坐标旋转法的收敛性, 误差估计及扩展*

谷　峰

(浙江经济职业技术学院)

摘　要

本文通过推广正规序列的定义, 得到一个不等式, 由此证明了统一计算初等函数数值的坐标旋转法的收敛性, 作了误差估计, 并拓广了所计算的函数范围, 成为快速计算所有初等函数的统一算法.

关键词: 坐标旋转法, 初等函数, 统一算法

CONVERGENCE AND ERROR ESTIMATION OF COORDINATE ROTATING ALGORITHM AND ITS EXPANSION

Gu Feng

(*Zhejiang Technology Institute of Economy, Hangzhou, Zhejiang* 310018)

Abstract

In this paper, the convergence of coordinate rotating algorithm, a united algorithm for computing elementary functions, has been proved by expanding the definition of normal series. The algorithm's error was also estimated. Furthermore the algorithm was expanded to computing more elementary functions and thus it really become a united algorithm for computing all elementary functions.

Key words: coordinate rotating, elementary functions, united algorithm

初等函数计算公式多采用多项式或有理分式逼近来实现. 文 [1] 对应用矩阵函数 e^{xA}(其中 A 是矩阵) 计算初等函数的算法 — 坐标旋转法, 进行了研究和推广, 但未能计算全部初等函数. 浙江大学蔡耀志老师给出了多个初等函数的算法, 对坐标旋转法作了推广, 但其算法未做收敛性证明和误差分析. 本文通过对正规序列定义的拓广, 给出了正规序列的一个不等式, 对蔡耀志老师给出的几个算法进行了收敛证明和误差分析, 使其理论上得以完善. 本文还进一步扩展了算法, 给出了计算 $tgx, ctgx, thx, cthx$ 的算法及其收敛证明和误差分析. 因为算法是着眼于在仅定义了加减法及逻辑运算的基本计算系统中应用实现的, 因此上述几个函数的坐标旋转算法是有意义的. 经过这样的推广, 本算法真正成了初等函数的快速统一算法.

* 2004 年 2 月 2 日收到.

§1. 几个算法的收敛性和误差分析

首先对 [1] 中的正规序列定义加以拓广, 给出正规序列的定义如下:

定义. 正规序列 $\{\delta_i\}_0^\infty$ 是一个单调下降收敛于 0 的正数序列, 其对所有

$$x \in \left\{x : |x| \le R(\delta) = \sum_{i=0}^{\infty} \delta_i\right\}$$

满足

$$|x - F_n(x, \delta)| \le \delta_n,$$

其中 $F_n(x,\delta)$ 定义为 $x_0 = x$, $x_{i+1} = x_i - sgn(x_i) * \delta_i$, $i = 0, 1, \cdots$.

$$F_n(x, \delta) = \sum_{i=\mathbf{0}}^{n} sgn(x_i) * \delta_i,$$

其中的 $R(\delta)$ 称为度量半径, $F_n(x,\delta)$ 称为 x 关于 $\{\delta_i\}_0^\infty$ 的度量展开.

常用的正规序列有

$$\left\{2^{-i}\right\}_0^\infty, \ \left\{arctg2^{-i}\right\}_0^\infty, \ \left\{arcth2^{-i+\left[\frac{i}{5}\right]}\right\}_1^\infty$$

等等.

下面给出关于正规序列的一个不等式及其收敛性.

引理 1. 对正规序列 $\{\delta_i\}_0^\infty$ 有

$$\delta_{n+1} < \sum_{i=n+1}^{\infty} \delta_i \le \delta_n.$$

证明. 取 $x = \sum\limits_{i=0}^{\infty} \delta_i$ 显然有

$$F_n(x, \delta) = \sum_{i=0}^{n} \delta_i.$$

由正规序列定义, 有

$$|x - F_n(x, \delta)| = \sum_{i=n+1}^{\infty} \delta_i \le \delta_n.$$

右边的不等式由此证明, 左边的不等式是平凡的. 证毕.

推论. 若 $\{\delta_i\}_0^\infty$ 为正规序列, 则 $\sum\limits_{i=0}^{\infty} \delta_i$ 收敛, 从而 $R(\delta)$ 为有限数.

这是引理 1 的显然推论. 无需证明.

由此引理, 我们即可讨论几个算法的收敛性和误差估计.

(A) $l_n x$ 的算法

取 $\{\delta_i\} = \left\{arcth2^{-i+[\frac{i}{5}]}\right\}_1^{+\infty}$,

$$\begin{cases} x_1 = 0, y_1 = k = \prod_{i=1}^{n} ch\delta_i, \\ S_i = sgn\left(x - y_i\right), \\ x_{i+1} = x_i + s_i * \delta_i, \\ y_{i+1} = y_i + s_i * 2^{-i+[\frac{i}{5}]} * y_i, \end{cases} \quad i = 1, 2, 3, \cdots, \tag{1}$$

当 $i = n$ 时, $x_{n+1} \approx l_n x$, $y_{n+1} \approx x$.

定理 1. 由 (1) 式得到的 $\{x_i\}$ 是收敛序列, 收敛至 $l_n x$, 且 $|x_{n+1} - l_n x| \le \delta_n$.

证明. 由递推式 (1) 易得 $x_{i+k} - x_i = \sum_{j=i}^{i+k-1} S_j * \delta_j$, 从而由引理 1

$$|x_{i+k} - x_i| \le \sum_{j=i}^{i+k-1} \delta_j \le \sum_{j=i}^{\infty} \delta_j \le \delta_{i-1}, \tag{*}$$

所以 $\{x_i\}$ 是 Cauchy 序列. 由算法的结构知 $x_i \to l_n x$, 在 (*) 中令 $k \to \infty$ 得 $|l_n x - x_i| \le \delta_{i-1}$. 证毕.

(B) $\arcsin x$, $\arccos x$ 的算法

取 $\{\delta_i\} = \{arctg2^{-i}\}$,

$$\begin{cases} x_1 = 0, y_1 = 0, z_1 = B = \prod_{i=0}^{n} \cos\delta_i, \\ S_i = sgn\left(x - y_i\right), \\ x_{i+1} = x_i + s_i * \delta_i, \\ y_{i+1} = y_i + s_i * 2^{-i} * z_i, \\ z_{i+1} = z_i - s_i * 2^{-i} * y_i, \end{cases} \quad i = 1, 2, 3, \cdots, \tag{2}$$

当 $i = n$ 时, $x_{n+1} \approx \arcsin x$, $y_{n+1} \approx x$, $z_{n+1} \approx \sqrt{1 - x^2}$.

定理 2. 算法 (2) 中的序列 $\{x_i\}$ 收敛于 $\arcsin x$ 且 $|x_{n+1} - \arcsin x| \le \delta_n$.

证明类似于定理 1.

在 (2) 中, 取 $s_i = -sgn\left(x - z_i\right)$, 便有 $x_{n+1} \approx \arccos x$, $y_{n+1} \approx \sqrt{1 - x^2}$, $z_{n+1} \approx x$.

定理 3. 上述给出的序列 $\{x_i\}$ 收敛于 $\arccos x$, 且 $|x_{n+1} - arc\cos x| \le \delta_n$.

证明类似于定理 1.

(C) $arcshx$, $arcchx$ 的算法:

取

$$\{\delta_i\} = \left\{arcth2^{-i+[\frac{i}{5}]}\right\}_1^{+\infty},$$

$$\begin{cases} x_1 = 0, y_1 = 0, z_1 = k = \prod_{i=1}^{n} \delta_i, \\ S_i = sgn\,(x - y_i)\,, \\ x_{i+1} = x_i + s_i * \delta_i, \\ y_{i+1} = y_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * z_i, \\ z_{i+1} = z_i - s_i * 2^{-i+\left[\frac{i}{5}\right]} * y_i, \end{cases} \qquad i = 0, 1, 2\cdots, \tag{3}$$

$x_{n+1} \approx arcshx,\ y_{n+1} \approx x,\ z_{n+1} \approx ch\,(arcshx)\,.$

定理 4. (3) 中给出的序列 $\{x_i\}$ 收敛于 $arcshx$, 且有 $|x_{n+1} - arcshx| \le \delta_n$.

证明类似定理 1.

在 (3) 中取 $S_i = sgn\,(x - z_i)$. 便有 $x_{n+1} \approx arcchx,\ y_{n+1} \approx sh\,(arcchx)\,,\ z_{n+1} \approx x$.

定理 5. 由上述给出的序列 $\{x_i\}$ 收敛于 $arcchx$, 且有 $|x_{n+1} - arcchx| \le \delta_n$.

证明类似于定理 1.

(D) $\sqrt{x}$ 的算法

$$\begin{cases} x_1 = 0 \cdots y_1 = -x, \\ S_i = -sgn\,(y_i)\,, \\ x_{i+1} = x_i + s_i * 2^{-i}, \\ y_{i+1} = y_i + s_i * 2^{-i+1} * x_i + 2^{-2i}, \end{cases} \qquad i = 1, 2, \cdots, \tag{4}$$

$$x_{n+1} \approx \sqrt{x}, \quad y_{n+1} \approx 0.$$

定理 6. 由 (4) 得到的序列 $\{x_i\}$ 收敛于 $\sqrt{x}$. 且有 $|x_{n+1} - \sqrt{x}| \le 2^{-n}$.

证明类似于定理 1.

(E) $x * y$ 及 y/x 的算法

$$\begin{cases} x_1 = 0, y_1 = y, z_1 = 0, \\ S_i = sgn\,(x_i)\,, \\ x_{i+1} = x_i - s_i * 2^{-i}, \\ z_{i+1} = z_i + s_i * 2^{-i} * y_1, \end{cases} \qquad i = 1, 2, \cdots, \tag{5}$$

$$x_{n+1} \approx 0, \quad z_{n+1} \approx x * y.$$

定理 7. 由 (5) 给出的序列 $\{z_i\}$ 收敛于 $x * y$, 且有 $|z_{n+1} - x * y| \le 2^{-n} y$.

证明类似于定理 1.

$$\begin{cases} x_1 = 0,\ y_1 = x,\ z_1 = -y, \\ S_i = -sgn\,(z_i)\,, \\ x_{i+1} = x_i + s_i * 2^{-i}, \\ z_{i+1} = z_i + s_i * 2^{-i} * y_1, \end{cases} \qquad i = 1, 2, \cdots, \tag{6}$$

$$x_{n+1} \approx {}^{y}\!/_{x}, \quad z_{n+1} \approx 0.$$

定理 8. 由 (6) 给出的序列 $\{x_i\}$ 收敛于 y/x, 且有 $|x_{n+1} - y/x| \le 2^{-n}$.
证明类似于定理 1.

§2. 坐标旋转法的进一步推广

在已有的算法中, 未包含计算 $tgx, ctgx, thx, cthx$ 的算法. 它们当然可以通过 $\sin x$, $\cos x$, shx, chx 并结合 y/x 的算法来计算, 但那需要运行三次算法 (例如, 运行一次计算 sinx, 第二次计算 cosx, 第三次计算 y/x, 由此得到 tgx). 这里给出一种直接的计算方法. 这些算法都可统一到坐标旋转法中去.

(A) $tgx, ctgx$ 的算法

设 $x \in \left(-\dfrac{\pi}{2}, \dfrac{3}{2}\pi\right)$, 且 $x \ne \dfrac{\pi}{2}$. 取 $\{\delta_i\} = \{arctg2^{-i}\}$, 对给定的精度 ε >0, 设 $\delta_{N-1} < \varepsilon \le \delta_N$.

(A-1) tgx 的算法

1^o $A = 0,\ B = \prod_{i=0}^{N} \cos\delta_i,\ L = 0.$

2^o $x_1 = 0,\ y_1 = B,\ z_1 = BA.$

3^o $s_i = sgn\,(x - x_i)\,,\ x_{i+1} = x_i + s_i * \delta_i,$
$y_{i+1} = y_i + s_i * 2^{-i} * z_i,\ z_{i+1} = z_i - s_i * 2^{-i} * y_i,\ i = 1 \cdots N.$

4^o $|z_{N+1}| \le \varepsilon$? 是, 转 6^o; 否, 转 5^o.

5^o $A = A + sgn\left(x - \dfrac{\pi}{2}\right) * sgn\,(z_{N+1}) * \delta_L * B^{-1}$
$L = L + 1$
转 2^o

6^o $tgx = A$, 输出, 结束.

(A-2) $ctgx$ 的算法

1^o $A = 0,\ B = \prod_{i=0}^{N} \cos\delta_i,\ L = 0.$

2^o $x_1 = 0,\ y_1 = A,\ z_1 = B.$

3^o $s_i = sgn\,(x - x_i)\,,\ x_{i+1} = x_i + s_i * \delta_i,$
$y_{i+1} = y_i + s_i * 2^{-i} * z_i,\ z_{i+1} = z_i - s_i * 2^{-i} * y_i,\ i = 1 \cdots N.$

4^o $|z_{N+1}| \le \varepsilon$? 是, 转 6^o; 否, 转 5^o.

5^o $A = A + sgn\left(x - \dfrac{\pi}{2}\right) * sgn\,(z_{N+1}) * \delta_L * B$
$L = L + 1$
转 2^o

6^o $ctgx = A$, 输出, 结束.

(A-3) 算法的收敛性及误差估计

计算 $-a\sin x+b\cos x$ 的算法是:

$$\begin{cases} x_1=x,\ y_1=Ba,\ z_1=Bb, \\ s_i=sgn\,(x-x_i)\,, \\ x_{i+1}=x_i+s_i*\delta_i,\ y_{i+1}=y_i+s_i*2^{-i}*z_i, \\ z_{i+1}=z_i-s_i*2^{-i}*y_i,\ i=1,2,\cdots. \end{cases}$$

当 n 充分大时, 有 $x_{n+1}\approx x,\ y_{n+1}\approx a\cos x+b\sin x,\ z_{n+1}\approx -a\sin x+b\cos x$. 令 $a\equiv 1$, $Bb=A$, 则当 $z_{n+1}\approx -a\sin x+b\cos x\approx 0$ 时, 便有

$$b=\frac{A}{B}\approx\frac{\sin x}{\cos x}=tgx.$$

同样, 令 $b\equiv 1, Ba=A$, 则当 $z_{n+1}\approx 0$ 时, 有

$$a=\frac{A}{B}\approx\frac{\cos x}{\sin x}=ctgx.$$

计算过程中, 根据 x 所在象限及 z_{N+1} 的符号不断校正 A 的值, 使 $z_{N+1}\to 0$, 从而求出 tgx 和 $ctgx$.

注意到 $\sum\limits_{i=0}^{\infty}\delta_i=\sum\limits_{i=0}^{\infty}arctg2^{-i}=1.74328662$, 而 tgx 和 $ctgx$ 的值域均为 $(-\infty,+\infty)$, 所以在具体计算过程中根据情况对初始的 A 值作平移, 使 tgx 或 $ctgx\in(A-1.74328662$, $A+1.74328662)$, 这是容易自动完成的.

由以上讨论可得下列定理:

定理 9. 计算 tgx 的算法收敛. 对任意给定的正数 ε, 当 $|z_{N+1}|\le\varepsilon$, 有 $\left|\dfrac{A}{B}-tgx\right|\le \varepsilon\Big/|\cos x|$.

定理 10. 计算 ctgx 的算法收敛. 对任意给定的正数 ε, 当 $|z_{N+1}|\le\varepsilon$, 有 $\left|\dfrac{A}{B}-ctgx\right|\le \varepsilon\Big/|\sin x|$.

计算实例: 对 $x=0.7$, 取 $N=28$, 计算得 $\dfrac{A}{B}=0.842288362$ 真值为 $tg0.7=0.84228838$, 其中最后一位 8 是可疑数字.

(B) thx 和 $cthx$ 的算法

设 $x\in(-\infty,+\infty)$, 取

$$\{\delta_i\}=\left\{arcth2^{-i+\left[\frac{i}{5}\right]}\right\},$$

对给定精度 ε, 设 $\delta_{N-1}<\varepsilon\le\delta_N$.

(B-1) thx 的算法

1^o $A=0, B=\prod\limits_{i=1}^{N}ch\delta_i, L=0.$

2^o $x_1=0,\ y_1=A,\ z_1=B.$

3^o $S_i=sgn\,(x-x_i),$

$x_{i+1} = x_i + s_i * \delta_i,$

$y_{i+1} = y_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * z_i,$

$z_{i+1} = z_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * y_i,\ i = 1, 2, \cdots N.$

4^o　$|z_{N+1}| \le \varepsilon$? 是, 转 6^o; 否, 转 5^o.

5^o　$A = A + sgn\,(x) * sgn\,(z_{N+1}) * \delta_L * B^{-1}$

$L = L + 1$, 转 1^o.

6^o　$thx = -A$, 输出, 结束.

(B-2) $cthx$ 的算法

1^o　$A = 0, B = \prod\limits_{i=1}^{N} ch\delta_i, L = 0.$

2^o　$x_1 = 0,\ y_1 = B,\ z_1 = A.$

3^o　$S_i = sgn\,(x - x_i)\,,$

$x_{i+1} = x_i + s_i * \delta_i,$

$y_{i+1} = y_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * z_i,$

$z_{i+1} = z_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * y_i,\ i = 1, 2, \cdots N.$

4^o　$|z_{N+1}| \le \varepsilon$? 是, 转 6^o; 否, 转 5^o.

5^o　$A = A + sgn\,(x) * sgn\,(z_{N+1}) * \delta_L * B^{-1}$

$L = L + 1$, 转 1^o.

6^o　$cthx = -A$, 输出, 结束.

(B-3) 算法的说明和误差估计

计算 $ashx + bchx$ 的算法是:

$$\begin{cases} x_1 = 0,\ y_1 = Ba,\ z_1 = Bb, & \\ S_i = sgn\,(x - x_i)\,, & \\ x_{i+1} = x_i + s_i * \delta_i, & i = 1, 2, 3, \cdots, \\ y_{i+1} = y_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * z_i, & \\ z_{i+1} = z_i + s_i * 2^{-i+\left[\frac{i}{5}\right]} * y_i, & \end{cases}$$

n 充分大时, 有 $x_{n+1} \approx x$, $y_{n+1} \approx achx + bshx$, $z_{n+1} \approx ashx + bchx$.

类似于 tgx 和 $ctgx$ 的算法讨论, 即可得出算法, 不再赘述.

定理 11. 计算 thx 的算法收敛. 对任意给定的正数 ε, 当 $|z_{N+1}| \le \varepsilon$ 时, 有

$$\left|\left(-\frac{A}{B}\right) - thx\right| \le \varepsilon / chx \le \varepsilon.$$

定理 12. 计算 cthx 的算法收敛. 对任意给定的正数 ε, 当 $|z_{N+1}| \le \varepsilon$ 时, 有

$$\left|\left(-\frac{A}{B}\right) - cthx\right| \le \varepsilon / shx.$$

§3. 方法的进一步说明

下面对坐标旋转法作一些进一步的说明.

1. 该算法是一个统一算法, 用同一个算式, 代入不同的初始值, 即可得到不同的函数值. 用软件实现时占用空间很小, 特别在一些特殊的场合 (例如容量有限的机器上) 应用, 有其特别的优点. 笔者曾将其嵌入掌上型游戏机的游戏程序中计算函数值, 效果理想. 必要时还可用硬件实现.

2. 在计算速度上, 求一个初等函数值只需花费相当于 4~5 次乘法的时间, 所以是一种快速计算方法.

3. 如需提高精度, 只需增大算法中的 n 即可, 算法本身无改变.

4. 由于两数的相乘, 相除也可包含到本方法中, 所以高位字长 (几百位以上) 的计算技术中也可应用该技术.

5. 本算法还可推广用于做多项式计算和代数方程式求解, 以及计算富氏级数.

以上即是对一种初等函数统一算法 — 坐标旋转法, 所做的讨论.

参 考 文 献

[1] 周长林, 初等函数的统一算法及其推广, 吉林大学自然科学学报, 1(1979).

[2] 蔡耀志, 初等函数统一算法, 江苏, 浙江, 上海三省市计算数学学术交流会论文, 1984.