第1章　误差理论

陈路　　CHENLU.SCIEN@GMAIL.COM
201800301206

本文首先简要总结误差理论的相关知识，进而以二次方程求根问题为例讨论减少误差的具体方法，并以幂次运算问题为例讨论减少运算次数的具体策略。

1. 误差理论知识总结

1.1 误差的来源及类型

计算机在计算过程中的误差主要来自于计算机存储空间的有限性和浮点数运算过程中的精度丢失，具体来说，由于计算机硬件只支持有限位机器数的运算，导致误差在运算中被引入和传播。误差大致可以分为以下四类：

1. **模型误差**　用计算机解决科学计算问题首先要建立数学模型，它是对被描述的实际问题进行抽象、简化而得到的，因而是近似的.我们把数学模型与实际问题之间出现的这种误差称为模型误差.
2. **截断误差**　通常指用一个基本表达式替换一个相当复杂的算术表达式时所引入的误差。该术语从用截断泰勒级数替换一个复杂表达式的技术中衍生而来。
3. **舍入误差**　计算机表示的实数受限于尾数的固定精度，因此有时并不能确切地表示真实值，这一类型的误差称为舍入误差。
4. **浮点运算舍入误差**

1.2 误差的度量方法

误差的测度可以分为绝对误差和相对误差，绝对误差仅仅是真实值与近似值之间的差，而相对误差很大程度上取决于真实值。定义如下：

定义 1
设$\hat{p}$是p的近似值，则**绝对误差**是$E_p = |p - \hat{p}|$，**相对误差**是$R_p = |p - \hat{p}|/|p|$，其中$p \neq 0$.

当$|p|$远离1时（大于或小于），相对误差R_p比绝对误差E_p能更好地表示近似值的精确程度。由于相对误差直接处理尾数，所以浮点表示主要采用相对误差。

1.3 序列的收敛性与收敛阶

定义 2

设$\lim_{n\to\infty} x_n = x$，有序列$\{r_n\}_{n=1}^{\infty}$，且$\lim_{n\to\infty} r_n = 0$。如果存在常量$K > 0$，满足

$$\frac{|x_n - x|}{|r_n|} \leqslant K, \quad n \text{ 足够大}$$

则称$\{x_n\}_{n=1}^{\infty}$以**收敛阶**$O(r_n)$收敛于x.

1.4 误差的传播

考虑数p和q（真实值），它们的近似值分别为$\hat{p}$和$\hat{q}$，误差分别为ϵ_p和ϵ_q，即$p = \hat{p}+\epsilon_p$，$q = \hat{q}+\epsilon_q$。分别观察在加法、乘法运算中误差的传播：

- **加法** $p + q = (\widehat{p} + \epsilon_p) + (\widehat{q} + \epsilon_q) = (\widehat{p} + \widehat{q}) + (\epsilon_p + \epsilon_q)$
- **乘法** $pq = (\widehat{p} + \epsilon_p)(\widehat{q} + \epsilon_q) = \widehat{pq} + \widehat{p}\epsilon_q + \widehat{q}\epsilon_p + \epsilon_p\epsilon_q$

可见，对于加法运算，和的误差是每个加数误差的和；而在乘法运算中，若$\hat{p}$和$\hat{q}$的绝对值大于1，那么原误差ϵ_p和ϵ_q将被放大为$\hat{p}\epsilon_p$和$\hat{q}\epsilon_q$；进一步观察其相对误差：

$$R_{pq} = \frac{pq - \widehat{pq}}{pq} = \frac{\widehat{p}\epsilon_q + \widehat{q}\epsilon_p + \epsilon_p\epsilon_q}{pq} = \frac{\widehat{p}\epsilon_q}{pq} + \frac{\widehat{q}\epsilon_p}{pq} + \frac{\epsilon_p\epsilon_q}{pq}$$

这表明p和q乘积的相对误差大致等于$\hat{p}$和$\hat{q}$相对误差的和.

定义 3

设ϵ表示初始误差，$\epsilon(n)$表示第n步计算后的误差增长. 如果$|\epsilon(n)| \approx n\epsilon$，则称误差线性增长. 如果$|\epsilon(n)| \approx K^n\epsilon$，则称误差呈指数增长. 若$K > 1$，则当$n \to \infty$时，指数误差的增长无界；若$0 < K < 1$，则当$n \to \infty$时，指数误差的增长趋于0.

2. 分析讨论题

2.1 二次方程求根

问题 1

求方程$x^2 + (\alpha + \beta)x + 10^9 = 0$的根，其中，$\alpha = -10^9, \beta = -1$，讨论如何设计计算格式才能有效地减少误差，提高计算精度.

解：我们已知，方程$ax^2 + bx + c = 0$（其中$a \neq 0, b^2 - 4ac > 0$）的根可以通过以下求根公式获得：

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{1}$$

对上式进行分子有理化，得到等价公式：

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}, \quad x_2 = \frac{-2c}{b - \sqrt{b^2 - 4ac}} \tag{2}$$

当$|b| \approx \sqrt{b^2-4ac}$时，为避免其值过小引起巨量消失（catastrophic cancellation）而造成精度损失，在求解时可以进行以下处理：

- 如果$b>0$，用公式(2)计算x_1，用公式(1)计算x_2
- 如果$b<0$，用公式(1)计算x_1，用公式(2)计算x_2

求解程序如下：

```
function [x1,x2] = solveQuadEq(a,b,c)
% Solve the roots of a quadratic equation
% Input - a,b,c    coefficients of the equation
% Output - x1,x2   roots of the equation

% Initialization
x1 = 0;
x2 = 0;
delta = b^2 - 4*a*c;

% Compute the roots
if delta>=0
  if b>0
    x1 = (-2*c)/(b+sqrt(delta));
    x2 = (-b-sqrt(delta))/2*a;
  else
    x1 = (-b+sqrt(delta))/2*a;
    x2 = (-2*c)/(b-sqrt(delta));
  end
end
end
```

2.2 幂次运算

问题 2
以计算x^{31}为例，讨论如何设计计算格式才能减少计算次数.

解：观察到$31=16+8+4+2+1$，采用以下“二次累积”计算策略

Step1: 计算$x\cdot x$得x^2

Step2: 计算$x^2\cdot x^2$得x^4

Step3: 计算$x^4\cdot x^4$得到 x^8

Step4: 计算$x^8\cdot x^8$得到 x^{16}

Step5: 将前几步的所有结果相乘，得：

$$x\cdot x^2\cdot x^4\cdot x^8\cdot x^{16}=x^{31}$$