

FORTRAN

程序设计导引

李 木 杨家全编

辽宁科学技术出版社

FORTRAN

程序设计导引

李 木 杨家全 编

辽宁科学技术出版社

1984年·沈阳

FORTRAN 程序设计导引

李 木 编
杨家全

辽宁科学技术出版社出版 (沈阳市南京街6段1里2号)
辽宁省新华书店发行 朝阳六六七厂印刷

开本: $850 \times 1168^{1/32}$ 印张: $11^{1/4}$ 插页: 2 字数: 260,000
1983年8月第1版 1984年9月第2次印刷

责任编辑: 张家振 责任校对: 李秀芝
封面设计: 张 悦

印数: 8,101—26,300
统一书号: 15288·38 定价: 1.45元

前 言

为普及电子计算机的使用，我们很想写一本适用于科技计算问题的 FORTRAN 程序设计的入门书。现在，这本书终于和读者见面了。这本书主要是写给工程技术人员看的，也可供大专院校有关师生参考。

FORTRAN 程序设计语言与其他语言一样，是有其严格描述方法的。在本书中，我们只对 FORTRAN 做具体的解说，不做严格地定义。本书把重点放在它的具体使用上，对每一种语句主要以实例说明之，并给出它的具体程序和必要的结果。本书之所以举了这么多的例子，目的在于使读者更好地了解这种语言的特点及其程序设计的基本方法。虽然程序设计入门并不难，但是没有足够多的例子还是不能很好理解和消化的。况且，一个好的例子，胜过一百句空话，它往往能给人以有益的启示。

由于程序设计语言是描述算法的，所以，我们尽量选一些与算法相关联的例子。书中牵涉的例题有下述几个方面：矩阵的基本运算；级数求和；数据排队；方程式的迭代解法；数值积分；求解线性方程组的高斯消去法以及高斯—赛德尔迭代法；解常微分方程初值问题的龙格—库塔方法等等。鉴于笔者水平所限，文中所举例题广泛性，普遍性及其先进性都还很不够。但读者只要懂得语言程序设计的基本方法，掌握语言的使用原则，了解必要的计算方法，就不难在自己的工作实践中运用它，以达到越来越熟练的程度。

人们常说：会画框图就会编程序。这是颇有道理的。事实上，一个实际课题，画出了它的框图就等于完成了它的程序设计的一大半任务，剩下的事情就是按照算法语言抄写框图的工作了。本书之所以选配这么多的例题也和这一点有关。书中大多数的例题都配有框图，目的在于使初学者能过好框图关。

为了照顾各方面的读者，各章所配的例题深浅程度不同。对希望掌握 FORTRAN 语言基本知识的读者，只需读本书的第一章。对需用 FORTRAN 语言算题的同志，我们建议在读完第一章之后，沿着三、四、五、六、七章的顺序读下去，至于第二章，可在读各章的同时随时去翻阅它，待读完第六章之后，回过头来再读它。对从未接触过计算机的读者，略去一些不易掌握的例题，对于掌握这种语言的基本内容和程序设计方法是没有多大影响的。

本书的编写工作曾得到我所有关领导和同志的支持，尤其是李明沛、何夫祥二同志始终给以热情的关怀和鼓励；东北工学院软件工程教研室的王惠德同志曾耐心地阅读了本书的初稿并提出了许多宝贵的意见。这里谨向他们致以深切的谢意。笔者还要感谢邱玉圃老师，因为他编著的《FORTRAN 程序设计》一书对笔者改写本书的初稿提供了不少的帮助。

由于我们水平不高，时间也比较仓促，错漏、疏忽和不妥之处在所难免，恳请读者批评指正。

中国科学院沈阳计算技术研究所

李 木 杨家全

1979年5月初稿

1980年6月修改

内 容 简 介

本书介绍 FORTRAN 语言及其程序设计的基本知识。全书共分七章。第一章是 FORTRAN 语言的概貌，第二章是输入输出，第三章至第七章较系统地介绍了 FORTRAN 程序设计的基本内容。

这是一本入门性读物。它的主要读者对象是初学 FORTRAN 语言并着手用它编写程序的一般工程技术人员。也可供大专院校有关专业的师生参考。

目 录

前言

第一章 FORTRAN 语言概貌	1
§ 1. FORTRAN 的基本字符和常数	2
§ 2. 类型的隐、显说明与维数语句	6
§ 3. 表达式	12
§ 4. 语句和输入/输出简介	17
第二章 输入/输出	62
§ 1. 格式读、写语句	63
§ 2. 格式语句	71
§ 3. 无格式读、写语句与辅助输入/输出语句	89
§ 4. 动态格式说明及其应用	98
第三章 三种基本语句	108
§ 1. 赋值语句	108
§ 2. 转向语句	113
§ 3. 条件语句	126
第四章 循环	148
§ 1. 循环语句	150

§ 2. 继续语句	170
第五章 过程	185
§ 1. 标准函数	186
§ 2. 语句函数	191
§ 3. 子例程子程序	209
§ 4. 函数子程序	253
§ 5. 外部语句	263
§ 6. 过程子程序的有关问题	270
§ 7. FORTRAN 通用程序的编写	282
第六章 程序段的数据联系与置初值	295
§ 1. 公用语句	295
§ 2. 等价语句	315
§ 3. 数据语句和数据块子程序	326
第七章 逻辑、双精度与复型计算的例	339
§ 1. 逻辑计算的例	339
§ 2. 双精度计算的例	343
§ 3. 复型计算的例	348
参考文献	353

第一章 FORTRAN 语言概貌

FORTRAN 是 FORMula TRANslator 的缩写。这种程序设计语言起源于美国。美国的 IBM 公司于 1957 年在 704 计算机上就引进了 FORTRAN I，接着在 709 计算机上引进了 FORTRAN II，在 1962 年为 7090 和 7094 两个机种引进了 FORTRAN IV，不久之后，也为其他厂家所采用。自 1956 年 FORTRAN 问世以来，它的发展很快，变种也较多，现已成为国际上十分流行的一种程序设计语言了。在我国，它也正在流行起来，国内研制的类似 FORTRAN 的语言，如 BD-200 的编译程序与 KCY 的编译程序都已投入运行，还有几个 FORTRAN II 与 FORTRAN IV 的编译程序也正在运行之中。

由于计算机系统的不同，所以产生了不同的 FORTRAN 语言。为了统一起见，1962 年就有人着手做标准化的工作，1966 年形成了美国标准 FORTRAN（分为基本 FORTRAN 与 FORTRAN 两种，分别大致对应于标准化前流行的 FORTRAN I 与 FORTRAN IV）；1972 年又形成了国际标准 (ISO) FORTRAN（分三级，除头尾两级分别大致对应于 FORTRAN I 与 FORTRAN IV 外，中间还加了一级）。国内现正在做 FORTRAN 语言的标准化工作，以形成我国自己的标准。

FORTRAN 是一种成熟的程序设计语言，特别是它的标准程序库非常丰富，许多生产、科技问题都有标准程序可利用，这对加快工作进度和提高工作效率是十分有利的。

本书介绍的 FORTRAN,基本上是以(ISO)FORTRAN 为基础的,为了具体说明问题,有时还引用了 FELIX C—256 机器上所使用的 FORTRAN 语言以及其他机种的 FORTRAN 语言。尽管 FORTRAN 语言反映了某些具体计算机的特点,但它并非真正依赖于这些特点。虽然机种不同,语言略有差异,但基本内容和形式却是相同的。因此,掌握本书内容之后,在具体计算机上使用 FORTRAN 语言编程序,只要熟悉一下具体计算机的语言说明书,注意到可能有的差异,自然是容易掌握的。

我们对 FORTRAN 只做具体解说,不讨论它的严格定义。在第一章里,分四节介绍 FORTRAN 语言的概貌,其中 § 4 对 FORTRAN 的各种语句给予了通俗的说明,以使读者对 FORTRAN 语言的全貌有所了解,同时也为学习以后各章打下一个良好的基础。

§1. FORTRAN的基本字符和常数

一、基本字符

每种语言文字都有它自己的基本字符, FORTRAN 语言规定它使用如下的基本字符:

1. 大写拉丁字母: 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
2. 阿拉伯数字: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
3. 专用字符:

专 用 字 符	字 符 的 名 称
	空白
=	等号, 赋值号
+	正号, 加号
-	负号, 减号
*	星号, 乘号
/	斜杠, 除号
(	左括号
)	右括号
,	逗号
.	黑点
\$	币号

除基本字符外,在 FORTRAN 程序中不得使用其他字符;不过,在有的 FORTRAN 语言中还使用了引号以及其他字符等等。

二、常数

FORTRAN 语言中规定了如下六种常数: 整型常数、实型常数、复型常数、逻辑型常数、双精度型常数和文字型常数。

1. 整型常数

如果一个常数, 它的值是一个非指型整数值, 那么它就是一个整型常数。例如, 3, -5, +10, 0, -23457 都是整型常数。整型常数在机器内能够精确表示。整型常数 N 的变化范围, 因机种不同而异, 例如在 FELIX C-256 机器中, $-2^{31} \leq N \leq 2^{31} - 1$ 。一个整型常数是一个单字, 在机内占用一个单元。

2. 实型常数

FORTRAN 中的实型常数有两种表示方法:

①基本实常数

通常的带小数点的数叫基本实常数，例如，12.0，-2.41，0.0 等都是。不过，在 FORTRAN 中，这种实常数和通常的书写方法还稍有不同，即允许略去小数点后面的纯零，例如 12.0 可以写作 12. 而 0.0 可以写作 0.。

在基本实常数中，小数点一定要有，这是和整型常数的区别所在。

②指型实常数

指型实常数是用以表示科学记法的数的。例如：

5.0E3	[5.0×10^3]
5E + 3	[5.0×10^3]
0.254E - 9	[0.254×10^{-9}]
1E5	[1.0×10^5]
5.0E + 03	[5.0×10^3]

注意，E 之前或为基本实常数或为整型常数，E 之后必须是整型常数。

如下写法是错误的：

3.23E5.1	×
-2.50E - 3.6	×

实型常数的变化范围也是因机而异，例如，在 FELIX C—256 机中，实型数 R 的绝对值范围是 $10^{-77} < |R| < 10^{77}$ ，当 $|R| < 10^{-77}$ 时，R 被当作 0；当 $|R| > 10^{77}$ 时，运算溢出，引起停机。一个实型常数也是一个单字，在机内占用一个单元。实型常数在机内往往不能精确表示，对于 FELIX C—256 机这种数具有 6 位有效数字。

3. 双精度型实常数

双精度型实常数的表示方式与指型实常数表示方式相同，只要把字母 E 换成 D 就行了。

这种双精度型实常数在机内往往不能精确表示，只能是个

近似值。它的变化范围也是因机而异，例如在 FELIX C—256 机中，双精度数D的绝对值满足不等式 $10^{-77} < |D| < 10^{77}$ 。这种数具有15位有效数字(十进制)。一个双精度数是一个双字，在机器内占用相邻的两个单元。

例：3.14159265D + 0 $[3.14159265 \times 10^0]$

16.23D - 5 $[16.23 \times 10^{-5}]$

0.0 D0 $[0.0 \times 10^0]$

88 D + 5 $[88 \times 10^5]$

4. 复型常数

复型常数的表示方法是：用括号内两个以逗号隔开的实型常数来表示，前者是实数部分，后者是虚部的系数。

例如：(6.2, 2.48) $[6.2 + 2.48 i]$

(3.0E5, 21.6) $[3.0 \times 10^5 + 21.6 i]$

(-1.E - 2, 2.5E + 3) $[-10^{-2} + 2.5 \times 10^3 i]$

一个复型常数在机器内占用相邻的两个单元。

5. 逻辑型常数

逻辑型常数就是逻辑值：•TRUE• (真)

•FALSE• (假)

逻辑型常数是一个单字，在机内占用一个单元。

6. 文字型常数

文字型常数通常有两种表示方法：

① $nHh_1h_2 \dots h_n$

例如：5HROOT1 [表示 5 个文字符号 ROOT1]

4HSUM = [表示 4 个文字符号 SUM =]

9HREAL □ PART [表示 9 个文字符号

REAL □ PART]

注意，在文字型常数中，空白也算作一个符号。

② $'h_1h_2\cdots h_n'$

例如: 'ROOT2' [表示符号 ROOT2]

'SIGMA' [表示符号 SIGMA]

这第二种表示方法不是标准 FORTRAN 所规定的表示法。

§ 2. 类型的隐、显说明与维数语句

一、符号名 (标识符)

在 FORTRAN 中, 用来给变量、数组等取名字的字符序列叫做符号名或称为标识符, 它是以拉丁字母开头的一串由拉丁字母与阿拉伯数字组成的字符序列。标准 FORTRAN 中规定, 一个符号名至多有 6 个字符; 在 FELIX C—256 机上规定一个符号名不得超过 8 个字符。不过, 一般常规定: 多余字符不起作用, 因此, 写多了也不要紧。例如, 在 FELIX C—256 机上, DIFFERENT 与 DIFFERENCE 被认为是同一个 DIFFEREN。

符号名的例: NUMBER	SUM
A	EPS
I	PI
X1	MATRIX
X2	VECTOR
QDRTIC	CUBE

下面的符号序列不能作为符号名:

3, 2AB (因为不是以拉丁字母开头);

X-1, πA , EPS.L (因为出现拉丁字母和阿拉伯数字以外的符号)。

符号名只起“标识”作用, 可由程序工作者自由选用。在

同一个程序段内，同一个符号名不能表示两个不同的成分。

二、类型的隐式说明与显式说明

由于变量的类型不易直接识别，因此需要进行类型说明好为编译程序提供信息，以便区分并安排工作单元。

在 FORTRAN 中，关于类型说明有两种方法：

1. 隐式说明法

凡以字母 I、J、K、L、M、N 开头的符号名皆为整型，否则为实型。

2. 显式说明法

凡实型和整型符号名不符合隐式说明规则的，必须用“类型语句”说明其类型。

例如：REAL L, NP1

这说明 L 和 NP1 为实型变量。

INTEGER A, B, FUNC

这说明 A 和 B 以及 FUNC 是整型变量。

隐式说明法只适用于实型和整型的情形，对于双精度型、复型、逻辑型的情形必须用类型语句加以说明。例如

DOUBLE PRECISION X, DATA, ER

COMPLEX ALFA, BETA

LOGICAL A, B, C, TEST

这里，第一个是双精度型的类型语句，它说明 X, DATA, ER 是双精度型变量。第二个是复型的类型语句，它说明 ALFA, BETA 是复型变量。第三个是逻辑型的类型语句，它说明 A, B, C, TEST 是逻辑型变量。

注意，类型语句是非执行语句，因此它必须出现在被说明的变量的前面，它只为编译程序提供信息。上述的两种说明方法同样适用于数组、语句函数和函数子程序。

综上所述，类型语句的一般形式是

$$t \quad d_1, d_2, \dots, d_n$$

其中 t —类型符: INTEGER, REAL, LOGICAL,
COMPLEX, DOUBLE PRECISION;

$d_1, d_2, \dots, d_n$ —变量名, 数组名, 数组说明符, 语句
函数名, 函数子程序名。

三、数组及其维数语句

在解决实际问题时除了各种单个变量以外, 还常常会遇到数组。按一定顺序排列的一组数, 称为数组。例如计算 $S = \sum_{i=1}^{10} a_i$, 我们就可以把 $a_1, a_2, \dots, a_{10}$ 作为一个数组, 而如下的一个矩阵

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

也是一个数组。

在 FORTRAN 中, 把一个向量或一个矩阵等叫做数组, 把向量的分量或矩阵元素等叫做数组元素。

对于数组元素, FORTRAN 中是如下表示的。例如上面的一维数组的数组元素表示为

$$A(1), A(2), \dots, A(10)$$

上面的二维数组的数组元素表示为

$$A(1,1), A(2,1), \dots, A(3,3)$$

数组中各元素在存贮器中的存贮位置, 是按数组元素的序号大小排列的。数组元素的存贮序号又叫下标值, 它的计算方法如下表所示:

维数	数组说明符	数组元素	下 标 值	总个数
1	$A(M)$	$A(I)$	I	M
2	$A(M,N)$	$A(I,J)$	$I + M * (J - 1)$	$M * N$
3	$A(M,N,L)$	$A(I,J,K)$	$I + M * (J - 1) + M * N * (K - 1)$	$M * N * L$

由下标值的求法可知：一维数组是连续存放的，二维数组是“按列存放”的，三维数组是“分层按列存放”的。关于二维数组和三维数组的存放现图式如下。

例如，由 $A(3,2)$ 指明的数组，存贮顺序如下图所示：

(1) $A(1,1)$	(4) $A(1,2)$
(2) $A(2,1)$	(5) $A(2,2)$
(3) $A(3,1)$	(6) $A(3,2)$

由 $A(3,3,3)$ 指明的数组，是依第三下标分层的，对于每一层又都是按列存放的，即上面所说的“分层按列存放”。设 A 的元素为 $A(I, J, K)$ ($I, J, K = 1, 2, 3$)，则存贮图式如下：

第一层 ($K = 1$)	(1) $A(1,1,1)$	(4) $A(1,2,1)$	(7) $A(1,3,1)$
	(2) $A(2,1,1)$	(5) $A(2,2,1)$	(8) $A(2,3,1)$
	(3) $A(3,1,1)$	(6) $A(3,2,1)$	(9) $A(3,3,1)$