计算方法

主编 张世禄

副主编 万 俊 陈豫眉 熊 华

Computing Methods

电子科技大学出版社

UESTCP

计算方法 主编 张世禄 副主编 万 俊 陈豫眉 熊 华

计 算 方 法

主　编　张世禄
副主编　万　俊　陈豫眉　熊　华

电子科技大学出版社

内容简介

计算方法是应用数学的重要分支。本书从应用的角度出发，逐一介绍代数方程(组)、超越方程、样条函数、数值微积分和常微分方程的基本的、实用的数值计算公式。作者遵从计算方法应着重计算这一原则，详细介绍了各算式的手算以及计算机(程序)计算步骤。书中首次将计算公式和程序模块结构挂钩，将一个公式为一算法变成具有相同基本程序模块结构的所有公式是一类算法。书的第十章介绍了书中绝大部分算式的算法组成以及利用基本程序模块垒积木式编程方法对应的C程序。本书将编程序重点从怎么编向编什么方向转移。书末附有实验让学生用机器验证。本书的宗旨是(数值)计算方法和所在学科Computing真正挂钩。

全书文字简练，叙述流畅，可作各工科及理科各专业教学用书和教学参考用书。本书配备有CAI，在CAI中将利用多媒体技术介绍书中的每一个例题的手算步骤，介绍各主要定理的证明过程以及如何利用基本程序模块垒积木式编程技术，需要CAI者可向电子科技大学出版社联系。

声　明

计算方法

主　编　张世禄

副主编　万　俊　陈豫眉　熊　华

出　　版：电子科技大学出版社　(成都建设北路二段四号)　邮编：610054
责任编辑：吴艳玲
发　　行：新华书店
印　　刷：电子科技大学出版社印刷厂
开　　本：787×10[illegible]2　1/16　印张 17.0　字数 413 千字
版　　次：1999 年　月第一版
印　　次：1999 年　月第一次
书　　号：ISBN 7-81065—241—9/TP・136
印　　数：1—4000 册
定　　价：20.00 元

前　言

计算方法是 Computing 学科的九大主干课之一。计算方法是大学各工科学生、理科不少学生的必修课。由于计算方法内容广而杂，几乎涉及整个数学领域的所有问题，加之计算方法章与章之间，节与节之间变化梯度有时相当大，因而被认为是既难教又难学的一门课程。

计算方法虽是一门古老的学科，但今天的计算方法和古代的计算方法已大大不同。古代的计算方法是为手算服务的，今天的计算方法却主要是为计算机的计算服务的，也就是说今天计算方法是程序设计的工具。不讲程序设计势必和计算方法的任务和宗旨相左，但是若按传统的方法介绍一个公式立即讲授该公式所对应的程序又会使这门本来就难教难学的功课更难教更难学。

受张景中院士“教育数学”理论的启示，我们将计算方法和程序设计的内容再加工，总结出不同的计算公式可以归结为同一算法，同一算法具有相同的基本程序模块结构。计算公式虽然千千万万，但是它们中的绝大多数（或者现有的计算公式）都可归结为六个基本算法的集合。这六个基本算法是：

一、连乘积计算法；

二、累加和计算法；

三、递推算法；

四、迭代算法；

五、尝试法；

六、递归算法。

计算方法中不讲递归算法。

虽然计算方法不专门介绍尝试法，但不少计算公式离不开尝试法，例如主元素消元法中选主元，计算 $\|\cdot\|_\infty$ 等都属尝试法。因此本书附带介绍尝试法。

本书将按基本算法和基本程序模块结构介绍手算过程及计算机（程序）计算过程，尽量使教和学从既要求知道干什么，又要求知道怎么干，向多知道干什么，少知道怎么干转移，尽量使计算过程像套公式、垒积木一样方便。

为了使学生更易掌握计算过程，对书中所介绍的所有公式都给出了手算算例，在第十章中再按基本算法介绍其程序，而其余的则让学生在数学实验中自己编写程序。

本书的作者全系四川师范学院计算机辅助教学硕士方向的导师和成员。为了教学的需要，本书配置了 CAI。在 CAI 中将演示：

1. 算例的手算过程；

2. 主要公式的推导和证明过程；

3. 程序编制和计算过程。

书中绪论、第一章、第六章、第十章及数学实验由张世禄先生编写；第二至五章由万俊先生编写；第七至九章由陈豫眉女士编写。书中所有程序都经熊华女士验证，书中所有手算结果也由熊华女士用程序验证核实。

教学改革和教材改革是21世纪高校教师的主要使命之一。作者在这方面仅作了初步尝试，加上成书仓促，作者水平有限，书中定有不少错误和不当，还请用过此书的师生斧正。

作　者

1999年5月

目　录

绪　论

高等数学或者其他数学所说的数学问题实际上是理想中的数学问题，理想中的数学问题虽然也是从实际中来，也能解决实际问题．但是在科学技术领域能把一具体问题用一数学公式描述的不多，能得到理论解的更少，即使得到理论解也和实际问题的解有差异．

要将科技领域中、现实生活中的具体问题归结成数学问题一般要经过两个过程：

1．建立物理模型；

2．建立数学模型．

要建立物理模型和数学模型常常要作若干次假设．每作一次假设，多多少少总要带来一些差异，假设愈多，可能造成的差异会愈大，但只要假设合理，其模型都是可取的．不少人头脑里有一个错误概念，认为物理模型、数学模型总是愈接近实际问题愈好．若能和实际问题完全一致则最好！

实际上，物理模型、数学模型的好坏取决于两点：

1．能否能抓住问题的本质；

2．是否能尽快得到满意解．

能否抓住问题的本质取决于模型建立者的知识和经验，这不属于本书研究课题．

尽快和满意都具有相对性，现举两例加以说明．

例 1　计算一圆桌的表面积．

本例不需要建立物理模型，从题目就可看出数学模型也已经建好，那就是桌面是一个直径为 D 的圆，其计算公式为：

$$S = \pi D^2/4$$

我们知道世界上不存在几何中的圆，任何高级木匠作的圆桌都不可能是真正的圆桌．但是只要能称得上木匠，他所作的圆桌都可认为是圆桌．用上面公式算得的面积都是满意的，显然要找一公式一丝不差地描述桌面边缘所围成的曲线既是不可能的也是不必要的，即使找到了，计算也将是很麻烦的！

例 2　科学院成都某研究所给某钢铁公司设计了一炼钢控制系统．该系统可计算并输出炼钢过程炼钢炉内各点钢水温度．在建立物理模型时作了如下假设：

1．炉体材料绝对均匀，性能永远一致；

2．绝对绝热；

3．满足热传导方程．

建立数学模型时作了如下假设：

1．炉体是标准直圆柱；

2．炉体无限长．

绝对均匀、性能永远一致的材料是找不到的，绝对绝热的材料世上也没有．几何中直圆柱炉任何人也制作不出，无限长更是办不到．这一切都和实际问题有明显的差异，但是用上述模型所得的计算结果虽和钢水的实际温度有差别，但差异在允许范围之内．计算结果是

令人满意的．作上述假设的目的在于将一个边界复杂的三维热传导方程变成边界条件简单的一维热传导方程，及时算出温度场中各点温度；相反若选用更接近炼钢过程的三维或二维模型，则不能及时算出指定各时刻指定点钢水温度，达不到自动控制目的．

科技领域中的不少数据都是通过仪表测得的，这些数据有以下三个特点：

1．和真实值多或少少有一些差异；

2．不可能连续；

3．无法用一个初等函数直接描述．

显然利用这些数据要得到数学中理论解是不可能的．通过这些数据建立的数学模型绝对不是“理想”中的数学模型．

综合上述可得出结论：建立同实际问题完全一致的数学模型不必要也不可能．现在假设和实际问题完全吻合的数学模型已经建立好．在此前提下能否就一定得到理论解——精确解呢？答案是否定的．

例如已知一个角，要想得到它的 1/3 角．对于如此简单的数学问题就只能得到近似解．至于常微分方程、偏微分方程、超越方程、代数方程、积分……能得到理论解的只有沧海一粟．

上面所说的所谓问题都是数学分析领域内所无法解决的问题．上述绝大多数问题又是科技领域、工农业生产领域所必须解决的问题．

计算方法也称之为计算数学，其功能就是计算近似解．近似解与真实解之间的差称为误差，误差在指定范围之内其解称为满意解，超出指定范围解就失去了意义．

计算方法所涉及的范围很广，它几乎包含了现代各种数学，不过可粗略地分成两大部分：

1．数值逼近；

所谓逼近是指用简单的初等函数代替原来的形式复杂的函数或者代替一时还找不到解析算式的函数．

2．数值代数

不少计算数学问题最后要归结为解一代数方程(组)，因而数值代数是计算数学中最主要内容．

计算方法是一门古老的数学，我国古代出了不少这方面的学者．祖冲之、秦九韶、杨辉等在计算方法方面都给人类作出了相当大的贡献．

计算机问世后，计算方法这门学科发展迅猛，计算方法才真正有了用途．计算方法中的计算都直接和计算机挂钩，计算机科学的学科名也由计算机(Computer)改成了计算(Computing)．这里的计算是指人们编出程序由计算机计算．因此编写程序也属计算方法的范畴，从某种意义上讲，会编程序才是真正掌握了计算方法．

计算方法难学，原因是这门学科内容多而杂，几乎包罗了应用数学中所有数学问题．计算方法所涉及的数学公式多，单就这本教材而言就将介绍数十个公式，要记住这些数学公式已够难了，若用传统方法对一个公式设计一个程序则更会增加这门学科难度．这也是不少计算方法教材只能口头强调程序设计的重要性而难以具体介绍各公式所对应的程序的设计的原因．

我们知道，不同的物理问题可能归结为同一数学问题，同一数学问题可用同一数学公式

描述．我们发现，不同的数学问题可以归结为同一算法，同一算法具有相同的基本程序模块．这里的算法和传统的算法有着本质区别．传统的一个算法对应一个公式，一个公式对应一个过程．第三代计算机语言正好适应传统算法．按传统算法编程序，不仅要知道干什么，更重要地是要知道怎么干．新算法将算法和程序模块结构紧密地结合在一起，不管计算公式如何，只要它所对应的基本程序模块结构一样就属同一算法，也就是说若干公式对应一个算法，同一算法对应相同的基本程序模块结构．例如梯形积分公式、辛普生积分公式、向量内积计算公式、函数内积计算公式、矩阵乘积计算公式、标准分（z 分）计算公式，虽然各自的数学含义大不相同，但它们的主体都是计算累加和，计算累加和就是新算法中的一个算法．所有计算累加和的基本程序模块都相同．

现在已为人知的数值计算公式千千万万，随着时代的进步，科学的发展，还将有不少新的先进计算公式出现．公式虽多，但它们中绝大多数（99%）都可归结于本书中所介绍的六个基本算法．或由其中几个算法组成，这六个基本算法是：

1. 计算连乘积；
2. 计算累加和；
3. 递推算法；
4. 迭代算法；
5. 递归算法；
6. 尝试法．

计算方法不介绍递归算法．

当熟悉了这几个算法的基本程序模块后，那么编写程序之前，只要知道干什么，所用计算公式属于何种算法，则编程序问题，即怎么编的问题就变得和套公式一样简单了．

手算是熟悉计算公式，洞察公式归属的算法及计算过程的重要一环，手算的例题虽然都较为简单，但是它的计算过程和步骤与计算机按程序计算的过程和步骤一致．利用计算机辅助教学软件和多媒体技术可将手算和程序计算过程形象生动地演示出来．不必要每个人都会设计辅助教学软件，但是学会使用辅助教学软件和多媒体技术对学习计算方法以及其他学科都大有帮助．

21 世纪即将来临，我们已处于知识经济时代，计算机知识是知识经济的知识核心，各种知识无不和计算机知识有关，计算方法是科技计算、计算机管理的重要工具，掌握这一工具，迎接知识经济对我们的挑战吧！

第一章 误　　差

误差是指实际值与计算值之间的差．误差有绝对误差和相对误差之分．当实际值在 0 附近时多用绝对误差，离 0 较远时多用相对误差．

由于实际值通常不知道，故误差是一估计值，无论是绝对误差还是相对误差，人们并不关心它到底是多大，只关心它是否落在指定范围之内．误差在指定范围之内其解是满意解，否则所得到的解无意义．因此软件人员的任务就是寻找合适的数值计算公式，使其误差落在指定范围之内．

§1-1 误差来源

误差主要来源于：

1. 模型误差

建立物理模型和数学模型都要作若干个假设，以求简化问题，使问题理想化，从而得出易于计算的算式．简化了的问题自然和原问题有差异．绪论中例 2 在建立物理模型时将材料认为是均匀的、性能不变的、绝对绝热的无疑都会产生模型误差．例 1 认为桌面边界是几何中的圆(实际上把桌面认为是平面也和实际问题有差异)．它们同属模型误差．

选择适宜的物理模型、数学模型，使能得到满意解不属于计算方法的内容，计算方法不考虑模型误差．

2. 测试误差

绝大多数程序都有数据输入，输入数据中的不少数据是用仪器仪表测试得出的，即使最精密的仪器也无法测得准确值．准确值和测试值之间的差称为测试误差．

绪论中例 1 里圆桌的直径，例 2 中炉体内径都须用量具测出，都只能使用测量值．是测量值就一定有误差．

3. 舍入误差

计算机中的数都存放在它的存储器中，计算机中的数有两大特征：

1)间断性；

2)有界．

实数是连续的，但计算机中的实型数是不连续的，原因是存储器的空间是有限的，无论用多少字节存放一个数，它所能容纳的数的位数总是有限的．对于定点数讲，绝对值比它所能表示的小数点后最后一位数还小的部分就无法表示，这一部分只能四舍五入，或者干脆全部舍去．四舍五入或者全部舍去均会引起误差，这种误差称为舍入误差．

对于浮点数也一样．

例如，设某计算机在某一确定软件支持下能表示的定点数绝对值不能小于 10^{-7}，舍取方式为四舍五入，则

$$0.1 + 10^{-8} = 0.1(0000000)$$

$$0.1 + 5 \times 10^{-8} = 0.1000001$$

这里我们用了=号，因为我们看见的就是 0.1 和 0.1000001.

计算方法中有一类问题称之为条件问题，条件问题是一个算法(公式)对初始数据或者中间某些数据微小摄动对计算结果影响的敏感性的问题．舍入误差、测试误差都属初始数据的摄动．研究坏条件问题的计算方法是十分重要的课题．有的时候，一些问题的条件并不坏，但由于计算顺序不恰当，初始数据的微小摄动或舍入误差在计算过程中不断被放大，轻者使计算结果精度大大降低，重者使计算失去意义．

例 1 计算 $I_n = \int_0^1 \frac{x^n}{5+x} dx, \ n = 0,1,\cdots,20$ (1-1-1)

解 当 $n=0$ 时

$$I_0 = \int_0^1 \frac{x^0}{5+x} dx = \ln(5+x)\Big|_0^1 = \ln 6 - \ln 5$$

$$I_n + 5I_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5} dx = \int_0^1 x^{n-1} dx = \frac{1}{n}$$

由此可得出递推计算公式：

$$\begin{cases} I_n = -5I_{n-1} + \dfrac{1}{n} \\ I_0 = \ln 6/5 \end{cases} \tag{1-1-2}$$

对于式(1-1-2)很容易编出程序计算 $I_0, I_1, \cdots, I_{20}$ 的值，计算结果见表 1-1.

表 1-1

j	$I[j]$	j	$I[j]$	j	$I[j]$	j	$I[j]$
1	0.08839	6	0.024428	11	−0.307181	16	1003.922729
2	0.05803	7	0.020719	12	1.619237	17	−5019.554688
3	0.04313	8	0.021407	13	−8.019263	18	25097.828125
4	0.03431	9	0.004076	14	40.167744	19	−125489.085938
5	0.02844	10	0.079618	15	−200.772049	20	627445.500000

容易看出，积分

$$I_n = \int_0^1 \frac{x^n}{5+x} dx$$

具有以下几个特点：

1) $0 < I_n < 1$

2) $I_n < I_{n-1}$

3) $n \to \infty, I_n \to 0$

表 1-1 中后面几个积分值符号交替变换且绝对值愈来愈大，大大超过 1，这显然和理论分析矛盾．原因何在?

假设计算机的舍入误差为 ε，令 $I_0 = \ln 6/5$ 为理论值，$\hat{I}_0$ 为计算值，即 $I_0 = \hat{I}_0 + \varepsilon$

由式(1-1-2)有

$$I_1 = -5I_0 + 1$$
$$\hat{I}_1 = -5\hat{I}_0 + 1$$
$$I_1 - \hat{I}_1 = -5(I_0 - \hat{I}_0) = -5\varepsilon$$
$$\cdots$$
$$I_n - \hat{I}_n = -5(I_{n-1} - \hat{I}_{n-1}) = (-1)^n 5^n \varepsilon \tag{1-1-3}$$

显然对式(1-1-3)有$\lim\limits_{n\to\infty}|(I_n-\hat{I}_n)|=\infty$,这就是引起表 1-1 中后面的积分值符号交替变换且绝对值愈来愈大并远远超过 1 的原因.

现考虑另外一个计算方案.

因 $I_n<I_{n-1}$, $I_n>0$

故 $5I_{n-1}<I_n+5I_{n-1}<6I_{n-1}$

考虑到 $I_n+5I_{n-1}=\frac{1}{n}$,由此有

$$\frac{1}{6\times 21} < I_{20} < \frac{1}{5\times 21}$$

取 $I_{20}=\frac{1}{2}(\frac{1}{6\times21}+\frac{1}{5\times21})$

利用递推公式

$$\begin{cases} I_{n-1} = \dfrac{I_n}{5} + \dfrac{1}{5n}, n = 20,19,\cdots,0 \\ I_{20} = \dfrac{1}{2}(\dfrac{1}{6\times 21} + \dfrac{1}{5\times 21}) \end{cases} \tag{1-1-4}$$

用式(1-1-4)计算结果见表 1-2.

表 1-2

j	$I[j]$	j	$I[j]$	j	$I[j]$	j	$I[j]$
1	0.088392	6	0.024325	11	0.014071	16	0.009898
2	0.058039	7	0.021233	12	0.012977	17	0.009336
3	0.043139	8	0.018838	13	0.012040	18	0.008876
4	0.034306	9	0.016926	14	0.011299	19	0.008354
5	0.028468	10	0.015368	15	0.010520	20	0.008230

设 I_{20}的计算误差为 ε,则 I_{19}的计算误差仅为 $\varepsilon/5,\cdots,I_0$ 的计算误差缩到 $\varepsilon/5^{20}$, $5^{-20}\varepsilon$ 已远远小于单精度数的舍入误差,所以表 1-2 中计算值 I_0 和直接计算 ln6/5(表中未列出)的值完全一样.

不同算法对初始数据的误差(或计算过程中某一步的舍入误差)的传播一般不同,一个算法,经过指定次数计算后,若仍能将初始数据的误差的影响限于一定范围之内,这个算法的稳定性就好,反之则稳定性差.

4. 截断误差

在不考虑测试误差和舍入误差的前提下,理论值和某种算法计算值之差称为截断误

差．截断误差是算法本身的误差，截断误差也称算法的先天误差．

例 2 利用公式

$$e^{-x}=\sum_{i=0}^{\infty}(-x)^{i}\frac{1}{i!}=1-x+\frac{1}{2}x^{2}-\frac{1}{6}x^{3}+\cdots \tag{1-1-5}$$

取部分和

$$E(x)=1-x+\frac{1}{2}x^{2}-\frac{1}{6}x^{3}+\frac{1}{24}x^{4} \tag{1-1-6}$$

作为 e^{-x} 的近似值，此时

$$e^{-x}-E(x)=\frac{1}{120}e^{-\xi}x^{5}$$

就是该算法的截断误差．由此也可看出截断二字的含义，对于数值逼近的每一种算法都必须考虑并记住它的截断误差的大小或者量级．

§1-2 误差表示法

1. 误差表示法

定义 1 设 x 为某个数的准确值，x^* 为计算值或测试值，记

$$\varepsilon(x)=x-x^{*} \tag{1-2-1}$$

为近似数 x^* 的绝对误差，简称误差．

由于 x 本身，即准确值本身并不知道，因此 $\varepsilon(x)$ 值通常不可能也不必要算出．人们常常不关心 $\varepsilon(x)$ 值的本身，只关心它不超过多少，也就是说只关心它的限．

定义 2 若要求 $|x-x^*|\leqslant\eta$，则称 η 为绝对误差限．

只要 x^* 落在 $[x-\eta,x+\eta]$ 内，则认为所得到的 x^* 是满意的，否则不符合要求．

对于测试误差讲，绝对误差的大小和仪器、仪表的精度直接相关，仪器、仪表的最小量度不能大于绝对误差限．

对于截断误差讲，它直接关联着算法的精度，以上一节例 2 为例，若 $0\leqslant x\leqslant 1$，当给定的绝对误差限 $\eta=10^{-5}$，则上一节中式(1-1-6)将达不到要求；若 $\eta=10^{-2}$，式(1-1-6)足够用，大家可以用手算验证．

绝对误差有时不足以刻划近似数的精度．例如测量圆桌直径时发生了 0.1 厘米的误差和测量手表表轴时发生了 0.1 厘米的误差大不一样．对于圆桌讲，0.1 厘米误差甚至更大的误差都是允许的，而手表的轴若有 0.1 厘米的误差则表轴将报废了．要决定一个近似值的精确程度，除了绝对误差之外，还必须考虑数本身的大小，这就要引进相对误差的概念．

定义 3 若 $\varepsilon(x)=x-x^*$ 为计算数的 x 的绝对误差，则称

$$\varepsilon_{r}(x)=(x-x^{*})/x \tag{1-2-2}$$

为近似数 x^* 的相对误差．有时也用 $\frac{x-x^*}{x^*}$ 表示相对误差．

同绝对误差一样，因 x 未知，$\varepsilon_r(x)$ 也仅是一估计值．人们也并不关心 $\varepsilon_r(x)$ 是多少，也只关心它是否落在指定范围之内．

定义 4 若规定 $|\varepsilon_r(x)=|(x-x^*)|/|x|\leqslant\delta$，则称 δ 为相对误差限．

2. 近似数的算术运算

现在讨论进行加、减、乘、除时，原始数据的相对误差和计算结果的相对误差之间的关系．

1)乘法和除法

设正数 x 的近似值 $x^*=x+\Delta x$，绝对误差 Δx 近似地等于 x 的微分，即 $\Delta x\approx \mathrm{d}x$，此时

$$\varepsilon_r(x)=\frac{\Delta x}{x}\approx\frac{\mathrm{d}x}{x}=\mathrm{d}\ln x$$

当 $x<0$ 时，可设 $y=-x$，则

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

总之，一个数的相对误差可用该数或相反数的自然对数的微分近似表示．由此有

$$\varepsilon_r(x_1x_2)\approx\mathrm{d}\ln x_1x_2=\mathrm{d}\ln x_1+\mathrm{d}\ln x_2\approx\varepsilon_r(x_1)+\varepsilon_r(x_2) \tag{1-2-3}$$

同样可得

$$\varepsilon_r(x_1/x_2)\approx\varepsilon_r(x_1)-\varepsilon_r(x_2) \tag{1-2-4}$$

式(1-2-3)、式(1-2-4)尚可推广，能表示更多个近似数的乘积和商的相对误差．

虽然在实际计算时式(1-1-3)和式(1-1-4)的结果有可能比单独一个数的相对误差还小，甚至可能几者最后抵消，但估计误差时，尤其是看误差是否超过相对误差限时，总是从最坏处着想，取其各相对误差绝对值之和！

2)加法和减法

设 x_1 和 x_2 的近似值分别为 $x_1+\Delta x_1$ 和 $x_1+\Delta x_2$，则

$$\varepsilon_r(x_1+x_2)=\frac{\Delta(x_1+x_2)}{x_1+x_2}=\frac{\Delta x_1+\Delta x_2}{x_1+x_2}=$$

$$\frac{x_1}{x_1+x_2}\cdot\frac{\Delta x_1}{x_1}+\frac{x_2}{x_1+x_2}\cdot\frac{\Delta x_2}{x_2}=\frac{x_1}{x_1+x_2}\cdot\varepsilon_r(x_1)+\frac{x_2}{x_1+x_2}\cdot\varepsilon_r(x_2)$$

若 x_1 和 x_2 同号，则上式右端 $\varepsilon_r(x_1)$ 和 $\varepsilon_r(x_2)$ 前的系数的绝对值都小于 1，且其和为 1. 此时

$$\begin{aligned}|\varepsilon_r(x_1+x_2)|&\leqslant\frac{x_1}{x_1+x_2}\max(|\varepsilon_r(x_1)|,|\varepsilon_r(x_2)|)+\frac{x_2}{x_1+x_2}\max(|\varepsilon_r(x_1)|,|\varepsilon_r(x_2)|)\\&\leqslant\max(|\varepsilon_r(x_1)|,|\varepsilon_r(x_2)|)\end{aligned} \tag{1-2-5}$$

当 $|x_1|\gg|x_2|$ 时，

$$\varepsilon_r(x_1+x_2)\approx\varepsilon_r(x_1) \tag{1-2-6}$$

当 x_1 和 x_2 异号，如果此时 x_1 和 $-x_2$ 相当接近，由式(1-2-5)可看出，虽然 $\varepsilon_r(x_1)$ 和 $\varepsilon_r(x_2)$ 的绝对值都不大，但 $\varepsilon_r(x_1+x_2)$ 的绝对值可能很大，甚至难以估计，而这类问题就是上一节所说的坏条件问题，此时计算结果失去意义．

3. 数的近似表示

对于近似值，我们还希望知道它的准确程度，为此我们再引进有效数字的概念．

定义 5　若近似值 x^* 的误差不超过某位数字的半个单位，而从该位数字到 x^* 最左边的那个非零数字共有 n 位，那么这 n 个数字都是有效数字，或称该数有 n 位有效数字(如图 1-1)．

例如用 $x^*=2.718$ 表示 e，则 x^* 有四位有效数字．它的绝对误差不超过 0.0001/2.

引入有效数字后要明确以下两点：

1)近似值50.3和50.300不一样，前者只有三位有效数字，后者有五位有效数字，后者的绝对误差(绝对值)小于等于前者.

$$x^* = \overbrace{xxx\cdots x}^{n}$$

误差不超过该数半个单位

自左向右看第一个非零数字

图 1-1

2)对于任意一个准确值 x，用四舍五入法得到的具有 n 位有效数字的近似值 x^*，总可以写成

$$x^* = \pm 10^m(a_1 \times 10^0 + a_2 \times 10^{-1} + \cdots + a_n \times 10^{-(n-1)}) \tag{1-2-7}$$

式(1-2-7)中，$a_1, a_2, \cdots, a_n$ 为 0～9 中某一数字($a_1 \neq 0$)，按有效数字的定义有

$$|x - x^*| \leqslant \frac{1}{2} \times 10^m \times 10^{-(n-1)} = \frac{1}{2} \times 10^{m-n+1} \tag{1-2-8}$$

4. 有效数字与相对误差之间的联系

对于准确值 x 的近似值 x^* 而言，数的有效位越多，绝对误差就越小，用式(1-2-8)则可看出它的绝对误差限；反之规定了绝对误差限后就可在确定位上作四舍五入. 下面是有效数字和相对误差间的关系.

设用式(1-2-7)表示的近似数具有 n 位有效数字，显然

$$a_1 \times 10^m \leqslant |x| \leqslant (a_1 + 1) \times 10^m$$

其相对误差为

$$|\varepsilon_r(x)| \leqslant \left|\frac{x - x^*}{x^*}\right| \leqslant \frac{\frac{1}{2} \times 10^{m-n+1}}{a_1 \times 10^m} = \frac{1}{2a_1} \times 10^{-(n-1)} \tag{1-2-9}$$

§1-3 算法选择

《计算方法》为程序设计提供了解决数学问题的计算公式. 解决同一问题常常可用不同的计算公式，选用不同的计算公式所得到的结果可能大不一样，因此存在公式选择问题.

前面已经提到，对于同一公式，计算顺序不同，算法的稳定性可能不同. 下面还会看到计算顺序不同，计算量也可能大大不同，因此，设计程序前，有时还须对公式作恒等变换，所以本节用算法选择而不用公式选择.

算法选择的原则是：

1. 正确性

正确性的含义是截断误差不超过用户指定的误差限；算法稳定；不溢出；若选用迭代法则必须考虑收敛(后面介绍).

例 1 方程

$$x^2 + 2px - q = 0$$
$$p = -0.5 \times 10^5$$
$$q = 1 \times 10^{-8}$$