

08345 5614687

战同胜 主编

实用数值算法

123456789
1234567890

567890

1234567890
1234

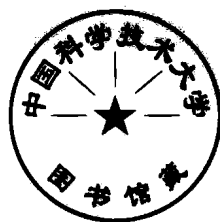
大连理工大学出版社

2

电子计算机应用数学

实用数值算法

主 编 战同胜
副主编 郭照隆 宋广仁
纪青君 丁 瑞



大连理工大学出版社

(辽)新登字 16 号

内容简介

本书是电子计算机的工具书,同时又是学习对数学模型进行离散化、算法化和程序化的一本简要参考书。全书选编了 63 个常用数值算法,内容较新且实用。每个算法都对数学方法作了简述,按结构优化思想编制了通用算法,给出数值例题和少量练习题,每个练习题都有答案。这些算法包括了现代数值计算的基本内容,具体请参看本书目录。

本书面向上机计算需要编制算法与程序设计的广大读者,适用于大学生、研究生、大中专教师、科研人员、工程技术人员和计算工作者,也适用于从事计算机应用和软件开发人员的使用。

实 用 数 值 算 法

Shiyong Shuzhi Suanfa

战同胜 主编

大连理工大学出版社出版发行 (邮政编码: 116024)

大连理工大学印刷厂印刷

开本: 787×1092 $\frac{1}{32}$ 印张: 8.125 字数: 172 千字

1992 年 1 月第 1 版 1992 年 1 月第 1 次印刷

印数: 0001—1500 册

责任编辑: 王佳玉

封面设计: 羊 戈

责任校对: 杜祖诚

ISBN 7-5611-0531-2/TP·33

定价: 4.50 元

序 言

当前,随着计算机应用的深入,数值算法普遍受到关注。由于计算科学和计算技术的飞快发展,使计算机型号和计算机语言多种多样,这就为编制计算程序带来许多困难;就是给出出现成程序语言汇编,也适应不了实际中灵活多变的科学与工程出现的计算。鉴于这种情况,本书编写了通用优化数值算法。把数值计算中经常遇到的典型的数学问题,进行离散化和算法化——用接近于语句形式编制成实用性较强的算法,指出上机计算的目的、输入和输出内容以及编程序的基本步骤等,为读者编程电算提供清晰思路 and 良好基础。这些算法通用性很强,它们适用于任何机型和语言,不管读者掌握哪种计算机语言(BASIC、FORTRAN、COBOL、PASCAL 和 PL/1, ALGOL₆₈等),在算法的指导下,都能比较容易地编制出具体计算机程序来。为使非数学专业人员能较好理解算法的基本原理,看清算法的结构,了解算法的优缺点、适用范围和局限性,本书在每个算法之前对其数学方法都进行了通俗易懂的简述和评述。

有高质量的算法,才能编制出高质量的程序,才能降低软件成本,从整体上提高经济效益。高质量算法应具有结构优化、容易阅读和理解。本书提供的算法都是在这种思想指导下选编的。它们都具有书写简练,清晰易懂;结构紧凑,运算时间少;内存压缩到合理的范围;尽量做到收敛速度快、精度高和

计算稳定等特点。这些算法在方法上具有通用性；内容上具有现代性；结构上具有优化性；书写上具有易读性；使用上具有方便性。

本书由战同胜主编、主审和统一定稿；参加编写的有郭照隆、宋广仁、纪青君和丁瑞同志。

编 者

1991 年 10 月于大连

目 录

第一章 插值算法	1
§ 1.1 Lagrange 插值算法	1
§ 1.2 Neville 迭代插值算法	4
§ 1.3 Newton 插值均差公式算法.....	7
§ 1.4 Hermite 插值算法	10
§ 1.5 三次自然样条算法.....	14
§ 1.6 三次固定样条算法.....	18
第二章 逼近与拟合算法	23
§ 2.1 多项式曲线拟合算法.....	23
§ 2.2 正交多项式曲线拟合算法.....	26
§ 2.3 最佳平方逼近求值算法.....	29
§ 2.4 快速 Fourier 变换(FFT)算法	32
第三章 求积算法	37
§ 3.1 Simpson 复化算法	37
§ 3.2 自适应的求积算法.....	39
§ 3.3 Romberg 算法.....	43
§ 3.4 Gauss 算法	45
§ 3.5 重积分的 Simpson 算法.....	49
第四章 超越方程求根算法	53
§ 4.1 逐次分半算法.....	53
§ 4.2 定点迭代算法.....	55
§ 4.3 Newton 算法	57

§ 4.4	割线算法	59
第五章	求多项式零点算法	61
§ 5.1	Horner 算法	61
§ 5.2	多项式方程求根的 Newton 算法	62
§ 5.3	多项式方程求根的 Müller 算法	65
第六章	矩阵的分解算法	69
§ 6.1	矩阵的 LU 分解算法	69
§ 6.2	矩阵的 LL^T 分解算法	72
第七章	线性方程组求解的直接算法(一般矩阵)	75
§ 7.1	Gauss 消去算法	75
§ 7.2	列主元消去算法	78
§ 7.3	标度化的列主元消去算法	81
§ 7.4	直接分解算法	84
第八章	线性方程组求解的直接算法	89(特殊矩阵)
§ 8.1	三对角方程组的追赶算法	89
§ 8.2	对称正定方程组的平方根算法	92
§ 8.3	改进的平方根算法	96
§ 8.4	共轭斜量算法	99
第九章	线性方程组求解的迭代算法	104
§ 9.1	Gauss-Seidel 迭代算法	104
§ 9.2	逐次超松弛(SOR)迭代算法	106
§ 9.3	病态方程组的迭代求精算法	109
第十章	计算矩阵特征值和特征向量的迭代算法	113
§ 10.1	幂法算法	113
§ 10.2	对称矩阵的幂法算法	116
§ 10.3	反迭代法算法	118

§ 10.4	压缩算法	121
第十一章	计算矩阵特征值和特征向量的变换算法	125
§ 11.1	Householder 算法	125
§ 11.2	QL 算法	128
第十二章	非线性方程组求解算法	137
§ 12.1	Newton 算法	137
§ 12.2	Broyden 算法	140
第十三章	常微分方程初值问题求解的单步算法	143
§ 13.1	Euler 算法	143
§ 13.2	改进的 Euler 算法	145
§ 13.3	Runge-Kutta 算法	148
§ 13.4	Runge-Kutta-Fehlberg 算法	151
第十四章	常微分方程初值问题求解的多步算法	156
§ 14.1	Adams 定步长预测-校正算法	156
§ 14.2	Adams 变步长预测-校正算法	159
§ 14.3	外推算法	164
第十五章	常微分方程组与 Stiff 方程求解算法	169
§ 15.1	常微分方程组与高阶方程的求解算法	169
§ 15.2	Stiff 方程求解算法	175
第十六章	常微分方程边值问题求解算法	179
§ 16.1	线性问题的打靶算法	179
§ 16.2	非线性问题的打靶算法	183
§ 16.3	线性问题的有限差分算法	188
§ 16.4	非线性问题的有限差分算法	192
§ 16.5	三次样条的 Rayleigh-Ritz 算法	197
第十七章	偏微分方程求解算法	203

§ 17.1	Poisson 方程的求解算法	203
§ 17.2	热传导方程的求解算法	209
§ 17.3	波动方程的求解算法	213
第十八章	无约束最优化算法	219
§ 18.1	变步长梯度算法	219
§ 18.2	FR 共轭梯度算法	221
§ 18.3	Partan 算法	225
§ 18.4	DFP 变尺度算法	227
练习题答案		231

第一章 插值算法

§ 1.1 Lagrange 插值算法

(一) 数学方法简述

设给定数据点 (x_i, f_i) , $i=0, 1, \dots, n$, 求不超过 n 次插值多项式 $p_n(x)$, 使之满足插值条件

$$p_n(x_i) = f(x_i) = f_i, \quad i = 0, 1, \dots, n$$

按下面构造性方法, 作出 Lagrange 插值多项式:

作基函数

$$\begin{aligned} L_i(x) &= \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} \\ &= \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \end{aligned}$$

则 $L_i(x)$ 是一个不超过 n 次多项式, 且满足

$$L_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad i, j = 0, 1, \dots, n$$

以此基函数作多项式

$$\begin{aligned} p_n(x) &= L_0(x)f_0 + L_1(x)f_1 + \cdots + L_n(x)f_n \\ &= \sum_{i=0}^n L_i(x)f_i \end{aligned}$$

$$= \sum_{i=0}^n f_i \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (1.1)$$

多项式 (1.1) 就是所求的 Lagrange 插值多项式。

方法评述：Lagrange 插值多项式简单、直观、对称。在构造上它是一个二重循环：内循环对 j 累积求基函数 $L_i(x)$ ，外循环对 i 累加求插值结果；它极容易上机实现求函数 $f(x)$ 在某点的近似值，应用很广。它的缺点是：每当增加一个节点时，计算过的函数 $L_i(x)$ 全部作废，还得重新计算，在这一点上它不如 Newton 插值。

(二) 通用算法

Lagrange 插值算法

目的：对数据表 (x_i, f_i) , $i=1, \dots, n$, 其中 $x_1 < x_2 < \dots < x_n$, 以及给定插值点 t , 构造 Lagrange 插值多项式 $p(x)$, 求出 t 对应的函数值 $p(t)$ 。

输入：整数 n , 数据表 (x_i, f_i) , $i=1, \dots, n$, 插值点 t 。

输出： $p(t)$ 的值 F_t 。

计算步骤：

1. 对 $i=1, \dots, n-1$ 做 2.
2. 如果 $x_i \geq x_{i+1}$ 那么
输出数据不合要求的信息。
过程终。
3. 对 $i=1, \dots, n$ 做 4. ~ 8.
4. 置 $A=1.0$; $F_i=0.0$
5. 对 $j=1, \dots, n$ 做 6. ~ 7.
6. 如果 $j \neq i$ 做 7.
7. 置 $A = A \cdot \frac{(t-x_j)}{(x_i-x_j)}$

8. 置 $F_t = F_t + A \cdot f_i$

9. 输出 t 对应的函数值 F_t .

过程终。

(三) 数值例题

给定数值表

x_i	0.2	0.4	0.6	0.8	1.0
$f(x_i)$	0.9798652	0.9177710	0.8080348	0.6386093	0.3843735

试构造 1 次、2 次、3 次和 4 次 Lagrange 插值多项式求 $f(0.5)$ 的值。

解 输入 $n=2, n=3, n=4, n=5$; 数值表中使用的值 x_i 和 $f(x_i)$ (见下表); 插值点 $t=0.5$, 打印的数值结果, 见下表。

使用的点	次 数	逼近值
0.4, 0.6	1	0.8629029
0.4, 0.6, 0.2	2	0.8688582
0.4, 0.6, 0.2, 0.8	3	0.8696111
所有点	4	0.8693047

(四) 练习题

1. 构造 1 次、2 次、3 次和 4 次的 Lagrange 插值多项式求 $f(0.2)$ 的近似值。给定的数据点为

$$f(0.1) = 1.2314028, \quad f(0.3) = 1.9121188,$$

$$f(0.4) = 2.3855409, \quad f(0.5) = 2.9682818,$$

$$f(0.6) = 3.6801196$$

2. 使用下列数据点构造 4 次 Lagrange 插值多项式求 $f(1.25)$ 的近似值。

$$f(1.0) = 1.00000, \quad f(1.1) = 1.23368,$$

$$f(1.2) = 1.55271, \quad f(1.3) = 1.99372,$$

$$f(1.4) = 2.61170$$

并同 $f(x) = e^{x^2-1}$ 比较。

§ 1.2 Neville 迭代插值算法

(一) 数学方法简述

设 f 是定义在 $x_0, x_1, \dots, x_n$ 上的函数, $m_1, m_2, \dots, m_k$ 是 k 个不同的整数, 且对每个 i , $0 \leq m_i \leq n$, 记 $p_{m_1, m_2, \dots, m_k}$ 表示 f 在 k 个点 $x_{m_1}, x_{m_2}, \dots, x_{m_k}$ 上的次数不超过 k 的 Lagrange 插值多项式。

Neville 插值多项式记为 $Q_{i,j}, i \geq j$, 它表示在 $j+1$ 个节点 $x_{i-j}, x_{i-j+1}, \dots, x_{i-1}, x_i$ 上的 j 次插值多项式, 计算公式为

$$Q_{i,j}(x) = \frac{(x - x_{i-j})Q_{i,j-1}(x) - (x - x_i)Q_{i-1,j-1}(x)}{x_i - x_{i-j}} \quad (1.2)$$

其中

$$Q_{i,j-1} = p_{i-j+1, \dots, i-1, i} \text{ 和 } Q_{i-1,j-1} = p_{i-j, i-j+1, \dots, i-1}$$

$$Q_{i,j} = p_{i-j, i-j+1, \dots, i-1, i}$$

$j=1, 2, 3, \dots$ 和 $i=j, j+1, \dots$, 且 $Q_{i,0} = f(x_i)$ 。

Neville 算法的计算流程见下表

x_0	$Q_{0,0}$				
x_1	$Q_{1,0}$	$Q_{1,1}$			
x_2	$Q_{2,0}$	$Q_{2,1}$	$Q_{2,2}$		
x_3	$Q_{3,0}$	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$	
x_4	$Q_{4,0}$	$Q_{4,1}$	$Q_{4,2}$	$Q_{4,3}$	$Q_{4,4}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

当 $|Q_{i,i} - Q_{i-1,i-1}| < \varepsilon$ (给定的精度) 停止计算。

方法评述: Neville 插值比 Lagrange 插值更为实用。从 Neville 插值的计算流程表来看, 每增加一个节点, 计算只增加一行。因此, 如精度不满足要求再增加一个节点时, 前面的所有计算完全有效。这个方法也适用于电算, 程序比较简单。

(二) 通用算法

Neville 插值算法

目的: 对数据表 (x_i, f_i) , $i=1, \dots, n$, 其中 $x_1 < x_2 < \dots < x_n$, 用 Neville 方法求插值点 t 处的函数值。

输入: 整数 n ; 数据表 (x_i, f_i) $i=1, \dots, n$, 容许误差 TOL, 插值点 t 。

输出: t 对应的函数值 f_t 或方法失败信息。

计算步骤:

1. 对 $i=1, \dots, n-1$ 做 2.
2. 如果 $x_i \geq x_{i+1}$ 那么
输出数据不合要求的信息,
过程终.
3. 对 $j=2, \dots, n$ 做 4. ~7.
4. 置 $j_1 = j-1$
5. 对 $i=n, n-1, \dots, j$

$$\text{置 } Q_i = \frac{(t-x_{i-j_1}) \cdot f_i - (t-x_i) \cdot f_{i-1}}{x_i - x_{i-j_1}}$$

6. 如果 $|Q_i - f_i| < \text{TOL}$ 那么

置 $f_i = Q_i$;

输出 f_i ; 过程终。

7. 对 $i=j, \dots, n$, 置 $f_i = Q_i$;

8. 置 $f_i = Q_n$ 。

输出未达到精度要求的信息。

过程终。

(三) 数值例题

给定数据: $f(1.0) = 1.00000$, $f(1.1) = 1.23368$,

$f(1.2) = 1.55271$, $f(1.3) = 1.99372$,

$f(1.4) = 2.61170$

使用 Neville 插值算法求 $f(1.25)$ 的近似值。

解 输入 $n=5$; 数据 x_i 和 $f(x_i)$; $\text{TOL}=10^{-5}$; $t=1.25$ 。

由 Neville 插值算法打印 $f(1.25)$ 的近似值为 1.75496. ■

(四) 练习题

1. 给定数据

$f(-0.3) = -0.20431$, $f(-0.1) = -0.08993$,

$f(0.1) = 0.11007$, $f(0.3) = 0.39569$,

$f(0.5) = 0.79845$

试用 Neville 算法求 $f(0)$ 的近似值。

2. 使用 Neville 算法求 $f(-0.78)$ 的近似值:

已知函数 $f(x) = x^2 e^x \cos x$, 利用节点 $x_0 = -1.0$,

$x_1 = -0.9$, $x_2 = -0.8$, $x_3 = -0.7$, $x_4 = -0.6$ 。

§ 1.3 Newton 插值均差公式算法

(一) 数学方法简述

设给定数据点 $(x_i, f_i), i = 0, 1, \dots, n$, 求不超过 n 次插值多项式 $p_n(x)$, 使之满足插值条件

$$p_n(x_i) = f(x_i) = f_i, \quad i = 0, 1, \dots, n$$

按下面构造性方法作出插值多项式:

首先构造 Newton 均差表:

$x_i \quad f(x_i)$	一阶均差	二阶均差
$x_0 \quad \underline{f[x_0]}$		
$x_1 \quad \underline{f[x_1]}$	$\underline{f[x_0, x_1]} = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$	
$x_2 \quad \underline{f[x_2]}$	$\underline{f[x_1, x_2]} = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$	$\underline{f[x_0, x_1, x_2]} = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$
$x_3 \quad \underline{f[x_3]}$	$\underline{f[x_2, x_3]} = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$	$\underline{f[x_1, x_2, x_3]} = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} \dots$
...	...	...

其次以上表中底下划横线的均差为系数作出 Newton 插值均差公式为

$$\begin{aligned}
 p_n(x) = & f[x_0] + f[x_0, x_1](x - x_0) \\
 & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\
 & + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots \\
 & (x - x_{n-1}),
 \end{aligned}$$

或

$$p_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, \dots, x_k](x - x_0) \cdots (x - x_{k-1}) \quad (1.3)$$

方法评述: Newton 插值公式 (1.3), 不但方法简单、容易编程电算, 而且也克服了 Lagrange 插值多项式的缺点, 即当增加一个节点时, Newton 插值均差公式也只相应地增加一项, 即

$$p_0(x) = f(x_0)$$

$$p_1(x) = f[x_0] + f[x_0, x_1](x - x_0)$$

$$= p_0(x) + f[x_0, x_1](x - x_0)$$

$$\vdots$$

$$p_n(x) = p_{n-1}(x) + f[x_0, \dots, x_n](x - x_0) \cdots (x - x_{n-1})$$

从而显出 Newton 插值均差公式的优越性。

(二) 通用算法

Newton 插值均差公式算法

目的: 对数据 $(x_i, f(x_i)), i = 0, 1, \dots, n$, 求插值多项式 (1.3) 的均差系数。

输入: 正整数 n ; 数 $x_0, x_1, \dots, x_n$; 值 $f(x_0), f(x_1), \dots, f(x_n)$ 作为零阶均差, 记为 $p_{0,0}, p_{1,0}, \dots, p_{n,0}$ 。

输出: 数 $p_{0,0}, p_{1,1}, \dots, p_{n,n}$, 其中

$$p(x) = \sum_{i=0}^n p_{i,i} \prod_{j=0}^{i-1} (x - x_j)$$

计算步骤:

1. 对 $i = 1, 2, \dots, n$

 对 $j = 1, 2, \dots, i$

$$\text{置 } p_{i,j} = \frac{p_{i,j-1} - p_{i-1,j-1}}{x_i - x_{i-j}}$$

2. 输出 $(p_{0,0}, p_{1,1}, \dots, p_{n,n})$; $(p_{i,i})$ 是 $f[x_0, x_1, \dots, x_i]$