

算法语言及其在测量计算中的应用

begin

曹 恒

integer n, n_1 ;

#read(0, '10', n); $n_1 := n - 1$;

begin array $ak, bk(100)[0:n_1]$;

real a, b, h, h_1, k, i, x, y ;

real procedure $f(x)$; value x ; real x ;

$f :=$ if #abs(x) > 1 then $x/2$ else $2 - 1.5 \times x$;

#read(0, '10', a, b);

$h := (b - a) / 2 / n$; $h_1 := 6.2832 / (b - a)$;

for $k := 0$ step 1 until $n - 1$ do

begin $ak[k] := bk[k] := 0$;

for $i := 0$ step 1 until $2 \times n - 1$ do

begin $x := a + i \times h$; $y := f(x)$; $x := k \times h_1 \times x$;

$ak[k] := ak[k] + y \times \# \cos(x)$;

$bk[k] := bk[k] + y \times \# \sin(x)$

end;

$ak[k] := ak[k] / n$; $bk[k] := bk[k] / n$

end;

#print(0, '2F14.6', ak, bk);

end

end 55...5 止



测绘出版社

算法语言及其在测量 计算中的应用

曹 恒

测绘出版社

本书结合 X-2 机、709 机、DJS-6 机、DJS-21 机较详细地介绍了算法语言 ALGOL 60 的语义和语法规则。书内还介绍了几个经多次使用、已有初步成效的平差计算通用程序。可供测绘科技人员和有关院校师生学习参考。

JSCC/68

算法语言及其在测量计算中的应用

曹 恒

测 绘 出 版 社 出 版

装甲兵“五·七”工厂印刷

新华书店北京发行所发行·各地新华书店经售

开本 $787 \times 1092^{1/32}$ ·印张 $7^{13/16}$ ·字数 175 千字

1978 年 7 月第一版·1978 年 7 月第一次印刷

印数 1—35,000 册·定价 0.66 元

统一书号: 15039·新91

前 言

电子计算机在测量计算中应用较早，由于这项先进技术的采用，使作业人员摆脱了繁重复杂的计算工作，大大地提高了质量和工效。但是，早期编制出的测量计算程序都是“手编程序”。由于它存在着难学、难编、难改的缺点，致使程序设计工作集中在少数专业人员手里，而大多数计算人员则只能套用，甚致连套用都有困难，这样就不利于发挥群众的智慧和创造才能，不利于普及推广。

六十年代初，国内有关单位就开始进行“用程序语言编程序”的研究和实验工作。由于这方面工作取得了显著进展，从而使程序设计工作大为简化。目前越来越多的测量人员能看懂程序，理解程序设计思想，进而自己动手编程序，测量计算工作的面貌正在发生深刻的变化。

一九七四年以来，我们结合生产需要，用算法语言编写了部分测量计算程序，经过多次使用，取得了初步成效。在此基础上整理编写了这本小册子。目的是在普及推广电算技术的过程中起一个抛砖引玉的作用。

本书共分六章，第一章介绍程序设计的基本概念；第二章和第三章介绍算法语言的常用部分，以及源程序的编写和改编方法，至于上机操作方法则从略，读者可参阅有关计算机的使用说明；第四，五，六章是部分通用程序。这些通用程序虽然都是按 X-2 机算法语言编写的，但由于各种计算机的算法语言大同小异，只要稍加修改就可以应用于其它配

备有算法语言编译系统的计算机，为了便于修改起见，我们在附录中列出了四种常用的国产计算机算法语言主要异同对照表，一当有了其中某一种计算机的源程序，就不难改编为其它三种计算机的源程序。

本书初稿完成后，曾打印寄送部分测绘单位审阅，得到了兄弟单位的大力支持，尤其是武汉测绘学院大地测量系，中国人民解放军 57653 部队，北京市地质地形勘测处等单位，提出了很多宝贵意见，在此表示感谢。

由于水平有限，经验不多，初稿虽经过修改，但书中的错误和不妥之处仍在所难免。现有的通用程序虽经过生产考验，但还需要不断改进和提炼，使之逐步完善，为此恳切地希望读者批评指正。

编 者

• 一九七七年十月

目 录

第一章 程序设计的基本概念

第一节	电子计算机的基本工作原理	1
第二节	程序设计方法的发展概况	10
第三节	程序设计步骤	12

第二章 算法语言简介

第一节	基本符号	20
第二节	标识符 数 变量 标准函数	23
第三节	表达式	30
第四节	语句	37
第五节	说明和分程序	61
第六节	过程	74
第七节	标准过程和输入输出语句	89
第八节	控制台变量	101

第三章 程序设计和改编中的几个问题

第一节	程序设计中的注意事项	105
第二节	通用程序的一般说明	108
第三节	改编方法	109

第四章 水准网平差计算通用程序

第一节	自由水准网按条件观测平差	114
第二节	附合水准网按逐渐趋近法平差	132

第五章 三角网按附有条件方程式的间接 观测平差通用程序

第一节	概略计算	146
第二节	按逐渐趋近法平差	169
第三节	按高斯法平差	190
第四节	概略计算与平差计算的合并	212

第六章 平面直角坐标按内插法换算程序

第一节	改算系数计算	222
第二节	应用改算系数换算控制点坐标及编表	231
附录一	常用的几种算法语言主要异同对照表	236
附录二	709 机与部分 X-2 机基本符号及标准过程 标识符缩写表	243
参考文献		244

第一章 程序设计的基本概念

电子计算机是一种先进的电子机器，它具有自动、高速、精度高等特点，能解算各种复杂的计算问题，因而在国防建设和经济建设的各个部门中应用愈来愈广泛。

为什么电子计算机能自动运算，求得我们所需要的结果呢？这固然是由于机器内部应用了大量的电子元件，巧妙地组成了各种电路，使之具备了记忆、运算、逻辑判断和控制能力，为自动运算提供了物质基础。但是，任何机器都是劳动者创造出来的生产工具，要由人来掌握使用，才能发挥其效能。电子计算机也不例外，它要能自动运算，还得有一个依据，这就是人们根据要解决的问题预先给它规定的详细操作内容和步骤，即解题程序。否则电子计算机自动运算求得正确结果是不可能的。由此可见程序设计是使用电子计算机的重要环节。

为了使初学者弄清什么是程序，怎样编程序，我们在本章中首先简单地介绍程序设计有关的一些基本知识。

第一节 电子计算机的基本工作原理

一 电子计算机的主要部件

电子计算机有哪些主要部件，它们的功能是什么，这是程序设计者应具备的基本知识。为了便于理解这个问题，我

们先回想一下在使用电子计算机以前，是怎样进行计算工作的。当我们用手摇计算机或算盘进行计算作业时，先要准备一份计算表格，表格上注明计算方法和步骤，安排好初始数据、中间结果和最后结果的空格，以便计算过程中由计算者来填写数据。有了计算表格后，我们先抄写初始数据（包括起算数据、观测数据和某些常数），然后按选定的公式在手摇计算机或算盘上对有关数据进行运算，并且把必要的中间结果填写在指定的空格内，直到算出最后结果填入表内为止。

根据以上情况，可以联想到电子计算机要能自动地完成计算工作的全过程，就必须具备相当于计算表格的存贮装置、相当于手摇计算机或算盘的运算装置以及代替人来指挥机器各部分协调工作的控制装置，此外还要有把程序和初始数据送进机器的输入装置，把计算结果告诉计算者的输出装置。因此，无论什么型号的电子计算机，尽管其构造和性能会有所不同，但都必须具备以下主要部件：

1. 存贮器 用来存放程序和数据。它由成千上万个存贮单元所组成，单元的位数称字长，单元的编号称单元地址，单元的总个数称存贮容量。显然存贮单元位数愈多计算精度愈高，存贮容量愈大，则解算问题的能力也愈大。

存贮器有内存与外存之分。内存贮器是主要的存贮器，目前一般由磁心体组成，其特点是存贮量小而存（“写入”）取（“读出”）速度较快，它直接与运算器相联，起着计算表格的作用。外存贮器又称为辅助存贮器或中间存贮器，由磁鼓、磁带、磁盘组成，其特点是容量大而存取速度慢，它不直接与运算器相联，只和内存相互传送信息，起着仓库的作用。当程序和数据超过了内存容量时，可将暂时不用的信息送到

外存贮器保存起来，待要用时再由外存送入内存，这样就可利用内存容量较小的计算机解算较大的题目。

2. 运算器 相当于手摇计算机或算盘，用来进行加、减、乘、除四则运算和逻辑运算等操作。

3. 控制器 在整个计算过程中，它根据程序中的各条指令不断地发出信号，指挥和协调其它部件的操作，使计算过程自动、连续地进行，在某种程度上起着人的组织、调度作用。但是机器终究不可能完全代替人的操作，因此还有一个人工控制台，起着人和计算机联系的作用。控制台的台面板上有各种指示灯、开关、扳键和按钮。人通过指示灯可以观察机器的运行情况；利用各种开关、扳键和按钮可以使机器启动或停止工作，临时输入或输出某些指令和数据，改变运算顺序或个别单元内容等。

4. 输入器 其作用是把穿孔纸带上程序和数据的信息转换成电脉冲输入到机器的存贮器中去。目前一般采用光电输入机，此外还有电容式输入机。纸带传送速度为2~3米/秒。

5. 输出器 其作用是把内存贮器中的数据打印出来，供算题者使用。常用的有电灼输出机和行式(宽行和狭行)打印机，目前宽行打印机已逐步代替电灼输出机和狭行打印机而被广泛采用，其输出速度一般在10行/秒左右，每行最多可印80~160个字符。

以上主要部件中，内存贮器、运算器和控制器三部分统称为中央处理装置(主机)，外存贮器和输入、输出部分统称为外部设备，它们之间的联系见图1。

二 电子计算机的解题过程

电子数字计算机虽然能解算复杂的题目，但是，由于它

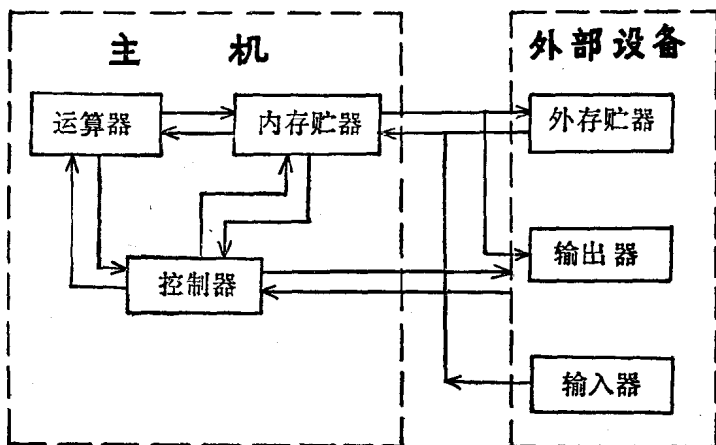


图 1 计算机主要部件之间的联系

只能进行四则运算和逻辑运算等基本操作，而且只能一项一项地执行，因此，在算题前必须把计算公式分解成一系列的基本操作，每一项基本操作给出一条相应的指令，机器才能一步一步地完成整个题目的运算。

所谓“指令”就是指示机器进行某种基本操作的命令，它包括两个方面的内容：一是执行什么操作，二是从哪一个单元中取数来参加该项操作或运算结果存放在何处，以及具体操作的信息等等。前者为操作码部分，后者为地址码部分，两者组成一条完整的指令，即

操作码 θ	地址码 D
--------------	---------

θ 和 D 均用八进制或十六进制代码表示。例如 X-2 机规定 θ 部分两位八进制代码组成。“71”代表取数，“50”代表乘，“04”代表送数……等； D 由五位八进制代码组成，如 00000，

00001, …… , 07777 等。

各种计算机在设计时就已经确定了它能执行哪些操作和相应的指令形式, 组成了该计算机的“指令系统”。所有需要上机解算问题, 归根到底都要有一个按该机指令系统的规定所表达的, 并且符合解题方法的程序, 才能在计算机上进行运算。

下面我们以计算 $h^2/2l$ 为例, 简单地说明计算机解题的基本过程

第一步: 根据计算机指令系统的规定, 设计好计算问题的程序, 并且把程序和初始数据(h , l 和常数 2)由输入设备依次送入存储器;

第二步: 从程序的开始地址启动后, 控制器先从存储器中取出第一条指令; 并按该指令的内容指挥机器各部件进行操作, 此后一条接一条地顺序执行下去 (在一般情况下, 控制器是按各条指令排列的先后次序执行的, 但是如果遇到转移指令, 就要向前或向后跳过若干条指令, 转移到预先安排的地址, 再从该地址起顺序执行下去)。在这中间遇有输出指令, 就把有关数据打印出来, 直到程序结束为止。设我们已按 X-2 机的指令系统设计好上述问题的计算程序, 则其执行情况如下。

下表中第一列是程序的各项指令和有关数据包括初始数据中间结果和最后结果存放在内存单元地址码, 是由程序设计者安排的, 这项工作一般称为内存分配。在 X-2 机中规定一条指令占半字长, 即一个单元中放两条指令, 分左、右指令, 先执行左指令, 后执行右指令, 例如在本例中第一, 第二两条指令放在 00010 号单元中, 第一条是左指令, 第二条是右指令, 而每个数占一个单元, 如 l 分配在 00016 号单元等等。

表 1-1

指令或地址	指 令		指 令 内 容	说 明
	操作码	地址码		
00010	71	00016	$(00016) \Rightarrow A$	从00016单元中取数(l)送运算器的A寄存器
	50	00020	$[(A) \times (00020)] \Rightarrow A$	A 寄存器中的数乘以00020单元中的数, 结果(即 $2l$)在A寄存器
00011	04	00021	$(A) \Rightarrow 00021$	A 寄存器中的数(即 $2l$)送 00021 单元
	71	00017	$(00017) \Rightarrow A$	从 00017 单元中取数(h)送A寄存器
00012	50	00017	$[(A) \times (00017)] \Rightarrow A$	A 寄存器中的数乘以 00017 单元的数, 结果(即 h^2)在A寄存器
	54	00021	$[(A) \div (00021)] \Rightarrow A$	A 寄存器中的数, 被 00021单元中的数除, (即 $h^2/2l$)在A 寄存器
00013	04	00022	$(A) \Rightarrow 00022$	A 寄存器中的数送 00022
	34	01120	按八进制格式输出 计算结果	本指令地址码部分表示八进制输出格式
00014	36	00001		本指令地址码部分表示交换代码个数
	37	00022		本指令地址码部分表示内存开始地址
00015	01	00000	Ω	停机
	00	00000		

续表

指令或 数的地 址	指 令		指 令 内 容	说 明
	操作码	地址码		
00016	<i>l</i>			初始数据
00017	<i>h</i>			初始数据
00020	常数 2			初始数据
00021	<i>2l</i>			中间结果
00022	<i>h/2l</i>			最后结果

上表第二，三列就是程序的指令部分或数据部分，第四，五列是让初学者了解各条指令内容和执行情况而写的，它们不是程序的组成部分。

三 数的表示

1. 十进制数和二进制数

我们日常接触到的数字大部分是十进制数。这种数的每一位数字都是 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 这十个数字当中的一个，如果某一位数字在运算中超过了 9 就要根据“逢十进一”的规则进到前一位上去。因此，数中不同位置上的数字代表该数字乘以 10 的若干次幂。例如十进制数

$$3.1416_{(10)} = 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} + 1 \times 10^{-3} + 6 \times 10^{-4}$$

这是大家都很熟悉的。但是，在电子计算机运算时，一般不是使用十进制数而是使用二进制数，后者根据“逢二进一”的规则进位，因此，二进制数中的每一位数不是 1 就是 0，它们代表该数字乘以 2 的若干次幂。例如二进制数

$$10010_{(二)} = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18_{(十)}$$

2. 八进制数和十六进制数

二进制数的表示形式很简单，运算很方便，但数位多，书写麻烦。为此，在数制又引进了八进制和十六进制，八进制数就是“逢八进一”的数制，每三个二进制数位（从小数点向左或向右依次提取）可以用一个八进制数位来代替，表示该数字乘以 8 的若干次幂。例如，二进制数

$$10010_{(二)} = 22_{(八)} = 2 \times 8^1 + 2 \times 8^0 = 18_{(十)}$$

$$110.001_{(二)} = 6.1_{(八)} = 6 \times 8^0 + 1 \times 8^{-1} = 6.125_{(十)}$$

十六进制数就是“逢十六进一”的数制，相当于我国旧制斤与两的关系。每四个二进制数位（同样由小数点开始向左或向右依次提取）可以用一个十六进制数来代替，表示该数字乘以 16 的若干次幂。由于常用的十进制数字无法表示 10 和 10 以上的数位。规定用 $\bar{0}$, $\bar{1}$, $\bar{2}$, $\bar{3}$, $\bar{4}$, $\bar{5}$, 分别表示 10, 11, 12, 13, 14, 15。例如

$$1101.01_{(二)} = \bar{3}.4_{(十六)} = \bar{3} \times 16^0 + 4 \times 16^{-1} = 13.25_{(十)}$$

3. 二—十进制数

在应用电子计算机解题时，往往要输入一些数据，如果要把它们都换算成二进制数（还要考虑具体的机器表示形式），那是很不方便的，为此，又引进了数的二—十进制形式。二—十进制的表示方法有好几种，通常是用四个二进制数位表示一个十进制数位。具体表示方法见下表。

采用这种记数方法，对于数据穿孔是很方便的。当我们使用穿孔机进行数据穿孔时，按“3”字键，纸带上就出现 0011；按“9”字键，纸带上就出现 1001（有孔为 1，无孔为 0）。这样既满足了计算机必须使用二进制的要求，又省去了数

表 1-2

十 进 制	0	1	2	3	4	5	6	7	8	9
二一十进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

制变换的麻烦，所以广泛采用。

在实际使用时，先要把十进制数写成浮点形式

$$x_{(+)} = \pm 10^{\pm P} \cdot q$$

其中， P 称为阶码； P 左边的符号称为阶符；10 左边的符号称为数符； q 称为尾数其值小于 1，例如

$$3.1416 = 10^1 \cdot 0.31416$$

$$-0.0001 = -10^{-3} \cdot 0.1$$

然后再把它表示为由阶码，阶符，数符，尾数组成的二一十进制的浮点形式。不同的计算机对阶码，阶符，数符，尾数的具体位置以及 P ， q 的位数的规定是不同的，现以 X-2 机和 709 机为例，说明二一十进制浮点形式的表示方法。

X-2 机的二一十进制浮点形式为

$$\underbrace{P_1 P_2}_{\text{阶 阶 码}} \pm \underbrace{\pm q_1 q_2 \cdots q_9}_{\text{数 符 尾 数}}$$

阶 阶 数 尾
码 符 符 数

其中，阶码用两位数表示，其值不大于 19，尾数用 9 位数表示，且 $q_9 \leq 3$ （实际有效数字只有 8 位），阶符和数符都用 0 表示正，1 表示负，例如，

十进制数	相应的二一十进制浮点数
0	0 0 0 0 0 0 0 0 0 0 0 0
3.14 16	0 1 0 0 3 1 4 1 6 0 0 0 0

20 62 64.80 62 0 6 0 0 2 0 6 2 6 4 8 1 0

-0.0001 0 3 1 1 1 0 0 0 0 0 0 0 0

0.017 0 1 1 0 1 7 0 0 0 0 0 0 0

等等。这里要附带说明的是，本书第四、五、六章中涉及到准备数据部分所举的例子都是十进制数，穿孔时应使用二一十进制浮点数。

709 机的二一十进制浮点形式与 X-2 机基本相同，所不同的是，尾数部分用 10 位数表示，且 $0 \leq q_{10} \leq 9$ ，因而实际有效数字就是 10 位。

这种二一十进制浮点数一经输入机器，就由机器内部的专用程序把它们转换成二进制的机器表示形式，从而就可以作为运算的对象了。

第二节 程序设计方法的发展概况

在电子计算机出现的初期，程序都是直接用机器指令编写的(例如前节所举的例子就是这样)。由于这种程序完全是由人工编写的，通常称为“手编程序”(即代码程序)，这是程序设计方法发展过程中的第一阶段。手编程序的优点是能充分发挥机器的性能，程序结构紧凑合理，效率较高，缺点是难学，难编，难改。不易为初学者所掌握，编程序花费的人力时间多，而程序的通用性不强，从而不利于电算技术的普及推广。

随着电子工业的迅速发展，电子计算机的产量愈来愈多，运算速度愈来愈快，用途愈来愈广，相形之下，用手工编程的方法就显得愈来愈落后了。

为了简化程序设计方法，节省编程序的时间。人们开始