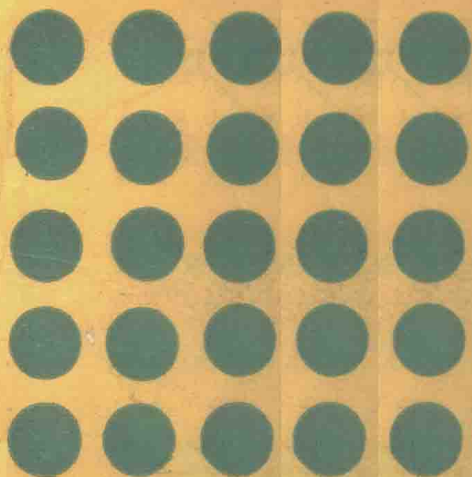


高等学校试用教材

计算实习

李志良 许梦杰 编



高等教育出版社

高等学校试用教材

计 算 实 习

李志良 许梦杰 编

高等教育出版社

内 容 提 要

本书分六章,前两章介绍模块结构化程序的设计方法,并具体介绍了数值分析中常用算法的模块程序设计方法。第三章列举了七个应用实例。第四章给出了二十个实习课题。后两章分别介绍了IBM-PC机的操作方法及科学计算绘图。

本书内容精炼,叙述严谨、通俗易懂,并有较多有特色的实习课题供学生上机实践,这对提高学生编程能力和加深对数值分析内容的理解都有很大益处。

本书可作为大专院校学生学习“计算实习”课的教材,也可作为研究生及有关科技人员的参考书。

高等学校试用教材

计 算 实 习

李志良 许梦杰

✱

高等教育出版社出版

新华书店北京发行所发行

河北省香河县印刷厂印装

✱

开本850×1068 1/32 印张8.25 字数210 000

1989 年5月第1版 1989 年5月第1次印刷

印数0 001-4 910

ISBN7-04-002017-3/TP·43

定价2.00元

序 言

本书可作为大专院校计算实习课的教材，也可作为算法语言与计算方法课的教学参考书，以及研究生和科技人员进行科学计算的实用指导书。

本书共分六章。第一章介绍编制科学计算程序的方法，提出模块结构图思想，实现模块化结构程序设计；第二章通过10个常用数值方法程序编制的介绍，使读者了解了程序设计方法的全过程，有利于读者掌握总体分析思想，从而提高编程能力；第三章分析7个应用例子，阐明在科学计算中数学模型的归结和计算方法的选取。这些实际例子是编者的实践工作成果和创见，有利于提高读者分析问题和解决问题的能力；第四章给出20个计算实习课题，提出“实习报告”规范，让读者按照规范要求，有目的、有创造地进行计算实习，并写出“实习报告”，作为一篇模拟学术论文，这样能加强科学计算实践环节；第五章主要介绍了IBM-PC微机的使用方法，这样有利于上机调试。同时推荐了2个应用软件，以提高读者应用计算机的能力，为科学计算服务。另外也介绍了编者在实践中查错的点滴经验；第六章着重叙述科学计算绘图方法，使数值方法大量的计算结果经过图象处理，能形象化、直观化、使之更有利于计算结果的分析，从而将科学计算和计算机绘图紧密结合。

学生在学了“计算方法”和“高级语言”课后，普遍存在的问题是只会套用“算法汇编”书中的标准库过程进行科学计算，不能独立地将数值方法编制成计算机程序，以致不会灵活应用已经掌握的方法计算综合性较大的课题。他们编出的计算程序经常是滥用GOTO语句，破坏了程序模块化结构，使程序静态和动态顺序不统一，故不能做到易读、易调试。为此，我们编写的“计算实习”教材，主要突出科学计算程序模块化思想，这是提高

读者编程能力的关键，从而使读者学会总体分析，奠定从事科学计算工作的基础。绝不是以解释某个数值算法程序或者解决它的编制问题作为我们编写的宗旨。

本书的具体特点如下：

(1) 根据数值方法的特点，我们总结出几种基本算法模块。例如累乘、累加、迭代、交换等语言模块，这样读者通过对某数值计算公式的分析，确定其有哪几种基本模块组合而成，就能灵活编写出一般数值公式的程序；

(2) 提出模块结构图的新思想。常用的程序框图太精细，结构不清晰，有时不易设计和编制。我们根据PC BASIC和FORTRAN 77算法语言结构性强的特点，采用模块结构图来代替以往的框图设计，其不仅符合“逐步求精”的程序设计方法，而且建立了结构图和程序语言相近对应的关系，有利于读者编制出模块结构程序；

(3) 通过一些应用实例分析，明确如何将一个综合问题进行模块结构化设计，介绍了“结构程序设计方法的全过程”，使读者掌握科学计算程序设计的总体设计方法，并且这个设计方法贯穿在各数值方法的程序编制中；

(4) 本教材中所列举的数值方法程序都经过我们重新编制，其结构性强，所以结合总体结构图和符号引用表，都能容易读懂。所用语言是结构BASIC和FORTRAN 77，由于微机调试BASIC语言程序方便，所以主要列出BASIC程序，同时介绍了FORTRAN 77程序的转换、调用和查错方法，使读者也能应用FORTRAN 77语言进行科学计算；

(5) 结合实习课题提出“实习报告”规范，使读者初步掌握写学术论文的模式，而且这些课题不是光套用标准过程所能完成的，而是把重点放在总体分析、方法选取、程序结构、上机调试和结果讨论等环节上；

(6) 介绍科学计算绘图方法。图象处理是计算机应用特点

之一，有必要将科学计算和计算机绘图紧密结合起来，在教材中介绍了基本绘图语句及曲线和曲面的绘制方法，从而使数值方法的大量计算结果能形象化。如插值、逼近、O.D.E等各种方法比较，也能用图象来显示。

本书在编写过程中，得到王德人、泮仲雄两位教授的支持和帮助，郑权教授详细审阅初稿，由清华大学李庆扬教授为本书作了认真审稿，并提出了宝贵意见。对此，编者均表示衷心感谢。

限于编者水平，书中难免有不妥或错误之处，恳请读者批评指正。

编 者

目 录

第一章 科学计算程序的编制方法	1
§1.1 程序的质量标准	1
§1.2 程序设计的风格	3
§1.3 结构程序设计方法	11
§1.4 基本算法模块	30
习题	35
第二章 常用数值方法的程序编制	33
§2.1 一元三点插值	38
§2.2 三次样条插值	44
§2.3 积分(龙贝格算法)	54
§2.4 初值问题	60
§2.5 列主元高斯消去法	66
§2.6 切比雪夫逼近	69
§2.7 Cholesky分解法	73
§2.8 Jacobi方法	80
§2.9 Householder变换	86
§2.10 求对称三对角矩阵特征值的二分法	94
习题	102
第三章 应用例子分析	106
§3.1 IMEV电子通量计算	106
§3.2 太阳能电池直流电压特性分析	112
§3.3 稀疏对称线性方程组解法	116
§3.4 储气柜水槽外井壁的应力计算	120
§3.5 船舶管系中弯管的数学模型	126
§3.6 单相电容式电机性能计算	132
§3.7 振动分析	136
第四章 计算实习课题	146
§4.1 怎样写实习报告	146
§4.2 课题1 圆周率 π 的计算	155

§4.3	课题2	迭代格式的比较	156
§4.4	课题3	Ronge现象的产生和克服	157
§4.5	课题4	离散函数的逼近	158
§4.6	课题5	奇异积分和无穷限积分	159
§4.7	课题6	计算逆阵和行列式值	159
§4.8	课题7	求矩阵的秩	160
§4.9	课题8	SOR方法的最佳超松弛因子选取	161
§4.10	课题9	结构振动计算	162
§4.11	课题10	磁体脉冲磁化的数值分析	163
§4.12	课题11	数值方法教学软件	166
§4.13	课题12	电通量的数值计算(综合题)	167
§4.14	课题13	用网络方法解最短路线问题	167
§4.15	课题14	多种物品的存储管理	169
§4.16	课题15	半导体器件的数值分析	170
§4.17	课题16	用差分法解均匀磁化磁体的磁场	172
§4.18	课题17	求总极值的统计试验最优化算法	175
§4.19	课题18	复合形法解优化设计问题	177
§4.20	课题19	等高线绘制软件	180
§4.21	课题20	逐步回归分析	181
第五章 上机指导			185
§5.1	IBM-PC微机DOS命令和BASIC命令		185
§5.2	行编辑程序(EDLIN)及其使用		188
§5.3	FORTRAN 77程序的编译、连接和查错		194
§5.4	应用软件Wordstar		200
§5.5	Personal Editor程序简介		215
第六章 科学计算绘图			220
§6.1	图形显示语句		220
§6.2	绘图基本概念		236
§6.3	二维曲线绘制		242
§6.4	曲面图形		247
§6.5	图形着色、存储和再现		253

第一章 科学计算程序的编制方法

在一般科学计算中，首先要将计算对象的物理过程或工作状态归纳为数学问题，即建立数学模型。其次，根据这个数学模型所确定的计算方法编制成计算机能接受的计算程序。最后，将计算程序送入计算机进行调试，执行运算，得出正确的计算结果。以上三个步骤就是一般科学计算的全过程。

本章所要论述的是从已知的数学模型和计算方法出发，如何编制高质量的科学计算程序。一般，人们只要求程序正确无误，能通过调试即可，而对程序是否容易被看懂，即程序的易读性以及是否便于人们检查和修改，即程序的易修改性是注意不够的。如往往由于程序中滥用了GOTO语句，缺乏清晰的结构，而使程序变得不易阅读和理解，增加了检查和修改程序所花的工作量，甚至使调试产生困难。随着计算机硬件的发展，内存量不断提高，程序的“可读性”已被列为评价程序好坏的重要标准。因此，在编制计算程序时，必须注意程序的质量标准和程序设计的方法，培养良好的程序设计风格，从而确保所设计的计算程序达到令人满意的效果。

§ 1.1 程序的质量标准

评价一个计算程序的质量如何，需要考虑各种因素，不同的科学计算课题所要求的标准也不一样。下面几个标准则是最重要的，它们对绝大多数程序都能作出较全面的评价。

一、正确性

正确性是评价程序质量的关键。在科学计算中，有时运用计算机来模拟物理实验，而研究对象的数学模型的建立，要根据计算结果的分析来不断完善，如果计算程序本身产生错误的结果，则该程序就毫无价值。所以首先要通过精心的程序设计和详细的检查、修改来实现程序的正确性，从而才能判断数学模型的合理性。同样，只有程序满足正确性这个条件，才能比较计算方法的优异性。

二、结构性

一个好的计算程序是按一定框架把程序的各个部分组合在一起的。这种框架称之为程序的结构。如何编制出结构清晰的程序呢？最广泛采用的方法叫模块化程序设计。模块是程序的一部分，它是用来完成一个具体任务的。可以把一个计算任务分解成若干个模块，每个模块可以独立地编写、测试、排错和修改，再把各模块“装配”在一起组成程序。这样编制的程序，结构清晰，便于理解，且可防止错误在不同模块之间扩散蔓延。衡量模块的好坏有两个标准，一个是耦合，即模块间相互联系的紧密程度；另一个是内聚性，即模块内诸内部成分相互联系的紧密和相关程度。一个好的模块要求有松散的耦合和很强的内聚性。

三、易读性

有相当长一段时期，多数人认为程序只是给计算机执行，而不是供人阅读的，程序只要逻辑正确，能被机器理解并执行就足够了，至于程序的“文体”如何，则是不必介意的。但是随着计算课题规模的扩大和复杂性的增加，人们意识到读程序是程序调试工作中的一个重要组成部分，而且读程序的时间恐怕比写程序的时间还要多。因此，程序实际上也是一种供人阅读的“文章”。一个逻辑上虽绝对正确，但杂乱无章的计算程序也是不可取的，因为它难以测试、排错和维护。“易读性”差的主要原因是由于滥用了GOTO语句，使程序的流程无规律，转来转去就象一团乱麻一样，使看

程序的人无法理解其思路，又由于程序的可读性直接影响了程序的移植、重用和维护，所以，编写计算程序时，尽可能要采用模块化程序设计，并且对符号变量的引用加以说明，即建立完整的文档，来实现易读性。

四、易改性

一个计算程序很少是一次执行就完全通过，总是需要反复调试、修改。另外，一个通过调试的计算程序不可能保持不变，几乎所有可供计算机使用的程序都需要经常修改以满足进一步的需要。所以，从很小的一段程序到全部完整的程序都要能很容易地进行不同程度的修改。如果一个计算程序是模块化结构，则就很容易检查到错误发生或要补充修改的相应模块，进行局部修改，对程序的总体结构不会产生或只产生很少的影响。

五、有效性

有效性是指计算程序精练，结构清晰，不繁琐重复。但不等于程序愈短愈好。在功能很强的计算机时代，简短性这一标准已经过时了，因为简短的程序有时无法让人看懂，很难维护。只有简单的程序结构才容易理解，容易校验，容易修改。把复杂的任务简化为简单有效的程序结构，绝不是轻而易举的。它需要具有清晰的总体结构思想，娴熟的技巧，足够的程序设计实践经验和耐心。

总之，在设计程序时，主要应考虑到正确性、结构性和易读性，强调了这些标准，必能做到计算程序的易改性和有效性。

§ 1.2 程序设计的风格

所谓程序设计风格是指使用计算机语言进行程序设计的习惯和方法，它与程序的易读性有十分密切的关系。程序不仅要由计算机来执行，还要由人来阅读，以便理解、修改扩充和维护。下面

从二个简单的例子来说明设计风格的重要性。

例1 利用幂级数

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots$$

计算 e^x 。

〔算法一〕

1. 步骤

(1) 输入X的值. 令幂次数 $N=1$, 计算级数和 $S=1$, 储存级数和 $F=0$. (这是初始部分)。

(2) 直接算出级数项 $\frac{X^N}{N!}$. ($T \leftarrow X^N$, $W \leftarrow N!$)

(3) ($N+1$)项级数和 $S \leftarrow S + T/W$.

(4) 精度EP控制. 若 $|S - F| < EP$, 则打印和值S且终止计算, 否则, $F \leftarrow S$, 且令 $N = N + 1$, 转(2)继续计算。

2. 计算程序1

```
10 REM ****1****
```

```
20 DEFDBL S,T,W,F:A=1
```

```
30 INPUT "---T1---",T1
```

```
40 X=A*T1:EP=.0001
```

```
50 N=1:S=1:K1=0:F=0
```

```
60 T=X^N:K1=K1+N-1
```

```
70 I=1:W=1
```

```
80 W=W*I:K1=K1+1
```

```
85 IF T>1.7D+38 OR W>1.7D+38 GOTO 155
```

```
90 I=I+1
```

```
100 IF I>N GOTO 120
```

```
110 GOTO 80
```

```
120 S=S+T/W:K1=K1+1
```

```
130 IF ABS(S-F)<EP GOTO 150
```

```
140 F=S:N=N+1:GOTO 60
```

```
150 PRINT"-----1-----"N,K1,S : GOTO 160
```

```
155 PRINT"----1---!!!!"N,K1,S
```

```
160 END
```

3 程序1注释

20句: DEFDBL为IBM-PC BASIC双精度变量说明. 引用变量A是为了在例2中扩充成指数矩阵函数的情况, 这里不妨取 $A = 1$.

30句至40句: 输入 T_1 , 则 $X = A \cdot T_1 = T_1$. EP 为精度控制量.

50句: 置变量初态值, K_1 是用来统计乘除法次数.

60句: 计算 X^N ($T \leftarrow X^N$).

70句至110句: 计算 $N!$ ($W \leftarrow N!$), 其中85句是给出溢出信息.

120句至140句: 执行计算步骤(3)和(4).

4 计算程序2

在程序2中, 计算 X^N 和 $N!$ 时改用重复模块语句, 如18句至28句所示. 16句, 30句和36句是外嵌套循环语句组, 用来计算 $(N+1)$ 项级数之和. 另外, 循环变量 N 的终值 M 取充分大的数, 以便能有足够的项相加, 达到计算的精度.

```
4 REM****2****
```

```
6 DEFDBL S,T,W,F : A = 1
```

```
8 INPUT"---T1---", T1
```

```
10 X = A * T1 : EP = .0001
```

```
12 M = 9000
```

```
14 S = 1 : K1 = 0 : F = 0
```

```
16 FOR N = 1 TO M
```

```
18 T = 1 : W = 1
```

```
22 FOR J = 1 TO N
```

```
24 T = T * X : W = W * J : K1 = K1 + 2
```

```

26 IF T > 1.7D + 38 OR W > 1.7D + 38 GOTO 40
28 NEXT J
30 S = S + T / W : K1 = K1 + 1
32 IF ABS(S - F) < EP GOTO 38
34 F = S
36 NEXT N
38 PRINT "----2----" N, K1, S : GOTO 42
40 PRINT "----2--1111" N, K1, S
42 END

```

在上面二个程序中，除了为控制溢出、精度和打印而使用 GOTO 语句外，“计算程序1”还要多用三条 GOTO 语句，从而使程序结构层次不清，流程转向频繁，可读性差。相反，在“计算程序2”中，由于引用了“累乘”和“累加”两个基本算法模块，避免使用 GOTO 语句，于是使程序结构清晰，内外层次分明，程序容易阅读。因此，少用 GOTO 语句，采用模块结构语句是一个良好的程序设计风格。

其次，设计风格与程序的有效性有关。要做到有效性，就是要设法减少计算工作量，提高算法的效率。例如上述“计算程序2”尽管易读，但存在两个缺点：一是每次计算级数中的某一项时，并没有利用前项的计算值，而是重新计算各项的值，所以乘法和除法的次数显然增多；二是当 X 取得适当大时，累积的值超过了机器的最大数，会产生溢出现象，如下列表1-2中所示。因此“计算程序2”还不是一个较好的程序设计。下面编制的“计算程序3”和“计算程序4”可进一步提高设计质量。

〔算法二〕

1 步骤

(1) 输入 X 值，令 $N = 1$ ，级数项值 $T = 1$ ，计算级数之和 $S = 1$ ，储存级数和 $F = 0$ 。（置初态）

(2) 递推算出级数项：

$$T \leftarrow T \cdot X / N.$$

(3) $(N+1)$ 项级数和 $S \leftarrow S + T$.

(4) 若 $|S - F| < EP$, 则打印和值 S , 终止计算. 否则 $F \leftarrow S$, 且令 $N = N + 1$, 转 (2) 继续计算.

2. 计算程序3

```

4 REM***3***
6 DEFDBL S,T,W,F:A=1
8 INPUT"----T1----",T1
10 X=A*T1:EP=.0001
12 M=9000
44 T=1:S=1:K1=0:F=0
46 FOR N=1 TO M
48 T=T*X/N:K1=K1+2
49 IF T>1.7D+38 GOTO 60
50 S=S+T
52 IF ABS(S-F)<EP GOTO 58
54 F=S
56 NEXT N
58 PRINT"-----3-----"N,K1,S:GOTO 62
60 PRINT"-----3----!!!!"N,K1,S
62 END

```

3. 程序3注释

46句至56句: 采用一个重复模块来算出 $(N+1)$ 项级数之和 S 其程序语句次序与算法步骤一致.

[算法三]

1. 步骤

(1) 输入 X 值. 令 $K=1$, $Q=5$, $F=0$, 则

$$X_1 \leftarrow X / 2^Q.$$

(2) 计算级数 $e^{\frac{x}{2Q}}$ 的 $(K+1)$ 项之和:

$$S \leftarrow 1 + \sum_{n=1}^K \frac{x_1^n}{n!} .$$

(3) 将 S 平方 Q 次, 得 e^x 的级数和.

(4) 若 $|S - F| < EP$, 则打印和值 S , 终止计算. 否则, $F \leftarrow S$, 令 $K = K + 1$, 转 (2) 继续计算.

2. 计算程序4

4 REM****4****

6 DEFDBL S,T,W,F : A = 1

8 INPUT "----T1----", T1

10 X = A * T1 : EP = .0001

12 M = 9000 : Q = 5

40 X1 = X / 2^Q : F = 0

66 FOR K = 1 TO M

68 T = 1 : S = 1 : K1 = 1

70 FOR N = 1 TO K

72 T = T * X1 / N : K1 = K1 + 2

74 S = S + T : IF T > 1.7D+38 GOTO 92

76 NEXT N : N = N - 1

78 FOR I = 1 TO Q

80 S = S * S : K1 = K1 + 1 : IF S > 1.7D+38 GOTO 92

82 NEXT I

84 IF ABS(S - F) < EP GOTO 90

86 F = S

88 NEXT K

90 PRINT "----4---- "N,K1,S : GOTO 94

92 PRINT "-----4---!!! "N,K1,S

94 END

3. 程序4注释

68句至76句：以〔算法二〕中“程序3”的语句设计来计算 $(k+1)$ 项级数之和 $e^{\frac{x}{Q}}$ 。

78句至82句：对 $e^{\frac{x}{Q}}$ 平方 Q 次，算出 e^x 值。

程序中的 M 用来保证有足够多的项相加。

由上可知，“计算程序3”和“计算程序4”皆克服了“计算程序2”的两个缺点，尤其是“计算程序4”的设计最佳，既做到易读，又使计算级数的项数 N 和乘除法次数 k_1 大大减少。这一点在表1-1和表1-2中明显可见。

表1-1

$X = 9$	级数项数 N	乘除法次数 K_1	计算结果 S (精度0.0001)
计算程序1	31	992	8103.0837
计算程序2	31	1023	8103.0839
计算程序3	31	62	8103.0839
计算程序4	9	24	8103.0839

表1-2

$X = 20$	级数项数 N	乘除法次数 K_1	计算结果 S (精度0.0001)
计算程序1	30	900	溢出
计算程序2	30	959	溢出
计算程序3	61	122	485165195.40977
计算程序4	15	36	485165195.40979

至于“计算程序4”的编制，在“计算程序3”的基础上，利用模块结构设计风格，就能很容易完成，并且，还可以利用某些算法模块，加以扩充和组合，从而能较易地编制出更复杂的算法程序。