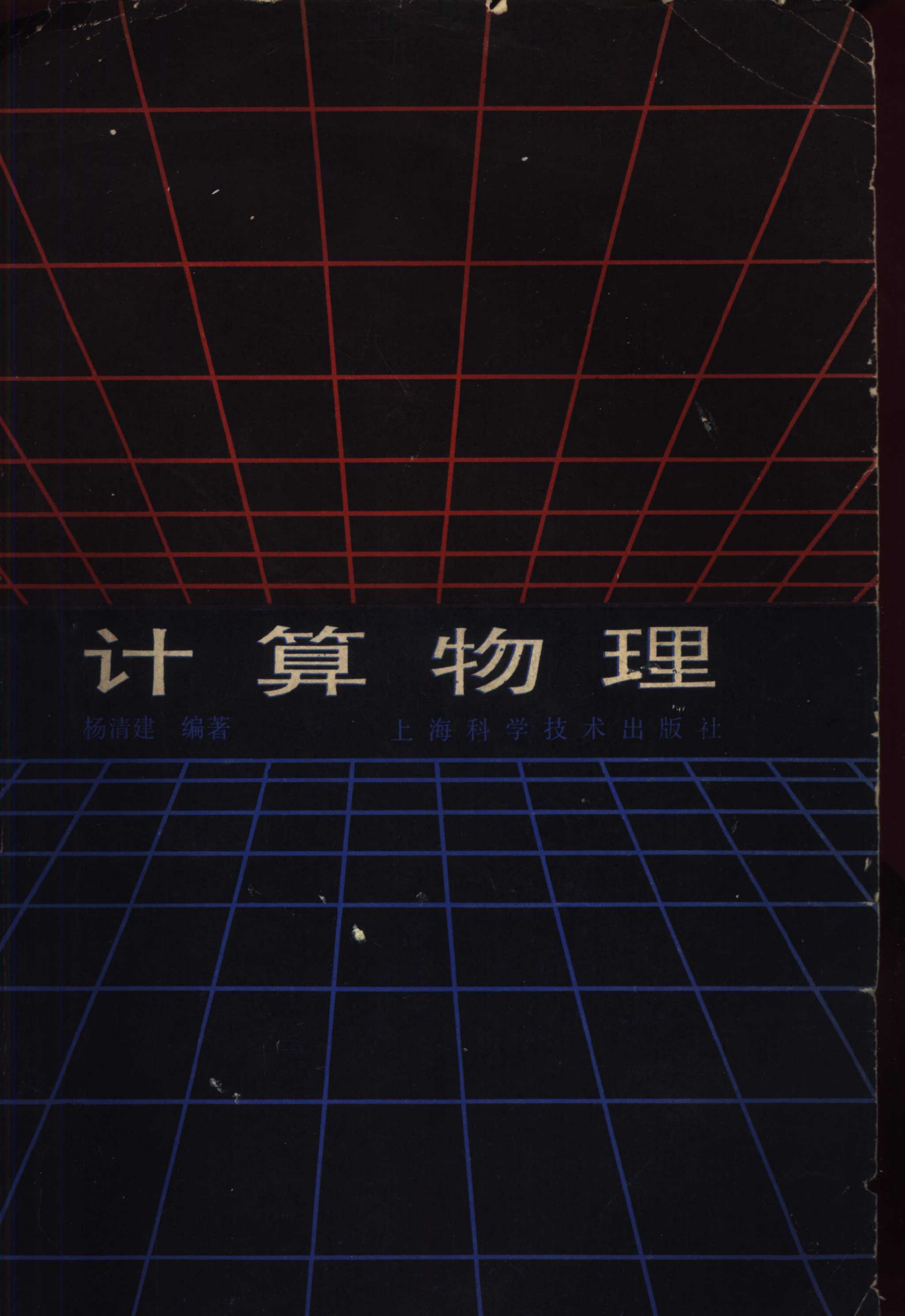




计算物理

杨清建 编著

上海科学技术出版社

内 容 提 要

本书共分九章,分别介绍了计算机在物理学中的应用,物理学中一些简单的计算以及用计算机解普通物理、理论物理、量子力学、电动力学、统计物理中常见的物理问题。书中实例均采用 FORTRAN 语言,附录中给出了相应的 BASIC 语言。

本书适合于理工科大学物理类专业师生阅读,也可供有关专业人员及自学者参考。

计 算 物 理

杨清建 编著

上海科学技术出版社出版

(上海瑞金二路 450 号)

新华书店上海发行所发行

江苏溧水印刷厂印刷

开本 787×1092 1/16 印张 11.75 字数 213,000

1988年5月第1版 1988年5月第1次印刷

印数:1—5,000

ISBN 7-5323-0644-5/O·66

定价: 3.15 元

前 言

近年来电子计算机技术的蓬勃发展,给物理学研究提供了新的强有力的工具,开辟了用计算机从事物理学研究的新途径,出现并形成了计算物理。从广义的角度看,计算物理是指电子计算机在物理学研究中的应用,它包括用计算机对实验物理进行控制、测量和数据处理,对物理问题进行科学计算及用计算机模拟物理过程和物理问题。狭义的角度是指用计算机对物理问题进行科学计算和模拟实验。

计算物理不同于理论物理,它利用计算机来完成理论物理所需要的数值计算。计算物理不同于实验物理,它利用计算机作模拟实验。它也不同于计算数学,计算物理的工作过程是从物理问题出发,以得出物理结果为终结,它往往根据物理图像选取计算方法,有些物理问题的微分方程原来是从差分形式取极限而导出的,在上机计算时又要将微分方程作离散化处理,但在计算物理中可以绕过建立微分方程的步骤,从物理模型的分析直接来编制计算程序,进行上机计算。它是一门与理论物理、实验物理、计算数学、计算机科学交叉的新学科。

一个计算物理工作者不仅要有理论物理方面的基础和解析分析的能力,而且要有实验物理的知识、技能,还要掌握计算方法和计算机的程序设计。本书是学习计算物理的入门教材,简单明确地介绍计算机在物理学上的一些应用及常用的计算方法。把物理问题、计算方法和计算机语言结合在一起,以一个个物理问题为主线索,介绍计算物理的基本知识,包括叙述解这些相应方程式的计算方法,计算流程图和上机实验的程序,对上机结果用物理知识进行判断等等。

在内容编排上本书力求由浅入深,由简单到复杂的顺序介绍计算机求解物理问题的过程。第一章简单介绍计算机语言和基本知识;第二章叙述一些简单的计算;第三章对普通物理中遇到的一些物理问题用微分方程的差分法和网格法及求定积分的辛普生方法进行求解;在初步入门的基础上,第四章专门叙述了用计算机求数值解的误差问题;第五章叙述用诺默罗浮方法求解量子力学中的薛定谔方程;第六章介绍用雅可比矩阵对角化方法求解薛定谔方程的矩阵形式,同时介绍常用线性代数方程组求解方法;第七章叙述用有限元方法求解电动力学中的泊松方程;第八章以统计物理中的物理问题为例介绍蒙特卡罗方法,这是用计算机模拟物理问题的常用方法之一;第九章简单介绍实验数据的处理。本书程序全部用FORTRAN语言编写,附录中给出了全部语句表和内部函数。为用BASIC语言进行学习的需要,附录中也给出了相应的用BASIC语言编写的程序。

本书编写过程中,承蒙李文铸教授大力支持和帮助,也受到美籍教授周传钧讲学的启示,特此表示感谢。限于水平,不妥之处请批评指正。

编 者

1985.5.

目 录

第一章 计算机在物理学中的应用	1
§ 1.1 概述	1
§ 1.2 计算机简介	1
§ 1.3 计算流程图	2
§ 1.4 BASIC 语言介绍	6
§ 1.5 FORTRAN 语言介绍	10
第二章 几个简单计算	18
§ 2.1 平均值和偏差, 气垫实验数据处理	18
§ 2.2 氢气在低温下热容量的计算	21
§ 2.3 多道分析器的数据处理	24
§ 2.4 猜数字, 求平方根	25
§ 2.5 泰勒展开近似计算	28
§ 2.6 π 的计算	32
习题	38
第三章 静电势、电子线路和扩散	39
§ 3.1 两个点电荷产生的静电势	39
§ 3.2 带电金属丝产生的静电势——定积分的近似计算	41
§ 3.3 RC 回路中电容器放电的研究——欧勒法应用	48
§ 3.4 RLO 电路中的放电现象	53
§ 3.5 二维扩散方程的差分解法	61
习题	64
第四章 误差及误差的改善	65
§ 4.1 计算过程的误差	65
§ 4.2 用近似方法计算定积分的误差	67
§ 4.3 龙格库塔方法	69
习题	72
第五章 薛定谔方程	73
§ 5.1 一维方势阱的计算机求解	73
§ 5.2 粒子在鞍力场中的运动	82
习题	95
第六章 用矩阵方法计算薛定谔方程的本征值	96
§ 6.1 本征值方程的矩阵表示	96
§ 6.2 矩阵的定义和一些性质	97

§ 6.3 计算矩阵本征值的雅可比方法	99
§ 6.4 线性代数方程组的高斯解法	104
习题	109
第七章 复杂电荷分布的电势计算	110
§ 7.1 泊松方程和变分法	110
§ 7.2 有限元法	112
§ 7.3 复杂电荷分布电势的有限元法计算	115
第八章 蒙特卡罗方法	125
§ 8.1 随机数的产生	128
§ 8.2 链式反应的模拟	127
§ 8.3 趋向平衡态的计算机模拟	140
§ 8.4 麦曲罗保利斯方法	142
§ 8.5 经典体系自由能的蒙特卡罗计算	147
习题	149
第九章 实验数据处理	150
§ 9.1 实验数据的拉格朗日插值法	150
§ 9.2 样条插值	153
§ 9.3 实验数据的曲线拟合	157
§ 9.4 实验数据频谱分析的快速傅立叶算法	161
附录一 FORTRAN 函数表和语句表	168
附录二 BASIC 语句的相应程序	171

第一章 计算机在物理学中的应用

§ 1.1 概 述

电子计算机是现代科学技术的重大成就之一,在物理领域中它是非常重要的研究工具。同时,研究的需要也促进了计算机硬件(即机器和专用设备)和软件(即计算程序和文本)的蓬勃发展。

在实验物理中,计算机不仅能迅速而精确地处理大量的实验数据,及时分析实验结果,而且能用于控制实验,使庞大和复杂的整套实验装置协调起来有效地自动工作,节省人力和时间,大大提高实验效率。在核物理和高能物理实验中,计算机能自动收集数据,并挑选出对研究最有用的一些数据。

在理论物理中,以前对于一些计算烦琐的问题,只能作一些依据不足、不严格的简化假设,以求得近似的结果。而有了高速、高精度的大型电子计算机,物理工作者就能部分地从大量烦琐计算中解脱出来,集中更多精力去探索物理机制的本质,用计算机去计算由理论推出的解析公式,其结果更接近原先的物理设想,也便于实验检验。对于一些计算实在太繁以前未解决的课题,现在也可用快速电子计算机求其数值解。

利用计算机对物理问题进行模拟是计算物理的一个重要而很有生命力的内容。在假定了一定的实验条件情况下,用计算机可以模拟一些物理过程,使人们能预测实际物理实验可能的结果,从而选择最佳的条件和安排合适实验。这在核工程和高能物理中已经成熟并且广泛地应用。近年来在凝聚态物理和生物物理等方面也有广泛应用。我们知道,研究一定条件下的物理过程,需要有一定的设备和大量的时间,用计算机模拟物理过程最经济,最省时间。这就是当前计算物理迅速发展的原因。

当前物理学的发展,在某种意义上,可以认为是在三个分支上向前发展:实验物理(包括应用物理);理论物理;计算物理。这三者互相联系又互相依赖,但是这三者也有相对的独立性,并互相促进。

计算机在物理学中的应用是一个相当广泛的课题,它不仅能求解物理问题,还能利用它建立物理模型。我们只涉及一部分内容,包括电学和电子线路中应用,解扩散方程,解薛定谔方程,热力学中趋向平衡问题,统计物理中求平均值以及实验数据的处理。

§ 1.2 计算机简介

计算机系统由硬件和软件两大部分组成,硬件是计算机的机器部分,包括控制器,运算器和内存储器,统称主机。存储器用来存储解题的步骤和数据,控制器按照一定顺序到存储器中把解题的步骤取出,然后利用运算器进行运算。硬件还包括输入输出系统及外存储器。输入系统有卡片读入器,纸带输入机,打字机等。输出系统有行式打字机,曲线输出仪,屏幕

显示器等。外存储器通常有磁带、磁盘和磁鼓等等。外存储器容量比内存储器容量大得多，但存取周期长、速度慢。

电子计算机中的电子元件采用“两状态”方式工作，即器件的导通或截止，电位的高或低，电流的有或无，正向或反向，磁场极性的南或北，所以计算机的计算也采用二进制。存储的每一个单元均由若干个二进位数组成，每个单元中包含的二进位数的个数称为该计算机的字长。四个二进位数可以代表一个十进位数。如果一架计算机的字长较长，则它的精确度就较高。计算机的速度是指每秒钟能做多少次运算。一般说，衡量一架电子计算机的硬件技术指标主要是指运算速度，字长和内存储量的大小。下面是常见几架计算机的指标：

表 1-1 常见计算机的技术指标

机 型	银 河	IBM4300 系列	MiCoVAXII	VAX8974	Honeywell
速 度	亿/秒	100万/秒	90 万/秒	2500 万/秒	100 万/秒
字 长		32 位	32 位	32 位	32 位
内 存 容 量		通常 60 兆	5~16 兆	通常 60 兆	2 兆

* 1 K=1024

计算机的软件是为了扩充计算机的功能而设计的。要让计算机代替人工进行运算，必

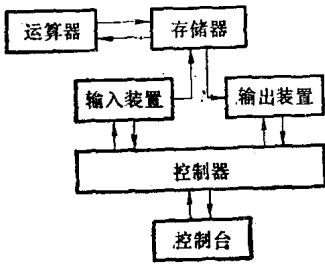


图 1-1 计算机方框图

须把解题的步骤告诉机器，采取人机对话方式工作，解题的步骤称为程序。现代程序用程序语言编写，若用汇编语言写的程序，则由汇编程序“翻译”成机器语言，进行运算，若用高级语言编写的程序（一般称源程序），则由编译程序“翻译”成为机器指令程序，然后执行。汇编程序和编译程序等等都通称为软件。一般把适用于各种用户和各种问题的软件称为通用软件，上面的汇编程序和编译程序就是通用软件，还有操作系统，管理程序等等。另外，还有专门用于特定领域的专用软件。

软件已成为计算机的一个不可分割的组成部分，没有适当的软件配合，计算机的效能根本无法发挥。软件和硬件都是计算机技术的重要研究领域。

计算机组成的方框图如图 1-1 所示：

用计算机解题一般是先根据实际问题抽象出数学模型，然后再选用合适的计算方法进行计算。通常我们用流图来具体描述解题的步骤，根据流图用计算机的程序语言可写出需要计算的源程序，将源程序输入计算机，进行上机调试，执行程序得到结果。对 FORTRAN 语言编写的程序，上机调试的大致过程如下：

1. 用键盘打字或纸带穿孔等方式输入源程序。
2. 由编译程序对源程序进行编译，如果发现语法有错，编译通不过，即修改源程序，一直到编译通过。
3. 程序编译通过，此时源程序已被翻译成机器指令程序，一般称为目标程序。机器执行目标程序就可得到结果。在执行程序的过程中，遇到读语句时，要从外部设备读入数据，遇到写语句时，则在外部设备上输出结果。
4. 如果输出结果不合题意，必须对原先的问题、数学模型、计算方法、程序及输入的数

据重新审查,修改后重复上述步骤,直到输出结果符合题意为止。

对用 BASIC 语言编写的程序,将源程序输入计算机以后,编译是在机器执行过程中逐句进行编译、运算,根据输出结果是否符合要求对源程序进行修改,直到符合题意为止。上机调试, BASIC 程序较简单,但同样课题,用 BASIC 语言进行运算所需的时间比用 FORTRAN 语言进行运算的时间要长。因此一般科技计算常采用 FORTRAN 语言编写源程序。

§1.3 计算流程图

对一个科学计算,我们常常用流程图来具体描述解题步骤及计算的先后次序,根据流程图可以写出相应的源程序,进行上机调试。要写出正确计算的源程序,我们必须学会画流程图。常用的流程图符号有六种。如图 1-2 所示。

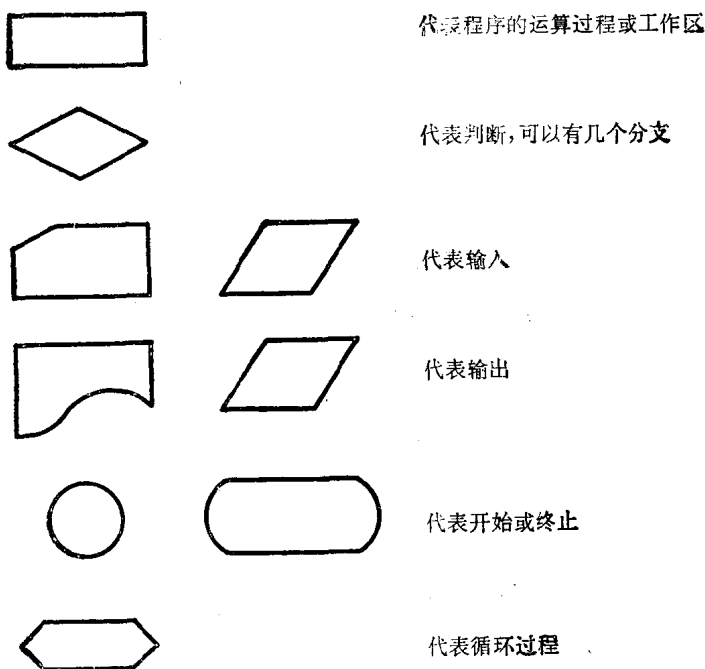


图 1-2 常用的流程图符号

利用流程图的符号,我们将前一节讲述的 FORTRAN 语言编写的源程序上机过程表示为图 1-3。

为了熟悉流程图,下面举三个简单的例子。

例 1 静电问题,有两个点电荷分别带有电荷 Q_1 和 Q_2 ,它们之间的距离为 1 米,求它们之间联线上电场为零点的位置。(如图 1-4 所示)

考虑联线上任意一点 P ,则 P 点电场可以写成

$$E_P = K \frac{Q_1}{r_1^2} - K \frac{Q_2}{r_2^2}$$

取 $K=1$,在数学上表示归一化。则

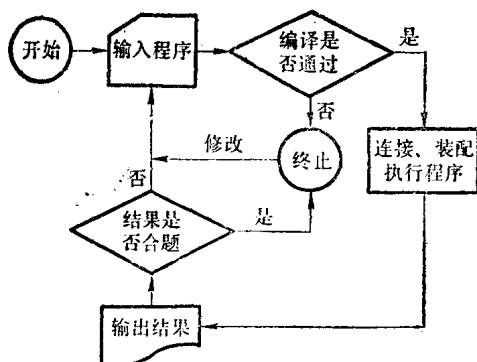


图 1-3 FORTRAN 语言编写源程序的上机过程

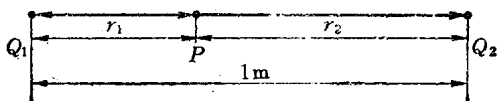


图 1-4 两个点电荷形成的电场

$$E_P = \frac{Q_1}{r_1^2} - \frac{Q_2}{r_2^2}$$

取坐标 X 轴在联线上, 原点在 Q_1 处, 设 $r_1 = X$, $r_2 = 1 - X$ 。我们要求电场为零点的位置, 即

$$\frac{Q_1}{X^2} - \frac{Q_2}{(1-X)^2} = 0$$

可以解 X 得的值为

$$X = \frac{1}{1 + \left(\frac{Q_2}{Q_1}\right)^{1/2}}$$

从上式可以知道, Q_1 不能为零值。以上的计算, 实际上是假定了两个点电荷是带同样符号的电荷, 若 Q_1 和 Q_2 不同符号, 例假设 Q_1 为负值, 如图 1-5 所示, 同样考虑任一点 P , 则 P 点的电场公式为

$$E_P = \frac{Q_1}{r_1^2} + \frac{Q_2}{r_2^2}$$

同样取 Q_1 为坐标原点, $r_1 = X$, 则 $r_2 = X - 1$, 令上述公式为零求解得到 X 值为

$$X = \frac{1}{1 - \left(-\frac{Q_2}{Q_1}\right)^{1/2}}$$

如果两个点电荷所带电量相等, 而符号相反, 这是属于电偶极子的情形, 上述公式亦不能用, 从公式看, 这时分母为零。

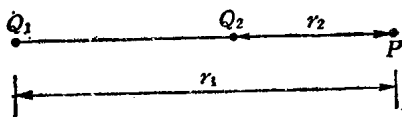


图 1-5 两个极性相反点电荷的电场

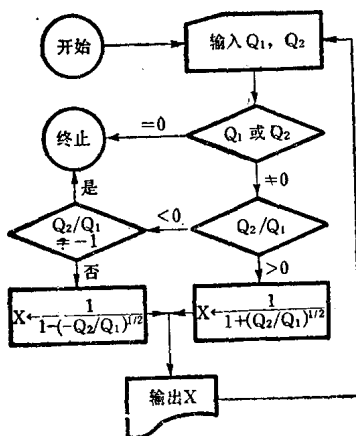


图 1-6 计算电场为零点的流图

针对上述各种情况,用计算机计算任意两个点电荷联线上电场为零点的位置,它的计算流程图如图 1-6 所示。

例 2 求电子计算机的精度

一台电子计算机,我们通常要知道它的精度,计算机的精度一般指计算机浮点数运算的值与通常实数间存在的最小差值,它们的相对误差就是计算机的精度。也就是计算机具有的最大有效数字的位数。我们可以用逐次逼近的方法通过一个程序求解。首先给任意变量 E 任意赋值,例如取为 2,取 E 值的一半为新的 E ,将它和 1 相加得到的值设为 T ,用计算机判断 T 是否大于 1,如果大于 1,则将新的 E 值再取一半为小的新 E 。同样与 1 相加得到新的 T 值,重复用计算机判断,继续上面过程,循环计算,这样 E 的数值越来越小,直到计算机分辨不出它和 1 相加后的值 T 是否大于 1,将此 E 值输出。它就是通常我们所说的电子计算机的精度。一架计算机的精度通常由它的字长确定。我们画出它的计算流程图如图 1-7 所示。

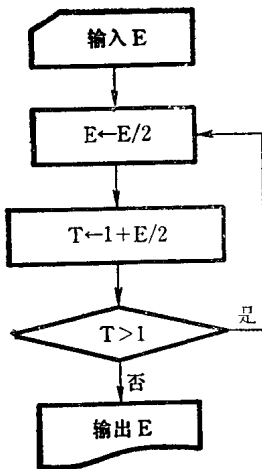


图 1-7 求计算机精度的流程图

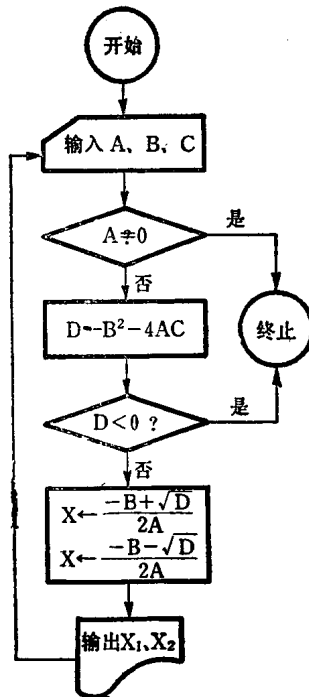


图 1-8 求二次方程实根的流程图

例 3 求二次方程 $AX^2 + BX + C = 0$ 的实根。其中方程的系数 A, B, C 是从外面输入的,要求输入一组系数以后就计算这个方程的两个实根,进行输出,接着又输入下一组系数继续计算,重复进行,直到输入的系数 $A=0$ 或输入的系数使方程不具有实根时才停止计算。

根据上述要求,可画出计算流程图如图 1-8 所示。

由以上几个例子可以看出,计算流程图是描述了计算的流程,它是程序设计的重要手段,它能简单明了地显示出解决具体问题的思路和所用的计算方法。根据流程图,可以用计算机的程序语言写出上机的计算程序,可以减少写源程序时出现的差错。画流程图是设计程序必须掌握的知识,即使不懂计算机的程序语言,也同样可以学会画计算流程图,当然画流程图的目的

是为了写出正确计算的源程序。

计算程序的编写经历了一个发展过程。开始时采用手编程序,用机器语言编写,程序很烦琐,容易出错,编写工作效率很低。后来用编译程序,采用汇编语言,用符号表示,这需要编译过程,编译程序与计算的公式差别很大,很难在不同型号的计算机上互换使用。随着高速和不同型号计算机的发展,五十年代中期以后相继出现了多种高级语言,如 FORTRAN, BASIC, COBOL, PASCAL, BCY 等等。BASIC 语言是一种易学、易写的普及性语言, FORTRAN 语言主要用于科技计算, COBOL 语言主要用于商业及管理系统。用这些语言编写的源程序与问题的原来形式非常接近,易于掌握。由于用高级语言编写源程序脱离具体使用的计算机,因此,程序在不同型号的计算机上可以互换,这给具体工作人员的使用提供了方便,为广泛应用计算机创造了良好的条件,这又促进了计算机的蓬勃发展。

本书简单介绍了 BASIC 语言和 FORTRAN 语言,但书中所举实例的程序都用 FORTRAN 编写,相应的 BASIC 语言程序在书后的附录中可查到。

§1.4 BASIC 语言介绍

BASIC 语言是一种易学、易写、而且具有会话性能的语言。1966 年一问世,就被广泛地用在中、小型计算机和具有多入口的分时终端设备的计算系统中,特别普及用于近年来大量生产的微型计算机中。

BASIC 的全名是 Beginners All-purpose Symbolic Instruction Code。意即初学者通用符号指令代码。

一、BASIC 的基本符号

任何语言都有它自己特定的一套基本字符。BASIC 语言也是有一些基本字符组成的。基本字符共分成三类:

1. 英文字母,共 26 个。

ABCDEFGHIJKLMNOPQRSTUVWXYZ。都采用大写字母。

2. 数字,共 10 个。

0123456789。

3. 符号,不同型号计算机有多有少,一般包括运算符号,关系符号,标点符号及专用符号。常用的有

$+$ $-$ $*$ $/$ $\uparrow$ $=$ $()$ $,$ $."$ $\$$ 等,分别代表加,减,乘,除,幂次,等号,左括号,右括号,逗号,圆点,引号,货币号。

用以上三类字符可组成全部 BASIC 语言及有关数学表达式。

二、BASIC 词法

1. 数

数都采用十进制,一般最多允许 7 位有效数字,即在第 7 位发生误差。因此当一个数有效数字大于等于 7 位时,在用计算机输出过程只保持 6 位的正确值。例

数	输出格式
-2000000	- 2.00000 E+6
200000000000	2.00000 E+10

108.999	108.999
-0.0000256789	--2.56789 E--5
5×10^3	5000

其中 $E+6$ 代表 10^6 , 一般机器输出时, $+$ 号可省略, $E-5$ 代表 10^{-5} 。一般计算机的精度就是有效数字的位数, 现在有的机器的有效数位数已超过上面的情况。数的大小范围通常在 10^{-39} 与 10^{39} 之间。

2. 变量

变量也是符号, 和代数中的变量完全一样, 都是用英文字母表示。也可以在字母后面再跟一个数字或字母作为变量。例如: X, Y, Z, A, B, C , 也可以是 AI, AD 等。因此可以组成许多变量。如果变量名超过两个符号, 同样是允许的, 但是只有最前面的两个符号, 计算机能够分辨, 才有意义, 后面符号不能起分辨不同变量的作用。

3. 数组和下标变量

数组的名字用一个字母表示。数组有一维数组和二维数组。有的机器也可用三维数组。

例如: $A(5), B(2, 6), C(2, 3, 6)$

其中 A 是一维数组, 有 $A(0), A(1), A(2), A(3), A(4), A(5)$ 共 6 个数组元素, B 是二维数组, 有 3×7 个数组元素, C 是三维数组, 有 $3 \times 4 \times 7$ 个数组元素。

一维数组可描写一组数据, 它的排列次序是 $A(0), A(1) \dots A(N-1), A(N)$ 。二维数组可描写行列式或平面上各点的情况, 它的排列次序是 $B(0, 0), B(0, 1), \dots, B(0, N), B(1, 0), \dots, B(1, N), \dots, B(M, 0), \dots, B(M, N)$ 。即按行的次序排列。

当用 $A(I), B(I, J)$ 表示时, 其中 I, J 称为下标变量。BASIC 规定, 下标变量可以从 0 或 1 开始, 但不能取负值。

4. 数学表达式

数学表达式由运算对象(数, 变量, 函数等)运算符($+, -, *, /, \uparrow$ 等)及括号、等式组成。表达式中的计算顺序与一般数学中一样, 先括号内, 后括号外, 运算符中乘方最优先, 乘除其次, 加减最后。例如对 $2 - (A+B)^D \cdot C$ 的运算, BASIC 语言应编写成 $2 - (A+B) \uparrow D * C$, 它的运算次序为先求 $A+B$ 的值, 接着作幂次 D 的计算, 再乘 C , 最后从 2 中减去上述结果。

5. 串变量

现代计算机不仅能作数值计算, 而且能直接储存和处理文字信息。

文字信息指由字母、符号、数字及空格等组成的字符串。计算机设有专门的内存区作为存储文字信息使用, 变量用 $A\$$ 表示, 称串变量, 字母后加上标识符 $\$$, 以示与通常的变量进行区别。例如程序中有语句 $A\$ = "HOW ARE YOU!"$, 它表示把引号中的字符串存放到了以 $A\$$ 命名的内存单元中去。不同型号的计算机能存储文字信息的容量也不同。

6. 内部函数

为了使用方便, 往往把数学中常用的一些函数事先排好一个一个的子程序, 放在计算机系统中, 常用函数的子程序称内部函数。在运算过程中, 当遇到数学表达式中包含内部函数时, 机器可以自动转入子程序进行计算, 完毕后又自动返回将结果代入表达式, 使用很方便。内部函数的多少随不同机器有所不同, 通常的内部函数有

SIN(X)	正弦函数,	X 以弧度表示;
COS(X)	余弦函数,	X 以弧度表示;

TAN(X) 正切函数, X 以弧度表示;
 ATN(X) 反正切函数, 结果以弧度表示;
 EXP(X) 指数函数 e^x ;
 LOG(X) X 的自然对数 $\ln X$;
 SQR(X) X 的平方根 $\sqrt{X}$;
 ABS(X) X 的绝对值;
 INT(X) 不大于 X 的最大整数, 如 $\text{INT}(2.99) = 2$;

RND(X) 随机数, 在 $0 \sim 1$ 之间, 其中 X 是数或数学表达式。但实际上不同机器的随机数表示不同, 例如微型机 TRS-80 其 $0 \sim 1$ 间的随机数用 $\text{RND}(0)$ 表示, 而 APPLE II 机 $0 \sim 1$ 间的随机数用 $\text{RND}(1)$ 表示;

SGN(X) X 的代数符号

表示: $X > 0$ 时 $\text{SGN}(X) = +1$

$X = 0$ 时 $\text{SGN}(X) = 0$

$X < 0$ 时 $\text{SGN}(X) = -1$

以上是常用的内部函数, 有些机器内部函数还要多些。关于这些函数的应用在以后实例中将遇到(附录中)

三、BASIC 语句

BASIC 源程序由一系列语句组成, 每一个语句有语句标号、定义符及语句体三部分组成。例如

10 LET $A = 4.17 + X$

其中 10 称为语句标号, LET 就是定义符, 它的作用是确定计算机执行某个规定的功能, 定义符后面是语句体, 它是根据定义符规定的功能执行具体的内容例 $A = 4.17 + X$ 表示变量 (A) = 数学表达式 ($4.17 + X$), 这样的数学表达式也可以是数或变量。BASIC 规定语句必须有标号, 它是按语句标号的顺序由小到大进行运算。标号可以从 1 (或 0) 开始, 编写程序可以不按标号顺序。一般在连续两个语句的标号之间留有空的数字, 以便修改程序时又可插入语句。

一般常用语句及功能列表如下

语 句	英 文 符 号	功 能
赋值语句	LET	对某一变量赋值。对 BASIC(II) 而言, LET 符号可省略
输出语句	PRINT	用来输出数据
输入语句	INPUT	用来输入数据
读 语 句	READ DATA	读入数据, 常用于数组输入, 与 DATA 语句联用。 供给可读数据(非执行语句)
转移语句	GO TO n	无条件转移到指定为 n 标号的语句
条件语句	IF	条件转移, 一般用法是 IF (比较) GO TO n 联用 或 IF (比较) THEN n 联用
循环语句	FOR I=N TO M STEP X NEXT I	建立程序的循环, 变量 I 从 N 开始到 M 止, 每一次增加步长为 X。
转 语 句	GOSUB n	转入子程序, 从标号为 n 的语句开始
调用语句	CALL	调用子程序
返回语句	RETURN	子程序结束处的语句, 返回主程序
停 语 句	STOP	中止程序执行
结束语句	END	结束程序
维数语句	DIM	说明数组的维数
函数定义语句	DEF MAT REM	定义一个用户函数 矩阵运算 注解

以上只是给出了语句的形式和一般功能,它的具体作用在实际应用中可逐渐理解。

有了 BASIC 语言的一般知识以后,我们可将 §1.3 计算流程图中的三个例子编写出 BASIC 的源程序。

例 1 求两个点电荷联线上电场为零点的位置。这个问题的流程图就是图 1-6,它的程序是

```

5  INPUT Q1, Q2
10* IF Q1=0 THEN 60
15 IF Q2=0 THEN 60
20 LET Y=Q2/Q1
25 IF Y<0 THEN 40
30 X=1/(1+Y↑0.5)
35 GO TO 50
40 IF Y=-1 THEN 60
45 X=1/(1-(-Y)↑0.5)
50 PRINT X
55 GO TO 5
60 STOP

```

程序说明: 标号 20 的语句为赋值语句有 LET, 当用 BASIC II 时可省略, 所以标号 30 和 45 的 LET 省略了。标号 55 的语句表示可以连续计算不同电荷值(Q1, Q2)的零点值位置。

例 2 求一台计算机的精度

它的计算流程图如图 1-7 所示, 相应的源程序是

```

5  INPUT E
10 E=E/2
15 T=1+E/2
20 IF T>1 GO TO 10
25 PRINT E
30 GO TO 5
35 STOP

```

说明: 加上语句标号为 30 的语句目的是观察一下输入不同的 E 值时, 输出的 E 值有什么变化。

例 3 求二次方程

$$AX^2 + BX + C = 0$$

的实根。具体的计算流程图如图 1-8 所示, 相应的源程序为

```

5  INPUT A, B, C
10 IF A=0 THEN 50
15 D=B↑2-4*A*C
20 IF D<0 THEN 50
25 X1=(-B+SQR(D))/(2*A)
30 X2=(-B-SQR(D))/(2*A)
40 PRINT X1, X2
45 GO TO 5
50 STOP

```

我们对 BASIC 语言只是作了简单介绍, 只有通过大量的实践, 才能掌握 BASIC 语句, 并能熟练地应用它, BASIC 语句简单, 而且和英文原来解释差不多, 也有的是取英文单字开

头几个字母,所以比较容易掌握和记忆。编写程序,必须严格按照要求写,要有语句标号,不要写错,错的语句机器通不过。本书以后各章节的内容以 FORTRAN 为主,在附录中有相应 BASIC 语言的程序供读者查阅。

§ 1.5 FORTRAN 语言介绍

第一个 FORTRAN 语言发表于 1954 年,随着应用的发展,有 FORTRAN I, FORTRAN II, FORTRAN III, FORTRAN IV。1966 年美国标准协会公布了标准文本:美国标准 FORTRAN。它相当于以前的 FORTRAN IV。以后仍有发展,但基本上相同。

FORTRAN 语言有许多优点:容易读,标准化程度高,易于交流,计算速度快,是块状结构,有利编写程序,因此 FORTRAN 语言是常用的计算机语言之一。国际上,大约 80% 以上的人在数字计算时用 FORTRAN 语言,许多科技资料也采用 FORTRAN 语言编写,因此应用相当广泛。

一、基本符号

FORTRAN 语言的基本符号与前面的 BASIC 语言类同,有 26 个英文字母、10 个数字及一些特殊字符(即 +, -, *, /, =, (,), ., ., 空白,币符等)。在 FORTRAN 语言中空白是一个有意义的符号。在程序中一般使用的符号都是计算机的键盘所能提供的字符。和 BASIC 语言的差别是:在 BASIC 中幂次用 $\uparrow$ 表示,而在 FORTRAN 语言中幂次用 ** 表示。

二、几点说明

1. 数

FORTRAN 语言中常数分许多类型,有整常数、实常数、双精度常数、复常数、逻辑常数和文字常数。常用的是整常数和实常数。由 0 至 9 的数字组成的数称为整常数,前面可以冠有“+”或“-”号。不同的计算机由于内存不同能够允许的最大整常数亦不相同。有小数点或指数部分的数称为实常数。例如 1.2, 0.12, 12.0, IE-10 等,其中 IE-10 表示 10^{-10} 。在 FORTRAN 语句中,计算公式中数的类型要统一,不同类型的数不能混合运算。

2. 变量

FORTRAN 语言中变量共有五种类型:整型、实型、双精度型、复型和逻辑型。变量名通过符号来定义,变量的符号名有效长度为六个字符,超过六个字符的变量超过部分的字符作无效处理。例如:AIJKS, B12K, S, AB1234K 等。最后一个变量 AB1234 为有效字符。变量以字母开头,由字母和数字组成。

变量的类型要通过类型语句说明。例如

```
REAL AIJKS, B12K, S
INTEGER AST, AB1234
DOUBLE PRECISION XT, YZ
COMPLEX VX, VXX
LOGICAL A1, A2
```

其中关键字 REAL, INTEGER, DOUBLE PRECISION, COMPLEX, LOGICAL 分别表示实,整,双精度,复,逻辑,说明这些字后面所跟的变量分别是实型,整型,双精度型,复

型和逻辑型。在科技计算中常常用到实型和整型的变量,为了方便,计算机的 FORTRAN 语言有一个隐含的规定:一个变量的符号名,它的第一个字母取成 I、J、K、L、M 和 N 中任一字符时,机器自动判别变量为整型,若变量的第一个字符不取上述的六个字母时,则机器自动判别变量为实型。隐含规定简便了程序的编写。

3. 数组

FORTRAN 语言中可以有一维数组,二维数组,三维数组,可分别描写一维、二维和三维的情况。例如 A(50), B(50, 50), C(50, 50, 50)。写成 A(I), B(I, J), C(I, J, K) 表示相应的数组元素的变量,要求下标变量 I, J 和 K 大于等于 1,不能取零和负值。

4. 函数

FORTRAN 中对于常用的各种函数,例如三角函数、指数、对数、开方、取绝对值等函数,都已预先配备好,需要时可直接引用。其中有一部分称为内部函数,另一部分称为基本外部函数。例如求绝对值,截断等,只需几条指令,因此,当源程序出现这些函数时,将几条指令插入目标程序中,遇到几次就插入几次,这类函数称为内部函数。而对正弦、余弦等函数,需要较多指令来现实,若遇到一次就插入一次,目标程序就变得冗长,于是采取另外的处理方式,先由编译程序把正弦、余弦等函数各自编成一个子程序,存放在机器中。当源程序中出现正弦或余弦函数时,只产生一条目标程序,把控制转到正弦或余弦函数的子程序,待算出结果后返回到原来引用处。这种处理方式的函数称为基本外部函数。两类函数的差别仅仅是机器在目标程序中实现的方式不同。在实际应用时,我们不必去注意它们是内部函数还是基本外部函数,都可直接引用编在程序中。但是,引用时要注意的是函数的类型与计算公式中的类型要一致。为了一般应用,下面列出最常用的几个函数:

引用形式	数 学 意 义	类 型	
		自 变 量	函 数
EXP(E)	指数函数 e^E	实 型	实 型
ALOG(E)	E 的自然对数 $\ln(E)$	实 型	实 型
SIN(E)	E 的正弦, E 取弧度值	实 型	实 型
COS(E)	E 的余弦, E 取弧度值	实 型	实 型
SQRT(E)	E 的平方根	实 型	实 型
ATAN(E)	E 的反正切的主值	实 型	实 型
ABS(E)	E 的绝对值	实 型	实 型
IFIX(E)	实数 E 化整数,即舍去 E 值的小数转化为整数。例如 IFIX(5.6)=5 IFIX(-5.8)=-5	实 型	整 型
FLOAT(I)	整数 I 化为实数。例如 FLOAT(3)=3.0	整 型	实 型

FORTRAN 提供了很多内部函数和基本外部函数,详细可查阅最后的附表。

函数引用时允许函数之间互相相套的情况。例如

SIN(SQRT(X)+1.0)

SQRT(FLOAT(I))

等,前面表示先开方将值加1.0再取正弦,这是求值,后者表示将整变数 I 化为实型再开方,这是函数类型规定所要求。

三、FORTRAN 语句

FORTRAN 语言是一条一条具有卡片性的语言,它比较严格的书写格式。源程序由一个或几个相对独立的程序段所组成。程序一般有注解行、语句行和结束行组成。注解行前面用 C 标记,以示区别,若有多个注解行,每行前面都要用 C 标记。注解行可以在程序中任意地方出现,它不是可执行语句,机器不作处理,仅仅是在打印出源程序时,作为说明源程序的作用而用。结束行一般用

END

表示,它本身是一个结束语句。对语句行而言,FORTRAN 语句很多,我们介绍常用的几个语句。

1. 输出和输入语句

输出语句亦称写语句,它的形式是

```
WRITE(3,17) I1, I2, K1, X1
```

```
17 FORMAT(1X, I4, 1X, I4, 1X, I4, 3X, F5.2)
```

其中 WRITE 表示输出语句,后面括号内的整常数 3 表示在编号为 3 号的外部设备上输出结果。因为一般计算机输出设备都编有设备号,在使用某一型号的计算机时,先询问输出设备的编号,然后选择在需要的输出设备上输出。也可以在程序中自己定义某一设备的具体编号值,例如我们将行式打印机规定为 3 号,定义方式随机型不同有所不同。再看括号内的整常数 17,它是一个语句的标号,表示按标号为 17 的格式语句(FORMAT)所提供的方式输出结果。后面的 I1, I2, K1, X1 表示输出三个整型变量和一个实型变量。FORMAT 是格式语句,格式语句一定要有语句标号,以备其他语句引用。在 FORTRAN 语言中除了格式语句外,其他语句一般不需要标号,机器运算按编写的顺序执行。FORMAT 后面括号内 X 字符表示输出时为空白,1X 代表空一格,5X 代表空五格。I 字符表示输出或输入整数, I4 表示输出最多只要四位数字。F 字符表示输出或输入实数,一般是输出或输入十进制的浮点数, F5.2 表示最多只有 5 个字符,包括小数点占有的一个字符在内,其中 2 表示要求小数点以后有二位数字,这样小数点以前只有二个符号位,考虑到数值可能有正负符号问题,最后 F5.2 的输出数在小数点以前只允许一个数字。上面的输出形式为

□××××□××××□××××□□□××.××

其中 □ 表示空格、×××× 为输出的数。输出的格式可以按问题的需要而定,但必须先对输出变量值有所估计,并留有一定余量。如果输出格式没有选择好,将导致正确的计算结果在输出时遭到破坏而失去意义。

输入语句亦称读语句,它的形式是

```
READ(4, 15) I, J, X
```

```
15 FORMAT(I4, I3, F10.5)
```

或 15 FORMAT(2I0, F0.0)

其中 READ 是读语句,它表示由设备编号为 4 的设备,按标号为 15 的格式语句所规定的格式输入整变量 I、J 和实变量 X 的值。标号为 15 的前一个格式语句表示输入形式为

$$\underbrace{\times \times \times \times \times \times \times}_{4} \underbrace{\times \times \times \times}_{3} \underbrace{\times \times \times \times \times \times \times}_{10}$$