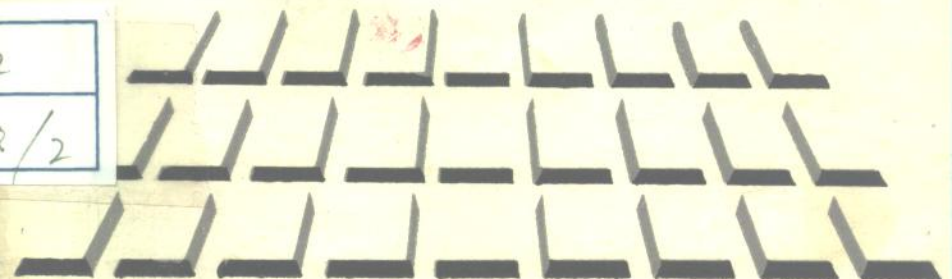
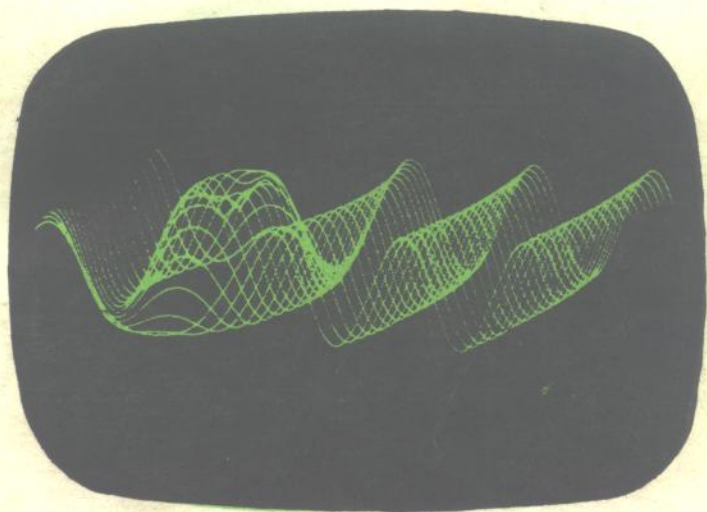


# FORTRAN 程序设计

谢柏青 编著



## **FORTRAN程序设计**

谢柏青 编著

责任编辑：李采华

•  
北京大学出版社出版

(北京大学校内)

北京通县向阳印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

787×1092毫米 32开本 9.75印张 200千字

1986年5月第一版 1986年5月第一次印刷

印数：00001—21,000册

统一书号：13209·128 定价：1.65元

## 内 容 简 介

本书系统地介绍了 FORTRAN 程序设计语言、程序设计方法和技巧,以及一些科研与工程技术常用算法。

书中的每个程序力求简明、扼要、准确,重要算法配有教学方法简述及程序说明,算法以子程序形式给出,便于读者直接使用。例题选自科研及工程技术中常用算法。每章附有一定数量习题,均已上机通过,可作为上机题选用。

本书适合作高等院校理工科大学生及研究生的教材,也可作为科研、工程技术及其他人员的自学参考书。

## 前 言

在科学研究、文化教育、工农业生产、商业和交通运输组织、经济管理和国防建设等各个领域已广泛地使用电子计算机技术。电子计算机的使用又加快了各个领域的发展速度。

人们要指挥电子计算机来完成一定的工作，就需要有可以为计算机了解的语言，就是机器语言。机器语言因机型而异，不便人们使用，随着计算机的广泛应用，逐步发展出算法语言，人们运用算法语言指挥计算机工作，它可由编译程序或解释程序翻译成机器语言。算法语言和一般人的语言不同，它具有严格的逻辑性和含义的明确性，即必须有严格的语法规则；同时由于算法语言是人们和计算机联系的媒介，又应该尽可能和人们一般惯用的语言相近。根据这两方面的要求，在计算机发展过程中，自五十年代以来已经提出了许多种算法语言(也称为计算机高级语言)，如FORTRAN, ALGOL60, COBOL, BASIC, PL/1, APL, LISP等,近年来又发展了PASCAL语言和C语言等。

FORTRAN语言是国际上广泛流行的算法语言之一。FORTRAN是英文FORMula TRANslator的缩写，原意是“公式翻译程序”。它于1954年提出，1956年开始使用，在此后的廿多年中有很发展，其原因是由于它适用于科学研究，同时在大型、小型和微型计算机上都可采用。

FORTRAN的发展过程：1958年出现FORTRAN I, 1962年出现FORTRAN IV, 由于种类规定不一，要求语言标准

化，于1964年美国标准化协会提出两个标准文本草案，定名为：

- 基本FORTRAN (相当FORTRAN I)
- FORTRAN (相当FORTRAN IV)

1966年正式公布了两个美国标准文本：

- 标准 FORTRAN (X3.9-1966) (相当 FORTRAN IV)
- 标准基本 FORTRAN (相当FORTRAN I)

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，其中一级相当 FORTRAN IV，二级相当 FORTRAN I。1976年开始美国标准化协会对美国国家标准 X3.9-1966 FORTRAN 进行修订，1978年将修订文本称为“FORTRAN 77”，并作为美国国家标准。

FORTRAN 语言在科学和工程计算中积累了许多算法，有通用的算法，也有针对解决某一类问题而编制的程序包。这种算法大部分是以过程的形式(即子程序或函数段)出现。

本书考虑到理工科大学生、研究生的特点以及科学工作者及工程技术人员的需要，结合实例讲解 FORTRAN 程序设计方法，并讲授子程序调用、数据交换等问题，使读者学会正确地编制程序和使用 FORTRAN 程序包来解决科研及生产中的实际问题。

在满足一定计算精度的要求下，一个程序质量好主要有两点标志：运行时占用机器时间较短及占用内存较少，而这两者又常常是互相矛盾的，前者在多次重复的计算中显得很突出，后者在大数据量的计算中处理不好就会产生内存溢出的问题，在使用微型机时更为突出。本书力图从程序设计的

方法上来说明这些问题，使读者掌握其要领。

目前国际上流行的 FORTRAN 为 FORTRAN IV 和 FORTRAN 77，尽管有标准文本，但尚没有一个正式使用的 FORTRAN 编译程序与标准文本完全相同。在计算机实现时都需作具体的修改和扩充，因此学习 FORTRAN 语言要注意每个计算机功能的差别。

为适应我国广泛使用微型机的形势，本书结合微型机使用的 FORTRAN-80来进行讲述，对标准 FORTRAN 有修改和扩充，如扩充了计算的功能，允许混合运算和混合赋值；无复型及双精度复型数；语句函数中可使用下标变量；继续行无限制；在可使用整型常数处可使用十六进制常数等等。

本书列举的程序及全部习题中的程序均在 CROMEMCO、APPLE II 及 NEC PC-8001 上通过。NEC PC-8001 和 APPLE II 型机上 F80、ED、L80、FORLIB 等软件均在软盘上提供与 CP/M 操作系统一起使用。CROMEMCO FORTRAN IV 在软盘上提供与 CROMEMCO 操作系统一起使用。用简单的几条指令可对 FORTRAN 源程序进行编辑、保存、编译、连接和运行，简化了操作手续，合理组织了计算机的工作流程，方便了用户。

本书是在北京大学教学讲义的基础上经修改而完成的。由于本人水平有限，有错误及不足之处欢迎批评指正。

周芝英同志对本书进行了认真的审阅，并提出许多宝贵意见，特在此致谢。

谢柏青

一九八四年十一月

# 目 录

<b>第一章</b>	<b>FORTRAN 的基本概念</b> .....	(1)
§1	FORTRAN 的书写格式与程序框图 .....	(1)
§2	FORTRAN 的程序结构 .....	(5)
§3	FORTRAN 中数的表示方法 .....	(7)
§4	变量 .....	(11)
§5	数组 .....	(13)
§6	函数 .....	(17)
§7	表达式 .....	(20)
	习题一 .....	(25)
<b>第二章</b>	<b>FORTRAN 的基本语句</b> .....	(27)
§1	赋值语句 .....	(27)
§2	有格式输入/输出语句 .....	(30)
§3	程序语句 .....	(50)
§4	停、暂停与结束语句 .....	(51)
§5	数据初值语句 .....	(52)
§6	两种条件语句 .....	(53)
§7	三种转移语句 .....	(59)
§8	循环语句与继续语句 .....	(66)
§9	定义函数语句 .....	(80)
§10	本章程序举例 .....	(82)
	(一) 用牛顿迭代法求方程的根 .....	(82)
	(二) 求多项式之值 .....	(86)
	(三) N 个数排序的三种方法 .....	(88)

(四) 验证哥德巴赫(Goldbach)猜想	(93)
(五) 计算微型机的相对机器准确度的程序	(95)
习题二	(96)
<b>第三章 子程序</b>	(102)
§1 函数子程序	(102)
§2 子例程子程序	(105)
§3 可调数组	(110)
§4 外部语句	(112)
§5 虚实结合	(117)
§6 本章程序举例	(132)
(一) 解 $n$ 阶线性方程组	(133)
(二) 求矩阵逆	(143)
(三) 使用差分插值方法求未知的函数值	(144)
(四) 随机数产生与检验	(148)
习题三	(153)
<b>第四章 数据联系语句与数据块子程序</b>	(160)
§1 公用语句	(160)
§2 等价语句	(168)
§3 数据块子程序	(175)
§4 本章程序举例	(178)
(一) 参数积分	(178)
(二) 重积分	(182)
(三) 线性曲线拟合	(187)
习题四	(195)
<b>第五章 输入与输出</b>	(200)
§1 格式语句的补充说明	(200)
§2 数组格式和自带格式输入/输出语句	(216)



§3	无格式输入/输出语句 .....	(221)
§4	辅助输入输出语句.....	(229)
§5	编码/译码语句 .....	(232)
§6	本章程序举例.....	(234)
	(一) 用两种方法打印直方图 .....	(234)
	(二) 打印正态分布随机数的分布直方图 .....	(243)
	习题五.....	(247)
<b>第六章</b>	<b>程序设计中应注意的问题及上机操作.....</b>	<b>(252)</b>
§1	程序设计中应注意的一些问题.....	(252)
§2	FORTRAN程序调试 .....	(262)
<b>附 录</b>	<b>.....</b>	<b>(280)</b>
附录一	ASCII 字符代码 .....	(280)
附录二	标准 FORTRAN 库函数 .....	(282)
附录三	FORTRAN 错误信息 .....	(285)
附录四	FORTRAN 语句一览表 .....	(290)
<b>参考文献</b>	<b>.....</b>	<b>(295)</b>

# 第一章 FORTRAN 的基本概念

## §1 FORTRAN 的书写格式与程序框图

本节以求一元二次方程  $ax^2+bx+c=0$  的实根为例说明书写格式与程序框图。要求从键盘输入方程的系数，在输入一组系数后打印出方程的两个根，当  $a=0$  或方程无实根时停止计算。

要用 FORTRAN 来解决这个问题，首先可画出如图 1-1 所示的程序框图。画程序框图是程序设计的重要一环，特别是复杂的程序用框图表示，就可以看出程序的走向。正确完整的框图可表达出程序设计思想，有助于正确迅速地编程序，也便于程序编制者与使用者交流思想。程序框图中各框的意义说明如图 1-2 所示。

框图也称为流程图，它是由图 1-2 各框加上文字说明再用指向线连接起来的图形，用来表示程序执行的顺序及程序的走向。图 1-1 就是解一元二次方程的程序框图。

下面写出此例的 FORTRAN 源程序：

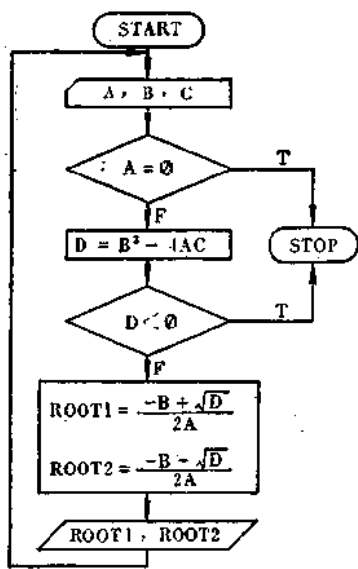
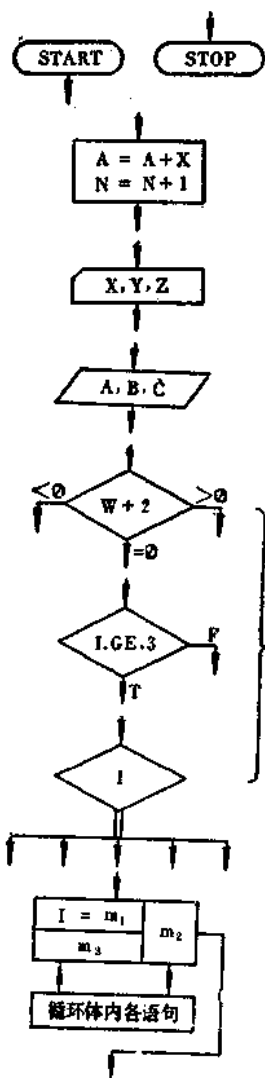


图 1-1 解一元二次方程的框图



圆框表示启动(START)、暂停(PAUSE)、停(STOP)、子程序入口、返回(RETURN)等

矩形框表示叙述肯定的动作(计算或其它功能), 只有一个入口, 一个出口

卡片形框表示将数据送入程序中的变量中去(例中为 X, Y, Z)

平行四边形框(或打印纸形框)表示将变量之值输出到显示器及打印机(此例中为 A, B, C)

菱形框表示逻辑判断的功能, 在FORTRAN中有左面三种形式:

- (1) 一个入口, 三个出口
- (2) 一个入口, 两个出口
- (3) 一个入口, 多个出口

循环框表示循环的功能

I 为循环控制变量,  $m_1$ ,  $m_2$ ,  $m_3$  为循环的初值、终值及步长值

图 1-2 FORTRAN 中使用的框及说明

1	5	6	7	72	73	80
C	THE		REAL ROOTS OF QUADRATIC EQUATIONS		1	
			PROGRAM F.NDRT		2	
50			READ(1, 10)A, B, C		3	
10			FORMAT(3F4.1)		4	
			IF (A.EQ.0.0) STOP		5	
			D=E*B-4.0*A*C		6	
			IF (D.LT.0.0) STOP		7	
			ROOT <sub>1</sub> =( -B+SQRT(D) )/(2.0*A)		8	
			ROOT <sub>2</sub> =( -B-SQRT(D) )		9	
		1	/(2.0*A)		10	
			WRITE(1, 20)ROOT <sub>1</sub> , ROOT <sub>2</sub>		11	
20			FORMAT (1X, F6.2, 5X, F6.2)		12	
			GOTO 50		13	
			END		14	
标号区	续行区		语 句 区			注释区

由以上看出一个 **FORTRAN** 源程序由若干行组成,行数的多少不受限制。每行有 80 列, 从左至右为 1—80 列编号, 每个格可填写一个 **FORTRAN** 字符。

要按规定将 **FORTRAN** 源程序书写在专用程序纸上, 上机操作时, 也必须按此格式送入源程序。其书写格式综述如下。

**FORTRAN** 源程序按列分为四个段:

语句标号段或称标号区(1—5列) 如果某一语句需要语句标号就写在这个区内。标号可用 1—99999 的无符号的整数, 数字前后的空格及中间的空格都不起作用, 如 123 与 1□2□3 作用相同。在每个程序块内不得重复使用相同的语句标号。在此例中第 3、4 和 12 行都使用了语句标号。在这个区内第 1

列若有字母 C, 该行为注释行, 若同时写有语句标号, 则语句标号不起作用。

**继续字符段或称续行区 (6 列)** 当一行写不完一个语句时, 可在下一行接着写, 这时下一行的第 6 列需要加一个非零、非空格的任一字符, 表示此行是前一行的继续行。第 6 列所加的字符也可叫做续行号。有的计算机中规定连着最多可用 19 个继续行, 而一般微型机可有任意个继续行。

**语句段或称语句区 (7—72 列)** 可以在这区域中填写任何 FORTRAN 语句, 注意语句可从其中任一列开始, 在任一列结束或换行。每一列只能打一个字符。

**说明段或称注释区 (73—80 列)** 该段为程序员作标记用, 可写行的编号, 以便查找, 也可写程序的注解等。这部分内容不进行编译, 也不影响程序的运行。

**FORTRAN 源程序中的行可分为四类:**

**注释行** 该行第 1 列有字母 C, 在其它列用户可写任何字符, 编译时该行内容不予编译, 对运行不产生任何影响。在列程序清单时原样印出, 起备忘录的作用。

**结束行** 该行为程序块的最后一行。1—5 列可填写语句标号, 第 6 列空白, 7—72 列有 END 三个字符相继出现, 中间可以有空格。结束行可跟在任何其它类型的行的后面, 表示程序块的结束。

**初始行** 该行为语句的第一行或单行的语句。可从语句区的任何部分开始写, 在语句段可填写全部或部分语句。该行 1—5 列可有语句标号, 第 6 列必须为空白或零。

**继续行** 若在第 6 列有非零、非空格的字符, 则表示该行是上一行的继续, 这时语句标号段的内容在编译时被忽略, 但若第 1 列有字母 C 仍应作注释行处理。

在此例中，第1行为注释行，2—9及11—13行为初始行，第14行为结束行，第10行为继续行。

## §2 FORTRAN 的程序结构

(一) FORTRAN 程序为块状结构，可由一个主程序块及多个(或一个)子程序块组成，也可只有主程序块而没有子程序块。下面举例说明 FORTRAN 程序的块状结构。

例 已知梯形上底为  $a$ ，下底为  $b$ ，高为  $h$ ，求梯形面积  $S = h(a+b)/2$  的 FORTRAN 源程序。

程序

```
PROGRAM S2
  READ(1, 1) A1, B1, H1, A2, B2, H2
1  FORMAT(6F5.1)
  CALL SS(A1, B1, H1, S1)
  CALL SS(A2, B2, H2, S2)
  WRITE(1, 2) S1, S2
2  FORMAT(1X, 2F10.2)
  STOP
  END
  SUBROUTINE SS(A, B, H, S)
  S=(A+B)*H/2.
  RETURN
  END
```

在此例中 FORTRAN 源程序由一个主程序块和一个子程序块组成，每块程序都用 END 作结束标志。每个程序块都是独立的。独立的含意有两层意思，一是指每个程序块中可以独立地使用变量或语句标号，如该例中主程序中用 A1，

$B_1, H_1$ 和 $A_2, B_2, H_2$ 分别表示两个梯形的上底、下底及高，用 $S_1$ 和 $S_2$ 分别表示两个梯形的面积，而在子程序中则用 $A, B$ 和 $H$ 表示梯形的上、下底和高，用 $S$ 表示梯形的面积；二是每个程序块独立地进行编译，即一块一块地编译，这样便于程序的调试，分块编译后一拼凑就可以了。在作较大程序时主程序及子程序可分别放在不同的文件中，可以单独编译，在运行时再进行连接。

(二) 每个程序块由一组有次序的语句组成。每个语句写一行，可以设标号，也可以不设标号。程序执行时按行执行，标号不起执行顺序的作用，只作为访问查找的标志。每个语句遵照FORTRAN语句的规定按FORTRAN源程序的书写格式来进行书写，可使用FORTRAN字符集的所有字符。

(三) FORTRAN字符集分四个子集：

字母集 共27个

A B C D E F G H I J K L M N  
O P Q R S T U V W X Y Z \$

数字集 共10个

0 1 2 3 4 5 6 7 8 9

注意：程序中的数字是有意义的数值时，不作特殊注释都可看成十进制数，但在特殊情况下看作16进制数，这时使用如下字符：

0 1 2 3 4 5 6 7 8 9 A B C D E F

字母数字集 所有字母和数字组成的子集。

专用字符集 共11个

+   -   \*   /   (   )  
空格 加号 减号 乘号 斜杠 左括号 右括号

逗号 圆点 单引号 等号

需要说明的是任何可打印的字符都可以出现在文字型的场中。因各机型允许使用的符号不同，当使用 FORTRAN 字符集以外的字符时，程序在不同机型的互换会带来一些麻烦，最好少使用。

另外，空格字符在打印纸上及显示器上都是空白，即不输出任何字符，空一个格，为读者看书方便，有时用记号 □ 表示空格符号，在程序书写时就不写 □ 记号，而留出空格。

### §3 FORTRAN 中数的表示方法

FORTRAN 语言中有七种类型的常数。

#### (一) 整型常数(INTEGER)

它在机内占两个字节(即 1/2 内存单元)。数的范围为  $-32768$  至  $+32767$  (即  $-2^{15}$  至  $+2^{15}-1$ )。它是正整数、零、负整数在此范围内的精确表示。如超过  $+32767$  或比  $-32768$  还小都称为溢出。数的有效数字可达十进制的五位。

在 FORTRAN 程序中用此范围的整数表示，如 382, -5, -32476, 2680 等不带小数点的数都是整型常数。

#### (二) 逻辑型常数(LOGICAL)

它在机内占一个字节(即 1/4 内存单元)。逻辑型常数只有两个值：真(.TRUE.)和假(.FALSE.)。在机内全零字节表示假，非零字节表示真。若将逻辑常数当作一个字节的整数使用，其值范围为  $-2^7$  至  $+2^7-1$  (即  $-128$  至  $+127$ )。有效位可达十进制三位。

FORTRAN 程序中逻辑型常数用 .TRUE. 和 .FALSE. 表示真和假。值得注意的是两边的圆点不能省去。



### (三) 实型常数(REAL)

它在机内占4个字节(即1内存单元)。内存中以浮点形式存放,它是正、负或零实数的近似表示。实型常数的范围约为 $\pm 10^{-38}$ 至 $\pm 10^{+38}$ ,有效数字可达十进制的七位。

在FORTRAN程序中实型常数表示形式为:

#### 1. 带小数点的数

$\pm.f \quad \pm i.f \quad \pm i.$

#### 2. 带指数的数

$\pm.fE\pm e \quad \pm i.fE\pm e \quad \pm i.E\pm e \quad \pm iE\pm e$

其中*i*,*f*及*e*分别表示整数部分、小数部分及指数部分;正(+)和负(-)号表示可任选。

实型常数若表示为 $rE\pm e$ ,则值为 $r \times 10^e$ ,其中 $0.1 \leq |r| < 1$ 时 $|e| \leq 38$ ,若*r*的有效数字超过十进制的七位,则机内只保留前七位有效数字。

实型常数举例:

345.    -.123476    12.8    为带小数点的数

.52E+3    -2.8E4    -2.E-6    -.638E+2

为带指数的数

由以上看出,一个实型常数可以写成许多形式:

1356.0    1.356E+3    13.56E+2

13560E-1    0.1356E+4    1356E0

.....

对实数FORTRAN规定一种规格化形式,其指数前面为小于1且小数点后第一位不为零的数,即上例中 $0.1356E+4$ 为规格化形式。这种规格化形式的实数称为基本实常数。在输入时实型常数按格式要求输入,取哪种形式都可以;输出时若按E转换取规格化形式。