

强清 浩淑 田
编著



FORTRAN

語言

清华大学出版社

FORTRAN 语言

清华大学出版社

FORTRAN 语言

譚浩強 田淑清 編著

清华大学出版社

1981

内 容 简 介

FORTRAN 语言，是国内外广泛使用的适于科技计算的计算机算法语言。本书是作者在清华大学校外多次讲授稿的基础上编写的。共分十二章，除介绍标准 FORTRAN 语言的基本规则和程序设计的基本技巧外，对某些广泛使用的非标准规定也作了简要介绍。本书是清华大学在教师、学生、研究生中普及该语言的教材，采取由具体问题入手，循序渐进，由浅入深的叙述方法，避免枯燥烦琐的缺点。本书是作者《BASIC 语言》一书的姊妹篇，例题较多，每章后附有习题（并另编有习题和解答）。本书通俗易懂，是一本入门书，适合初学者使用。可作大专院校和计算机学习班教材，也可供从事计算机工作的工程技术人员及自学读者阅读。

FORTRAN 语言

谭浩强 田淑清 编著

清华大学出版社出版

北京 海淀 清华园

北京印刷一厂印刷

新华书店北京发行所发行·各地新华书店经售

开本：787×1092 $\frac{1}{16}$ 印张 15 字数 369 (千字)

1981 年 7 月第一版 1981 年 7 月第一次印刷

印数 1~100000

统一书号：15235·10 定价 1.55 元

前 言

FORTRAN 是目前国际上广泛流行的一种适于科学计算的高级语言。它具有标准化程度高,便于程序交换,较易优化,执行效率高等显著优点。随着我国四个现代化的逐步实现和国内外文化科学交流的逐步扩大, FORTRAN 将会在我国占愈来愈重要的地位。

本书是为初学 FORTRAN 的同志编写的。由于 FORTRAN 语言各种严格的规定很多,初学者往往感到枯燥烦琐。因此本书在写法上,不采用手册的方法,不准备在一开始就罗列一大堆规定而令人望而生畏。而是根据初学者的特点从具体例子入手,引出概念。由浅入深,循序渐进。先介绍最简单的几种语句,读者学完前二章后即可自己动手编写一些简单的而且是完整的 FORTRAN 程序,有条件的还可边学边上机调试程序,在此基础上逐步深入,这样就不会感到枯燥无味。FORTRAN 中最令人头痛的是输入输出的格式,我们把它分成两部分介绍,在第二章中介绍最基本的最常用的一些格式,并在以后各章的学习中逐步熟悉了以后在第十一章中再作深入的介绍。

本书是一本 FORTRAN 的入门书。考虑到 FORTRAN 语言本身比较烦琐,因而不打算再引人太多的有关算法方面的问题,以免分散注意力。而把重点放在常用的基本语句的使用和程序编制的技巧上。对每一个基本语句,都通过举例反复说明和比较,以求读者对它有一个准确而又具体的概念。

考虑到目前学习 FORTRAN 的同志大多数是接触过计算机的,不少同志是学习过其它语言的。因此我们不再对计算机的基本知识等作过多的介绍。但同时兼顾那些从未接触过计算机和未学过其它语言的同志的特点,在写法上力求通俗易懂,使这部分同志也不致感到困惑。

本书介绍了一些常用的例题,并在每章后都附有习题(习题和解答另编成一本“FORTRAN 语言习题和解答”),在例题和习题中介绍了二百多个程序例子,可供读者参考。

本书基本上是按标准 FORTRAN (FORTRAN X 3.9~1966)介绍的。由于各种不同的计算机使用的 FORTRAN 有一些不同,我们在本书中对那些广泛使用的非标准的规定和语句也作简单的介绍。

本书各章中的例题只是从教学的角度为了说明某些语句的概念和使用而编写的例子,并不一定是该问题的最佳解法或最完善的程序。在本书中所介绍的程序基本上适用于任何一种配有 FORTRAN 编译系统的计算机系统。

本书是作者在近年来在教师、研究生和学生中讲授“FORTRAN 语言程序设计”课程的基础上编写的。本书初稿作为清华大学教材在校内印刷使用后,受到校内外同志的关心和鼓励。经过一段时间的试用后,我们现在又对它作进一步的修改和补充,正式出版。在此我们谨对曾对本书给予过关心和帮助的同志表示感谢。

本书由谭浩强、田淑清同志编写。其中谭浩强编写第一、二、三、四、五、十二章。田淑清编写第六、七、八、九、十、十一章。由于我们水平所限,错误在所难免,敬请读者批评指正。

编 者

1981年1月于
清华大学计算中心

目 录

第一章 关于 FORTRAN 的初步知识	1
§ 1. 从棍石记数到电子计算机	1
§ 2. 从机器语言到高级语言	4
§ 3. FORTRAN 语言的发展概况	5
§ 4. 最简单的 FORTRAN 程序分析	6
§ 5. FORTRAN 源程序的书写格式	9
§ 6. 常数	13
§ 7. 变量	16
§ 8. 算术运算符和算术表达式	18
§ 9. 标准函数	21
习题	22
第二章 最基本的 FORTRAN 语句	24
§ 1. 算术赋值语句	24
§ 2. 参数说明语句	26
§ 3. 输入和输出的概念	26
§ 4. 最简单的带格式的输出	28
§ 5. 最简单的带格式的输出	33
§ 6. 停(STOP)语句和暂停(PAUSE)语句	36
§ 7. 举例	38
习题	40
第三章 控制转移语句	41
§ 1. 无条件转移(GO TO)语句	41
§ 2. 算术条件转移(算术 IF)语句	41
§ 3. 简单的逻辑条件转移(逻辑 IF)语句	49
§ 4. 计算转移语句(计算 GO TO 语句)	55
§ 5. 赋标号(ASSIGN)语句和赋标号转移(赋标号 GO TO)语句	61
§ 6. 几种控制转移语句的小结	66
习题	68
第四章 双精度型、复型和逻辑型	70
§ 1. 双精度型常数和双精度变量	70
§ 2. 复型常数和复型变量	73
§ 3. 双精度复常数和双精度复变量	77
§ 4. 几种数值型量的小结	78
§ 5. 逻辑型常数和逻辑型变量	82
习题	90
第五章 循环	93
§ 1. 引言	93
§ 2. 循环(DO)语句	94
§ 3. 对循环的一些规定	100
§ 4. 继续(CONTINUE)语句	104

§ 5. 多重循环.....	106
§ 6. 应用举例.....	111
习题	116
第六章 数组	117
§ 1. 数组和数组元素的概念.....	117
§ 2. 数组说明语句(DIMENSION 语句)	119
§ 3. 数组元素的排列顺序和数组的输入输出.....	121
§ 4. 程序举例.....	126
习题	132
第七章 语句函数	133
§ 1. 语句函数的定义.....	133
§ 2. 语句函数的引用和使用举例.....	134
§ 3. 程序举例.....	137
习题	144
第八章 子程序	146
§ 1. 函数子程序.....	146
§ 2. 子例程子程序.....	159
§ 3. 可调数组.....	162
§ 4. 子程序的多重入口.....	168
§ 5. 可变返回点.....	170
§ 6. 外部语句(EXTERNAL 语句).....	172
§ 7. 过程的小结.....	176
习题	178
第九章 数据联系语句与数据置初值	180
§ 1. 等价语句(EQUIVALENCE 语句)	180
§ 2. 公用语句(COMMON 语句)	182
§ 3. 给变量赋初值(DATA 语句和数据块子程序).....	187
习题.....	190
第十章 文字型常数和格式数组	192
§ 1. 文字型常数.....	192
§ 2. 给变量赋文字型常数.....	192
§ 3. 字符的输出与输入.....	193
§ 4. 格式数组.....	196
习题	200
第十一章 输入输出的补充	201
§ 1. 纵向控制符.....	201
§ 2. FORMAT 语句中的斜杠符	202
§ 3. G 型格式说明符.....	203
§ 4. 比例因子.....	203
§ 5. H 型格式说明符在输入语句中的使用.....	204
§ 6. FORMAT 语句中的格式组	205
§ 7. 记录.....	206
§ 8. 固定格式输出、自由格式输入语句和无格式转换输入输出语句.....	207

§ 9. 文件和辅助输入输出语句.....	208
第十二章 FORTRAN 语言小结.....	210
§ 1. FORTRAN 语句小结.....	210
§ 2. FORTRAN 程序结构.....	212
§ 3. FORTRAN 语句的排列顺序.....	213
§ 4. FORTRAN 库函数.....	214
§ 5. 提高程序质量的方法.....	216
§ 6. FORTRAN 程序的运行.....	221
附录.....	226
附录 I FORTRAN 字符集.....	226
附录 II 字符-ASCII 代码-五单位码对照表.....	227
附录 III 中英名词对照表.....	229

第一章 关于 FORTRAN 的初步知识

§ 1. 从棍石记数到电子计算机

古代人们最早是用木棍和石子来记数的，随着生产的发展，逐步发展了各种计算工具（例如算盘、计算尺、机械式计算机），最后发展为电子计算机。自 1946 年电子计算机问世至今不过三十多年，但它的迅猛发展，开辟了科学技术的新的阶段。现在计算机已经开始进入人们生产和生活中的几乎一切领域。它是我们实现“四化”宏伟目标必不可少的重要手段。

目前，电子计算机主要用于以下几个方面：(1) 科学计算；(2) 自动控制系统；(3) 数据处理；(4) 人工智能；(5) 计算机辅助设计等等。

电子计算机的主要特点是：运算速度快（目前国外已有每秒几亿次的计算机）、存贮数据多、精确度高、具有记忆和逻辑判断的能力，而且它的内部操作全部都是自动控制进行的。它不仅大大节省人力，提高计算效率，而且有许多工作离开了计算机是简直无法完成的。

一个计算机系统的硬设备包括：

(1) **输入设备**。将信息（如程序或数据）送入计算机内，是人——机联系的主要设备之一。常用的有光电输入机、卡片输入机和电传打字机，磁带机以及磁盘机等。

(2) **存贮器**（简称“内存”，在计算机主机体内）。内存有磁芯存贮器和半导体存贮器。一台计算机的速度、容量、精度，主要取决于它的内存贮器。它是计算机的一个重要的组成部分。计算一个问题的步骤及原始数据，中间结果及最后结果都要存放在存贮器中。（除计算机内存外，还有外存贮器 它是计算机的外部设备。外存贮器有磁盘、磁带等。它的特点是容量大，但存取速度慢，算题时是直接使用内存，而不是外存。但外存可存放大量信息，在操作系统控制下，它在方便用户和提高计算机使用效率方面能发挥很大作用。）内存被分为一个一个的单元，每个单元包括若干位，每个单元（或几个单元）中存放一个数据。存贮器的存贮单元总数是 1K（即 1024）的倍数。如一台内存为 256 K 的计算机系统有 $256 \times 1024 = 262144$ 个存贮单元。一个存贮单元可以包括 8 位（即一个字节），16 位（一个字）或 24 位、32 位、……等，这是随计算机而异的。存贮器的一个重要特点是，从它的每个单元中取出信息后，该单元中原有信息不破坏。只有当送入新的信息后，才取代了原来的信息。

(3) **控制器**。由它操纵计算机各部分按照程序的要求协调动作，也可由人工用面板开关或控制台键盘进行操作。

(4) **运算器**。计算机的各种算术运算和逻辑运算都是在运算器中进行的。

(5) **输出设备**。计算机的计算结果，或程序员需要了解的程序运行数据都必须通过输出设备显示出来。常用的有：终端显示器、电传打字机、宽行打印机、x-y 绘图仪以及磁带机，磁盘机等。

计算机系统各部分的联系见图 1.1。

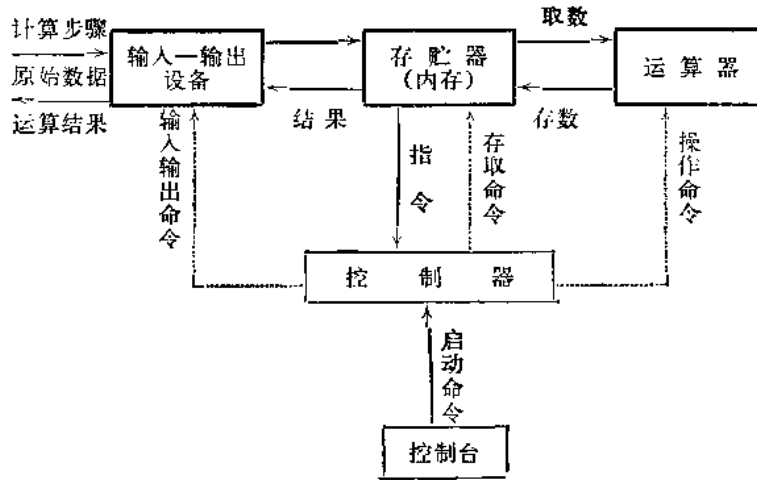


图1.1 计算机各部分联系示意图

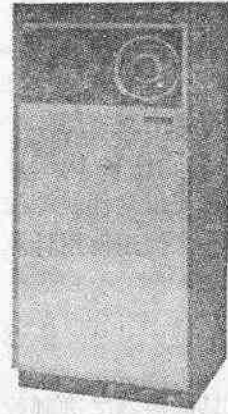
运算器和控制器合称为中央处理机(C.P.U)。计算机主机包括中央处理机和内存贮器(内存)。输入和输出设备称为计算机的外部设备(或外围设备)。计算机和外部设备的外形可见图 1.2 和图 1.3。



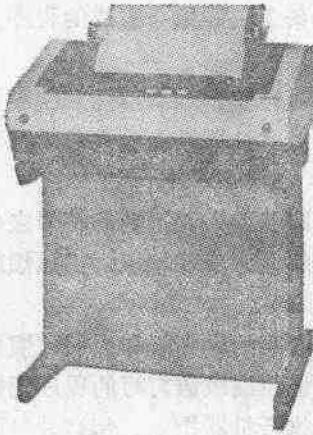
图 1.2



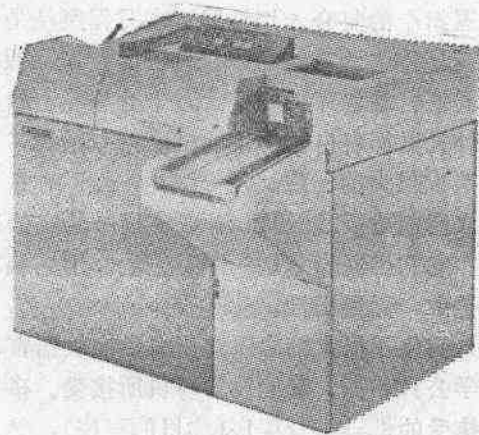
终端显示器



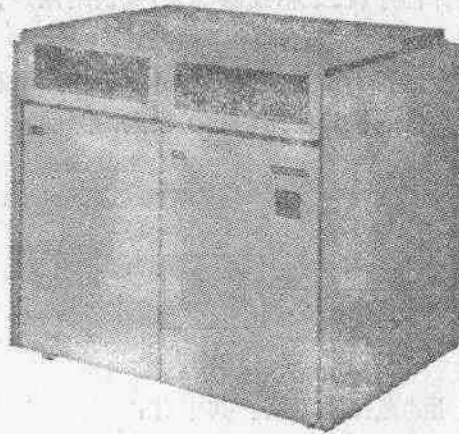
磁带机



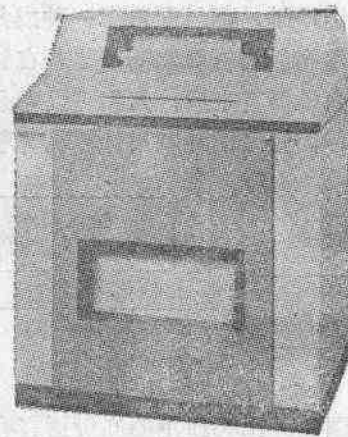
电传打字机



卡片读入机



磁盘机



宽行打印机

图 1.3

计算程序和原始数据通过输入设备输入到计算机内。根据输入的程序，由控制器发出指令，要求运算器进行操作运算，最后又根据程序的要求，将结果通过输出设备输出（如打印机在纸上打印出一些信息）。

一个计算机要正常工作和发挥效能，除了需要以上设备(硬件)外，还需要有功能完善的程序系统(称软件)，以使计算机自动进行各种所要求的工作。对一般的用户来说，更重要的是了解和使用软件。算法语言的出现，使用户可以完全不顾计算机的内部结构而如同使用算盘一样方便地使用计算机。

§ 2. 从机器语言到高级语言

计算机只能识别二进制数。由一组按一定规则由 0 或 1 组成的数字代码构成一条指令，它控制计算机进行相应的动作。这种二进制的指令是面向计算机的，因此称机器指令。

人与人对话需要用语言，人与计算机进行“对话”，也要一种特定的语言。所谓机器语言是指机器指令的集合。用机器语言写程序就是要写出由一条条机器指令组成的程序。这样编出的程序称之为手编程序。机器语言又称低级语言。

用机器语言编程序工作量大，十分繁琐，枯燥无味，直观性差，难看难懂难编难记。而且每一种机器都有自己特定的机器指令系统，互不通用。对非计算机专业人员学习和使用计算机造成很大的困难。

初期人们创造了“汇编语言”，它用一特定的助记符号，即帮助人们记忆的符号来代表数字代码，它和机器指令是一一对应的。用它编写程序与手编程序步骤相似，仍然烦琐枯燥，工作量大，无通用性。

后来人们化了很大精力研究出各种高级语言，用高级语言写的程序既和人们习惯用的语言和数学公式相似，又能为计算机所接受。事实上，是先把用高级语言写的程序翻译成计算机能接受的机器指令程序(称目的程序)，然后计算机再执行机器指令。起这种“翻译”作用的称“编译程序”。见图 1.4。每一种高级语言都有自己的编译程序。即使同一种高级语言，对于不同的计算机其编译程序也是不同的，因为每种机器有不同的机器指令系统。在计算机出厂时，都已配备了一种或几种语言的编译程序(在纸带或磁盘上)提供给用户。应该知道用高级语言算题的前提是在计算机中已装入该高级语言的编译程序。

有了高级语言之后，人们可以不必学习机器指令，也不需要懂得计算机的工作原理和内部结构，而能方便地利用计算机算题。这就为计算机的推广使用扫除了极大的障碍。

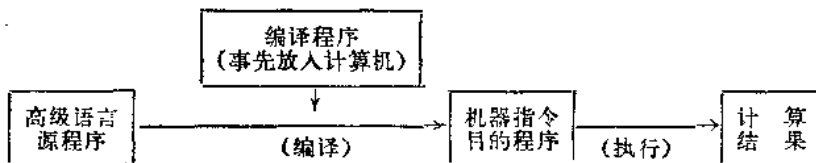


图1.4

目前国内外比较通用的高级语言有十几种，适用的范围也不同。其中有：
FORTRAN, ALGOL 60(适合于科技计算)
COBOL (适合于数据处理)

PL/1, ALGOL 68 (大型通用语言)
 BASIC (小型会话式语言)
 PASCAL (结构程序设计语言)

国内也有一些使用汉字拼音的高级语言, 如 BCY 等。

下面介绍一下用高级语言解题的过程:

- (1) 根据物理问题构造数学模型, 即归纳为数学公式。
- (2) 选择适当的计算方法。把数学公式化成适于计算机解题的方法。
- (3) 用高级语言编写程序。复杂的程序, 为使思路清楚, 往往先设计框图, 然后再编写程序。

(4) 上机算题。可事先将程序或数据穿孔(在卡片或纸带上), 然后输入计算机。一般先调试程序(试算), 试算是为了检查程序编写有否错误。试算无误后才正式算题。

用计算机进行科技计算的工作流程见图 1.5。

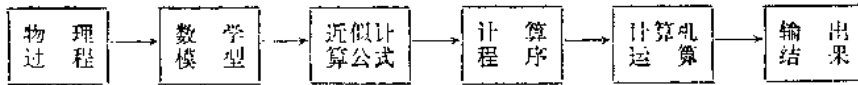


图 1.5 计算机进行科技计算的工作流程

使用电子计算机解题的全过程通常是一个循环反复、渐趋正确的过程。一个复杂的程序往往不是一次能通过的。特别对初学者, 有时反复上机, 修改几次甚至十几次才获通过。学习高级语言必须采取理论与实际相结合的方法, 除了学习语言本身的规定外, 更重要的是多编程序多上机实践。在反复实践中理解和熟悉语言的使用。

§ 3. FORTRAN 语言的发展概况

FORTRAN 是目前国际上广泛流行的一种高级语言, 它适于科学计算。FORTRAN 是英文 FORMula TRANslation 的缩写, 意为“公式翻译”。

FORTRAN 语言允许使用者将程序编写成类似数学式子的形式, 例如:

FORTRAN 语句	数学式子
X = 3.0	$X = 3$
Z = X + Y	$Z = X + Y$
T = (A + B) / (C + D)	$T = \frac{A + B}{C + D}$
W = 2. * SIN(X)	$W = 2 \sin X$

两种写法很相似(但不是完全相同)。

FORTRAN 编译系统将这种用 FORTRAN 语言写的程序翻译成机器指令, 然后由计算机执行机器指令(目的程序)。FORTRAN 语言与人们习惯用的语言(用英文表示)和数学语言比较接近, 便于推广使用。

第一个 FORTRAN 语言是 1954 年提出的, 1956 年开始使用。随后不断发展形成很多版本。其中最流行的是 1958 年出现的 FORTRAN II, 和 1962 年出现的 FORTRAN IV。由于 FORTRAN 种类多, 规定不一, 乃要求语言标准化。1964 年, 美国标准化协会提出了两个标准文本的草案, 定名为:

- 基本 FORTRAN(相当 FORTRAN II)

- FORTRAN(相当 FORTRAN IV)

1966年,又正式公布了两个美国标准文本:

- 标准 FORTRAN(X 3.9-1966)(大致相当 FORTRAN IV)

- 标准基本 FORTRAN(X 3.10-1966)(大致相当 FORTRAN II)

由于 FORTRAN 语言在国际上被广泛采用,1972年国际标准化组织 ISO(International Standard Organization)公布了完全的(一级)、中间的(二级)、基本的(三级) FORTRAN,其中一级的相当 FORTRAN IV,三级的相当于 FORTRAN II,在两者之间增加了一个二级。

根据其十多年来实践中的发展,美国标准化协会76年对1966年美国国家标准的 X 3.9-1966 FORTRAN 进行修订,增加了不少新的内容,1978年4月美国国家标准协会正式公布将它作为美国国家标准,为区别于旧的标准,把它称作“FORTRAN 77”。同时宣布撤销 FORTRAN ANSI X 3.10-1966(即标准基本 FORTRAN)。在本书中提到的标准 FORTRAN 仍然是指 1966 年的标准 FORTRAN(X 3.9-1966)。

目前在国际上最广泛流行的是 FORTRAN IV。应该指出,尽管有一个标准 FORTRAN 文本,但是几乎没有一个正式使用的 FORTRAN 编译程序是完全与标准 FORTRAN 相同的。在具体计算机上实现时,总是根据计算机的具体情况作某些修改或扩充。而且随着计算技术和计算方法的发展,又不断补充新的内容。因此,即使都是 FORTRAN IV,在每个计算机上的规定和功能也还是有所差异的。

本书以介绍标准 FORTRAN(X 3.9-1966)为主,同时对一些广泛使用的非标准规定也作简要的介绍。

§ 4. 最简单的 FORTRAN 程序分析

在介绍 FORTRAN 语言的具体内容之前,我们根据自己的体会对如何学习 FORTRAN 语言发表一些看法。

FORTRAN 语言有许多具体而繁琐的规定,有一些初学 FORTRAN 的同志因而望而生畏,甚至半途而废。我们认为:初学者不应一开始就陷入到苦学硬记一大堆规定和格式之中,而应当抓住重点,学会语句的基本用法和编制程序的技巧。在本书的编写上,不准各开头就抛出一大堆繁琐的规定,而从一、二个具体问题入手,再引出一些概念。正如学习英语不能从背英文文法开始,更不能学完文法再写句子,而必须把文法和句子结合起来学。同样,学习程序设计,也应当把学习语法规定和使用语言编制程序紧密结合起来。

对初学者来说,应当抓住基本要求,即通过较短时间的学习能够看懂和编写一般的小型程序。我们建议,在开始时不必在一些次要的、细小的问题上太钻牛角尖,更不要死背规定和定义。一些基本规定在做几次练习和上几次机以后就自然会理解和记住的。在掌握了 FORTRAN 语句的基本用法和程序设计的基本技巧以后,我们完全可以编写出一般的计算程序。其它一些“锦上添花”的工作(例如打印格式的多样化,提高计算机运行效率等),可以在以后的实践中逐步地学习。至于某些用得较少的语句和具体规定,如果读者在今后编程序时需要使用它们时,可随时查阅有关说明书。这样就可以在学习本课程时把精力集

中在主要的方面。这是我们在教学实践中的一个深切的体会。

为了使初学者对 FORTRAN 语言程序从一开始就有一个具体的感性知识，我们首先介绍一些最简单的 FORTRAN 程序，然后从中引出关于 FORTRAN 的一些基本知识。

【例 1】用 FORTRAN 语言写出程序，计算：

$$X_{1,2} = (-B \pm \sqrt{B^2 - 4AC}) / (2A)$$

其中：A=1, B=3, C=2。

程序为：

1	5 6 7	
C	THE	ROOTS OF THE QUADRATIC EQUATION
		A = 1.
		B = 3.
		C = 2.
		ROOT 1 = (-B + SQRT(B * * 2 - 4.0 * A * C)) / (2.0 * A)
		ROOT 2 = (-B - SQRT(B * * 2 - 4.0 * A * C)) / (2.0 * A)
		WRITE (10,101) ROOT 1, ROOT 2
101		FORMAT (1 X, 2 F10.4)
		STOP
		END

先对这个程序作简单的说明（在本章中，只要求大体知道程序各部分的作用即可，不要求对所有语句全部弄懂）。

第一行是注释行，说明程序的内容（求二次方程式的根）。第二、三、四行是赋值部分，分别把 1, 3, 和 2 三个值送到变量单元 A, B 和 C 中。第五、六行是运算部分，通过等号右边的表达式求出根 ROOT 1 和 ROOT 2 的值。第七、八行是输出部分，将 ROOT 1 和 ROOT 2 二个根的值按第八行给出的规定格式打印出来。第九行 STOP 表示程序停止执行。第十行 END 表示程序块至此结束。

【例 2】我们再举一个复杂一些的例子：

已知三角形的二边(A和B)和其夹角(α)，求第三边C（设第一组值为：A1=1, B1=2, $\alpha_1=0.2$ 弧度，第二组值为：A2=2, B2=4, $\alpha_2=0.1$ 弧度）

计算公式为：
$$C = \sqrt{A^2 + B^2 - 2AB \cos \alpha}$$

这个 FORTRAN 源程序分为两部分：

(1) 主程序：

1 2 3 4 5 6 7 8

5	A 1 =1.0 B 1 =2.0 ALFA 1 =0.2 A 2 =2.0 B 2 =4.0 ALFA 2 =0.1 C 1 =C (A 1, B 1, ALFA 1) C 2 =C (A 2, B 2, ALFA 2) WRITE (10, 10) C 1, C 2	
10	FORMAT (1 X, 2 F10.4) STOP END	

(2) 子程序:

1 2 3 4 5 6 7

C F U N C T I O N	S U B P R O G R A M FUNCTION C (A, B, ALFA) D=A*A+B*B-2.0*A*B*COS (ALFA) G=SQRT (D) RETURN END	
-------------------	---	--

对这个程序作些简单介绍: 整个程序由一个主程序和一个子程序组成。子程序是根据公式 $C^2 = A^2 + B^2 - 2AB\cos\alpha$ 计算 C 值。子程序的第四行 SQRT 是平方根函数。我们再来看主程序, 第一到六行是赋值部分, 依次把 1.0, 2.0, 0.2, 2.0, 4.0, 0.1 六个数值赋给 A1, B1, ALFA 1, A2, B2, ALFA 2 六个变量。第七行是调用子程序, 将 A1, B1, ALFA 1 的值代入到子程序 C 中代替 A, B 和 ALFA。并将子程序求出的 C 值带回主程序赋予 C1。同样, 第八行的作用是用通过子程序求出 C2 值。子程序中的 RETURN 语句表示执行完子程序后返回主程序。主程序中第九、十行为输出部分, 它将 C1 和 C2 值按第十行给定的格式输出。最后停止运行。(对以上程序如未能全弄懂, 可以先采取“不求甚解”的态度。以后会详细介绍的)。

从以上两个简单的例子中, 可以看到:

一、一个 FORTRAN 源程序由一个主程序或者一个主程序和若干个子程序组成。主程序和每一个子程序分别是一个独立的程序块(又称程序单位)。每一个程序块(程序单位)都要以 END 行作为结束。因此, 一个源程序如果包括几个程序块(程序单位), 则每一个程序块都应分别以 END 行结束(即除了主程序外, 每一子程序也都要以 END 行结束)。

在编译时, FORTRAN 系统对源程序按块进行编译, 各块分别通过编译变为以机器指令表示的目的程序, 然后再装配成一个统一的程序, 进行运算。主程序或子程序如同积木一样, 根据需要进行装配, 组成一个整体。这样可以把常用的子程序编好, 调试通过后

放入磁盘(磁带)。编程序时,主程序就可以尽量简单,以提高编制和调试程序的效率。

二、每一个程序块是由若干行组成的。行是FORTRAN程序的基本单位。一行内只能写一个语句,不能在一行内连续写几个语句。而一个语句在一行内写不完,可以写在几行内。程序中的行可以是语句行(如READ语句、WRITE语句),也可以不是语句行(如注解行、结束行)。

三、一个简单的FORTRAN源程序一般主要由以下几个部分组成:

1. 提供数据部分
2. 运算部分
3. 输出部分

此外,还可以有说明部分等。

四、语句前面可以没有标号。在需要时也可以设标号。程序的执行是按行的出现先后次序进行的,标号大小不起执行顺序的作用,只作为查找访问的标志。如:

```
10      WRITE(10,100)C
5       A=1.0
1       B=5.3
```

执行的顺序是10→5→1,而不是1→5→10。标号的大小范围为从1到99999之间,标号必须是无符号的整数(不能是带小数点的实数,负数或变量)。一个程序块中不能有两个或二个以上标号相同的行。

五、FORTRAN源程序各行必须按一定格式书写,每一字符写在一格内。(详见下节)。

概括起来说,FORTRAN程序的特点是:程序分块,块分为行,行分为格,顺行执行。

§ 5 FORTRAN源程序的书写格式

FORTRAN源程序有严格的书写格式,必须按规定书写。标准的FORTRAN程序格式如图1.6所示。在它上面写的是上一节中例1的程序。

一张程序纸包括若干行(不规定固定的行数),但必须是80列,即每一行上有80格,每一格内可填写一个字符。

一、每一行分为三个区:

1. 标号区。由第1列到第5列,如果某一个语句需要写标号,就写在这个区内。标号最多为5位数字,它的范围是1到99999,可以是任意的无符号的整数。数字前、后和数字中间的空格不起作用。如123和123和12 3, 1 2 3的作用完全一样。

注意,标号不能超越出标号区的范围,如写成:“12 345”,虽然也是五位数字,但由于中间加了空格,占了六列,超过了标号区范围,就会出现错误。

2. 续行区。第6列为续行标志区。当一个语句在一行内写不下时,可以继续在下—行接着写。在下—行第6列的位置上写一个非零和非空格的任一字符(必须是FORTRAN规定的字符范围内),则表示下—行是上一行的继续行。如图1.6中的第七行的第6列上写了“1”,表示这一行中从第7列到72列的字符是上一行的继续。在编译时把继续行的第一个字符紧接着上一行的最后一个字符。一个语句最多可以有19个继续行(即一个语句最

清华大学计算中心	FORTRAN		程序名称 求一元二次方程式的根	设计人	共 页 第 页	日期									
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
C, THE	ROOTS OF	THE QUADRATIC EQUATION													
	A=1.0														
	B=3.0														
	C=2.0														
	ROOT 1 = (-B - SQRT (B**2 - 4.0 * A * C)) / (2.0 * A)														
	ROOT 2 = (-B + SQRT (B**2 - 4.0 * A * C)) / (2.0 * A)														
	WRITE (1, 0) ROOT 1, ROOT 2														
	FORMAT (1X, 2F10.4)														
	STOP														
	END														

图 1.6