



21 世纪数学系列教材

实用计算方法

徐长发 王 邦



华中科技大学出版社

<http://press.hust.edu.cn>

- 立足数值建模 介绍应用背景
- 突出算法思想 注重形象直观
- 总结实用经验 指导应用实践
- 内容通俗易懂 分析深入浅出

实用计算方法

徐长发 王 邦

华中科技大学出版社

图书在版编目(CIP)数据

实用计算方法/徐长发 王 邦

武汉:华中科技大学出版社,2005 年 1 月

ISBN 7-5609-3301-7

I. 实…

II. ①徐… ②王…

III. 计算数学-高等学校-教材

IV. O241

实用计算方法

徐长发 王 邦

责任编辑:徐正达

封面设计:刘 卉

责任校对:刘 竣

责任监印:熊庆玉

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉皇荣文化发展有限公司

印 刷:湖北省通山县九宫印务有限公司

开本:787×960 1/16

印张:11.25

字数:195 000

版次:2005 年 1 月第 1 版

印次:2005 年 1 月第 1 次印刷

定价:14.00 元

ISBN 7-5609-3301-7/O · 338

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书是理工科本科各专业计算方法课程的教材,内容包括曲线插值和曲线拟合、数值积分和数值微分、非线性方程求根的迭代法、解线性方程组的迭代法、解线性方程组的直接法、解常微分方程的差分方法、矩阵特征值与特征向量的数值方法等。每章提供了不同类型的习题,并在书末给出了习题解答的提示。

本书突出了常见数值问题的应用背景,突出了各类数值方法的构造思想、算法实现和实用范围。其内容是实用的,有关概念和算法都是直观易懂的。本书可读性强且利于自学,能指导读者应用现有软件去解决常见的实际问题。

本书也可作为有关工程技术人员的参考书。

前 言

《实用计算方法》是为计算方法(或称数值分析)课程而编写的本科教材。

理工科各专业在本科阶段应开设计算方法课程,这是时代发展的需要。早在20世纪70年代末和80年代初,计算机还处于穿孔输入低级阶段时,发达国家的综合性大学和专业技术学院就已全面开设了计算方法课程。如今,计算机已发展到相当成熟的地步,各类应用软件、包括计算方法方面的软件和程序都已普及,各行各业的发展都已离不开计算机了。我国高校的本科生几乎人人都会使用计算机,理工科的本科学生若不懂得用计算机处理简单的工程技术问题,则是不合时宜的。

计算方法所学习的内容是理工科各专业的必备知识。大家知道,高等数学和线性代数课程是必修课,因为它们在理论上提供了一些重要的思维方法和计算分析方法。然而,计算方法课程则出于应用的目的将上述课程中的思维方法和计算分析方法在计算机上予以实现,可见计算方法课程是多么基础,多么重要,多么实用。计算方法课程应该首先在我国高校理工科本科专业中普及,计算数学工作者应该推动这项普及工作。

计算方法教材应通俗易懂,好教好学。本书采用了许多与众不同的写作方式来体现本书特色。第一,本书增写了各类数值方法的应用背景,这有利于增强读者的学习动力和应用热情。第二,本书立足于数值方法建模,形象直观地分析总结了算法思想,读者只需掌握算法思想,而不需要死记那些形式复杂的计算公式。第三,本书以实用为目的,从介绍典型算法入手来展开全书内容。要求学生能掌握算法思想、算法特点、算法改进、算法总结和算法应用这个主线条;对于简单算法和简单问题,要求学生在理解算法思想的基础上能手工演算;对于较复杂的算法和较复杂的问题,则要求学生能应用 Mathematica 和 Matlab 去计算解决。第四,在写作方式方面,为了符合教学习惯,本书采用提出问题、分

析问题和解决问题的形式;为了便于阅读,本书力求通俗易懂和深入浅出,对较抽象的数学知识和数学形式给予形象直观的解释.

本书是在华中科技大学机械类平台学术委员会的委托和支持下完成的. 全书内容需讲授40多学时,若学时数偏少可据需要适当删减书中打“*”标记章节的内容.

华中科技大学机械学院戴同教授审阅了全部书稿并提出了不少宝贵的、有利于教和学的修改意见. 借本书出版之际,作者对所有支持本书写作的同志表示衷心的感谢.

作 者

2004年9月

目 录

绪论	(1)
0.1 学习好的算法	(1)
0.2 误差和精度	(2)
0.3 注意学习方法	(6)
第1章 曲线插值和曲线拟合	(7)
1.1 Lagrange 插值多项式	(7)
1.2 分段低次Lagrange 多项式插值方法	(18)
1.3 Hermite 插值和分段三次Hermite 插值方法	(21)
* 1.4 三次样条插值	(29)
1.5 曲线拟合的最小二乘方法	(39)
习题1	(45)
第2章 数值积分和数值微分	(47)
2.1 插值型数值积分的基本思想	(47)
2.2 插值型数值积分公式的确定办法及其代数精度	(48)
2.3 分段低阶数值积分和外推	(54)
2.4 Gauss 求积公式	(60)
2.5 数值微分及其外推	(63)
习题2	(69)
第3章 非线性方程求根的迭代法	(71)
3.1 实根隔离与二分法	(71)
3.2 基本迭代法及其外推	(72)
3.3 Newton 迭代法	(77)
* 3.4 解非线性方程组的Newton 迭代法	(83)
习题3	(86)

第4章 解线性方程组的迭代法	(87)
4.1 Jacobi 方法和 Gauss-Seidel 方法	(87)
4.2 向量和矩阵的模	(90)
4.3 线性方程组基本迭代法的收敛性	(95)
4.4 Jacobi 方法和 Gauss-Seidel 方法的敛散性	(97)
4.5 SOR 方法	(100)
习题4	(101)
第5章 解线性方程组的直接法	(103)
5.1 直接消去法	(103)
5.2 矩阵分解法	(106)
5.3 直接法的误差分析	(110)
习题5	(113)
第6章 解常微分方程的差分方法	(114)
6.1 一阶常微分方程初值问题及其差分方法	(114)
6.2 Euler 方法	(115)
6.3 梯形方法	(118)
6.4 Runge-Kutta 方法	(119)
6.5 显式单步法的稳定性问题	(123)
* 6.6 Adams 多步方法	(128)
* 6.7 常微初值问题数值解的外推方法	(133)
* 6.8 常微边值问题的差分离散化方法	(134)
* 6.9 常微特征值问题的差分离散化方法	(136)
习题6	(142)
* 第7章 矩阵特征值与特征向量的数值方法	(145)
7.1 幂法	(145)
7.2 反幂法	(148)
7.3 计算对称矩阵特征值的 Jacobi 方法	(151)
习题7	(153)
习题解答的提示	(155)

绪 论

大家知道,各行各业中的所有工程技术问题都需要定性、定量地去研究和解决,都离不开数学和计算机的帮助. 现在解决工程技术问题的流程大致如图0.0.1所示.

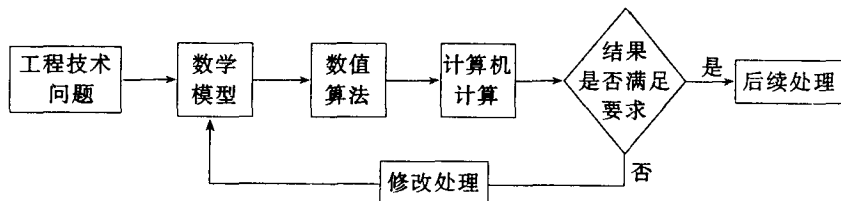


图 0.0.1 解决工程技术问题流程图

由此可见,数学是理性描述科技问题的工具,数值算法是数学模型和计算机之间的纽带和桥梁,可看做用于实际计算的离散化的数学模型. 利用数学工具和计算机,设计者可以反复修改和计算,可实现理性和感性的统一,可实现理论和实用的统一. 这种充分使用数学和计算机的设计方式极大地推动了科学技术的发展和应用. 尽管建立数学模型可能会涉及较多的数学知识,然而要实现数值计算,就必须作离散化处理,就必须将它归结为一些最基本的数学问题和算法问题. 本课程要学习的就是在科技计算中必须掌握的最基本、最常用的算法.

0.1 学习好的算法

对于某个具体的数学模型,人们可提出多种算法来实现其数值计算的目的,然而,这些算法有好的也有差的. 只有好的算法、实用的算法才是值得学习的.

1. 好的算法必须是合理、实用的

以求解线性代数方程组 $Ax=b$ 为例,其中 A 是一个 $n \times n$ 非奇异矩阵;若选用线性代数中介绍的Crame法则作为求解方法,则 $x_k = D_k/D, k=1, 2, \dots, n$,其中 D 是系数矩阵行列式的值, D_k 是相应的增广矩阵行列式的值. 在这种求解方法下,共需计算 $(n+1)$ 个行列式的值且需做 n 次除法运算,每个行列式共有 $n!$ 个乘积,每个乘积需做 $n-1$ 次乘法运算,这样共需 $[(n+1)!(n-1)+n]$ 次乘除运算. 当 $n=20$ 时,需做 9.7×10^{20} 次乘除运算,即使采用每秒运算1亿次的计算机也需计算30多万年! 显然,Crame法则在设计求解线性方程组算法时是不能被采用的.

2. 好的算法必须在计算量和计算精度方面是可满足实际需求的

仍以求解线性代数方程组 $Ax=b$ 为例,选用线性代数中介绍的Gauss消去法

(在第5章中将具体介绍这种数值方法),若消去过程能顺利进行,则求解过程所需的乘除法运算的次数可以估计出来,为 n^3 的同阶量,记为 $O(n^3)$,其中 n 是未知数的个数.当 $n=20$ 时,Gauss消去过程仅需几万次乘除运算就可完成求解工作;当 $n=100$ 时,求解工作也只需几百万次乘除运算.可见Gauss消去算法在计算量方面是可以满足实际需求的.然而,由线性代数知识大家也知道,若算法设计不合理,消去过程可能会产生较大的误差,甚至消去过程不能进行下去,只有改进了的Gauss消去法——选主元Gauss消去法——才是合理、实用的,满足适当精度要求的.

3. 好的算法必须是高效率的

衡量一个算法的效率可从总体计算量和计算精度这两个方面分析比较.例如,求解线性代数方程组时,用三种不同算法求得近似解的精度是差不多的,但它们需分别花费 $O(n^3)$, $O(n^2)$, $O(n)$ 的乘除运算,当然仅花费 $O(n)$ 计算量的算法在三者中是更好的算法.同样地,在计算花费差不多的前提下,计算结果精度高的算法是更好的算法.

4. 好的算法是有一定实用范围的

对于同一个数学问题,本书介绍了几个实用的算法.但一个好的算法往往都有它的实用范围,超出这个范围它就不再是好算法了.因此,读者只有懂得不同算法的不同特长,才能选用最有效的算法解决实际问题.

0.2 误差和精度

大家知道,解决实际工程技术问题要经过若干环节,每个环节都会产生误差,最后用计算机计算的结果当然也是有误差的.然而只要最后的计算结果能控制在所要求的误差范围内,这种解决问题的过程就是实用、有效的.因此,需要对所有可能的误差归类分析,分析其产生的原因和度量办法,找出减小误差和控制误差的办法.

1. 误差的来源和归类

只要分析一下解决工程技术问题的流程,就不难发现误差的来源和归类大致如图0.2.1所示.

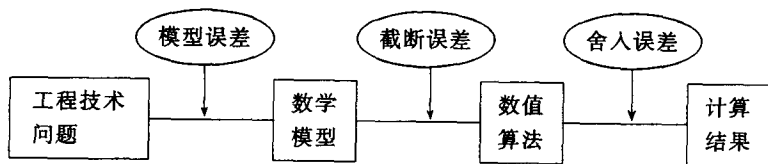


图 0.2.1 误差的来源和归类

模型误差是在工程技术问题向数学模型转化过程中产生的.在建立数学模型时,通常要加上一些限制,抓住主要的因素,忽略次要的因素,因此数学模型带有理

想化的描述,它与实际问题之间难免存在误差. 本书不讨论模型误差.

截断误差是在连续型的数学问题向离散型的数值算法转化过程中产生的. 例如, Taylor 展式

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + \text{余项}$$

有无限项,只有作有限项截断,才能形成具体计算 $f(x)$ 的近似值的公式

$$f_n(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n,$$

其中丢掉的“余项”直接反映了近似值 $f_n(x)$ 与准确值 $f(x)$ 之间的误差,这就是截断误差. 一般地讲,连续型的数学问题常常可表示为一个极限问题,为了实现计算,需将无穷过程作有限截断处理,具体计算出近似值,由此产生的误差就是截断误差. 截断误差是算法加工时产生的,它既是现实的又是理想的.

舍入误差是计算机作数值计算时产生的. 例如,计算 3π ($\pi=3.141\,592\,65\cdots$) 时, π 有无限个小数位,计算机只能取一定字长来计算. 据四舍五入的规则,若取 4 位字长计算,计算结果为 3×3.142 ; 若取 8 位字长计算,计算结果为 $3 \times 3.141\,592\,7$. 一般地讲,计算过程中某些数据的位数可能很多,也可能是无穷小数,计算机只能用有限字长计算,必须舍掉尾数,这就产生了舍入误差. 舍入误差是实实在在的,虽然它微不足道,但在复杂的计算过程中舍入误差可能会传播和积累,可能会被放大,可能会严重地影响计算结果.

2. 近似解的精度

将数学问题改写为计算机可执行的算法时存在截断误差,计算机在计算中每一步会产生舍入误差,后续计算又会在前步计算的基础上产生误差的积累,计算过程中处处是近似的且处处有误差. 那么计算结果怎样才能保证实际要求呢? 这就需要关于近似解靠近准确解程度的度量标准.

这里就单个近似值 S 和准确值 S^* 之间的关系来说明问题. 按照通常的习惯,采用三个标准来说明近似值 S 对于 S^* 的“准确”程度,即精度. 第一个标准是绝对误差,第二个标准是相对误差,第三个标准是数值的有效位数.

绝对误差表示形式为

$$|S - S^*| \leq \epsilon,$$

其中, ϵ 是一个较小的数值,例如 10^{-4} 或 10^{-8} 等. 显然 ϵ 越小, S 靠近 S^* 的程度越满意,此时可称近似值 S 具有绝对误差精度 ϵ . 这种精度标准类似于机械零件尺寸加工的情形,准确尺寸 S^* 是不知道的,加工出来的尺寸 S 是近似的,但加工的尺寸满足 $S^* - \epsilon \leq S \leq S^* + \epsilon$, 即 $S = S^* \pm \epsilon$ 时,就是满足精度要求的.

相对误差的表示形式为

$$\frac{|S - S^*|}{|S|} \leq \epsilon.$$

它和绝对误差在表现误差大小方面有什么区别呢?下面用一个例子来说明.

例如,有 2 个无量纲测量值的表示形式为

$$S_1^* = 15 \pm 2, \quad S_2^* = 1\,000 \pm 5.$$

这表明准确值 S_1^* 未知,测量得到的近似值为 $S_1 = 15$,绝对误差为 $|S_1 - S_1^*| \leq 2$;准确值 S_2^* 未知,测量得到的近似值为 $S_2 = 1\,000$,绝对误差为 $|S_2 - S_2^*| \leq 5$. 仅比较绝对误差,前者小,后者大,但这不能说明后者的测量精度低. 因为绝对误差仅表现了误差值本身的大小,没有考虑准确值的大小. 相对误差则考虑了待测量值的大小. 这两种测量的相对误差分别为 $2/15 = 13.3\%$ 和 $5/1\,000 = 0.5\%$,这说明后者的测量精度高.

又例如,测量 100 m 长度时有 1 m 的误差,绝对误差为 1 m,相对误差为 1%;测量 1 m 长度时有 2 cm 的误差,绝对误差为 2 cm,相对误差为 2%. 二者比较,前者测量精度高. 这是因为绝对误差中可带有不同的单位长度;相对误差是一种无量纲的精度表示形式,表明单位长度的误差量. 显然,相对误差越小,近似值的精确度越高.

计算机数值计算的输出结果仅仅是一个近似的数值 S ,能否直接观察 S 对准确值 S^* 的近似程度呢? 近似值 S 所具有的有效位数字这个概念可直观表现 S 对 S^* 的近似程度.

大家知道,计算机中所有的数据都要按四舍五入的规则表示为规格化形式,如

$$S = \pm 10^k \times 0.a_1a_2\cdots a_n,$$

其中, k 是整数; $0.a_1a_2\cdots a_n$ 是规格化小数, $a_1 \neq 0, a_2 \sim a_n$ 这些数字可以是零或非零值. 显然, S 是在规格化小数的小数点后第 $(n+1)$ 位上发生舍入的结果,舍入量不超过舍入位的半个单位,即

$$|S - S^*| \leq \frac{1}{2} \times 10^{-n} \times 10^k.$$

由此可以看出,用近似值 S 表示准确值 S^* 时,其规格化小数的小数点后的 n 位数据都是有效的. 这种表示形式直观表明了 S 对 S^* 的准确程度,此时称 S 具有 n 位有效数字.

例如,在充分字长的情况下,计算机将大整数 $S^* = \pm 123\,456\,789$ 规格化处理为 $S = \pm 0.123\,456\,789 \times 10^9$. 若保留小数点后 4 位,按四舍五入的规则,计算机输出结果为 $S_1 = \pm 0.123\,5 \times 10^9$,这表示 S_1 有 4 位有效数字, $|S_1 - S^*| \leq \frac{1}{2} \times 10^{-4} \times 10^9$. 若保留小数点后 10 位,计算机输出结果为 $S_2 = \pm 0.123\,456\,789\,0 \times 10^9$,这表示 S_2 有 10 位有效数字, $|S_2 - S^*| \leq \frac{1}{2} \times 10^{-10} \times 10^9$.

例如,计算机可将纯小数 $S^* = \pm 0.002\,314\,159\,26\cdots$ 规格化表示为 $S = \pm 0.231\,415\,926\cdots \times 10^{-2}$. 输出值 $S_1 = \pm 0.231\,42 \times 10^{-2}$ 具有 5 位有效数字,

$|S_1 - S^*| \leq \frac{1}{2} \times 10^{-5} \times 10^{-2}$; 输出值 $S_2 = \pm 0.231\ 415\ 9 \times 10^{-2}$ 具有 7 位有效数字,

$|S_2 - S^*| \leq \frac{1}{2} \times 10^{-7} \times 10^{-2}$.

例如, 已知准确值 $S^* = 36.435\ 98$, 近似值

$$S_1 = 36.436\ 66, \quad S_2 = 36.436\ 47, \quad S_3 = 36.436\ 28,$$

求 S_1, S_2 和 S_3 的有效位数. 先用规格化形式表示为

$$S^* = 0.364\ 359\ 8 \times 10^2, \quad S_1 = 0.364\ 366\ 6 \times 10^2,$$

$$S_2 = 0.364\ 364\ 7 \times 10^2, \quad S_3 = 0.364\ 362\ 8 \times 10^2,$$

再据有效位的定义, 有

$$|S^* - S_1| = 0.000\ 006\ 8 \times 10^2 \leq \frac{1}{2} \times 10^{-4} \times 10^2,$$

$$|S^* - S_2| = 0.000\ 004\ 9 \times 10^2 \leq \frac{1}{2} \times 10^{-5} \times 10^2,$$

$$|S^* - S_3| = 0.000\ 003\ 0 \times 10^2 \leq \frac{1}{2} \times 10^{-5} \times 10^2.$$

由此可知 S_1 具有 4 位有效数字, S_2 和 S_3 都是具有 5 位有效数字的近似数.

以上几个例子介绍的是在准确值 S^* 已知的情况下, 关于确定近似值 S 的有效位的办法. 在实际计算过程中常常会发生准确值 S^* 未知的情形, 按某种算法算得 S^* 的近似值 S , 该算法能保证

$$|S - S^*| \leq \frac{1}{2} \times 10^{-m},$$

计算结果输出 S 的形式为

$$S = S \text{ 规格化小数} \times 10^k.$$

那么此时 S 是具有多少有效位的近似值呢?

首先由 S 充分接近 S^* 可知, S^* 可表示为

$$S^* = S^* \text{ 规格化小数} \times 10^k,$$

那么

$$|S - S^*| = |S \text{ 规格化小数} - S^* \text{ 规格化小数}| \times 10^k \leq \frac{1}{2} \times 10^{-m}$$

可改写为

$$|S \text{ 规格化小数} - S^* \text{ 规格化小数}| \leq \frac{1}{2} \times 10^{-(m+k)}.$$

因此 S 的规格化小数具有 $(m+k)$ 位有效数字, 近似值 S 具有 $(m+k)$ 位有效数字.

综上所述, 近似数值 S 的有效位数越多, 它对准确值 S^* 的接近程度就越好. 近似值 S 的有效位数的多少是评定计算结果好坏的标准之一, 也是评价算法好坏的标准之一.

0.3 注意学习方法

1. 要带着实用的欲望学习该课程

本课程所学习的一些计算方法,简言之,是关于高等数学和线性代数课程中的基本数学问题数值化处理的方法,它是科技计算的基础,也是IT行业的基础. 该课程中的每一类问题都有着广泛的应用背景,既有算法又有具体实现过程,因此读者应摆脱关于理论课程的学习方法,带着分析问题、解决问题和实际应用的目的来学习本课程中的有关内容.

2. 要注重算法思想

本课程在将某一类数学问题转化为数值算法的过程中,综合使用了多种基础知识,具体算法似乎显得繁杂. 但要注意,设计算法的基本思想始终是直观明快的. 因此,只有注重算法思想的学习,才能无碍地记住算法和相应的计算公式,才能培养关于分析问题和解决实际问题的能力,死记硬背是无效的.

3. 要注重实践性环节

现今阶段,无需对每个算法编程上机了,每种算法都有相应的软件,学习阶段的实践仅需要调用这些程序去解决一个简单问题,观察其解决的效果. 这样的实践过程虽然简单,但却是必要的,因为这样可一般性地了解具体的计算过程,可加深对各种算法优缺点和最佳适应范围的理解.

第1章 曲线插值和曲线拟合

曲线插值问题,直观地说,认为已知的一批数据点 $(x_k, f_k)_{k=0}^n$ 是准确的,这些数据点所表现的准确函数关系 $f(x)$ 是未知的;在这种情况下要作一条近似曲线 $S(x)$ 且点点通过这些数据点. 插值问题不仅要讨论这种近似曲线 $S(x)$ 的构造方法,要讨论这种近似所产生的误差,还要讨论当插值点增多时这种近似曲线 $S(x)$ 是否稳定地收敛于未知函数 $f(x)$.

曲线拟合问题,直观地说,认为已知的一批数据点 $(x_k, f_k)_{k=0}^n$ 可能是不准确的,在这种情况下要作一条近似曲线 $S(x)$ 能宏观地描述这些数据点所反映的变化规律.

无论曲线插值问题还是曲线拟合问题,都可看做近似函数 $S(x)$ 的构造问题. 人们可用已知的 $S(x)$ 来近似讨论未知的 $f(x)$,这种近似方法在工程技术理论研究和应用研究中都是非常重要的.

1.1 Lagrange 插值多项式

1. 背景要求与 Lagrange 插值问题的提法

在生产实践和科学实验中,常常不能具体写出反映真实现象的函数 $f(x)$ 的表达式,但可通过实验测得若干离散的函数值 $\{f(x_k)\}_{k=0}^n$,如表 1.1.1 所示.

表 1.1.1 离散的函数值

x_k	x_0	x_1	$\cdots$	x_n
$f(x_k)$	f_0	f_1	$\cdots$	f_n

人们希望利用这些有限的信息构造一个近似函数,该近似函数对表中的实测值是准确的,该近似函数能近似描述 $f(x)$ 的变化规律.

由于 $n+1$ 个无关的条件能唯一地确定一个 n 次代数多项式函数 $L_n(x)$,所以用多项式 $L_n(x)$ 近似表述 $f(x)$ 是合理的. 于是,有 Lagrange 插值多项式问题的提法:

已知区间 $[a, b]$ 的等距或不等距节点 $\{x_k\}_{k=0}^n$ 和这些节点处的函数值 $\{f_k\}_{k=0}^n$,求 n 次插值多项式 $L_n(x)$,使得

$$L_n(x_k) = f_k, \quad k=0, 1, \cdots, n.$$

人们可用一种笨拙的办法获得 $L_n(x)$,即令

$$L_n(x) = a_0 + a_1x + \cdots + a_nx^n,$$

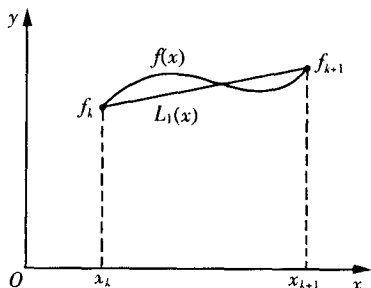
将 $n+1$ 个无关的插值条件代入,从而获得关于 $n+1$ 个未知数的线性代数方程组

$$f_k = a_0 + a_1 x_k + \cdots + a_n x_k^n, \quad k=0, 1, 2, \cdots, n,$$

解得 $a_0, a_1, \cdots, a_n$ 就可得到 $L_n(x)$. 然而这种办法不仅麻烦、计算量大, 而且求解方程组的计算过程会出现较大的误差, 是不可取的. 下面从简单的线性插值和二次插值例子中总结出一套构造一般 Lagrange 插值的简便方法.

2. 两点确定线性插值函数

假设已知两个插值节点 x_k 和 x_{k+1} , 已知相应的函数值 f_k 和 f_{k+1} , 要构造一个线性插值函数 $L_1(x) = ax + b$, 使得 $L_1(x_k) = f_k, L_1(x_{k+1}) = f_{k+1}$.



该线性插值函数 $L_1(x)$ 如图 1.1.1 所示, 它是很容易构造的, 可利用构造直线的点斜式方程写出, 即

$$L_1(x) = f_k + \frac{f_{k+1} - f_k}{x_{k+1} - x_k}(x - x_k). \quad (1.1.1)$$

特别地, 式 (1.1.1) 可变形为两点式方程

$$L_1(x) = f_k l_k(x) + f_{k+1} l_{k+1}(x) \quad (1.1.2)$$

其中

$$l_k(x) = \frac{x_{k+1} - x}{x_{k+1} - x_k}, \quad l_{k+1}(x) = \frac{x - x_k}{x_{k+1} - x_k}.$$

称 $l_k(x)$ 是关于节点 x_k 的节点基函数, 称 $l_{k+1}(x)$ 是关于节点 x_{k+1} 的节点基函数.

$L_1(x)$ 用式 (1.1.2) 的表示形式是有特点的, 它突出了每个节点处对应着一个基函数, 插值函数是这些节点基函数的线性组合形式, 其组合系数就是节点处的函数值. 也正是由于这种表现形式上的特点, 节点基函数可相应地表现为

$$l_k(x_m) = \begin{cases} 1, & m=k, \\ 0, & m \neq k, \end{cases} \quad l_{k+1}(x_m) = \begin{cases} 1, & m=k+1, \\ 0, & m \neq k+1, \end{cases}$$

它们都是 $[x_k, x_{k+1}]$ 上的线性多项式函数, 其图形如图 1.1.2 所示.

线性插值函数 $L_1(x)$ 近似 $f(x)$ 的效果可由其误差函数

$$R_1(x) = f(x) - L_1(x)$$

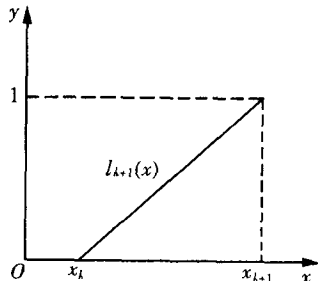
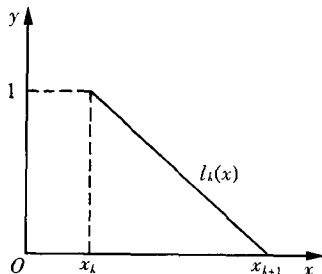


图 1.1.2 线性插值基函数图形

来描述,那么线性插值函数的误差函数有什么特点又如何确定呢?第一,由于插值的原因,线性插值函数在节点处的误差为零,其误差函数应具有的形式为

$$R_1(x) = c(x-x_k)(x-x_{k+1}).$$

第二,当 $f(x)$ 就是线性函数时, $f(x)$ 和 $L_1(x)$ 就是一致的.所以,线性插值函数的误差函数一定是一个二次多项式.第三, $R_1(x)$ 的具体形式为

$$R_1(x) = f(x) - L_1(x) = \frac{f''(\xi)}{2!}(x-x_k)(x-x_{k+1}),$$

$$\xi \in [x_k, x_{k+1}], \quad (1.1.3)$$

此误差表达式的具体证明可参见定理1.1.1.

3. 三点确定二次多项式插值函数

假设已知三个节点 x_k, x_{k+1}, x_{k+2} ,它们可能是等距离的,也可能是不等距离的,还已知相应的互异的函数值 f_k, f_{k+1}, f_{k+2} ,要构造一个二次多项式插值函数 $L_2(x)$,使得

$$\begin{aligned} L_2(x_k) &= f_k, \\ L_2(x_{k+1}) &= f_{k+1}, \\ L_2(x_{k+2}) &= f_{k+2}. \end{aligned}$$

三个插值条件可唯一地确定一个二次多项式插值函数,如图1.1.3所示.

需要特别强调的是,我们需要像式(1.1.2)那样的节点基函数的线性组合形式,即

$$L_2(x) = f_k l_k(x) + f_{k+1} l_{k+1}(x) + f_{k+2} l_{k+2}(x). \quad (1.1.4)$$

因为 $L_2(x)$ 是一个二次多项式,所以可推知式(1.1.4)中的三个节点基函数都是二次多项式;因为在节点 x_k 处的函数值 f_k 在式(1.1.4)中仅仅由 $f_k l_k(x)$ 这一项来表现,所以可推知 $l_k(x_k) = 1, l_k(x_{k+1}) = l_k(x_{k+2}) = 0, \dots$.总之,式(1.1.4)的形式特点决定了节点基函数的性质;三个节点基函数 $l_k(x), l_{k+1}(x)$ 和 $l_{k+2}(x)$ 都是二次多项式;它们在插值节点处的表现如表1.1.2所示.

表 1.1.2 节点基函数在插值节点处的表现

	x_k	x_{k+1}	x_{k+2}
$l_k(x)$	1	0	0
$l_{k+1}(x)$	0	1	0
$l_{k+2}(x)$	0	0	1

下面介绍确定节点基函数 $l_k(x)$ 的简便办法.首先因为 $l_k(x)$ 是一个二次多项式,且 $l_k(x_{k+1}) = l_k(x_{k+2}) = 0$,所以 $l_k(x)$ 中必定含有 $(x-x_{k+1})(x-x_{k+2})$ 因式,且 $l_k(x)$ 必定具有如下形式:

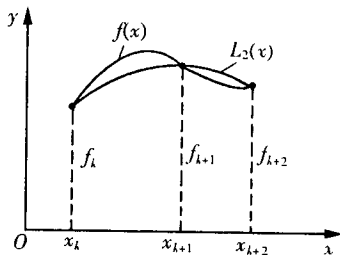


图 1.1.3 二次插值函数的图形