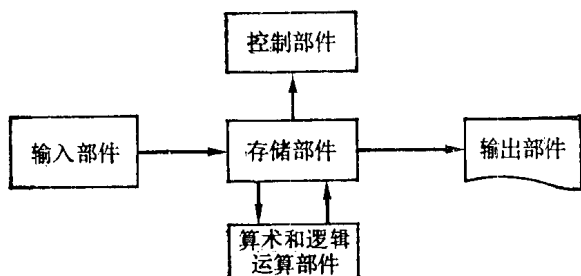


图1.2 数字计算机的主要部分

五个部件共同执行现代数字计算机的基本运算，每一部件有其单独的功能。

输入部件 输入部件的功能是从程序员那里接受数据和指令语句，然后在部件内将这些字母数字（字母和数）转换成电脉冲。这些脉冲再被传送到存储部件存储起来或做其它用途。

计算机程序员将要使用几种输入设备，最广泛使用的输入设备是卡读机，见图 1.3。程序员将计算机卡片送入卡读机，在卡片上已经穿好专门的命令或所期望的动作。卡读机扫描此卡片并启动对应于每一穿孔位置的电定位装置。我们可以认为这些被启动的位置处于“开”的位置，所有其它未被启动的位置是处于电器上“关”的位置。

这个卡片语句的开关电映象立即被传送到存储单元被存储起来以备以后使用。

与此类似，但使用的没有那么广泛的另一种输入设备是终端打字机，见图 1.4。用这种设备可以把命令句打入系统，每次打一个字母，这个命令句的电映象和卡读机所产生的电映象一样被存入存储器中。

还有一种输入设备就是高速磁带机，见图 1.5。这是一种和录一放磁带机相似的一种磁带机，它像卡读机和终端打字机一样把

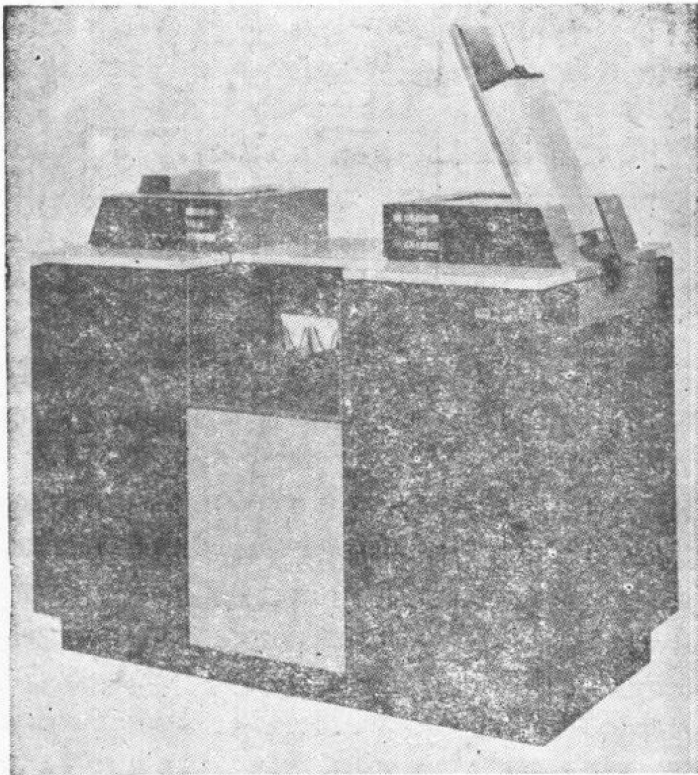


图1.3 输入卡读机

开关脉冲传送到存储部件中。自然，在输入以前要把命令预先存储在磁带上。由于带上的输入信息异常紧凑，因此它每秒可以向系统传送100000个以上的字母数字。与它比较一下，卡读机一般每分钟只能读入62000~65000个字符。

再比较一下，每个卡片最多能包含80个字符（包括字母和数字）。如每分钟要输入65000个字母数字，卡读机每分钟就要处理812张卡片，每张卡片大约为7×3英寸。同样的一分钟输入时间磁带机可以处理6000英寸磁带，每英寸包含1000个字母数字或者说总共为6兆个字母数字。

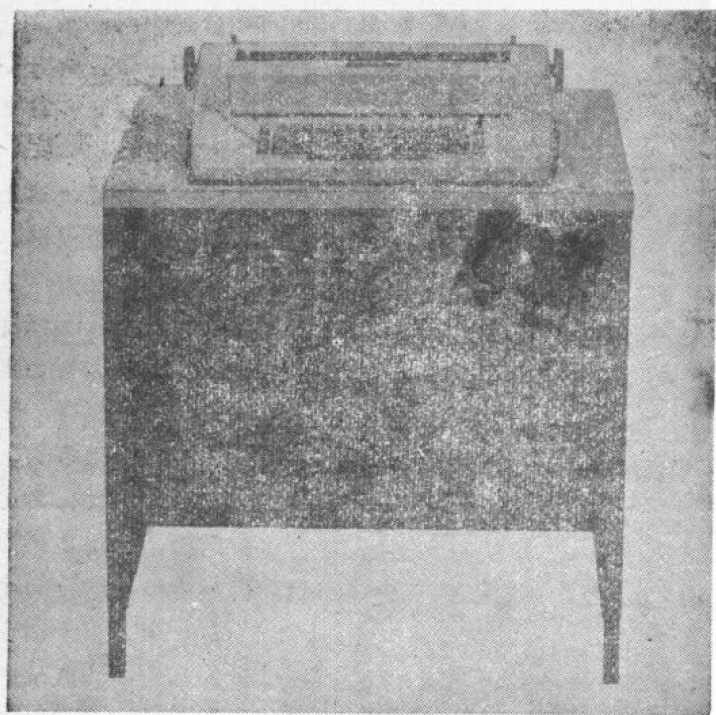


图1.4 输入打字机

存储部件 此部件的功能就是绝对准确地保存所有可能的字母数字和程序符号中的数码，数据和指令语句等。它是由成千个单独的磁心“位”所组成。最常见的存储器有磁盘存储器，磁鼓存储器，磁心存储器和磁带存储器。

存储部件接受两种基本类型的输入，第一，逐张卡片，逐个字符的接受源程序，计算机将根据这些卡片和字符连续不断的运行。存储部件将根据源程序的输入程序卡所给定的顺序，每次一张或每次一行的运行。

第二，它接受两种类型的数据。第一种数据是做为输入源程序的一部分，按前面的程序句指定的形式存入存储器。第二种，它也接受和存储在已存入程序的算术运算过程中所产生的数据。