全国高职高专规划教材

计算方法

Computing Methods

江爱民 主 编
张有正 副主编

全国
高职高专
规划教材

科学出版社
www.sciencep.com

全国高职高专规划教材

计 算 方 法

江爱民　主　编
张有正　副主编

科学出版社
北 京

内 容 简 介

本书介绍了计算机常用的计算方法及其基础理论。主要内容包括误差及误差分析、插值与逼近、非线性方程的数值解法、数值积分、常微分方程初值问题的数值解法、矩阵及线性方程组和遗传算法等。

本书针对高职高专计算机专业学生的教学特点和需要取材，内容由浅入深，比较全面地介绍各种数值计算的算法。对每种算法简化理论上的严格推导与证明，力求做到简洁、易懂、实用，注重培养学生的工程计算能力。每一章均附有适量的习题。

本书可作为高职高专计算机专业计算方法课程的教材，也可作为相关专业工程技术人员的参考书。

图书在版编目（CIP）数据

计算方法/江爱民主编，一北京：科学出版社，2004
（全国高职高专规划教材）
ISBN 7-03-012698-X

Ⅰ.计… Ⅱ.江… Ⅲ.电子计算机—计算方法—高等学校：技术学校—教材 Ⅳ.TP301.6

中国版本图书馆 CIP 数据核字（2003）第 123097 号

责任策划：李振格/责任编辑：王日臣
责任印制：吕春珉/封面设计：东方人华平面设计部

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
新蕾印刷厂 印刷
科学出版社发行 各地新华书店经销
*
2004年2月第 一 版 开本：787×1092 1/16
2004年2月第一次印刷 印张：11 1/4
印数：1—5 000 字数：249 000
定价：15.00元

全国高职高专规划教材编委会名单

本书编写人员名单

主　编　江爱民

副主编　张有正

撰稿人　朱益民　李祥龙　江爱民　张有正　周柏青

　　　　郑启富

前　言

计算机在科学研究、工程设计和管理中得到日益广泛的应用。计算机常用的数值计算方法（计算方法），是提高学生的工程计算能力的必修课程，已受到学校的高度重视，“计算方法”也已成为高职高专计算机专业学生必修的专业基础课程。

本书是教师在总结多年的教学经验基础上编写而成的。内容力求较全面地介绍各种数值计算的算法，对每种算法简略理论上的严格推导与证明，力求做到简洁、易懂、实用，注重培养学生的工程计算能力。

全书安排了 7 章内容，每章均附有适量的习题，对部分标有*号的内容可作为选学或参考阅读资料。要强调指出，为使学生掌握这门课程的基本知识和计算技能，教师应在每章讲授后布置一定量课外作业，本课程还应安排学生完成三次以上计算机实习。

本书可作为高职、高专计算机专业的教材，也可以作为自学用书或参考资料。

本书由江爱民任主编，张有正任副主编，其中，第 1 章由朱益民编写，第 2 章由李祥龙编写，江爱民编写第 3 章，张有正编写第 4、5 章，第 6 章由周柏青编写，第 7 章由郑启富编写。

受水平所限，书中的错漏和不妥之处敬请兄弟院校同行和读者指正。

编　者

2003年6月

目　录

第 1 章　误差及误差分析......1

1.1　误差的来源与分类......1

1.2　绝对误差与相对误差......2

1.2.1　绝对误差......2

1.2.2　有效数字......3

1.2.3　相对误差......4

1.3　运算误差分析......6

1.3.1　和、差、积、商的误差......6

1.3.2　在近似计算中需要注意的一些现象......7

第 2 章　插值法与逼近......10

2.1　插值与逼近的基本概念......10

2.1.1　插值与逼近的概念......10

2.1.2　插值问题的提法......10

2.1.3　插值多项式的存在惟一性......11

2.2　拉格朗日插值多项式......12

2.2.1　插值基函数......12

2.2.2　拉格朗日插值多项式......12

2.2.3　插值余项......17

2.2.4　误差的事后估计......18

2.3　牛顿插值多项式......19

2.3.1　差商（均差）和牛顿插值多项式......19

2.3.2　差分和牛顿插值公式......22

*2.4　Hermite 插值......29

2.4.1　Hermite 插值......29

2.4.2　二重密切 Hermite 插值多项式......30

*2.5　分段低次插值......33

2.5.1　分段线性插值......34

2.5.2　分段二次插值......35

2.5.3　分段三次 Hermite 插值多项式......35

*2.6　三次样条插值......36

2.6.1　样条函数的概念......37

2.6.2　三次样条插值函数的定义......37

2.6.3　边界条件和三次样条插值多项式......38

2.7　最小二乘法......48

2.7.1 样最小二乘法的概念 48
2.7.2 法方程组和最小二乘解的求法 49
*2.7.3 利用正交函数做多项式拟合 56
2.8 数值微分 60
2.8.1 差商型数值微分公式 60
2.8.2 插值型求导公式 61
2.8.3 三次样条插值数值微分公式 62
2.8.4 数值微分问题化为数值积分问题 64
2.8.5 数值微分的外推算法 66
第 3 章 非线性方程的数值解法 71
3.1 二分法 71
3.1.1 根所在范围的确定 71
3.1.1 二分法 72
3.2 简单迭代法 74
3.2.1 简单迭代法的基本过程 74
3.2.2 简单迭代法的几何解释与收敛性 75
3.3 牛顿迭代法 78
3.3.1 公式的导出 78
3.3.2 牛顿迭代法的几何意义及收敛讨论 79
3.4 插值法 82
第 4 章 数值积分 87
4.1 数值积分基本思想 87
4.1.1 数值积分的必要性 87
4.1.1 数值积分的基本思想 88
4.2 牛顿-柯特斯公式 88
4.3 梯形求积公式 91
4.3.1 基本公式 91
4.3.2 变步长梯形公式 93
4.4 辛卜生公式及龙贝格求积法 95
4.4.1 辛卜生基本求积公式 95
4.4.2 复化辛卜生公式 96
4.4.3 龙贝格求积法 97
4.5 高斯（gauss）求积公式 101
4.5.1 代数精度的概念 101
4.5.2 高斯求积公式 101
第 5 章 常微分方程初值问题 的数值解法 106
5.1 数值解法的基本思想 106

5.2 欧拉方法 109
5.2.1 基本公式 109
5.2.2 欧拉公式的几何解释 110
5.2.3 欧拉方法的误差估计 111
5.2.4 改进的欧拉公式 112
5.3 龙格-库塔法 114
5.3.1 原理 114
5.3.2 龙格-库塔公式 115
5.3.3 一阶微分方程组的龙格-库塔公式 118
5.4 线性多步法 120
第 6 章 矩阵及线性方程组 124
6.1 线性方程组的直接解法 124
6.1.1 直接法概述 124
6.1.2 三角形线性方程组的解法 125
6.1.3 Gauss 消去法 126
6.1.4 选主元素的 Gauss 消去法 129
6.2 矩阵的三角分解 134
6.3 解三对角线方程组的追赶法 137
6.4 矩阵求逆 140
6.4.1 求逆矩阵的 Gauss-Jordan 列主元素法 140
6.4.2 算法设计 144
***第 7 章 遗传算法** 148
7.1 概述 148
7.1.1 遗传算法是一种仿生优化算法 148
7.1.2 遗传算法的发展及现状 149
7.2 基本遗传算法 150
7.2.1 基本遗传算法的构成要素 150
7.2.2 基本遗传算法的实现 151
7.2.3 基本遗传算法应用举例 154
7.3 遗传算法的改进 156
7.3.1 分层遗传算法 156
7.3.2 CHC 算法 157
7.3.3 自适应遗传算法 159
7.3.4 基于小生境技术的遗传算法 160
7.3.5 混合遗传算法 162
7.4 遗传算法的应用 165

第 1 章　误差及误差分析

一个实际问题的解决往往是通过对这个问题实际背景进行简化、假设，建立相应的数学模型，通过对数学模型的求解来实现问题的解决。而模型的求解一般要用数值计算的方法，所谓数值计算是指将所求解的数学模型简化成一系列的算术运算，以便在计算机上求出问题的数值解，并对算法的收敛性、稳定性和误差进行分析、计算。

1.1　误差的来源与分类

用数学工具来解决实际问题在哪些地方会产生误差呢？为了分析误差的来源，我们先来分析一下实际问题的解决需要经过哪些步骤，在哪些环节会产生误差。

首先在建立数学模型当中，由于数学模型是实际问题的简化与抽象，同实际问题有差别，会产生误差，这类误差我们称为模型误差，比如说，根据牛顿定理建立起来的自由落体的下降的高度与时间的关系式为：

$$s=\frac{1}{2}gt^2$$

在这个关系式中，我们把重力加速度看成常数，物体不受空气阻力影响，这当然与实际有区别，所以模型本身也有误差。

数学模型建立后，模型的解往往不能用解析式来表示，要借助于计算机进行数值求解。而计算机只能执行人们事先编制好的指令，作一些规定的、且是有限次的运算或判断，以及在一些规定的设备上进行数据输入或输出。因此，对于计算工作者来说，还必须解决计算的近似公式，以及求解近似公式的计算方案。以便于利用计算机来进行计算。那么，在这过程中会有多个环节产生误差：

（1）公式中可能有一些参数，这些参数可能是从通过观测的或实验得到的，它们与真数之间有一定的差异，这就给计算带来了一定的误差。这种误差称为观测误差。比如$s=\frac{1}{2}gt^2$中的g。

许多数学运算（如微分、积分与无穷级数求和等）是通过极限过程来定义的，而实际上计算机只能完成有限次的算术运算与逻辑运算。因此，在实际应用时，还需将解题方案加工成算术运算与逻辑运算的有限序列，而这种加工又往往表现为某无穷过程的“截断”。例如，对于收剑的无穷级数，常用它的前面有限项的和来代替无穷级数的和，实际上抛弃了无穷级数的后段，由此便产生了误差。如用梯形公式计算积分的近似值，这方法本身就有误差。这类误差统称为方法误差或截断误差。

（2）公式中的一些常数比如π，e，$\sqrt{2}$等等，它在参加运算时只能取有限位有效

数字，这样就产生了舍入误差。

综上所述，误差在数值计算中是不可避免的。也就是说，在数值计算方法中，绝大多数情况下不存在绝对的严格和精确。对于一个好的计算工作者来说，主要在于能够分析误差产生的原因，并将误差限制在许可的范围之内。

当然除了上所这些原因引起的误差不可避免以外，有些工作中粗心大意等人为的原因引起的差错要尽量加以避免。比如说误将 688 写成为 868 及公式应用错误等。

最后还要说明的是，除了对这些误差本身要引起足够的重视外，还要注意到这些误差在计算过程中所产生的效应。例如，某个参数由于观测引起的误差可能是微不足道的，或者少量的舍入误差对中间的计算结果影响并不大，但是，这些误差经过计算机的千百万次运算以后，误差的积累就可能大得惊人。初始数据的微小误差也可能会引起严重错误，甚至会导致完全错误的结果。因此，挑选和设计好的算法，是一个很重要的环节，必须加以足够的重视。

1.2 绝对误差与相对误差

1.2.1 绝对误差

设 x^* 是精确值 x 的近似值，则 $\Delta x = x - x^*$ 称为近似值 x^* 的绝对误差，简称误差。由于精确值 x 往往是不知道的，所以 Δx 难以求得，只能根据具体的测量或计算求得它的范围，也就是估计 $|\Delta x|$ 的上界，记 $|\Delta x|$ 的上界为 η，称 η 为 x^* 的绝对误差限：

$$|\Delta x| = |x - x^*| \le \eta \tag{1.2.1}$$

进而有

$$x^* - \eta \le x \le x^* + \eta \tag{1.2.2}$$

有时也用

$$x = x^* \pm \eta \tag{1.2.3}$$

来表示 x 的精确度和取值范围。

由微分性质，在计算函数值时，如果给出了自变量的绝对误差，需要估计函数值的绝对误差时，通常可以用下式估计：

$$\Delta[f(x)] = f'(x)\Delta x$$

例 1.1 用有毫米刻度的尺测量不超过 1m 的长度 l。读数方法如下：如果长度 l 接近于毫米刻度 l^*，就读出那个刻度数 l^* 作为长度 l 的近似值。显然，这个近似值的绝对误差限就是 0.5mm，则有

$$|\Delta l| = |l - l^*| \leq 0.5\text{mm}$$

如果读出的长度是 314mm，则有

$$|l - 314| \leq 0.5$$

这样，我们虽然仍不知道准确长度 l 是多少，但由（1.2.2）可得：

$$313.5 \leq l \leq 314.5\text{mm}$$

这表明长度 l 在[313.5，314.5]这个区间内，可写成

$$l = 314 \pm 0.5\text{mm}$$

再比如，光速的近似值目前公认的是：

$$c^* = 2.997925 \times 10^{10}\,\text{cm/s}$$

通常记为：

$$c = (2.997925 \pm 0.000001) \times 10^{10}\,\text{cm/s}$$

1.2.2 有效数字

取 x 的近似值 x^*，我们通常用四舍五入的方法取前面几位

例 1.2　$x = \pi = 3.14159265\cdots$，按四舍五入原则有

取一位：$x_1^* = 3$，$\Delta x_1 \approx 0.14$

取三位：$x_3^* = 3.14$，$\Delta x_3 \approx 0.00146$

取五位：$x_5^* = 3.146$，$\Delta x_5 \approx -0.000007$

取六位：$x_6^* = 3.14159$，$\Delta x_6 = 0.000003$

如果近似值的误差限是某一位上的半个单位，该位到 x^* 的第一位非零数字共有 n 位，我们就说 x^* 有“n 位有效数字”。或者说 x^* 准确到该位。用四舍五入法取准确值的前 n 位作为近似值 x^*，则 x^* 有 n 位有效数字。比如，$x_3^* = 3.14$　是以三位有效数字来表示 π，它的误差限为：

$$|\pi - x_3^*| \leq 0.5 \times 10^{-2}$$

$x_5^* = 3.1416$ 是以五位有效数字来表示 π，它的误差限为：

$$|\pi - x_5^*| \le 0.5 \times 10^{-4}$$

一般，如果将x^*写成如下形式：

$$x^* = \pm 10^m (x_1 \times 10^{-1} + x_2 \times 10^{-2} + \cdots + x_n \times 10^{-n})^n \tag{1.2.4}$$

其中，x_1，x_2,⋯，x_n是 0 至 9 这 10 个数字之一，且$x_1 \ne 0$，n是正整数，m是整数。若

$$|x - x^*| \le 0.5 \times 10^{m-n} \tag{1.2.5}$$

则称近似数x^*有n位有效数字。

1.2.3 相对误差

绝对误差是不能准确地反映x^*的精确度的，例如：$x = 5000 \pm 10$ 与$x = 100 \pm 10$两者的绝对误差都是 10，但是显然前者的精确度大大优于后者。故我们引进

$$\delta x = \frac{x - x^*}{x} = \frac{\Delta x}{x}, \quad x \ne 0 \tag{1.2.6}$$

称δx为x^*的相对误差。

如上例中，前者的相对误差为±0.002，而后者的相对误差为±0.1，用相对误差较好地反映了x^*的近似程度。

相对误差说明了近似x^*的绝对误差与x^*本身比较起来所占的比例，从另一方面反映了一个近似数的准确程度。

和绝对误差一样，由于准确值x并不知道，其绝对误差$\Delta x = x - x^*$无法准确地算出，因此也就无法确定出相对误差δx的准确值，只能估计出它的一个范围。有时也用

$$\delta x = \frac{x - x^*}{x^*} = \frac{\Delta x}{x^*}$$

来定义相对误差。我们来估计$|\delta x|$的界：

$$|\delta x| = \left|\frac{\Delta x}{x^*}\right| \le \left|\frac{\eta}{x^*}\right| = \mu$$

称 μ 为 x^* 的相对误差限。

通常也把它简称为相对误差。显然，相对误差限 μ 与绝对误差限 η 有如下关系：

$$\mu = \left|\frac{\eta}{x^*}\right|$$

根据有效数字的基本概念，可以推出以下两个结论。

结论 1.1　若近似值 x^* 具有 n 位有效数字，则其相对误差满足

$$|\delta x| \le \frac{1}{2x_1} \times 10^{-(n-1)} \tag{1.2.7}$$

其中 x_1 为最左边的一位有效数字。

证明：　由 x^* 的表示形式（1.2.4）可知

$$|x^*| \ge x_1 \times 10^{m-1}$$

又因为已知 x^* 具有 n 位有效数字，则由式（1.2.5）知

$$|x - x^*| \le 0.5 \times 10^{m-n}$$

因此，x^* 的相对误差为

$$|\delta x| = \left|\frac{x - x^*}{x^*}\right| \le \frac{\frac{1}{2} \times 10^{m-n}}{x_1 \times 10^{m-1}} = \frac{1}{2x_1} \times 10^{-(n-1)}$$

证毕。

例 1.3　用 x^*=2.72 来表示 e 具有三位有效数字的近似值，则相对误差限是：

$$\mu \le \frac{1}{2 \times 2} \times 10^{-2}$$

在实际应用时，为了要使取的近似数的相对误差满足一定的要求，可以用式（1.2.7）来确定所取的近似数应具有多少位有效数字。例如，为了要取一个 $\sqrt{20}$ 的近似数其相对误差不超过 0.1%，则可以用式（1.2.7）来确定所取的近似数应有多少位有效数字。我们可以确定 $x_1 = 4$，代入式（1.2.7）即可求出有效数字的位数。

结论 1.2 若一个近似数 x^* 的相对误差为：

$$|\delta x| \le \frac{1}{2(x_1+1)} \times 10^{-(n-1)} \tag{1.2.8}$$

则这个近似数准确到 n 位有效数字。其中 x_1 为最左边的一位有效数字。

证（略）。

在实际应用时，为了要使取的近似数具有 n 位有效数字，要求所取的近似数的相对误差满足条件（1.2.8）。

1.3 运算误差分析

1.3.1 和、差、积、商的误差

设 x^* 是精确值 x 的近似值，设 y^* 是精确值 y 的近似值，用 $x^* \pm y^*$ 来表示 $x \pm y$ 的近似值，则它的误差为：

$$(x \pm y)-(x^* \pm y^*)=(x-x^*)\pm(y-y^*) \tag{1.3.1}$$

式（1.3.1）说明和的误差是误差之和，差的误差是误差之差。由于

$$|(x \pm y)-(x^* \pm y^*)| \le |x-x^*|+|y-y^*|$$

所以，和或差的误差限是误差限之和。以上的结论适用于任意多个近似值的和或差。任意多个数的和或差的误差等于各数的误差之和。

若我们把 x^* 的误差 $\Delta x = x - x^*$ 看成是 x 的微分

$$\mathrm{d}x = x - x^*$$

x^* 的相对相对误差是：

$$\delta x = \frac{x-x^*}{x} = \frac{\mathrm{d}x}{x} = \mathrm{d}\ln x \tag{1.3.2}$$

它是对数函数的微分。

设 $u = xy$，则 $\ln u = \ln x + \ln y$，因而

$$\mathrm{d}\ln u = \mathrm{d}\ln x + \mathrm{d}\ln y$$

这就是说，乘积的相对误差是各乘数的相对误差之和。同样可证，商的相对误差是

被除数与除数的相对误差之差。这是因为，若

$$u = \frac{x}{y}$$

则

$$\mathrm{d}\ln u = \mathrm{d}(\ln x - \ln y) = \mathrm{d}\ln x - \mathrm{d}\ln y$$

由此得到，任意多次连乘、连除所得的相对误差限等于各乘数和除数的相对误差限之和。

例 1.4　设

$$u = \frac{xy}{zw}$$

则因

$$\ln u = \ln x + \ln y - \ln z - \ln w$$

从而得到

$$\mathrm{d}|\ln u| \le \mathrm{d}|\ln x| + \mathrm{d}|\ln y| + \mathrm{d}|\ln z| + \mathrm{d}|\ln w|$$

u 的相对误差限等于乘 x、y 和除数 z、w 的相对误差限之和。

设 $y = f(x)$ ，$y^* = f(x)^*$，则 y^* 的相对误差是：

$$\mathrm{d}\ln y = \frac{f'(x)}{f(x)}\mathrm{d}x$$

例 1.5　设 $y = x^\alpha$，则 $\ln y = \alpha \ln x$，因此

$$\mathrm{d}\ln y = \alpha \mathrm{d}\ln x$$

x^α 的相对误差是 x 的相对误差的 α 倍。

1.3.2　在近似计算中需要注意的一些现象

（1）要避免两个相近的数相减。因为两正数之差 $u = x - y$（不妨设 $x > y$）的相对误差是：

$$\mathrm{d}\ln u = \mathrm{d}\ln(x - y) = \frac{\mathrm{d}x - \mathrm{d}y}{x - y}$$

如果 x 和 y 很接近，它们的差很小，因而 u 的相对误差很大。这是由于 x^* 和 y^* 的前几位有效数字必然相同，相减后有效数字会大大减少。遇到此类情况应当多保留几位

有效数字，或转化公式以防止此类问题的出现。

下面是一些常见公式转化的例子：

当 x 接近于 0 时，$\dfrac{1-\cos x}{\sin x}$ 应转化为 $\dfrac{\sin x}{1+\cos x}$。

当 x_1 与 x_2 很接近时，$\lg x_1 - \lg x_2 = \lg\dfrac{x_1}{x_2}$

当 x 很大时，$\sqrt{1+x}-\sqrt{x}=\dfrac{1}{\sqrt{1+x}+\sqrt{x}}$

（2）两个相差很大的数进行运算时，要防止“大数吃小数现象”。我们先看一个例子，设 $a=10^{11}$，$b=10$，$c=-10^{11}$，为了计算 $a+b+c$，如果按 $(a+b)+c$ 顺序来计算，而有效数字为十位以下，则在作 $a+b$ 的运算时，a“吃掉”了 b，但若按 $(a+b)+c$ 运算顺序，其结果为首先 a 与 c 互相抵消，再与 b 相加，其最后结果为 10，从而保护了数 b。所以在运算过程中必须注意合理安排运算顺序，以便提高运算的精度或保护重要的数。

例 1.6 说明了在运算过程中会有大数“吃掉”小数的现象。

例 1.6 在具有四位有效数字的计算机上作下列运算：

（1）10^3（0.8961）+10^{-5}（0.4688）

$\to 10^3$（0.8961）+10^3（0.0000） （对阶）

$\to 10^3$（0.8961） （规格化）

其结果是大数“吃掉”了小数。

（2）10^0（0.6108）+10^3（0.6871）

$\to 10^3$（0.0006）+10^3（0.6871） （对阶）

$\to 10^3$（0.6877） （规格化）

其结果是大数“吃掉”了部分的小数。

（3）10^{-1}（0.3311）-10^{-1}（0.3307）

$\to 10^{-1}$（0.0004）

$\to 10^{-4}$（0.4000） （规格化）

其计算结果的有效数字位数大大减少，尽管印出的结果为 10^{-4}（0.4000），但包括 4 在内都不一定是有效数字，之所以在小数点印出了四位，是计算机系统进行了规格化的结果。

（3）要注意计算步骤的简化，减少运算次数。从前面我们已经看到，运算过程中的每一步都有可能产生误差。而且这些误差还会积累到最终的结果中去。所以简化计算公式，减少运算次数十分重要，它直接影响计算的速度和误差的积累。下面我们以计算多项式的值为例来说明简化计算公式的重要性。

例 1.7 计算多项式

$$P(x)=a_n x^n + a_{n-1}x^{n-1}+\cdots+a_0 \tag{1.3.3}$$