关于 lnx 在 785 机上实现的一种算法

周泽滋

提　要　本文提出了关于 lnx 的一种算法，它综合了算法Ⅰ、算法Ⅱ、算法Ⅲ的优点，克服了它们的缺点，并给出了误差分析。

785 机是一台具有多功能、多部件和有大容量内存的向量流水线巨型计算机，在为其配置标准子程序时，必须注意使之满足两点要求：(1) 保证计算结果的精确性，(2) 尽量提高计算速度。关于 lnx 的算法，所见也有多种，本文仅就其中三种算法作一比较，然后陈述本方案，最后讨论本方案在计算机上实现时所能满足的精度。

一、三种算法的比较

三种算法是指 CRAY—1 机标准子程序库提供的算法[1]（简称算法Ⅰ）和朱展能同志提出的一种算法[2]（简称算法Ⅱ），以及为 785 机双精度标准子程序设计的一种算法[3]（简称算法Ⅲ）.

1.1 算法Ⅰ与算法Ⅱ的比较

(1) 对变元 x 的表示式的要求

算法Ⅰ：　$x=2^p\cdot f$　　$\left(\frac{1}{\sqrt{2}}\leqslant f<\sqrt{2}\right)$;

算法Ⅱ：　$x=2^m\cdot c$　　$\left(\frac{1}{2}\leqslant c<1\right)$,

其中 p、m 均为整数。

算法Ⅱ对变元表示式的要求就是变元在计算机上关于浮点数的规格化形式，而在计算机上实现算法Ⅰ对变元表示式的要求，须作如下处理：

$$x=2^p\cdot f=\begin{cases}2^{m-1}\cdot(2\cdot c) & \left(\frac{1}{2}\leqslant c<\frac{1}{\sqrt{2}}\right)\\ 2^m\cdot c & \left(\frac{1}{\sqrt{2}}\leqslant c<1\right)\end{cases}$$

本文 1981 年 10 月 28 日收到

即当$\frac{1}{2}\leqslant c<\frac{1}{\sqrt{2}}$时，$\begin{cases}p=m-1\\f=2\cdot c;\end{cases}$

当$\frac{1}{\sqrt{2}}\leqslant c<1$时，$\begin{cases}p=m\\f=c.\end{cases}$

(2) 变换公式

算法Ⅰ：$t=\frac{f-1}{f+1}$，从而 $f=\frac{1+t}{1-t}$；

算法Ⅱ：$t=\frac{c-\frac{1}{\sqrt{2}}}{c+\frac{1}{\sqrt{2}}}$，从而 $c=2^{-\frac{1}{2}}\cdot\frac{1+t}{1-t}$，

变换公式基本相同。

(3) 计算公式

算法Ⅰ：$\ln x=p\cdot\ln 2+\ln\frac{1+t}{1-t}$；

算法Ⅱ：$\ln x=m\cdot\ln 2-\frac{1}{2}\cdot\ln 2+\ln\frac{1+t}{1-t}$.

关于$\ln\frac{1+t}{1-t}$的处理，算法Ⅰ采用有理逼近，算法Ⅱ则用多项式逼近，算法Ⅱ较算法Ⅰ多$-\frac{1}{2}\ln 2$这一项。

(4) 可以证明，在计算机上实现算法Ⅰ时精度容易得到保证。在1点附近用算法Ⅱ计算只能得到少数几位有效数字。

1.2 算法Ⅱ介绍[1],[5]

(1) 设变元x表示为

$$x=2^m\cdot c\qquad\left(\frac{1}{2}\leqslant c<1\right),\ m\text{为整数。}\tag{1.1}$$

(2) 将区间$\left[\frac{1}{2},\ 1\right)$分成$2^N$（$N$为非负整数）个等子区间：

$$I_k=[a_{k-1},\ a_k)\qquad(k=1,2,\cdots,2^N),$$

其中

$$a_l=\frac{1}{2}+l\cdot 2^{-(N+1)}\qquad(l=0,1,\cdots,2^N).\tag{1.2}$$

(3) 当$c\in I_k$时，在I_k中选取一点b_k，使得作变换

$$t=\frac{b_k-c}{b_k+c}\tag{1.3}$$

时，I_k映到以原点为中心的对称区间$(-d_k,\ d_k](d_k>0)$. 由于变换(1.3)的严格单调性，可以推出

$$b_k=\sqrt{a_{k-1}a_k}=\frac{1}{2}\cdot[1+(k-1)\cdot 2^{-N}]^{\frac{1}{2}}\cdot[1+k\cdot 2^{-N}]^{\frac{1}{2}}.$$

由(1.3)得

$$c=b_k\cdot\frac{1-t}{1+t}.$$

(4) 计算公式

$$\ln x=m\cdot\ln 2+\ln b_k+\ln\frac{1-t}{1+t}, \tag{1.4}$$

其中 $x=2^m\cdot c$，且 $c\in I_k$。

(5) 关于 $\ln\frac{1-t}{1+t}$ 的处理是用其台劳多项式

$$-\sum_{s=0}^{n}\frac{2}{2s+1}\cdot t^{2s+1}$$

近似表示，这里根据要求适当确定出 N 和 n .

此算法要求占部份内存以存放一部份常数(b_k，$\ln b_k$ 等).

为减少乘法次数，作者对于 $\sum_{s=0}^{n}\frac{1}{2s+1}(t^2)^s$ 的处理还作了一些工作，这里就不介绍了。

1.3 算法Ⅰ与算法Ⅲ的比较

如果不考虑 $\ln\frac{1-t}{1+t}$ 的处理方法，那么算法Ⅰ就是算法Ⅲ中 $N=0$ 时的特殊情况，显见在提高计算速度方面来说，选取 $N=0$ 并非最好的方案。

二、本文关于计算 lnx 采用的方案

(1) 设变元 x 表示式为

$$x=2^m\cdot c \qquad \left(\frac{1}{2}\leqslant c<1\right), \tag{2.1}$$

其中 m 为整数。

(2) 将 $\left[\frac{1}{2},\ 1\right)$ 分为 2^N+1 (N 为非负整数) 个子区间：

$$I_k=[a_{k-1},\ a_k) \qquad (k=1,2,\cdots,2^N+1),$$

其中

$$a_0=0.5,$$

$$a_{2^N+1}=1.0,$$

$$a_k=\frac{1}{2}[1+(k-1)2^{-N}]+\varepsilon_N .$$

$(k=1,2,\cdots,2^N)$，这里 ε_N 是一个与 N 有关的量* $(0<\varepsilon_N<2^{-(N+2)})$.

* 此处 ε_N 的选取原则是使在变换(2.3)下，I_1 及 $[a_1,b_2]$ 的象区间等长为最佳。

记 $$b_k=\frac{1}{2}+(k-1)2^{-(N+1)} \qquad (k=1,2,\cdots,2^N+1) \tag{2.2}$$

则有 $$a_k=b_k+\varepsilon_N \qquad (k=1,2,\cdots,2^N).$$

(3) 当 $c\in I_k$ 时，作变换

$$t=\frac{c-b_k}{c+b_k}, \tag{2.3}$$

从而 $$c=b_k\cdot\frac{1+t}{1-t}.$$

(4) 计算公式

$$\ln x=(m\cdot\ln 2+\ln b_k)+\ln\frac{1+t}{1-t}, \tag{2.4}$$

而 $\ln\frac{1+t}{1-t}$ 则用它的台劳多项式

$$2t\sum_{s=0}^{n}\frac{1}{2s+1}(t^2)^s$$

近似表示，对有链接技术的 785 机来说是可取的。

(5) 关于 N、n 的选取

由于此算法可能失去有效数字的位数与 N 有关，在一定范围内它随 N 的增大而增大，但由于 N 增大时子区间的长度变小，从而使得满足给定方法误差限的 n 值能变小，因而能提高计算速度，并能减少捨入误差的积累。

在浮点数尾数为 48 位（二进制，以下同）的 785 机上，希望计算结果具有 44 位有效数字，只须取 $N=2$，$n=5$ 即可。

(6) 算法 I 其区间分法和所作变换基本上是此方案 $N=0$ 的情况。

(7) 此算法需占部份内存以存放数据（b_k，$\ln b_k$ 等），对于具有大容量内存的计算机来说，这是可行的。

三、误 差 分 析

当取 $N=2$，$n=5$ 时，此算法引起的相对误差限 $<2^{-53}$，如果在尾数为 48 位的 785 机上计算

$$2t\sum_{s=0}^{5}\frac{1}{2s+1}(t^2)^s$$

时，其捨入误差积累引起有效数字位数的损失以 1～2 位为估计值，则对公式(2.4)进行计算时，其有效数字位数的损失，最多不超过 4 位。实际上当 $m=1$ 且 $c\in I_2$ 时，计

算 $m\ln 2+\ln b_k$ 可能损失的有效数字位数最多，但也不会超过 2 位。

这里我们认为在内存单元中给出变元 x 的值是 x 的精确值，由于 b_k 的取值在计算机中能精确表出，从而变换(2.3)能使 t 具有足够位数的有效数字。

关于 $\sum_{s=0}^{n}\frac{1}{2s+1}(t^2)^s$ 的处理，如果用关于 t^2 的较低次的契比雪夫多项式逼近，还可降低多项式的次数，但由于 $\ln 1=0$，从而对 1 点附近的 x 算出 $\ln x$ 之值其有效数字位数可能要减少一些。

四、求数组的对数的并行计算

在用向量机计算一组数各自的对数值时，用此法在 $N\neq 0$ 时将增加对内存的访问，同时还可能出现体碰头现象，但取 $N=0$，并对此方案的变换(2.3)及计算公式作如下处理

$$t=\frac{c-b_k}{c+b_k}=\frac{(1+\delta_k)c-1}{(1+\delta_k)c+1}, \tag{2.3'}$$

其中

$$\delta_k=\begin{cases}1 & k=1\\ 0 & k=2\end{cases}$$

及

$$\ln x=(m\cdot\ln 2+\ln b_k)+\ln\frac{1+t}{1-t}=(m-\delta_k)\cdot\ln 2+\ln\frac{1+t}{1-t} \tag{2.4'}$$

这样处理在运算中虽然增加了几步，但可减少对内存的访问次数，更不会出现体碰头。

五、总　结

此算法在确保一定精度情况下可提高计算速度。

此算法在 $N=0$ 时经过一些处理后可在并行机上实现求一组数相应的对数值。

参 考 文 献

[1] CRAY—1 数学子程序库使用手册(2240014)

[2] 朱展能：关于 e^x，$\ln x$，$\sqrt{x}$ 在 785 机上实现的算法，《计算机工程与科学》，国防科技大学研究所，1981 年第 1 期。

[3] 汪裕武："关于 $\ln x$ 双精度的计算方案"，国防科技大学七系学术报告会报告。1981.

[4] 李岳生、黄友谦：《数值逼近》，人民教育出版社，1978.

[5] 李家楷、史应光：对数函数和指数函数的子程序方案，《计算数学》，科学出版社，1979 年第 2 期。

An Algorithm of lnx on the 785 Computer

Zhou Ze—zi

Abstract

In this paper an algorithm of $\ln x$ is presented. It sums up some of algorithms Ⅰ、Ⅱ、Ⅲ's advantages and overcomes some of their disadvantages. Moreover, an estimation of error is rigorously established.