第一章　　数值计算中的误差

§1.1　误差的来源

§1.2　绝对误差、相对误差和有效数字

§1.3　数值计算中误差的传播

§1.4　数值计算中应注意的几个问题

§1.1　误差的来源

- 误差按来源可分为：
  - 模型误差
  - 观测误差
  - 截断误差
  - 舍入误差

- **模型误差** 数学模型通常是由实际问题抽象得到的，一般带有误差，这种误差称为模型误差.

- **观测误差** 数学模型中包含的一些物理参数通常是通过观测和实验得到的，难免带有误差，这种误差称为观测误差.

- **截断误差** 求解数学模型所用的数值方法通常是一种近似方法，这种因方法产生的误差称为截断误差或方法误差.

例如，利用 $\ln(x+1)$ 的Taylor公式计算 $\ln 2$ ，

$$\ln(x+1) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 - \cdots$$

实际计算时只能截取有限项代数和计算，如取前5项有：

$$\ln 2 \approx 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5}$$

这里产生误差 (记作R_5) 截断误差

$$\boxed{R_5 = -\frac{1}{6} + \frac{1}{7} - \frac{1}{8} + \cdots}$$

舍入误差

由于计算机只能对有限位数进行运算，在运算中像 $\sqrt{2}$, e, $1/3$ 等都要按舍入原则保留有限位，这时产生的误差称为舍入误差。

在数值分析中，均假定数学模型是准确的，因而不考虑模型误差和观测误差，只讨论截断误差和舍入误差对计算结果的影响.

§1.2 绝对误差、相对误差和有效数字

➢ 绝对误差

⊗设 x^* 是准确值 x 的一个近似值，记

$$e = x - x^*$$

称 e 为近似值 x^* 的**绝对误差**，简称误差.

绝对误差一般很难准确计算，但可以估计上界.

⊗若 ε 满足 $|e| \le \varepsilon$

则称 ε 为近似值 x^* 的**绝对误差限**，简称误差限.

$\varepsilon > 0$ 不唯一，当然 ε 越小越具有参考价值.

例 用毫米刻度的尺子测量一长度 x, 如读出的长度是 x^* =765 mm,

由于误差限是 0.5 mm，故准确值

$$x \in [764.5\ \mathrm{mm},\quad 765.5\ \mathrm{mm}].$$

- 精确值 x , 近似值 x^* 和误差限 ε 之间满足：

$$x^* - \varepsilon \le x \le x^* + \varepsilon$$

通常记为

$$\boxed{x = x^* \pm \varepsilon}$$

- 绝对误差有时并不能完全地反映近似值的好坏，如测量 100 m 和 10 m 两个长度，若它们的绝对误差都是 1 cm，显然前者的测量结果比后者的准确.

- 因此，决定一个量的近似值的精确度，除了要看绝对误差外，还必须考虑该量本身的大小.

相对误差

- 记 $e_r = \dfrac{e}{x} = \dfrac{x - x^*}{x}$，称 e_r 为近似值 x^*的相对误差.

由于 x 未知，实际使用时总是将 x^*的相对误差取为

$$\boxed{e_r \approx \frac{e}{x^*} = \frac{x - x^*}{x^*}}$$

- $\boxed{\varepsilon_r = \varepsilon / | x^* |}$ 称为近似值 x^*的相对误差限.

$$| e_r | \le \varepsilon_r .$$

例　设 x^*=1.24是由精确值 x 经过四舍五入得到的近似值, 求 x^*的绝对误差限和相对误差限.

解　由已知可得: $1.235 \le x < 1.245$

所以　$\varepsilon = 0.005$,

$$\varepsilon_r = 0.005/1.24 \approx 0.4\%.$$

● 一般地, 凡是由准确值经过四舍五入得到的近似值，其绝对误差限等于该近似值末位的半个单位.

● 若近似值 x^* 满足 $\boxed{|x-x^*|\le\frac{1}{2}\times10^{-n},}$ 则称 x^* 准确到小数点后第n位.
并把从第一个非零数字到这一位的所有数字均称为有效数字.

例 $\pi=3.14159265358979 32......;\quad \pi^*=3.1415$

问: π^*有几位有效数字?

解: $\because|\pi^*-\pi|<0.5\times10^{-3}$

$\therefore\ \pi^*$ 有4位有效数字，精确到小数点后第3位

有效数字的另一定义

数x^*总可以写成如下形式

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

其中m是整数, a_i 是0到9中的一个数字, $a_1 \neq 0.$

x^* 作为x的近似值, 具有n位有效数字当且仅当

$$\boxed{\left|x^* - x\right| \leq \frac{1}{2} \times 10^{m-n}}$$

由此可见, 近似值的有效数字越多, 其绝对误差越小.

例 为了使 $x=\sqrt{2}$ 的近似值的绝对误差小于10^{-5}，问应取几位有效数字?

解 由于$\sqrt{2}=1.414\cdots$，则近似值x^*可写为

$$x^*=0.a_1a_2\cdots a_n\times 10^1,\quad a_1=1\neq 0.$$

令 $$\left|\sqrt{2}-x^*\right|\leq\frac{1}{2}\times 10^{1-n}\leq 10^{-5}$$

故取 $n=6$，即取 6 位有效数字．此时 $x^*=1.41421$.

- 相对误差限与有效数字之间的关系.

☞ 有效数字 $\Rightarrow$ 相对误差限

已知 $x^* = \pm 0.a_1a_2\ldots a_n \times 10^m$ 有n位有效数字，则其相对误差限为

$$\varepsilon_r = \left|\frac{\varepsilon}{x^*}\right| = \frac{0.5\times 10^{m-n}}{0.a_1a_2\ldots a_n \times 10^m} = \frac{10^{-n}}{2\times 0.a_1\ldots} \le \frac{1}{2a_1}\times 10^{-n+1}$$

☞ 相对误差限 $\Rightarrow$ 有效数字

已知 x^* 的相对误差限可写为 $\varepsilon_r = \dfrac{1}{2(a_1+1)} \times 10^{-n+1}$，则

$$|x - x^*| \le \varepsilon_r \cdot |x^*| = \frac{10^{-n+1}}{2(a_1+1)} \times 0.a_1a_2\ldots \times 10^m$$

$$< \frac{10^{-n+1}}{2(a_1+1)} \cdot (a_1+1) \times 10^{m-1} = 0.5 \times 10^{m-n}$$

可见 x^* 至少有 n 位有效数字.

➢ 基本运算中(+ − × ÷)的误差估计

如 $|\sqrt{2}-1.414|\le 0.5\times10^{-3}$,

$|\sqrt{5}-2.236|\le 0.5\times10^{-3}$,

问 $|\sqrt{5}\times\sqrt{2}-2.236\times1.414|\le ?$

$$\left|\frac{\sqrt{2}}{\sqrt{5}}-\frac{1.414}{2.236}\right|\le ?$$

事实上, 对于$A=f(x_1, x_2)$，如果 x_1, x_2 的近似值为 x_1^*, x_2^*, 则A 的近似值为 $A^*=f(x_1^*, x_2^*)$, 用多元函数微分近似公式可以得到

$$
\begin{aligned}
e(A^*) &= A - A^* = f(x_1, x_2) - f(x_1^*, x_2^*) \\
&\approx \frac{\partial f(x_1^*, x_2^*)}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f(x_1^*, x_2^*)}{\partial x_2}(x_2 - x_2^*) \\
&= \frac{\partial f(x_1^*, x_2^*)}{\partial x_1} e(x_1^*) + \frac{\partial f(x_1^*, x_2^*)}{\partial x_2} e(x_2^*)
\end{aligned}
$$

- 绝对误差 e 运算可近似看成微分运算.

由此可以得到基本运算中 (+ − × ÷) 的误差估计,

$$e(x_1 \pm x_2) = e(x_1) \pm e(x_2),$$

$$\boxed{|e(x_1 \pm x_2)| \leq |e(x_1)| + |e(x_2)|}$$

- 和差的误差限不超过各数的误差限之和.

$$e(x_1x_2) = x_2e(x_1) + x_1e(x_2),$$

$$e_r(x_1x_2) = \frac{x_2e(x_1)}{x_1x_2} + \frac{x_1e(x_2)}{x_1x_2} = e_r(x_1) + e_r(x_2)$$

$$\boxed{|e_r(x_1x_2)| \le |e_r(x_1)| + |e_r(x_2)|}$$

- 乘法相对误差限不超过各数相对误差限之和.

$$e\left(\frac{x_1}{x_2}\right)=\frac{x_2 e(x_1)-x_1 e(x_2)}{x_2^2},$$

$$e_r\left(\frac{x_1}{x_2}\right)=\frac{x_2 e(x_1)-x_1 e(x_2)}{x_2^2}\cdot\frac{x_2}{x_1}=e_r(x_1)-e_r(x_2).$$

$$\boxed{\left|e_r\left(\frac{x_1}{x_2}\right)\right|\le|e_r(x_1)|+|e_r(x_2)|.}$$

- 除法相对误差限不超过各数相对误差限之和.

例 设 $y=x^n$, 求 y 的相对误差与 x 的相对误差之间的关系.

解 $$e(y)=e(x^n)=nx^{n-1}e(x)$$

$$e_r(y)=\frac{e(y)}{y}=\frac{nx^{n-1}e(x)}{x^n}=n\frac{e(x)}{x}=ne_r(x)$$

所以x^n 的相对误差是 x 的相对误差的n倍.

x^2的相对误差是 x 的相对误差的 2 倍,

$\sqrt{x}$的相对误差是 x 的相对误差的 1/2 倍.

算法的数值稳定性

- 一种数值算法，如果其计算舍入误差积累是可控制的，则称其为数值稳定的，反之称为数值不稳定的.

例 计算积分 $I_n = \int_0^1 x^n e^{x-1} \mathrm{d}x$

解 利用分部积分法可得计算 I_n 的递推公式

$$\boxed{I_n = 1 - nI_{n-1}, \quad n = 1,2,\cdots}$$

算法1： $I_n = 1 - nI_{n-1}, \quad n = 1,2,\cdots$

$$I_0 = \int_0^1 e^{x-1} \mathrm{d}x = 1 - e^{-1} = 0.632120558\cdots \approx 0.6321$$

由此递推计算 $I_1, I_2, \ldots, I_9$.

算法2： $\frac{e^{-1}}{10} \le \int_0^1 x^9 e^{-1} dx \le I_9 \le \int_0^1 x^9 dx \le \frac{1}{10}$

取近似值 $I_9 \approx \frac{1}{2}\left(\frac{e^{-1}}{10} + \frac{1}{10}\right) \approx 0.0684,$

此时 $|I_9 - I_9^*| \le \frac{1}{2} \cdot \frac{1-e^{-1}}{10} \approx 0.0316.$

并将计算公式改写为

$$I_{n-1} = \frac{1}{n}(1 - I_n), \quad n = 9,8,\cdots,2,1$$

由此计算 $I_8, I_7, \ldots, I_0.$

I_n	算法1	算法2	真值
I_0	0.6321	0.6321	0.6321
I_1	0.3679	0.3679	0.3679
I_2	0.2642	0.2642	0.2642
I_3	0.2074	0.2073	0.2073
I_4	0.1704	0.1709	0.1709
I_5	0.1480	0.1455	0.1455
I_6	0.1120	0.1268	0.1268
I_7	0.2160	0.1121	0.1124
I_8	- 0.7280	0.1035	0.1009
I_9	7.5520	0.0684	0.0916

- 对任何 n都应有$I_n>0$, 但算法1的计算结果显示$I_8<0$, 可见, 虽然I_0的近似误差不超过0.5×10^{-4}, 但随着计算步数的增加，误差明显增大. 这说明算法1给出的递推公式是数值不稳定的.

- 而对于算法2，虽然初始给出的 I_9没有一位有效数字，但算至 I_6已有4位有效数字．这说明算法2中误差随着计算过程的深入是逐步递减的，因而是数值稳定的.

例 计算积分 $I_n = \int_0^1 x^n e^{x-1} \mathrm{d}x$

对于算法1： $I_n = 1 - nI_{n-1}, \quad n = 1,2,\cdots$

$$I_0 = \int_0^1 e^{x-1} \mathrm{d}x = 1 - e^{-1} \approx 0.6321 = I_0 *$$

由 $I_n = 1 - nI_{n-1}$ 和 $I_n^* = 1 - nI_{n-1}^*$， $n = 1,2,\cdots$

可得 $\boxed{I_n - I_n^*} = -n(I_{n-1} - I_{n-1}^*) = \cdots = \boxed{(-1)^n n!(I_0 - I_0^*)}.$

$$\boxed{| I_9 - I_9 * | = 9! | I_0 - I_0 * | = 362880 | I_0 - I_0 * |}$$

可见, 随着计算步数的增加, 误差迅速放大， 使结果失真.

例　计算积分 $I_n = \int_0^1 x^n e^{x-1} \mathrm{d}x$

$$I_n = 1 - nI_{n-1}, \quad n = 1,2,\cdots$$

算法2的计算公式为

$$I_{n-1} = \frac{1}{n}(1 - I_n), \quad n = k, k-1, \cdots, 2, 1$$

类似地可得

$$\boxed{I_n - I_n^* = (-1)^{k-n}\frac{n!}{k!}(I_k - I_k^*), \quad n = k, k-1, \cdots, 1, 0}$$

$$\boxed{|I_0 - I_0^*| = \frac{1}{9!}|I_9 - I_9^*| = \frac{1}{362880}|I_9 - I_9^*|}$$

可见，近似误差 $I_n - I_n^*$ 是可控制的，算法是数值稳定的.

§4 数值计算中应注意的问题

为了减少舍入误差的影响，设计算法时应遵循如下的一些原则.

➢ 1. 避免两个相近的数相减

如果 x, y 的近似值分别为x^*, y^*，则 $z^*=x^*-y^*$ 是 $z=x-y$ 的近似值.

此时，相对误差满足估计式

$$\left|e_r(z^*)\right| = \left|\frac{e(x^*-y^*)}{x^*-y^*}\right|,$$

可见，当x^*与 y^*很接近时，z^*的相对误差有可能很大.

- 在数值计算中，如果遇到两个相近的数相减，可考虑改变一下算法以避免两数相减.
- 例如

当$x_1 \approx x_2$时， $\log x_1 - \log x_2 = \log \frac{x_1}{x_2}$

当$x \approx 0$时， $1 - \cos x = 2\sin^2 \frac{x}{2}$

当$x >> 1$时， $\sqrt{x+1} - \sqrt{x} = \frac{1}{\sqrt{x+1} + \sqrt{x}}$

例 求方程 $x^2-64x+1=0$的两个根，使它们至少具有四位有效数字.

$$(\sqrt{1023}\approx 31.984)$$

解 由求根公式有 $x_1=32+\sqrt{1023}\approx 63.984$

若由 $x_2=32-\sqrt{1023}\approx 0.016$, 仅有两位有效数字,

但若采用 $x_2=1/x_1\approx 0.01563$, 则有四位有效数字.

- 对两个相近的数相减，若找不到适当方法代替，只能在计算机上采用双精度进行计算，以提高精度.

➢ 2. 防止大数“吃掉”小数

- 因为计算机上只能采用有限位数计算, 若参加运算的数量级差很大, 在它们的加、减运算中，绝对值很小的数往往被绝对值较大的数“吃掉”，造成计算结果失真.

- 在求和或差的过程中应采用由小到大的运算过程.

➢ 3. 绝对值太小的数不宜作除数

由于除数很小,将导致商很大, 有可能出现“溢出”现象．另外，设x , y 的近似值分别为x^* , y^*,则$z^*=x^*/y^*$是$z=x/y$的近似值．此时z^*的绝对误差满足估计式

$$|e(z)| = \left|\frac{ye(x)-xe(y)}{y^2}\right| \le \frac{|y||e(x)|+|x||e(y)|}{y^2}$$

● 可见，若除数太小, 则可能导致商的绝对误差很大.

➢ 4. 注意简化计算程序，减少计算次数

- 首先，若算法计算量太大，实际计算无法完成

例　用Cramer法则求n阶线性方程组$Ax=b$的解，用n阶行列式定义来计算

$$\text{总乘法运算次数} > (n-1)(n+1)!$$

当$n=25$时，在每秒百亿次乘除运算计算机上求解时间为

$$\frac{24*26!}{10^{10}\times 3600\times 24\times 365} \approx 307\ (\text{亿年})$$

● 其次，即使是可行算法，则计算量越大积累的误差也越大. 因此，算法的计算量越小越好.

例 计算n次多项式：

$$p_n(x)=a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$$

若直接逐项计算，大约需要乘法运算次数为

$$n+(n-1)+\ldots+2+1=\frac{n(n+1)}{2}$$

如果利用分配律，则能使计算量大为降低

$$P_2(x)=a_2x^2+a_1x+a_0$$
$$=(a_2\cdot x+a_1)\cdot x+a_0,$$

2次乘法+2次加法

$$P_3(x)=a_3x^3+a_2x^2+a_1x+a_0$$
$$=(a_3x^2+a_2x+a_1)x+a_0$$
$$=((a_3\cdot x+a_2)\cdot x+a_1)\cdot x+a_0$$

3次乘法+3次加法

一般地，对于n次多项式将它改写为

$$\boxed{p_n(x)=(\cdots((a_n\cdot x+a_{n-1})\cdot x+a_{n-2})\cdot x+\cdots)\cdot x+a_0}$$

则只需n次乘法和n次加法运算.

5. 要关注计算效率，注意收敛速度

例：用下述级数计算 $\ln 2$，且要求误差小于 10^{-7}.

$$\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \ldots + (-1)^{n-1}\frac{1}{n} - \ldots$$

解: 用计算公式 $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots + \frac{(-1)^{n-1}x^n}{n}$

$$\ln 2 \approx S_n = 1 - \frac{1}{2} + \frac{1}{3} - \ldots + (-1)^{n-1}\frac{1}{n}$$

误差 $|\ln 2 - S_n| \le \frac{1}{n+1} < 10^{-7}$， 得 $n > 10^7 - 1$.

需前一千万项求和!

如果用级数

$$\ln\frac{1+x}{1-x}=2x\left(1+\frac{1}{3}x^2+\frac{1}{5}x^4+\ldots+\frac{1}{2m+1}x^{2m}+\ldots\right),$$

$$|x|<1$$

取 $x=\frac{1}{3}$ 来计算 $\ln 2$，用前9项求和(取 $m=8$) 就达到精度要求，即

$$\ln 2\approx S_8=2x\left(1+\frac{1}{3}x^2+\frac{1}{5}x^4+\ldots+\frac{1}{17}x^{16}\right),\qquad x=\frac{1}{3}$$

则误差 $|\ln 2-S_8|<10^{-7}$.

➢ 6.选用数值稳定性好的算法

在数值计算方法中, 一般注重误差的定性分析, 即只讨论各种算法的数值稳定性, 只要算法是数值稳定的, 就不再进行舍入误差估计; 截断误差将针对各种问题结合具体算法讨论.

知识结构图二

误差

- 分类
  - 模型误差、观测误差
  - 截断误差、舍入误差
- 度量
  - 绝对误差（限）
  - 相对误差（限）
  - 有效数字
  - 三者的联系
- 传播
  - 基本运算的误差估计
  - 算法的稳定性
- 注意
  - 避免误差扩大
  - 提高算法效率
  - 注重算法稳定性

- 作业 P13: 4， 7, 9, 10, 12

- 数值实验 P14: 2， 3