面向 21世纪 高级应用型人才

中国高等职业技术教育研究会推荐

高职高专系列教材

计算方法与实习

田祥宏 编著

西安电子科技大学出版社
http://www.xduph.com

□ 中国高等职业技术教育研究会推荐

高职高专系列教材

计算方法与实习

田祥宏　编著

西安电子科技大学出版社

2004

内 容 简 介

本书是在中国高等职业技术教育研究会的指导下，由西安电子科技大学出版社组织编写的高职高专系列教材之一。本书在介绍算法时尽量由浅入深，有意忽略复杂繁琐的理论证明和推导，着重点在算法的理解与应用方面。本书主要内容包括误差知识、非线性方程的数值解法、线性方程组的数值解法、插值法、曲线拟合与最小二乘法、数值微分与数值积分、常微分方程数值解法等方面的基本概念、原理及算法。书中附有主要算法的框图和C语言程序实现。每一章都给出适量的习题，并附有部分习题的参考答案。本书最后一章(计算实习)供学生理解算法实习之用。

本书可作为普通高校非数学类专业本科生和计算机专业专科生以及成教本科、专科起点本科生的教材或参考书，也可作为有关工程技术人员和自学者的参考书。

★本书配有电子教案，需要者可与出版社联系，免费提供。

图书在版编目(CIP)数据

计算方法与实习/田祥宏编著．—西安：西安电子科技大学出版社，2004.4

(高职高专系列教材)

ISBN 7-5606-1363-2

Ⅰ．计…　Ⅱ．田…　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(2004)第 009266 号

策　　划　马乐惠

责任编辑　阎彬　马乐惠

出版发行　西安电子科技大学出版社(西安市太白南路2号)

电　　话　(029)88242885　88201467　　邮　编　710071

http://www.xduph.com　　E-mail：xdupfxb@pub.xaonline.com

经　　销　新华书店

印刷单位　西安兰翔印刷厂

版　　次　2004年4月第1版　2004年4月第1次印刷

开　　本　787毫米×1092毫米　1/16　印张9.5

字　　数　219千字

印　　数　1～4 000册

定　　价　11.00元

ISBN 7-5606-1363-2/O·0068(课)

XDUP 1634001-1

序

1999年以来，随着高等教育大众化步伐的加快，高等职业教育呈现出快速发展的形势。党和国家高度重视高等职业教育的改革和发展，出台了一系列相关的法律、法规、文件等，规范、推动了高等职业教育健康有序的发展。同时，社会对高等职业技术教育的认识在不断加强，高等技术应用型人才及其培养的重要性也正在被越来越多的人所认同。目前，高等职业技术教育在学校数、招生数和毕业生数等方面均占据了高等教育的半壁江山，成为高等教育的重要组成部分，在我国社会主义现代化建设事业中发挥着极其重要的作用。

在高等职业教育大发展的同时，也有着许多亟待解决的问题。其中最主要的是按照高等职业教育培养目标的要求，培养一批具有“双师素质”的中青年骨干教师；编写出一批有特色的基础课和专业主干课教材；创建一批教学工作优秀学校、特色专业和实训基地。

为解决当前信息及机电类精品高职教材不足的问题，西安电子科技大学出版社与中国高等职业技术教育研究会分两轮联合策划、组织编写了“计算机、通信电子及机电类专业”系列高职高专教材共100余种。这些教材的选题是在全国范围内近30所高职高专院校中，对教学计划和课程设置进行充分调研的基础上策划产生的。教材的编写采取公开招标的形式，以吸收尽可能多的优秀作者参与投标和编写。在此基础上，召开系列教材专家编委会，评审教材编写大纲，并对中标大纲提出修改、完善意见，确定主编、主审人选。该系列教材着力把握高职高专“重在技术能力培养”的原则，结合目标定位，注重在新颖性、实用性、可读性三个方面能有所突破，体现高职教材的特点。第一轮教材共36种，已于2001年全部出齐，从使用情况看，比较适合高等职业院校的需要，普遍受到各学校的欢迎，一再重印，其中《互联网实用技术与网页制作》在短短两年多的时间里先后重印6次，并获教育部2002年普通高校优秀教材二等奖。第二轮教材预计在2004年全部出齐。

教材建设是高等职业院校基本建设的主要工作之一，是教学内容改革的重要基础。为此，有关高职院校都十分重视教材建设，组织教师积极参加教材编写，为高职教材从无到有，从有到优、到特而辛勤工作。但高职教材的建设起步时间不长，还需要做艰苦的工作，我们殷切地希望广大从事高等职业教育的教师，在教书育人的同时，组织起来，共同努力，编写出一批高职教材的精品，为推出一批有特色的、高质量的高职教材作出积极的贡献。

中国高等职业技术教育研究会会长 李宗尧

IT 类专业系列高职高专教材编审专家委员会名单

主　任：高　林　（北京联合大学副校长，教授）

副主任：温希东　（深圳职业技术学院电子通信工程系主任，教授）

李卓玲　（沈阳电力高等专科学校信息工程系主任，教授）

李荣才　（西安电子科技大学出版社总编辑，教授）

计算机组：组长：李卓玲(兼)　（成员按姓氏笔画排列）

丁桂芝　（天津职业大学计算机工程系主任，教授）

王海春　（成都航空职业技术学院电子工程系副教授）

文益民　（湖南工业职业技术学院信息工程系主任，副教授）

朱乃立　（洛阳大学电子工程系主任，教授）

李　虹　（南京工业职业技术学院电气工程系副教授）

陈　晴　（武汉职业技术学院计算机科学系主任，副教授）

范剑波　（宁波高等专科学校电子技术工程系副主任，副教授）

陶　霖　（上海第二工业大学计算机学院教授）

徐人凤　（深圳职业技术学院计算机应用工程系副主任，高工）

章海鸥　（金陵科技学院计算机系副教授）

鲍有文　（北京联合大学信息学院副院长，副教授）

电子通信组：组长：温希东(兼)　（成员按姓氏笔画排列）

马晓明　（深圳职业技术学院电子通信工程系副主任，副教授）

于　冰　（宁波高等专科学校电子技术工程系副教授）

孙建京　（北京联合大学教务长，教授）

苏家健　（上海第二工业大学电子电气工程学院副院长，高工）

狄建雄　（南京工业职业技术学院电气工程系主任，副教授）

陈　方　（湖南工业职业技术学院电气工程系主任，副教授）

李建月　（洛阳大学电子工程系副主任，副教授）

李　川　（沈阳电力高等专科学校自动控制系副教授）

林训超　（成都航空职业技术学院电子工程系主任，副教授）

姚建永　（武汉职业技术学院电子信息系主任，副教授）

韩伟忠　（金陵科技学院龙蟠学院院长，高工）

项目总策划：梁家新

项目策划：马乐惠　云立实　马武装　马晓娟

电子教案：马武装

前　　言

随着现代科学技术的发展和计算机及网络技术的广泛应用，信息传播的速度已经明显提高，数值计算方法不再仅仅应用于纯数学领域，在工程实践中它也得到了很广泛的应用。因此，普通的工程技术人员和各行各业的设计人员都要用到数值计算方法。作为信息时代的大学生，更应了解数值计算方法这方面的知识，特别对于高等院校计算机专业的大学生、成教本科学生和自学考试本科学生来说，为面对今后的实际应用，学习数值计算方面的知识更为重要。

目前，关于计算方法的教材很多，但大多适用于理工科各专业的本科生和研究生，其内容对高等数学的基础要求很高，这类教材对于高职高专的学生不太适合。因此，本教材减少了许多数学方法的证明，而更注重于数值计算方法的实现，目的在于让学生理解算法，并能熟练使用各种数值计算方法。

本书共分 8 章。第 1～7 章系统介绍了各种常用的数值计算方法，包括计算方法的主要原理、思想及算法的实现。各章配有大量的例题解析和一定数量的习题，并附有参考答案，以便让读者通过对例题的学习和习题的练习而进一步理解和掌握各种数值计算方法。这部分建议用 32～40 学时讲授较为合适。第 8 章是计算实习，用于学生上机实习。通过这些实习题，可让学生更加深刻地理解各种数值计算方法。为更有效地指导学生上机实习和供学生自学，在每个实习题中对主要算法都给出了流程图和用标准 C 语言编写的程序，并附有上机实习的练习题。学生上机实习时间建议为 12～18 学时较为合适。

本书适合高职高专计算机专业的本、专科生和其它非数学专业工科学生或成人教育的本科生使用。

在本书的编写过程中，得到了金陵科技学院龙蟠学院计算机信息管理教研室许多老师的关心和支持，作者在此深表谢意。

由于作者水平有限，书中的缺点与错误在所难免，恳请读者批评指正。作者联系方式：avatian@sohu.com。

作　者

2003 年 12 月

目　录

第 1 章　绪　　论

1.1　计算方法的任务与算法的概念

计算方法是研究并解决数学问题的数值近似解的方法，是在计算机上使用的解数学问题的方法之一。其任务是研究适用于科学计算的数值计算方法及有关的数学理论，它是程序设计和对数值结果进行分析的依据和基础，是用计算机进行科学计算全过程的一个重要环节。

计算机实际上只会做加减乘除等算术运算和一些逻辑运算。由这些基本运算及运算顺序的规定构成的完整解题步骤，称为算法。它可用框图、算法语言、数学语言或自然语言描述。用计算机算法语言描述的算法称为计算机程序。

算法的优劣可以从运算工作量、占用的存储单元、逻辑结构、计算机程序编写的难易程度、计算结果的可靠性以及算法的收敛性和稳定性来加以鉴定。

1.2　误 差 知 识

1.2.1　误差的来源

科学计算中所处理的数据和计算的结果通常都是在一定范围内的近似数值，它们与实际的真实值之间总存在着误差。也就是说，一个物理量的真实值和我们算出的值往往不相等，其差值称为误差。引起误差的原因是多方面的，其主要来源有下面几种。

1. 模型误差

我们在解决实际问题时，通常需要将实际问题转化为数学问题，然后加以求解。这样的转化过程就是建立数学模型的过程。建立数学模型时，需要对实际问题进行抽象和简化，忽略一些次要因素。这样建立的数学模型虽然“精确”，但其实只是客观现象的一种近似。这种数学模型与实际问题之间的误差称为模型误差。

2. 测量误差

在给出的数学模型中往往会涉及到一些通过测量得到的物理量，比如电流、电压、电阻、长度、温度等，而在测量过程中由于所使用的测量工具和人为因素的影响，不可避免地会带来误差，这种误差称为测量误差。

3. 截断误差

在使用无穷级数求和时，只能取前面有限项的和来近似作为该级数的和，于是就产生了有限过程代替无限过程的误差。这种在计算中通过用有限过程的计算结果代替无限过程

的结果而造成的误差，称为截断误差，这是计算方法本身出现的误差，故又称为方法误差。

4. 舍入误差

在计算中由于四舍五入而产生的误差，称为舍入误差。这种误差的产生是由于在计算中遇到的数据可能位数很多，也可能是无穷小数(如 1/3 等)，但计算时只能对有限位进行运算，所以必须进行四舍五入。少量的舍入误差可能微不足道，但在计算机上完成了千百万次运算后，经过积累的舍入误差有时会非常惊人。

1.2.2 绝对误差、相对误差、有效数字

为了从不同的侧面表示近似数的精确程度，通常运用绝对误差、相对误差和有效数字的概念。

1. 绝对误差与绝对误差限

设 x^* 为准确值，x 是 x^* 的一个近似值，称 $e=x^*-x$ 为近似值 x 的绝对误差，简称误差。误差 e 可正可负，所以绝对误差不是误差的绝对值。通常无法准确地算出绝对误差的真值，只能根据具体测量或计算的情况估计它的大小的范围。设估计出 $|e|$ 的上界为 ε(某正数)，即

$$|e|=|x-x^*|\leqslant\varepsilon$$

则称 ε 为近似值 x 的绝对误差限，简称误差限，通常也把它称为绝对误差。

例如，用毫米刻度的直尺测量一长度为 x^* 的物体，测得的长度近似值为 $x=25$ mm，由于直尺以毫米为刻度，所以其误差不超过 0.5 mm，即

$$|x^*-25|\leqslant 0.5$$

从这个不等式不能得出精确值 x^*，但却知道 x^* 的范围，即

$$24.5\leqslant x^*\leqslant 25.5$$

显然这个近似值的绝对误差为 0.5 mm。由这个例子可以看出，绝对误差是有量纲单位的。同时可以看出，对于给定的正数 ε，若近似值 x 满足 $|x^*-x|\leqslant\varepsilon$，则在允许误差 ε 范围内认为 x 就是 x^*，也即近似值 x 和真值 x^* 关于允许误差 ε 可以看成是一样的，或说值 x 关于允许误差 ε 是“准确”的。

2. 相对误差与相对误差限

绝对误差的大小不能完全表达出近似值的准确程度。比如测得光速的近似值为 299 796 km/s，产生 4 km/s 的误差，而测量 100 m 的长度产生了 1 m 的误差。就绝对误差而言，前者是后者的 4000 倍，但是否就说明后者测量得更精确呢？其实不然，如果考虑被测量的数量本身的大小，前者的误差所占的比例为 4/299 796，而后者的误差所占的比例为 1%，显然前一种测量要精确些。由此可见，要看近似值的精确度，不仅要看绝对误差的大小，而且还要考虑近似值本身的大小，这就需要引入相对误差的概念。

设 x^* 为准确值，x 是 x^* 的一个近似值，则称

$$\frac{x^*-x}{x^*}=\frac{e}{x^*}$$

为近似值 x 的相对误差，记作 e_r。

但在实际计算中，通常真值 x^* 总是很难求得的。故常以

$$\bar{e}_r=\frac{x^*-x}{x}$$

作为相对误差。

事实上：

$$\bar{e}_r - e_r = \frac{\bar{e}_r^{2}}{1+\bar{e}_r} = \frac{e_r^2}{1-e_r}$$

所以，当$\bar{e}_r$和 e_r 有一个为小量时，$\bar{e}_r - e_r$ 就是该小量的二阶小量。

在实际计算中，计算相对误差与计算绝对误差具有相同的困难，因此通常也只考虑相对误差限，即如果有某正数 ε_r，使

$$|e_r| \leqslant \varepsilon_r \quad \text{或} \quad |\bar{e}_r| \leqslant \varepsilon_r$$

则称 ε_r 为 x 的相对误差限。

3. 有效数字

为了给出一种近似数，使之既能表示其大小，又能表示其精确程度，需要引进有效数字的概念。当准确值 x^* 有很多位数时，通常按四舍五入原则得到 x^* 的前几位近似值 x。例如：$x^*=\pi=3.1415926\cdots$，经四舍五入，若取前三位数字，则得 $x=3.14$，$\varepsilon\leqslant 0.002$；若取前五位数字，则得 $x=3.1416$，$\varepsilon\leqslant 0.000008$。它们的绝对误差都不超过末位数字的半个单位，即

$$|\pi - 3.14| \leqslant \frac{1}{2} \times 10^{-2}$$

$$|\pi - 3.1416| \leqslant \frac{1}{2} \times 10^{-4}$$

由上面的分析可知，如果近似值 x 的绝对误差限是某一位的半个单位，从该位到 x 的第1位非零数字共有 n 位数，我们就说 x 有 n 位有效数字(见图 1.1)。

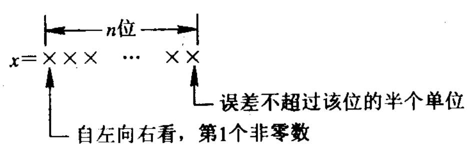

图 1.1　有效数字

如 π 的近似值取 $x=3.14$，则有 3 位有效数字；取 $x=3.1416$，则有 5 位有效数字。

例 1.1　对下列各数写出具有 5 位有效数字的近似值：

$$4565.3445,\ 0.004\,223\,45,\ 9.000\,023\,4,\ 9.000\,023\,4 \times 10^3$$

解　按有效数字的定义，上述各数具有 5 位有效数字的近似值分别是：

$$4565.3,\ 0.004\,223\,5,\ 9.0000,\ 9.0000 \times 10^3$$

注意 $x^*=9.000\,023\,4$ 的 5 位有效数字近似值是 9.0000，而不是 9，因为 9 只有 1 位有效数字。

例 1.2　指出下列各数有几位有效数字：

$$3.000\,044,\ -0.003\,00,\ -9000,\ 8 \times 10^3,\ 3 \times 10^{-3}$$

解　按定义，上述各数的有效位数分别是 7，3，4，1，1。

1.2.3　误差的危害及防止

在数值计算过程中，由于计算工具只能对有限位进行运算，因而在运算过程中不可避

免地要产生误差。如果对每步运算都分析误差，这是不可能的，也是不必要的。而实际上，在运算过程中所产生的误差的大小，通常又与运算步骤有关。为了鉴别计算结果的可靠性和防止误差危害现象的产生，在分析运算误差时，要考虑以下一些原则。

1. 尽量避免两相近数相减

在数值计算中，两个相近的数作减法时有效数字会有损失。例如，求下式的值：

$$y = \sqrt{2001} - \sqrt{1999}$$

设 $x_1^* = \sqrt{2001}$，$x_2^* = \sqrt{1999}$。它们的 6 位有效数字分别为 $x_1 = 44.7325$，$x_2 = 44.7102$。

第一种方法：$x_1^* - x_2^* \approx x_1 - x_2 = 44.7325 - 44.7102 = 0.0223$

分析一下这种方法的精度，由

$$|e(x_1 - x_2)| \approx |e(x_1) - e(x_2)| \leqslant |e(x_1)| + |e(x_2)| \leqslant \frac{1}{2} \times 10^{-4} + \frac{1}{2} \times 10^{-4}$$

$$= 10^{-4} < \frac{1}{2} \times 10^{-3}$$

可知，这种方法的计算结果有 3 位有效数字。

第二种方法：$x_1^* - x_2^* = \dfrac{2}{x_1^* + x_2^*} = \dfrac{2}{44.7325 + 44.7102} = 0.022\,360\,684\,5\cdots$

用这种方法可以取得有 6 位有效数字的近似结果 0.022 360 7。

由以上两种方法的对比来看，如果避免了两相近的数直接相减，那么得到的结果的精度比较高。

2. 防止大数“吃掉”小数

在数值计算中参加运算的数有时数量级相差很大，而计算机的位数有限，因此在进行加减法运算时，要对阶和规格化。对阶是以大数为基准，小数向大数对齐，即比较相加减两个数的阶，将阶小的尾数向右移，每移一位阶码加 1，直到小数阶码与大数阶码一致时为止，并将移位后的尾数多于字长的部分进行四舍五入，然后对尾数进行加减运算，最后将尾数变为规格化形式。所以当参加运算的两个数的数量级相差很大时，如不注意运算次序，就可能出现大数“吃掉”小数的现象，从而影响计算结果的可靠性。如在四位浮点机上作运算

$$0.7315 \times 10^3 + 0.4506 \times 10^{-5}$$

对阶后是 $0.7315 \times 10^3 + 0.0000 \times 10^3$，规格化后为 0.7315×10^3，造成大数“吃掉”小数的结果。

对于多个数相加，应按绝对值从小到大的顺序依次相加。设有

$$S_N = \sum_{j=2}^{N} \frac{1}{j^2 - 1}$$

在 486 PC 机上计算 S_{10}^3，S_{10}^4，S_{10}^5。按从小到大的顺序依次相加，计算结果可达到 8 位有效数字；若按从大到小的顺序相加，计算结果依次有 6 位、4 位及 3 位有效数字。

3. 尽量避免用绝对值太小的数做除数

用绝对值太小的数做除数，舍入误差会增大，而且当太小的除数稍有一点误差时，对计算结果影响很大。

例 1.3

$$\frac{2.7182}{0.001}=2718.2$$

如分母变为 0.0011，即分母只有 0.0001 的变化时，

$$\frac{2.7182}{0.0011}=2471.1$$

计算结果却有了很大变化。

4. 注意简化计算步骤，减少运算次数

运算过程的每一步都有可能产生误差，而且这些误差可能会积累到最终的结果中去，只不过这种误差的积累有时是增加的，有时因互相抵消而减少。因此在数值计算中，必须要考虑尽量简化计算步骤，这样一方面可以减小计算量，节省计算机的计算时间，另一方面，由于减少了运算次数，从而减少了产生误差的机会，也使误差积累有可能减小。

例 1.4 计算 x^{31}的值，如果逐个相乘，则要作 30 次乘法，但若改变成

$$x^{31}=xx^2x^4x^8x^{16}$$

则只要 8 次乘法运算就可以了。

又如计算多项式

$$f(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$$

的值。若直接计算 a_ix^{n-i}并逐项相加，一共需做

$$n+(n-1)+\cdots+2+1=\frac{n(n+1)}{2}$$

次乘法和 n 次加法。但是如果将前 n 项提出 x，则有

$$f(x)=(a_0x^{n-1}+a_1x^{n-2}+\cdots+a_{n-1})x+a_n$$

再对括号内 $n-1$ 次多项式施行同样方法，又有

$$f(x)=((a_0x^{n-2}+a_1x^{n-3}+\cdots+a_{n-2})x+a_{n-1})x+a_n$$

对内层括号的 $n-2$ 次多项式再施行上述同样的方法，又得一个 $n-3$ 次多项式，这样每作一步，最内层的多项式就降低一次，最终可将多项式表述成如下形式

$$f(x)=[(\cdots((a_0x+a_1)x+a_2)x+\cdots+a_{n-2})x+a_{n-1}]x+a_n$$

利用此式结构上的特点，从里往外一层层地计算。设

$$\begin{aligned}
b_0&=a_0\\
b_1&=b_0x+a_1\\
b_2&=b_1x+a_2\\
&\vdots\\
b_k&=b_{k-1}x+a_k\\
&\vdots\\
b_n&=b_{n-1}x+a_n=f(x)
\end{aligned}$$

得递推公式：

$$\begin{cases}b_k=b_{k-1}x+a_k & (k=1,2,\cdots,n)\\ b_0=a_0\end{cases}$$

于是 $f(x)=b_n$。这种多项式求值的算法称为秦九韶法，它是我国宋代数学家秦九韶最先提

出的。按此法求 $f(x)$只需做 n 次乘法和 n 次加法，工作量少，又便于编程实现。

若采用计算器或手算也很方便。我们把 $f(x)$按降幂排列的系数写在第 1 行，把欲求某点之值 x_0 及 b_kx_0 写在第 2 行，第 3 行为第 1、第 2 两行相应值之和 b_k，最后得到的 b_n 即为所求 $f(x_0)$的值，如下所示：

	a_0	a_1	a_2	$\cdots$	a_{n-1}	a_n
$x=x_0$		b_0x_0	b_1x_0	$\cdots$	$b_{n-2}x_0$	b_nx_0
	b_0	b_1	b_2	$\cdots$	b_{n-1}	b_n

例 1.5　计算 $f(x)=2+x-x^2+3x^4$ 在 $x_0=2$ 时的值。

	3	0	-1	1	2
$x=2$		6	12	22	46
	3	6	11	23	$48=f(2)$

得 $f(2)=48$。

5. 选用数值稳定的计算公式

定量地分析舍入误差的积累对大多数的算法来说是非常困难的，如果在执行算法的过程中舍入误差在一定条件下能够得到控制(或者说舍入误差的增长不影响产生可靠的结果)，则该算法是数值稳定的，否则是数值不稳定的。

例 1.6　计算积分

$$I_n = e^{-1}\int_0^1 x^n e^x \, dx$$

由分部积分法可得递推公式：

$$\begin{cases} I_0 = e^{-1}\int_0^1 e^x \, dx = 1 - e^{-1} \approx 0.6321 \\ I_n = 1 - nI_{n-1} \quad (n = 1,2,\cdots) \end{cases} \tag{1.1}$$

利用上式，用四位小数计算依次得到：0.6321，0.3679，0.2642，0.2074，0.1704，0.1480，0.1120，0.2160，-0.7280，7.5520。由此看到 I_8 为负值，显然与一切 $I_n>0$ 矛盾。事实上由估计式

$$\frac{e^{-1}}{n+1} = e^{-1}(\min_{0\leqslant x\leqslant 1} e^x)\int_0^1 x^n \, dx < I_n < e^{-1}(\max_{0\leqslant x\leqslant 1} e^x)\int_0^1 x^n \, dx = \frac{1}{n+1}$$

必有 $I_7<\frac{1}{8}=0.125$，但上面算得的 $I_7=0.2160$ 已经连一位有效数字也没有了。发生这种现象的原因是，I_0 带有不超过$\frac{1}{2}\times 10^{-4}$的误差，虽然在后面的递推计算中没有新的舍入误差，但这个初始数据的误差在以后的递推中顺次乘以 $n=1,2,3,\cdots$而传播积累到 I_n 中，使得计算到 I_7 时就完全不精确了。

但如果将递推公式(1.1)改为

$$I_{n-1} = \frac{1-I_n}{n} \tag{1.2}$$

再由$\frac{e^{-1}}{10}<I_9<\frac{1}{10}$，粗略取 $I_9\approx\frac{1}{2}\left(\frac{1}{10}+\frac{e^{-1}}{10}\right)=0.0684$，按递推公式(1.2)对 $n=9,8,\cdots,1$

倒推计算，计算中的小数点后5位要四舍五入，得I_9，I_8，…，I_0依次为：0.0684，0.1035，0.1121，0.1268，0.1455，0.1709，0.2073，0.2642，0.3679，0.6321。由此可见$I_0=0.6321$全部为有效数字。这样计算的结果如此精确的原因是，I_9的误差传播到I_8时要乘以1/9，直到计算I_0时，I_9的误差已缩小为原来的$1/9!$。

我们将运算过程中舍入误差不增长的计算公式称为数值稳定的，否则称为不稳定的。如递推公式(1.2)就是数值稳定的，而式(1.1)是不稳定的递推式。

由上面的例子可见，选用数值稳定性好的计算公式，可以防止误差的递推积累，从而使计算结果更精确。

本章小结

本章主要介绍了计算方法和误差的基本概念，并且对误差的产生与防止进行了分析；同时介绍了绝对误差、相对误差、有效数字的概念及计算的方法。

用秦九韶法求多项式的值能减少运算次数，非常适合于计算机编程实现，建议熟练掌握。

习 题 1

1.1 计算方法的主要研究对象是什么？

1.2 误差的来源有哪些？

1.3 什么叫绝对误差、相对误差和有效数字？

1.4 为防止误差的危害，应遵循的原则有哪些？

1.5 指出下列各数各有几位有效数字：

6.8675，6.086 75，0.086 75，196.4730，196×10^5，0.000 196

1.6 将下列各数舍入至5位有效数字：

3.258 94，3.258 96，4.382 000，0.000 789 242

1.7 若近似数x具有n位有效数字，且表示为

$$x=\pm(a_1+a_2\times10^{-1}+\cdots+a_n\times10^{-(n-1)})\times10^m \qquad (a_1\neq0)$$

证明其相对误差限为

$$\varepsilon_r\leqslant\frac{1}{2a_1}\times10^{-(n-1)}$$

并指出近似数75.894和0.0498的相对误差限分别是多少？

1.8 设$a=1000$，取四位有效数字进行运算，求近似值x：

(1) 按算式$x=\sqrt{a+1}-\sqrt{a}$计算。

(2) 按算式$x=\dfrac{1}{\sqrt{a+1}+\sqrt{a}}$计算。

将两结果与准确值$x^*=0.015\,807\,437\cdots$比较，各有多少位有效数字。

1.9 设$f(x)=8x^5-0.4x^4+4x^3-9x+1$，用秦九韶法求$f(3)$。

第 2 章　非线性方程的数值解法

科学研究和工程设计中的许多问题常常归结为解一元函数方程 $f(x)=0$。而对于此类方程，即使有根存在，也难以求出根的表达式，更因为复杂而难以计算根的近似值；况且在实际问题中，欲求方程的根时，不一定需要得到根的准确值，只需求得满足一定精度的近似根就可以了。所以我们可以用数值方法计算非线性方程的近似根。使用数值方法求根主要分两步：

第 1 步：确定初始近似根或确定近似根存在的区间；

第 2 步：将根精确化，即将根的近似值逐步精确化使之满足实际问题的需要(达到一定的精度)。

本章将介绍几种数值计算方法用于求非线性方程的近似根。

2.1 二 分 法

二分法(或称对分法)是求方程近似解的一种最简单、最直观的方法。

设函数 $f(x)$在区间$[a,b]$上单调连续，且 $f(a)f(b)<0$(区间两端点的函数值 $f(a)$和 $f(b)$异号)，则方程 $f(x)=0$ 在(a,b)内有惟一的实根 x^*。这是微积分中的中值定理，也是二分法使用的前提条件。

2.1.1 算法原理及思想

二分法是用对分区间的方法根据分点处函数 $f(x)$值的符号，逐步将有根区间缩小，使在足够小的区间内，方程有且仅有 1 个根。

首先，为便于讨论不妨设 $f(a)<0$，$f(b)>0$，如图 2.1 所示。取区间$[a,b]$的中点 $x_0=\frac{1}{2}(a+b)$，计算函数值$f(x_0)$，若正好 $f(x_0)=0$，则说明已经找到方程的根：$x^*=\frac{1}{2}(a+b)$；否则 $f(x_0)$与 $f(a)$异号，或与 $f(b)$异号。若 $f(x_0)>0$，则 $f(x_0)$与 $f(a)$异号，设 $a_1=a$，$b_1=\frac{1}{2}(a+b)$(即只取原来区间的左半部分)；若 $f(x_0)<0$，则 $f(x_0)$与 $f(b)$异号，设 $a_1=\frac{1}{2}(a+b)$，$b_1=b$(即只取原来区间的右半部分)。这样区间$[a_1, b_1]$是方程新的有根区间，此区间被包含在旧的有根区间$[a, b]$内，且长度是原区间长度的

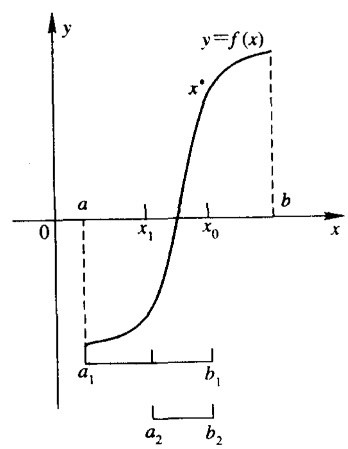

图 2.1　二分法的思想

一半，即 $b_1-a_1=\frac{1}{2}(b-a)$。再将区间 $[a_1, b_1]$ 对分，得中点 $x_1=\frac{1}{2}(a_1+b_1)$，计算函数值 $f(x_1)$，重复上述过程，则可得到长度又缩小一半的有根区间 $[a_2, b_2]$，如此反复对分下去，可得到一系列的有根区间：$[a, b]$，$[a_1, b_1]$，$[a_2, b_2]$，…，$[a_k, b_k]$，…。其中，每个区间都在前一区间内，且长度是前一区间的一半，因此区间 $[a_k, b_k]$ 的长度为

$$b_k - a_k = \frac{1}{2}(b_{k-1} - a_{k-1}) = \cdots = \frac{1}{2^k}(b - a) \tag{2.1}$$

每次对分后，如果取有根区间 $[a_k, b_k]$ 的中点 $x_k=\frac{1}{2}(a_k+b_k)$ 作为 $f(x)=0$ 的近似根，则在对分过程中将得到一系列的近似根：

$$x_0, x_1, x_2, \cdots, x_k, \cdots$$

而且这些近似根会以实际的根 x^* 为极限，即近似根的准确性越来越高。但是在实际计算中不必进行无限次的重复过程，因为数值分析的结果可以允许带有一定的误差，只要该误差在一定范围之内，即可将此时的近似根作为根。

由于

$$|x^* - x_k| \leqslant \frac{1}{2}(b_k - a_k) = b_{k+1} - a_{k+1} \tag{2.2}$$

因此对于预先给定的精度 ε，若有 $b_{k-1}-a_{k+1}<\varepsilon$，则可认为结果满足了方程 $f(x)=0$，即得到满足精度要求的近似根。

由式(2.1)、(2.2)可得误差估计式为

$$|x^* - x_k| \leqslant \frac{1}{2^{k+1}}(b - a) \tag{2.3}$$

由式(2.3)可知，可以根据预先给定的精度 ε，得到需要对分的次数 k。

例 2.1　用二分法求方程

$$f(x) = x^3 - 2x - 5 = 0$$

在区间[2, 3]内的根的近似值，并指出其误差。

解　因为 $f(2)=-1<0$，$f(3)=16>0$，故(2, 3)为有根区间。令 $a=2$，$b=3$，取中点 $x_0=(2+3)/2=2.5$，$f(2.5)=5.625>0$，则有根区间为：(2, 2.5)，所以取 $a_1=2$，$b_1=2.5$。再取中点 $x_1=(2+2.5)/2=2.25$，$f(2.25)>0$，则有根区间为：(2, 2.25)，所以取 $a_2=2$，$b_2=2.25$。如此继续，即得计算结果，列于表 2.1。

表 2.1　二分法算例

k	a_k($f(a_k)$的符号)	x_k($f(x_k)$的符号)	b_k($f(b_k)$的符号)
0	2(−)	2.5(+)	3(+)
1	2(−)	2.25(+)	2.5(+)
2	2(−)	2.125(+)	2.25(+)
3	2(−)	2.0625(−)	2.125(+)
4	2.0625(−)	2.093 75(−)	2.125(+)
5	2.093 75(−)	2.109 375(+)	2.125(+)
6	2.093 75(−)	2.101 563(+)	2.109 375(+)
7	2.093 75(−)	2.097 656(+)	2.101 563(+)