

837602

FORTRAN语言程序设计

王开铸 张不同 王宇颖 编

5087

1678

哈尔滨工业大学出版社

87

78

FORTRAN语言程序设计

王开铸 张不同 王宇颖 编

哈尔滨工业大学出版社

内 容 简 介

本书主要介绍FORTRAN程序设计语言及简单的程序设计方法。全书共分八章：第一章，FORTRAN程序基础；第二章，直接程序设计；第三章分枝程序设计；第四章，循环程序设计；第五章，过程程序设计及调用语句；第六章，数据联系语句与数据块子程序；第七章，输入输出及格式语句的进一步描述；第八章，FORTRAN程序的运行。各章后均配有适量的习题。

本书可作为工科院校本科生的教材，也可供工程技术人员参考、自学。

FORTRAN 语言程序设计

王开铸 张不同 王宇颖 编

哈尔滨工业大学出版社出版

新华书店首都发行所发行

鸡西市印刷二厂印刷

开本787×1092 1/16 印张14.5 字数329000

1987年9月第1版 1987年9月第1次印刷

印数 1—12000

书号15341·58

定价2.45

ISBN T—5603—0017—0/TP·2

前 言

FORTRAN语言是一种计算机的程序设计语言。不论是大、中、小、微型计算机，几乎都配有FORTRAN语言，它是国际上应用最广泛的程序设计语言之一，尤其适用于解决工程和科学研究的数值计算问题。目前，程序设计语言已成为我国高等院校学生的必修课。为了满足教学需要，编写了“FORTRAN语言程序设计”这本书。

本书主要介绍FORTRAN程序设计语言及程序设计的基本方法。全书分为两大部分：前一部分介绍了基本的程序设计方法，让读者尽早地学会编写简单的程序；后一部分较深入地介绍了怎样才能编写出一个好的程序，使程序和输出的结果具有可阅读性和直观性。

根据多年的教学体会，对本书内容的安排遵循如下三条原则。首先，为避免引起初学者对FORTRAN语言各种概念和规定的枯燥繁琐感，将FORTRAN语言中的基本概念和规定，采用集中和分散相结合的原则加以处理，把基本概念集中放在前几章，而把一些难以理解的概念和内容联系起来，分散到其它有关章节避免了概念的罗列。其次，强调FORTRAN语言各种语句的语法、语义和语用相结合的原则，以便读者准确地掌握每一个基本的语句的含义。结合程序设计的方法把各个语句分散到各种程序设计方法中介绍，可使读者尽早地步入程序设计阶段。第三，全书按照由浅入深、循序渐进编排的原则，先介绍四种程序设计方法，即直接、分枝、循环和子程序设计方法。在这四种程序设计方法的基础上，进而介绍了一些程序设计的技巧，以使读者能够顺利地编写出较好的程序。

本书是在1981年由王开铸、张不同、王宇颖、姜文清、王国鸣、褚宾生同志编写及李光汉、陈骏林同志审阅的讲义“FORTRAN程序设计”的基础上，由王开铸、张不同、王宇颖同志几经修改、增新而写成的。书中的全部程序由吉朝艳同志调试通过。本书经王义和同志作了详细审阅，提出了许多宝贵意见。在此，向给予多方支持的同志们，谨致谢意。

由于水平所限，经验不足，书中错误和不妥之处在所难免，敬请读者批评指正。

编 者 一九八六年六月

绪 论

自世界上第一台计算机“ENIAC”(electronic numerical intergrator and computer电子数值积分计算机)诞生以后,在短短的四十年的时间里,计算技术迅猛发展,取得了卓越的成就,不仅形成了一个庞大的工业部门,而且形成了一门新兴学科——计算机科学。

从第一台计算机问世到现在,计算机已经发展了四代产品,如表1所示:

表1 计算机系统的发展阶段及其特征

代	特征	基本电路	存贮器	速度级和 代表机器	软 件	用 途
第一 代 1946~1954		电子管	延迟线 磁 鼓	千 次 ENIAC和 IBM704	符号语言	科学计算
第二 代 1955~1964		晶体管	磁 芯	10 万 次 CDC6600 IBM7090	编译系统	科学计算 数据处理
第三 代 1965~1972		集成电路	磁 芯 磁 盘	百 万 次 IBM360	操作系统	科学计算 数据处理 实时控制
第四 代 1973~		大规模 集成电路	磁 芯 MOS	亿 次 ILLIACIV	计算机网 络、数据 库	综合 信息处理

第一代计算机,主要采用电子管线路作基本逻辑电路。当时用的软件,主要是符号语言(汇编语言),计算机主要用于科学计算。第二代计算机,主要采用晶体管结构作基本逻辑电路。采用的软件,主要是管理程序和编译系统,计算机除主要用于科学计算以外,还从事少量的数据处理。第三代计算机,主要采用集成电路代替分立晶体管线路。采用的软件,主要是增加了管理系统资源的操作系统。计算机的应用领域,已扩展到实时控制。第四代计算机,主要采用大规模集成电路。这样体积大大缩小、功能大大提高了,已采用半导体MOS作为内存储器。采用的软件主要是增加了计算机网络和数据库。计算机用途更为广泛,几乎一切领域都能使用。第五代计算机正在研制(日本、美国),预计不久的将来即将问世。总之,计算机正在向、巨、微、网、智方向发展。

计算机能如此迅猛地发展,主要是由其特点所决定的。

电子计算机的运算速度快(国外的巨型机已达到几亿次每秒)精度高,具有逻辑判断和记忆能力,并且工作是自动完成的。计算机的应用领域可大体归纳出以下几个方面:

①科学计算。

纯数值计算，如导弹轨迹计算、桥梁结构计算，水坝应力计算、强度计算等。应用于这方面的计算机占8%左右。

②数据处理和信息加工。

要计算的公式并不复杂，但处理的数据量大，如银行业务、商业往来帐目、企业管理中的报表统计分析、成本核算等等。用于这方面的计算机占80%左右。

③自动控制系统，工业生产自动控制。如化工部门，实现了合成氨生产过程44个回路的闭环控制，水泥烧成配料的生产过程控制等等。用于这方面的计算机占12%左右。

④计算机辅助设计与辅助制造。利用计算机帮助人们进行产品设计，如设计飞机、轮船等等。

⑤利用计算机模拟人的智能、制造智能机器人、进行声音识别、图象识别及处理等等。

近十多年来，随着计算机系统的迅速发展和应用范围的日益广泛，软件的规模也越来越大。从七十年代开始，国外建立了软件产业，使软件生产达到了“系列化、产品化、工程化、标准化”。软件的主要任务是充分提高机器效率、发挥和扩大计算机的功能和用途，合理地、有效地使用计算机。软件的功能是由程序来完成的，编制各种程序必须掌握程序设计语言，本书介绍的，就是算法语言中的一种——FORTRAN。

§1 电子计算机的结构

电子计算机按结构分，可分为电子数字计算机和电子模拟计算机；按用途分，可分为通用计算机和专用计算机。不管哪一种计算机其结构都具有统一的模式，即Von Neumann式的计算机结构。这种结构的计算机，主要由硬件和软件两大部分组成。

1.1 硬 件

硬件即：组成计算机的物理设备，它可分为以下五大部分：运算器、控制器、存储器、输入设备和输出设备。它们之间的关系如图1所示：

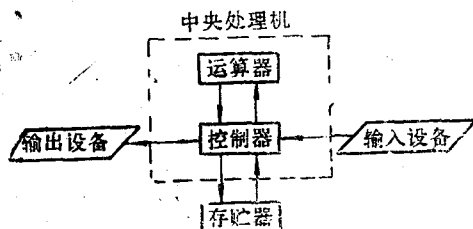


图1 电子计算机简易结构图

中间保存数据和程序。内存有磁芯存储器、MOS电路组成的随机存取存储器RAM、只读存储器ROM和可编程的只读存储器PROM。另一类称为外存储器（外存），它的主要特征是存储容量大，专用于长期或永久地存储数据和程序。如磁带、磁鼓和磁盘

中央处理机CPU(Central Processing Unit)是计算机的控制中心。它由两部分组成：控制器和运算器。控制器指挥计算机各部分自动、协调地进行工作；运算器在控制器的指挥下，对数据进行各种算术和逻辑运算。

存储器是存放数据和程序的地方，存储器按其作用分为两类：一类称为内存储器（内存），其主要特征是存取速度快，专用于运算

等。

计算机除CPU和存储器（内存）外，其余部分统称为外部设备，是用于支持CPU正常工作的辅助设备。如打印机、读卡机、光电机、磁鼓、磁盘等。

1.2 软件

计算机的另一组成部分是软件。软件是更好地发挥计算机效率的程序和程序系统。软件按用途分，可分为系统软件和应用软件。系统软件又可分为：面向用户的系统软件，如编译系统、汇编程序和调试程序等；面向管理人员的系统软件，如诊断程序。面向计算机本身的系统软件，如计算机的操作系统。应用软件是解决某些问题的程序。归纳如图2所示。

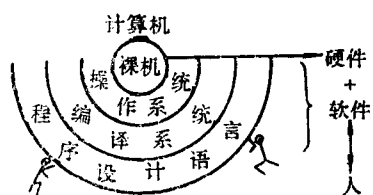


图2 计算机系统

§2 程序设计语言简介

2.1 什么是计算机语言

语言是人们交流思想的工具。在人类社会，人们为了交流思想，出现了各种各样的语言，如汉语、英语、日语、德语、……。人们通过这些“自然语言”相互“交流信息”。其实，人使用计算机进行解题的过程，也可以看成是人与计算机相互“交流信息”的过程。也就是说，人们通过编制的计算程序，去命令计算机怎样做，做什么，并命令计算机怎样把结果告诉人们。人与计算机之间的这种“交流信息”的过程，是通过执行计算程序中的一条条“指令代码”实现的。指令代码就是人与计算机交流信息的语言。由于这种语言是计算机所特有的，所以称之为计算机语言。

随着计算机的发展，计算机语言经历了从低级到高级的发展过程。

2.2 机器语言到算法语言

在使用计算机的最初年代里，程序是手工编制的。所谓手工编制程序，就是程序员直接写出能在机器上执行的、由一系列机器指令组成的程序。这样的由机器代码组成的语言，称为机器语言。用机器语言所编写的程序，称为机器语言程序，或手编程序。当然，不同的机器具有不同的指令系统，如DJS-130机的一条指令由16位二进制代码组成。如0010100000011110

每一位上的0或1，决定了计算机控制和运算线路的唯一状态。此指令的第0位的0表示访内指令，第1、2位的01表示取数操作，第3、4位的01表示参加运算的数据放在累加器AC₁中，第5、6、7位的000表示直接地址访问，第8~15位的00011110表示由内存地址36（八进制数）中取数参加运算，故这条指令的含义是：由内存单元（36）中取数，存入机器的累加器AC₁中。

用户在这样的计算机上解题时，不得不用极其繁琐的方法，把解题的算法编写成机

器语言的程序。如公式 $y = 3 + 5$ 的机器语言程序如下:

单元号	机器语言表示	意 义
00060	000003	整数 3
00061	000005	整数 5
00062	000000	结果单元 Y
00063	024060	取数 3 送入 AC_1
00064	030061	取数 5 送入 AC_2
00065	133000	$(AC_1) + (AC_2) \Rightarrow AC_2$
00066	050062	结果送入 Y 中

这种繁琐的编制程序工作,既费时间又容易出错,是很伤脑筋的。程序员要自己处理存储分配和输入输出。所以写出的程序直观性差,不可读。

机器语言程序在裸机上的解题流程如图 3 所示。将手编程序作为目标程序输入计算机,计算机执行程序就产生出结果数据。

为了简化机器语言程序, Wilkes 等在 50 年代初,引进了子程序、伪指令、操作码、地址码、十进制数表示、相对寻址功能及十二转换的概念。从而在 IBM650 计算机上产生了

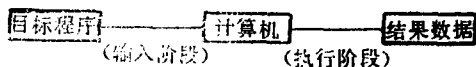


图 3 机器语言程序上机流程

第一个汇编程序。当时,汇编程序是一个最复杂、最精巧的程序,它在机器语言的基础上,提供一些容易记忆的助记符来代替原来的代码,利用计算机的基本操作,将特定的伪指令转换成相应的机器指令。如 DJS-130 机的一条访内指令的符号形式为:

LDA 1 , 60

LDA 表示取数操作码,位于 LDA 后的第一个数字 1,表示累加器 AC_1 , 数字 60 表示内存的单元号,这里的逗号或空格为各项间的分隔符。

用户使用这种软件工具——汇编语言来编写解题程序,称为汇编语言的源程序。如公式 $y = 3 + 5$ 的汇编语言的源程序表示为:

标号	操作码	操作数	意 义
A:	3		
B:	5		
Y:	0		
	LDA	1, A	取标号 A 中内容 3 送 AC_1
	LDA	2, B	取标号 B 中内容 5 送 AC_2
	ADD	1, 2	$(AC_1) + (AC_2) \Rightarrow AC_2$
	STA	2, Y	将结果 $(AC_2) \Rightarrow Y$

这时上机流程如图 4 所示。用汇编语言编写程序,仍要求用户对所用的计算机的逻辑结构及其性能完全了解。所编写的符号指令仍与机器指令一一对应。所以用它编程序仍是一件不容易的事。

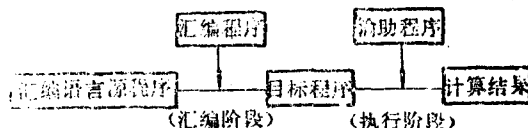


图 4 汇编程序上机流程

第二代计算机问世后,计算机的运算

速度和处理能力不断提高,计算机的高速度和准备程序的低效率之间的矛盾越来越突出。1955年前后,在美国出现了程序设计语言发展中的第一个最重要的里程碑——FORTRAN语言。这种语言吸收了机器语言和数学语言的优点,编出的程序具有很强的直观性。上述公式 $y = 3 + 5$,用FORTRAN语言来编写其程序就极为简单,用赋值语句可直接写出如下:

$$Y = 3 + 5$$

FORTRAN语言的成功实现,证明了高级程序设计语言是可行的和有生命力的。它引进了许多新的语言概念,如变量、表达式、数组、语句、输入输出格式等。用户编写程序时,再也不依赖于机器逻辑结构了。至今,FORTRAN仍是应用最广泛的程序设计语言之一。

六十年代是程序设计语言发展的兴旺时代,用于各种目的的程序设计语言层出不穷。如今世界上已有数百种语言了。这些语言从应用范围来划分,可分为三类:第一类是科学计算语言,如FORTRAN、ALGOL、BASIC、PL/1等。第二类是数据处理语言,如COBOL、LISP、APL、SNOBOL语言等。第三类是实时控制语言,如PL/1和实时FORTRAN等。这些语言按应用方法来划分,可分为两类:一类称为交互式语言,即用户和计算机彼此构成交互的双方,使程序的编写、验证、修改和运行交错在一起,如BASIC、APL、LISP语言等;另一类是非交互式语言,如ALGOL、FORTRAN语言等。用高级语言编写的程序,在计算机中的工作流程如图5所示:

用高级语言编写的程序,不依赖于具体机器,与数学公式非常相近,所以算题人员不需了解计算机具体结构,只要掌握高级语言就可以使用计算机来解题了。这给计算机的推广、应用带来了广泛的前景

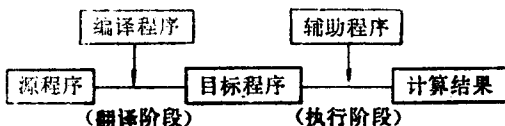


图5 有编译程序的计算机工作流程图

§3 怎样学习算法语言

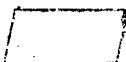
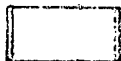
3.1 为什么要学习算法语言

目前,世界上大、中、小型计算机已有数百万台,广泛地应用于社会各个领域,推动了科学技术、经济和军事技术的发展,成为现代社会不可缺少的强有力的科学工具。所以,计算机技术水平、生产规模和应用程度,不仅是衡量一个国家现代化水平的重要标志,而且也是衡量任何一门自然科学的学科水平的重要标志。如今,计算机相当于工科大学生的计算尺,文科大学生的笔,理科大学生的推导工具,它不仅能帮助人们快速计算、方案设计和方案择优,而且还能用来写作、编辑和印刷。所以,现代科学家、发明家离不开计算机,即使是一般工作人员,也离不开计算机。人们与计算机打交道的最直接的介面,就是程序设计语言。因此,程序设计语言已成为高等学校中的一门必修课。

3.2 怎样才能学好算法语言

初学程序设计语言的人，会感到它错综复杂，似乎难于下手。其实，只要我们掌握学习程序设计的方法、工具和要点，就能学会程序设计。所谓方法，就是指解题的算法，必须做到“全局在胸”，即在编写计算机程序时，必须把一个问题分解成一个个简单的步骤，将各个步骤排好顺序，并仔细考虑到可能发生的每一种情况，然后，用框图的形式把“全局”描绘下来。为此，要遵循描绘框图的规则。绘制框图的流程图符号见表2所示：

表2 流程图符号

	处理框 操作数据的说明		开始、结束框程序中开始、结束 或程序中断点的说明
	输入输出框机器输出数据或用户 输入数据的说明		连接符进入另一个流程图的入口 或出口点的说明
	判定框根据条件判定下一步分枝 的说明		流向框与框间加工步骤流向的说 明
	循环框循环参数和控制变量变化 的说明		注解对框图功能的附加说明
	辅程序段框辅程序段段名、参数 调用的说明		屏幕显示终端显示说明

例如，用椭圆框来表示框图（或程序）的开始和结束。在框图中，把各种简单的运算写在长方形框格里，把判断决策的条件写在菱形框里，有关输入输出的数据写在平行四边形框中，其它的一些约定待后面叙述。对描绘成的框图，应仔细核对，确保①步骤的完整性；②框图的通用性。

例如，某天下午，张生与李生决定：如果下午六点天不下雨，他们就骑自行车出去蹓跹；如果下雨，张生就到李生家去；如果李生的父亲在家，就能帮助他们在家中制作飞机模型，否则就看电视。要求绘出张生活动的程序框图，见图6所示。

学习程序设计语言必须要有工具，即计算机。能绘出正确的框图，还不能算学会了程序设计。还必须选择某一台计算机，用该机提供的语言编出程序，上机运行，这才是学习语言的捷径。因为验证程序正确性的最好方法是上机实践，只有在机器上通过的程序才算是正确的程序。最后，为了学好程序设计语言，必须掌握程序设计语言的要点，这就是对语言中每个句子的句法、语义和语用，要深刻理解、灵活运用。掌握了上

述方法、工具和要点，就不难学好程序设计语言了。

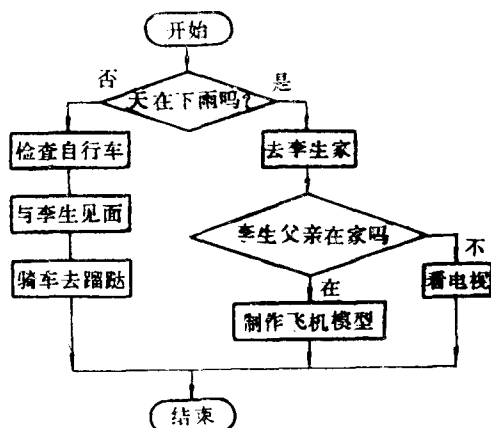


图6 张生活动程序框图

3.3 FORTRAN发展史和现状

FORTRAN一词来源于FORMula TRANslation，即“公式翻译”之意。第一个FORTRANⅡ语言的编译程序于1955年在IBM7040计算机上实现。1958年到1963年FORTRAN相继在许多计算机上实现。1962年初，FORTRANⅣ的编译程序开始出现了，并不断扩大使用。FORTRANⅣ不仅扩大了FORTRANⅡ的功能，而且作了一些改动，致使FORTRANⅡ的程序难于在FORTRANⅣ的编译程序下直接运行。因此，1962年5月美国标准协会（简称ASA）成立了有关工作组，开展FORTRAN语言的标准化工作。经过几年的讨论，于1966年3月正式公布了两个标准文本：美国标准基本FORTRAN和美国标准FORTRAN。美国标准基本FORTRAN是美国标准FORTRAN的一个子集。也就是说，这两个语言是向上兼容的。

由于FORTRAN在国际上广泛使用，国际标准化组织（简称ISO）于1972年7月发表了FORTRAN的推荐文本。ISO的FORTRAN分为三级：完全FORTRAN、中间FORTRAN和基本FORTRAN。其中，完全FORTRAN相当于美国标准FORTRAN，中间FORTRAN介于美国标准FORTRAN和美国标准基本FORTRAN之间，而基本FORTRAN相当于美国标准基本FORTRAN。1978年美国又根据国际上使用FORTRAN的情况，对FORTRAN重新发表了两个文本：美国国家标准程序设计语言FORTRAN（全集）和美国国家标准程序设计语言FORTRAN（子集），它们简称FORTRAN 77。这使FORTRAN语言的文本进入了一个新的发展阶段。

在科学计算中，FORTRAN是当今世界上最广泛流行的一种程序设计语言。它自问世以来，一直居于领先地位。在我国，1975年首次在DJS-8/6计算机上编制了FORTRAN编译程序后，许多机型上都配制了FORTRAN。目前，国际交流日益增多，许多科技题目的程序又多采用FORTRAN语言编写。为便于程序交流，本书着重介绍FORTRAN语言，且以标准FORTRAN为蓝本。

要强调的一点是，具体计算机上的FORTRAN语言，都是美国标准FORTRAN的一个扩充或限制。因此，当你使用此语言时，应参阅你所使用计算机的FORTRAN使用手册。

目 录

绪 论

§1 电子计算机的结构·····	2
§2 程序设计语言简介·····	3
§3 怎样学习算法语言·····	5

第一章 FORTRAN程序基础

§1 FORTRAN程序引例·····	1
§2 基本字符集与常数·····	5
§3 变量与符号名·····	9
§4 变量的类型·····	12
§5 基本函数·····	14
§6 FORTRAN语句分类·····	16
习 题·····	18

第二章 直接程序设计

§1 变量命名和定值语句·····	19
§2 进行计算语句·····	20
§3 输出结果语句·····	27
§4 停止计算语句·····	35
§5 READ(输入)语句·····	36
§6 举例·····	37
习 题·····	41

第三章 分枝程序设计

§1 单向分枝程序·····	45
§2 逻辑赋值语句·····	46
§3 双向分枝程序·····	50
§4 三向分枝程序·····	55
§5 多向分枝程序·····	60
§6 程序的模拟执行·····	69
习 题·····	70

第四章 循环程序设计

§1 数组和数组元素	74
§2 维数语句 (DIMENSION 语句)	77
§3 数组元素的排列顺序	78
§4 数组的输入输出	79
§5 循环程序设计	81
§6 (CONTINUE) 继续语句	88
§7 多重循环	90
习 题	101

第五章 过程程序设计

§1 语句函数	105
§2 函数过程	111
§3 子程序	118
§4 可调数组	124
§5 函数段与子程序段综合应用例题	129
§6 外部语句	132
§7 四种过程的对比	135
习 题	137

第六章 数据联系语句与数据块子程序

§1 共享逻辑单元的等价语句 (EQUIVALENCE)	142
§2 公用语句 (COMMON)	147
§3 等价语句与公用语句的联用	153
§4 数据置初值语句和数据块子程序	154
习 题	156

第七章 输入输出及格式语句的进一步描述

§1 几种输入输出语句	159
§2 格式语句	167
§3 FORMAT 语句中的字域说明符组	174
§4 格式语句与输入输出名表的关系	174
§5 输入输出语句与格式语句的应用	177
§6 格式数组	182
习 题	186

第八章 FORTRAN 程序的运行

§1 Z-80A 档微型机系统概貌介绍	189
---------------------	-----

§2 使用文本编辑程序上机的步骤与过程·····	190
§3 屏幕编辑程序的命令使用说明·····	195
附录一 文本编辑命令汇总·····	199
附录二 FORTRAN 编译程序错误信息·····	200
附录三 连接装配出错信息·····	202
附录四 FORTRAN IV 运行时错误信息·····	203
附录五 屏幕编辑命令一览表·····	204
附表一 FORTRAN IV 基本函数一览表·····	206
附表二 ASCII ——十六进制表·····	208
附表三 FORTRAN 77 语句一览表·····	209
参考文献·····	214

第一章 FORTRAN程序基础

§ 1 FORTRAN程序引例

用FORTRAN语言编写的程序，与用自然语言写出来的文章的基本形式类似。如文章分有各种体裁，每种体裁的文章书写时，都有一定的格式要求。然而FORTRAN语言是一种程序设计语言，用它编写的程序是让计算机执行的。因此，FORTRAN程序在体裁结构和书写形式上，都有一套固定的格式要求。下面通过阅读一个简单的程序引例，来介绍用FORTRAN语言编写程序的基础知识。

〔引例〕编写一个求一批数开平方的程序。

1.1 选择算法

怎样求一个数A的平方根（如 $x = \sqrt{A}$ ），在计算机上采用手算开平方的方法是不方便的，最好是选择一种基本步骤很简单，又可以多次重复的“逐步求精”的算法，这样，便可以编写出简单的程序来完成开平方的计算。

设 x_0 为平方根的近似值，如果 x_0 小于实际应有的值，那么 $x_0^2 < A$ ，则 $A/x_0 > x_0$ ；如果 x_0 大于实际应有的值，即 $x_0^2 > A$ ，则 $A/x_0 < x_0$ 。在这两种情况下，取 x_0 与 A/x_0 的平均值，都会得到比 x_0 好一些的近似值，记作 x_1 ：

$$x_1 = \frac{1}{2} (x_0 + A/x_0)$$

如果对 x_1 还不满意，可以再如上法求得：

$$x_2 = \frac{1}{2} (x_1 + A/x_1)$$

如此多次重复地做下去，求得：

$$x_{n+1} = \frac{1}{2} (x_n + A/x_n) \quad n = 1, 2, \dots$$

这种求平方根的方法叫迭代法或逐步求精法，早在两千年前，希腊数学家Heronus的著作已描述过。迭代到什么时候为止呢？上述“逐步求精”的过程重复多次之后， x_{n+1} 和 x_n 的差别就会很小，或者说，只要

$$|x_{n+1} - x_n| < \epsilon$$

这时 x_n 和 x_{n+1} 的前若干位可认为是相同的，则应该结束计算。这里的 ϵ 是一个恰当选择的小实数。

1.2 绘制框图

为确保框图的通用性，我们不限定被开方数的个数。要求用户在给出这批数的末尾，专门给一个数0.0，以表示计算结束。在计算中计算机不能违反数学中的规则，即在实数范围内，负数不能开平方。计算机对每个数开平方后，需输出三个值，即该数的序号、该数和该数的平方根。当被开方数出现负数时，计算机输出该数的序号和此负数，表示此数不能开平方。这种思想可概括为如下的步骤：

- ① 读入一个数并记序号；
- ② 判定此数：若为0.0，则停止计算；若为负数则输出序号和此负数，且转向步骤①，若此数大于0.0，则转入步骤③；

③ 按“逐步求精”法求平方根；

④ 输出该数的序号、该数和该数的平方根，且转向步骤①。

上述步骤的通用框图如图1-1所示：

1.3 程序及其解释

FORTRAN源程序和数据，书写在FORTRAN程序纸上，见图1-2所示。程序纸上的每一行被分为80列（格），源程序或数据从左往右书写，每一列只能写一个字符。源程序由若干行组成，每行分三个部分：第一部分称为语句标号区，从第一列到第五列，用于写语句标号；第二部分称为续行区，为第六列，用于标识语句的续行；第三部分称为语句区，从第七列至七十二列，用于书写FORTRAN语句。第七十三列至第八十列可作为语句序号之用。源程序中的行，有下列几类：起始行、继续行、注解行和结束行。起始行是语句的第一行，要求第六列必须为空白。一个语句可以分多行写完，除语句的第一行外，都为继续行，继续行要求第六列必须为除“0”外的数字和符号。注解行是程序的说明，增加程序的可阅读性，要求第一列上必须是字母“C”。结束行表示程序段结束，要求在第七列以后必须是字母END。

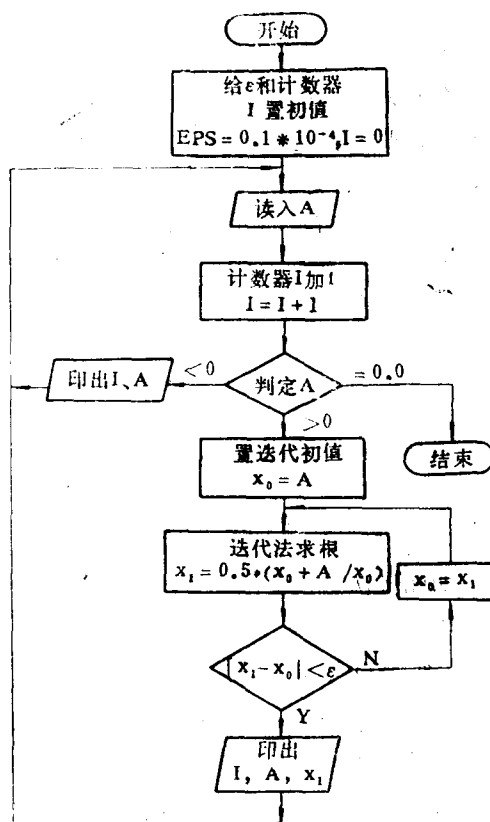


图 1-1 成批数开平方框图