

附录 B 数值计算中的误差

数值方法的特点之一就是所求得解是近似解,总是存在一定的误差.因此,误差分析是数值分析中一个很重要的课题.

误差是人们用来描述数值计算中近似解的精确程度,是科学计算中的一个十分重要的概念.

误差大致可分为以下几种类型:

- **模型误差**: 数学模型是对实际问题的数学描述,它往往是抓住问题的主要因素而略去次要因素,因此,它是实际问题的一个近似.
- **观测误差**: 在数学模型中通常包含一些参量(数据),这些参量的值一般都是通过测量或实验的方法所得到的,因此也存在误差.
- **截断误差**: 也称**方法误差**,在对数学模型进行数值求解时,需要做一些近似,如对导数离散时可用差商代替.
- **舍入误差**: 由于机器字长有限,由于机器字长有限,计算机对浮点数的表示和算术运算都存在一定的误差.

在数值分析中,我们总是假定数学模型和所给的数据都是准确的,因而不考虑模型误差和观测误差,主要研究截断误差和舍入误差对计算结果的影响.

例 B.1 近似计算 $\int_0^1 e^{-x^2} dx$ 的值.

解. 这里我们采用 Taylor 展开,即

$$\begin{aligned}\int_0^1 e^{-x^2} dx &= \int_0^1 \left(x - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \dots \right) \\ &= 1 - \frac{1}{3} + \frac{1}{2!} \times \frac{1}{5} - \frac{1}{3!} \times \frac{1}{7} + \frac{1}{4!} \times \frac{1}{9} - \dots \\ &\triangleq S_4 + R_4\end{aligned}$$

其中 S_4 为前四项的部分和, R_4 为剩余部分. 如果我们以 S_4 作为定积分的近似值, 则 R_4 就是由此产生的误差, 这种误差就称为截断误差.

在计算 S_4 的值, 假定我们保留小数点后 4 位有效数字, 则

$$S_4 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} \approx 1 - 0.3333 + 0.1000 - 0.0238 = 0.7429$$

这就是我们最后得到的近似值. 这里, 在计算 S_4 时所产生的误差就是舍入误差. \square

B.1 误差与有效数字

定义 B.1 设 x^* 是精确值, x 是其近似值, 则**绝对误差** e 和**相对误差** e_r 分别定义为

$$e = x - x^* \quad \text{和} \quad e_r = \frac{x - x^*}{x^*}.$$

若存在 $\varepsilon > 0$ 满足 $|e| = |x - x^*| \leq \varepsilon$, 则称 ε 为**绝对误差限**, 简称**误差限**. 类似地, 若存在 $\varepsilon_r > 0$ 满足 $|e_r| \leq \varepsilon_r$, 则称 ε_r 为**相对误差限**.

几点说明:

- 绝对误差可能为正, 也可能为负;
- 绝对误差越小越具有参考价, 但绝对误差却不能很好地表示近似值的精确程度;
- 近似值的精确程度取决于相对误差的大小;
- 实际计算中我们所能估计的通常是误差限或相对误差限;
- 由于真值难以求出, 通常也使用下面的定义作为相对误差限

$$e_r = \frac{x - x^*}{x}.$$

- 工程中通常用下面的表达式来刻画近似值的精度:

$$x^* = x \pm \varepsilon,$$

表示精确值在区间 $[x - \varepsilon, x + \varepsilon]$ 中.

定义 B.2 若近似值 x 的误差限是某一位的半个单位, 且该位到 x 的第一位非零数字共有 n 位, 则称 x 有 n 位**有效数字**.

关于有效数字的判断, 我们可以使用下面的方法.

定理 B.1 设 x 是 x^* 的近似值, 若 x 可表示为

$$x = \pm 0.a_1a_2 \dots a_n \dots \times 10^m,$$

其中 a_i 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若

$$|x - x^*| \leq 0.5 \times 10^{m-n},$$

则 x 至少有 n 位有效数字.

换言之, 若 $|x - x^*| \leq 0.5 \times 10^k$, 则 x 至少有 $m - k$ 位有效数字.

例 B.2 设 $x_1 = 3.1415$ 和 $x_2 = 3.1416$ 是 $\pi = 3.14159265\dots$ 的近似值, 则 x_1 有 4 位有效数字, 而 x_2 有 5 位有效数字.

例 B.3 根据四舍五入原则, 写出下列各数的具有 5 位有效数字的近似值:

$$187.9325, \quad 0.03785551, \quad 8.000033.$$

解. 这三个数的具有 5 位有效数字的近似值分别为: 187.93, 0.037856, 8.0000. □

有两点需要注意的是:



- 按四舍五入原则得到的数字是有效数字;
- 一个数末尾的 0 不可以随意添加或省略.

定理 B.2 (有效数字与相对误差限) 设 x 是 x^* 的近似值, 若 x 可表示为

$$x = \pm 0.a_1a_2 \dots a_n \dots \times 10^m,$$

其中 a_i 是 0 到 9 中的数字, 且 $a_1 \neq 0$. 若 x 具有 n 位有效数字, 则其相对误差限满足

$$\varepsilon_r \leq \frac{1}{2a_1} \times 10^{-n+1}.$$

反之, 若 x 的相对误差限满足

$$\varepsilon_r \leq \frac{1}{2(a_1 + 1)} \times 10^{-n+1}.$$

则 x 至少有 n 位有效数字.

证明. 由 x 的表达式可知

$$a_1 \times 10^{m-1} \leq |x| \leq (a_1 + 1) \times 10^{m-1}.$$

若 x 具有 n 位有效数字, 则

$$\varepsilon_r = \frac{|x - x^*|}{|x|} \leq \frac{0.5 \times 10^{m-n}}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \times 10^{-n+1}.$$

反之, 若 $\varepsilon_r \leq \frac{1}{2(a_1+1)} \times 10^{-n+1}$, 则

$$|x - x^*| = |x| \cdot \varepsilon_r \leq 0.5 \times 10^{m-n}.$$

故 x 至少有 n 位有效数字. □

从这个定理可以看出, 有效数字与相对误差是紧密相关的: 有效数字越多, 相对误差就越小; 反之, 相对误差越小, 有效数字就越多.

B.1.1 基本算术运算的误差估计

误差估计主要是指如何估计误差限或相对误差限. 我们用 $\varepsilon(x)$ 表示 x 的误差限, 则有

$$\begin{aligned} \varepsilon(x_1 \pm x_2) &\leq \varepsilon(x_1) + \varepsilon(x_2), \\ \varepsilon(x_1 x_2) &\leq |x_2| \varepsilon(x_1) + |x_1| \varepsilon(x_2), \\ \varepsilon\left(\frac{x_1}{x_2}\right) &\leq \frac{|x_1| \varepsilon(x_1) + |x_1| \varepsilon(x_2)}{|x_2|^2}. \end{aligned}$$

B.1.2 函数求值的误差估计

一般地, 设 $f(x)$ 是可微函数, x 为 x^* 的近似值, 则由 Taylor 公式可知, 存在 ξ 使得

$$f(x) - f(x^*) = f'(x^*)(x - x^*) + \frac{1}{2} f''(\xi)(x - x^*)^2.$$

所以有

$$\varepsilon(f(x)) \leq |f'(x^*)| \varepsilon(x) + \frac{|f''(\xi)|}{2} \varepsilon^2(x).$$



当 $|f''(\xi)|$ 与 $|f'(x^*)|$ 的比值不是很大时, 我们可以舍去二次项, 从而得到

$$\varepsilon(f(x)) \lesssim |f'(x^*)|\varepsilon(x).$$

由于 x^* 通常是不知道的, 所以我们也用 $f'(x)$ 来近似 $f'(x^*)$, 即

$$\varepsilon(f(x)) \lesssim |f'(x)|\varepsilon(x).$$

关于相对误差限, 我们有如下的估计:

$$\varepsilon_r(f(x)) = \left| \frac{f(x) - f(x^*)}{f(x^*)} \right| \approx \left| \frac{f'(x^*)(x - x^*)}{f(x^*)} \right| = \left| \frac{x^* f'(x^*)}{f(x^*)} \right| \cdot \left| \frac{x - x^*}{x^*} \right| = C_p \varepsilon_r(x),$$

其中 $C_p = \left| \frac{x^* f'(x^*)}{f(x^*)} \right|$ 称为 $f(x)$ 的 **条件数**.

对于多元可微函数 $f(x_1, x_2, \dots, x_n)$, 设 $x = (x_1, x_2, \dots, x_n)$ 是 $x = (x_1^*, x_2^*, \dots, x_n^*)$ 的近似值, 则有

$$\varepsilon(f(x)) \approx \sum_{k=1}^n \left| \frac{\partial f(x)}{\partial x_k^*} \right| \varepsilon(x_k).$$

B.2 误差分析

- 数值计算中的误差分析很重要, 但也很复杂;
- 在计算过程中, 误差会传播、积累、抵消;
- 实际计算中的运算次数通常都在千万次以上, 因此对每一步运算都做误差分析比较不切实际.

误差分析一般可分为定量分析和定性分析.

B.2.1 定量分析

- 主要方法有: 向前误差分析法, 向后误差分析法, 区间误差分析法, 概率分析法等.
- **向前误差分析**: 用输入数据的误差和数值方法本身的误差来分析计算结果的误差.
- **向后误差分析**: 用某个算法计算 $f(x)$, 得到的近似解为 \tilde{f} , 假定 \tilde{f} 是 $f(x)$ 对应于某个数据 \tilde{x} 的精确解, 即 $\tilde{f} = f(\tilde{x})$, 分析 $\tilde{x} - x$ 的大小就是向后误差分析.

向后误差分析法 (backward error analysis) 由著名数值分析专家 J. H. Wilkinson 于 1960 年提出, 这是误差理论中最基本的误差分析方法之一.

向后误差分析是一种 **先验误差估计** 方法, 不仅可以用来讨论算法的稳定性, 还可以用于讨论算法的收敛性.


后验误差估计 则是利用得到的数值结果来估计近似解的误差, 如在解方程组时, 可以利用残量来估计解的误差.

B.2.2 定性分析

- 目前在数值计算中更关注的是误差的定性分析;
- 定性分析包括研究数学问题的适定性, 数学问题与原问题的相容性, 数值算法的稳定性, 避免扩大误差的准则等;



- 定性分析的核心是原始数据的误差和计算过程中产生的误差对最终计算结果的影响.

 算法有“优劣”之分, 问题有“好坏”之别, 即使不能定量地估计出最终误差, 但是若能确保计算过程中误差不会被任意放大, 那就能放心地实施计算, 这就是研究定性分析的初衷.

B.3 数值稳定性


数值计算中的稳定性包括数学问题的稳定性和数值算法的稳定性.

B.3.1 数学问题的稳定性

如果数学问题满足

- (1) 对任意满足一定条件的输入数据, 存在一个解,
- (2) 对任意满足一定条件的输入数据, 解是唯一的,
- (3) 问题的解关于输入数据是连续的,

则称该数学问题是**适定的** (well-posed), 否则就称为**不适定的** (ill-posed).

 如果输入数据的微小扰动会引起输出数据 (即计算结果) 的很大变化 (误差), 则称该数值问题是**病态的**.

例 B.4 解线性方程组
$$\begin{cases} x + \alpha y = 1 \\ \alpha x + y = 0 \end{cases}$$

解. 易知当 $\alpha = 1$ 时, 方程组无解. 当 $\alpha \neq 1$ 时, 解为

$$x = \frac{1}{1 - \alpha^2}, \quad y = \frac{-\alpha}{1 - \alpha^2}.$$

当 $\alpha \approx 1$ 时, 解的误差可能会很大. 比如当 $\alpha = 0.999$ 时, $x \approx 500.25$. 假定输入数据 α 带有 0.0001 的误差, 即输入数据为 $\alpha^* = 0.9991$, 则此时有 $x^* \approx 555.81$, 解的误差约为 55.56, 是输入数据误差的五十多万倍, 因此该问题的病态的. □

B.3.2 病态问题与条件数

设 $f(x)$ 可导, 则其**条件数**定义为

$$C_p = \left| \frac{x f'(x)}{f(x)} \right|.$$

- 一般情况下, 条件数大于 10 时, 就认为问题是病态的;
- 条件数越大问题病态就越严重;
- 病态是问题本身固有的性质, 与数值算法无关;
- 对于病态问题, 选择数值算法时需要谨慎.



B.3.3 算法的稳定性

例 B.5 近似计算

$$S_n = \int_0^1 \frac{x^n}{x+5} dx, \quad n = 1, 2, \dots, 8.$$

解. 通过观察可知

$$S_n + 5S_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5} dx = \int_0^1 x^{n-1} = \frac{1}{n},$$

因此,

$$S_n = \frac{1}{n} - 5S_{n-1}. \quad (\text{B.1})$$

易知 $S_0 = \ln 6 - \ln 5 \approx 0.182$ (保留三位有效数字), 利用上面的递推公式可得 (保留三位有效数字)

$$S_1 = 0.0900, \quad S_2 = 0.0500, \quad S_3 = 0.0833, \quad S_4 = -0.166,$$

$$S_5 = 1.03, \quad S_6 = -4.98, \quad S_7 = 25.0, \quad S_8 = -125.$$

另一方面, 我们有

$$\frac{1}{6(n+1)} = \int_0^1 \frac{x^n}{6} dx \leq \int_0^1 \frac{x^n}{x+5} dx \leq \int_0^1 \frac{x^n}{5} dx = \frac{1}{5(n+1)}. \quad (\text{B.2})$$

因此, 上面计算的 S_4, \dots, S_8 显然是不对的. 原因是什么呢? 误差!

设 S_n^* 是 S_n 的近似值, 则

$$e(S_n^*) = S_n^* - S_n = \left(\frac{1}{n} - 5S_{n-1}^* \right) - \left(\frac{1}{n} - 5S_{n-1} \right) = -5(S_{n-1}^* - S_{n-1}) = -5e(S_{n-1}^*).$$

即误差是以 5 倍速度增长, 这说明计算过程是不稳定的, 因此我们不能使用该算法.

事实上, 递推公式 (B.1) 可以改写为

$$S_{n-1} = \frac{1}{5n} - \frac{1}{5}S_n.$$

因此, 我们可以先估计 S_8 的值, 然后通过反向递推, 得到其它值.

我们可以根据 (B.2) 对 S_8 做简单的估计, 即

$$S_8 \approx \frac{1}{2} \left(\int_0^1 \frac{x^8}{6} dx + \int_0^1 \frac{x^8}{5} dx \right) \approx 0.0204.$$

于是

$$S_7 = 0.0209, \quad S_6 = 0.0244, \quad S_5 = 0.0285, \quad S_4 = 0.0343,$$

$$S_3 = 0.0431, \quad S_2 = 0.0580, \quad S_1 = 0.0884, \quad S_0 = 0.182.$$

对比精确值 S_n^* 可知, 此时计算出来的 S_n 精度要好很多.

通过误差分析可知, 误差是以 $\frac{1}{5}$ 的速度减小, 因此计算过程是稳定的. \square

算法的稳定性: 通俗地讲, 对于某个给定的算法, 如果输入数据的误差在运算过程不断增长而得不到控制, 那么我们就说该算法是数值不稳定的, 否则就是数值稳定的.

假设输入数据的误差为 e_0 , 经 n 次运算后的计算结果的误差为 e_n . 如果 $e_n \approx c_1 n e_0$, 其中 c_1 是与 n 无关的常数, 则称误差是线性增长的. 如果 $e_n \approx c_2 k^n e_0$, 其中 c_2, k 是与 n 无关的常数且



$k > 1$, 则称误差是指数增长的. 如果算法的误差增长是线性的, 则该算法是数值稳定的, 如果算法的误差是指数增长的, 则该算法是数值不稳定的.

显然误差的线性增长是不可避免的, 而指数增长是必须避免的. 在数值计算中, 误差不可避免, 因此算法的稳定性是一个非常重要的性质.

在数值计算中, 不要采用不稳定的算法!

定义 B.3 (算法的向后稳定性) 设用某个算法来计算 $f(x)$, 得到的近似值为 $\tilde{f}(x)$. 若对任意的 x , 都存在一个“小”的 δx , 使得 $f(x + \delta x) = \tilde{f}(x)$, 则称该算法是**向后稳定的**, 其中 δx 称为**向后误差**. 这种误差分析方法就是**向后误差分析**.

若一个算法是向后稳定的, 则有

$$|\tilde{f}(x) - f(x)| = |f(x + \delta x) - f(x)| \approx |f'(x)| \cdot |\delta x|.$$

由于 δx 很小, 所以当 $|f'(x)|$ 不是很大时, 误差总是很小. 因此向后稳定是一个好的算法的基本性质.

向后误差分析将舍入误差归入到截断误差中, 使得误差分析相对简单化.

B.3.4 数值计算注意事项

在用计算机进行数值计算时, 舍入误差不可避免, 但我们要尽可能地减小舍入误差对计算结果的影响. 在计算过程中, 我们应注意以下几点.

- (1) **避免相近的数相减**. 如果两个相近的数相减, 则会损失有效数字, 如 $0.12346 - 0.12345 = 0.00001$, 操作数有 5 位有效数字, 但结果却只有 1 为有效数字. 下面给出几个避免相近的数相减的方法:

$$\begin{aligned}\sqrt{x + \varepsilon} - \sqrt{x} &= \frac{\varepsilon}{\sqrt{x + \varepsilon} + \sqrt{x}} \\ \ln(x + \varepsilon) - \ln(x) &= \ln\left(1 + \frac{\varepsilon}{x}\right) \\ 1 - \cos(x) &= 2 \sin^2 \frac{x}{2}, \quad |x| \ll 1 \\ e^x - 1 &= x \left(1 + \frac{1}{2}x + \frac{1}{6}x^2 + \cdots\right), \quad |x| \ll 1\end{aligned}$$

- (2) **避免数量级相差很大的数相除**. 可能会产生溢出, 即超出计算机所能表示的数的范围.
- (3) **避免大数吃小数**. 如直接计算 $(10^9 + 10^{-9} - 10^9)/10^{-9}$ 时, 结果可能为 0. 另外, 在对一组数求和时, 应按绝对值从小到大求和.
- (4) **简化计算**. 尽量减少运算次数, 避免误差积累.
- (5) **选用稳定的算法**. 尽可能避免由于算法本身导致的误差增大.



B.4 课后习题

实践题

练习 2.1 数值计算中的误差. 已知 $\sin(x)$ 的幂级数展开为

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

试利用该公式编程计算 $\sin(\pi/2)$ 和 $\sin(33\pi/2)$ 的值, 并与实际值做比较, 误差分别多大? 分析原因.

