

高等学校教材

结构力学程序设计

王焕定 主编

王焕定 朱本全

张永山 张金生

孙 锋

合编

高等教育出版社

前 言

本书是根据国家教委高等工科大学力学课程教学指导委员会结构力学课程教学指导小组制订的结构力学课程教学基本要求和有关提高内容,以及我校多年教学实践编写而成的。

全书内容主要结合结构力学中的计算问题,从分析、建立模型、确定算法等开始,按软件工程学思想。介绍了程序设计的全过程。为便于使用和自学,也简述了FORTRAN 77算法语言、结构矩阵分析原理和上机实习的程序设计任务书等。

本书可作为高等工科院校土建、水利、道桥、力学等各专业学生学习结构力学程序设计和上机实习的教材;若补充现有结构力学教材的矩阵位移法内容则本书也可作为结构矩阵分析和程序设计选修课的教材。本书还可作为有关各专业研究生、教师及工程技术人员学习用书。

本书的教学程序全部在微机上调通。均可作相应的实际工程结构分析计算使用。

本书由王焕定主编,并编写第二、三、五章,朱本全编写第四章,张金生编写第一章,孙锋编写第六章,张永山编写附录内容。

本书审阅人为北方交通大学王道堂教授、清华大学匡文起教授和重庆建筑工程学院吴德伦教授,在编写过程中也得到了西安冶金建筑学院刘铮教授等的指教,他们对书稿提出了许多宝贵的意见和建议,我们特此表示衷心的感谢。

限于编者理论水平和教学经验。不足之处敬请读者指正。

编 者

哈尔滨建筑工程学院

1992.6

(京)112号

内 容 提 要

本书以一些较简单的结构计算问题为对象,介绍了程序设计(设计、阅读、修改和调试等)的各方面内容。全书共分六章:FORTRAN 77 算法语言概述,结构力学问题的程序设计,矩阵位移法概述,平面刚架程序的使用及阅读,平面桁架静力分析程序的设计,电算实习任务书。书后还附有 BASIC 源程序和 IBM-PC 机的操作等八个附录。

本书主要特点有:架起了结构计算问题到计算机程序的桥梁;融算法语言、矩阵分析、软件工程与上机操作等于一书,适于自学;始终注意按软件工程学思想组织内容,使初学者养成良好而科学的程序设计习惯。

本书可作为高等工院校土建、水利、道桥、力学等各专业学生学习结构力学程序设计和上机实习的教材,也可作为有关各专业研究生、教师及工程技术人员学习用书。

本书责任编辑 余美茵

高等学校教材

结构力学程序设计

王焕定 主编

王焕定 朱本全 合编
张永山 张金生 孙锋

高等教育出版社出版
新华书店总店科技发行所发行
河北省香河县印刷厂印装

开本 787×1092 1/16 印张 14.5 字数 350 000

1993年4月第1版 1993年4月第1次印刷

印数 0001—3 638

ISBN7-04-004112-X/TB·214

定价 5.65 元

目 录

第一章 FORTRAN 77算法

语言概述	1
§ 1-1 程序结构	1
§ 1-2 说明	2
§ 1-3 语句	5
§ 1-4 函数	12
§ 1-5 子程序	13
§ 1-6 输入/输出语句和格式控制	18

第二章 结构力学中较简单结构的程序设计

§ 2-1 程序设计概述	22
§ 2-2 多跨静定梁的程序设计	28
§ 2-3 三铰拱的程序阅读	32
§ 2-4 连续梁支座弯矩影响线程序	41
§ 2-5 连续梁支座弯矩影响线程序修改	46
§ 2-6 程序调试概述	50
§ 2-7 程序调试举例	57
§ 2-8 程序质量评价	69
习题	70

第三章 矩阵位移法概述

§ 3-1 杆件结构离散化	73
§ 3-2 单元分析	75
§ 3-3 整体分析——直接刚度法	79
§ 3-4 引入支承条件	82
§ 3-5 结构分析中几种常用的单元刚度矩阵	83
习题	87

第四章 平面刚架程序的使用及阅读

§ 4-1 平面刚架教学参考源程序	89
§ 4-2 教学参考源程序的使用说明	89

§ 4-3 教学参考源程序的阅读	106
§ 4-4 平面刚架静力计算程序的拓广应用	112
§ 4-5 扩充教学程序功能的一些说明	117
习题	121

第五章 平面桁架静力分析程序的设计

§ 5-1 程序的总体设计——模块的划分	126
§ 5-2 模块接口设计——数据交换问题	128
§ 5-3 模块 JCZG和DYZL 的求精展开	131
§ 5-4 平面桁架静力分析源程序	134
§ 5-5 平面桁架静力分析程序的使用说明及算例	139

第六章 电算实习任务书

§ 6-1 平面刚架静力计算程序修改的任务书	145
§ 6-2 交叉梁程序任务书	149
§ 6-3 空间桁架程序任务书	152

附录A 常见荷载类型

附录B 连续梁静力计算程序	159
---------------	-----

附录C BASIC教学参考程序	163
-----------------	-----

附录D 教学程序库清单	170
-------------	-----

附录E 第一类(分支)稳定问题教学程序	171
---------------------	-----

附录F 常用标准算法	192
------------	-----

附录G IBM-PC机的操作说明	196
------------------	-----

附录H QUICK-BASIC 教学参考程序	203
------------------------	-----

附录I 部分习题答案	222
------------	-----

参考文献	224
------	-----

第一章 FORTRAN 77 算法语言概述

FORTRAN 77 是一种主要用于科学技术计算的计算机高级语言，其内容十分丰富。本章旨在对编制一般计算程序所经常遇到的和本书中出现的语法内容作一简单介绍，为已学过该语言的读者提供有关语法规定的简要手册。对于未接触过该语言的读者，建议学习有较多例子的 FORTRAN 77 教材。

§ 1-1 程序结构

一、FORTRAN 77 程序结构与组成

FORTRAN 77 是一种块结构语言。用它编写的程序可以分为若干个形式上独立的程序单位。其中必有且只有一个程序单位是主控单位，也可称为主程序。其余均为辅程序，它又可分成子例行、函数和数据块辅程序，本书分别简称为子程序、函数和数据块。一个程序也可以只有主程序而没有辅程序，当两者都有时一般习惯是将主程序放在最前面，其它辅程序以任意次序放在主程序之后。但这也并非必须如此，§ 2-7 中将主程序放在最后也是允许的。

每个程序单位均由语句序列组成，这些语句可分为可执行语句和非执行语句两类。非执行语句用来描述数据的结构和特征、编辑信息、语句函数及程序单位的分类等。可执行语句用来指明要做什么操作，如循环、转移和输入、输出等。一些常用语句的分类情况见表1-1。

一个程序单位中各类语句的位置有一定的规定，说明语句应在可执行语句之前。程序单位中语句顺序如表1-2。

表1-2中水平线上线下的顺序不能改变，必须由上向下。竖线左右两边的语句顺序允许任意穿插。

二、FORTRAN 77 源程序的书写格式

FORTRAN 77 源程序的书写有严格的要求，必须按规定书写。源程序每行有 80 列，可分为四个区：标号区、续行区、语句区和标识区。一般标识区除注解行等外，在程序设计时不用。其它三个区说明如下：

表 1-1 FORTRAN 77 语句简表

可执行语句	非执行语句
赋值语句	COMMON 语句
CALL 语句	DIMENSION 语句
CONTINUE 语句	类型说明语句
DO 语句	EXTERNAL 语句
ELSE 语句	FORMAT 语句
ELSE IF 语句	FUNCTION 语句
END 语句	IMPLICIT 语句
END IF 语句	INTRINSIC 语句
GO TO 语句	PARAMETER 语句
IF 语句	PROGRAM 语句
PAUSE 语句	语句函数
PRINT 语句	SUBROUTINE 语句
READ 语句	
RETURN 语句	
STOP 语句	
WRITE 语句	
OPEN 语句	

表 1-2 FORTRAN 77 语句顺序表

注解行	PROGRAM 语句、FUNCTION 语句、 SUBROUTINE 语句		
	FORMAT 语句	PARAMETER 语句	IMPLICIT 语句
			其它说明语句
		语句函数定义语句	
		可执行语句	
END 语句			

1. 标号区

占每行的第一至第五列，用于书写语句标号。在任何被引用语句的前五列中，都必须有语句标号，它为 1 到 99999 之间的任一无符号整数。标号的数值本身无意义，它只起标识作用，并不代表执行的顺序。在同一程序单位中不能有重复的语句标号，而在不同程序单位里可以用相同的标号。

2. 续行区

仅占第六列，用于填写续行标志。若某行的前五列为空白，而第六列是一个非 0 非空格的任意字符，则该行是上一行的继续行，而该字符称续行标志。

3. 语句区

占第七至七十二列，用于书写语句。每一行的语句区只能书写一条语句，写不下时在下一行用续行标志继续写，最多允许有十九个续行。

当一行的第一列为字符 C（大、小写均可）或 * 时，此行称为注解行。它的 80 列均可利用，且可在任何处写注解内容。在程序的任何位置均可用注解行进行注释。

§ 1-2 说 明

说明语句的作用是向编译系统提供程序中各量的类型和存储分配信息等。在程序单位里，说明语句必须放在第一个可执行语句的前面（与子程序有关的一些说明见 § 1-5）。

一、常数、变量、数组及其类型

1. 常数

在程序执行过程中不改变其值的量称为常数。它分为整型、实型、双精度型和字符型等类型（另二类为长整型和复型，本书只介绍上述四类）。

（1）整型常数 不带小数点的常数即整数。如 2，+3，-21 等。而 3.，-5.0, 5.0 等都不是整型常数。在微机上整型常数的合法范围一般（指 16 位机）为 -32768 至 32767。

（2）实型常数 带有小数点或者指数的常数^①。如 3.0，2.，30.2，-31.23E-3（等于 -31.23×10^{-3} ）等。

（3）双精度型常数 有效数字更多的实常数。它比实常数更精确，也占用更多的字

① 实型常数小数点后的零可写或不写。

节。如0.3141592653589D 1 (也即3.141592653589)。

(4) 字符型常数 用单引号括起来的由字符组成的一串符号。如'ABC', 'FORCE', '369', 'A1 B 2' 等均为字符型常数。字符常数的长度由字符个数确定, 中间的空格符也作字符计数。如'External Force' 长度为14。

2. 变量

在程序中其值可变的量称为变量。变量具有一定的名字和类型。

变量名必须是字母开头的、一个由字母(或数字或数字与字母)组成的字符串。机器(指系统软件)能够识别的字符串中字符个数, 称有效长度, 一般机器有效长度为六位。如ABCDE 1 和ABCDE 2 机器认为是不同的变量。对ABCDEF 1 和ABCDEF 2 一些机器可能认为此两变量名是相同的。

变量与常数一样也有六类, 程序中每一变量都必须有确定的类型。FORTRAN 语言对类型存在隐含约定: 凡以 I、J、K、L、M、N (含小写) 为首的变量均为整型。除此之外的变量都是实型的。这种约定简称 I-N 规则。如果要改变这种规则, 必须在程序中用 IMPLICIT 和类型说明语句进行说明。

3. 数组

数组是用一个名字表示的一组顺序排列的同类型变量。在程序里可用类型说明语句、维数语句或公用语句对数组类型和规模(也称维数)等进行定义, 同一程序单位中不允许有重名的数组定义。用以确定变量在数组中位置的整数称为下标, 下标所指明的变量称为下标变量。如A(4)表示一个名为A的数组的第4个变量(也称为元素), 4为下标, A(4)为下标变量。为了与下标变量加以区别, 前述变量一般称简单变量。

二、类型说明语句

格式: $t \quad V_1, V_2, \dots, V_n$

其中, t 可以是 INTEGER、REAL 和 CHARACTER 等类型说明关键词之一, 分别表示其后 V_i 是整型、实型和字符型等。 V_i 可以是变量名、数组名、函数名或数组说明符(函数名见 § 1-5-1、数组名和数组说明符见本节五)。

类型说明语句优先于隐含说明语句和隐含约定, 即当某变量用类型说明语句说明后, 隐含规则和其前的隐含说明语句均不再起作用。

当 V_i 代表数组时可有两种表示法。若该数组未经维数语句说明, V_i 应是数组说明符, 否则应该用数组名。

例 INTEGER Y1, DP, H, A2
 REAL IMIN, L
 CHARACTER *5 A, B
 CHARACTER C*3, D*4

此例中, Y1, DP, H, A2 均为整型变量, IMIN, L 为实型变量, 它们均改变了 I-N 规则。变量 A, B 是长度为 5 的字符变量, 而字符变量 C、D 的长度分别为 3 和 4。

三、隐含说明语句 (IMPLICIT 语句)

格式: $IMPLICIT \quad t \quad (\text{字母}, \dots)$

其中，t 的含义同本节二，字母，...可以是“字母-字母”或26个字母的某个子集。

隐含说明的作用是改变 I-N 规则确定的类型，自行定义一些新的隐含规则（仅在该程序单位中有效）。如为了将 L、M 和 N 开头的变量定义作实型，而以 A、B 开头的变量定义作整型，可用如下说明语句：

```
或      IMPLICIT INTEGER (A, B), REAL (L-N)
        IMPLICIT REAL (L, M, N)
        IMPLICIT INTEGER (A, B)
```

等等。

四、参数说明语句 (PARAMETER 语句)

格式： PARAMETER (C₁=e₁, C₂=e₂, ...)

其中，C_i 是常量名，e_i 是常数表达式。

参数说明语句的作用是为某常数命名。当程序中用到该常数时均可用常量名代替。

例 PARAMETER (PAI=3.141592)

表示常数 3.141592 的名为 PAI，程序中凡用到 3.141592 处均可用 PAI 代替。

常量名的命名与变量相同，也有类型区别，类型的确定也与变量相同。在常数表达式 e_i 中可以出现其它常量名，但必须是在同一程序单位且已定义的常量名。

```
例      REAL MT
        PARAMETER (MT=0.001, I=100)
        PARAMETER (N1=2*I, A=3.976*MT)
```

五、维数语句 (DIMENSION 语句)

DIMENSION 语句用于说明数组的体积、维数等。

格式： DIMENSION V₁(i₁), V₂(i₂), ..., V_n(i_n)

其中，V_i(i_i) 称为数组说明符，V_i 称为数组名。i_i 称为下标说明符，它由若干个数组 (A:B) 组成，数组个数称维数，最多可达七维。数组 A:B 中 A 称下界，B 称上界，上、下界可为整数、整型变量或整型表达式，其值可以大于、等于或小于零^①，但下界必须小于等于上界。当下界为 1 时可以省略。

例 DIMENSION A(-5:-1, -10:2), CE(2, 2), DA(5)

表示 A 和 CE 是二维数组，DA 是一维数组。A (-5:-1, -10:2) 等称为数组说明符，A、CE 等称为数组名。因 A 的元素共 65 个，故称数组 A 的体积为 65。

当上下界表达式中出现变量时，此数组称为可调数组（参见 § 1-5 的说明）。

FORTRAN 77 中数组元素在计算机内是按列存储的。例如：

```
DIMENSION B(2, 3)
```

^① 一些微机 FORTRAN 编译系统不允许小于等于零。

为二维数组共六个元素，用它可代表数学中的一个 2×3 的矩阵。数组 B 的六个元素在计算机中的排列的顺序是：

$B(1, 1), B(2, 1), B(1, 2), B(2, 2), B(1, 3), B(2, 3)$

正因如此，当子程序中可调数组在调用时的体积小于实元所说明体积时，可调数组由于实际结合（见 § 1-4）将按实元的列顺序排列。例如：

```

        DIMENSION A(10,3)
        -----
        CALL EXAM(A,3,...)
        -----
        END

c
        SUBROUTINE EXAM(B,N,...)
        DIMENSION B(N,N),...
        -----
        END
    
```

若数组 A 在 CALL 语句前为零，则经 CALL EXAM (A, 3, ...) 调用后，A 第一列前 9 个元素为子程序 EXAM 中数组 B 的全部元素，A 中其余元素仍全为零。初学者必须十分注意，以免产生错误。上述说明中至此尚未介绍的内容，可参见 § 1-5。

同一程序单位中，切记数组名和变量名不能相重，数组说明符也不允许重复出现。但数组名可在同一程序单位说明部分出现多次。例如：

```

        DIMENSION MOMENT(9)
        REAL MOMENT
    
```

是正确的。

§ 1-3 语 句

一、算术、关系和逻辑表达式

1. 算术表达式

算术表达式是由操作数（如变量、常量、下标变量等）、算术运算符、括号等组成的式子。算术运算符有 +、-、*、/、**（** 代表乘方）。运算的优先次序为：（1）括号，（2）函数，（3）**，（4）* 和 /，（5）+ 和 -。算术表达式的例子见表 1-3。

算术表达式的值是一个数，其类型由操作数类型确定。当操作数均为整型时，算术表达式的值为整型。当操作数均为实型时，表达式的值就为实型数。若操作数既有整型又有实型，算术表达式将为实型。

表 1-3 算术表达式算例

算 术 式	FORTRAN 77 的算术表达式
$a + (-b)$	$A + (-B)$
$(a \cdot b)^2$	$(A*B)**2$ 或 $(A**F)*(A*B)$
$\frac{a}{b \cdot c}$	$A/B/C$ 或 $A/(B*C)$
$a(b+c)$	$A*(B+C)$
$\sin 3x$	$SIN(3*X)$
$a \cdot b \cdot (-c)$	$A*B*(-C)$ 或 $-A*B*C$

注意：两整数相除时，结果为商的整数部分。例如：5/2 结果为 2，4/5 结果为 0。两

实数相除时由于商也是实数，所以5.0/2.结果为2.5。

2. 关系表达式

关系表达式是由关系运算符、括号、算术表达式组成的式子。它比较两个算术表达式的值并产生逻辑真 (TRUE) 或假 (FALSE)。关系运算符如表1-4所示。

例如:

表 1-4 关系运算符

关系运算符	含 义
.LT.	< (小于)
.LE.	≤ (小于等于)
.EQ.	= (等于)
.GT.	> (大于)
.GE.	≥ (大于等于)
.NE.	≠ (不等于)

(X+Y*A).LT.(C+D)
(SIN(Y)+COS(X)).GE.A

都是合法的关系表达式。

3. 逻辑表达式

逻辑表达式是由逻辑运算符、括号、关系表达式等组成的式子。逻辑运算符是: .NOT., .AND., .OR.。它们分别表示逻辑非、逻辑乘和逻辑和 (也称为“非”、“与”和“或”。)

当多种运算符在逻辑表达式中出现时, 优先次序为:

**→*, /→+, -→.LT., .LE., .EQ., .NE., .GT., .GE., →.NOT.→
.AND.→.OR.

其中逗号分隔的运算符的优先级别是相同的。

二、算术赋值语句

格式: $V=e$

其中, V是变量名, =称赋值号, e是算术表达式。

变量名可以是整型、实型等的简单变量, 也可是下标变量。赋值语句的作用是把算术表达式e的值计算出来并赋给变量V, 它与数学中的等式是根本不同的。若V与e的类型不同, 赋值语句按表1-5把e的类型转换后再赋给V。

表 1-5 类型转换

e 类型 \ V 类型	整 型	实 型
整 型	照 赋	取整 (去掉小数)
实 型	转 实 型	照 赋

三、GO TO 语句

GO TO 语句用来改变程序按行自上而下执行的顺序, 实现跳过某段程序、返回已执行过的语句重又执行, 或根据某种条件来确定执行路线的目的。FORTRAN 77 里有三种GO TO 语句:

1. 无条件 GO TO 语句

格式: $GO TO n$

其中, n是语句标号。

当此语句被执行时, 程序将无条件地转移到本程序单位标号为n的语句上继续执行。为了实现程序的结构化 (参见 § 2-1), 应避免向前转移。在能用别的语句代替实现相同转移时, 向后转移最好也不用。

2. 计算 GO TO 语句

格式: $\boxed{\text{GO TO}(K_1, K_2, \dots, K_n), I}$

其中, K_i 是语句标号, I 是一个整型变量或整型表达式, 通常它应满足条件 $1 \leq I \leq n$, 而且在执行到这条语句时, 它的值应已确定。

计算 GO TO 语句的作用是, 按 I 的值 i 转向标号为 K_i 的语句。如:

```
.....  
M=3  
GOTO (4, 21, 4, 3, 9), M  
.....  
.
```

当执行到计算 GO TO 语句时, 程序转移到第三个语句标号所对应的语句, 即标号为 4 的语句继续执行。

3. 赋值 GO TO 语句

赋值 GO TO 语句必须与标号赋值语句配合使用。

标号赋值语句

格式: $\boxed{\text{ASSIGN } M \text{ TO } K}$

赋值 GO TO 语句

格式: $\boxed{\text{GO TO } K, (L_1, L_2, \dots, L_N)}$

其中, K 是整型变量名。M 是语句标号, 它必须是本程序单位中语句标号 L_1, L_2, \dots, L_N 之中的一个。

执行标号赋值语句, 就将语句标号 M 赋给了 K 。而当执行赋值 GO TO 语句时, 就把控制转移到 K 所标识的语句。

四、IF 语句

IF 语句常称作条件语句。利用它可实现有条件的执行某语句或语句段, 从而实现程序的多分支。IF 语句包括算术 IF 语句、逻辑 IF 语句和分块条件语句。

1. 算术 IF 语句

格式: $\boxed{\text{IF} (\text{算术表达式}) K_1, K_2, K_3}$

其中, K_1, K_2, K_3 为所在程序单位中可执行语句的标号。执行此语句时, 先计算表达式的值, 按其值为负、零和正分别转向 K_1, K_2, K_3 所标识的语句。

2. 逻辑 IF 语句

格式: $\boxed{\text{IF} (\text{逻辑表达式}) \text{可执行语句}}$

其中, 可执行语句为表 1-1 中除以下语句之外的任一语句:

DO 语句、块 IF 语句、ELSE IF 语句、ELSE 语句、END IF 语句、END 语句和另一逻辑 IF 语句。

执行此语句, 先计算括号中逻辑表达式的值, 若为“真”, 则执行表达式后的可执行语句并继续, 否则执行此语句的下一条语句。

3. 分块条件语句

分块条件语句有以下四类语句：

块 IF 语句：格式为 IF (逻辑表达式) THEN

ELSE IF 语句：格式为 ELSE IF (逻辑表达式) THEN

ELSE 语句

END IF 语句

用它们可构成单边、双边和多重选择结构。

(1) 单边选择结构

格式：

```
IF (逻辑表达式) THEN
    语句序列
END IF
```

执行此语句时，先计算逻辑表达式的值。其值为“真”，按顺序执行语句序列。否则执行 END IF 后的语句。

格式中的语句序列（或称为程序段）一般称为 IF 块。在 IF 块中可包含另一个单边选择结构，从而实现嵌套结构。例如：

```
IF (X.GE.0.0) THEN
    IF (Y.GE.0.0) THEN
        Z=SQRT(X)+SQRT(Y)
    END IF
END IF
```

(2) 双边选择结构

格式：

```
IF(逻辑表达式) THEN
    程序段 A
ELSE
    程序段 B
END IF
```

执行本语句也是先计算逻辑表达式的值，其值为“真”，执行程序段 A，否则执行程序段 B。当 A 或 B 程序段执行完后，均从 END IF 后面的语句继续执行。程序段 B 称为 ELSE 块。

(3) 多重选择结构

格式：

```
IF (逻辑表达式 E) THEN
    程序段 A
ELSE IF (逻辑表达式 F) THEN
    程序段 B
ELSE IF (逻辑表达式 G) THEN
    程序段 C
ELSE
    程序段 D
END IF
```

执行本语句时，首先求表达式E的值。若其值为“真”，则执行程序段A，然后执行END IF的下一语句。否则，就计算表达式F的值，由其值的“真”或“假”分别执行程序段B或去计算表达式G，…。程序段A称为IF块，程序段B、C称为ELSE IF块，程序段D称为ELSE块。无论哪种块中均可嵌套另一个选择结构。但必须注意：IF…THEN语句必须与END IF语句配对出现，为便于检查，读者应养成使程序嵌套层次分明的习惯。此外，还需切记无论何种选择结构都不准将控制从各种块外转到块内。例如下面的程序片断是错误的：

```

10      IF (N.EQ.M) THEN
          X=X+0.23
          Y=SIN(X)
        END IF
        .
        .
        .
        GOTO 10

```

五、循环语句 (DO 语句)

在程序中若需要对某一部分执行多次，可采用DO语句来完成。DO语句又称为循环语句，它可分单重循环和多重循环两类。

1. 单重循环

格式：

DO n I=M1, M2, M3 可执行语句序列 n CONTINUE	}	循环体
--------------------------------------------	---	-----

其中，n是循环终端语句的语句标号，I是循环控制变量(简称循环变量)，M1、M2和M3分别是循环的初值、终值和步长。循环步长为1时，M3可省略。

循环变量I可以是整型或实型变量，M1、M2和M3可以是整型或实型常数、整型或实型表达式，其值可正、可负，M1和M2可以为零，M3不能为零(如§1-2中注解所指出，对数组循环时必须注意系统软件允许下标是否为负)。一般情况下I、M1、M2和M3取整型量较好。

终端语句必须是可执行语句，但不能是下述语句：

DO、GO TO、IF、RETURN、STOP、PAUSE 语句

终端语句最好为CONTINUE语句，

格式：

CONTINUE

它虽是一个可执行语句，但并不产生任何操作，只起执行继续的作用。

循环语句的执行过程如下：

(1) 计算M1、M2和M3的表达式值。

(2) 把M1的值赋给I。

(3) 确定循环次数n1

$$n1 = \text{MAX} (\text{INT}((M2 - M1 + M3) / M3), 0)$$

(MAX和INT是FORTRAN内部函数，见§1-4)。

- (4) 检查循环次数，若 $n1$ 为零则执行终端语句后面的语句，否则执行循环体。
- (5) 当执行到终端语句时， $I+M3$ 赋给 I 。
- (6) 循环次数 $n1$ 减 1，然后从 (4) 继续执行。

```

例      X=0.0
        Y=0.0
        DO 10 I=1,100
           X=X+1.0/I
           Y=Y+1.0/(I*I)
10      CONTINUE

```

对循环语句有如下要求：

- (1) 循环变量不允许出现在赋值号左边。
- (2) 循环初值、终值和步长在循环体内也不许改变。
- (3) 可用 GO TO 语句从循环体内转出，而绝不允许从循环体外转入循环体，也即“只出不入”。

2. 多重循环

若单重循环的循环体内又有循环语句，则构成多重循环，又称为循环的嵌套。

若用开口方框表示循环，则多重循环示意图如图 1-1 所示。

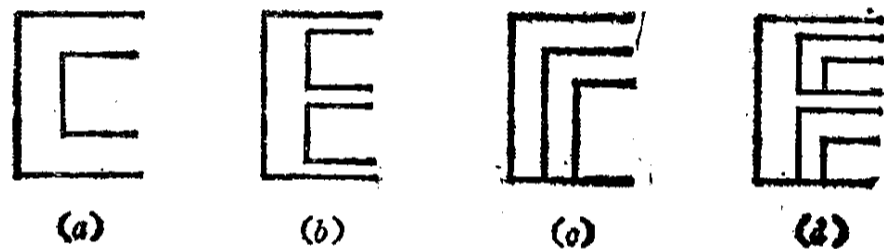


图 1-1 多重循环示意图

图 1-1 a 表示嵌套的二重循环，图 1-1 b 表示一个循环内有两个内层循环，图 1-1 c 表示一个三重循环且公用一个终端语句，图 1-1 d 表示外层循环里有两个二重循环，第一个二重循环是公用一个终端语句的，第二个二重循环与外循环公用一个终端语句。例如：

```

        DO 10 I=1,M
           DO 10 J=1,N
              X(J,I)=0.0
10      CONTINUE

```

就是一个公用终端语句的二重循环。

对多重循环的要求如下：

- (1) 有嵌套关系的循环变量不允许同名，反之可以同名。
- (2) 内循环必须完整地嵌套在外层循环里，不允许交叉。同时，循环与 IF 语句中的各种块也不允许交叉。
- (3) 同单重循环一样，不管哪个层次的循环都必须遵循“只出不入”的规则。各种允许的转移如图 1-2 所示，各种不允许的转移如图 1-3 所示。

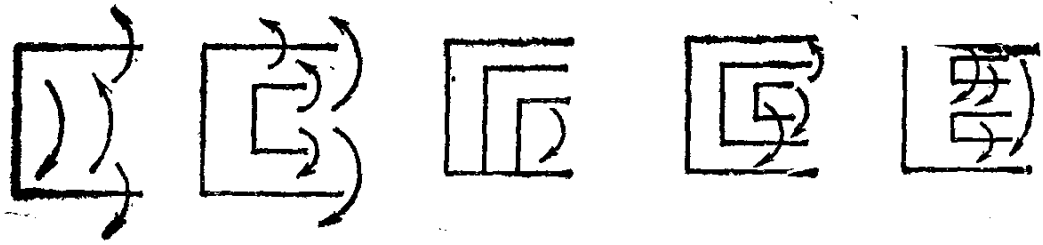


图 1-2 循环的允许转移示意图

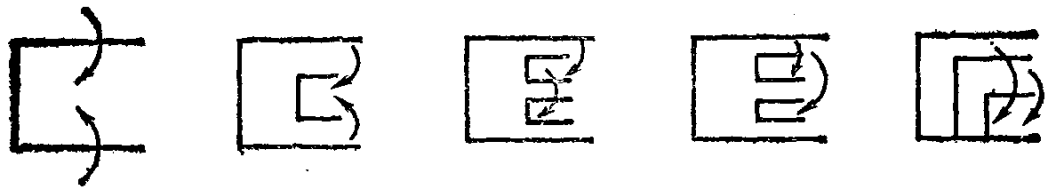


图 1-3 循环的不允许转移示意图

以下程序片断由于违反第(2)条要求, 故是错误的:

```

DO 10 I=1, 10
  A=5.
  IF (I.GT.5) THEN
    B=I+3.
    C=A*I
    X=B/C
10  CONTINUE
    Y=X+A
  END IF

```

六、停语句和结束语句

1. 暂停语句 (PAUSE 语句)

格式: PAUSE n

中, n 是 1 至 5 位的数字串或字符常数, 也可略去。

执行 PAUSE 语句导致程序执行的暂停, 若有 n 则在屏幕上显示它。程序暂停执行后可通过键盘打入命令 (因随系统软件而变, 故不介绍) 使在暂停点继续运行。

2. 停语句 (STOP 语句)

格式: STOP n

它与 PAUSE 语句类似, 区别仅在 STOP 后是不能在停止点恢复程序运行的, 故不再解释。

3. 结束语句 (END 语句)

格式: END

它是一个可执行语句, 其作用主要是标志程序单位的结束, 每个程序单位的最后一个语句必须是它, 程序单位中也只能有一个 END。此外, END 还兼具 STOP 和返回语句 RETURN 的作用, 因此在主程序和辅程序的 END 前可省略 STOP 和 RETURN (关于辅程序见 § 1-5)。

§ 1-4 函 数

函数由一个语句或一组语句组成，用来定义某个函数的计算方法。它分为内部函数、语句函数和外部函数（外部函数将在 § 1-5 中介绍）。

一、内部函数

凡是由计算机系统软件提供、用户可直接使用的函数称内部函数。调用一个内部函数会得到一个函数值，所以它与变量、常数一样可直接出现于表达式中。在调用时，填写在内部函数括号内的自变量（实元）的个数、类型和顺序，应完全与规定的哑元的个数、类型和顺序一致。实元也可以是表达式，但其类型应与规定相符。

使用内部函数时，还应遵守通常情况下数学方面的基本规定。

FORTRAN 77 提供了 81 种内部函数，根据本章开始说明的取材原则，仅列一部分于表 1-6。

表 1-6 FORTRAN 77 常用内部函数简表

功 能	属 名	专用名	参 数 个 数	类 型		功 能	属 名	专用名	参 数 个 数	类 型	
				P	F					P	F
转整型	INT	INT	1	R	I	求余	MOD	MOD	2	I	I
		IFIX		R	I	平方根	SQRT	SQRT		1	R
转实型	REAL	REAL	1	I	R	指 数	EXP	EXP	1	R	R
		FLOAT		I	R						
截去小数	AINT	AINT	1	R	R	自然对数	LOG	ALOG	1	R	R
					常用对数	LOG 10	ALOG 10				
舍入	ANINT NINT	ANINT	1	R	R	三角函数	SIN COS TAN	SIN COS TAN	1	R	R
		NINT		R	I						
取绝对值	ABS	IABS ABS	1	I R	I R						
符号传送	SIGN	ISIGN	2	I	I	反三角函数	ASIN ACOS ATAN ATAN2	ASIN ACOS ATAN ATAN2	1	R	R
		SIGN		R	R						
选最大值	MAX	MAX0 AMAX1	≥2	I R	I R				2		
选最小值	MIN	MIN0	≥2	I	I	双曲函数	SINH COSH TANH	SINH COSH TANH	1	R	R
		AMIN1		R	R						

说明：P 为参数(PARAMETER)，F 为函数(FUNCTION)

I 为整型(INTEGER)，R 为实型(REAL)

在 FORTRAN 77 中，函数专用名的自变量类型有严格要求，必须按规定给出自变量的类型，而在使用函数属名时，只要保证自变量有意义，对类型没有严格要求；因此，函数的属名实际上具有通用的性质。

SIGN(X₁, X₂) 使 X₁ 与 X₂ 的符号相同。

MAX(X₁, X₂, ...) 在 X₁, X₂, ... 中选最大值。

MOD(X₁, X₂) 取 X₁/X₂ 的余数。

ATAN 2(X₁, X₂) 求 X₁/X₂ 的反正切。

三角函数中的角度均为弧度。

二、语句函数

语句函数是用户编写的用一个语句定义的函数。

格式: $F(a_1, a_2, \dots, a_n) = e$

其中, F 是函数名 (其取名规则与变量相同, 其类型说明也与变量同), a_i 是哑元 (也称为形式参数), 哑元的集合称为哑元表, e 是含哑元的表达式。

哑元是语句函数定义句赋值号右部表达式 e 中所用到的虚拟自变量, 当调用语句函数时它被某些具体的数值或变量所取代。这些代替哑元的量称实元。用实元代替哑元的过程称为哑实结合。哑元不能是常数、表达式和数组名。

语句函数定义句的位置在所有说明的后面, 在第一个执行语句之前。语句函数仅在定义句所在程序单位中有效, 它的引用与内部函数相同。

定义语句函数时, 表达式中不能出现自身函数名, 也不能间接地调用自身函数, 即不能回归调用。但表达式中可以出现已定义且不包含语句函数自身调用的其它语句函数。

例: 设 $x = 0.12$ $y = 0.32$, 试用程序计算

$$Q = \sin y * \sqrt{\cos^2 x + \sin^2 y}$$

计算 Q 的程序段如下:

```
C      定义语句函数
      F(X, Y) = X * X + Y * Y
      SQ(X, Y) = Y * SQRT(F(X, Y))
C      计算并输出
      A = COS(0.12)
      B = SIN(0.32)
      WRITE(*, *) SQ(A, B)
      END
```

§ 1-5 子 程 序

在 § 1-1 中已介绍, 一个程序必有且只有一个主程序, 此外可有若干个函数与子程序等独立的程序单位。这些程序单位都可以单独进行编译并生成目标代码, 最后将所有的程序单位的目标代码进行连接, 即可生成一个可执行程序 (或称可执行文件)。外部函数 (本节中简称函数) 和子程序在结构上与主程序是类似的, 其区别为

1. 主程序段头语句为 PROGRAM 程序名, 它也可以省略。函数段头语句:

格式: $\text{类型 FUNCTION 函数名 (哑元表)}$

子程序段头语句

格式: $\text{SUBROUTINE 子程序名 (哑元表)}$

它们是不能省略的, 但可以没有哑元表, 具体规定见本节一、二。

2. 除段头语句外, 主程序中不能出现 RETURN 语句, 函数与子程序中不能有文件说明语句 (即 FILE 语句, 本书不介绍)。

3. 主程序自身能构成一个可执行程序, 函数或子程序自身不能构成可执行程序。