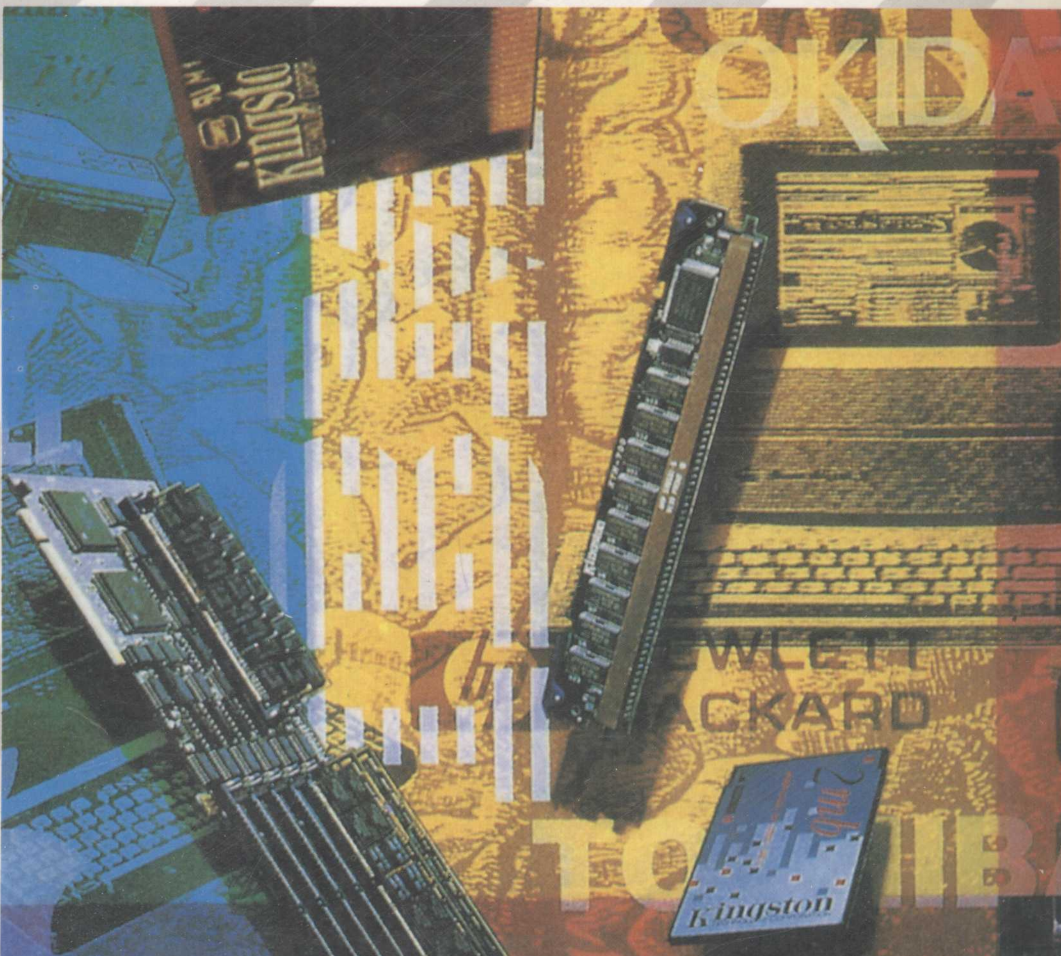
计算机科学大众丛书

DZ

微机实用数值计算
——算法与程序

任开隆　杨奇峰等　编著



電子工業出版社

微机实用数值计算
——算法与程序

任开隆　杨奇峰等　编著

電子工業出版社

内容简介

本书介绍了科学和工程中常用的数值计算方法及其在微机上的应用。给出了工程技术中常见问题的各种算法和实用程序，每个程序都用三种语言 Fortran. Pascal 和 C 编制，并配有软盘。

本书可供具有中专以上文化水平的工程技术人员自学，亦可作为大专院校的有关专业及培训班的教材。

《计算机科学大众》丛书编委会

主任委员　林定基

副主任委员　曹东启　丁嘉种

编　　委(以姓氏笔划为序)

刘兆毓　刘彦明　刘克武　朱家维

宋玉升　郑锡琏　彭裕禄　秦志斌

秘　书　袁　玫　徐海波

微机实用数值计算——算法与程序

任开隆　杨奇峰等　编著

责任编辑　宋玉升

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行　各地新华书店经销

北京大中印刷厂印刷

★

开本：787×1092毫米/1/16印张：16　字数：380千字

1996年2月第一版　1996年2月第一次印刷

印数：3000册　定价：22元

ISBN 7-5063-3254-6/TP·1207

前　言

近年来，微型计算机的迅速发展和普及，使微型计算机在各个领域中得到广泛的应用。科学和工程中常用的数值计算方法，是微型计算机应用中的极为重要的基础。无论是计算机专业还是非计算机专业的技术人员，都需要掌握数值计算的基本概念和方法，并能把它应用到微计算机上，这正是本书的目的。

本书不注重数值计算方法在数学上的理论推导，而主要介绍数值计算的基本思想和工程技术中常见问题的各种算法及其实现，并对常用算法提供了以三种语言 Fortran、Pascal 和 C 编写实用程序装入一张软盘，供读者进行练习和计算时使用。考虑到微型计算机的存储容量和运算速度，本书提供的一些算法尽量占用较少的存储空间，同时保证了计算精度。本书在一些章的结尾有方法评述，可以帮助读者选择解决问题的最佳算法，并提示读者在微机上应用这些算法时要注意的某些问题。本书还介绍了广泛应用的统计方法及其在微机上的实现，如随机变量的抽样、蒙特卡洛方法、假设检验、方差分析、判别分析及预测技术等，并附有一些应用实例，便于读者理解和应用。

本书在内容选材方面，力求体现科学性、实用性和先进性，做到内容全面、重点突出，在编写上力求做到深入浅出、通谷易懂。本书不仅适合具有中专以上文化水平的工程技术人员和管理干部自学，达到学会应用微机解决实际的数值计算问题，而且可作为大专院校的计算机专业及培训班的教材。

本书编写过程中，得到林定基教授的指导和帮助，作者在此表示衷心地感谢。

本书由任开隆负责全书内容的设计与构思，杨奇峰副教授编写了一至六章的初稿，在初稿的基础上修改补充，最后完成本书，其中第一、四、五章由王信峰同志完成，第三、六章由江新华同志完成，第二、七章由任开隆完成，张翼给出了全部 C 语言程序和部分 Pascal 语言程序，车燕给出了部分 Pascal 语言程序和全部 Fortran 语言程序，最后由任开隆对全书统编和审阅。

由于作者水平所限，不妥之处在所难免，敬请读者不吝赐教。

编者

1995 年 5 月于北京

目　　录

第一章　数字计算中的误差

第二章　非线性方程的解

此为试读，需要完整PDF请访问：www.ertongbook.com

第四章 逼近与插值

第五章 数值积分与数值微分

第六章 常微分方程数值解法

第七章　统计方法

第一章　数值计算中的误差

科学的迅猛发展，使计算机已成为科学研究与工程设计的主要工具之一。在实际工作中，由于受计算机字长的限制，我们不得不将无限小数截取有限位，转化为计算机能表示出的近似值，并进行这些近似值间的运算，这就是数值计算。将待求解问题的数学模型（即能描述并等价于实际问题的数学问题）简化为一系列算术运算、逻辑运算等，以便在计算机上求出问题解的近似值（常称为数值解）的方法，叫做数值计算方法，简称算法。我们的问题是，如何规定运算的次序、运算方式等，提出相应算法，并对算法的收敛性、稳定性，特别是误差加以分析？

本章的目的是帮助我们认识误差，了解误差对运算结果的影响，掌握误差分析的基本方法。

§1.1　误差的种类及来源

产生误差的原因多种多样，但常见的有以下几种。

1.1.1　模型误差

解决工程技术实际问题常要首先建立对实际问题的数学描述——数学模型。建立模型的时候，为便于分析和计算，需要对问题进行合理的简化，忽略掉一些次要因素，使建立的模型既能解决实际问题，又简单易行。但无论如何，合理简化的结果必会使建立的模型与实际问题间产生一定的差异，使模型仅能近似地描述实际问题，由此而产生的误差称为模型误差。

模型误差过大时，则要考虑修改模型。

1.1.2　观测误差

为研究实际问题，常要首先搜集有关的数据。实验观察、测量等都是获取数据的重要手段。受观测手段（实验仪器与测量仪器等）的限制，得到的数据只能是一些近似值，带有一定的误差，这种误差称为测量误差，或观测误差。

通常，观测误差可根据测量工具、实验仪器本身的精度来确定。但有时也受到某种干扰而带有随机性，也很难估计。

1.1.3　舍入误差

数据计算过程中，要对无限小数（如 $\pi=3.14159265\cdots$，$\frac{1}{3}=0.3333\cdots$ 等）进行舍入。常见的舍入规则有："只舍不入"、"只入不舍"与"四舍五入"。至于要用哪种舍入规则常由实际问题来定。由舍入而引起的误差称为舍入误差。

本书中仅考虑由“四舍五入”而引起的舍入误差。

1.1.4 截断误差

计算机只能完成有限次算术与逻辑运算。对于一些超越运算(如微分、积分、无穷级数求和等),则要化为一系列有限次的算术运算或逻辑运算来处理,处理的结果也会产生一定的误差。例如,利用函数 $\sin x$ 的幂级数展开式:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

计算 $\sin 0.5$ 的近似值时,取

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!}$$

则会产生误差

$$-\frac{x^7}{7!} + \frac{x^9}{9!} - \cdots \triangleq e_4(x)$$

(在此为 $e_4(0.5)$,是由甩掉无穷级数自第四项开始后的所有项产生的)。计算 $\sin 0.5$ 时产生的误差满足

$$|e_4(0.5)| \leqslant \frac{0.5^7}{7!} \approx 1.55 \times 10^{-6}$$

这种由仅保留了超越运算过程中的有限次运算部分而引起的误差,称为截断误差。被舍去的部分称为余项。

一般来说,截断误差常可根据算法所舍去的余项来估计。

上述四种误差,尽管单独看来常是微不足道的,但科学研究与工程设计常要进行成千上万次的计算;计算中的多次舍入、截断等,均会造成误差的积累,甚至使结果严重偏离真值(这在后面例子中会看到),因而要对误差予以足够的重视。

通常,模型误差和观测误差往往是计算工作者不能独立解决的。本书主要讨论截断误差与舍入误差。

§1.2 误差的表示

要认识误差,分析误差,则应给出能体现误差的量,本节的绝对误差与相对误差描述了误差在量上的本质特征。

1.2.1 绝对误差

设 a 是精确值 x(即真值)的一个近似值,常称

$$e(x) = x - a,$$

为近似值 a 的绝对误差,简称误差。又若 ε 是 $e(x)$ 的一个上界,即

$$|e(x)| \leqslant \varepsilon$$

则称 ε 为近似值 a 的绝对误差限或绝对误差界,简称误差限。

例如,用刻有毫米的米尺测量一个桌子,测得桌子长为 985mm,设 x 是桌子的真正长度,则 985 实际上为 x 的一个近似值,由于测量误差不会超过半个毫米,即

此为试读,需要完整PDF请访问: www.ertongbook.com

$$|985 - x| \leqslant 0.5(\text{mm})$$

从而

$$985 - 0.5 \leqslant x \leqslant 985 + 0.5$$

这说明 x 的真值在区间[984.5,985.5]内,通常也记为

$$x = 985 \pm 0.5$$

上例中,由于真值 x 无法测得,绝对误差 $985-x$ 实际上是根本无法知道的,但我们却很容易得到了其绝对误差限。实用中,往往能得到误差限也就够了。

1.2.2 相对误差

用绝对误差及其误差限来刻划近似值的精确程度是有局限性的。例如,现设你要买煤,由于秤的原因,无论你买多少煤,都会少给你 5 千克,那么你会选择买 50 千克还是选择买一吨煤?当然你会毫不犹豫地选择买一吨煤。为什么呢?因为你认为买 50 千克煤“差的太多了”。这里两种情况下的绝对误差都是 5 千克,“差的太多”不能用绝对误差来刻划,它是相对于被衡量值本身的大小而言的。

我们称绝对误差与被衡量值真值之比,即

$$\frac{e(x)}{x} = \frac{x-a}{x} \triangleq e_r(x) \tag{1.1}$$

为近似值 a 的相对误差。(注意,今后我们常使用符号$\triangleq$表示“记为”)若有一个正数 δ 使

$$|e_r(x)| \leqslant \delta$$

则称 δ 为近似值 a 的相对误差限。注意,相对误差与相对误差限是无量纲的。

前述例子中,选择买一吨煤的相对误差为 5/1000=1/200,而买 50 千克的相对误差为 5/50=1/10,为前种相对误差的 20 倍!

由(1.1)式可知,相对误差与绝对误差间有如下关系:

$$e(x) = e_r(x)x$$

实际上,(1.1)式蕴含有:单位真值中所含的绝对误差就是相对误差。因此,在进行误差分析时,相对误差常比绝对误差更能准确描述近似值的精确程度。然而,x 的真值通常总是无法知道的,即使可以估计出绝对误差与绝对误差限,也无法由(1.1)式给出相对误差与相对误差限。实际应用中,常以

$$e'_r(x) = \frac{e(x)}{a} \tag{1.2}$$

作为相对误差。容易证明,$e'_r(x)$与 $e(x)$间仅差一个 $e_r(x)$的高阶无穷小量,因而用 $e'_r(x)$近似代替 $e_r(x)$不会引起更明显的误差。

另外,相对误差也常用百分数表示,如本节开始的例子中,两个相对误差可以分别表示为 0.5%与 10%。

最后,需要指明的是,分析误差常要同时考虑相对误差与绝对误差。

1.2.3 有效数字

设 x 是一个实数,用四舍五入规则舍入后,近似值的误差限为其末位的半个单位。如,$\pi=3.1415926\cdots$,四舍五入取四位小数,近似值为 3.1416,此时

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

若 x 的近似值 a 的误差限是其某一位的半个单位，则从这位起直到左边第一个非零数字为止的所有数字都称为 a 的有效数字。

具体地说，若 x 的近似值

$$a = (\pm d_1 \times 10^{m-1} + d_1 \times 10^{m-2} + \cdots + d_l \times 10^{m-l} + \cdots + d_s \times 10^{m-s}$$

其中 $s \geqslant l$，简记为($d_1 \neq 0$)

$$a = \pm 0.d_1 d_2 \cdots d_l \cdots d_s \times 10^m \tag{1.3}$$

若误差满足

$$|e(x)| = |x - a| \leqslant \frac{1}{2} \times 10^{m-l}$$

则称近似值 a 具有 l 位有效数字，且 $d_1, d_2, \cdots, d_l$ 即为 a 的 l 位有效数字。

例如，3.1416 是 π 具有 5 位有效数字的近似值，3.1415 是 π 具有 4 位有效数字的近似值。

用计算机进行数值计算时，受计算机字长的限制，参加数值计算的数有一定的位数，计算结果也只能保留有限位，所保留下来的各位数字不一定都是有效数字；同时，即使保留下的都是有效数字，也往往只是计算舍去一些有效数字后的结果。

1.2.4 有效数字与误差的关系

设 a 是 x 的近似值且具有式(1.3)的形式，则

$$|a| \geqslant d_1 \times 10^{m-1}$$

故，由(1.2)式，忽略 $e_r(x)$ 与 $e'_r(x)$ 的差别

$$|e_r(x)| = \left|\frac{e(x)}{a}\right| \leqslant \frac{0.5 \times 10^{m-l}}{d_1 \times 10^{m-1}} = \frac{1}{2d_1} \times 10^{-l+1}$$

因此，当 $a = \pm 0.d_1 d_2 \cdots d_l \cdots d_s \times 10^{m-1}$ 作为 x 的近似值具有 l 位有效数字时，相对误差满足

$$|e_r(x)| \leqslant \frac{1}{2d_1} \times 10^{-l+1}$$

反之，若数 $a = 0.d_1 d_2 \cdots d_l \cdots d_s \times 10^{m-1}$(或 $a = -0.d_1 d_2 \cdots d_l \cdots d_s \times 10^{m-1}$)是 x 的一个近似值，相对误差限为：

$$\frac{1}{2(d_1 + 1)} \times 10^{-l+1}$$

则易证明，a 至少具有 l 位有效数字。这就是有效数字与误差间的关系。

利用此关系，不仅可以估计近似值的相对误差，而且还可以求得为使精度(相对误差)在一定范围内时应取的有效位数。

例如，用 3.1416 作为 π 的近似值，其相对误差必满足：

$$|e_r(\pi)| \leqslant \frac{1}{2 \times 3} \times 10^{-5+1} \approx 1.667 \times 10^{-5}$$

又如，为使 π 的近似值 a 的相对误差不超过 0.2%，注意到 $\pi = 3.14159\cdots$，应使

$$|e_r(\pi)| = \frac{1}{2 \times (3+1)} \times 10^{-m+1} \leqslant 0.2\%$$

可求出 $m \geqslant 3$，从而，取三位有效数字即可，即应取 $\pi = 3.14$。

§1.3 误差分析

1.3.1 算术运算中误差的传播

由误差概念与微分定义比较可以发现，真值 x 处的绝对误差可以看做此处的微分，即

$$e(x) = dx$$

则

$$e_r(x) = \frac{1}{x}e(x) = \frac{1}{x}d(x) = d(\ln x)$$

这说明，x 的近似值 a 的相对误差可以看做 $\ln x$ 的微分。可见，微分是描述相对误差和绝对误差的一个有效工具。

利用微分与绝对误差、相对误差的上述关系，容易得到

$$e(x \pm y) = e(x) \pm e(y)$$

$$e(xy) = xe(y) + ye(x)$$

$$e\left(\frac{x}{y}\right) = \frac{1}{y}e(x) - \frac{x}{y^2}e(y) = \frac{x}{y}[e_r(x) - e_r(y)]$$

$$e_r(x + y) = \frac{e(x) + e(y)}{x + y} = \frac{xe_r(x) + ye_r(y)}{x + y}$$

$$e_r(x - y) = \frac{xe_r(x) - ye_r(y)}{x - y}$$

$$e_r(xy) = e_r(x) + e_r(y)$$

$$e_r\left(\frac{x}{y}\right) = e_r(x) - e_r(y)$$

由这些公式可知

1. 大小相近的两个同号数相减时，结果的相对误差可能会很大，而使计算结果的有效数字严重丢失，精度降低。

2. 两个数相乘，其中一个数很大时，乘积的绝对误差可能会增大。

3. 当除数的绝对值很小时，商的绝对误差可能增大。

因此，应尽量避免大小相近的两个同号数相减，避免乘数的绝对值很大以及除数接近于零。如何避免呢？下面对此进行讨论。

1.3.2 减小误差的方法

误差是不可避免的，这是由我们所使用的计算工具——计算机所决定的。但根据经验，通过误差分析，避免出现较大误差，保证精度在允许范围内是可以做到的。关于误差分析方法，将在下一节中讨论；在此我们介绍几种常用的减小误差的方法。

1. 对于前边 1.3.1 中三种误差可能增大的情况，常用将公式变形的方法来解决。例如，当 x 很小时，$\cos x$ 接近等于 1。若我们要计算 $1-\cos x$ 的值，则可变形为

$$1 - \cos x = 2\sin^2 \frac{x}{2}$$

或用 $\cos x$ 的泰勒(Taylor)展开式

$$1-\cos x=\frac{x^2}{2!}-\frac{x^4}{4!}+\cdots$$

来计算。

2. 调整运算次序,常可以防止大数“吃”小数现象。为了对位,计算机中常要进行小数形式 0.00153、0.153、15.3 等与形式 $1.53\times^{-3}$、1.53×10^{-1}、1.53×10^{1} 等的互相转换。这种允许小数点移位的表示方式称为数的浮点表示。若用五位浮点十进制数计算:

$$54321+0.4+0.3+0.3$$

计算机上先对位后运算,结果为

$$54321$$

小数被大数吃掉了,为避免这种现象,改用计算顺序由小到大

$$0.3+0.3+0.4+54321$$

则可得到精确结果 54322。

3. 减少运算次数,可以减少误差积累。因此,尽量寻找运算次数较少的算法,是算法设计中常要考虑的,例如,计算多项式

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

在某点的值,常采用

$$a_n+x(a_{n-1}+x(\cdots+x(a_1+x))\cdots)$$

由里向外计算的方式进行。注意,这样做不仅减少了误差积累,同时也减少了运算量。

1.3.3 误差分析方法

由误差与微分间关系,利用函数的全微分公式,可以进行误差分析。

例如,当计算函数 $p(x,y)=5x^3+4xy+7y^2$ 在 $x=1.414\pm0.5\times10^{-4}$,$y=2\pm0$ 处的函数值时,误差为:

$$|e(1.414,2)|=|dp(x,y)|_{(1.414,2)}=|(\frac{\partial p}{\partial x})_{(1.414,2)}||dx|+|(\frac{\partial p}{\partial y})_{(1.414,2)}||dy|$$

注意到 $dx=e(x)=0.5\times10^{-4}$、$dy=e(y)=0$ 并将各导数值求出,代入上式

$$|e(1.414,2)|\leqslant[15\times1.414^2=4\times2]\times0.5\times10^{-4}=1.899547\times10^{-3}$$

取小数点后 4 位小数计算结果为:

$$P(1.414,2)=53.4478$$

不难发现,上述误差估计并未考虑到计算过程中产生的舍入误差,这就必然使估计的误差偏小。当计算过程简单时,计算过程中产生的误差也许很小;但当计算过程复杂时,其计算过程中的舍入误差就会对结果影响很大,因此必须加以考虑。实际中还采用以下方法:

1. 向前误差分析

从初始数据的误差出发,随着计算过程把逐次发生的计算误差累积起来,获得关于计算结果的误差或误差界,这种方法称为向前误差分析法。

例如,计算 $x=\sum_{i=1}^{n}x_i$ 时,若每个 x_i 的误差 e_i 满足

$$|e_i|\leqslant0.5\times10^{-r}\quad(i=1,2,\cdots,n)$$

其中 r 为某一正整数，则和的误差

$$|e(x)| = |e_1 + e_2 + \cdots + e_n| \leqslant \frac{n}{2} \times 10^{-r}$$

这种方法实际中很难实施，且对于复杂的计算来说，分析过程也会很令人乏味。

2. 向后误差分析

为估计计算误差，假定初始值存在某一误差，并使这一误差的影响等效于计算过程中所产生计算误差的影响，并估计出误差的方法，称为向后误差分析法。

例如，解方程

$$ax = b \tag{1.4}$$

由于求解过程中发生了误差，最终使求得的解 $\bar{x}$ 产生了一个误差(也常称为扰动)$x^* - \bar{x}$(x^* 代表方程的精确解)，将其等效地看成仅由数据存在误差 δb 而引起的，方程求解过程本身不再产生误差，则 $\bar{x}$ 必为方程

$$ax = b + \delta b$$

的精确解，即

$$a\bar{x} = b + \delta b \tag{1.5}$$

将 x^* 代入(1.4)式减(1.5)式可得

$$x^* - \bar{x} = -\delta b/a$$

这种方法常用于解线性方程组的误差分析，也很有效。

3. 舍入误差的统计分析方法

向前误差分析方法，尽管计算起来有时过于乏味，但最终总能给出误差界。然而，问题却常会是误差界过于保守而失去估计的意义。究其原因则是忽视了带有不同符号的误差，在运算过程中相互抵消。

如何才能恰当估计出误差总积累的值？对于成千上万次的计算，计算的舍入误差会遵从统计规律。将计算的舍入误差看成随机变量，设其概率密度函数为 $p(x)$，则可以用数理统计的方法估计误差。以下通过例子说明误差估计的统计分析方法。

考虑 n 个数的相加 $x = \sum_{i=1}^{n} x_i$。用小数点后 r 位的浮点数表示每个 x_i，则 x_i 的舍入误差 e_i 相互独立，且

$$|e_i| \leqslant 0.5 \times 10^{-r}$$

e_i 在区间$[-0.5\times10^{-r}, 0.5\times10^{-r}]$内等可能地取每一个值。因此，$e_i$ 的分布密度函数 $p_i(x) = 10^r (i=1,2,\cdots,n)$，均值为零，方差 σ_i^2 为：

$$\int_{-\infty}^{\infty} x^2 p_i(x) dx = \int_{-e}^{e} x^2 p_i(x) dx = \frac{e^2}{3}$$

$$(i = 1, 2, \cdots, n)$$

其中 $e = 0.5\times10^{-r}$。n 个数相加的总体误差

$$e(x) = \sum_{i=1}^{n} e_i$$

方差为

$$\sigma^2 = \sum_{i=1}^{n} \sigma_i^2 = \frac{n}{3}e^2$$

当 n 充分大时，由概率论的中心极限定理，相加的总误差 $e(x)$ 服从均值为零，方差为 $\frac{n}{3}e^2$ 的正态分布，即 $e(x)$ 的分布密度函数为

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{x^2}{2\sigma^2}\}$$

其中 $\sigma^2 = \frac{n}{3}e^2$。令 $P(|e(x)| \leqslant \varepsilon) = \frac{1}{2}$，即使 $e(x)$ 取值于 $(-\varepsilon,\varepsilon)$ 的概率为 0.5，则有

$$\varepsilon = 0.6745\sigma = 0.3894e\sqrt{n}$$

这就是 $e(x)$ 的一个大致误差限，它与 $\sqrt{n}$ 成正比，与向前误差分析中估计的上限值 $\frac{n}{2} \times 10^{-r}$ 比较可知，n 较大时，统计方法估计出的误差限小得多。

用向前误差分析方法求得的误差限常与运算次数 n 成正比。实际应用的经验告诉我们，用 $\sqrt{n}$ 代替向前误差分析的误差限中 n，可以得到一个较可靠的误差限。

§1.4 算法的稳定性与收敛性

计算机上解数学问题常用逼近方法。当误差上限太大，不可接受时，往往采用修正目前估计值的方法加以改善，直到误差满足要求。这样就产生了一列逼近值 $a_1, a_2, \cdots, a_n, \cdots$。若当 n 趋于无穷大时 a_n 的极限为要逼近的真值 a^*，则称上述修正算法为收敛的。

1.4.1 收敛性

收敛的算法，使我们有了求出满足精度要求的解的希望。理论上讲，求出一个较好的近似解是不成问题的。但事实却并非如此。

例如，给定级数

$$1 - \sum_{i=1}^{n} \frac{2}{16i^2 - 1} \triangleq a_n$$

易知，

$$\lim_{n\to\infty} a_n = \frac{\pi}{4}$$

取前 n 项逼近时的余项为 $\frac{1}{4n+3}$。现我们用它来计算 π 的精确到 10^{-6} 的近似值，则令

$$\frac{4}{4n+3} \leqslant 10^{-6}$$

得到 n 至少为 10^6，也就是要计算约 10^6 项的和。这么多次运算后，舍入误差的总积累也许已超过了 10^{-6}。此算法用来计算 π 的值时，收敛太慢了，实际上根本不能用。

因此，实用中，不仅要用收敛的算法，而且要用收敛更快的算法。

1.4.2 稳定性

一个算法，若其对舍入误差不敏感时，称为稳定的。换句话说，算法的稳定性是指其计算结果受舍入误差影响的大小。受影响小或不受影响的算法为稳定的，否则为不稳定的。

此为试读，需要完整PDF请访问：www.ertongbook.com