

Numerical Analysis

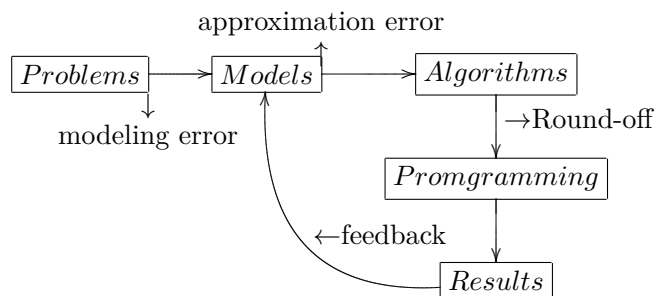
Iydon

2018 年 9 月 4 日

目录

1	Preface	3
2	Mathematical Preliminaries and Error Analysis	4
2.1	Round-off Errors and Computer Arithmetic	4
2.1.1	Binary machine numbers	4
2.1.2	Decimal machine numbers	4
2.1.3	Machine Operators	6
2.1.4	Nested method (秦九韶算法)	6
2.1.5	Convergence (收敛性)	7

1 Preface



2 Mathematical Preliminaries and Error Analysis

2.1 Round-off Errors and Computer Arithmetic

2.1.1 Binary machine numbers

定义 2.1 舍入误差 舍入误差形成原因:进行有限位的运算(*finite digits arithmetic*)

其中, *IEEE:754-2008* 规定二进制机器数 (*Binary machine numbers*) 中浮点数 (*floating-point*) 存储规范如下:

$$(-1)^S 2^{c-1023} (1+f)$$

$$\begin{cases} S : 0/1 & \text{signpart} \\ c : 11 \text{ digits} & \text{exponential part.} \\ f : 52 \text{ digits} & \text{mantissa part} \end{cases}$$

由于实数的稠密性,可知找不到比某一个数大的最小的数或小的最大的数,但是在计算机中可以找到,所以计算机不能表示所有的数。

2.1.2 Decimal machine numbers

$\pm 0.d_1 d_2 \cdots d_n \times 10^n$ 其中 $1 \leq d_1 \leq 9$, $0 \leq d_i \leq 9$, $\forall i \geq 2$ 。如果记真实的数为 y , 其浮点数表示为 $fl(y)$ 。

当存在 $y = 0.d_1 d_2 \cdots d_k d_{k+1} \cdots \times 10^n$, 其浮点数表示有如下两种方式:

1. Chopping: chop off digits, say $d_{k+1} d_{k+2} \cdots$.
2. Rounding: $y + 5^{n-(k+1)}$, then chopping.

例 2.1

$\pi = 3.14159265 \dots$, 取 5 位。

- Chopping: $fl(y) = 0.31415 \times 10^1$.
- Rounding: $fl(y) = 0.31416 \times 10^1$.

 π

定义 2.2 Suppose p^* is an approximation of p .

$$\begin{cases} \text{absolute error} &= |p^* - p| \\ \text{relative error} &= \frac{|p^* - p|}{p} \end{cases}$$

定义 2.3 有效数字 (Significant digits) p^* is said to approximate p with t significant digits. If t is the largest nonnegative integer, s.t.

$$\frac{|p - p^*|}{p} \leq 5 \times 10^{-t}$$

Chopping floating:

$$y = 0.d_1 \cdots d_k d_{k+1} \cdots \times 10^n$$

$$fl(y) = 0.d_1 \cdots d_k \times 10^n$$

Chopping: (其有效位数至少为 $k-1$)

$$\frac{|fl(y) - y|}{|y|} = \frac{0.0 \cdots 0 d_{k+1} \cdots \times 10^n}{0.d_1 \cdots d_k d_{k+1} \cdots \times 10^n} \leq 10^{1-k}$$

Rounding: (其有效位数至少为 k)

$$\frac{|fl(y) - y|}{|y|} \leq \frac{0.0 \cdots 1 d_{k+1} \cdots \times 10^n}{0.d_1 \cdots d_k d_{k+1} \cdots \times 10^n} \leq 10^{-k}$$

2.1.3 Machine Operators

记计算机的加减乘除为 $\oplus \ominus \otimes \oslash$ ，于是有

$$x \oplus y = fl(fl(x) \oplus fl(y))$$

Four cases to avoid:

1. 两个十分接近的数 (two nearly equal)。
2. 分子远大于分母 (numerator » denominator)。
3. 避免大数吃掉小数。

2.1.4 Nested method (秦九韶算法)

input : $a_0, a_1, \dots, a_n(\text{given}); x$

output: $P_n(x)$

```

1  $S_n \leftarrow a_n;$ 
2 for  $k \leftarrow n - 2$  to 0 do
3   |  $S_k \leftarrow xS_{k+1} + a_k;$ 
4 end
5  $P_n(x) \leftarrow S_0;$ 
```

```

1 def nested(poly:list=[1], x:float=0.0)->float:
2     """
3     Horner nested polynomial calculation.
4
5     Args:
6         poly: List, store the coefficient of the polynomial.
7         x: Float, specify the variable in the polynomial.
8
9     Returns:
10        Float, result.
11
12    Raises:
```

```
13         If 'poly' is empty, raise IndexError.
14         If type(args) does not correspond, raise TypeError.
15         """
16         result = poly[0]
17         for i in range(1, poly.__len__()):
18             result = x*result + poly[i]
19         return result
```

2.1.5 Convergence (收敛性)

Stable: small change in initial data and the error is small.

若 E_0 为初始值误差, E_n 为 n 步的误差,

- $E_n \approx C$ (不依赖 n), 称之为线性。
- $E_n \approx C^n E_0$ 则可由 C 的取值判断是否稳定。

定义 2.4 Rates of Convergence 当 $n \rightarrow \infty$, $\alpha_n \rightarrow \alpha$, $\beta_n \rightarrow 0$ (与 n 的取值无关), 则称 α_n 是以 β_n 的速度收敛到 α 的。