· 高等学校计算机基础教育教材精选 ·

数值方法与计算机实现
（第2版）

徐士良 编著

清华大学出版社

·高等学校计算机基础教育教材精选·

数值方法与计算机实现
（第2版）

徐士良 编著

清华大学出版社
北京

内容简介

本书以数值分析为基础，介绍算法设计与分析，并具体给出了工程上常用的、行之有效的数值型算法。

全书共分9章。主要内容包括误差与运算误差、线性代数方程组与矩阵运算、矩阵特征值、非线性方程、代数插值、函数逼近与曲线拟合、数值积分与数值微分、常微分方程数值解、连分式及其新计算法。附录中给出了各章习题的参考答案。

本书可以作为高等理工科院校非数学专业的《数值分析》或《计算方法》等课程的教材，也可供广大工程技术人员参考。

图书在版编目（CIP）数据

数值方法与计算机实现/徐士良编著. —2版. —北京：清华大学出版社，2010.2
（高等学校计算机基础教育教材精选）
ISBN 978-7-302-21701-5

Ⅰ.①数…　Ⅱ.①徐…　Ⅲ.①电子计算机—数值计算—高等学校—教材　Ⅳ.①TP301.6

中国版本图书馆CIP数据核字(2009)第243041号

责任编辑：焦　虹　徐跃进
责任校对：时翠兰
责任印制：孟凡玉

出版发行：清华大学出版社　　**地　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印　装　者：北京鑫海金澳胶印有限公司
经　销：全国新华书店
开　本：185×260　　**印　张**：26.75　　**字　数**：608千字
版　次：2010年2月第2版　　**印　次**：2010年2月第1次印刷
印　数：1～3000
定　价：38.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177转3103　　产品编号：035658-01

出版说明

在教育部关于高等学校计算机基础教育三层次方案的指导下，我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践，全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验，取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快，目前我国的高等教育事业正面临着新的发展机遇，但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要，进一步推动我国高等学校计算机基础教育事业的发展，我们在全国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀的教学成果，编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次，包括面向各高校开设的计算机必修课、选修课，以及与各类专业相结合的计算机课程。

为了保证出版质量，同时更好地适应教学需求，本套教材将采取开放的体系和滚动出版的方式（即成熟一本、出版一本，并保持不断更新），坚持宁缺毋滥的原则，力求反映我国高等学校计算机基础教育的最新成果，使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作，在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度，以适应高校计算机基础教育课程层次化、多样化的趋势，从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来，把自己的教学成果与全国的同行们分享；同时也欢迎广大读者对本套教材提出宝贵意见，以便我们改进工作，为读者提供更好的服务。

我们的电子邮件地址是 jiaoh@tup. tsinghua. edu. cn。联系人：焦虹。

清华大学出版社

第 2 版前言

本书是从应用的角度来描述数值方法,并且直接用计算机实现这些方法,这不仅对于学生,而且对于广大工程技术人员来说,都是很有帮助的。本版在前面各版的基础上做了如下几方面的修订。

(1) 删除了一些内容。如自适应算法、连分式的性质等内容。

(2) 增加了一些内容。如病态系统、非线性方程求根的试位法、多变量线性拟合、数值微分、常微分方程边值问题等内容。

(3) 对章节做了重新编排。如将原来第 1、第 2 章的内容做删改后合并为一章,对原来的第 3 章与第 4 章做了重新安排。

(4) 在本书附录中给出了各章习题的参考答案。

书中所有算法均用 C 语言描述,并通过实际调试。

阅读本书只需要具备微积分与线性代数方面的基础知识。当然,还需要熟悉 C 语言方面的知识。

本书通俗易懂,例题和习题丰富。本书可以作为高等理工科院校非数学专业的数值分析或计算方法等课程的教材,也可作为广大工程技术人员的自学教材与参考书。本书所有源程序可以在清华大学出版社网站上下载,网址为 http://www.tup.com.cn。

限于水平,书中难免会有错误和不当之处,恳请读者批评指正。

徐士良

于清华大学

第 1 版前言

本书是在清华大学出版社出版的《计算机常用算法(第 2 版)》(前两版共发行 9 万册)基础上做了修改和补充,去掉了多项式与非数值问题的常用算法等部分,并对章节进行了重新编排,并改名为《数值方法与计算机实现》。其内容是以数值分析为基础,以实际应用为目的,以计算机作为计算工具,对工程中常见的数值计算问题建立行之有效的算法。为了使算法行之有效,本书的一开始就重视算法。本书主要强调问题的分析和算法的设计,通过例题说明方法的本质,忽略了许多数学上的烦琐证明过程。

书中所有算法均用 C 语言描述,并通过实际调试。

阅读本书只需要具备微积分与线性代数方面的基础知识。当然,还需要熟悉 C 语言方面的知识。

本书通俗易懂,例题和习题丰富。本书可以作为高等理工科院校非数学专业的数值分析或计算方法等课程的教材,也可作为广大工程技术人员的自学教材与参考书。本书所有源程序可以在清华大学出版社网站上下载,网址为 http://www.tup.com.cn。

限于水平,书中难免会有错误和不当之处,恳请读者批评指正。

徐士良

于清华大学

目录

第1章 绪论

由于计算机技术的飞速发展，现代计算机作为数据处理的工具，使得许多复杂的问题(特别是数值计算问题)能够得到较佳的解答。但是，一个数学问题，乃至一个具体的计算公式，是否一定能够在计算机上实现，这是以计算机作为计算工具所面临的新问题。计算机算法实际上就是研究如何用计算机这个工具来处理实际问题。计算机算法一般简称为算法。

概括地说，所谓算法是指解题方案的准确而完整的描述。算法强调的是解决实际问题时计算机的执行过程，它与静态的计算公式是有区别的。

本章首先讨论数值型算法的一些主要特点；然后介绍数值计算过程中特别需要注意的误差与运算误差问题，并以三项递推关系为例说明数值型算法的稳定性问题；最后介绍工程中常用的几个正交多项式，这些正交多项式在后面的有些章节中要用到。

1.1 数值型算法的特点

算法的执行总是与特定的计算工具有关，而每一种计算工具的有效数字位数总是有限的。在用计算机作数值计算时，计算机系统分配给一个数据的存储空间也是有限的。一般来说，一个实数无法转换成与之等值的有限位的二进制数，其有限位以后的数字将被舍去。下面的例子说明了这个问题。

例 1.1 下列 C 程序的功能是将 10 个实型数 0.1 进行累加，然后将累加结果输出。

```
#include <stdio.h>
main()
{ int k;                               /* 定义整型变量 k */
  double x,z;                          /* 定义双精度实型变量 x 与 z */
  z= 1.0;                              /* 实数 1.0 赋给变量 z */
  x= 0.0;
  for (k=0; k<10; k++)x=x+0.1;         /* 10个 0.1 累加到变量 x 中 */
  printf("z=%20.17f\n",z);             /* 输出变量 z 的值 */
  printf("x=%20.17f\n",x);             /* 输出变量 x 的值 */
}
```

运行这个程序后，输出的结果如下：

```
z=1.0000000000000000
x=0.99999999999999989
```

由这个运行结果可以看出，10 个实型数 0.1 累加后并不等于 1.0。

由上述例子可以看出，实数 0.1 确实无法精确地转换成计算机中的二进制数据，即实型常量 0.1 在计算机中只能近似表示。因此，对于一般的数值计算问题，在实际计算过程中，所有参与运算的数通常都是近似的。

作为数值型算法，除了具有一般算法的基本特点外，还具有以下几个基本特点。

1. 理论上的精确运算与实际运算之间存在着很大差异

数学本身是严密的，在进行数学推导过程中，一般都要用到一些运算规则、恒等变换公式等，而变换前后的式子是互相等价的。但在实际运算时，数学上完全等价的式子，其运算结果的差异会很大。

下面举一个例子来说明。

例 1.2 某计算工具具有四位有效数字(如四位数学用表)，现要计算当 $x=1000$ 时的下列函数值：

$$f(x)=\sqrt{x+1}-\sqrt{x}$$

解：直接将 $x=1000$ 代入函数表达式，其计算结果如下：

$$f(1000)=\sqrt{1000+1}-\sqrt{1000}=\sqrt{1001}-\sqrt{1000}=31.64-31.62=0.02$$

但如果将该函数表达式作如下恒等变换：

$$f(x)=\sqrt{x+1}-\sqrt{x}=\frac{1}{\sqrt{x+1}+\sqrt{x}}$$

然后将 $x=1000$ 代入，其计算结果如下：

$$f(1000)=\frac{1}{\sqrt{1000+1}+\sqrt{1000}}=\frac{1}{\sqrt{1001}+\sqrt{1000}}=\frac{1}{31.64+31.62}=0.01581$$

由上述的计算可知，函数表达式经过恒等变换后，其计算结果有很大的差异。实际上，通过分析，后者的计算结果要比前者的计算结果更准确，因为在前者的计算中，两个相近的近似数相减，严重丢失了有效数字。

这样的例子在初等函数的计算中是很常见的。在数学上两个恒等的式子，但在实际计算时可以得到不同的结果，这主要是受计算机有效数字位数的限制所造成的。例如，对于数学上的恒等式 $10^{12}+1+(-10^{12})\equiv10^{12}+(-10^{12})+1$，如果用具有 7 位有效数字的计算工具(如各种程序设计语言中的单精度运算)，分别按恒等式两边的顺序进行计算，其结果也是不同的(前者的计算结果为 0，而后者的计算结果为 1)。

利用数值型算法的这个特点，可以将一个运算式子通过恒等变换变成与其等价的另一个运算式子，从而提高运算结果的准确程度。

2. 理论上的解题方案与实际能用性之间存在着很大差异

一个数值计算问题通常有多种求解方案；但并不是每一种求解方案都能实际可行，一

种求解方案也并不是在所有情况下都能使用。下面举例说明这种情况。

例 1.3 求一元二次方程

$$x^2-(10^{12}+1)x+10^{12}=0$$

的两个实根。如果用通常的求根公式，并且假设计算工具具有 7 位有效数字，则计算过程如下：

$$\begin{aligned} x_{1,2} &= \frac{(10^{12}+1)\pm\sqrt{(10^{12}+1)^2-4\times 1\times 10^{12}}}{2\times 1} \\ &= \frac{10^{12}\pm\sqrt{10^{24}-4\times 10^{12}}}{2} \\ &= \frac{10^{12}\pm 10^{12}}{2} = \begin{cases} 10^{12} \\ 0 \end{cases} \end{aligned}$$

最后的计算结果为

$$x_1=10^{12},\quad x_2=0$$

经检验发现，$x_1=10^{12}$ 满足原方程，是方程的根，$x_2=0$ 不满足原方程。

这个例子表明，理论上的解题方案，在有些情况下不一定能用。实际上，一元二次方程

$$Ax^2+Bx+C=0$$

的求根公式

$$x_{1,2}=\frac{-B\pm\sqrt{B^2-4AC}}{2A}$$

对于一般的系数 A、B、C 来说可以使用，但如果对于某些系数 A、B、C 使求根公式分子中的两项$(-B)$与$\sqrt{B^2-4AC}$很接近时，计算结果就会严重丢失有效数字，在这种情况下就不能简单地使用求根公式了。

一般来说，对于一元二次方程

$$Ax^2+Bx+C=0$$

可以用以下方法求出两个实根：利用求根公式先计算一个实根（使求根公式分子中的两项为同号相加，从而避免两个同号数相减的情况），然后利用韦达定理计算另一个实根。即

$$\begin{cases} x_1=\dfrac{-B-\operatorname{sgn}(B)\sqrt{B^2-4AC}}{2A} \\ x_2=\dfrac{C}{Ax_1} \end{cases}$$

其中，$\operatorname{sgn}(B)$为 B 的符号函数，即

$$\operatorname{sgn}(B)=\begin{cases} +1, & B\geqslant 0 \\ -1, & B<0 \end{cases}$$

由这个例子可以看出，有些数学上的计算公式有时不能真正用于实际计算，这些公式理论上是正确的，但由于参与公式中各种运算的数是有误差的近似数，而误差在运算过程中会发生某些不良效应，从而导致计算结果的误差变得很大，使结果不可靠。这就是理论

上的解题方案与实际能用性之间所存在差异的一个方面。

理论上的解题方案与实际能用性之间所存在差异的另一种表现是计算工作量。例如，在线性代数中所介绍的求解线性代数方程组的克拉默(Cramer)法则，在理论上可以用克拉默法则求解所有线性代数方程组，但实际上是行不通的。因为用克拉默法则求解较大规模的线性代数方程组，所需要的运算时间是不能容忍的。因此，对于求解较大规模的线性代数方程组，克拉默法则并不实用。

3. 精确解法与近似解法一般没有本质差别

所谓精确解法是指公式解法或直接解法。近似解法一般是指迭代解法或数值解法。从理论上讲，用精确解法所得到的结果是精确的，而用近似解法所得到的结果是近似的。但实际情况并非如此。在所谓的精确解法中，参与运算的数都是近似的，并且每一步的运算又都有可能产生误差，这些误差通过运算后最终会影响结果。因此，即使使用精确的计算公式，其最后得到的结果一般不是绝对精确的，都包含有误差成分。甚至，如果算法选得不合适，即使是精确解法，也会得到错误的结果。在近似解法中，一般都要事先给定精度要求，在算法收敛情况下，计算过程中一般都要保证最后结果满足事先给定的精度要求。即近似解法得到的结果虽然是近似的，但一般都满足事先规定的精度要求，也就是说，近似解法的精度是可以控制的。因此，精确解法所得的结果未必一定准确，而近似解法所得的结果未必一定不准确，两者没有本质的差别。

1.2 误差与运算误差分析

1. 数值计算中误差的不可避免性

误差在数值计算中是不可避免的。也就是说，在数值方法中，绝大多数情况下不存在绝对的严格和精确。

有人会说，计算机科学的发展，为科学计算以及数据处理提供了高速和高精度的计算工具。这只是问题的一个方面，不可否认，由于计算机技术的发展，许多复杂的数值计算问题才能得以解决(这些问题用手算是不可想象的)，但计算机与任何计算工具一样，它所处理的数值型数据只能是近似的，如果处理不当，这些近似的数据经过大量的运算后，其后果有可能是相当严重的。实际上，在用计算机进行数值计算时，在各个环节上都有可能产生误差。

为了说明数值计算过程中误差的来源，下面简单介绍在用计算机解决实际问题的各主要步骤中所引进的各种误差。

1) 构造数学模型

为了便于进行数值计算，一般首先需要将实际问题归纳为数学问题，这就是常说的需要建立一个合适的数学模型。

在将实际问题归纳为数学问题时，通常总要附加许多限制，并且要忽略一些次要的因

素，以便建立起一个“理想化”的数学模型。因此，这样得到的数学模型实际上只是客观现象的一种近似描述。而这种经过归纳后的数学描述上的近似，必然也就引进了误差。这种数学描述上的近似所引进的误差称为模型误差。

在将实际问题归纳为数学问题的过程中，除了模型误差外，还有一种很重要的误差。在构造数学模型时，为了对问题本身作抽象近似，除了忽略一些次要因素外，还需要对主要因素通过实验观测取得各种有效数据，根据实验观测到的数据进行分析总结，从而确定数学模型中的各种参数。由于条件的限制，通过实验观测到的数据与真值之间往往是有一定差异的，这也就给计算引进了一定的误差。这种误差称为观测误差。

2）制定解题方案，确定计算的近似公式

数学模型建立后，计算机还不能直接解决。这是因为，对于计算机来说，只能作一些它所规定的、并且是有限次的运算或判断，以及在一些规定的设备上进行输入与输出。因此，还必须为数学模型建立一个便于用计算机进行计算的近似公式。

大家知道，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，还需要将数学模型变成实际可行的解题方案，即将数学模型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对某种无穷过程的“截断”或计算方法的近似。例如，对于收敛的无穷级数，通常用它前面的有限项之和来近似代替无穷级数的和，实际上抛弃了无穷级数后面的无穷多项，由此便产生了误差。又例如，用梯形公式计算积分的近似值，这方法本身就有一定的误差。这类误差统称为方法误差或截断误差。

3）上机计算

解题方案确定之后，就可以通过某种工具来具体描述解题步骤，然后编制计算机程序，调试通过后就可以在计算机上正式运行，最后得到所需要的结果。

计算机与任何计算工具一样，总是受有效数字位数的限制，在进行数值计算时，其处理的数据总是近似的。在计算机中，任何数据都要转换成二进制形式才能进行处理，而绝大部分的数值型数据是无法精确地用二进制形式表示的，也就是说，即使是一个准确的数，为了用计算机进行处理，在转换成二进制数时就变成近似的。这就说明，在计算机中，参加运算的数据只能具有有限位的有效数字，其超过部分将被无情地处理掉，即产生了误差。这种误差称为舍入误差。

由上所述，在数值计算过程中，误差的产生是不可避免的，其误差的类型也是各种各样的，它们会直接影响到计算结果的准确性。

虽然数值计算中的误差是不可避免的，但是，在解决实际问题时，应该尽量减少产生误差的机会，尽量减小某些误差或将它们限制在许可的范围之内。这是因为，误差在计算过程中会产生不好的效应。例如，某个参数由于观测引进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不大，但是，这些误差经过计算机的千百万（甚至更多）次的运算以后，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。

2. 绝对误差与相对误差

1）绝对误差

定义 1.1 设 x 为准确数，x^* 为其近似数。则

$$E(x) = x - x^*$$

称为近似数 x^* 关于准确数 x 的绝对误差。

一般来说，由于准确数 x 是未知的，因此，无法根据定义 1.1 准确地计算出某个近似数的绝对误差，而只能根据测量或计算的具体情况估计出误差绝对值的一个范围，也就是估计出 $|E(x)|$ 的上界。

设

$$|E(x)| = |x - x^*| \leqslant \eta$$

则称 η 为近似数 x^* 关于准确数 x 的绝对误差限。

前面说过，一般无法计算出由定义 1.1 所定义的绝对误差，因此，工程上就将绝对误差限称为绝对误差。在本书中，如果没有特殊说明，绝对误差即指绝对误差限，有时简称为误差。

当估计出近似数 x^* 关于准确数 x 的绝对误差限 η 后，工程上可以用以下两种方法表示准确数 x 所在的范围：

$$x - \eta \leqslant x \leqslant x^* + \eta$$

或

$$x = x^* \pm \eta$$

在计算函数值 $f(x)$ 时，当自变量 x 有一个误差时，其计算得到的函数值也有一个误差。如果给出了自变量 x 的绝对误差为 $E(x)$，则函数值的绝对误差可以用下式来估计：

$$E[f(x)] = f'(x)E(x)$$

2）相对误差

绝对误差的大小反映了近似数偏离准确数的程度，还不能完全反映近似数的准确程度。例如，设有两个量 x 和 y，其中 $x=10\pm1$，$y=1000\pm5$。显然，近似数 $y^*=1000$ 的绝对误差比近似数 $x^*=10$ 的绝对误差大了 4 倍，但并不能说 y^* 的准确程度要比 x^* 差，实际上正好相反，y^* 的准确程度要优于 x^*。

为了能够确切地表示一个近似数的准确程度，需要引进一个相对误差的概念。

定义 1.2 设 x 为准确数，x^* 为其近似数。则

$$E_r(x) = \frac{E(x)}{x} = \frac{x - x^*}{x}$$

称为近似数 x^* 关于准确数 x 的相对误差。

实际上，由于准确数 x 一般是不知道的，因此，相对误差通常又定义为

$$E_r(x) = \frac{E(x)}{x^*} = \frac{x - x^*}{x^*}$$

由上述定义可以看出，相对误差说明了近似数 x^* 关于准确数 x 的绝对误差 $E(x)$ 与近似数本身比较起来所占的比例，因而更客观地反映了该近似数的准确程度。