割线法——一种求函数近似值的快速硬件算法

张 民 选

提 要 本文研究了计算机硬件求函数近似值的一种快速算法——割线法，并在实际应用中对其进行了改进。

一、研 究 背 景

在国内外近期推出的大型、巨型计算机中，为满足现代科学计算，信息处理，实时控制对计算机速度的需要，用硬件直接实现x^{-1}、$\sqrt{x}$、$\ln x$、e^x、$\sin x$、$\cos x$等常用函数求值的机器为数不少。为进一步挖掘潜力，在某些机器中，设置了专用的求x^{-1}（或$x_1 * x_2^{-1}$）、$\sqrt{x}$值的流水线部件。要使实现常用函数求值的硬件设计，成本开销小，速度增益大，就必须研究函数求值的硬件算法。在文献[2]中，对倒数迭代算法进行研究和实现方案进行探讨的结果，已使我们体会到对硬件算法进行研究是大有益处的。

CRAY—1 高性能计算机系统，其中央处理机中，设有全流水线式的多单功能部件，且各功能部件又能“链接”成复合功能部件。为提高除法速度，设置了专用的“倒数近似值”部件，求取x^{-1}的半精度值后，再利用软件迭代一次求得全精度值。这样设计，使得求x^{-1}的近似值部件不至太大，且又适合全流水线结构的机器特点，除法运算每三拍可得一个结果。CRAY—1 这种既考虑提高机器速度，又兼顾机器全流水线化的特点，采用软硬结合的设计思想值得借鉴。

通过对经典的函数求值算法的分析、研究，我们引进了一种适于计算机实现的求函数近似值的硬件算法——割线法。基于求取常用函数的半精度（27位以内）值的硬件设备，在现有器件水平，进行一次近似计算就够了。所以，我们只以一次近似计算为基础来讨论割线法。

二、割线法的基本概念

1. 一般描述

设函数$f(x)$在闭区间$[a,b]$上连续，在开区间(a,b)内有限导数$f'(x)$，$f''(x)$存在且不变号，则函数$f(x)$满足使用割线法求值的条件。

本文1988年8月4日收到

一直线和函数 $f(x)$ 曲线在 $[a, b]$ 区间上相割，且有两个交点，则称这直线为函数 $f(x)$ 曲线的割线。在 $[a,b]$ 区间内用求割线方程 $g(x)$ 的值代替求函数 $f(x)$ 的值的近似计算方法，称为割线法。

函数 $f(x)$ 的曲线和它的割线 $g(x)$ 在 $[a,b]$ 区间内相交于 c、d 两点（见图1），c 点横坐标为 $(a+\Delta_1)$，d 点横坐标为 $(a+\Delta_1+\Delta_2)$，则割线的通式为：

$$g(x)=f(a+\Delta_1)+\{[f(a+\Delta_1+\Delta_2)-f(a+\Delta_1)]/\Delta_2\}*[x-(a+\Delta_1)] \tag{1}$$

式中：$0\leqslant\Delta_1<(b-a)$，$0<\Delta_2\leqslant(b-a)$，且有：$(\Delta_1+\Delta_2)\leqslant(b-a)$。

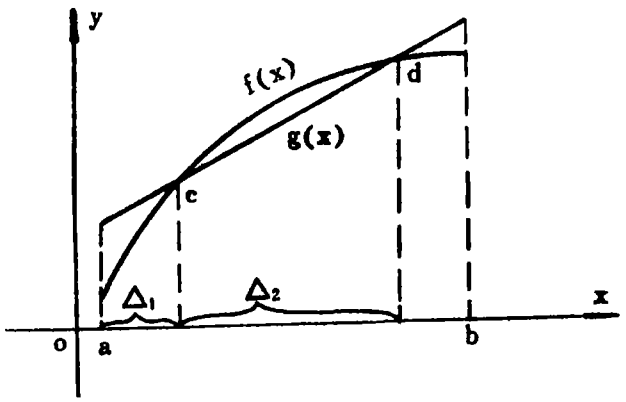

图 1 割线法

2. 三种典型算式

若以割线的端点为起点描述割线，则可得割线法在实际应用中的两个基本算式：

$$g(x)=Y_a+T(x-a) \tag{2a}$$

$$g(x)=Y_b+T(x-b) \tag{2b}$$

式中：Y_a、Y_b 分别为割线在 a 端和 b 端的取值，T 为割线斜率。

若令割线参数 $\Delta_1=0$，$\Delta_2=(b-a)$，则可得典型算式Ⅰ：

$$g_1(x)=Y_{1a}+T_1(x-a) \tag{3a}$$

$$g_1(x)=Y_{1b}+T_1(x-b) \tag{3b}$$

式中：$Y_{1a}=f(a)$，$Y_{1b}=f(b)$，$T_1=(f(b)-f(a))/(b-a)$。

若令割线参数 $\Delta_1=(c-a)$，（其中 c 满足方程：$f'(c)=(f(b)-f(a))/(b-a)$，$\Delta_2\to0$，则此割线方程为：$g_2(x)=f(c)+f'(c)(x-c)$。以端点为起点描述此割线，可得典型算式Ⅱ：

$$g_2(x)=Y_{2a}+T_2(x-a) \tag{4a}$$

$$g_2(x)=Y_{2b}+T_2(x-b) \tag{4b}$$

式中：$Y_{2a}=f(a)+\{f(c)-[f(a)+T_2(c-a)]\}$，

$Y_{2b}=f(b)+\{f(c)-[f(b)+T_2(c-b)]\}$，$T_2=[f(b)-f(a)]/(b-a)$。

由算式Ⅰ和算式Ⅱ，可导出一条逼近函数 $f(x)$ 时误差最小的割线，用此割线逼近函数，则得典型算式Ⅲ：

$$g_3(x)=Y_{3a}+T_3(x-a). \tag{5a}$$

$$g_3(x)=Y_{3b}+T_3(x-b) \tag{5b}$$

式中：$Y_{3a}=f(a)+\{f(c)-[f(a)+T_3(c-a)]\}/2$，

$Y_{3b}=f(b)+\{f(c)-[f(b)+T_3(c-b)]\}/2$，$T_3=[f(b)-f(a)]/(b-a)$.

3. 误差估计

由高等数学知识，不难得到割线法三种典型算式的误差估计式。

若令 $A=|f(c)-[f(a)+f'(c)(c-a)]|$，则：

$$\varepsilon_1(x)=|f(x)-g_1(x)|\leqslant A \tag{6a}$$

$$\varepsilon_2(x)=|f(x)-g_2(x)|\leqslant 2^{-1}|f''(z)(x-c)^2|\leqslant A \tag{6b}$$

$$\varepsilon_3(x)=|f(x)-g_3(x)|\leqslant 2^{-1}A \tag{6c}$$

式中：$z=c+\theta(x-c)$，$0\leqslant\theta<1$，$f'(c)=[f(b)-f(a)]/(b-a)$.

从误差估计式 (6a)、(6b)、(6c) 可知，采用割线法逼近函数，其误差成平方级减小，即每逼近一次，精度提高一倍以上（指位数）。在初值精度相同的基础上，分别用三种典型算式逼近函数一次，算式Ⅲ求得的函数近似值比算式Ⅰ、算式Ⅱ求得的函数近似值高一位二进制精度。

三、割线法的硬件实现

由割线法求函数近似值的基本算式 $g(x)=Y+T(x-a)$可知，用割线法逼近函数一次有四步运算：①查T、Y_a或Y_b两个初始常数表，②做减法求出$(x-a)$或$(x-b)$，③做乘法求出减法结果Δx和T相乘的积，④做加法求出乘积和Y_a或Y_b的和，得一次近似的最终结果。

从上看来，用割线法求函数近似值其计算步骤并不少，然而在计算机硬件实现时，其中$(x-a)$或$(x-b)$不需做一次真正的减法，一般按 2^{-n} 步长分段用割线法逼近函数，求$(x-a)$或$(x-b)$只要将操作数 $x(2^{-1}\leqslant x<1)$ 的前 n 位抹去，余下的数值(或正或负)参加乘法运算即可。由于 $g(x)$ 逼近函数有误差，乘法运算只要保证运算不增加近似值的误差，对不影响精度的低位可以截尾，以节省硬件。最后一次加法运算，可和乘法运算的求全积合并起来，进行一次三操作数的加法。这样，一次近似计算的时间实际上只是进行一次查表和一次截断乘法的时间。若查T表查出的是乘法译码信号、或在查T表的同时用$(x-a)$的结果进行乘法译码，那么，一次截断乘法的运算时间就完成了一次函数逼近。可见，其运算速度是非常快的。

上面是基于有Y_a或Y_b和T两个表格的情况下进行讨论的，若采用 $g_2(x)=f(c)+f'(c)(x-c)$式逼近函数，其逼近精度比常用的切线法精度高。对于某些函数，只需查一个常数因子表，在表格硬件成本高时，这也是很可取的。例如函数 x^{-1}，第一次迭代时用 $Y_1=c^{-1}-(c^{-1})^2(x-c)$ 式（其中 $c=a+(b-a)/2$）进行计算，和用 $Y_1=a^{-1}-(a^{-1})^2(x-a)$ 式进行计算，若c^{-1}和a^{-1}的表格容量相同，则前者比后者的计算结果高两位二进制精度。或者说，在要求相同精度的情况下，前者的表格容量是后者表格容量的二分之一，这可大大地节省表格硬件。

要进行多次迭代时，第一次采用割线法的三种典型算式之一去逼近函数，第二次以后可采用割线法的简单算式(如令$\Delta_1=0$，$\Delta_2\to 0$)或其它的迭代法进行计算。

四、割线法的改进

计算机常采用分段逼近的方法提高一次近似计算的精度，在$[a,b)$区间内，按2^{-m}等长分割成n个子区间，在各子区间中分别使用割线法逼近函数。相应割线法的基本算式为：

$$g(x)=Y_{ai}+T_{ai}(x-a_i) \qquad a_i\leqslant x<a_{i+1} \qquad (i=0,1,2,\cdots,(n-1).) \tag{7a}$$

$$g(x)=Y_{bi}+T_{bi}(x-b_i) \qquad b_{i-1}\leqslant x<b_i \qquad (i=1,2,3,\cdots,n). \tag{7b}$$

式中：Y_{ai}为$[a_i,a_{i+1})$区间中的割线在a_i点的取值，Y_{bi}为$[b_{i-1},b_i)$区间中的割线在趋近于b_i点的取值，$T_{ai}=2^m(f(a_{i+1})-f(a_i))$，$T_{bi}=2^m(f(b_i)-f(b_{i-1}))$.

若在$[a,b)$区间的每个子区间中单独使用(7a)式或(7b)式，则整个区间中的Y_a或Y_b、T初始常数共有$2n$个，这些常数都需有相应的硬件来产生或存贮器空间来存放，是否可以减少一些常数呢？

在不损失整个区间的计算精度且不增加运算时间的前提下，提供分段使用割线法求函数近似值的一个改进方案：函数$f(x)$在$[a,b)$区间内，等长分割成n($n=(b-a)/2^{-m}$)个子区间，n个子区间又划分为奇子区间集$(1,3,5,\cdots,(n-1))$和偶子区间集$(2,4,6,\cdots,n)$，在$n/2$个奇子区间集中使用(7b)式逼近函数，在$n/2$个偶子区间集中使用(7a)式逼近函数，这样，相邻奇偶子区间对的初始常数Y_{bi}、$Y_{ai}(i=1,3,\cdots,(n-1))$在同一分割点取值，但其取值并不一定相等，可在其中选取一个做为公用的初始常数Y_i，选取的原则为：

$$Y_i=\begin{cases} Y_{bi} & |Y_{bi}-f(b_i)|\geqslant|Y_{ai}-f(b_i)| \\ Y_{ai} & |Y_{bi}-f(b_i)|\leqslant|Y_{ai}-f(b_i)| \end{cases} \tag{8}$$

这样处理后，减少了二分之一的初始常数，可望节省一半的表格硬件。交替使用(7a)和(7b)式逼近函数，相邻子区间对公用一个Y_i初始常数的改进割线法示于图2。

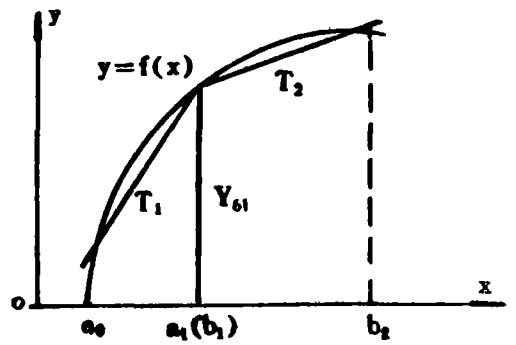

图 2 改进割线法

$$g(x)=Y_{b1}+T_1(x-b_1)$$
$$b_0\leqslant x<b_1$$
$$g(x)=Y_{a1}+T_2(x-a_1)$$
$$a_1\leqslant x<a_2$$

更正：图中横坐标a_0、$a_1(b_1)$、b_2改为：b_0、$b_1(a_1)$、a_2.
Y_{b1}改为：$Y_{b1}(Y_{a1})$.

五、割线法的应用

在计算机中，对常用函数如x^{-1}、$\sqrt{x}$、$\sqrt[3]{x}$、$\ln x$、e^x、$\sin x$、$\cos x$等提供专用指令，设置计算这些函数的硬件，是提高计算机效能的行之有效的措施。若采用改进的割线法进行计算，则只要在机器中设置Y、T两个常数存贮器和一条站数较少的运算流水线，就能求取这些常用函数的半精度或其它精度的近似值，再经软件用设计好的算法进行处理，则可较快地求得全精度值，从而增加机器的处理能力。软件迭代时，第一次计算用改进的割线法逼近函数，也能节省存贮空间和缩短计算时间，从而提高机器的处理效率。

割线法是为计算机能又快又省又简地求取函数近似值而研究的硬件算法。应用该算法，我们设计了一个求倒数近似值的流水线部件，该部件与文献[2]中的相同部件比较，其流水线站数从六站减少到三站，插件量也略有减少，其它指标基本相同。

胡守仁教授、周兴铭副教授、杨晓东副教授对本研究工作给予了大力支持和热情指导；钟士熙讲师、何能富讲师在百忙之中审阅了本文并提出了宝贵意见；在此，作者深表谢意！

参 考 文 献

[1] 慈云桂 胡守仁 国外巨型计算机系统发展动向 1982.3

[2] 周兴铭 张民选 倒数迭代算法的理论分析与方案探讨，计算机学报 第四卷第五期 1981.5

[3] 黄凯 计算机算术运算原理、结构与设计，科学出版社，1980.

The Secant Algorithm-A Hardware Algorithm of the Computer to Evaluate Approximation of the Functions at High Speed

Zhang Min-xuan

Abstract

In this paper, the secant algorithm—a hardware algorithm of the computer to evaluate approximation of the functions at high speed is suggested. First, the definition of the secant algorithm is described and its scheme is discussed. Then, an improvement of the secant algorithm is given, and an economical method of the structure of the initial value table is recom mended. Finally, the application of the secant algorithm to the design of hardware of the computer is discussed.