

数模与电算基础

(化学化工专用)

上 册

程 光 钺 编 著

一 九 八 〇 年 四 月

前 言

电子计算机的出现和发展在化学化工领域内引起了一场深刻的革命，它促使这一领域的面貌发生巨大的变革，而这种变革又在不断加速地进行着。它也迫使在这一领域内工作的科技人员去接触那些本来并不熟悉的众多问题，而数学模型方法就是其中较突出的一个。

依靠电子计算机在化学化工领域内建立数学模型主要牵涉以下四方面的内容。首先是化学化工本身。这当然是最基本的，是数学模型的物理——化学基础。第二是概率与数理统计。这是因为建立数学模型，特别是比较复杂体系的模型，不可避免地要对大量具有随机性的实验资料进行归纳，因此研究随机性事件的概率论与数理统计是必不可少的基本工具。第三是计算方法。第四是程序设计。这两方面的内容都是与电子计算机的应用直接有关的。因为在建立数学模型的过程中总会遇到极大量的计算工作，这些工作通常是无法用人力或通过机械计算机以必要的精度在必要的时间内完成的。而应用电子计算机，首先要掌握的基础知识就是程序设计与计算方法。

在这里我们主要从化学和化工工作者的实用角度出发，介绍后三方面的基本原理与方法，并在此基础上讨论数学模型的建立和与此有关的问题。

这三方面的内容是属于数学的三个分支学科——概率论与数理统计、数值分析和电子计算机程序设计，因此相应的内容也散见于这三个数学分支学科的著述之中。因为数学著述就其内容的全面与系统以及论述之精密与严格而言，常常远超出一般化学化工工作者的实用要求。所以面对这一现实，我们常有为制作耗料不多的木器却不得不在大片丛林中挑选少数几种木材的类似感觉。另外，多数数学著述对非数学专业的科技人员讲来，常感其文字过于精炼与艰涩，难于阅读。有鉴于此，在这里我们将试图集中那些为在化学化工领域内依靠电子计算机建立数学模型所必不可少的基础知识，并尽可能通过化学实例加以阐明与解释。同时在叙述上也在不破坏必要的严谨性的前提下，尽量用化学化工工作者易于接受的语言与表达方式。

本讲义是笔者为化工部西南化工研究院讲授“数模与电算基础”课程而编写的，分上下两册。上册介绍电子计算机应用的基础知识，由两篇组成。第一篇（包括第一章至第四章）以ALGOL—60语言为中心介绍程序设计和程序设计语言。第二篇（包括第五章至第十章）介绍电子计算机常用的基本算法与程序。下册也是由两篇组成。其中第三篇（包括第十一章至第十五章）介绍概率论与数理统计基础。第四篇（包括第十六章至第二十）介绍数学模型的建立方法。各部分虽彼此衔接但仍有一定程度的独立性。

在讲义编写过程中北京大学化学系徐光宪教授在题材组织、选择等方面曾给予指导。四川大学数学系敖硕昌教授和向重伦老师分别审阅了手稿的第三、四篇和第一、二篇。段世清、谭立业、陈一权、党秀文等同志以及川大化学系物化教研室很多其他同志都给予了热情的鼓励和帮助。笔者在此一并表示感谢。

0242

目 录

—— 上 册 ——

前 言

第一篇 电子计算机的程序设计

第一章 电子计算机与程序设计	(1)
第一节 电子计算机简介	(1)
第二节 用电子计算机解题的一般步骤	(3)
第三节 程序设计	(5)
第四节 程序设计语言	(5)
第二章 ALGOL-60语言	(6)
第一节 基本符号	(9)
第二节 数、逻辑值、变量、标准函数	(10)
§ 2 · 1 数	(10)
§ 2 · 2 逻辑值	(11)
§ 2 · 3 简单变量和类型说明	(12)
§ 2 · 4 下标变量和数组说明	(12)
§ 2 · 5 标准函数	(14)
第三节 表达式	(15)
§ 3 · 1 算术初等量和简单算术表达式	(16)
§ 3 · 2 关系式、布尔初等量和简单布尔表达式	(17)
§ 3 · 3 条件算术表达式	(19)
§ 3 · 4 条件布尔表达式	(22)
第四节 语句	(24)
§ 4 · 1 赋值语句	(24)
§ 4 · 2 输入语句	(26)
§ 4 · 3 输出语句	(28)
§ 4 · 4 转向语句	(32)
§ 4 · 5 条件语句	(33)
§ 4 · 6 复合语句与分程序	(36)
§ 4 · 7 空语句	(41)
§ 4 · 8 循环语句	(42)
第五节 开关与过程	(52)
§ 5 · 1 开关	(52)

§ 5 · 2	命名表达式	(56)
§ 5 · 3	过程的一般介绍	(58)
§ 5 · 4	过程说明	(62)
§ 5 · 5	过程语句	(67)
§ 5 · 6	函数过程	(70)
§ 5 · 7	标准过程	(72)
第三章	FORTRAN 语言	(72)
第一节	基本组成部分	(72)
§ 1 · 1	字符集	(72)
§ 1 · 2	数	(73)
§ 1 · 3	变量与类型说明	(73)
§ 1 · 4	数组与数组说明	(73)
§ 1 · 5	标准函数	(74)
§ 1 · 6	表达式	(74)
第二节	语句	(74)
§ 2 · 1	赋值语句	(74)
§ 2 · 2	转向语句	(75)
§ 2 · 3	条件语句	(75)
§ 2 · 4	循环语句	(76)
§ 2 · 5	空语句	(76)
§ 2 · 6	输入输出语句与格式语句	(77)
§ 2 · 7	简单FORTRAN程序的一个实例	(77)
第三节	子程序与函数	(78)
§ 3 · 1	子程序	(78)
§ 3 · 2	函数	(79)
§ 3 · 3	语句函数	(80)
第四节	一些其他问题	(81)
§ 4 · 1	FORTRAN程序的结构	(81)
§ 4 · 2	公共语句	(82)
§ 4 · 3	等价语句	(82)
§ 4 · 4	穿孔卡与程序纸	(83)
§ 4 · 5	FORTRAN 语言的各种版本	(83)
第四章	语言翻译	(83)
第一节	问题的提出	(83)
第二节	解决的途径	(84)
第三节	FORTRAN—ALGOL 翻译实例	(85)
参考文献		(86)

第二篇 电子计算机常用的基本算法与程序

第五章	计算方法与程序的质量要求	(87)
------------	---------------------	--------

第一节 计算量	(88)
第二节 存贮量	(90)
第三节 误差	(91)
第四节 程序结构	(94)
第六章 线性代数方程组	(95)
第一节 简单消去法	(95)
§ 1 · 1 二元线性方程组	(95)
§ 1 · 2 n元线性方程组	(97)
§ 1 · 3 在计算机上的实现——组合法介绍	(99)
§ 1 · 4 程序与框图	(101)
§ 1 · 5 实例——二氯甲烷蒸气压计算	(103)
第二节 主元消去法	(104)
§ 2 · 1 方法梗概	(105)
§ 2 · 2 程序与框图	(105)
§ 2 · 3 对程序的分析——追踪法介绍	(107)
§ 2 · 4 实例——乙炔的克分子热容—温度关系	(114)
第七章 行列式与矩阵	(116)
第一节 行列式	(116)
§ 1 · 1 行列式的定义	(116)
§ 1 · 2 行列式的基本性质	(118)
§ 1 · 3 Cramer (克莱姆) 规则	(118)
§ 1 · 4 行列式值的计算	(119)
§ 1 · 5 计算程序与框图	(122)
§ 1 · 6 实例——方程组相容性检验	(124)
第二节 矩阵	(125)
§ 2 · 1 矩阵的定义	(125)
§ 2 · 2 几种特殊形式的矩阵	(126)
§ 2 · 3 矩阵的加减与数乘及其计算程序	(127)
§ 2 · 4 矩阵的乘积及其计算程序	(128)
§ 2 · 5 矩阵的转置及其计算程序	(131)
§ 2 · 6 矩阵的逆及其计算程序	(135)
§ 2 · 7 实例——不平衡Wheatstone (惠斯顿) 电桥分析	(144)
§ 2 · 8 矩阵的秩及其计算程序	(145)
§ 2 · 9 实例——化学体系的独立反应数	(149)
第八章 非线性方程与方程组	(151)
第一节 迈步法	(151)
§ 1 · 1 方法梗概	(151)
§ 1 · 2 程序与框图	(152)
第二节 二分法和优选法	(152)

§ 2 · 1	二分法	(152)
§ 2 · 2	优选法	(154)
§ 2 · 3	实例——甲醇合成反应的化学平衡	(158)
第三节	下降法	(160)
§ 3 · 1	方法梗概	(160)
§ 3 · 2	程序与框图	(163)
§ 3 · 3	实例——正戊烷的沸点—压力关系式	(164)
第九章	插值、微分与积分	(166)
第一节	插值	(166)
§ 1 · 1	方法梗概	(166)
§ 1 · 2	Lagrange (拉格朗日) 一元插值程序	(168)
§ 1 · 3	实例——实验数据的内插	(169)
第二节	数值微分	(170)
第三节	数值积分	(171)
§ 3 · 1	方法梗概	(171)
§ 3 · 2	Simpson (辛普森) 法程序	(173)
§ 3 · 3	实例——反应热效应的计算	(173)
第十章	常微分方程与方程组	(176)
第一节	Euler (尤拉) 法	(176)
§ 1 · 1	方法梗概	(176)
§ 1 · 2	程序与框图	(180)
§ 1 · 3	实例——一级化学反应动力学	(183)
第二节	Runge—Kutta (龙格—库塔) 法	(184)
§ 2 · 1	方法梗概	(184)
§ 2 · 2	计算程序	(188)
§ 2 · 3	实例——连续反应动力学	(188)
参考文献		(190)

第一篇 电子计算机的程序设计

第一章 电子计算机与程序设计

第一节 电子计算机简介

电子计算机是一种能够自动和高速进行大量计算工作的电子设备，它的诞生和发展是当代重大的科学成就之一。

一九四六年世界上出现了第一台电子计算机，这台计算机共用了一万八千多个电子管，消耗近150瓩的电力，但由于采用了电子管，其运算速度与以往的计算工具相比有了惊人的提高。随着电子技术的迅速发展以及卓有成效的应用实践，在短短的三十多年的时间里，不但电子计算机本身有了极大的改进与发展，同时它也促使现代科学技术发生巨大的变化。目前电子计算机的应用已遍及科学研究、工农业生产、军事、商业、交通直至行政管理、文化教育、医药卫生等等各种不同领域，并渗入到日常生活中去。

电子计算机可分为三类：电子数字计算机、电子模拟计算机和电子数字——模拟混合计算机。电子数字计算机以数字形式的量值在机器内部进行运算。电子模拟计算机是用连续变化的物理量来表示被运算的变量以进行计算。电子数字——模拟混合计算机是将两者结合起来的计算机。电子数字计算机的精度高、使用方便灵活，因此目前广泛使用的（特别是进行数据加工处理用的）电子计算机大多属于此类；在以下的讨论中，除特别申明外，谈到电子计算机时，所指的都是电子数字计算机。

电子计算机的发展一般按使用的元器件划分为四代。第一代是采用电子管的计算机，其时间大约是计算机问世后的前十年。1958年左右出现采用晶体管的第二代计算机。1965年左右开始采用的集成电路的计算机称为第三代计算机。七十年代又出现了第四代计算机，即采用大规模集成电路的计算机。

从广泛的意义上讲来，电子计算机系统是由两大类部件所组成：硬件和软件。硬件是电子计算机系统中实际装置（主要是机械装置、电子装置等等）的总称，主要包括：

一、主机：通常把运算器、控制器和内存贮器称为主机。

主机部分是构成电子计算机的主要装置。我们经常把使用电子计算机进行的计算工作通俗地比做打算盘，因此首先要有一个相当于算盘的部件。这一部件就是运算器。运算器能快速地对数据进行算术和逻辑运算。用电子计算机解决复杂的实际问题时，总是预先通过一定方法将它转化为一系列简单的算术或逻辑运算问题，在运算器里进行运算。

通过打算盘算题时如果用的原始数据较多或计算分若干步进行，那么往往就要借助于纸把原始数据先写出来，把中间的运算结果记下来等等，以便顺利无误地进行计算。电子计算机中相当于纸的这个部分就是存贮器。存贮器能存放大量的数据和指令。数据就是参加运算

的原始数据或运算的中间或最后结果，而指令简单说来就是通知机器每一步应做什么运算的命令。这些指令是使用计算机的人预先安排好的。由这一系列指令所构成的完整的运算步骤就称为程序。因此在存储器中所存放的是数据和程序。内存贮器是包括在主机之内的存储器，它与运算器有直接的联系。一方面它不断地向运算器提供运算所需要的数据，另一方面它又能保存从运算器送出的计算结果。

电子计算机还必须具有相当于人在打算盘时起指挥作用的部分，这部分称为控制器。控制器能按照程序自动控制计算机各部分连续而协调地工作。它的任务是按照程序给出的顺序一步步地分析每一条指令，并命令计算机的各个部件执行这些指令所规定的基本操作。

二、外部设备：通常把外存储器、输入、输出设备等称为外部设备。

外存储器是在主机以外的存储器，它与运算器没有直接的联系。它一般有较大的存储量并可与内存贮器进行信息交换。当进行具有大量数据或指令的计算工作时，内存贮器不够用就可以利用外存储器存放计算中暂时不用的数据或指令。外存储器主要有磁鼓、磁带、磁盘三种。它们的工作原理就和磁带录音机的磁带相似。它们与通常用做内存贮器的磁芯存储器和半导体存储器相比，存取信息的速度要慢些，但其存储量大，价格低。由于它们是在主机之外，可以更换，因此计算机在长期运算中所积累的有价值资料都可保存在磁带、磁盘上。当再度需要使用时，就可以把相应的磁带或磁盘重新装入计算机调用。所以对于一个配有外存储器的电子计算机，其外存容量实际上是无限限制的。近来又发展了一种柔软可弯曲的塑料磁盘称为软盘，其体积小而存储量大，可靠性高，可随身携带与邮寄，使用方便。

输入输出装置是电子计算机与外界联系的部件，这种联系包括计算机与人，计算机与其他计算机或计算机与其他设备。输入装置的职能是将外界的信息送入计算机。目前最典型的方法是首先将信息转化为数，然后再以数据形式输入。这些数一般是打在穿孔纸带或穿孔卡片上，通过光电纸带输入机或卡片输入机将光信号转换为电信号并为计算机所接受。输入还可采用其他方式如通过磁盘或磁带等直接输入，通过控制台按键或电传打字机输入等。目前又出现一些更先进的输入装置，它们可以把文字、图象，以至人的语言直接向计算机输入。比如磁墨水读出器（简称MICR，可对用磁性物质做墨水所书写的文件进行扫描、读出并转换成电信号向计算机输入）、光学字符读出器（简称OCR，利用写有文字的纸面上各点光的强弱不同而将它转换成为电信号、输入并识别）等等。这样人们就可以用打字的甚至手写的文件直接向计算机输入。更方便的输入装置是声音输入装置（简称AID），它可以通过讲话对计算机发出指令。简略地讲来其原理是：当声音发出后，由一组滤波器对它进行频谱分析，每隔一定时间将这些分析的结果与存储于计算机中的“标准”进行比较从而实现识别并输入。目前这些新型的输入装置还在研究与发展阶段。

输出装置的任务是把计算机的计算结果以各种形式进行输出：（1）以人们所能识别的形式做为最后结果而输出；（2）输出某种信息以控制其他设备工作；（3）送到外存储器暂存（如输出至磁带、磁盘）。作为最后结果而输出时，一般的输出形式有数字、符号、字母、图象、声音等。目前常见的输出装置是快速打印机（主要输出数字），宽行打印机和电传打字机（输出数字、符号、字母），快穿机（输出穿孔纸带），绘图机或曲线记录仪（输出曲线图形）等等。这些输出设备都是以打印文件的形式将信息输出的。目前还有一些以其他形式输出的装置，如声音应答装置，图形字符显示器等。例如图形字符显示器可通过阴极射线管显示、激光显示等形式将计算机处理的结果以数字、文字、曲线或其他图象的形式显示出来；它

可进行大屏幕的彩色或立体显示也可以与光笔相结合又将新的信息向计算机输入。光笔是一种手执的如铅笔状的部件,可用光笔在显示屏上对输出的符号、文字、图象等进行修改、补充,然后经修改补充后的信息又可以直接重新输入计算机。

三、外围设备:通常把主机与外部设备以外的其他的硬件称为外围设备。如模一数、数一模转换设备、远距离终端设备等等。随着外部设备的发展与多样化,外围设备与外部设备的界限逐渐不清。因此目前不少人已将外部设备与外围设备不做严格区分。由于一般外围设备与我们即将介绍的内容关系不大,所以在此不再详述。

电子计算机系统的另一个组成部分是软件,软件又称为程序系统,指的是事先编好的整套程序。它不同于普通的算题程序,其着眼点在于扩大和提高计算机的效率。简化或代替在各环节上人所承担的部分工作。事实表明,软件的功能与质量在很大程度上左右整个计算机系统的效能。目前软件建设已越来越引起人们的重视,一些工业国家软件研制经费已占整个计算机研制经费的一半以上,从事软、硬件工作的人员之比已达三比一。软件按其作用分成系统软件和应用软件,系统软件的目的在于为计算机提供某种能力,或执行某种特定的功能(如操作系统、编译程序、检查诊断程序、程序设计语言等)。应用软件是指为解决某类问题而编制的规模较大且功能广泛的程序。在计算机发展的早期,人们针对各应用领域中不同的具体问题编制不同的程序。目前已从这个状况逐步过渡到针对某一领域,研制能解决该领域内一部分或大部分问题的总的程序,或称程序包。每一个具体问题可通过使用此程序包的一部分而求解。这种程序包即称为应用软件。

目前还发展了所谓的固件,即具有软件功能的硬件。随着电子计算机的发展,出现一种使软件与硬件相“融合”的趋势,即所谓“软件硬化”。固件是这一种趋势的产物,目前固件仍处于研制阶段。

第二节 用电子计算机解题的一般步骤

使用电子计算机解决实际问题,一般须经以下几个步骤:

1 建立数学模型

很多实际的科学技术问题都可以在经过一定的简化之后用数学语言,也就是说用一定的数学关系式来描述。如果我们的简化是一种合理的简化,对各种影响因素进行了去粗取精、取主舍次的分析,那么所抽象出的描述问题的数学关系式就称为此问题的数学模型。

一般说来电子计算机不是直接对于待解决的实际问题本身进行计算,而是对描述此问题的数学模型进行计算。

一个合理可靠的模型能合理而可靠地模拟它所描述的问题。相反,如果模型不可靠,那么计算出的结果就不能代表实际问题的解答。因此用电子计算机解决实际问题的第一步就是要建立一个合理可靠的数学模型。

2 选择计算方法

有了描述实际问题的数学模型,也就是说有了所描述的问题中各变量间的数学方程式和求解条件(如初始条件、边界条件等),下一个问题就是如何具体进行计算以求解。

一般说来如果所描述的是一个稍复杂的实际问题,那么能简便地通过推导而得到解析解

(即用数学表达式表示的解)的情况是较为少见的。在大多数的情况下,这种解析解并不存在,人们能够求得的只是用具体数值表示的解,即数值解。但通常这种数值解对于实际应用而言也已够用了。

在电子计算机问世以前,这种求数值解的工作(即所谓数值计算)是靠人工或机械计算器来进行的。其精度、速度都是很有限的。电子计算机的出现与发展,使得数值计算工作发生了巨大的变化。

用电子计算机进行数值计算,需要把在数学模型中出现的各种数学运算(如微分、积分、微分方程、代数方程等的求解)通过简单的算术运算和逻辑运算来进行。因此如何在保证精度高、计算量小、所需存贮单元少的条件下选择最适宜的计算方法就成为使用电子计算机进行计算中第二个十分重要的问题。

目前计算方法已是数学学科中的一个重要分支,并且随着电子计算机的发展,计算方法的发展也很迅速。一方面为了适应新的需要出现了许多新的方法,另一方面许多不适应电子计算机的古典方法被淘汰或改造。同时一些计算量很大,过去难于实现的方法,现在又得到了新生。

当前很多数学问题已有了多种相当成熟的计算方法供选用。

3 程序设计

有了数学模型并选择了适当的计算方法之后,就需要详细安排完整的解题计划与步骤,

并把它们以程序的形式描述出来,以便命令计算机按此计划与步骤进行计算。这一工作就是所谓的程序设计。它是本书前几章所要讨论的中心问题。

4 上机计算

将所编写出的程序,用电子计算机所能接受的方式(如穿孔纸带、穿孔卡片、电传打字等)输入到计算机中去。在进行计算后,由计算机输出计算结果。

5 分析结果

对计算机输出的结果进行分析,判断是否正确,如正确则计算结束,否则应进一步分析错误的性质,以便进行修改、重算直至正确为止。

综上所述,用电子计算机解决实际问题的步骤可用图 1—1 的示意图表示。

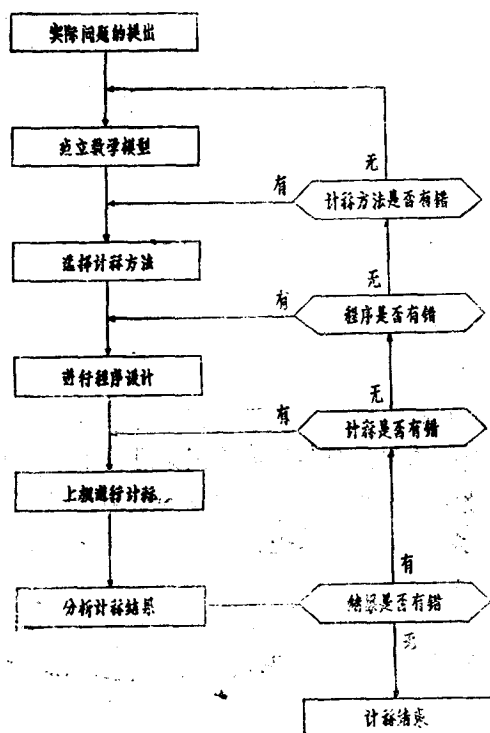


图 1—1 用电子计算机解题步骤示意图

第三节 程序设计

前面已讲过，用电子计算机解题时，人们必须事先详细安排完整的解题计划，并以程序的形式把解决问题的方法与步骤连同数据一起都送入计算机。这样，计算机才能按照人们的意图而运转，并计算出所需之结果。

由于计算机所能直接接受的是二进制数码，所以我们欲送入的任何信息都必须最终化为二进制的数码（又称代码）。另外，任何一个指定的计算机都只能进行指定的一些基本操作，也就是说只能进行指定的一些算术、逻辑或其他规定动作（比如进行加、减、乘、除运算，比较两个数据的大小，把数据自计算机中某一部分送至另一部分等等），所以从原则上讲来，程序设计归根结底要把我们的计算问题化为上述这些基本操作的组合，进而把它们通过二进制数码告诉计算机。

比如我们欲求为配制 $2.5\text{N H}_2\text{SO}_4$ 250 毫升需 $5.5\text{N H}_2\text{SO}_4$ 多少毫升。这个问题可化为求

$$x = 2.5 \times 250 / 5.5$$

为了进行这一计算，就应首先把2.5、250和5.5这三个数据存贮到存贮器的三个指定单元A、B、C（即地址分别为A、B、C的三个单元）中，同时又分配一个单元D来存放计算结果。以上这一工作叫做分配存贮单元。单元分配完毕之后就要编写程序。此时应十分详细地规定计算机对哪些数据执行哪些基本操作，以及这些操作的执行顺序。规定计算机的操作种类以及操作对象（一般为数值或地址）的命令，一般被称为机器指令。因此在这里所谓编写程序就是根据问题的要求。按一定的顺序写出机器指令。比如对本问题应写出：

- （1）取出A单元中的数据，并把它放入运算器（实际上是运算器中的某个寄存器）。
- （2）把运算器中的这个数乘以B单元中的数，所得之积仍保留在运算器中。
- （3）将运算器中的数除以C单元的数所得之结果仍保留在运算器中。
- （4）将运算器中的数放入D单元中，并把D单元中之数打印出来。

以上这些指令都必须通过代码传递给计算机。为此每种计算机有自己的一套所谓指令系统，在指令系统中规定该机器能执行那些指令和如何用代码来表示它们。而上述的程序就应根据指令系统的要求用代码表示，并经穿孔纸带或穿孔卡输入到计算机中去。

第四节 程序设计语言

上述这种用机器指令编写的程序称为手编程序。从上例可以看出这种手编程序有很大的缺点。首先是编写手编程序的工作量十分大。上例中这个简单的计算就要写出好几条指令。如是复杂的数学问题，其相应手编程序之庞大与繁复必可想而知。因此往往计算机能在几分钟或几秒钟算出的问题，却需要几天甚至几星期来编写其程序。其次这种手编程序是由一大堆数字（代码）所组成的，很不直观，不便于阅读和交流；并且在编写时很容易出错，出错后又不容易检查。最后，手编程序是用机器指令编写的，不同型号的计算机，其机器指令系统不同，所以针对某种型号计算机所编写的程序，完全不适用于另一种型号的计算机。

为了克服这些缺点，从五十年代中期开始，发展了另一种程序设计方法，即用所谓程序

设计语言来进行程序设计的方法。

所谓“语言”一般是指人们相互间交流思想的工具。如汉语、日语、英语等等，这种语言称为自然语言。但在计算机科学中所讲的“语言”，则指的是人和电子计算机间交换信息的工具。比如我们要指示计算机进行某种指定的操作，或需要计算机将计算的结果告知我们等，都需要通过计算机科学中的所谓“语言”来进行。前面讲的手编程序中的机器指令系统就是一种语言，称为机器语言。只不过这种语言依赖于机器，即对不同的电子计算机相应的语言彼此不同。

用于进行程序设计，并不依赖于机器的语言，称为高级语言。高级语言有多种。据统计，目前世界各国使用的这类程序设计语言不下数百种，而其中较通用的也有十余种。有些程序设计语言最适用于科学数值计算（如ALGOL—60^①，FORTRAN^②等）；有些适用于进行商用数据处理（如COBOL^③等）；有些适用于机床工具数字控制（如APT^④等）；有些适用于非数值的符号处理（如SNOBOL^⑤等）；有些适用于人机对话（如BASIC^⑥，FOCAL^⑦等）。另外，还有一些大型的所谓“可扩充型”或“汇集型”的语言（如ALGOL—68，PL/1^⑧等）在一定程度上兼收与并蓄上述各种语言的功能。以上所有这些语言都有一个公共的特点，就是其陈述规则与自然语言（比如英语）的表达方式相似，同时与一般陈述数学问题的方式也很相近。利用这类程序设计语言进行程序设计通常只要求我们按照该种语言的规则（语法规则）将要解决的问题或计算的内容写出来就可以了。这样写出来的程序叫做源程序。当源程序被输入计算机后，在大多数情况下计算机就通过一个所谓的编译程序，自动地将用程序设计语言所写出的源程序翻译成机器语言（用代码表示的机器指令）编写的程序。这后一程序称为目标程序。按照目标程序所规定的步骤，计算机即可进行运算。编译程序作为计算机软件的一部分，已事先存放在计算机中。使用者只需用程序设计语言编写程序并穿孔输入，计算机就会通过原已存入的编译程序进行翻译和识别并据此进行计算直至输出计算结果。

在以上所述的各种程序设计语言中广泛用于本书所涉及的数值计算的是ALGOL—60和FORTRAN。前者是我国目前应用最广泛的，后者是国际当前最流行的。除此以外，BASIC和FOCAL也常用于小规模的科学数值计算。

第二章 ALGOL — 60 语言

由于算法语言ALGOL—60是我国目前常用的程序设计语言之一，且适于描述一般科学

① ALGOL—60是Algorithmic Language—60（算法语言—60）的简称。

② FORTRAN是Formula Translator（公式翻译语言）的简称。

③ COBOL是Common Business Oriented Language（通用商业语言）的简称。

④ APT是Automatically Programmed Tools（自动程序设计工具）之简称

⑤ SNOBOL是String—Oriented Symbolic Language（行符号语言）之简称

⑥ BASIC是Beginners All—purpose Symbolic Instruction Code（初学者通用符号指令代码）之简称

⑦ FOCAL是Formula Calculator（公式计算语言）之简称

⑧ PL/1是Programming Language—1（程序设计语言—1）之简称

技术中的计算过程，因此在本书中将主要使用 ALGOL—60 语言

将 ALGOL—60 语言用于具体的计算机时，常根据计算机的具体结构与性能对原语言进行一些限制和扩充。经限制和扩充后的语言虽然有时取各种不同的名称，但它们都是属于 ALGOL—60 系统的，彼此大同小异。ALGOL—121 就是在 ALGOL—60 基础上针对 DJS—21 机（简称 121 机）经限制与扩充而得到的。下面我们将通过 ALGOL—121 来介绍算法语言 ALGOL—60。

在正式介绍以前，首先通过一个简单的例子对程序的编写与结构做一粗略的介绍。我们仍举在前面提到过的实例。

例2—1 求为配制 2.5N H_2SO_4 250ml，需要 5.5N H_2SO_4 多少毫升？

解： 设 N_1 ， N_2 分别为稀硫酸和浓硫酸的浓度； V_1 ， V_2 是它们的体积，则有

$$V_2 = \frac{N_1 V_1}{N_2} = \frac{2.5 \times 250}{5.5}$$

用 ALGOL—121 语言编写此问题的程序应包括以下内容：

- (1) 首先宣告程序开始。
- (2) 说明所用符号的意义。
- (3) 指明所要进行的运算。
- (4) 把结果打印出来。
- (5) 宣告程序结束。

与此计算问题相应的 ALGOL—121 语言源程序可写为：

程序 2—1

```
      Y
      'BEGIN'
      'REAL' N1, N2, V1, V2;           说明部分
      READK ( 3, N1, N2, V1 );         语句部分
      V2 := N1 * V1 / N2;
      PRINT ( V2 )
      'END'
      S
      2.5; 5.5; 250;                   数据区
```

现对以上程序简单解释如下：

Y 表示其后为所编之源程序。

S 表示其后为待输入之数据。

'BEGIN' 和 'END' 分别表示源程序开始与结束。

'REAL' N1, N2, V1, V2 是告之机器这四个符号是实型的变量。

READK (3, N1, N2, V1) 是输入三个数据并赋于 N1, N2 和 V1。

V2 := N1 * V1 / N2 表示先将 N1 乘以 V1 再除以 N2，并把所得之值赋于 V2。

PRINT (V2) 表示将 V2 打印出来。

2.5; 5.5; 250; 是三个待输入的数据，通过 READK (3, N1, N2, V1) 而被输入，并分别赋于 N1, N2, V1。

〔例2—1完〕

上例程序中的'REAL', N1, N2, V1, V2 用来指明运算对象的性质, 称为程序的说明部分。READK (3, N1, N2, V1); V2 = N1 * V1 / N2; PRINT (V2), 是用来描述运算(操作)的过程与步骤称为程序的语句部分。因此这个简单的源程序的基本结构是:

```
'BEGIN'
说明部分
语句部分
'END'
```

如果用D代表说明, G代表语句。那么这种简单程序的结构可表示为①:

```
'BEGIN'
D, D, .....D,
G, G, .....G
'END'
```

比较复杂的源程序一般可以是由若干个上述结构经嵌套(内层还可以并列)而构成。上述结构的每一层就称为一个分程序。如

```

Y
'BEGIN'
D, D, ...D,
G, G, ...G,
  'BEGIN'
    D, D, ...D,
    G, G, ...G,
      'BEGIN'
        D, D, ...D,
        G, G, ...G,
        'END',
      G, G, ...G,
      'BEGIN'
        D, D, ...D,
        G, G, ...G,
        'END',
      G, G, ...G,
    'END'
  G, G, ...G
'END'

```

} 第三层
 } 分程序
 } (之一)

} 第二层
 } 分程序

} 第一层
 } 分程序

} 第三层
 } 分程序
 } (之二)

程序的嵌套结构是ALGOL程序的一个重要特点和优点。嵌套在里面的称为内层分程序, 在外面的称为外层分程序。如对上面的第二层分程序而言, 第一层分程序是其外层分程序。

① 在ALGOL中把内容分写成若干行或合并写成一行都是可以的。只要一行中符号的总数不超过一定数额(在ALGOL—121中要求不超过69个符号)

序, 第三层分程序是其内层分程序。

第一节 基 本 符 号

任何语言都是由一些基本符号按一定语法规定而组成的。例如对英语而言, 这些基本符号就是二十六个字母和若干个标点符号。ALGOL—121语言也是由一些基本符号按ALGOL—121语法规则组成的。这些基本符号有以下一些:

一) 字母: 共26个

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

二) 数字: 共10个

0 1 2 3 4 5 6 7 8 9

三) 逻辑值(或称布尔值): 共2个

TRUE(真)①

FALSE(假)

四) 定义符:

(1) 运算符: 共23个

a) 算术运算符: 共6个

+ (加)

- (减)

* (乘)

/ (除)

// (整除)

** (乘方)

b) 关系运算符: 共6个

'LS' (小于, <)

'LQ' (小于等于, ≤)

'GQ' (大于等于, ≥)

'GR' (大于, >)

= (等于)

'NQ' (不等于, ≠)

c) 逻辑运算符: 共5个

'EQV' (全等, ≡)

'NOT' (逻辑非, ¬)

'OR' (逻辑加, ∨)

'AND' (逻辑乘, ∧)

'IMP' (蕴涵, ⊃)

d) 顺序运算符: 共6个

'FOR' (对于)

'DO' (做)

'IF' (如果)

'THEN' (则)

'ELSE' (否则)

'GOTO' (转向)

(2) 分隔符: 共11个

, (逗号)

. (小数点)

: (冒号)

; (分号)

10 (小拾)

: = (赋值号)

:: = (双点赋值号)

'STEP' (步长)

① 括号内是此符号的简单解释和中文读法(下同)

'UNTIL' (直到)

'WHILE' (当)

△ (空格)

(3) 括号: 共 8 个

((开圆括号)

) (闭圆括号)

[(开方括号)

] (闭方括号)

' (定义符括号)

" (行括号)

'BEGIN' (开始)

'END' (结束)

(4) 说明符: 共 9 个

'INTEG' (整型)

'REAL' (实型)

'ARRAY' (数组)

'BOOLE' (布尔型)

'SWITCH' (开关)

'PROC' (过程)

'DRUM' (鼓)

'LIBR' (库)

'TAPE' (带)

(5) 分类符: 共 3 个

'LABEL' (标号)

'VALUE' (值)

'STRING' (行)

在以上各基本符号中不少符号的意义是显而易见, 无需进一步解释。对于那些比较陌生的符号, 我们将在后面相应的各节中加以介绍。

在书写时应注意避免某些基本符号之间的混淆, 为此可对那些易于混淆的写法做一些规定, 在本书中采用以下一些规定:

符 号	含 意	符 号	含 意
O	字母O	Z	字母Z
0	数字0	2	数字2
X	字母X	U	字母U
*	运算符乘	V	字母V
I	字母I	△	空格
1	数字1		

第二节 数、逻辑值、变量、标准函数

§2.1 数

在ALGOL—121中数的表示不能是任意的而只能采取以下两种形式: 整型 ('INTEG' 型) 和实型 ('REAL' 型)。

整型数与通常的习惯写法无区别, 比如 + 3, - 2 等等。数字前面的 +、- 称为数符。数符为正时可省略。

实型数一般可写成如 $-64.53_{10} \cdot 3$ 之形式。它相当于通常的 -64.53×10^{-3} 或 -0.06453 。在这里的 -64 是整数部分, 它是数符 (-) 和无符号的整数 (64) 所组成。53 是小数部分, 它是由小数点 (.) 和无符号整数 (53) 所组成, $_{10} \cdot 3$ 是指数部分, 它是由小拾 ($_{10}$)

①和数符(-)以及无符号的整数(3)所组成。由此可知实数可表示为

整数部分		小数部分		指数部分		
(数符)	(无符号整数)	(小数点)	(无符号整数)	(小拾)	(数符)	(无符号整数)

例: - 64 . 53 ₁₀ - 3 = -64.53 × 10⁻³

实际上可以认为这就是数的一般表示式。因为在以上表示式中只有整数部分时,这个一般的表示式就变成整型数的表示式,所以我们可以说在数的一般表示式中只有整数部分时,就是整型数,否则就是实型数。另外实型数也可以缺以上三个组成部分中的一部分或两部分(只要不同时缺小数部分和指数部分)如.53₁₀-3、-64₁₀-3、-64.53₁₀-3、.53等皆为实型数^②。

例2—2 用正确的ALGOL语言写出(1)0.00057,(2)-7.25 × 10³

解 (1) 0.00057写成0.00057,.00057,0.57₁₀-3,5.7₁₀-4等皆可。

(2) -7.25 × 10³可写成-7.25₁₀3,-0.725₁₀4等,如写成整数形式则为-7250。

[例2—2完]

使用整型数与实型数有什么区别呢?区别主要有三个:

(1) 对于指定的电子计算机可使用的实数范围要比整数范围大。比如在121机中实数的绝对值不得超过约 9.2×10^{18} ^③,但整数绝对值则不得超过约 1.7×10^{10} ^④。

(2) 在计算机中进行运算时,整型数的运算与实型数的运算方式不同。整型数间进行的是所谓定点运算。实型数间进行的是所谓浮点运算。一般讲来,定点运算的速度要高于浮点运算速度。

(3) 整型数在计算机内表示为精确的数,在进行定点运算时计算结果也是准确的,如4 * 2的结果为准确值8。实型数在计算机内进行浮点运算时往往会带有误差。如4.0 * 2.0,其结果可能是8.0000001,也可能是7.9999999。由于整型与实型数间存在有这样的区别,使用者就应根据自己的需要,考虑各方面的利弊而选用。至于在什么时候最适于使用什么形式,以后将在相应的部分加以介绍。

§ 2.2 逻辑值(布尔数)

电子计算机的一个重要特点是能够进行逻辑判断。比如它可以判断5 > 3是正确的。而13 > -2是错误的。为了描述这类判断的结果,在ALGOL语言中引入了两个基本符号TRUE和FALSE。如判断为正确我们就表示为TRUE,反之则为FALSE。

我们知道5 + 3为8,就是说8是5与3进行算术运算(5 + 3)的结果。现在5 > 3为TRUE,所以我们说TRUE是5与3经布尔运算(5 > 3)的结果。从这一意义上讲,

① 注意:这是基本符号中的分隔符小拾(₁₀),而不是由数字1与数字0所组成的一个数。换言之它是一个基本符号。它与后画的-3组成一个整体,代表拾的负三次方。单独的一个小拾是无意义的。小拾后面必须是而且只能是整数。

② 实数也可以写成整数部分加小数点,如64..

③ 准确为 $2^{33} \times (1 - 2^{-34})$

④ 准确为 $2^{34} - 1$