

电子计算机 算法手册

```
PROCEDURE LAG1(N,X,Y,U,V)  
  VALUE N,U; INTEGER N; R  
BEGIN INTEGER I,J,L;  
  V:=0;  
  FOR I:=0 STEP 1 UNT  
  BEGIN L:=1;  
    FOR J:=0 STEP  
    IF J=I THEN L  
    V:=V+L*Y[I]  
  END  
END LAG1;
```

电子计算机算法手册

上海计算技术研究所

上海教育出版社

电子计算机算法手册

上海计算技术研究所

上海教育出版社出版

(上海永福路 123 号)

新华书店上海发行所发行 江西印刷公司印刷

开本 850×1156 1/32 印张 33.5 插页 4 字数 1,014,000

1982 年 12 月第 1 版 1982 年 12 月第 1 次印刷

印数 1—16,300 本

统一书号: 7150·2307 定价: (精)4.45 元

序 言

随着我国国民经济和科学技术的发展,电子计算机的使用日益广泛.作为电子计算机应用的一个重要方面——科学计算也日新月异地迅速发展.科学计算中所使用的计算方法及其程序(简称算法)一方面以计算机作计算工具,另一方面以数学(特别是计算数学)作理论依据.所以近来人们称它为电子计算机的数学软件(也有人称数值软件,当然数学软件指的范围更广,不只限于算法).它不断地被完善和标准化,使之能有效地解决各种问题.同时,为了解决各领域提出来的越来越大、越来越复杂的问题,人们也在大力研究、发明各种新的算法.算法一经提出,并证明为有效后就被大量地、广泛地、重复地使用.为了普及和发展计算机的应用技术,提高机器使用效率,缩短解题周期,减少重复劳动,发展计算方法,我们编写了本书.

编写本书的指导思想是:兼顾算法的数学理论和实现程序,偏重于程序;尽量汇编新的算法,也选编了有效的经典算法;以求解大量常见问题的算法为主,也考虑了解决特殊问题的算法;既参考国内外的已有算法,又吸收了我们实际工作中的经验;各个数学分支的算法力求齐全.

本书共分十四章.第一章数值逼近,包括经典的方法及近十几年才发展起来的如样条函数等方法.它们能解一、二维插值、拟合、光顺等问题.多年的使用证明它们是有效的.

第二章数值求积,大致可分为辛普生型积分、龙贝格型积分,高斯型积分.CCQUAD是较新的有效求积方法.若被积函数在积分区间内激烈振荡,则可用菲隆方法.

快速傅里叶变换(FFT)的应用已引起了人们的普遍重视,发展极快.在第三章中给出了实数据和复数据的一维和二维的快速傅里叶变换算法.

第四、五两章是线性代数算法,包括解线性代数方程组,求最小二乘解,求矩阵的逆和行列式值,解标准和广义特征值问题.矩阵可以是实

阵、复阵或病态阵,满阵或稀疏阵,对称阵或非对称阵以及某类特殊阵。为了尽力扩大使用范围,除采用了各种经典的和较新的数学方法外,还涉及了使用外存贮器技术,此外还给出了一些和有限元素法有关的算法。

第六、七、八章是求多项式(实的、复的或病态的)零点、解超越方程、解非线性方程组和最优化问题。解超越方程的方法也可用于求单变量函数的极值。它们大致可分成区间套方法和迭代法两类,可求单根或复根,求一个根或求某区间上的全部根,可解实方程或复方程。第八章的最优化算法大多是用来解无约束极值问题。罚函数法可解有约束极值问题。

第九章规划问题和第十章图论都偏重于解一些运筹学问题。前者解线性规划,互补性问题、运输问题、分配问题和计划评审。后者用于解运输、通讯和电路等网络中的诸如最短距离、最低造价等问题。

第十一章解常微分方程组初值问题,除了解一般方程组外,吉尔方法和埃尔米特插值法都可解刚性方程组。

第十二章概率统计计算,包括计算常见分布的随机数、分布函数和分位数,对时间序列进行估计分析,对实验数据进行回归分析,方差分析和识别分析等几方面。

第十三章的算法主要是用逼近方法计算常用的特殊函数。

第十四章分类和检索属数据处理算法,它们都不用外存贮器,只对数据进行加工。

本书共有 207 个算法,每个算法大致分功能、方法、使用说明(包括过程标识符和形式参数表、说明、例子,使用较复杂的还包括考试程序)和过程等部分,最后附上参考文献。在功能部分,将简要地说明算法所能解决的数学问题、算法的适用范围和主要性能。在方法上,经典的或熟知的算法的叙述从简,新的或特殊的算法的叙述稍详;算法的性质和定理的证明一般不给出,但算法的思想、公式、步骤和结论则均有介绍。若要详细了解,可读参考文献;在使用说明中,对各形式参数的意义和调用过程时要注意的事项作了说明。例子的选取既为考试和核对,有时还用来比较各方法的优劣,说明算法的特殊功能。

算法的程序用 ALGOL 60 语言编写。这是考虑到我国目前大多数计算机所使用的算法语言仍是 ALGOL 60 或 ALGOL 60 的变种之故。由于过程是在 709 机上考试通过的,所以本书的过程和考试程序除了尽量

用标准 ALGOL 60 语言外,个别地方可能受到 709 机语言的限制。当然用户可根据所使用的计算机容易地将它们加以改变。为了使用户能灵活地编写输入输出语句,本书的输入输出语句对格式不予考虑。例如,如果 a 是说明了的标识符,则 $read(a)$ 和 $print(a)$ 表示输入和输出 a 的内容。否则后者表示输出字符 a 。

本书的程序和过程中用到了一些标准函数,其名称和意义见下表。

标准函数	数 学 含 义	标准函数	数 学 含 义
$abs(E)$	E 的绝对值	$exp(E)$	E 的指数函数
$entier(E)$	不大于 E 的最大整数	$sqrt(E)$	E 的算术平方根, $E \geq 0$
$sign(E)$	E 的符号,即 $sign(E)$ $= \begin{cases} 1 & \text{若 } E > 0 \\ 0 & \text{若 } E = 0 \\ -1 & \text{若 } E < 0 \end{cases}$	$cubrt(E)$	E 的立方根
$\sin(E)$	E 的正弦值	$ranum(E)$	产生一个 $[0,1]$ 中均匀分布的伪随机数,这里 E 只有形式上的作用,可任取一值
$\cos(E)$	E 的余弦值	$ranump(E)$	将 $ranum(E)$ 的伪随机数置成初态,然后转下一 $ranum(E)$ 取得伪随机数,其中 E 只有形式上的作用,可任取一值
$tg(E)$	E 的正切值		
$\arcsin(E)$	E 的反正弦值, $ E \leq 1$		
$\arctg(E)$	E 的反正切值		
$\ln(E)$	E 的自然对数, $E > 0$		

参加本书编写工作的主要有冯伯培、江善标、刘万勋、张国彬、严洪范、杨明三同志;张宝康、江乃雄、裴广生、于春生、陈淦诚、李庆芝、华伯浩、郑家栋、施政文、顾鼎铭、陈培华、刘长学、张汝芬、孙鸿君、钟银珠、叶瀚、邓礼武、张明廉、吴幼贞、朱爱芳、毛乾霖、许言午、范力、王金汉、华孝先、熊明光等同志也参加了编写工作;图论一章中的十一个算法由上海铁道学院楼世博、李鸿祥、孙洲、葛燧和同志编写;倪素文、胡志琴、张建明等同志承担了部分工作。最后由冯伯培和江善标修改定稿。

由于我们水平所限,加上时间仓促,本书难免错误和不足,恳请同志们批评指正。

目 录

序 言

第一章 数值逼近	1
1.1 拉格朗日一元 n 点插值	2
1.2 一元三点插值	3
1.3 拉格朗日二元 n 点插值	5
1.4 二元三点插值	7
1.5 埃尔米特插值	9
1.6 三次和三次周期样条函数插值、微商、积分	11
1.7 三次自然样条函数插值、微商、积分	20
1.8 用局部数据的三次曲线拟合法	25
1.9 等距节点正交多项式曲线拟合	29
1.10 曲线拟合的数值磨光方法	32
1.11 样条函数插值和光顺	35
1.12 曲面样条函数	41
1.13 双三次样条函数	47
1.14 双圆弧样条曲线	52
第二章 数值求积	61
2.1 变步长辛普生求积	62
2.2 自动调节步长辛普生求积	63
2.3 龙贝格求积	66
2.4 带误差界的龙贝格求积	69
2.5 高斯型求积	72
2.6 带误差估计参考量的高斯-克朗罗德(Kronrod)积分	81
2.7 克伦肖-柯特斯(Clenshaw-Curtis)积分	84
2.8 计算振荡函数积分的非隆方法	91

2.9	用高斯求积公式计算重积分	96
2.10	求三维单位球体上的函数的积分	102
第三章	快速傅里叶变换和复运算	105
3.1	复数据快速傅里叶变换算法(一)	105
3.2	复数据快速傅里叶变换算法(二)	111
3.3	复数据快速傅里叶变换算法(三)	116
3.4	实数据快速傅里叶变换算法(一)	122
3.5	实数据快速傅里叶变换算法(二)	126
3.6	复数据二维快速傅里叶变换算法	129
3.7	复数的模	133
3.8	复数除法	134
3.9	复数的根	135
3.10	复数的实幂指数	137
3.11	复数的复幂指数	138
3.12	复变量自然对数	139
第四章	解线性代数方程组、求最小二乘解、求矩阵的逆阵和行列式值	142
4.1	全主元高斯-约当(Jordan)消去法	144
4.2	LU 分解法	146
4.3	逐行主元消去法	150
4.4	乘积形逆阵方法	154
4.5	解复阵方程的 LU 分解法	159
4.6	解方程组、求逆、求行列式值的加边法	164
4.7	解带型线性方程组的列主元高斯消去法	170
4.8	解对称及非对称带型方程组的局部选主元方法	172
4.9	解对称方程组的顺序高斯消去法	182
4.10	解对称方程组的改进平方根法	184
4.11	解对称方程组的豪斯霍尔德变换法	187
4.12	解对称方程组的块对角主元素法	193
4.13	解病态对称方程组的直接-迭代校正法	199

4.14	解对称正定方程组(兼求行列式值)的平方根法	204
4.15	解对称正定方程组的改进平方根法	210
4.16	解对称正定带型方程组的顺序消去法	213
4.17	解对称正定带型方程组的平方根法	215
4.18	解对称正定带型方程组的改进平方根法	218
4.19	解对称正定方程组的共轭斜量法	220
4.20	超松弛法	224
4.21	解中心对称带型方程组的对称高斯消去法	227
4.22	解三对角方程组的追赶法	234
4.23	解拟三对角方程组的直接法	236
4.24	解变带宽对称正定方程组的改进平方根法	240
4.25	解高阶稀疏对称正定方程组的变带宽法	242
4.26	解非对称稀疏方程组的波阵法	253
4.27	结点近似最佳编序算法	266
4.28	带宽极小化法(RCM)	272
4.29	求实满秩矩阵的逆阵的主元消去法	276
4.30	消秩法求逆的快速紧凑格式	283
4.31	求对称阵逆阵的消去法	286
4.32	求对称正定矩阵的逆阵的消去法	288
4.33	求对称正定矩阵的逆阵的改进平方根法	292
4.34	求一般实阵行列式值的行主元三角化方法	295
4.35	求对称阵行列式值的改进平方根法	297
4.36	求实对称正定带型矩阵行列式值的平方根法	299
4.37	求最小二乘解的豪斯霍尔德变换法	301
4.38	双对角化法	312
4.39	奇异值分解和最小二乘解	319
第五章	解特征值问题	336
5.1	求实对称矩阵特征值的雅可比方法	337
5.2	解实对称矩阵的标准或广义特征值问题的隐式QL 算法	342

5.3	计算实对称矩阵部分特征值的二分法——反迭代法	350
5.4	求实对称矩阵的标准或广义部分特征值与特征向量的子空间迭代法	356
5.5	求复对称矩阵的特征值和特征向量的雅可比方法	370
5.6	求实矩阵的部分特征值和特征向量的幂法和穷举法	380
5.7	求实矩阵全部特征值和特征向量的 QR 方法	383
5.8	求复矩阵全部特征值和特征向量的 LR 方法	403
5.9	求带阵部分特征值和特征向量的反迭代法	413
第六章	求多项式的根	421
6.1	用霍纳法计算实多项式及其导数的值	424
6.2	解二、三、四次多项式方程的直接法	426
6.3	求实系数多项式实根的切线法	432
6.4	求实系数多项式实根的割线法	435
6.5	求实系数多项式根的劈因子法(一)	438
6.6	求实系数多项式根的劈因子法(二)	442
6.7	解实多项式方程的根平方-子结式方法	446
6.8	求复系数多项式零点的牛顿法	460
6.9	求复系数多项式零点的拟线性逼近法	463
6.10	求复系数多项式零点的三步变量移位法	470
第七章	解超越方程	481
7.1	二分法	482
7.2	插值和二分法的结合方法(一)	483
7.3	插值和二分法的结合方法(二)	487
7.4	三种改进的割线法	493
7.5	改进的佩加苏方法	497
7.6	导数估值的带存贮的一点迭代方法	499
7.7	多点迭代法	503
7.8	求函数零点的米勒(Muller)方法	506
7.9	求函数零点的钱伯斯(Chambers)方法	515
7.10	求复函数方程的根的下山法	522

第八章 解非线性方程组和最优化	527
8.1 用差商代替导数的牛顿法	528
8.2 布罗伊登方法	531
8.3 布罗伊登-戴维登科方法	535
8.4 正交三角化牛顿法	541
8.5 阻尼最小二乘法	546
8.6 单纯形法	552
8.7 正交化法	555
8.8 秩二变尺度法	561
8.9 二参数变尺度法	569
8.10 用差商(自动选步长)计算导数和梯度	574
8.11 解非线性规划的罚函数(SUMT)法	577
第九章 规划问题	616
9.1 解线性规划的用乘数的单纯形法	616
9.2 解线性规划的 LU 分解法	622
9.3 解 0-1 变量线性规划的隐算法	633
9.4 解互补性问题	638
9.5 解运输问题的原始-对偶方法	643
9.6 解分配问题	653
9.7 计划评审计算	659
9.8 拓扑顺序计算	662
第十章 图论	666
10.1 求线图中所有两顶点间的最短通路值	667
10.2 求线图中部分顶点对间的最短通路	672
10.3 求线图中一定点到各顶点间的最短通路	675
10.4 求线图中两点间的最短通路	678
10.5 求无向图的最优树	682
10.6 求无向连通图的支撑树	685
10.7 求无向图的一棵支撑树或一片森林	688

10.8	求无向图的森林	690
10.9	求无向连通图的基本回路	693
10.10	最大网络流	696
10.11	求网络的最大流	702
10.12	最小运费网络流	706
10.13	求二分图的最大对集	714
10.14	求赋权完备二分图的最优对集	720
第十一章	解常微分方程(组)初值问题	727
11.1	定步长改进欧拉法	728
11.2	定步长吉尔法	729
11.3	变步长龙格-库塔法	731
11.4	四阶、六阶、十阶龙格-库塔法	733
11.5	库塔-墨森(Merson)法	743
11.6	墨森法	745
11.7	哈明(Hamming)方法	749
11.8	外推法	752
11.9	解刚性(Stiff)常微分方程初值问题的埃尔米特插值 型法	758
11.10	解一般和刚性常微分方程初值问题的吉尔方法	764
第十二章	概率统计计算	788
12.1	满足均匀分布的随机数的产生(一)	789
12.2	满足均匀分布的随机数的产生(二)	790
12.3	满足正态分布的随机数的产生	792
12.4	满足普阿松分布的随机数的产生	793
12.5	正态分布的分布函数	796
12.6	正态分布的分位数	799
12.7	二项分布的分布函数	801
12.8	普阿松分布的分布函数	804
12.9	χ^2 分布的分布函数	806
12.10	χ^2 分布的分位数	809

12.11	t 分布的分布函数	811
12.12	t 分布的分位数	815
12.13	F 分布的分布函数	817
12.14	F 分布的分位数	821
12.15	数学期望和自协方差函数值估计	824
12.16	数学期望和互协方差函数值估计	826
12.17	时间序列的线性平滑	828
12.18	时间序列的三重指数平滑	832
12.19	多元三角回归分析	836
12.20	逐步回归分析	846
12.21	非线性参数估计	855
12.22	单因素方差分析	861
12.23	多因素方差分析	864
12.24	最短距离法	871
12.25	主成分分析	874
12.26	非线性映照法	879
12.27	逐步判别分析	892
第十三章	特殊函数	908
13.1	指数积分	908
13.2	定指数积分(一)	910
13.3	定指数积分(二)	911
13.4	正弦积分、余弦积分和非涅耳积分	914
13.5	计算第一、二类全椭圆积分的多项式逼近法	920
13.6	计算第一、二、三类全椭圆积分	921
13.7	Γ 函数	924
13.8	Γ 函数的自然对数	925
13.9	Γ 函数的倒数	927
13.10	零阶和一阶第一、二类贝塞尔函数	929
13.11	零阶和一阶第二类变形贝塞尔函数	934
13.12	球诺伊曼(Neumann)函数	936

13.13	第一类分数阶贝塞尔函数和变形贝塞尔函数	938
13.14	实误差函数	945
13.15	大 x 的误差函数	946
13.16	大 x 的余误差函数	948
13.17	高斯误差函数	949
13.18	复误差函数	952
13.19	正交多项式	956
13.20	第一类实元或虚元的连带勒让德函数	961
第十四章	分类与检索	967
14.1	二次插入分类	968
14.2	地址计算分类	969
14.3	Shell 分类	971
14.4	Shuttle 分类	973
14.5	部分分类	974
14.6	最小存储分类	977
14.7	堆分类	979
14.8	树分类(一)	982
14.9	树分类(二)	984
14.10	目录合并分类	985
14.11	快速分类与直插结合法	988
14.12	顺序检索	992
14.13	二次树检索	995
附表一	高斯-勒让德积分的结点和权系数	1000
附表二	拉登积分的结点和权系数	1006
附表三	洛巴多积分的结点和权系数	1015
附表四	埃尔米特积分的结点和权系数	1021
附表五	拉盖尔积分的结点和权系数	1028
附表六	高斯求积公式 $G(f)$ 和改进高斯求积公式 $K(f)$ 的结点和权系数	1039
附表七	以 x^2 为权的高斯型求积公式的结点和权系数	1049
附表八	书内外国人姓氏中外文对照表	1053

第一章 数值逼近

本章共有十四个算法。第一、三个分别是一、二维经典拉格朗日 (Lagrange) 插值。它们计算简便, 但用于高次插值时会产生令人讨厌的龙格 (Runge) 现象, 即节点处满足插值要求而节点间有时会出现振荡, 因此实用时通常以不超过五、六次为宜。为避免出现振荡, 可以采用低次多项式进行分段逼近。第二、四个算法就是一维分段和二维分块的二次抛物插值。它们不仅简便, 而且逼近效果往往优于高次拉格朗日插值, 不足之处是仅保证函数的连续性, 在节点处导数不连续, 所以比较适用于一般光滑性不高、计算量较少的场合。若导数连续性也要予以保证, 则可选用样条 (Spline) 函数方法。若实际问题不仅在节点处给出了函数值而且给出了它的一阶导数值, 则可用第五个埃尔米特 (Hermite) 插值算法。

近十几年来, 样条函数的理论和应用有了很大发展。本章从第六个算法开始, 除第九个多项式拟合外都是与样条函数有关的算法, 用户可根据这些算法的特点和实际问题的要求合理地选用。第六个算法中包含两个过程。前一个即常用的根据三弯矩方程建立的三次样条函数, 端点边界条件可有三种给定方法, 算法除插值外还能计算微商和积分值; 后一个是它在周期情况的应用。第七个算法是三次自然样条函数, 相当于端点二阶导数为零, 它适用于边界条件无法给出而可作自然处理的问题。在某些情况下, 比如逼近某段是“曲”而另一段是“直”的函数曲线时, 样条函数往往也会产生不应有的显眼的拐曲。有各种方法可避免这种情况, 如第八个算法采用了二步法: 先由局部数据点用最小平方拟合方法估算节点的导数, 然后在每两个相邻节点间用三次埃尔米特插值求所需的插值函数。虽然这时在节点处只保证了一阶导数的连续性, 然而这样往往能避免一些不应有的拐曲。第十个算法是一种适用于外形设计的能够更好地适应保凹凸性要求的曲线拟合方法, 即常称的 B 样条。这里的磨光算法

不是插值方法,它不保证通过数据型值点,但通过盈亏修正可以达到较好的逼近精度,故是外形设计和数据处理的行之有效的简便方法。第十一个算法既能用于插值,又能在一定逼近精度范围内进行数值光顺,这只需通过参数的适当选取便能实现;另外端点边界条件是自然的,不必提供;实践表明它是个很有效的算法。在生产实践和计算机绘图中圆弧样条已日益广泛应用,第十二个算法圆弧样条由两部分构成,先由数据点用局部三次样条估算型值点的切线方向,然后在相邻数据点之间用分角线方法作出双圆弧插补,算法具有保凹凸性。第十三、十四个是二维样条函数。前者在平面数据点任意分布的情况下得到整个任意阶可微的光滑曲面,后者在数据点为平面矩形节点分布情况下得到分块双三次曲面,具有二阶连续可微性,因它们都采用了自然边界条件,故边界条件不必另行给定。

1.1 拉格朗日一元 n 点插值

一、功能

给定互不相等的一元函数插值节点 $x_i (i=0, 1, \dots, n)$ 及相应的函数值 $y_i (i=0, 1, \dots, n)$, 用拉格朗日插值多项式求插值自变量 u 所对应的函数值 v 。

二、方法

对于给出的插值自变量 u , 求对应函数值 v 的公式为

$$v = \sum_{i=0}^n \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(u-x_j)}{(x_i-x_j)} y_i.$$

三、使用说明

(一) 过程标识符和形式参数表

LAG1(n, x, y, u, v);

n 插值节点数减 1;

x, y 数组 $x[0:n], y[0:n]$, 分别存放插值节点值及相应函数值;

u, v 分别为插值自变量和插值结果。

(二) 例子

x	y	u_k	v_k
-2	24	-2	24
		-1.5	5.625
		-1	$-0.1818989404 \times 10^{-11}$
-0.4	-0.2688	-0.4	-0.2688
-0.2	-0.0768	-0.2	-0.0768
		0	$0.4547473509 \times 10^{-12}$
		0.4	-0.2688
		0.8	-0.4608
1	0	1	0
		1.5	5.265
4	480	4	480

四、过程

```

procedure LAG1( $n, x, y, u, v$ );
  value  $n, u$ ; integer  $n$ ; real  $u, v$ ; array  $x, y$ ;
begin integer  $i, j, l$ ;
     $v := 0$ ;
    for  $i := 0$  step 1 until  $n$  do
      begin  $l := 1$ ;
        for  $j := 0$  step 1 until  $n$  do
          if  $j \neq i$  then  $l := l \times (u - x[j]) / (x[i] - x[j])$ ;
           $v := v + l \times y[i]$ 
        end
      end LAG1;

```

1.2 一元三点插值

一、功能

给定互不相等的一元函数插值节点 $x_i (i=0, 1, \dots, n)$ 及相应函数值 $y_i (i=0, 1, \dots, n)$, 用一元三点插值多项式求取插值自变量 u 所对应的函数值 v .