

引论

数值计算的误差

内容提要

■ 误差

- 误差的来源
- 绝对误差与相对误差
- 误差限
- 有效数字
- 误差估计

■ 误差分析与数值稳定性

■ 数值计算中算法设计的技术

■ 数学软件（略）

什么是误差

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。

● 误差的来源

- 从实际问题中抽象出数学模型 —— **模型误差**
- 通过测量和实验得到模型中的各种数据 —— **观测误差**
- 数学模型的数值求解 —— **截断误差（方法误差）**
- 机器字长有限 —— **舍入误差**

在数值分析中，我们总假定数学模型是准确的，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。

误差举例

例：近似计算 $\int_0^1 e^{-x^2} dx$

解：将 e^{-x^2} 作Taylor展开后再积分

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

取 $\int_0^1 e^{-x^2} dx \approx S_4$

则 $R_4 = \frac{1}{4!} \times \frac{1}{9} - \frac{1}{5!} \times \frac{1}{11} + \dots$ 称为 截断误差

误差举例

$$\int_0^1 e^{-x^2} dx \approx S_4 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42}$$

保留小数点后 4 位数字

$$\approx 1 - \underline{0.3333} + 0.1000 - \underline{0.0238}$$

$$= 0.7429$$

舍入误差

绝对误差

定义： 设 x 为精确值， x^* 为它的一个近似值，则称

$$e^* = x^* - x$$

为近似值 x^* 的**绝对误差**，有时简称**误差**。

x	— 精确值
x^*	— 近似值

- 绝对误差可正可负
- 绝对误差通常是不可知的

定义： 存在一个正数 ε^* ，使得，

$$|e^*| = |x^* - x| \leq \varepsilon^*$$

则称 ε^* 为**绝对误差限**，简称**误差限**。记： $x = x^* \pm \varepsilon^*$

- 做误差估计时所求的是**绝对误差限**，越小越好！
- 但绝对误差限却不能很好地表示近似值的精确程度

相对误差

I can tell that distance between two planets is 1 million light year ± 1 light year.

I can tell that this part's diameter is $20\text{cm} \pm 0.1\text{cm}$.

Of course mine is more accurate ! The accuracy relates to not only the **absolute error**, but also to the size of the **exact value**



相对误差

定义： 设 x 为精确值， x^* 为它的一个近似值，则称

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

为近似值 x^* 的 **相对误差**。

- 由于精确值难以求出，通常也采用下面的定义

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

- 若存在正数 ε_r^* ，使得 $|e_r^*| \leq \varepsilon_r^*$ ，则称 ε_r^* 为 **相对误差限**

- 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得到的是 **绝对误差限** 或 **相对误差限**

有效数字

定义：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。

例： $\pi = 3.14159265 \cdots$ ，近似值

$$x_1 = 3.14, \quad x_2 = 3.1416, \quad x_3 = 3.1415$$

问： x_1, x_2, x_3 分别有几位有效数字？

(3, 5, 4)

例：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, 0.03785551, 8.000033

(187.93, 0.037856, 8.0000)

- 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

有效数字

- 另一个比较实用的描述

设 x^* 为 x 的近似值，若 x^* 可表示为

$$x^* = \pm 0.a_1a_2\cdots a_n \cdots \times 10^m$$

其中 a_i 是 0 到 9 中的数字且 $a_1 \neq 0$ ，且有

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

则 x^* 有 l 位有效数字。

有效数字与相对误差限

定理： 设近似值 x^* 可表示为

$$x^* = \pm a_1.a_2\cdots a_n \cdots \times 10^m \quad (a_1 \neq 0),$$

若 x^* 具有 n 位有效数字，则其相对误差限满足

$$\left| \frac{x^* - x}{x^*} \right| \leq \frac{1}{2a_1} \times 10^{-(n-1)}$$

反之，若

$$\left| \frac{x^* - x}{x^*} \right| \leq \frac{1}{2(a_1+1)} \times 10^{-(n-1)}$$

则 x^* 至少有 n 位有效数字。

- 有效数字越多，相对误差限越小

误差估计

- 误差估计：估计误差限或相对误差限

- 简单算术运算的误差估计

记 $\varepsilon(x^*)$ 为 x^* 的误差限，则有

$$\varepsilon(x_1^* \pm x_2^*) \leq \varepsilon(x_1^*) + \varepsilon(x_2^*)$$

$$\begin{aligned}\varepsilon(x_1^* x_2^*) &\leq |x_2^*| \varepsilon(x_1^*) + |x_1^*| \varepsilon(x_2^*) + \varepsilon(x_1^*) \varepsilon(x_2^*) \\ &\approx |x_2^*| \varepsilon(x_1^*) + |x_1^*| \varepsilon(x_2^*)\end{aligned}$$

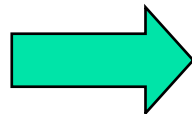
$$\varepsilon\left(\frac{x_1^*}{x_2^*}\right) \leq \frac{|x_2^*| \varepsilon(x_1^*) + |x_1^*| \varepsilon(x_2^*)}{|x_2^*|^2} + \dots \approx \frac{|x_2^*| \varepsilon(x_1^*) + |x_1^*| \varepsilon(x_2^*)}{|x_2^*|^2}$$

误差估计

- 一元可微函数 $f(x)$ 的误差估计

设一元函数 $f(x)$ 可微, x^* 为 x 的近似值, 则有

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


$$|f(x^*) - f(x)| \approx |f'(x^*)| |x^* - x|$$

$$\varepsilon(f(x^*)) \approx |f'(x^*)| \varepsilon(x^*)$$

内容提要

■ 误差

■ 误差分析与数值稳定性

- 误差分析方法
- 算法的数值稳定性
- 病态问题
- 避免误差危害

■ 数值计算中算法设计的技术

■ 数学软件（略）

误差分析

● 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、对消
- 对每一步运算都做误差分析比较不切实际
(运算次数通常都在千万次以上)

● 定量分析

- 向后误差分析法：比较有效的方法
- 向前误差分析法，区间误差分析法，概率分析法

定量分析工作量大，得到的误差界往往不太实用。
目前在数值计算中更关注的是误差的定性分析

误差分析

● 定性分析

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，但是若能判别计算过程中误差不会被任意放大，那就能放心地实施计算，这就是定性分析的初衷。

- 定性分析包括研究数值问题的**适定性**，数值问题与原问题的**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中产生的误差对最终计算结果的影响

数值稳定性

● 数学问题的适定性和数值算法的稳定性

● 数学问题的**适定性**（well-posedness）：满足

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

否则就称问题是**不适定**的（ill-posed）

通俗描述：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，否则就是**良态**的。

病态问题举例

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

解：当 $\alpha=1$ 时，无解

当 $\alpha \neq 1$ 时，解为 $x = \frac{1}{1-\alpha^2}, y = \frac{-\alpha}{1-\alpha^2}$

若 $\alpha \approx 1$ ，则 α 的微小误差可能会引起解的很大变化

比如 $\alpha=0.9990$ 时， $x \approx 500.25$;

如果输入数据有 0.0001 的误差，即 $\alpha^*=0.9991$,

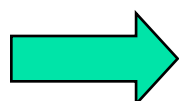

则 $x^* \approx 555.81$ ，误差约为 55.56!

因此，此时的问题就是病态的!

算法的稳定性

例：近似计算 $S_n = \int_0^1 \frac{x^n}{x+5} dx$ ，其中 $n=1, 2, \dots, 8$

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

 $S_n = \frac{1}{n} - 5S_{n-1}$  此公式精确成立

易知 $S_0 = \ln 6 - \ln 5 \approx 0.182$  保留 3 位有效数字

通过递推公式可得（每次都保留 3 位有效数字）

$$S_1 = 0.090, S_2 = 0.050, S_3 = 0.0833, S_4 \stackrel{?}{=} -0.166,$$
$$S_5 \stackrel{?}{=} 1.03, S_6 \stackrel{?}{=} -4.98, S_7 \stackrel{?}{=} 25.0, S_8 \stackrel{?}{=} -125.$$

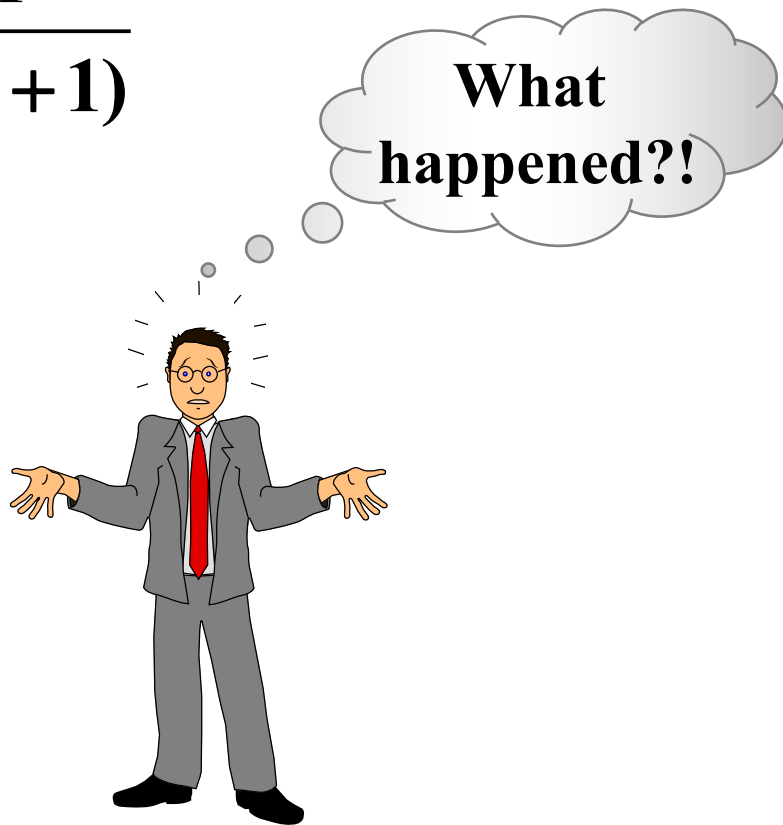
算法稳定性

但显然有

$$S_n = \int_0^1 \frac{x^n}{x+5} dx \leq \int_0^1 \frac{x^n}{5} dx = \frac{1}{5(n+1)}$$

$$S_n \geq \int_0^1 \frac{x^n}{6} dx = \frac{1}{6(n+1)}$$

$$\frac{1}{6(n+1)} \leq S_n \leq \frac{1}{5(n+1)}$$



算法稳定性

考察第 n 步的误差

$$e_n = S_n^* - S_n = \left(\frac{1}{n} - 5S_{n-1}^* \right) - \left(\frac{1}{n} - 5S_{n-1} \right) \approx -5(S_{n-1}^* - S_{n-1}) = -5e_{n-1}$$

即有

$$|e_n| = 5 |e_{n-1}| = 5^2 |e_{n-2}| = \cdots = 5^n |e_0|$$

误差以 **5 倍** 的速度增长！

说明该计算过程是 **不稳定** 的！

我们需要改变算法！


数值稳定性

$$\frac{1}{6(n+1)} \leq S_n \leq \frac{1}{5(n+1)}$$

解法二：

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

具体思路：先估计一个 S_N ，再反过来求 S_n ($n < N$)

$$S_8 \approx \frac{1}{2} \left(\frac{1}{5(8+1)} + \frac{1}{6(8+1)} \right) \approx 0.0204$$


$$S_1 = 0.0884, S_2 = 0.0580, S_3 = 0.0431, S_4 = 0.0343, \\ S_5 = 0.0285, S_6 = 0.0244, S_7 = 0.0209, S_8 = 0.0204.$$

[demo_1_2.m](#)

在数值计算中，误差不可避免，
算法的稳定性是一个非常重要的性质。

数值稳定性

- 算法的稳定性

在计算过程中，如果误差不增长或能得到有效控制，则称该算法是稳定的，否则为不稳定的。

- 数值计算中，不要采用不稳定的算法！

数值计算注意事项

- 避免相近的数相减

例： $a_1 = 0.12345$, $a_2 = 0.12346$, 各有5位有效数字。
而 $a_2 - a_1 = 0.00001$, 只剩下1位有效数字。

- 几种经验性避免误差危害的方法：

$$\sqrt{x + \varepsilon} - \sqrt{x} = \frac{\varepsilon}{\sqrt{x + \varepsilon} + \sqrt{x}}$$

$$\ln(x + \varepsilon) - \ln x = \ln\left(1 + \frac{\varepsilon}{x}\right)$$

当 $|x| \ll 1$ 时：

$$1 - \cos x = 2 \sin^2 \frac{x}{2}$$
$$e^x - 1 = x \left(1 + \frac{1}{2}x + \frac{1}{6}x^2 + \dots \right)$$

数值计算注意事项

- 避免数量级相差很大的数相除

可能会产生溢出的情形（超出计算机所能表示的范围）

- 避免大数吃小数

例：计算 $(10^9 + 10^{-9} - 10^9) / 10^{-9}$

[demo_1_3.m](#)

例：按从小到大、以及从大到小的顺序分别计算

$$S = 1 + 2 + 3 + \dots + 40 + 10^8$$

求和时 **从小到大** 相加，可使结果的误差减小

数值计算注意事项

- 简化计算，避免误差积累

例：已知 $p(x) = x^n + x^{n-1} + \cdots + x + 1$ ，
计算 $n = 20$ 时， $p(8)$ 的值。

如果直接代入计算，则需 $n(n-1)/2$ 次乘法和 n 次加法运算

如果将 $p(x)$ 改写为：

$$p(x) = (\cdots(((x + 1)x + 1)x + 1)\cdots)x + 1$$

[demo_1_1.m](#)

则只需 $n - 1$ 次乘法和 n 次加法运算。



秦九韶算法 或 Horner 算法

- 选用稳定的算法

内容提要

- 什么是误差
- 误差分析与数值稳定性
- 数值计算中算法设计的技术
- 数学软件（略）

算法设计

- 多项式计算：秦九韶算法 或 Horner算法
- 迭代法与开方求值：如非线性方程的迭代法解法
- 以直代曲与化整为零：如差商代替微商
- 加权平均的松弛技术：如 Simpson 算法