数值分析

夏银华

中国科学技术大学

误差类型

- 舍入误差 (round-off error)：由于计算机只能表示有限位数的数字
- 截断误差 (Truncation error)：由于数值算法对数学运算或数量的近似
- 其它误差：模型误差，数据／参数输入误差，人为误差...

舍入误差

- 有效数字 (significant digits)：能够被相信的数字
- 舍入误差：（计算机中）被忽略的有效数字
- 精度：近似值 $\hat{x}$与真值 x间的差距
- （准确）绝对误差：$\epsilon = \hat{x} - x$
- （准确）相对误差：$\alpha = \frac{\hat{x}-x}{x}$

（计算机）数字表示

数字系统

- 十进制：$1,234_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$
- 二进制：$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$

整数表示

- 首“位” (first bit) 是正负号，剩余位存储数字
- 例如在16位计算机中，$-13_{10} = 1000000000001101$，最大能表示的数 $2^{15} - 1$

（计算机）数字表示

浮点数(floating number)表示

- $x = mb^e$,
  m: 小数部分(Mantissa/Significand)
  b: 基(Base)
  e: 指数(Exponent)
- 例如: (规范化小数 $b^{-1} \leq m < b^0$)
  十进位： $0.00527_{10} = 0.527_{10} \times 10^{-2_{10}}$
  二进位： $10.1_2 = 0.101_2 \times 2^{2_{10}} = 0.101_2 \times 2^{10_2}$

（计算机）数字表示

例：

- 考虑二进制7位浮点计算机
  7位 = 1位符号 + 3位指数部分（1位符号两位数字） + 三位小数部分，
  问计算机能表示的最大最小正数是?

例：

- 考虑二进制7位浮点计算机
  7位 = 1位符号 + 3位指数部分（1位符号两位数字） + 三位小数部分，
  问计算机能表示的最大最小正数是?
-

（计算机）数字表示

局限性

- 有限的范围（最大数和最小数）
- 有限的精度（舍入误差）
- 有限的距离（数字间的距离 Δx 随数字本身增长）
  若t =计算机中数字的小数部分的有效数字的位数，
  机器误差：$M_\epsilon = b^{1-t}$，
  相对误差：$|\frac{\Delta x}{x}| \leq \frac{M_\epsilon}{2}$
  绝对误差：$\Delta x \leq \frac{M_\epsilon}{2}|x|$

（计算机）算术运算

加、减法：

$$x_1 \pm x_2 = m_1 b^{e_1} \pm m_2 b^{e_2}$$

- 假设 $e_1 > e_2$，

$$x_1 \pm x_2 = (m_1 \pm m_2 b^{e_2 - e_1}) b^{e_1} = m b^{e_1}$$

- 绝对误差： $\epsilon \leq \epsilon_1 + \epsilon_2$
- 相对误差： $\alpha = \frac{m - \hat{m}}{m}$ （无界！当 $m \to 0$）

乘、除法：

$x_1 x_2 = m_1 m_2 b^{e_1+e_2}$

- 指数相加 $\Rightarrow$ 小数相乘 $\Rightarrow$ 正则化并舍入
- $m = m_1 m_2 < 1$
- 相对误差： $\alpha \leq \alpha_1 + \alpha_2$

$x_1/x_2 = (m_1/m_2) b^{e_1-e_2}$

- 指数相加 $\Rightarrow$ 小数相除 $\Rightarrow$ 正则化并舍入
- 相对误差： $\alpha \leq \alpha_1 - \alpha_2$

问题

- 大规模加减法（循环）
- 大数小数相加减
- 相近的数相减
- 级数求和时湮灭 （如：向量内积）

Horner's 递归求和公式

- 多项式：

$$
\begin{aligned}
p(x) &= a_0x^3 + a_1x^2 + a_2x + a_3 \\
&= ((a_0x + a_1)x + a_2)x + a_3
\end{aligned}
$$

Horner's 递归求和公式

- 多项式：

$$\begin{aligned} p(x) &= a_0x^3 + a_1x^2 + a_2x + a_3 \\ &= ((a_0x + a_1)x + a_2)x + a_3 \end{aligned}$$

- 一般多项式

$$p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

Horner's 递归求和公式

- 多项式：

$$\begin{aligned} p(x) &= a_0x^3 + a_1x^2 + a_2x + a_3 \\ &= ((a_0x + a_1)x + a_2)x + a_3 \end{aligned}$$

- 一般多项式

$$p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

- 递归公式

$$b_0 = a_0, \quad b_i = a_i + b_{i-1}x, i = 1, \cdots, n$$
$$p(x) = b_n$$

例：

求积分：$y_n = \int_0^1 \frac{x^n}{x+5} dx,\ n = 0, 1 \cdots$

例：

求积分：$y_n = \int_0^1 \frac{x^n}{x+5} dx,\ n = 0, 1 \cdots$

递推关系：$y_n = \frac{1}{n} - 5y_{n-1}$

例：

求积分：$y_n = \int_0^1 \frac{x^n}{x+5} dx,\ n = 0, 1 \cdots$

递推关系：$y_n = \frac{1}{n} - 5y_{n-1}$

三数位循环：

$$
\begin{aligned}
y_0 &= \ln 6 - \ln 5 \approx 0.182 \\
y_1 &= 1 - 5y_0 \approx 0.090 \\
y_2 &= 0.5 - 5y_1 \approx 0.050 \\
y_3 &= 0.333 - 5y_2 \approx 0.083 \quad (> y_2!) \\
y_4 &= 0.083 - 5y_3 \approx -0.165 \quad (< 0!!)
\end{aligned}
$$

例：

求积分：$y_n = \int_0^1 \frac{x^n}{x+5} dx,\ n = 0, 1 \cdots$

递推关系：$y_n = \frac{1}{n} - 5y_{n-1}$

三数位循环：

$$
\begin{aligned}
y_{10} &\approx y_9 \Rightarrow y_9 + 5y_9 = 0.1 \Rightarrow y_9 = 0.017 \\
y_8 &= 1/45 - y_9/5 = 0.019 \\
y_7 &= 1/40 - y_8/5 = 0.021 \\
y_6 &= 0.025 \\
\cdots & \\
y_1 &= 0.088 \\
y_0 &= 0.182
\end{aligned}
$$

绝对误差

考虑 $y = f(x_1, x_2, x_3, ..., x_n)$，如果 ϵ_i 是 x_i的绝对误差， y的绝对误差？

绝对误差

考虑 $y = f(x_1, x_2, x_3, ..., x_n)$，如果 ϵ_i 是 x_i的绝对误差， y的绝对误差？

- 单变量: $y = f(x)$。回忆Taylor展开

$$f(x+\epsilon_x) = f(x) + f'(x)\epsilon_x + \frac{\epsilon_x^2}{2!}f''(x) + \cdots + \frac{\epsilon_x^n}{n!}f^{(n)}(x) + R_n,$$

其中 $R_n = \frac{\epsilon_x^{n+1}}{(n+1)!}f''(\xi_i)$， 因此 $\epsilon_y = \epsilon_x|f'|$，当 $\epsilon_x \ll 1$。

绝对误差

考虑 $y = f(x_1, x_2, x_3, ..., x_n)$，如果 ϵ_i 是 x_i的绝对误差， y的绝对误差？

- 单变量: $y = f(x)$。回忆Taylor展开

$$f(x+\epsilon_x) = f(x) + f'(x)\epsilon_x + \frac{\epsilon_x^2}{2!}f''(x) + \cdots + \frac{\epsilon_x^n}{n!}f^{(n)}(x) + R_n,$$

其中 $R_n = \frac{\epsilon_x^{n+1}}{(n+1)!}f''(\xi_i)$， 因此 $\epsilon_y = \epsilon_x|f'|$， 当 $\epsilon_x \ll 1$。

- 多变量:

当 $\epsilon_i \ll 1$， $\epsilon_y \leq \sum_{i=1}^{n} |\frac{\partial f(x_1,x_2,x_3,...,x_n)}{\partial x_i}|\epsilon_i.$

相对误差

- 乘法: $y = x_1 x_2$,

$$|\frac{\epsilon_y}{y}| \leq \sum_{i=1}^{2} |\frac{\epsilon_x}{x_i}|.$$

相对误差

- 乘法: $y = x_1 x_2$,

$$|\frac{\epsilon_y}{y}| \leq \sum_{i=1}^{2} |\frac{\epsilon_x}{x_i}|.$$

- 更一般的情形: $y = x_1^{m_1} x_2^{m_2} \cdots x_n^{m_n}$,

$$|\frac{\epsilon_y}{y}| \leq \sum_{i=1}^{n} |m_i| |\frac{\epsilon_i}{x_i}|.$$

条件数 (condition number)

- 数学问题的**条件 (condition)**：问题对于输入值的敏感程度。
- 数值算法 **不稳定 (unstable)**：如果输入值的误差被数值算法放大。
- 考虑 x and $f(x)$，**相对条件数 (the relative condition number)** ： $f(x)$ 和 x相对误差的比值。

条件数 (condition number)

- 数学问题的条件 **(condition)**：问题对于输入值的敏感程度。
- 数值算法 **不稳定 (unstable)**：如果输入值的误差被数值算法放大。
- 考虑 x and $f(x)$，**相对条件数 (the relative condition number)**： $f(x)$ 和 x相对误差的比值。

$$\kappa = \frac{\alpha_f}{\alpha_x} = \frac{\frac{f(x(1+\alpha_x))-f(x)}{f(x)}}{\alpha_x} \approx \frac{xf'(x)}{f(x)}.$$

条件数 (condition number)

- 数学问题的**条件 (condition)**：问题对于输入值的敏感程度。
- 数值算法 **不稳定 (unstable)**：如果输入值的误差被数值算法放大。
- 考虑 x and $f(x)$，**相对条件数 (the relative condition number)**： $f(x)$ 和 x相对误差的比值。

$$\kappa = \frac{\alpha_f}{\alpha_x} = \frac{\frac{f(x(1+\alpha_x))-f(x)}{f(x)}}{\alpha_x} \approx \frac{xf'(x)}{f(x)}.$$

- 当 $\kappa \gg 1$, 则称问题是“病态的 (ill-conditioned)”。

例：

- $f(x) = \sqrt{x^2 + 1} - x + 200$; $x = 100$,

例：

- $f(x)=\sqrt{x^2+1}-x+200$; $x=100$,
  $\kappa_f=|100\frac{-0.5\cdot 10^{-4}}{200.005}|=0.2\cdot 10^{-4}$, 问题是“良态的(well-conditioned)”。

例：

- $f(x) = \sqrt{x^2+1} - x + 200$; $x = 100$,
  $\kappa_f = |100\frac{-0.5\cdot 10^{-4}}{200.005}| = 0.2 \cdot 10^{-4}$, 问题是“良态的 (well-conditioned)”。
- $f(x) = \sqrt{x^2+1} - x$,

$$x = 100 \Rightarrow \kappa_f = -1.0$$

“算法”条件数 (condition number)

- $f(x)=\sqrt{x^2+1}-x$, 假设“四位十进制”计算机

“算法”条件数 (condition number)

- $f(x)=\sqrt{x^2+1}-x$, 假设“四位十进制”计算机

$$
\begin{aligned}
\hat{f}(0.1000\cdot 10^2) &= \sqrt{0.1000\cdot 10^5+0.1000\cdot 10^1}-0.1000\cdot 10^3\\
&= \sqrt{(0.1+0.0000{\color{red}1})\cdot 10^5}-0.1000\cdot 10^3=0,
\end{aligned}
$$

“算法”条件数 (condition number)

- $f(x) = \sqrt{x^2+1} - x$, 假设“四位十进制”计算机

$$\begin{aligned}\hat{f}(0.1000 \cdot 10^2) &= \sqrt{0.1000 \cdot 10^5 + 0.1000 \cdot 10^1} - 0.1000 \cdot 10^3 \\ &= \sqrt{(0.1 + 0.00001) \cdot 10^5} - 0.1000 \cdot 10^3 = 0,\end{aligned}$$

$$\begin{aligned}|\alpha_f| &= 1 \\ |\alpha_x| &= \left|\frac{\hat{x} - x}{x}\right| \le \frac{1}{2}10^{1-t} \simeq \frac{1}{2}10^{-3} \\ \kappa_A &= \frac{\alpha_f}{\alpha_x} \simeq 2000.\end{aligned}$$

“算法”条件数 (condition number)

- κ_A 称为“算法”条件数。由于数位表示的原因，κ_A 可能会大于问题本身的条件数κ_f。

“算法”条件数 (condition number)

- κ_A 称为“算法”条件数。由于数位表示的原因，κ_A 可能会大于问题本身的条件数κ_f。
- 解决办法：
  - 提高计算精度
  - 重写算法

$$f(x) = \frac{1}{\sqrt{x^2+1}+x},$$

$$\hat{f}(0.1000 \cdot 10^2) = \frac{1}{0.1000 \cdot 10^3 + 0.1000 \cdot 10^3} = 0.5000 \cdot 10^{-2}$$

$$\kappa_A \simeq 0 \ll 1.$$