

李树忱 著
王兆清

高精度无网格重心插值 配点法——算法、程序及工程应用



科学出版社

高精度无网格重心插值配点法

——算法、程序及工程应用

李树忱 王兆清 著

国家重点基础研究发展计划(2010CB732002)

国家自然科学基金(51179098)

科学出版社

北 京

内 容 简 介

本书论述了基于重心型插值的高精度无网格配点法的基本算法和计算程序;详细讨论了常微分方程(组)边值问题和初值问题、积分方程和积分-微分方程、二维椭圆型偏微分方程边值问题、波动方程和热传导方程的重心插值配点法计算公式和程序;论述了不规则区域上重心插值配点法的具体算法;给出了重心插值配点法在结构变形、屈曲和振动分析方面的算法和程序;通过大量算例说明重心插值配点法的有效性和计算精度。

本书可供从事数值分析领域研究的工程技术人员和高等院校计算数学、计算力学、土木工程等专业本科生、研究生参考。

图书在版编目(CIP)数据

高精度无网格重心插值配点法:算法、程序及工程应用/李树忱,王兆清著. —北京:科学出版社,2012

ISBN 978-7-03-035595-9

I. ①高… II. ①李… ②王… III. ①插值法-应用-研究 IV. ①O241.3

中国版本图书馆 CIP 数据核字(2012)第 223342 号

责任编辑:刘宝莉 / 责任校对:宋玲玲

责任印制:张 倩 / 封面设计:陈 敬

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

中国科学院印刷厂印刷

科学出版社发行 各地新华书店经销

*

2012年9月第 一 版 开本:B5(720×1000)

2012年9月第一次印刷 印张:31 1/4

字数:618 000

定价:120.00 元

(如有印装质量问题,我社负责调换)

前 言

工程中的许多问题都可以应用相关学科知识和数学理论描述为数学问题或建立数学模型，一般采用微分方程来描述。只有少数问题可以得到问题的解析解，大部分的问题需要借助数值方法求解。数值求解微分方程初值问题的经典方法是有限差分法，数值求解微分方程边值问题的经典方法是有限元方法。要得到问题的高精度解，差分法一般需要采用高阶差分格式或采用较小的差分步长，有限元方法一般需要采用高阶单元或采用较小尺寸的单元，这将极大地降低分析问题的效率。

求解工程微分方程的高效率高精度数值分析方法，是数值分析工作者不断努力追求的目标。本书的目的就是为读者介绍一种高效率、高精度数值分析方法——重心插值配点法。重心插值配点法采用重心型插值作为近似函数，利用重心插值的微分矩阵离散微分算子，不需要划分计算网格，计算程序编写简便，是一种高效率、高精度的无网格数值方法。

作为本书的基础，首先论述了重心型插值基本理论和计算程序；随后各章详细讨论了常微分方程（组）边值问题和初值问题、积分方程和积分-微分方程、二维椭圆型偏微分方程边值问题、波动方程和热传导方程的重心插值配点法计算公式和程序；论述了不规则区域上重心插值配点法的具体算法和计算程序；给出了重心插值配点法在结构变形、屈曲和振动分析方面的算法和计算程序；通过大量算例说明重心插值配点法的有效性和计算精度。

为方便读者，本书对于具有代表性的数值算例，给出详细的采用 MATLAB6.5 编写的程序代码，所有计算程序都经过作者的检验。作者在计算程序编写方面考虑了程序的通用性和可扩充性，对于其他的数值算例，只需对程序做少量修改即可使用。

本书出版得到国家重点基础研究发展计划（2010CB732002）、国家自然科学基金（51179098）和山东建筑大学科研基金的资助。

由于作者水平有限，书中或有不当之处，敬请各位同仁批评指正。意见可发至邮箱：sdjzuwang@126.com。

目 录

前言

第 1 章 重心型插值及其微分矩阵	1
1.1 数值计算方法的本质	1
1.2 Lagrange 插值	2
1.2.1 Lagrange 插值和 Newton 插值	2
1.2.2 改进的 Lagrange 插值	3
1.3 重心 Lagrange 插值	3
1.4 重心有理插值	5
1.4.1 重心有理插值的一般形式	5
1.4.2 高阶重心有理插值	5
1.5 函数的重心插值计算程序	7
1.5.1 重心 Lagrange 插值的计算程序	7
1.5.2 重心有理插值的计算程序	8
1.6 重心插值的微分矩阵	10
1.7 微分矩阵的计算程序	11
1.8 任意连续函数的多项式逼近	13
1.9 微分矩阵计算函数导数	16
1.10 本章小结	17
第 2 章 边值问题的重心插值配点法	19
2.1 二阶边值问题的重心插值配点法	19
2.1.1 重心插值配点法计算公式	19
2.1.2 边界条件的施加方法	20
2.1.3 二阶边值问题的算例和计算程序	21
2.2 奇异摄动二阶边值问题的重心插值配点法	25
2.2.1 奇异摄动二阶边值问题的计算公式	25
2.2.2 数值算例	26
2.3 高阶两点边值问题的重心插值配点法	30
2.3.1 三阶两点边值问题	30
2.3.2 四阶两点边值问题	32
2.3.3 五阶两点边值问题	39

2.3.4	六阶两点边值问题	42
2.3.5	七阶两点边值问题	48
2.3.6	八阶两点边值问题	50
2.3.7	九阶两点边值问题	54
2.3.8	十阶两点边值问题	57
2.3.9	十阶以上两点边值问题	61
2.4	多点边值问题的重心插值配点法	64
2.4.1	三阶三点边值问题	65
2.4.2	四阶多点边值问题	68
2.5	本章小结	74
第3章	初值问题的重心插值配点法	75
3.1	一阶初值问题的重心插值配点法	75
3.1.1	重心插值配点法计算公式	75
3.1.2	一阶初值问题的数值算例和程序	76
3.2	二阶初值问题的重心插值配点法	78
3.2.1	重心插值配点法计算公式	78
3.2.2	初始条件的施加方法	79
3.2.3	重心插值配点法的数值计算步骤	80
3.2.4	数值算例和计算程序	80
3.3	高阶初值问题的重心插值配点法	83
3.3.1	三阶初值问题	83
3.3.2	四阶初值问题	85
3.3.3	五阶初值问题	89
3.3.4	八阶初值问题	89
3.4	线性微分方程组初值问题的重心插值配点法	90
3.4.1	线性微分方程组的离散公式	90
3.4.2	线性微分方程组的数值算例和程序	92
3.5	微分方程初值问题的区间分解法	99
3.6	脉冲激励振动的数值分析	100
3.7	本章小结	103
第4章	积分方程的重心插值配点法	105
4.1	Fredholm 积分方程	105
4.1.1	Fredholm 积分方程的重心插值离散公式	105
4.1.2	积分向量和数值积分	106
4.1.3	Fredholm 积分方程的数值算例	110

4.2	一阶 Fredholm 积分-微分方程	113
4.2.1	一阶 Fredholm 积分-微分方程的重心插值离散格式	113
4.2.2	一阶 Fredholm 积分-微分方程的数值算例	115
4.3	二阶 Fredholm 积分-微分方程	121
4.4	高阶 Fredholm 积分-微分方程	129
4.5	Volterra 积分方程	133
4.5.1	Volterra 积分方程的重心插值配点法离散格式	133
4.5.2	Volterra 积分方程的数值算例和计算程序	134
4.6	一阶 Volterra 积分-微分方程	138
4.6.1	一阶 Volterra 积分-微分方程的重心插值离散格式	138
4.6.2	一阶 Volterra 积分-微分方程的数值算例和计算程序	139
4.7	二阶 Volterra 积分-微分方程	142
4.8	四阶 Volterra 积分-微分方程	148
4.9	混合型积分-微分方程	152
4.10	积分-微分方程组	154
4.10.1	Fredholm 积分-微分方程组	155
4.10.2	Volterra 积分-微分方程组	161
4.11	本章小结	167
第5章	二维边值问题的重心插值配点法	168
5.1	二维椭圆型方程边值问题的重心插值配点法计算公式	168
5.1.1	二维椭圆型方程的重心插值离散公式	168
5.1.2	边界条件的离散公式和施加方法	174
5.2	二维 Poisson 方程的计算	175
5.3	二维广义 Poisson 方程的计算	185
5.4	双调和方程的计算	190
5.5	双调和方程的降阶计算	197
5.6	极坐标下的边值问题	200
5.7	轴对称问题的重心插值配点法	206
5.8	本章小结	209
第6章	重心插值区域分解法和正则区域法	210
6.1	一维不连续边值问题的区域分解法	210
6.1.1	不连续边值问题的重心插值区域分解法计算格式	210
6.1.2	边界条件和单元间连接条件的处理	211
6.1.3	数值算例和计算程序	212
6.2	二维边值问题的重心插值区域分解法	218

6.2.1	重心插值区域分解法的计算公式	218
6.2.2	数值算例和计算程序	224
6.3	重心插值正则区域法	230
6.3.1	重心插值正则区域法的计算格式	230
6.3.2	不规则域上 Poisson 方程的正则区域法计算	233
6.3.3	不规则域上扩散方程的正则区域法计算	247
6.4	不规则域上双调和方程的正则区域法计算	257
6.5	界面问题的正则区域分解法	262
6.6	渗流自由面问题的正则区域法	276
6.6.1	渗流自由面问题的控制方程	276
6.6.2	渗流自由面问题的正则区域法计算	277
6.7	本章小结	281
第 7 章	动态扩散问题的数值分析	282
7.1	热传导问题的基本方程	282
7.2	一维热传导方程的重心插值配点法	284
7.2.1	一维热传导方程的离散方程	284
7.2.2	一维热传导方程的计算	288
7.3	一维非局部边界条件热传导方程的重心插值配点法	294
7.3.1	非局部边界条件热传导方程举例	294
7.3.2	未知函数整体约束的非局部边界条件热传导方程的计算	295
7.3.3	一般形式的非局部边界条件热传导方程的计算	302
7.4	对流-扩散方程的重心插值配点法计算	308
7.5	动态 Schrödinger 方程的数值计算	318
7.6	反应-扩散方程的数值计算	329
7.7	本章小结	333
第 8 章	振动问题的数值分析	334
8.1	一维波动方程的重心插值配点法	334
8.1.1	一维波动问题的控制方程和定解条件	334
8.1.2	一维波动问题的重心插值离散公式	335
8.1.3	一维波动问题的数值计算	339
8.2	一维非局部边界条件波动方程的数值计算	343
8.2.1	一维非局部边界条件波动方程的计算	343
8.2.2	一维非局部边界条件广义波动方程的计算	347
8.3	电报方程的重心插值配点法计算	351
8.4	等直杆的纵向自由振动数值分析	355

8.5	变截面杆的自由振动	367
8.6	本章小结	371
第9章	梁结构力学问题的数值分析	372
9.1	Euler-Bernoulli 梁弯曲变形分析	372
9.1.1	简单载荷作用下梁的变形分析	373
9.1.2	复杂载荷作用下梁的变形分析	379
9.2	Euler-Bernoulli 梁自由振动数值分析	387
9.3	Euler-Bernoulli 梁受迫振动数值分析	391
9.4	Euler-Bernoulli 梁屈曲的数值分析	401
9.5	不连续 Euler-Bernoulli 梁的变形分析	407
9.6	圆环变形和屈曲的数值分析	416
9.6.1	圆环的变形分析	417
9.6.2	均布压力下圆环的屈曲分析	419
9.7	Timoshenko 梁振动的数值分析	421
9.7.1	Timoshenko 梁的控制方程与边界条件	421
9.7.2	Timoshenko 梁自由振动的重心插值配点法计算格式	422
9.7.3	Timoshenko 梁自由振动的数值算例	425
9.7.4	Timoshenko 梁受迫振动的数值分析	428
9.8	本章小结	430
第10章	弹性力学问题的重心插值配点法	431
10.1	直角坐标系下平面问题的重心插值配点法	431
10.1.1	平面问题的基本方程	431
10.1.2	平面问题位移解法的心插值配点法	432
10.1.3	平面问题应力解法的心插值配点法	445
10.2	极坐标系下平面问题的重心插值配点法	449
10.2.1	极坐标系下平面问题的基本方程	449
10.2.2	极坐标系下平面问题的重心插值配点法	451
10.3	柱体扭转问题的重心插值配点法	466
10.3.1	柱体扭转问题的基本方程	466
10.3.2	直角坐标系下扭转问题的计算	469
10.4	本章小结	481
	参考文献	482

第 1 章 重心型插值及其微分矩阵

1.1 数值计算方法的本质

许多物理现象和工程问题的分析,可以归结为求解微分方程的初边值问题。只有少数简单情况可以得到问题的解析解,大部分工程问题需要采用数值方法求解。经典的数值分析方法有有限差分法(finite difference method, FDM)、有限元方法(finite element method, FEM)和边界元方法(boundary element method, BEM)。有限差分法采用差分算子近似微分算子,将微分方程通过差分离散为差分方程求解。有限元方法将求解区域划分为若干单元,在每一个单元上采用低阶多项式构造单元插值函数,利用 Galerkin 法得到求解问题的代数方程组。边界元方法利用所求问题的基本解将微分方程转化为边界积分方程,通过离散边界积分方程得到求解问题的代数方程组。

有限差分法和有限元方法是低阶的数值分析方法,要得到问题的高精度数值解,有限差分法需要非常小的差分步长,有限元方法需要划分稠密的计算网格。这极大地降低了分析问题的效率。边界元方法是高精度算法,利用边界元方法可以降低分析问题的维数,简化问题的分析。但该方法的构造依赖于问题的基本解,对于一些不存在基本解的工程问题,不能采用边界元方法进行分析。

不同数值计算方法本质区别在于:①未知函数近似函数的构造方法;②微分方程的离散技术。不同的未知函数近似函数构造方法和微分方程离散技术,构成不同的数值分析方法。

未知函数近似函数的构造方法,是数值分析方法的核心。大部分的数值计算方法采用多项式函数构造未知函数的近似函数。未知函数近似函数的构造方法分局部构造法和整体构造法。局部构造法是采用分段(片)多项式函数近似未知函数,典型代表方法是有限元方法。整体构造法是在整个计算区域上采用多项式函数近似未知函数,典型代表方法拟谱方法(pseudo-spectral method, PSM)。数值计算方法的计算精度主要依赖于未知函数近似函数的近似精度。一般来说,整体构造法的近似精度优于局部构造法,高阶多项式的近似精度高于低阶多项式。

微分方程的离散技术可分为配点法和 Galerkin 法两种。配点法是基于微分方程强形式的离散方法,也就是强迫微分方程在给定的离散节点上精确成立,由此得到求解问题的代数方程组。Galerkin 法是基于微分方程弱形式的离散方法,也

就是使微分方程在加权意义下成立。强形式离散方法要求构造的未知函数近似函数具有与方程阶数相同的光滑性要求,对未知函数近似函数的光滑性要求较高。弱形式的离散方法可以降低未知函数近似函数的光滑性要求,但是数值计算过程中需要进行积分运算,增加了计算公式和计算程序的复杂性。

本书的重心插值配点法是基于微分方程强形式的配点型方法。未知函数的近似函数采用离散节点上的重心型插值。利用重心型插值的数值稳定性好和近似精度高的优点,实现微分方程的高精度数值计算。

1.2 Lagrange 插值

插值在工程数值计算方面有广泛的应用,如有限元方法中形函数的建立、科学计算可视化过程中图形图像的显示等,都需要应用到插值理论。在一维的情况下,插值方法主要有 Lagrange 插值、分段线性插值、样条插值和有理函数插值等^[1]。Lagrange 插值是一种多项式插值,在数值分析理论方面有重要的作用,但是在等距节点插值过程中,当节点数量较大时,Lagrange 插值表现出极大的数值不稳定性,使得 Lagrange 插值精度急剧下降。著名的 Runge 函数 Lagrange 插值的振荡性,正说明了 Lagrange 插值的这一缺陷。

近年来,国际数值分析学者致力于 Lagrange 插值数值稳定性方面的研究,取得了许多新的成果,改进了 Lagrange 插值数值不稳定的缺陷;同时研究发现采用有理函数作为插值基函数,可以显著地提高插值精度;并且 Lagrange 插值和有理函数插值都可以表示为一种重心插值的形式。通过不同权的选取,可以得到不同的插值格式,丰富了一维插值的类型^[2]。

1.2.1 Lagrange 插值和 Newton 插值

设有 $n+1$ 个不同的插值节点 $x_j (j=1, 2, \dots, n)$ 和相对应的一组实数 f_j 。若采用多项式插值,则在次数不超过 n 的多项式空间中,求一个插值多项式 $p(x)$,使多项式 $p(x)$ 满足 $p(x_j) = f_j, j=1, 2, \dots, n$ 。

多项式插值问题有唯一解,插值多项式可以写成 Lagrange 插值公式

$$p(x) = \sum_{j=1}^n L_j(x) f_j, \quad L_j(x) = \frac{\prod_{k=1, k \neq j}^n (x - x_k)}{\prod_{k=1, k \neq j}^n (x_j - x_k)}, \quad j = 1, 2, \dots, n \quad (1.1)$$

式中: $L_j(x)$ 为 Lagrange 插值基函数,其满足以下性质:

$$L_j(x_k) = \delta_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases} \quad (1.2)$$

Lagrange 插值公式(1.1)在插值节点较少时,具有很好的插值精度,其存在的主要缺陷如下^[3]:

- (1) 计算量是 $O(n^2)$ 。
- (2) 增加新的插值节点,需要重新计算。
- (3) 当节点数量很大时,插值是数值不稳定的。

我们知道,多项式插值也可以写成 Newton 形式,通过一系列的差商计算,Newton 插值公式可以写成

$$p(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2) + \cdots + f[x_1, x_2, \cdots, x_n](x - x_1)(x - x_2) \cdots (x - x_{n-1}) \quad (1.3)$$

Newton 插值公式当增加新的插值节点时,只需计算最后一个差商,相比 Lagrange 插值公式要方便得多,但是当节点数量很大时,插值仍是数值不稳定的。

1.2.2 改进的 Lagrange 插值^[4]

令

$$l(x) = (x - x_1)(x - x_2) \cdots (x - x_n) \quad (1.4)$$

定义重心权

$$w_j = \frac{1}{\prod_{j \neq k} (x_j - x_k)}, \quad j = 1, 2, \cdots, n \quad (1.5)$$

也就是 $w_j = 1/l'(x_j)$, 则插值基函数可以表示为

$$L_j(x) = l(x) \frac{w_j}{x - x_j}, \quad j = 1, 2, \cdots, n \quad (1.6)$$

将式(1.6)代入 Lagrange 插值公式(1.1),得到 Lagrange 插值的另一种表现形式

$$p(x) = l(x) \sum_{j=1}^n \frac{w_j}{x - x_j} f_j \quad (1.7)$$

Lagrange 插值公式(1.7)称为改进的 Lagrange 插值公式。改进的 Lagrange 插值公式是向后稳定的,增加一个新的插值点仅需 $O(n)$ 次运算。

1.3 重心 Lagrange 插值^[4]

利用插值公式(1.7)插值常数 1, 可得下面恒等式:

$$1 = \sum_{j=1}^n L_j(x) = l(x) \sum_{j=1}^n \frac{w_j}{x - x_j} \quad (1.8)$$

将式(1.8)两边分别去除式(1.7)的两边,得到重心 Lagrange 插值公式

$$p(x) = \frac{\sum_{j=1}^n \frac{w_j}{x - x_j} f_j}{\sum_{j=1}^n \frac{w_j}{x - x_j}} \quad (1.9)$$

重心 Lagrange 插值公式在分子和分母都包含插值权 w_j , 因此 w_j 的任何非零倍数仍是插值权。由式(1.5)可知, 插值权仅依赖于插值节点的分布, 因此对于一些特殊的节点分布可以得到简化插值权。重心 Lagrange 插值是向前稳定的^[4], 但对于数值分析中常用的等距节点插值, 其插值是病态的。通过选择一些特殊的插值节点分布, 重心 Lagrange 插值具有非常好的数值稳定性。

对于一些特殊的节点分布, 重心 Lagrange 插值权具有非常简单的表达式。对于区间 $[-1, 1]$ 上的等距分布节点, $h = 2/n$, 通过对式(1.5)直接计算得到 $w_j = (-1)^{n-j} C_n^j / (h^n n!)$, 消除与 j 无关的因子, 得到等距节点重心 Lagrange 插值的插值权

$$w_j = (-1)^{n-j} C_n^j \quad (1.10)$$

除非 n 较小, 多项式插值对于等距节点是病态的。当节点分布密度与函数 $(1-x^2)^{-1/2}$ 成比例, 也就是节点分布在区间的两端较密, 在区间中间较稀疏时, 多项式插值具有很好的数值稳定性^[3]。满足这样条件的最简单节点分布是 Chebyshev 点族。

第一类 Chebyshev 节点为

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \quad j = 0, 1, \dots, n \quad (1.11)$$

其重心 Lagrange 插值的插值权为

$$w_j = (-1)^j \sin\left(\frac{2j+1}{2n+2}\pi\right) \quad (1.12)$$

第二类 Chebyshev 节点为

$$x_j = \cos\left(\frac{j}{n}\pi\right), \quad j = 0, 1, \dots, n \quad (1.13)$$

其重心 Lagrange 插值的插值权为

$$w_j = (-1)^j \delta_j, \quad \delta_j = \begin{cases} 1/2, & j = 0 \text{ 或 } n \\ 1, & \text{其他} \end{cases} \quad (1.14)$$

Chebyshev 节点族定义在区间 $[-1, 1]$ 。利用坐标变换 $y = x(b-a)/2 + (b+a)/2$ 可将 $[-1, 1]$ 的 Chebyshev 节点变换为任意区间 $[a, b]$ 上具有相同分布形式的节点。

1.4 重心有理插值

1.4.1 重心有理插值的一般形式

在数值分析中,有些时候插值节点是无法自由选择的,如事先给定了插值节点及其对应函数值的插值问题。特别是等距节点插值问题,Lagrange 多项式插值是不稳定的。可以采用分段多项式插值,如样条插值,克服多项式的数值不稳定性问题^[5]。

有时有理函数插值比多项式插值具有更高的插值精度,采用有理函数插值可以有效克服插值的不稳定性问题。记一个有理函数空间 $R_{m,n}$,其元素是由分子次数不超过 m 的多项式和分母次数不超过 n 的多项式构成的有理函数。问题是对于给定的节点分布和对应的函数值,是否存在一个有理函数满足插值条件? 下面的定理说明有理函数插值问题一定有解^[3]。

定理 设 $\{(x_j, f_j), j=0, 1, \dots, n\}$ 为 $n+1$ 个实数对, x_j 互不相同。 $\{u_j, j=0, 1, \dots, n\}$ 为 $n+1$ 个实数, 则

(1) 如果 $u_k \neq 0$, 存在有理函数 $r(x) \in R_{n,n}$,

$$r(x) = \frac{\sum_{j=0}^n \frac{u_j}{x - x_j} f_j}{\sum_{j=0}^n \frac{u_j}{x - x_j}} \quad (1.15)$$

使得 $r(x)$ 在 x_k 处插值 f_k , 也就是 $\lim_{x \rightarrow x_k} r(x) = f_k$ 。

(2) 反之, 任何一个插值实数对 $\{(x_j, f_j), j=0, 1, \dots, n\}$ 的有理函数均可以表示为式(1.15)的重心插值形式。

此定理说明, 对于给定的插值节点及函数值, 存在无穷多重心有理函数插值, 并且所有的有理函数插值都可以写成式(1.15)的重心插值形式。

如何选择插值权 u_k , 使得重心有理函数插值具有尽可能高的插值精度, 是实际应用有理函数插值的中心问题。

Berrut 最初建议采用

$$u_k = (-1)^k, \quad k = 0, 1, \dots, n \quad (1.16)$$

作为重心有理函数插值的插值权^[6], 其插值近似阶为 $O(h)$, 这里 $h = \max_{j=0, 1, \dots, n-1} \{|x_{j+1} - x_j|\}$ 。后来 Berrut 及其合作者采用特殊的插值节点分布和增加极点的方法, 提高重心有理函数插值的插值精度^[7~13]。

1.4.2 高阶重心有理插值

我们知道将一种插值嵌入另外一种插值中, 可以提高插值的阶次, 如将具有

C^0 连续性的自然邻点插值嵌入 Bernstein 基函数, 可以构造出具有 C^1 连续性的插值函数^[14]。基于这种混合函数的思想, Floater 和 Hormann 提出了高阶重心有理函数插值^[15]。

给定插值节点 $x_i, i=0, 1, \dots, n$ 及其对应的函数值 f_i , 选择一个整数 d 满足 $0 \leq d \leq n$, 对于每一个 $i=0, 1, \dots, n-d$, 令 p_i 为插值 $d+1$ 个点 $(x_i, f_i), (x_{i+1}, f_{i+1}), \dots, (x_{i+d}, f_{i+d})$ 的次数至多为 d 的多项式, 则令

$$r(x) = \sum_{i=0}^{n-d} \lambda_i(x) p_i(x) / \sum_{i=0}^{n-d} \lambda_i(x) \quad (1.17)$$

其中

$$\lambda_i(x) = \frac{(-1)^i}{(x-x_i) \cdots (x-x_{i+d})} \quad (1.18)$$

式(1.17)是将多项式插值 $p_0, p_1, \dots, p_{n-d}$ 通过权函数 $\lambda_i(x)$ 混合得到的。容易证明式(1.17)的 $r(x)$ 插值所给的点对 $\{(x_j, f_j), j=0, 1, \dots, n\}$ 。这样对于不同的 $d=0, 1, \dots, n$ 得到一族有理函数插值。Floater 和 Hormann 证明, 只要被插值函数具有足够的光滑性, 式(1.17)构造的有理函数插值的近似阶为 $O(h^{d+1})$, 由此通过适当的选择参数 d , 可以得到我们所需要的插值精度。

有理函数插值公式(1.17)可以写成重心插值的形式。令 $I=\{0, 1, 2, \dots, n\}$ 为一指标集。将多项式 $p_i(x)$ 写成 Lagrange 公式形式

$$p_i(x) = \sum_{k=i}^{i+d} \prod_{j=i, j \neq k}^{i+d} \frac{x-x_j}{x_k-x_j} f_k \quad (1.19)$$

将式(1.19)代入式(1.17)的分子, 得

$$\sum_{i=0}^{n-d} \lambda_i(x) p_i(x) = \sum_{i=0}^{n-d} (-1)^i \sum_{k=i}^{i+d} \frac{1}{x-x_{k_j=i, j \neq k}} \prod_{j=i, j \neq k}^{i+d} \frac{1}{x_k-x_j} f_k = \sum_{k=0}^n \frac{w_k}{x-x_k} f_k \quad (1.20)$$

其中

$$w_k = \sum_{i \in J_k} (-1)^i \prod_{j=i, j \neq k}^{i+d} \frac{1}{x_k-x_j} \quad (1.21)$$

指标集 $J_k = \{i \in I; k-d \leq i \leq k\}$ 。对于式(1.17)的分母, 根据常数 1 的 Lagrange 插值公式有恒等式

$$1 = \sum_{k=i}^{i+d} \prod_{j=i, j \neq k}^{i+d} \frac{x-x_j}{x_k-x_j} \quad (1.22)$$

由此可得

$$\sum_{i=0}^{n-d} \lambda_i(x) = \sum_{k=0}^n \frac{w_k}{x-x_k} \quad (1.23)$$

将式(1.20)和式(1.23)代入式(1.17), 得到高阶重心有理插值公式

$$r(x) = \frac{\sum_{j=0}^n \frac{w_j}{x - x_j} f_j}{\sum_{j=0}^n \frac{w_j}{x - x_j}} \quad (1.24)$$

其中插值权 w_j 由式(1.21)确定。注意到,当 $d=0$ 时,插值权(1.16)和(1.21)等价。

今后将重心 Lagrange 插值和重心有理插值统称为重心插值。在计算程序中,只需改变插值节点和插值权,重心 Lagrange 插值的计算程序就成为重心有理插值的计算程序。

1.5 函数的重心插值计算程序

1.5.1 重心 Lagrange 插值的计算程序

以 Runge 函数 $f(x)=1/(1+25x^2)$ 为例,说明重心 Lagrange 插值的良好数值稳定性及计算程序^[16]。在区间 $[-1,1]$ 上令 $n=100$,以第二类 Chebyshev 节点作为插值节点,插值计算区间 $[-1,1]$ 上的 2000 个点的近似值。利用 MATLAB 编写计算程序,计算程序清单见图 1.1。为便于阅读和使用,程序文件名标示在图题的【】中。

```
clear;clc;n=100;d=3;fun=inline('1./(1+25*x.^2)');
x=-cos(pi*(0:n)'/n);n=length(x);f=fun(x);
w=[0.5;ones(n-1,1);0.5].*(-1).^((0:n)');
xx=linspace(-1,1,2000)';numer=zeros(size(xx));
denom=zeros(size(xx));exact=zeros(size(xx));
for j=1:n
    xdifff=xx-x(j);
    if xdifff~=0
        temp=w(j)./xdifff;
        numer=numer+temp*f(j);
        denom=denom+temp;
    end
    exact(xdifff==0)=1;
end
ff=numer./denom;jj=find(exact);ff(jj)=fun(xx(jj));
plot(x,f,'k',xx,ff,'-k');fe=fun(xx);er=max(abs(ff-fe))
```

图 1.1 重心 Lagrange 插值程序【Bary_Lag_intp.m】

程序绘制的图形如图 1.2 所示,本算例计算的插值点最大绝对误差为 9.3940×10^{-4} 。通过大量的数值计算表明,对于等距节点插值重心 Lagrange 插值表现为极大的数值不稳定性,但是对于 Chebyshev 节点的插值,重心 Lagrange 插值具有极好的数值稳定性。

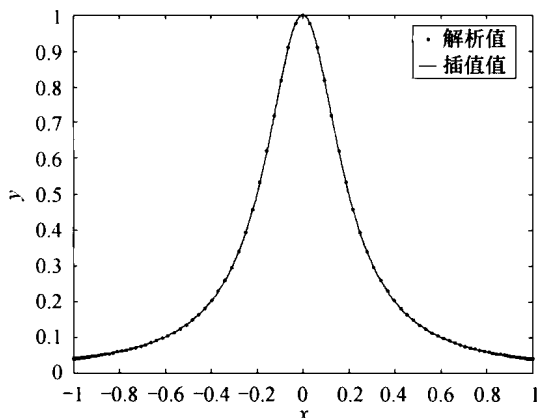


图 1.2 Runge 函数的重心 Lagrange 插值

1.5.2 重心有理插值的计算程序

由于重心 Lagrange 插值(1.9)和重心有理插值(1.24)的插值公式形式完全相同,区别仅在于插值权的不同;因此在 1.5.1 节介绍的重心 Lagrange 插值计算程序完全适用于重心有理插值,但是要对插值权单独编写计算程序。利用重心有理插值权公式(1.21),编写计算插值权的 MATLAB 函数程序见图 1.3,重心有理插值的计算程序见图 1.4。

```
function w=wvalue(d,x)
n=length(x);
for k=1:n
    if k<=d
        Jk=1:k;
    else
        Jk=k-d:k;
    end
    w(k)=0; m=length(Jk);
    for i=1:m
        pw(i)=1;
        for j=Jk(i):Jk(i)+d
            if (j~=k&j<=n)
                pw(i)=pw(i)*1/(x(k)-x(j));
            end
            if j>n
                pw(i)=0
            end
        end
        w(k)=w(k)+(-1)^(Jk(i)+1)*pw(i);
    end
end
```

图 1.3 重心有理插值权程序【wvalue.m】