普通高等教育“十二五”规划教材

工程分析程序设计

魏进家 陈 斌 周屈兰 刘小民

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

普通高等教育“十二五”规划教材

工程分析程序设计

魏进家 陈 斌 周屈兰 刘小民

Fortran & C

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书系统地介绍了当前在科学与工程计算领域广为使用的 Fortran 90 和 C 语言，以 Fortran 90 语言为主进行详细介绍，对 C 语言进行简要介绍，使读者在精通 Fortran 90 语言程序设计的同时，还可以读懂和使用 C 语言程序。最后介绍 Fortran 和 C 的混合语言编程，使读者了解 Fortran 和 C 例程的相互调用，从而较为全面地掌握工程计算的主要常用语言并发挥其各自的优点。书中语言精炼易懂并配有解读，同时注意结合工程应用特点，选择典型的例题和习题，有利于培养读者从事工程计算分析的程序设计能力。本书可作为高等院校程序设计课教材和工程技术人员的参考用书。

图书在版编目(CIP)数据

工程分析程序设计/魏进家等编著. —西安：西安交通大学出版社，2015.1
ISBN 978-7-5605-6929-1

Ⅰ.①工… Ⅱ.①魏… Ⅲ.①工程分析-程序设计
Ⅳ.①TU712 ②TP311.1

中国版本图书馆 CIP 数据核字(2014)第 300136 号

书　　名　工程分析程序设计
编　　著　魏进家　陈　斌　周屈兰　刘小民
责任编辑　任振国

出版发行　西安交通大学出版社
（西安市兴庆南路 10 号　邮政编码 710049）
网　　址　http://www.xjtupress.com
电　　话　(029)82668357　82667874(发行中心)
(029)82668315　82669096(总编办)
传　　真　(029)82668280
印　　刷　陕西时代支点印务有限公司

开　　本　787mm×1 092mm　1/16　**印张**　19.625　**字数**　476 千字
版次印次　2015 年 1 月第 1 版　2015 年 1 月第 1 次印刷
书　　号　ISBN 978-7-5605-6929-1/TU·139
定　　价　35.00 元

读者购书、书店添货、如发现印装质量问题，请与本社发行中心联系、调换。
订购热线：(029)82665248　(029)82665249
投稿热线：(029)82664954
读者信箱：jdlgy@yahoo.cn

前　言

本书系统地介绍了当前在科学与工程计算领域广为使用的计算机语言 Fortran 90 和 C，内容包括程序设计基础、结构化程序设计、数组、派生类型、指针以及格式化输入输出和文件操作等。本书以 Fortran 90 语言为主进行详细介绍，对 C 语言进行简要介绍，使学生在精通 Fortran 90 语言程序设计的同时，还可以读懂和使用 C 语言程序。最后介绍 Fortran 和 C 的混合语言编程，使学生了解 Fortran 和 C 例程的相互调用，从而较为全面地掌握工程计算的主要常用语言并发挥其各自的优点。

本书的程序举例浅显易懂并配有解读，同时注意结合工程应用特点，选择部分常用数值算法程序分析讲解，使读者能够深入体会计算机语言在解决工程问题中的作用，培养其学习兴趣。除了主要讲解 Fortran 90 外，还穿插部分 Fortran 77 程序和格式，并与 Fortran 90 对比，使读者在掌握 Fortran 90 的同时，对 Fortran 77 的格式和语法有所了解，具备读懂 Fortran 77 程序的能力。本书多数章节后面都配有一定数量的习题，便于读者通过课后练习加深对所学内容的理解和掌握。

本书内容遵循由浅入深的原则，力求主次分明，重点突出。在 Fortran 90 语言部分，首先按照程序设计基础、结构化编程与逻辑运算、循环的顺序学习进行程序设计的基本知识，然后在此基础上介绍模块化程序设计、数组、派生数据类型、指针等深入的内容，使读者容易接受。在学完 Fortran 90 后，读者对程序语言设计就具有一定的基础。在 C 语言部分，按照基础知识、输入输出、指针和数组以及函数等内容进行介绍，并与 Fortran 90 语言的异同进行比较，使读者更易掌握。本书最后介绍 Fortran 和 C 的混合语言编程，加深读者对两种语言的理解，掌握相互调用的能力，这也是本书的一大特色。

本书是在西安交通大学《工程分析程序设计》课程讲义的基础上，结合三年的教学实践经验进行修改完成的。由魏进家、陈斌、周屈兰和刘小民编写，其中刘小民编写第 1、2 和 5 章，周屈兰编写第 3、4 和 6 章，魏进家编写第 7、8 和 9 章，陈斌编写第 10 和 11 章。魏进家负责全书统稿和审定工作。

本书由西安交通大学李会雄教授悉心审阅，提出许多宝贵意见，谨致衷心谢忱。

由于编者水平有限，书中难免存在不妥之处，敬请读者予以批评指正。

编　者

2014 年 5 月

目 录

第 1 章　Fortran 背景知识

本章首先简述 Fortran 语言的历史，让大家对 Fortran 的过去和未来有所了解；其次，会讲述 Fortran 90/95 新的语言特征；最后还会简述 Fortran 编译器的演化历史及其基本的编译方法。

1.1　Fortran 语言简史

Fortran 语言是一种在国际上广泛流行的适于科学计算的程序语言，也是世界上产生最早的高级程序设计语言。Fortran 是 Formula Translation 的缩写，即数学公式翻译器。

Fortran 的起源要追溯到 1954 年 IBM 公司的一项计划。IBM 尝试着在 IBM 704 计算机上开发一套程序，它可以把接近数学语言的文本翻译成机器语言。1957 年，他们开发出第一套 Fortran 编译器，一个革命性的产品 Fortran 也随之诞生了。20 世纪 60 年代初，在国防、教育和科技领域对高性能计算工具的迫切需求下，Fortran 语言蓬勃发展，成为当时统治计算机世界的高级语言之王，有很多软件公司都推出了自己的编译程序。但是，各个公司为了强调自己产品的功能，都在原来的 Fortran 语言之外添加了一些自己的独门语法，从而导致了 Fortran 语言移植上的困难。

1962 年，为了统一不同公司、不同硬件平台上的 Fortran 语言，美国国家标准局(ANSI)开始了语言标准化的尝试，并在 1966 年制定了 Fortran 语言的统一标准，即 Fortran 66。由于标准文档过于简单，约束力不强，Fortran 66 标准发布后，语言的统一问题并没有得到彻底解决。

1978 年，美国国家标准局正式公布了 Fortran 语言标准的第一个修订版本，这套标准就是所谓的 Fortran 77。Fortran 77 除了保留了 Fortran 66 标准的大部分内容外，还添加了许多适于结构化程序设计与维护的新特性。Fortran 77 让 Fortran 成了一种真正规范、高效和强大的结构化程序设计语言。

继 Fortran 77 标准之后，1992 年国际标准组织 ISO 又正式公布了崭新的 Fortran 90 标准。Fortran 90 标准除了引入自由的代码风格外，还引入了模块、接口块、自由定义(派生)数据类型和运算符、可动态分配和参与复杂运算的数组、例程重载、指针、递归等重要的语法特征。这不但使结构化的语言更趋完善，也使其具备了少量的面向对象语言特征。

1997 年 ISO 又发布了 Fortran 95 标准。Fortran 95 在 Fortran 90 的基础上，加强了 Fortran 语言在并行运算方面的支持，并进一步完善了派生类型、指针、数组等要素的相关语法。

2004 年 5 月，在 ISO、IEC 的联合工作组 JTC1/SC22/WG5 以及美国标准委员会的共同努力下，终于推出了 Fortran 2003 标准。Fortran 2003 对 Fortran 95 做了较多的改进，添加了很多新特性，例如增强的派生数据类型、面向对象编程、增强的数据操作功能和与 C 语言互操作等。Fortran 2003 近乎彻底地解决了语言现代化的问题。

Fortran 2003 之后的下一个版本是 Fortran 2008。和 Fortran 95 一样，Fortran 2008 也只是一个小改版，略微更正了 Fortran 2003 的一些问题，并且合并了 TR－19767 的语言功能。

以 Fortran 66 为基准，我们可以把后续的 Fortran 77/90/95 以及 Fortran 2003/2008 均视为对 Fortran 语言标准的修订。在历次修订中，Fortran 77、Fortran 95 和 Fortran 2008 是修订幅度相对较小的版本，而 Fortran 90 和 Fortran 2003 则是锐意变革的大修版本。

由于正式支持 Fortran 2003 标准的编译器还没有出来，部分编译器只支持部分 Fortran 2003 的语法，如 ivf 等。所以，本书重点讲解 Fortran 90 的语法及应用，必要时也介绍 Fortran 95 语法。

1.2 Fortran 90/95 新的语言特征

本节将分别介绍 Fortran 90 和 Fortran 95 新的语言特征。

1. Fortran 90 的新语言特征

(1)自由书写格式。Fortran 90 提供的新的自由书写格式，取消了许多限制，没有保留列，没有规定每行的第几个字符有什么作用，而且尾部可以出现注释，空格在某些情况下是有意义的。例如：F U N C T I O N 并不等于 FUNCTION。

(2)模块。模块是 Fortran 90 引入的一种新的程序单元，它的功能远比 Fortran 77 的数据块程序单元强大。模块包含了数据、例程、例程接口等要素的声明，可以供其他程序单元引用和支持面向对象的程序设计。模块还能将模块内部的实体隐藏起来，以提供数据抽象、编写安全、可移植的程序代码。

(3)自定义(派生)数据类型和操作符。Fortran 90 除了提供固有数据类型，还允许用户定义自己的数据类型，即派生类型。派生类型可以整体访问，也可以直接访问当中的元素；既可以重载固有操作符(如＋、＊)，也可以定义新的操作符，以适应派生类型数据操作要求。

(4)数组功能加强。在 Fortran 90 中，可以直接对整个数组或数组段进行操作。可以对数组整体、部分及隐式赋值(包括可选择性赋值的 WHERE 语句)，声明数组常量，定义派生类型数组值函数，用数组构造子规定一维数组的值，利用 ALLOCATABLE 或 POINTER 属性为数组动态分配内存。新的固有例程能够创建和使用多维数组，支持数组计算(如 SUM 函数累加数组元素的和)。

(5)例程重载。

(6)同 C＋＋相似，Fortran 90 也支持例程重载。例程重载是指不同参数列表的例程被赋予相同的名字，例程调用时，编译器会依据所传递的实参类型，按参数匹配的原则调用相应的例程。如固有函数 ABS，其参数既可以是整数，也可以是实数或复数。例程重载必须使用接口块，并以调用时的例程名命名接口块。

(7)指针。Fortran 90 指针允许动态访问和处理数据，可以用来创建动态数组和派生类型的动态数据结构(如链表)。指针可以指向固有数据类型或派生类型，一旦和同类型的目标变量相关联，指针可以代替目标出现在表达式和赋值语句中。

(8)递归。

(9)Fortran 90 例程可以实现递归，但必须在(FUNCTION 或 SUBROUTINE)例程原型中添加关键字 RECURSIVE，当例程是函数时，还必须添加 RESULT 子句。

(10)接口块。Fortran 90 提供了接口块，用来向调用程序描述外部例程的接口信息，可以包含在主程序、外部例程和模块这三种程序单元中。

(11)封装机制。类似于 C++中的类,Fortran 90 可以通过模块程序单元实现对派生类型数据连同其操作例程的封装,通过其公有接口,供别的程序单元使用,以扩展 Fortran 功能,使之适合特殊的应用需求,例如模拟面向对象的程序设计、数据库管理、建立动态数据结构等。

2. Fortran 95 的新语言特征

(1)FORALL 语句和构造。在 Fortran 90 中,可以通过数组构造子、RESHAPE 和 SPREAD 固有例程逐元素地构造数组,Fortran 95 的 FORALL 语句和构造则提供了另一种数组操作方式。FORALL 允许通过元素下标对数组元素、数组段、字符字串或指针目标进行操作;类似于隐式 DO 循环,FORALL 构造可使几个数组赋值语句共享相同的下标循环控制表达式。

FORALL 是 WHERE 的一般形式,两者都通过隐式循环对数组进行操作,只不过 FORALL 是使用元素下标,WHERE 则针对整个数组。

(2)PURE 用户定义例程。在用户定义的例程(子程序或函数)原型前添加 PURE 关键字,向系统表明该用户定义例程没有任何负面影响。也就是说,它们不会修改输入参数,也不会修改任何在例程外部可见的其他数据。

(3)ELEMENTAL 用户定义例程。在用户定义例程原型前添加 ELEMENTAL 关键字,它是 PURE 例程的特殊形式。当传递数组参数时,每次对一个数组元素进行操作。要使用 ELEMENTAL 例程,必须在调用程序中建立其接口块。

(4)CPU_TIME 子程序。该子程序通过参数返回特定 CPU 处理器的时间,单位为秒,参数须是单一的实数。

(5)NULL 函数。在 Fortran 90 中不能直接初始化指针为空指针。Fortran 90 指针必须先与目标变量相关联,或动态分配内存,然后才能使用 NULLIFY 函数设置空指针。Fortran 95 则可利用 NULL 函数直接初始化指针为空指针。

(6)WHERE 结构的扩展。WHERE 结构经过扩展,可以包含一个 FORALL 结构的嵌套,而 FORALL 结构又可以包含 WHERE 语句或结构的嵌套。两者的配合使用具有强大的功能。

(7)默认初始化。默认初始化可以应用于派生类型,包括形参,这样就能保证具有指针成员的派生类型对象能够一直可以访问,从而避免了出现内存可分配,但是不能去分配的情形。

对于指针,可以使用新的固有函数 NULL 来给出初始的关联状态,也可以在使用数据之前,使用 NULLIFY 语句来获得初始化。

(8)固有函数 CEILING,FLOOR,MAXLOC,MINLOC 的扩展。在 Fortran 90 和高性能 Fortran 之间,某些固有函数以及相关函数存在某些不兼容,因为在高性能 Fortran 里面,在不同的变元位置增加了一个 DIM 函数。

在 Fortran 95 里面,就放松了对于变元顺序的要求。这样在变元序列当中,MASK 和 DIM 的出现就可以是任意的了,从而保证了 Fortran 95 与高性能 Fortran 的兼容性。

(9)可分配数组的动态去分配。在 Fortran 95 里面,当退出一个给定的作用域时,其中没有通过使用 SAVE 而得到保留的可分配数组,就自动地去分配,和使用 DEALLOCATE 语句的效果一样。这样就可以防止可能发生的内存遗漏,从而规范分配过程。

(10)名称列表输入里面的注释。在名称列表输入记录当中可以使用注释,从而方便了用户。

(11)最小域宽格式说明。

(12)在使用数值的格式输出的时候,运用增强的输出格式编辑描述符,就可以对域宽进行极小化,从而避免输出时的白边。

(13)用于支持 IEEE 754/854 浮点运算标准的某些修改。在所有以前的 Fortran 版本里面,都不区分+0. 和-0.,即正 0 和负 0,在 Fortran 95 版本里面,就能够区分这两者了,即在使用 SIGN 函数时,可以通过让第一个变元为 0,而第二个变元为负号,就得到负 0。

值得注意:Fortran 95 已经删除了 Fortran 90 中标明为过时的某些语言特征,也新标出了一些过时的语言特征,但是还没有删除。例如:算术 IF 语句,计算 GO TO 语句,语句函数和固定源码形式等语言特征在 Fortran 90 里面还允许正常使用,但在 Fortran 95 里面已被划归为过时的语言特征;实型和双精度实型的 DO 变量,PAUSE 语句,ASSIGN 语句,带标签的 GO TO 语句等语言特征在 Fortran 90 里面还可以使用,但在 Fortran 95 里已经被删除了。

1.3 Visual Fortran 编译器的演化和编译

1. Visual Fortran 编译器的演化

当前,国内使用较多的 Fortran 编译器或可视化集成开发环境为 Visual Fortran,它起源于 Microsoft 的 Fortran PowerStation 4.0。这套工具后来卖给了 Digital 公司继续开发,第二个版本称为 Digital Visual Fortran 5.0。Digital 被 Compaq 并购后,接下来的 6.0、6.1、6.5 和 6.6 版本称为 Digital Visual Fortran。本书使用目前最新的(Digital)Visual Fortran 6.6 专业版开发实例。其实不管是 6.0,还是 5.0,甚至是 4.0 的 PowerStation,其使用方法都是一样的。

Visual Fortran 被组合在 Microsoft Visual Studio 集成开发环境中。Visual Studio 提供一个统一的操作界面,这个界面包括文字编辑器、Project 的管理、调试工具等。而编译器则是使用类似 Plug In 的方法组合到 Visual Studio 中,用户在使用 Visual Studio 和 Visual C++时,看到的都是相同的操作界面。

Visual Fortran 6.6 除了完全支持 90/95 语法外,扩展部分还提供完整的 Windows 应用程序开发工具,可以直接建造 Win 32 DLL(动态链接库)、基于组件对象模型 COM 的组件等,专业版还包含了 IMSL 的数值链接库。另外,它还可以和 Visual C++ 6.0 互相链接,将 Fortran 和 C/C++语言的程序代码混合编译成同一个执行文件(EXE 或 DLL)。

2. Visual Fortran 编译器的编译过程

Visual Fortran 编译器的功能十分强大,包括编译器(Compile)、连接器(Link)、链接库(Library)、说明文件(Help)、分析工具(Profile)。本节只介绍编译器最基本的编译方法。

安装好 Visual Fortran 后,运行 Developer Studio 就可以开始编译 Fortran 程序了。编译程序的过程可以归纳为:

(1)建立一个新的 Project,Project 会保存成 *. dsw 的文件。建立新的 Project 的方法为:打开 File,选择 New,选择 Project 选项卡,选择 Fortran Console Application 格式,给定 Project 名称。

(2)生成一个新的程序文件(打开 File,选择 New,选择 Files 选项卡,选择 Fortran Free

Format Source File，给定文件名称），或是插入一个已有的程序文件（选项 Project/Add to Project/Files）。程序代码会保存成 *. f90 或 *. for 的文件。单击 File/Save Workspace 后，会记录 Project 所包含的程序文件。

（3）用 Build 菜单中的 Execute 选项来编译并运行程序，或只是单击 Build 选项来只做编译不做运行。

（4）要写新的程序可以另外再建立一个新的 Project，或是直接更换 Project 中的文件。千万不要把两个独立的文件放入同一个 Project 中，否则会导致编译过程出现错误。

（5）下次要修改程序时，可以直接使用 File/Open Workspace 来打开 *. dsw 的 Project 工作文件。

第 2 章　Fortran 程序设计基础

本章主要介绍 Fortran 的基础知识：程序书写、字符集及标识符、数据类型、声明的有关事项、算术表达式及表控输入/输出语句。通过本章的学习，可以编写简单的 Fortran 程序。

2.1　程序书写

1. Fortran 程序结构

Fortran 程序是一种段式结构。每个 Fortran 程序由一个主程序段和若干个子程序段及模块组成。子程序段和模块根据需要可有零个或多个，但主程序段有且仅有一个。每个程序段都有自己的段头语句，但主程序段的段头语句可以省略。每个程序段可以独立编写，程序运行总是从主程序段开始。

下面我们来看一个加法计算器程序的实例，它只包含一个程序单元，也就是主程序的 FORTRAN 程序。

例 2-1　简单 Fortran 程序的构造形式

```
1.  PROGRAM MAIN
      ! 加法计算器程序
2.    REAL A, B, C
3.    A = 1000
4.    B = 200
5.    C = A + B
6.    PRINT *,'C = ',C
7.  END PROGRAM MAIN
```

第一行的 PROGRAM 关键字标识 Fortran 主程序，后接程序名，这一行是可选的；以感叹号开始的第二行是注释，不参加编译；第三行中的 REAL 将其后面的变量声明为实数型。这三行为非执行部分，之后的部分(END 语句之前)为执行部分。

由此给出简单的 Fortran 90 程序的构造形式：

[PROGRAM 程序名]
[声明语句]
[执行语句]
END[PROGRAM[程序名]]

方括号内的部分是可选的，END 语句是唯一必须的，它通知编译器：程序编译到此结束。END 语句中的程序名可以省略，但若出现程序名，必须同时出现 PROGRAM 关键字。

下面是一个最简单的 Fortran 程序：

例 2-2

```
END
```

下面是一个稍微简单的打印输出程序：

例 2-3

```
PROGRAM HELLO
  PRINT *,"HELLO WORLD!"
END
```

2. Fortran 语句

语句是构成 Fortran 程序的基本单位。按照 Fortran 对程序进行编译、执行过程中所起的作用，Fortran 的语句分为非执行语句和可执行语句。当需要引入或说明一个程序单元或子程序，或者是说明数据类型时，就需要使用非执行语句。程序执行非执行语句时不会对计算机产生任何操作。当需要计算机进行一个指定动作时，就需要使用可执行语句，如：赋值、输入输出、控制转移等。

Fortran 语句在使用时有特定的顺序要求。合适的语句位置如表 2-1 所示。

表 2-1　Fortran 语句的顺序

<table>
<tr><td colspan="3">PROGRAM、FUNCTION、SUBROUTINE、BLOCK、DATA 、MODULE</td><td rowspan="8">注释行</td></tr>
<tr><td colspan="3">USE</td></tr>
<tr><td rowspan="3">FORMAT</td><td colspan="2">IMPLICIT NONE</td></tr>
<tr><td rowspan="2">PARAMETER</td><td>IMPLICIT 及其他说明语句</td></tr>
<tr><td>DATA</td><td>可执行结构</td></tr>
<tr><td colspan="3">CONTAINS</td></tr>
<tr><td colspan="3">内部例程或模块例程</td></tr>
<tr><td colspan="3">END</td></tr>
</table>

注：其中处于同一水平位置的各语句之间没有严格的前后顺序，而不同的行则表示了严格的在程序当中出现的前后顺序。

下面给出语句顺序所应遵守的一般原则：

(1)程序段的段头语句，只能出现在每个程序段开始的位置。如：PROGRAM、FUNCTION、SUBROUTINE、BLOCK、DATA、MODULE 等；

(2)如果出现 USE 语句，则只能出现在段头语句之后、其他语句之前；

(3)IMPLICIT NONE 语句应紧跟在 USE 语句之后，在其他说明语句之前；

(4)FORMAT 语句和 DATA 语句也可以放置在可执行语句中间，不过把 DATA 语句放置在可执行语句中间是一种过时的做法；

(5)PARAMETER 语句可以出现在 DATA 语句和可执行语句之前、IMPLICIT NONE 语句之后的任何位置上；

(6)其他说明语句应出现在 DATA 语句和可执行语句之前；

(7)注释行可以写在程序的任何位置上；

(8)如果出现内部例程或模块例程，则必须跟在 CONTAINS 语句后面；

(9)END 语句是程序段的结束语句，只能出现在整个程序段的最后。

3. 空格

在 Fortran 语言中，空格通常没有意义，它不参加编译。适当地运用空格能够增加程序的可读性，如程序中的代码缩进。由于空格默认的功能就是分割不同的词汇，所以在代表有意义字符序列的记号(token)内，比如：标号、关键字、变量名、操作符等，不能随意使用空格。例如，RE AL、SUBRO UTINE 、MO NEY 和< =都是非法的，但如果变量名字符之间以连接号_连接起来，则属于合法的，例如 NA_ME，ADD_RESS，CHIN_PEO_NUM 都是合法的变量名。

一般情况下，记号之间需要用空格隔开，例如：100CONTINUE 是非法的，因为标号 100 和关键字 CONTINUE 是两个独立的记号。但并不是所有情形下的记号之间都必须要有空格，在不会产生混乱的前提下，有些语句关键词之间的空格是可以省略的，例如：END PROGRAM 和 ENDPROGRAM 都是合法的。

4. 注释

注释不是语句，不影响程序的执行，在编译时被忽略。但适当的注释能增强程序的可读性。

Fortran77 的注释方式为：第一列上由字符“C”或“ * ”作为注释的标志，第 7 至 72 列上写上注释内容。

Fortran 90 只提供了一种注释方式：以感叹号开始的语句作为注释，字符串内的感叹号除外。注释可以是一整行，也可以是空白行。注释的位置可以是任意的，关键是一行的任意位置只要出现了注释符“!”，那么它后面直到行末，都会被编译器认为是注释而不加理会。因此不要把语句放置在一行内的注释后面。

5. Fortran 程序书写格式

Fortran 程序的书写格式有两种：一种是 Fortran 90 之后的自由格式，一种是 Fortran 90 之前的固定格式

1)固定格式

早期的计算机，还没有使用键盘/显示器作为输入/输出设备，那时的程序是利用穿孔卡片一张张地记录下来，再让计算机来执行。固定格式正是为了配合早期使用穿孔卡片输入程序所发明的格式。

固定格式是 Fortran 程序的一种旧式写法，采用这种写法的程序文件扩展名为“. F”或“. FOR”。

在固定格式中，每行有 80 列，这 80 列被分为四个区，分别书写不同的内容：

(1)第 1～5 列为标号区。可以写 1 至 5 位整数作为语句标号，也可以没标号，但转向、格式等语句中必须有标号。同一程序单元中，不得采用重复的标号。另外，标号不得全为零字符，且标号区中的空格不起作用。标号区中某一行的第一列若出现“C”或“ * ”字符，说明该行是注释行，仅起说明作用，不会被编译。该程序中的第一行即为注释行，10 为格式标号。

(2)第 6 列为续行区。如果此列出现数字“0”和空格以外的任何字符，且第 1～5 列均为空白时，则该行作为上一行的续行。例 2-4 程序中的“ * ”即为续行标志。需要注意的是，注释

行没有续行。

(3)第 7～72 列为语句区。语句可以从第 7 列以后任何位置开始书写,但一行只能写一个语句。语句区内的空格(不包括引号内字符串中的空格)在编译时被忽略。

(4)第 73～80 列为注释区。注释区用于程序员书写提示信息,在编译时不予处理。

下面是一个固定格式的 Fortran 程序实例。

例 2-4　固定格式的 Fortran 程序

为了直观说明固定格式,本例以表格的形式列出固定格式的行与列。

1	2	3	4	5	6	7 至 72	73 至 80
C						FIXED FORMAT	
*						已知 a,b,c,求一元二次方程的根	
						PROGRAMMAIN	
						a=1.0	
						b=3.0	
						c=−6.0	
						x1=(−b+ sqrt(b * b−4.0 * a * c))/(2.0 * a)	
						x2=(−b− sqrt(b * b−4.0 * a * c))/	
					*	(2.0 * a)	
						WRITE(* , 10)x1, x2	
1	0					FORMAT(1x, 2f6.2)	
						END	

随着穿孔卡片的淘汰,固定格式已经没有必要再继续使用下去。但由于现在仍然可以找到很多用固定格式编写的旧程序,所以对于固定格式的使用规则,读者还是需要了解的。

2)自由格式

自由格式是 Fortran 90 之后的新写法,是目前最流行的书写格式,它取消了许多限制。它没有规定每行的第几个字符有什么作用。自由格式的 Fortran 90 文件扩展名为. f90。对于自由格式,需要注意的事项有以下几点:

(1)每行最多写 132 个字符;

(2)叹号"!"后面的内容为注释;

(3)如果需要写语句标号,标号可放在每行程序的最前面;

(4)一行之内可以不止包含一条语句,但语句之间必须用(;)加以分隔;

(5)一行程序代码的最后如果是符号"&",则代表下一行是该行的继续;

(6)如果一行程序代码开头是符号"&",则其上一行的最后非空格符必须是一个 &;且 & 号前不能有空格,代表该行是上一行的继续;

(7)Fortran 90 允许出现多达 39 个续行。

以下是用自由格式书写的 Fortran 程序实例。

例 2-5 自由格式的 Fortran 程序

```
! FREE FORMAT
1.  PROGRAM MAIN
2.    REAL A,B,C,P,Q,X1,X2
3.    A=1.0; B=3.0; C=-6.0              ! 一行书写多个语句
4.    P=-B/(2.0*A)
5.    Q=SQRT(B*B-4.0*A*C)/&             ! 下一行是续行
6.    (2.0*A)
7.    X1=P+&!   下一行语句是续行,下一行开头是&,则该行
                ! 最后非空格符必须是&,且&号前不能有空格
      &Q        ! 此行是上一行的续行
8.    X2=P-Q
9.    WRITE(*,10)X1, X2
10.   10 FORMAT(1X, 2F6.2)
11. END
```

2.2 字符集和标识符

1. 字符集

"字符集"是指编写程序时所能用到的全部字符和符号。Fortran 语言的基本字符集由下列字符组成：

(1)26 个英文字母(A～Z 和 a～z)；

(2)数字 0～9；

(3)下划线(_)；

(4)特殊字符 :=+-*/(),.'!"%&;<>?_$ 以及空格符(书写时用空格键表示)。

值得注意的是，Fortran 不区分大小写英文字母。READ、ReaD、read，都会被视为相同的命令。但在以下位置时，因为大小写是字符型数据的不同数据取值，所以大小写必须区分：

(5)作为字符常量的字符串里面；

(6)输入输出的记录里面；

(7)作为编辑描述符的引号或撇号里面。

特殊字符主要具有功能的意义，如编辑功能，运算功能，语法功能等。

除了上面列出的基本字符集外，还有一些辅助的字符，它们在不同的平台有不同的用法约定。

辅助字符分两类：可打印字符和不可打印字符。

(8)可打印字符。各种本地化语言的字符，例如，汉字、希腊字母等，都可以应用在字符串、注释和输入输出纪录当中。

(9)不可打印字符。主要就是控制字符，例如制表符 Tab 键。