# 計算机数値方法

《计算机数值方法》编写组

福建省计算数学学会编辑

# 計算机数值方法

[illegible]

[illegible]

计算机数值方法

（福建中学数学增刊）

编辑者：福建省计算数学学会
印刷者：厦门大学印刷厂
订购处：厦门大学702信箱转
厦门市计算机学会

1984年2月闽版刊字第001号

# 目　录


**绪论** ……（1）

§1 为什么要研究数值方法？ ……（1）

§2 什么是计算机数值方法？ ……（2）

§3 误差的基本概念 ……（3）

§4 浮点数和浮点运算 ……（7）

§5 选用和设计算法应注意什么？ ……（10）

**第一章　方程求根** ……（16）

§1 根的隔离与对分法 ……（16）

§2 迭代法 ……（22）

§3 弦截法 ……（25）

§4 切线法 ……（28）

习题一 ……（33）

**第二章　插值法** ……（35）

§1 线性插值与二次插值 ……（36）

§2 差分与差商 ……（40）

§3 牛顿(Newton)插值法 ……（45）

§4 分段插值 ……（55）

§5 样条插值法 ……（58）

习题二 ……（68）

**第三章　数据处理方法** ……（70）

§1 曲线拟合与经验公式 ……（70）

§2 最小二乘法 ……（72）
§3 回归分析 ……（80）
§4 一元线性回归 ……（86）
§5 非线性回归方程的线性化 ……（101）
§6 二元线性回归 ……（101）
§7 多元（$n>2$）线性回归 ……（109）
习题三 ……（114）

**第四章 数值积分** ……（117）
§1 梯形公式与抛物线公式 ……（118）
§2 龙贝格（Romberg）求积公式 ……（129）
习题四 ……（135）

**第五章 常微分方程初值问题数值解** ……（137）
§1 欧拉（Euler）折线法及其改进 ……（137）
§2 龙格一库塔（Runge-Kutta）方法 ……（144）
§3 阿达姆斯（Adams）公式 ……（149）
习题五 ……（159）

**第六章 数值代数** ……（160）
§1 高斯（Gauss）消去法 ……（160）
§2 对称正定矩阵的平方根方法 ……（169）
§3 三对角线性方程组的追赶法 ……（175）
§4 雅可比（Jacobi）迭代法和高斯—赛德尔（Gauss—Seidel）迭代法 ……（178）
§5 矩阵的特征值 ……（185）
习题六 ……（197）

# 绪　论

## §1　为什么要研究数值方法？

计算机数值方法的主要作用，在于更有效地使用计算机，以求取各种数学问题的数值解。在使用电子计算机已成为人们的常识的今天，对于计算机上使用的数值方法也提出了更为迫切的要求。大家知道，计算机只能作加、减、乘、除算术运算和逻辑运算，而数学运算的范围是极为广阔的，既有算术运算，也有代数运算，还有各种各样的函数运算。由于生产实践和科学实验中提出的各种问题，在建立了数学模型之后，还不能立刻用计算机直接求解，因此，当用计算机求解时，就要研究计算机数值方法。也就是说，需要研究确定解决这些数学模型的适合于计算机上采用的数值方法，将数学公式转换成一系列相应的算法步骤，并从此出发，编制出一个正确的计算程序，再上机计算，得出有用的结果。

随着我国社会主义建设事业的迅速发展，电子计算机的应用日益扩大，使计算机数值方法在现代科学研究中的地位与作用不断提高，适合于计算机求解的实际问题，无论是规模还是种类都在迅速增长。因此，建立数值方法的基本概念和计算机上常用的算法，这对计算机使用者来说是完全必要的。

计算机数值方法对于自然科学和社会科学的发展及国防现代化建设都起着重要的推动作用。具有非凡计算能力的电子计算机从诞生时起，就与科学研究工作结下了不解之缘。

可以说，近代尖端技术的发展是建立在电子计算机的基础上，而计算机数值方法的发展将会导致许多科学研究的新突破。

电子计算机在工业部门广泛应用的结果，是给整个工业生产带来一场新的革命。计算机数值方法广泛用来进行各种各样的设计。如产品设计、工程设计、材料设计等等，从而大大地提高了设计的质量和效率。计算机数值方法也是现代农业的好助手。以农业机械为例，为了保证农机具合理的刚度、强度、耐磨损能力与作业性能，并降低钢材的消耗，大型农机具的设计都要用计算机进行计算。在水利工程中使用计算机，经济效果显著，提高了质量，缩短了施工周期，以我国一个高水坝的设计为例，如果用人工计算，需要几个月才能完成设计任务，采用了计算机数值方法，只用三个星期就顺利完成了，而且节约工程投资达一千万元。

除外，还有原子反应堆设计、天气数值预报、弹道计算和分析、飞机结构计算、流体力学问题计算、分子结构计算、爆炸问题的数值计算、数值试验、计划管理等等。总之，计算机数值方法是与实现四个现代化密切相关的重要应用领域，今后它将为加速现代化的建设发挥更大的作用。

## §2 什么是计算机数值方法？

数值计算方法是近代数学的一个重要的分支，它专门研究数学问题的数值解法，这包括方法的推导、对方法的描述以及对整个求解过程的分析。它不同于分析数学，分析数学主要研究运算规则和表达的方式。数值计算关心的是运算的数值结果，虽然多数数值方法是为使用数字计算机而提出

的，但是数值方法的研究对象与计算机程序设计、信息处理也是不同的。

计算机数值方法是计算机科学的重要内容。如果数值方法选用不当，编出的计算程序所需的存贮容量可能太大，以致超过计算机的存贮能力而无法进行；或是计算出的结果精确度不高达不到要求；或是需要计算机化费太多的时间才能完成等等。但如果采用的数值方法较好，同样一台计算机就能发挥更大的作用。因此，研究如何把数学模型归结为数值问题；如何制定快迅的算法；如何估计一个给定算法的精度，分析误差在计算过程中的积累和传播；如何构造精度更高的算法；如何使一个算法较少地占用内存和调用外存；如何对现有的和新提出来的算法的优缺点进行分析比较，分析解的特性和适用范围以及对平行算法的研究等等，构成了计算机数值方法的内容。

计算机数值方法涉及的问题相当广泛，本书仅介绍必要的基本概念，选择一些常用的行之有效的算法，作为进一步学习这方面内容的入门。解决实际问题时，应当根据问题的要求，计算机的性能，选用良好的算法，以便得出精确的结果。

## §3 误差的基本概念

误差是由各种原因产生的。用数值计算方法解决科学技术中的具体问题，首先必须建立这个具体问题的**数学模型**。具体问题的数学描述都是在一定条件下理想化了，所以，数学模型总是近似的，它与实际现象之间总存在误差，其误差称为**模型误差**。数学模型中通常包含观测数据，观测的结果

和实际的大小之间有误差，这种误差称为**观测误差**。当实际问题的数学模型很复杂，因而不能获得其精确解时，就必须建立一套行之有效的数值方法，模型的精确解与数值方法准确解之差称为**截断误差**。由于实际计算是按有限位数进行的，所以数值解的每一步都可能有误差，这种误差称为**舍入误差**。例如，对某个问题，我们需要精确到小数点后五位，虽然对于普通算术运算中的实际数字，可以采用十分精确的数值方法来计算，但遗憾的是，由于计算机所表示的数字仅仅是该数的开头的若干位，因而精确度可能达不到要求，这种由于使用计算机而引起的误差就是舍入误差。

舍入误差（包括原始数据的误差）和截断误差将是数值方法研究的对象，讨论它们在计算过程中的传播和对计算结果的影响，并找出误差的界。误差的界和估计对于研究误差的渐近特性和改进算法的近似程度具有重大的实际意义。

**1. 绝对误差**

一个数的绝对误差是它的精确值减去它的近似值。若某一数的精确值为 $x$，其近似值为 $x^*$，那么 $x$ 与 $x^*$ 之差

$$E(x)=x-x^* \qquad (0.1)$$

称为近似值 $x^*$ 的绝对误差。简称**误差**。

精确值 $x$ 一般是未知的，因而 $E(x)$ 不能求出，但往往可以估计出它的大小范围，亦即可以确定一正数 $\eta$，使

$$|E(x)|=|x-x^*|\leqslant\eta \qquad (0.2)$$

此时，$\eta$ 称为 $x^*$ 的**绝对误差界**。有时也用

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

表示 $x^*$ 的精确度或精确值所在的范围。绝对误差是有量纲单位的。

### 2．相对误差

由于对各种不同的问题计算所得的结果，其数值大小相差很大，只用绝对误差还不能说明数的近似程度，例如，甲穿孔时平均每百个符号错一个，乙穿孔时大约每五百个符号错一个，他们的误差都是错一个，但显然乙穿孔时要准确些。因此，除了要看绝对误差大小外，还必须顾及量本身，这就需要引入相对误差的概念。绝对误差与精确值之比，即

$$E_r(x)=\frac{E(x)}{x}=\frac{x-x^*}{x} \qquad (0.3)$$

称为 $x^*$ 的相对误差。由于精确值 $x$ 一般不知道，实际计算时常用下式作为 $x^*$ 的相对误差

$$E_r^*(x)=\frac{x-x^*}{x^*} \qquad (0.4)$$

同样的，若能求出一正数 $\delta$，使 $|E_r(x)|\leqslant\delta$，则 $\delta$ 称为 $x^*$ 的**相对误差界**。相对误差是无量纲的数，通常用百分比表示，称百分误差。

根据上述定义可知，甲穿孔时的相对误差 $|E_r^*(x)|\leqslant\frac{1}{100}=1\%$。乙穿孔时的相对误差 $|E_r^*(x)|\leqslant\frac{1}{500}=0.2\%$，比甲穿孔时精确。所以，在分析误差时，相对误差更能刻划误差的特性。

### 3．有效数字

一个实数 $x$ 总可以表示成无穷小数的形式

$$x=\pm(0.\alpha_1\alpha_2\cdots\alpha_n\alpha_{n+1}\cdots)_{10}\times10^m \qquad (0.5)$$

其中 $\alpha_1,\alpha_2,\cdots\alpha_n,\alpha_{n+1},\cdots$ 都是0,1,2,…9中的一个数字，$m$，$n$ 均为整数。在实际计算时，我们只能取有限位数字，而把其后的数字都去掉。为了方便起见，假定 $m=0$，$\alpha_1\neq0$，现

把（0.5）式分成两个部分

$$x=0.\alpha_1\alpha_2\cdots\alpha_n+0.\underbrace{0\cdots0}_{n}\alpha_{n+1}\cdots$$

称前一部分为有效部分，后一部分为残余部分。为了尽可能减少误差，应根据残余部分的大小来调整有效部分的最后一个数字 $\alpha_n$，一般**舍入规则**如下：

（ i ）若 $\alpha_{n+1}<5$，则 $\alpha_n$ 后面数字全部舍去。

（ii ）若 $\alpha_{n+1}>5$，则 $\alpha_n$ 进 1 。

（iii ）当 $\alpha_{n+1}=5$ 时，若 $\alpha_{n+1}$ 后面有非零数字则 $\alpha_n$ 进 1；若 $\alpha_{n+1}$ 后面无非零数字则视 $\alpha_n$ 奇偶性，采用“奇进偶不进”。即 $\alpha_n$ 是奇进 1，$\alpha_n$ 是偶不进。

例如，将下列各数简化为只保留小数点后 4 位。

$\pi=3.1415926\cdots\approx3.1416$（入）　$4.93125\approx4.9312$　（舍）

$4.165238\approx4.1652$　（舍）　$9.5431500\approx9.5432$（入）

$0.13405006\approx0.1341$（入）

为了简单起见，我们把通常所说的“四舍五入”抽象成数学语言，对于形如（0.5）式的 $x$，经舍入规则后，得到近似数为

$$x^*=\begin{cases}\pm(0.\alpha_1\alpha_2\cdots\alpha_n)_{10}\times10^m, \\ \qquad\qquad 0\leqslant\alpha_{n+1}<10/2 \\ \pm[(0.\alpha_1\alpha_2\cdots\alpha_n)_{10}+10^{-n}]\times10^m, \\ \qquad\qquad 10/2\leqslant\alpha_{n+1}<10\end{cases}\qquad(0.6)$$

其中 $\alpha_1\neq0$。为了可以从近似数的有限小数表示本身，就能知道近似数的精度。我们引入有效数字。显然，由舍入规则可知，近似数 $x^*$ 的绝对误差不超过末位数的半个单位。即

$$|x-x^*|\leqslant\frac{1}{2}\times10^{m-n}\qquad(0.7)$$

**定义** 对于形如（0.6）式的 $x^*$，若其绝对误差界满足条件（0.7），便说 $x^*$ 为具有 $n$ 位有效数字的有效数，而每一位数字 $\alpha_1,\alpha_2,\cdots\alpha_n$ 都叫做 $x^*$ 的**有效数字**。

例如，易从定义验证3.1416是 $\pi$ 的具有5位有效数字的近似值。有效数字不但给出了近似数的大小，而且还给出了它的绝对误差界。如有效数字3587.64，$0.158\times10^{-2}$，$0.1580\times10^{-2}$ 的绝对误差界分别为 $\frac{1}{2}\times10^{-2}$，$\frac{1}{2}\times10^{-5}$，$\frac{1}{2}\times10^{-6}$。特别要注意有效数字的指数记法，$0.158\times10^{-2}$ 为3位有效数字，而 $0.1580\times10^{-2}$ 是4位有效数字。

## §4 浮点数和浮点运算

### 1．浮点数

数字电子计算机进行运算时，必须按一定方法确定小数点的位置。针对小数点的处理，机器中表示数有两种方法，一种叫定点表示法。另一种叫浮点表示法。如果在数的表示中，小数点的位置均是固定的，些种表示法称为定点表示法。如果在数的表示中，小数点的位置是可以变动的，这种表示法称为浮点表示法。

一般地，我们把计算机中浮点数的全体组成的集合记为 $G$，那么 $G$ 中的**浮点数**具有如下形式

$$z^*=\sigma(0.\alpha_1\alpha_2\cdots\alpha_t)_\beta\times\beta^e\qquad(0.8)$$

其中 $\sigma=\pm1$ 称为数符；$\alpha_1,\alpha_2,\cdots\alpha_t$ 均为整数，它们满足关系 $0\leqslant\alpha_i\leqslant\beta-1$，$(i=1,2,\cdots t)$；$\beta$ 为浮点数的基底，一般

$\beta=10$，2或16；自然数 $t$ 称为计算机字长；数 $e$ 为浮点表示的阶码，它有固定的上限 $M$ 和下限 $N$，通常取为 $-M\leqslant e\leqslant M$；$\beta^e$ 称为定位，用来确定浮点数的小数点的真实位置；浮点数的阶码其范围可以用表示阶数的位数来确定，这个位数仍记为 $e$；数 $(0.\alpha_1\alpha_2\cdots\alpha_t)_\beta$ 称为尾数，这些数的乘幂是以 $\beta$ 为基底的。因此，浮点数分为阶码和尾数两个部分，均有各自的符号位，对不同的计算机，尾数和阶码是不同的。

若对 $z^*\neq 0$ 规定（0.8）式中的 $\alpha_1\neq 0$，则 $G$ 为规格化的浮点数系。当 $\beta=10$，$t=4$，$e=99$ 时，$-0.0001\times 10^{-99}$ 和 $+0.0001\times 10^{-99}$ 是数系 $G$ 中绝对值最小的非零数，而 $-0.9999\times 10^{99}$ 和 $+0.9999\times 10^{99}$ 是此数系中的最小数和最大数。

若记 $a^*=\sigma(0.\alpha_1\alpha_2\cdots\alpha_t)_\beta$，则（0.8）式改写为

$$z^*=a^*\times\beta^e$$

其中 $|a^*|<1$，这样在计算机的存贮单元中需用一对数 $(a^*,e)$ 来表示浮点数。用 $(0,0)$ 表示数零。例如，$11.375=0.11375\times 10^2=(0.11375,+2)$ 在程序中常写成 $0.11375\times 10^2=0.11375_{10}2$，而二进制的浮点表示如下

$$0.1011011_2100$$

**2．浮点运算**

我们知道，浮点数系 $G$ 是一个离散的有限集合，在利用计算机进行计算时，初始数据和中间结果都可能不在 $G$ 中，于是就产生用 $G$ 中的数来近似地表示相应数据的问题。假设数 $z\notin G$，我们要用 $G$ 中的一个浮点数作为 $z$ 的近似，记这个浮点数为 $f^l(z)$，它应有最好的逼近性质，即

$$|z-f^l(z)|=\min_{g\in G}|z-g|$$

如果 $z$ 的绝对值大于 G 中的最大正数或小于 G 中的最小正数，则称为溢出。对于二进制系统，即 $\beta=2$，$a_l$ 为 0 或 1，一般也仿照十进制的四舍五入那样“1 进 0 舍”。因为后一位的 1 恰好是前一位的 1/2。通常为避免见 1 就进而造成统计偏差。因此，确定规则：当 $a_{t+1}=0$ 时，则舍去，当 $a_{t+1}=1$ 时，若 $a_t=0$ 则进 1，若 $a_t=1$ 则舍去。有的计算机却只舍而不入，这时产生的误差要增大二倍。

类似十进制数的讨论，对于给定的浮点数 $z$，设

$$z=\sigma(0.a_1a_2\cdots a_ta_{t+1}\cdots)_\beta\times\beta^e$$

经舍入规则后为

$$f^l(z)=\begin{cases}\sigma(0.a_1a_2\cdots a_t)_\beta\times\beta^e, \\ \qquad\qquad 0\leqslant a_{t+1}<\beta/2 \\ \sigma[(0.a_1a_2\cdots a_t)_\beta+\beta^{-t}]\times\beta^e, \\ \qquad\qquad \beta/2\leqslant a_{t+1}<\beta\end{cases} \tag{0.9}$$

其中 $a_1\neq 0$。显然，由舍入规则可知，$f^l(z)$ 的绝对误差界为

$$|z-f^l(z)|\leqslant\frac{1}{2}\times\beta^{e-t} \tag{0.10}$$

由于 $\beta^{-1}\leqslant|\sigma(0.a_1a_2\cdots a_ta_{t+1}\cdots)_\beta|<1$，所以 $|z|\geqslant\beta^{e-1}$，因此，$f^l(z)$ 的相对误差界为

$$\frac{|z-f^l(z)|}{|z|}\leqslant\frac{1}{2}\times\beta^{-t+1} \tag{0.11}$$

记 $\mathrm{eps}=\frac{1}{2}\times\beta^{-t+1}$ 为 $f^l(z)$ 的相对误差界，我们称 $\mathrm{eps}=\frac{1}{2}\times\beta^{-t+1}$ 为计算机的精度。由于舍入方法不同，亦即按“舍入规则”及“只舍不入”，在计算机中进行浮点运算分

别称为舍入运算和切断运算。

若把 $\varepsilon$ 定义为 $f^l(z)$ 的相对误差，些么由（0.11）式可知

$$f^l(z)=(1+\varepsilon)z, \qquad |\varepsilon|\leqslant\frac{1}{2}\times\beta^{-t+1} \qquad (0.12)$$

所以，$z\cdot\varepsilon$ 为用浮点数 $f^l(z)$ 表示 $z$ 时的绝对误差。

设 $x,y(y\neq0)$ 均为规格化的浮点数，$x,y\in G$，它们的算术运算 $x+y$，$x-y$，$x\times y$，$x/y$ 的精确结果则不一定是 $G$ 中的浮点数。在计算机中进行浮点运算时，计算机自动把运算结果用 $G$ 中的浮点数表示出来，记算术运算的结果为 $f^l(x+y)$，$f^l(x-y)$，$f^l(x\times y)$，$f^l(x/y)$。这些运算结果怎样用机器数系来表示，随计算机的功能而不同。由（0.12）式可知浮点运算和精确运算之间的关系为

$$f^l(x+y)=(x+y)(1+\varepsilon_1),f^l(x-y)=(x-y)(1+\varepsilon_2)$$
$$f^l(x\times y)=(x\times y)(1+\varepsilon_3),f^l(x/y)=(x/y)(1+\varepsilon_4),$$

其中 $|\varepsilon_i|\leqslant\text{eps}=\frac{1}{2}\times\beta^{-t+1}$，$(i=1,2,3,4)$，由于大多数计算机都配置有使舍入为最佳的算法，所以，上述条件是满足的。

## §5 选用和设计算法应注意什么？

算法是为使用数字计算机而提出的，一般可分为数值算法和非数值算法。我们把利用计算机来解决本质上是非数值问题的方法，称为非数值算法，它用的是计算机进行判断的能力。而把计算机传统地同数值问题的求解方法，称为数值算法，它用的是计算机进行算术运算的能力。从本书的目的出发，我们将算法定义为数学问题构造性解法的一个完备而

确切的描述。且仅讨论数值算法。

解决数值问题，不仅需要算法，而且要求选用和设计出好的算法。许多事例说明，如果算法选用不当，计算机的利用效率就得不到充分的发挥。衡量算法的标准一般有：运算次数的多少；算法的存贮量多少；逻辑结构是否简单；算法是否稳定等等。当这些要求不能兼备时，就应根据需要，权衡利弊，综合平衡而作决择。

**1．注意算法步骤简化，减少运算次数**

现以多项式的求值问题为例，说明减少运算次数的重要性和可能性，例如，计算多项式

$$P_n(x)=\sum_{k=0}^{n}a_kx^k$$

的值。若直接逐项求和运算，算 $a_kx^k$ 要 $k$ 次乘法，一共要 $\frac{1}{2}n(n+1)$ 次乘法和 $n$ 次加法，但若按下列递推关系式

$$\begin{cases}u_0=a_n\\ u_k=u_{k-1}\cdot x+a_{n-k}\end{cases}\qquad(0.13)$$

对 $k=1,2,\cdots$ 直到 $n$，反复执行算式（0.13），最终得到的 $u_n$ 就是所求的结果。只要 $n$ 次乘法和 $n$ 次加法，而且逻辑结构简单。算式（0.13）就是著名的秦九韶算法。一般说来，在一个工程问题中，通过算法步骤的简化不仅能减少运算次数、提高计算速度，而且还能简化逻辑结构，减少误差的积累。

**2．防止两数相近进行减法运算**

两个正数之差 $z=x-y$ 的相对误差是

$$E_r(z)=\frac{E(x)-E(y)}{x-y}$$

若两个数 $x$ 和 $y$ 很接近，它们的差的相对误差就很大，这是由于 $x^*$ 和 $y^*$ 的前几位相同的有效数字在它们的差 $x^*-y^*$ 内全被减掉了，所以，遇到这种情况，在 $x^*$ 和 $y^*$ 内应多保留几位有效数字，或者对公式进行处理，避免减法，特别要避免再用这个差作为除数。例如，求 $z=\sqrt{a+1}-\sqrt{a}$ 之值，当 $a=1000$，取 4 位数字计算 $\sqrt{a+1}=31.64$，$\sqrt{a}=31.62$，两者直接相减得 $z=0.02$。这个结果只有一位有效数字（$z$ 的实际值为 0.01580），但若对公式处理成

$$z=\sqrt{a+1}-\sqrt{a}=\frac{1}{\sqrt{a+1}+\sqrt{a}},$$

则按后一公式求得的结果 $z=0.01581$ 有三位有效数字，可见改变计算公式可以避免相近两数相减引起的有效数字损失，而得出比较精确的结果。

**3．设法控制误差的传播和积累**

用不同的算法解决同一问题时，所得结果的精度可能大不相同；这是由于初始数据的误差或计算中某一步产生的舍入误差，在计算过程中的传播常因算法而异，于是就产生了算法的数值稳定性问题。一个算法，如果初始数据的误差对计算结果的影响较小，便说这个算法具有较好的数值稳定性；反之，如果初始数据的误差对计算结果影响很大，便说这个算法的数值稳定性不好，解决数值问题，要选用稳定性较好的算法。

怎样减少计算中误差的积累呢？

首先要注意浮点运算的特点，作为一个例子说明误差在