



新世纪土木工程系列教材

结构力学程序 设计及应用

王焕定 张永山 编著



高等教育出版社

新世纪土木工程系列教材

结构力学程序设计及应用

王焕定 张永山 编著

高等教育出版社

内容提要

本书为新世纪土木工程系列教材之一。全书由浅入深地介绍了结构力学计算问题的程序设计,包括程序设计、阅读、修改和调试等各方面内容,并在书末提供了可供教学和工程应用的光盘。全书共分6章:FORTRAN 90 语言概述(含 FORTRAN 90 精选子集的 elf90 编译器介绍与相关语法)、结构力学问题的程序设计、矩阵位移法、桁架静力分析程序设计、杆系结构静力分析计算程序的使用、结构动力分析程序的使用。

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、结构矩阵分析、软件工程于一书,便于自学;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;所有程序均采用 FORTRAN90 语言标准语句编写,易于修改和移植;本书光盘既可供配合教学内容应用,也可供实际工程结构分析使用;既可解决静力分析问题,也可解决一些动力计算问题。

本书可作为高等工科院校土建、水利和力学类等各专业学生学习结构力学程序设计及应用的教材,也可供有关专业研究生、教师及工程技术人员参考。

图书在版编目(CIP)数据

结构力学程序设计及应用/王焕定,张永山编著. —北京:高等教育出版社,2001

ISBN 7-04-009386-3

I. 结… II. ①王…②张… III. 结构力学—程序设计
IV. 0342

中国版本图书馆 CIP 数据核字(2001)第 031237 号

结构力学程序设计及应用

王焕定 张永山 编著

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

排 版 高等教育出版社照排中心

印 刷 中国农业出版社印刷厂

开 本 787×1092 1/16

版 次 2001 年 7 月第 1 版

印 张 15

印 次 2001 年 7 月第 1 次印刷

字 数 360 000

定 价 21.10 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

教育部高等教育出版社土建类系列教材

编辑委员会委员

主任委员：沈蒲生（湖南大学）
副主任委员：白国良（西安建筑科技大学）
 邹超英（哈尔滨工业大学）
 强士中（西南交通大学）
委 员：王 健 叶志明 江见鲸
 关宝树 刘 明 张印阁
 张家良 朱彦鹏 吴胜兴
 杨和礼 尚守平 黄醒春
 梁兴文 焦兆平 廖红建

(以上排名先后按姓氏笔画)

出版者的话

新世纪土木工程系列教材是我社组织编写出版的“大土木”范畴的专业系列教材。1998年教育部颁布了新修订的《普通高等学校本科专业目录和专业介绍》，新专业目录中土建类土木工程专业覆盖了原来建筑工程和交通土建工程等8个专业。1999年各高校已按新的专业目录招生。开设土木工程专业的各院校把近年来在教育思想与教学观念、教学内容与课程体系、教学方法与教学手段等方面取得的改革成果固化到教学计划和人才培养过程中，设计了从教学思想到教学模式等一系列教学改革方案。大家在教学实践中体会到：专业、课程教学改革必然引起相应的教材改革。我社从1999年开始进行土木工程专业系列教材的策划工作，并于2000年成立了“教育部高等教育出版社土建类系列教材编辑委员会”。

我们编辑出版土木工程系列教材的指导思想是：

1. 紧密结合人才培养模式改革，根据拓宽专业基础、提高综合素质、增强创新能力的要求，调整学生的知识结构。

2. 从各院校调整土建类各专业教学计划出发，加强基础课程到专业课程的有机沟通，用系统的观点和方法建立新的课程体系结构，包括对课程的整合与集成，组织和建设专业核心课程，成套成系列地推出土木工程系列教材。

3. 各门课程教材要具有与本门学科发展相适应的学科水平，以科技进步和社会发展的最新成果充实、更新教材内容，贯彻理论联系实际的原则。

4. 要正确处理继承、借鉴和创新的关系，不能简单以传统和现代划线，决定取舍，而应根据教学要求进行取舍。继承、借鉴历史和国外的经验，注意研究结合我国的现实情况，择善而从，消化创新。

5. 随着高新技术、特别是数字化和网络化技术的发展，在土木工程系列教材建设中，要充分考虑文字教材与音像、电子、网络教材的综合结合，发挥综合媒体在教学中的优势，提高教学效率。在开发研制教学软件的同时，要注意使文字教材与先进的软件接轨，明确不同形式教材之间的关系是相辅相成、相互补充的。

6. 坚持质量第一。图书是特殊的商品，教材是特殊的图书。教材质量的优劣直接影响教学质量和教学秩序，最终影响学校人才培养的质量。教材不仅具有传播知识服务教育、积累文化的功能，也是沟通作者、编辑、读者的桥梁，一定程度上还代表着国家学术文化或学校教学、科研水平。因此，遴选作者、审定教材、贯彻国家标准和规范等方面需严格把关。

为了实现本套教材的指导思想，我们组建了由有丰富的教学经验、有较高的学术水平和学术声望的教师组成的编委会，由编委会研究提出土木工程系列教材的选题及其基本内容与编审原则，并推荐作者。

我们出版本系列教材，旨在为新世纪的土木工程专业学生提供一套经过整合优化的比较系

统的专业系列教材,以期为我国的土木工程专业教材建设贡献自己的一份力量。

本系列教材的编写大纲和初稿都经过了编委会的审阅,以求教材质量更臻完善。如有疏漏之处,请读者批评指正!

2001年3月

前 言

本书为新世纪土木工程系列教材之一。本书的编写提纲经土建类系列教材编辑委员会认可,以教育部颁发的结构力学课程教学基本要求为基础,考虑了本教材作为矩阵位移法和程序设计相结合单独设课需要,根据多年教学实践的经验编写而成。

全书首先从矩阵位移法和程序设计单独设课出发,结合课程需要简述了 FORTRAN 90 算法语言(含 FORTRAN 90 精选子集的 elf90 编译器介绍与相关语法)。接着,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,按照软件工程学思想,介绍了程序设计的全过程。在此基础上,安排了结构矩阵分析的相关内容及用矩阵位移法知识解决结构计算问题的程序设计内容。最后,介绍了本书光盘杆系结构静力分析程序和结构动力分析程序的使用。这些应用程序虽然不是结构辅助设计软件,但解题规模完全可供工程结构分析计算所用。

本书可作为高等工科院校土建、水利和力学类等各专业学生学习结构力学程序设计及应用的教材,也可作为有关各专业研究生、教师及工程技术人员的参考书。

本书凡属完整的教学程序(包括程序段),在本书光盘上均有源程序,书中例题以及部分习题的答案,也均在本书光盘上。

本书由王焕定、张永山编著,其中王焕定编写第 3、5 章、§1—8 节和本书光盘内容编写制作,张永山编写 1、2、4 章,孙峰编写第 6 章。

本书由东南大学单建教授审阅,对书稿提出了许多宝贵的意见和建议。此外,在编写过程中还得到了编委会主任委员沈蒲生教授、哈尔滨工业大学和广州大学有关领导、高等教育出版社的大力支持与帮助。为此,特向给予本书支持的专家和领导表示衷心的感谢。

限于编者的水平,书中难免存在一些不足之处,欢迎读者批评指正。

编著者

2000 年 12 月

责任编辑	余美茵
封面设计	王 睢
责任绘图	陈钧元
版式设计	周顺银
责任校对	朱惠芳
责任印制	杨 明

目 录

第1章 FORTRAN 90 语言概述	1
§ 1-1 FORTRAN 90 特点	1
§ 1-2 程序结构	2
§ 1-3 控制语句	5
§ 1-4 数组	8
§ 1-5 自定义数据类型	12
§ 1-6 模块	13
§ 1-7 指针	16
§ 1-8 elf90 简介	18
第2章 结构力学问题的程序设计	23
§ 2-1 程序设计方法概述	23
§ 2-2 只有一个基本部分的多跨静定梁 内力计算程序设计	29
§ 2-3 三铰拱内力计算程序设计	38
§ 2-4 连续梁支座弯矩影响线程序 设计	57
§ 2-5 连续梁支座弯矩影响线程序 修改	66
§ 2-6 程序调试方法概述	71
§ 2-7 程序调试举例	78
§ 2-8 程序质量评价	93
§ 2-9 结论和讨论	94
习题	95
第3章 矩阵位移法	97
§ 3-1 概述	97
§ 3-2 局部坐标单元刚度方程	99
§ 3-3 坐标转换问题	106
§ 3-4 整体分析	111
§ 3-5 结构刚度和综合荷载元素速算法 及单元内力计算	122
§ 3-6 平面刚架程序的主程序及部分 子程序框图	128
§ 3-7 本章结论和一些扩展	131
习题	132
第4章 桁架静力分析程序设计	135
§ 4-1 桁架静力分析程序的总体设计 ..	135
§ 4-2 数据结构与模块接口设计	142
§ 4-3 部分模块详细设计	147
§ 4-4 总刚变带宽存储与方程求解	152
§ 4-5 桁架静力分析源程序	157
§ 4-6 结论和讨论	176
习题	177
第5章 杆系结构静力分析计算程序 的使用	178
§ 5-1 梁结构静力计算算例	178
§ 5-2 桁架静力计算算例	188
§ 5-3 刚架静力计算算例	192
§ 5-4 组合结构静力计算算例	197
习题	198
第6章 结构动力分析程序的使用	199
§ 6-1 结构自振分析	199
§ 6-2 线性体系地震反应分析	214
习题	229
主要参考书目	230

第1章

FORTRAN 90 语言概述

§ 1-1 FORTRAN 90 特点 (1)	§ 1-5 自定义数据类型 (12)
§ 1-2 程序结构 (2)	§ 1-6 模块 (13)
§ 1-3 控制语句 (5)	§ 1-7 指针 (16)
§ 1-4 数组 (8)	§ 1-8 elf90 简介 (18)

FORTRAN 语言是最适合于工程与科学数值计算的一种计算机高级程序设计语言,目前流行的结构计算分析软件一般都采用 FORTRAN 语言作为编程工具。FORTRAN 90(也可简称 F90)是 20 世纪 90 年代发展起来的,它比 FORTRAN 77 更适合于编写大型计算分析软件。为使学过 FORTRAN 77 语言的读者尽快掌握 FORTRAN 90,本章将简要介绍 FORTRAN 90 的一些新功能、新特点。对未接触过 FORTRAN 语言的读者,建议参考有关 FORTRAN 90 语言的教材。此外,本章还向读者推荐了 FORTRAN 90 语言的一个精选子集 elf90,在介绍 FORTRAN 90 过程中指出了 elf90 中不同的语法规定。

§ 1-1 FORTRAN 90 特点

FORTRAN 90 兼容和继承了 FORTRAN 77 的所有先进特性,同时舍弃了一些已经过时的语法规定,结合现代计算机语言发展和科学计算需求,对 FORTRAN 77 的语法和功能进行了大幅度扩展与完善,使得 FORTRAN 有了突破性的发展,具体表现在以下一些方面:

- 数组运算能力增强:数组与数组、数组与普通变量(或常数)之间可以直接进行加、减、乘、除运算操作,而不必再利用循环语句编写程序来完成数组运算,这一功能大大简化了程序的编写工作,同时使程序格式简单、易读、易于修改调试,且更接近于数学描述。

- 自定义数据类型(结构):FORTRAN 90 除继承了原有的内部数据类型外,还提供了用户可根据实际需要设计新数据结构类型(也称结构或导出类型)的功能,比以前版本增强了复杂数据结构描述和处理的能力。

- 模块:模块是 FORTRAN 90 的一个新特性,它是一种在基本程序单元外独立编写的程序单元,用户可以将说明语句、一个或若干个子程序放在一起,定义成一个模块单元以供其他程序单元引用。模块起到共享存储单元与复制程序的功能,使程序编写得到简化。在模块中的子程序也可以设置访问权限,使得某些子程序仅能够被同一模块中的子程序所调用,因此,模块实

现了程序功能的封装,增强了程序的可读性。

● 指针:为了方便处理动态数据、定义更为复杂的数据结构(如树和链表等),FORTRAN 90 借鉴了 PASCAL 和 C 语言中的指针功能,增加了指针操作,使得在处理复杂数据的插入、删除等操作时非常方便。

● 动态变量(或动态数组):为了合理利用存储空间,FORTRAN 90 具有动态分配与释放存储空间的功能,可以申请动态数组,或将指针指向动态空间。从而改变了 FORTRAN 77 采用开定长大公共区,分段存放动态数组,既浪费资源,又存在稍不小心就可能出错的弊端。

● 具有语言扩展功能:FORTRAN 的运算符有固有的功能和定义,当这些已有运算符不能满足用户要求时,可以自己定义运算符,既可以扩充原有运算符的功能,也可以定义新的运算符,这使得 FORTRAN 90 语言具有了扩展功能。

● 数值计算能力得到极大提高:程序计算速度比 FORTRAN 77 有较大提高,能够处理的数据规模有了质的改变,数组大小突破了内存的限制,不再局限于基本内存(或扩展内存)可以自动使用硬盘空间进行计算,使得解决问题的规模大大提高。

● 内部子程序功能:FORTRAN 90 语言为了减少外部子程序(子程序单元)的数量,在外部子程序中可以定义仅供该外部子程序使用的内部函数子程序(或内部子程序),这一功能大大扩展了语句函数的功能。同时,内部子程序的这一特点也使得它不被其他外部程序访问,实现了程序单元的封装功能。

§ 1-2 程序结构

1-2-1 FORTRAN 90 源程序结构与书写格式

FORTRAN 90 是结构化、模块化的程序设计语言,程序单元是构成 FORTRAN 90 程序的基本成分。一个程序单元可以是主程序、外部子程序,也可以是模块,各个程序单元之间是相互独立的。其中内部子程序是包含在程序单元内的程序块,仅供属于同一个程序单元的其他子程序调用,在一个程序单元中可以包含若干个内部子程序。在一个完整的程序中,主程序个数是唯一的,且必须有一个程序单元为主程序。FORTRAN 90 源程序的每个程序单元的组成结构如图 1-1 所示。

FORTRAN 90 注释语句采用感叹号‘!’，在一行中感叹号后面的所有字符都将被编译器忽略而作为注释内容。注释语句可以出现在一行的任何位置。

FORTRAN 90 程序语句可以选择固定格式和自由格式两种方式之一进行书写,自由格式的每行不再分为标号区、续行区、语句区、注释区等 4 个区,且一行可以书写多个语句;当一行有多个语句时,语句间要用分号隔开,一般为了程序可读性较好,仅将较短的语句放在一行,例如:

```
XIC=10.0;XITA=60.0;DELTAL=0.02;IF(XIC<10)XIC=1.0
```

注意:在 `elf90` 中一行只能书写一个语句。

FORTRAN 90 程序的一行最多可以有 132 个字符,当程序语句太长或从阅读角度想要分成几行书写时,可以利用续行符‘&’来完成,注意注释语句不能续行。例如:

PROGRAM、FUNCTION、SUBROUTINE、MODULE 语句	
USE 语句	
FORMAT 语句 ^①	IMPLICIT NONE 取消隐含说明语句
	PARAMETER 与 DATA 语句
	自定义类型 接口块 类型说明说明语句 其他说明语句
可执行语句部分	
CONTAINS 语句	
内部子程序或模块子程序	
END 语句	

图 1-1 FORTRAN 90 程序单元的语句顺序表

```
A=SIN(X)*COS(X) + X*X + 3 &      ! 这里 & 续行符
+ Y*Y + SIN(Y)*COS(Y)           ! 与上一行构成一个语句
```

注意:在 eif90 中 WRITE(或有格式 READ)语句中的格式说明部分不能使用续行符。

1-2-2 变量

FORTRAN 90 的变量名、子程序名可由 1~31 个字符(字母、数字或下划线)组成,但第一个字符必须是字母。与 FORTRAN 77 一样,变量不区分大小写字母。除兼容以前版本外,FORTRAN 90 建议不使用 I-N 隐含规则(因为使用隐含规则,编译器将无法检查出变量的书写错误,而变量的书写错误对程序应该是致命性的错误),也就是建议编程时采用强类型方式,要求在程序单元的说明语句前使用如下语句:

```
IMPLICIT NONE    ! 该程序单元内的所有变量必须进行类型声明才能使用
```

FORTRAN 90 中可以采用新的变量类型声明格式:

```
类型说明符[[([KIND=]种类类别)][[,属性说明]... ::]变量表
```

注意:[]表示可选项,下面不再赘述。

● 类型说明符:可以是内部的(INTEGER、REAL、CHARACTER、LOGICAL、COMPLEX 等 5 种)或派生的(自定义类型)。例如:

```
REAL :: A=10.0,B      ! 定义变量 A,B,并给 A 赋初始值 10.0
```

● 种类类别:为了节省内存单元,同时又能够实现用户编程的需要,对于同一种类型的数据,按照其大小和精度的要求不同,在内存中所占有的存储单元数目也可以是不同的,FORTRAN 90 中提供了对数据的精度和范围进行选择控制的功能,使用种类类别参数对数据的范围和精度进行控制。具体种类类别值将视具体的计算机处理系统而定,当用户在编写程序时,查阅有关手册以便确定每种数据类型的可用种类类别及缺省类别定义。例如:

^① 图中表示各类语句在程序中的书写顺序,从上到下顺序不能改变,左、右无先后顺序。注意:在 eif90 中不允许使用 FORMAT、PARAMETER 和 DATA 语句,在 END 语句前必须要有 STOP 语句(主程序)或 RETURN 语句(子程序或函数),IMPLICIT NONE 语句是必须要有的。

```

INTEGER(KIND=1) :: A      ! 表示定义了种类类别为 1 的整型变量 A
REAL :: B                ! 表示定义了种类类别缺省(4)的实型变量 B
INTEGER*4 POINTER :: A1  ! 整型指针变量 A1 占 4 字节
REAL*8 :: C              ! 实型变量 C 占 8 字节,也可写为 REAL(8)::C

```

注意:在 `elf90` 中整型的种别只能等于 4。

● 属性说明:可以是常数说明符 `PARAMETER`、数组说明符 `DIMENSION`、指针类型说明符 `POINTER`、动态数组类型 `ALLOCATABLE` 等。

- (1) `PARAMETER`:将说明对象定义为常数。
- (2) `DIMENSION`:设置说明对象的数组维数及大小。
- (3) `PUBLIC`:用于设置说明对象具有公共可访问属性。
- (4) `PRIVATE`:将说明对象设置为本程序单元私有的。
- (5) `ALLOCATABLE`:说明用于动态建立指针目标和可分配数组。
- (6) `EXTERNAL`:说明对象名是一个外部函数或虚拟函数。
- (7) `POINTER`:说明对象为指针。
- (8) `TARGET`:说明的对象可以作为与指针结合的目标。
- (9) `SAVE`:说明该名字的结合状态、分配状态、定义状态,且值是保留的。

(10) `INTENT`:意图属性,用它说明子程序中哑元(虚参)的使用意图。其中有三种用法:
`INTENT(IN)`表示将调用程序实参的值传递给子程序有该属性的变量,此类变量不能出现在赋值号左边;
`INTENT(OUT)`表示将子程序的计算结果返回给调用程序,此类变量只能出现在赋值号左边;
`INTENT(IN OUT)`表示调用程序的实参传递给子程序,子程序再将计算结果返回到调用程序,这类变量可以出现在赋值号的两边。例如:

```

REAL(8),DIMENSION(6,6)::A ! 双精度(8字节)实型 6×6 个元素的二维数组 A
REAL*8 ::A(6,6)           ! 与上面语句的作用等价
REAL*4,DIMENSION(6,6)::B ! 单精度(4字节)实型二维数组 B

```

1-2-3 各种程序单元的结构

FORTRAN 90 语言的各种程序单元的结构如下(注意:方括号[]表示可省略项):

● 主程序结构:

```

[PROGRAM 程序名]
  [说明语句部分]
  可执行语句部分
  STOP
END [PROGRAM [程序名]]

```

例如:PROGRAM ZYS1

```

  INTEGER::I=2,J
  J=I+1
  PRINT *,I,J ! 表控输出语句
  STOP
END PROGRAM ZYS1

```

注意:`elf90` 必须要有 `STOP` 语句。

● 内部函数子程序格式

```
FUNCTION 函数子程序名([参数表])
  [说明语句部分]
  [可执行语句部分]
  RETURN
END FUNCTION[函数子程序名]
```

注意:ef90 必须要有 RETURN 语句。

● 外部函数子程序格式

```
FUNCTION 函数子程序名([参数表])
  [说明语句部分]
  [可执行语句部分]
  RETURN
  [CONTAINS
  内部子程序(函数或子例行)]
END [FUNCTION[函数子程序名]]
```

注意:ef90 必须要有 RETURN 语句。

● 模块单元的程序格式

```
MODULE 模块名[(参数表)]
  [使用语句部分]
  IMPLICIT NONE
  [说明语句部分]
  [可执行语句部分]
  [CONTAINS
  内部子程序]
END [MODULE[模块名]]
```

模块的使用格式:USE 模块名 1[, 模块名 2, …… , 模块名 N]

● 内部子例行程序格式

```
SUBROUTINE 子程序名[(参数表)]
  [说明语句部分]
  [可执行语句部分]
  RETURN
END SUBROUTINE[子程序名]
```

● 外部子例行程序格式

```
SUBROUTINE 子程序名[(参数表)]
  [说明语句部分]
  [可执行语句部分]
  RETURN
  [CONTAINS
  内部子程序(函数或子例行)]
END [SUBROUTINE[子程序名]]
```

§ 1-3 控制语句

1-3-1 关系表达式

在 FORTRAN 90 中,为了更接近数学语言描述,比较两个算术表达式值的关系运算符可以直接使用(=, >, <, >=, <=, /=),见表 1.1。

表 1.1 关系运算符表

数学符号	=	≠	≥	≤	>	<
关系运算符	==	/=	>=	<=	>	<

注意:关系表达式 $4 < X < 6$ 是错误的,正确格式应该是 $(4 < X) .AND. (X < 6)$ 。

1-3-2 CASE 选择结构

FORTRAN 90 提供的 CASE 选择结构可以直接处理多分支选择(类似于 C 语言的 SWITCH 语句)。CASE 结构的一般形式为:

```
SELECT CASE(表达式)
  CASE(情况选择 1 列表)
    语句块 1          ! 情况选择 1 条件满足时,仅执行语句块 1
  CASE(情况选择 2 列表)
    语句块 2          ! 情况选择 2 条件满足时,仅执行语句块 2
    .....
  CASE DEFAULT
    语句块 N          ! 前面条件都不满足时,执行此语句块 N
END SELECT
```

表达式的值必须是整型、字符型、逻辑型之一,不能是实型或复型的。块 CASE 与块 IF 结构的区别在于,块 IF 中的分支条件是逻辑表达式,而块 CASE 中的分支条件是一些值的列表。

当表达式的值是数值型(整数)时,情况选择语句的几种表达形式如下:

- 单值表示:如 CASE(1),CASE(10,20),CASE(2,10,30)等。
- 用始值表示:如 CASE(3:)表示 3,4,5,6,⋯及大于 2 的所有整数。
- 用终值表示:如 CASE(: 4)表示小于 5 的所有整数。
- 始终值表示:如 CASE(2: 6)表示 2,3,4,5,6 五个整数。

上述表示可以混合使用,如 CASE(2: 6,8,10: 20,30:)等。

1-3-3 DO 循环结构

FORTRAN 90 扩展了 DO 语句的功能,增加了一种不确定循环次数的 DO 循环结构。确定次数的 DO 循环结构与原 FORTRAN 77 的基本相同,但由于 FORTRAN 90 不建议使用标号功能,循环开始语句和循环终端语句形式有所变化,其他规定不变。两种 DO 循环结构的使用格式分别如下:

- 确定次数的 DO 循环结构的一般形式为:

```
[结构名:] DO 循环变量=初值表达式,终值表达式[,步长表达式]
           可执行语句块 ! 也可以包含新的循环结构
END DO [结构名]
```

```
例如:  LJQ:  DO I=1,100          ! 循环变量 I 从 1 到 100 变化
           J=J + I              ! 将 J 与 1,2,⋯,100 累加
           END DO LJQ          ! 循环终端语句,LJQ 为循环结构名称
```

循环结构名是可以省略的,但对于多重循环使用循环结构名可增加程序的可读性、同时为使用 EXIT 语句提供转移目标。

- 不确定次数的 DO 循环结构的一般形式为:

```
[结构名:] DO
    可执行语句块          ! 也可以包含新的循环结构
END DO [结构名]
```

注意:这种 DO 循环语句一般要配合 EXIT 语句使用,详细用法见 1-3-5 中例题。

在 FORTRAN 90 循环结构中建议循环变量采用整型。特别注意与 FORTRAN 77 语言一样,循环也可以嵌套,但绝对不能出现交叉情况。

1-3-4 DO WHILE 循环结构

对于循环次数确定的循环,使用 DO 语句是比较方便的。但是,对循环次数事先无法确定的循环问题来说,使用次数确定的 DO 语句实现起来是比较麻烦的。利用 DO WHILE 语句可以很方便地实现执行次数不确定的循环。DO WHILE 循环结构的一般形式为:

```
DO WHILE(逻辑表达式)      ! 逻辑表达式为真值时进行循环
    语句块
END DO
```

```
例如: INTEGER :: I1=0, J1 =0      ! 定义两个初值为 0 的整型变量
DO WHILE(I1 >= 0)                ! 非负数时执行循环,当读入负数时结束循环
    READ *, I1                    ! 读入一个整数
    J1 = J1 + I1                  ! 将读入的数累加
END DO
```

1-3-5 EXIT 语句和 CYCLE 语句

EXIT 语句和 CYCLE 语句是循环结构中的专用语句,对循环有控制作用。

● EXIT 语句的一般格式: EXIT [DO 结构名]

EXIT 语句的作用是停止当前循环,将控制转移到当前循环(或指定名称的循环结构)之外,因此也称 EXIT 为循环出口语句,实现循环的异常结束。例如:

```
REAL :: A1, B1 = 0
DO
    READ *, A1 ; IF(A1 < 0) EXIT      ! 无限次(或称次数不定)循环
    B1 = B1 + A1                    ! 读入负数结束循环
END DO                               ! 将非负数累加
```

对于单重循环 EXIT 后不能有循环结构名,对多重循环最好加 DO 结构名,有利于增强程序的可读性。如果 EXIT 后面无循环结构名,执行到 EXIT 后程序将结束当前循环,执行该循环下面的语句。

● CYCLE 语句的一般格式: CYCLE [DO 结构名]

当执行 CYCLE 语句时,程序跳过 CYCLE 下面语句,控制又返回到所指定的 DO 结构名继续执行。它与 EXIT 语句结束循环的功能不同,而是跳过一些语句后继续重复循环。例如:


```

PROGRAM LC
  IMPLICIT NONE                ! 取消隐含规则,所有的变量必须说明
  REAL :: A1, B1=1.0          ! 定义两个实数,且 B1 初始值为 1.0
  DO                          ! 无限次循环
    READ *, A1 ; IF(A1<0)CYCLE ! 读入负数时重复循环
    B1 = B1 *A1 ; IF(B1>1.0E6)EXIT ! 将非负数累乘,当 B1>106 时结束循环
  END DO
  STOP
END PROGRAM LC

```

§ 1-4 数组

所谓数组就是具有相同数据类型及种类的标量数据(称为数组元素)的集合,数组可使变量个数减少、程序简练,它是 FORTRAN 语言提供的一种数据结构,在编制科学计算程序中数组的应用非常普遍。由于 FORTRAN 90 对数组的特性有了较大的扩充,在数组的操作方面提供了强大的内部函数和增加了很多新功能,使所编制程序的结构更加清晰,易于调试、维护和移植。下面将介绍 FORTRAN 90 的新特性和新功能的最基本内容(更多的功能参见有关参考书或 FORTRAN 软件的联机帮助信息)。

1-4-1 数组定义

FORTRAN 90 中数组定义的一般形式为:

类型说明, DIMENSION(数组形状), 其他属性说明 :: 数组名列表

或 类型说明, 属性说明 :: 数组名(数组形状)[, 数组名(数组形状), ……]

数组最多可以有 7 维,每一维用界偶(下界:上界)来表示,数组的上界必须声明,而数组的下界可不必声明,省略时下界默认值为 1。例如:

```

REAL, DIMENSION(10)::A, B      ! 说明 A, B 是一维实型数组,有 10 个元素
REAL :: A(10), B(20)          ! 一维实型数组, A 有 10 个元素, B 有 20 个元素
REAL, DIMENSION(6)::T, X(3)   ! 一维数组 T 有 6 个元素, X 有 3 个元素
INTEGER, DIMENSION(1:10, -3:8)::C ! 说明 C 是二维整型数组, 120 个元素
REAL *8, DIMENSION(6, 6)::T   ! 说明 T 是二维双精度数组, 36 个元素
REAL(8)::G(6, 6)             ! 说明 G 是二维双精度数组, 36 个元素

```

1-4-2 数组片段

所谓数组片段,是指由数组中一些确定的元素组成的新集合,它是原数组的一个子集,数组片段本身也是一个数组,具有与原数组同样的特性。数组片段有如下几种表示方法:

● 下标偶表示

数组名(整数表达式 1:整数表达式 2)