一个自我修正的迭代法及其收敛性*1)

黄清龙
(江苏工业学院信息科学系 江苏常州 213016)

A SELF-MODIFIED ITERATIVE METHOD AND ITS CONVERGENCE

Huang Qinglong
(*Dept. of Information Science, Jiangsu Polytechnic University, Changzhou,* 213016)

Abstract

This paper discusses a self-modified iterative method. It is a new method for simultaneously finding all roots of a nonlinear algebraic equation. The convergence and the convergence rate with higher order are obtained. The results of efficiency analysis and numerical example are satisfactory.

Key words: roots, iteration method, self-modification, convergence, efficiency

§1. 引 言

设有 n 次代数方程

$$f(x)=x^n+a_{n-1}x^{n-1}+\cdots a_1x+a_0=\prod_{i=1}^{n}(x-r_i)=0, \tag{1}$$

其中 $r_i\neq r_j\ (i\neq j)$.

作为解代数方程时牛顿法的一种改进，文 [1,6] 讨论了一个在没有重根的情况下可同时求解出 n 次代数方程 (1) 的 n 个根且 3 阶收敛的算法，其迭代公式为

$$x_i^{(k+1)}=x_i^{(k)}+\frac{\alpha_i^{(k)}}{1+\alpha_i^{(k)}\beta_i^{(k)}}, \tag{2}$$

其中

$$\alpha_i^{(k)}=-\frac{f\left(x_i^{(k)}\right)}{f'\left(x_i^{(k)}\right)}, \tag{3}$$

$$\beta_i^{(k)}=\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}, \tag{4}$$

* 2003 年 6 月 2 日收到.
1) 江苏省高校自然科学研究项目 (02KJD110001).

$x_j^{(0)}\ (j=1,2,\cdots,n)$ 是方程 (1) 的 n 个根的初始近似值.

文 [2] 以文 [3] 中的迭代法为基础讨论其四种不同的修正迭代，提高了文 [3] 中的迭代法的收敛阶. 文 [2] 实际上是利用不同的迭代函数相复合产生新的迭代函数. Gauss-Seidel 技巧是迭代函数自己与自己的结合，但通常不提高收敛阶. 受文 [2] 和 Gauss—Seidel 技巧的启发，本文将迭代公式 (2) 与它自己相结合，构造出一个新的迭代式

$$x_i^{(k+1)}=x_i^{(k)}+\frac{\alpha_i^{(k)}}{1+\alpha_i^{(k)}\gamma_i^{(k)}}, \tag{5}$$

其中

$$\gamma_i^{(k)}=\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{1}{x_i^{(k)}-u_j^{(k)}}, \tag{6}$$

$$u_j^{(k)}=x_j^{(k)}+\frac{\alpha_j^{(k)}}{1+\alpha_j^{(k)}\beta_j^{(k)}}. \tag{7}$$

$\alpha_i^{(k)},\beta_i^{(k)}(i=1,2,\cdots n)$ 仍由 (3) 式，(4) 式决定.

按照文 [2] 的观点，这是迭代公式 (2) 自己对自己进行修正. 这使得在进行第 $k+1$ 步迭代时更充分地利用了第 k 步的计算结果. 本文将证明迭代式 (5) 是收敛的而且收敛阶至少为 5, 因而比迭代式 (2) 高 2 阶敛速，而迭代式 (5) 比迭代式 (2) 新增加的主要工作量只是计算 $\gamma_i^{(k)}$.

文 [1, 2, 3, 5] 都是讨论同时求解方程 (1) 的 n 个根的方法，但都没有分析计算效率. 事实上本文的迭代式 (5) 的计算效率比文 [1, 2, 3, 5] 的迭代法都高，而且比文 [2, 3, 5] 的迭代式更简洁，在计算机上的算法实现更容易. (5) 式的数值计算结果也是满意的.

§2. 迭代法的收敛性

为了证明需要，将 (5) 改写成

$$h_i^{(k+1)}=\frac{A_i^{(k)}}{1+A_i^{(k)}}h_i^{(k)}, \tag{8}$$

其中 $h_i^{(k)}=x_i^{(k)}-r_i\,,i=1\,,2\,,\cdots\,,n;\ \ k=0\,,1\,,2\,,\cdots,$

$$A_i^{(k)}=\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{(x_i^{(k)}-r_i)(r_j-u_j^{(k)})}{(x_i^{(k)}-r_j)(x_i^{(k)}-u_j^{(k)})}. \tag{9}$$

记

$$h^{(k)}=\max_{1\leq i\leq n}\left|h_i^{(k)}\right|\ (k=0\,,\,1\,,\,2\,,\,\cdots), \tag{10}$$

$$d=\min_{1\leq i<j\leq n}|r_i-r_j|, \tag{11}$$

$$p=\frac{n-1}{(s-1)\,(s-2)}\,,\quad q=\frac{p}{1-p}. \tag{12}$$

取常数 s 满足 $s>\max\{4,n\}$, 其中 n 是代数方程 (1) 的次数 (不妨设 $n\geq 2$).

引理 1. 设 $u_j^{(k)}$ 由 (7) 式确定, 则当迭代初值 $x_j^{(0)}$ $(j=1,2,\cdots,n)$ 满足 $\left|x_j^{(0)}-r_j\right|\leq\frac{d}{s}$ 时, $\left|u_j^{(k)}-r_j\right|\leq\frac{s^2}{d^2}h^{(k)3}, j=1,2,\cdots n;\ k=0,1,2,\cdots$.

引理 1 的证明参见文 [6] 的定理 2.

定理 1. 当迭代初值 $x_j^{(0)}$ $(j=1,2,\cdots,n)$ 满足 $\left|x_j^{(0)}-r_j\right|\leq\frac{d}{s}$ 时, 由 (5) 式产生的序列 $\left\{x_i^{(k)}\right\}_{k=0}^{\infty}$ 收敛于 r_i, 且收敛阶至少为 5.

证明. 先用数学归纳法证明收敛性. 当 $i\neq j$ 时, 由引理 1 易知

$$\left|x_i^{(0)}-r_j\right|\geq(s-1)\frac{d}{s},\quad\left|r_j-u_j^{(0)}\right|\leq\frac{d}{s},$$

$$\left|u_j^{(0)}-x_i^{(0)}\right|\geq\left|r_i-r_j\right|-\left|u_j^{(0)}-r_j\right|-\left|x_i^{(0)}-r_i\right|\geq(s-2)\frac{d}{s},$$

由此可得

$$\left|A_i^{(0)}\right|\leq\sum_{\substack{j=1\\j\neq i}}^{n}\frac{\left|h^{(0)}\right|^4s^4}{(s-1)(s-2)d^4}\leq\frac{n-1}{(s-1)(s-2)}=p<\frac{1}{2},\tag{13}$$

则由 (8) 式得

$$\left|h_i^{(1)}\right|\leq\frac{\left|A_i^{(0)}\right|}{1-\left|A_i^{(0)}\right|}\left|h_i^{(0)}\right|\leq q\left|h_i^{(0)}\right|.\tag{14}$$

一般地, 设 $\left|h_j^{(k)}\right|\leq\frac{d}{s}$ $(j=1,2,\cdots,n)$, 则类似地估计可得

$$\left|A_i^{(k)}\right|=\sum_{\substack{j=1\\j\neq i}}^{n}\frac{\left|h^{(k)}\right|^4s^4}{(s-1)(s-2)d^4}\leq p<\frac{1}{2},\tag{15}$$

$$\left|x_i^{(k+1)}-r_i\right|\leq\frac{d}{s},\qquad\left|h_i^{(k+1)}\right|\leq q\left|h_i^{(k)}\right|,\tag{16}$$

这里 p, q 仍由 (12) 式确定, 且 $0<q<1$.

因此由数学归纳法知当定理的条件满足时 (15) 式, (16) 式总成立. 反复利用 (16) 式则得 $\left|h_i^{(k)}\right|\leq q^k\left|h_i^{(0)}\right|\leq\left(\frac{d}{s}\right)q^k$.

由于 $0<q<1$, 故当 $k\to\infty$ 时, $h_i^{(k)}\to 0$, 即 $x_i^{(k)}\to r_i$ $(i=1,2,\cdots,n)$.

下面讨论收敛阶. 由 (15) 式知

$$\left|A_i^{(k)}\right|\leq\frac{(n-1)s^4}{(s-1)(s-2)d^4}\left|h^{(k)}\right|^4.$$

又由 (15) 式知 $\left|A_i^{(k)}\right|<\frac{1}{2}$, 故由 (8) 式得

$$\left|h_i^{(k+1)}\right|\leq\frac{2(n-1)s^4}{(s-1)(s-2)d^4}\left|h^{(k)}\right|^5,$$

所以迭代式 (5) 是至少 5 阶收敛的. 证毕.

§3. 迭代法的效率分析和初值选择

1. 关于计算效率

在进行效率分析时，采用 $e=\dfrac{Ln\,m}{w}$ 作为迭代法的计算效率，这里 $m>1$ 是迭代法的收敛阶， w 是每一步迭代的计算工作量 [4].

我们假设用秦九韶算法计算多项式的值，这时计算一个 n 次多项式的值只需 n 次乘法运算. 我们假设 n 次乘 (或除) 法运算作为一个计算工作量单位，加减运算量忽略不计，则可以粗略地认为 $f(x_i^{(k)})$, $f'(x_i^{(k)})$, $\beta_i^{(k)}$, $\gamma_i^{(k)}$ 的计算量都为 1. 对每个根平均而言， Newton 法每迭代一步的计算量为 2. 利用文 [1] 的迭代法即 (2) 式每迭代一步的计算量为 3. 利用本文 (5) 式每迭代一步的计算量为 4. Newton 法的计算效率为 $\dfrac{Ln\,2}{2}$. 文 [1] 迭代法的计算效率为 $\dfrac{Ln\,3}{3}$. 迭代式 (5) 的计算效率为 $\dfrac{Ln\,5}{4}$. 可见本文新获得的迭代法的计算效率高于文 [1] 的迭代法，当然更高于 Newton 法的效率.

事实上虽然迭代式 (5) 的构造思想来源于文 [2], 但我们新构造的这个迭代法比文 [2, 3, 5] 中的迭代法都更简洁，计算效率也更高. 和文 [1] 中的迭代法相比，迭代式 (5) 不光提高了 2 阶敛速，而且改善了计算效率. 在利用文 [2] 的思想构造新的迭代法时，不光要注意收敛阶的提高，也要注意这样做引起的计算量的增加，把收敛阶和计算量结合起来考察是否改善了计算效率.

2. 关于迭代初值

初值选择是所有的局部收敛方法都面临的问题，比如文 [1, 2, 3, 5] 中的迭代法，本文的迭代式 (5), 甚至著名的 Newton 法都是局部收敛的，但它们仍然是重要的. 用它们求解时需要选择适当的迭代初值，通常的做法是用对初值要求低的迭代法如文 [7] 的递推二分搜索法 (recursive interval bisection search procedure), 文 [8] 中的二分法或 Bernoulli 方法求得较精确的初始近似 (虽然这些迭代法收敛慢但总是可行的), 然后再用高速收敛的迭代法计算.

特别是文 [8] 中改进的 Bernoulli 方法，是专门用于求解多项式方程的迭代法，不需要特别选取初值并且可以求出方程 (1) 的所有实根或虚根，因而可用它求出迭代式 (5) 所需初值.

另一方面，当 $|x|\geq 1$ 时显然有

$$\left|\frac{x^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0}{x^{n-1}}\right|\geq|x|-|a_{n-1}|-|a_{n-2}|-\cdots-|a_0|.$$

可见取 $R=\max\{1\,,\,|a_0|+|a_1|+\cdots+|a_{n-1}|\}$, 则当 $|x|>R$ 时,

$$|f(x)|=\left|x^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0\right|>0.$$

也就是说，方程 (1) 的全体实根都在区间 $[-R,R]$ 内，因而可用算法更加简单的二分法来确定实根的初始近似.

文 [1, 2, 3, 5] 也是讨论同时求解方程 (1) 的全部根的方法，而且都是局部收敛的，那里并没有讨论初值选择，但事实上初值选择还是非常重要的. 我们这里讨论的确定迭代初值的思想方法对它们也是适用的. 本文侧重于收敛速度和计算效率的研究，对初值选取我们只做上面简单的讨论.

顺便指出，由于本文的 (5) 式是由文 [1] 的迭代法进行自我修正而得，因此 (5) 式对迭代初值的要求并不比文 [1] 中的迭代法更苛刻. 虽然 (5) 式的计算效率高于文 [1] 的迭代法，但使文 [1] 的迭代法收敛的初值，用于 (5) 式时也同样收敛.

§4. 数值例子

为了考察文 [1] 的迭代法和 (5) 式用于实际求解代数方程的情形，我们使用这两个迭代法利用 Matlab 编程计算. 现将计算结果附于后.

例 1. 求解方程 [5] $128x^4-256x^3+160x^2-32x+1=0$, 迭代初值分别取 0, 0.3, 0.6, 1, 精度要求 10^{-12}, 计算结果见表 1.

例 2. 求解地震波理论中的 *Rayleigh* 方程 $32x^3-56x^2+24x-3=0$, 迭代初值分别取 0, 0.5, 1, 精度要求 10^{-12}, 计算结果见表 2.

表 1 求解方程 $128x^4-256x^3+160x^2-32x+1=0$ **的计算结果**

迭代公式	迭代次数 k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
文 [1] 的迭代式	0	0	0.3	0.6	1
	1	0.038461538462	0.308747673491	0.690918635171	0.963060686016
	2	0.038060233496	0.308658283776	0.691341713184	0.961939772593
	3	0.038060233744	0.308658283817	0.691341716183	0.961939766256
本文的 (5) 式	0	0	0.3	0.6	1
	1	0.038058405380	0.308657860567	0.691251235869	0.961945290150
	2	0.038060233744	0.308658283817	0.691341716183	0.961939766256

表 2 求解方程 $32x^3-56x^2+24x-3=0$ **的计算结果**

迭代公式	迭代次数 k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
文 [1] 的迭代式	0	0	0.5	1
	1	0.200000000000	0.375000000000	1.176470588235
	2	0.243808087597	0.323805689748	1.183011463175
	3	0.249955665119	0.317035707337	1.183012701892
	4	0.249999999979	0.316987298131	1.183012701892
	5	0.250000000000	0.316987298108	1.183012701892
本文的 (5) 式	0	0	0.5	1
	1	0.223048327138	0.337264150943	1.181268882175
	2	0.249914402269	0.317056482451	1.183012702162
	3	0.250000000000	0.316987298108	1.183012701892

从表 1 可见，从同样的迭代初值出发，要使全部根都达到精度要求，用文 [1] 的迭代法需要 3 步迭代，利用本文的 (5) 式只要迭代 2 步. 从表 2 可见，利用文 [1] 的迭代法需要

5 步，而利用本文的 (5) 式只需要 3 步，所以 (5) 式确实比文 [1] 的方法收敛更快. 从这里的数值例子可见，定理 1 中的条件 $\left|x_j^{(0)} - r_j\right| \le \frac{d}{s}(j = 1, 2, \cdots n)$ 只是保证收敛的充分条件，当迭代初值不满足这个条件时对有些方程而言迭代式 (5) 仍可能收敛.

参 考 文 献

[1] L W. Ehrlich, A modified Newton method for polynomials. Comm ACM, 10(1967) 107-108.

[2] De-ren Wang,Yu-jiang Wu, Some modifications of the parallel Halley iteration method and their convergence. Computing, 28(1987) 75-87.

[3] Wang Xinghua, Zheng Shiming, Parallel Halley iteration method with circular arithmetic for fining all zeros of a polynomial. A Journal of Chinese University, Numer. Math, 4(1985) 308-314.

[4] 李庆扬，莫孜中，祁力群，非线性方程组的数值解法，科学出版社， 1987, 35-37

[5] 张志海，田伶改，求无重根时代数方程根的一种数值迭代方法. 高校计算数学学报， 23:1 (2001) 38-44.

[6] 黄清龙，解代数方程时牛顿法的一种改进. 应用数学， 8(增) (1995) 73-76.

[7] R. E. Moore, S. T. Jones, Safe starting regions for iterative methods. SIAM J. Numer. Anal., 14:6 (1977), 1051-1065.

[8] 曹志浩，张玉德，李瑞遐，矩阵计算和方程求根，人民教育出版社， 1979, 230-240.