二、数值计算中的一些基本原则

- 避免绝对值小的数作除数
- 避免两个相近的数据相减
- 要防止大数“吃掉”小数
- 尽量减少计算工作量
- 选用数值稳定性好的算法

■ 避免绝对值小的数作除数

这一原则主要指尽量避免除数绝对值远远小于被除数绝对值的除法。

设 $z=y/x$（$x\neq 0$），如果 x 的绝对值远小于 y 的绝对值，由于

$$\varepsilon\left(\frac{y}{x}\right)\approx\frac{|x|\varepsilon(y)+|y|\varepsilon(x)}{|x|^2}$$

$$=\frac{\varepsilon(y)}{|x|}+\frac{|y|}{|x|^2}\varepsilon(x)$$

▪ 避免两个相近的数据相减

如果 $y \approx x$，现分析两个数的近似数作减法所得结果的误差. 设 $z = y - x$，则利用误差估计

$$\varepsilon(z) \approx \varepsilon(y) + \varepsilon(x)$$

有相对误差估计

$$\varepsilon_r(z) \le \frac{|y|}{|z|}\varepsilon_r(y)| + \frac{|x|}{|z|}\varepsilon_r(x)$$

当 $y \approx x$ 时，有 $z \approx 0$，计算结果的相对误差限可能很大，导致数值计算结果的有效数字位数减少。

▪ 要防止大数“吃掉”小数

一个绝对值很大的数和一个绝对值很小的数直接相加时，很可能发生所谓“大数吃小数”的现象。

例如，$a = 10^{13}$，$b = 4$，设想这两个数在具有12位浮点数计算机系统（12位有效位数系）中相加

$$a + b = 10^{13} + 4 = 1.0000000000000 \times 10^{13}$$

$$+0.0000000000004 \times 10^{13}$$

实际加法操作如下

$$1.0000000000\,0 \times 10^{13} + 0.0000000000\,0 \times 10^{13}$$

$$= 1.0000000000\,0 \times 10^{13}$$

二、数值计算中的一些基本原则

▪ 尽量减少计算工作量

在考虑算法时应注意简化计算步骤，减少运算次数 。

计算工作量小的算法不仅节约运行时间，而且使误差积累小。

例2 设计算法用于计算多项式

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

▪ 尽量减少计算工作量

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

算法一：

$$S_0 = a_0\,,$$
$$S_k = S_{k-1} + a_k x^k\,,\quad（k = 1，2，\ldots，n）$$
$$P_n（x）= S_n$$

这种算法计算复杂性怎么样？

$$x^k = x \cdot x^{k-1}$$

计算一个 n 次多项值需要用 **$2n$-1** 次乘法。

这种算法是否是最优的呢？

■ 尽量减少计算工作量

另一种典型算法是秦九韶算法

$$P_4(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

$$= a_0 + x(a_1 + x(a_2 + x(a_3 + x a_4)))$$

算法二：

$S_n = a_n$,

$S_{k-1} = a_{k-1} + xS_k$, （$k = n$，$n-1$，…，1），

$P_n（x）= S_0$

计算一个 n 次多项值需要用 n 次乘法。

■ 选用数值稳定性好的算法

不同的算法在执行过程中对数据误差的影响是不一样的。舍入误差对计算结果影响不大的算法被称为数值**稳定的算法**.

例3 利用递推式计算定积分

$$I_n = e^{-1}\int_0^1 x^n e^x dx$$

(n= 0， 1， 2， …， 20)的值。

算法一：

$$I_n = e^{-1}\left(x^n e^x\Big|_0^1 - n\int_0^1 x^{n-1}e^x dx\right) = 1 - nI_{n-1}$$

其中 $$I_0 = e^{-1}\int_0^1 e^x dx = e^{-1}(e-1) = 1 - e^{-1}$$

得递推关系式

$$\begin{cases} I_0 = 1 - e^{-1} \\ I_n = 1 - nI_{n-1}, (n = 1,2,\cdots) \end{cases}$$

利用递推式可得20个数据如下表：

S_1	0. 36787944117144	S_{11}	0. 07735222935878
S_2	0. 26424111765712	S_{12}	0. 07177324769464
S_3	0. 20727664702865	S_{13}	0. 06694777996972
S_4	0. 17089341188538	S_{14}	0. 06273108042387
S_5	0. 14553294057308	S_{15}	0. 05903379364190
S_6	0. 12680235656152	S_{16}	0. 05545930172957
S_7	0. 11238350406936	S_{17}	0. 05719187059731
S_8	0. 10093196744509	S_{18}	-0. 02945367075154
S_9	0. 09161229299417	S_{19}	1. 55961974427919
S_{10}	0. 08387707005829	S_{20}	-30. 19239488558378

对积分值有估计式：

$$\frac{e^{-1}}{n+1} \leq e^{-1}\int_0^1 x^n e^x dx \leq \frac{1}{n+1}$$

算法二：

由递推公式

$$I_n = 1 - nI_{n-1}$$

有

$$I_{n-1} = \frac{1}{n}(1 - I_n)$$

由 I_n 的估计式

$$\frac{e^{-1}}{n+1} \le I_n \le \frac{1}{n+1}$$

取

$$I_{30} \approx S_{30} = \frac{1}{31}$$

有

$$S_{n-1} = \frac{1}{n}(1 - S_n), \qquad n = 30, 29, 28, \cdots, 2$$

利用递推式可得20个数据如下表:

S_1	0. 36787944117144	S_{11}	0. 07735222886266
S_2	0. 26424111765712	S_{12}	0. 07177325364803
S_3	0. 20727664702865	S_{13}	0. 06694770257562
S_4	0. 17089341188538	S_{14}	0. 06273216394138
S_5	0. 14553294057308	S_{15}	0. 05901754087930
S_6	0. 12680235656153	S_{16}	0. 05571934593124
S_7	0. 11238350406930	S_{17}	0. 05277111916899
S_8	0. 10093196744559	S_{18}	0. 05011985495809
S_9	0. 09161229298966	S_{19}	0. 04772275579621
S_{10}	0. 08387707010339	S_{20}	0. 04554488407582

■ 结论

初始误差在算法执行过程中不断减小, 这种算法称为数值稳定算法。

在算法执行过程中, 舍入误差对计算结果影响不大的一类算法被称为数值稳定算法; 否则称为不稳定算法.