2012 International Conference on System Modeling and Optimization (ICSMO 2012)
IPCSIT vol. 23 (2012) © (2012) IACSIT Press, Singapore

A Fast Algorithm for Solving Nonlinear Equations

Xiaodong Wang [1], Lei Wang [2]

[1] Quanzhou Normal University, Quanzhou 362000, China

[2] Georgia Institute of Technology, Atlanta GA 30332, USA

Abstract. In this work, we develop a simple yet practical algorithm for finding a root of a real function $f'(x)$ with a good local convergence. The algorithm uses a continued fraction interpolation method that can be easily implemented in software packages for achieving desired convergence orders. For the general *n*-point formula, the order of convergence rate of the presented algorithm is τ_n , the unique positive root of the equation $x^n - x^{n-1} - \cdots - x - 1 = 0$ .Computational results ascertain that the developed algorithm is efficient and demonstrate equal or better performance as compared with other well known methods.

Keywords: Iterative methods, higher order, convergence rate, algorithm

1. Introduction

One of the most frequently occurring problems in scientific work is to locate a real root α of a nonlinear equation

$$f'(x) = 0 \tag{1}$$

Various problems arising in diverse disciplines of engineering, sciences and nature can be described by a nonlinear equation of the form (1) . For example, many optimization problems and boundary value problems appearing in many applied areas are reduced to solving the preceding equation. Therefore solving the equation (1) is a very important task and there exists numerous methods for solving the above equation [9, 13, 6, 11, 12]. Though there exists many iterative methods but still the Newtons method, one of the best known and the probably the oldest, is extensively used for solving the nonlinear equation (1) [3, 5, 10]. Many methods have been developed which improve the convergence rate of the Newton's method [4, 10, 7, 1, 2, 8].

In this paper, we present an iterative algorithm for the problem based on the continued fraction interpolation method. For the general iteration scheme of n points, the algorithm constructs the $n+1$ th point according to the information given by the previous n points. Thus the algorithm has a good local convergence. Furthermore, the scheme of the algorithm is very simple and easy to implement on a computer. The numerics experiments indicate that the scheme of the algorithm even for three points converges very fast.

2. The Design of the Algorithm

Let $f \in C^{(1)}[a,b], y(x) = f'(x)$. We want to find $x^* \in (a,b)$ such that $y(x^*) = f'(x^*) = 0$. Suppose we are given n approximated values $x_0, x_1, \cdots, x_{n-1}$ of x^* and the function f with corresponding derivatives $y_i = y(x_i), 0 \le i < n$.

In order to find the real root of y , we construct the following scheme of n points based on the continued fraction interpolation of $y = f'(x)$ as follows.

$$\psi(y) = a_0 + \cfrac{y - y_{n-1}}{a_1 + \cfrac{y - y_{n-2}}{\ddots \cfrac{y - y_2}{a_{n-2} + \cfrac{y - y_1}{a_{n-1}}}}} \tag{2}$$

The k th order difference of function f can be defined recursively as follows.

$$\begin{cases} \Phi_0(y) = x \\ \Phi_k(y_1, \cdots, y_k, y) = \dfrac{y - y_k}{\Phi_{k-1}(y_1, \cdots, y_{k-1}, y) - \Phi_{k-1}(y_1, \cdots, y_k)}, k = 1, 2, \cdots \end{cases} \tag{3}$$

It follows from the inverse interpolation condition $\psi(y_i) = x_i, 0 \le i < n$ that

$$\begin{cases} a_0 = x_{n-1} = \Phi_0(y_{n-1}) \\ a_i = \Phi_i(y_{n-1}, \cdots, y_{n-i-1}), i = 1, 2, \cdots, n-1 \end{cases} \tag{4}$$

It follows that

$$\psi(0) = a_0 - \cfrac{y_{n-1}}{a_1 - \cfrac{y_{n-2}}{\ddots \cfrac{y_2}{a_{n-2} - \cfrac{y_1}{a_{n-1}}}}} \tag{5}$$

According to the formula described above, we can design a very simple iteration algorithm to find x^* as follows.

Algorithm 2.1: Inverse-Interpolation(f)

Input $x_0, x_1, \cdots, x_{n-1}, \varepsilon$

for $i \leftarrow 0$ **to** $n-1$

do $y_i \leftarrow f'(x_i)$

while $y_{n-1} > \varepsilon$

do $\begin{cases} g(y_{n-1}, \cdots, y_1) \\ x_n \leftarrow a_0 - \cfrac{y_{n-1}}{a_1 - \cfrac{y_{n-2}}{\ddots \cfrac{y_2}{a_{n-2} - \cfrac{y_1}{a_{n-1}}}}} \\ y_n \leftarrow f'(x_n) \\ n \leftarrow n+1 \end{cases}$

Output $(x_n, y_n, f(x_n))$

where the function $g(y_{n-1}, \cdots, y_1)$ is used to compute the coefficients $a_0, \cdots, a_{n-1}$ according to the formula (4) .

3. The Convergence of the Algorithm

Suppose $f \in C^{(1)}[a,b], x \in [a,b]$. If for $x_i \in [a,b], 0 \le i \le k$, the limit $\lim\limits_{x_0,\cdots,x_k \to x} \Phi_k(f'(x_0),\cdots,f'(x_k)) = \Phi_k(x)$ exists, then the function f is k th inverse differentiable and $\Phi_k(x)$ is called the k th inverse derivative of f at x. (6)

Let $x^* \in [a,b]$ and $f'(x^*) = 0$. The function type $T^{(n)}[a,b] \subseteq C^{(1)}[a,b]$ can be defined as

$$T^{(n)}[a,b] = \{f \mid f \text{ } k\text{th inverse differentiable in } [a,b], \Phi_k(x^*) \ne 0, 1 \le k \le n\}$$

Suppose for any function $f \in T^{(n)}[a,b]$, the point sequence generated by algorithm 2.1 be $\{x_k\}_1^\infty$. From the computing steps of the algorithm we know x_{k+1} can be formulated as

$$x_{k+1} = a_0 - \cfrac{y_k}{a_1 - \cfrac{y_{k-1}}{\ddots \cfrac{y_{k-n+3}}{a_{n-2} - \cfrac{y_{k-n+2}}{a_{n-1}}}}}$$

where $a_0 = x_k = \Phi_0(y_k)$, and $a_i = \Phi_i(y_k,\cdots,y_{k-i}), i = 1,2,\cdots,n-1$.

Now, we can compte $\psi_1(y)$, the $n+1$ points inverse continued fraction interpolation of $y = f'(x)$ at $x^*, x_{k-n+1},\cdots,x_k$ as follows

$$\psi_1(y) = a'_0 + \cfrac{y - y_k}{a'_1 + \cfrac{y - y_{k-1}}{\ddots \cfrac{y - y_{k-n+3}}{a'_{n-2} + \cfrac{y - y_{k-n+2}}{a'_{n-1} + \cfrac{y - y_{k-n+1}}{a'_n}}}}} \tag{7}$$

It follows from the inverse interpolation condition $\psi_1(y_i) = x_i, k-n+1 \le i \le k$, and $\psi_1(0) = x^*$ that $a'_i = a_i, 0 \le i < n$, and $a'_n = \Phi_n(y_k,\cdots,y_{k-n+1},0)$.

It follows that

$$\psi_1(0) = a_0 - \cfrac{y_k}{a_1 - \cfrac{y_{k-1}}{\ddots \cfrac{y_{k-n+2}}{a_{n-1} - \cfrac{y_{k-n+1}}{a_n'}}}} = x^* \tag{8}$$

Denote $x^* = \dfrac{P_n}{Q_n}$, and $x_{k+1} = \dfrac{P_{n-1}}{Q_{n-1}}$. It is not difficult to see that

$$|x_{k+1} - x^*| = \left|\frac{P_{n-1}}{Q_{n-1}} - \frac{P_n}{Q_n}\right| = \frac{\left|\prod_{i=1}^{n} y_{k-n+i}\right|}{|Q_{n-1}Q_n|}$$

It follows that

$$|x_{k+1} - x^*| = \prod_{i=1}^{n} |x_{k-n+i} - x^*| \left|\frac{\prod_{i=1}^{n} \Phi_1(y_{k-n+i},0)}{Q_{n-1}Q_n}\right| \tag{9}$$

Theorem 1 *Let* $f \in T^n[a,b]$. *There must be* $\varepsilon > 0$ *and* $\delta_1 > 0$ *such that when* $\{x_{k-n+1}, \cdots, x_k\} \subset o(x^*, \delta_1)$, $| Q_{n-1} Q_n | \geq \varepsilon$.

Theorem 2 *Let* $f \in T^n[a,b]$. *There must be* $\delta > 0$ *such that when* $\{x_0, \cdots, x_{n-1}\} \subset o(x^*, \delta)$, *the sequence* $\{x_k\}_1^\infty$ *generated by the algorithm 2.1 converges to* x^* .

4. References

[1] S. Amat, C. Bermudez, S. Busquier, D. Mestiri, A family of Halleys Chebyshev iterative schemes for non-Frchet differentiable operators, J. Comput. Appl. Math. , 216, 486-493, 2009.

[2] S. Amat, S. Busquier, J.M. Gutierrez, Geometric constructions of iterative functions to solve nonlinear equations, J. Comput. Appl. Math. , 157, 197-205, 2003.

[3] I. K. Argyros, Computational theory of iterative methods, Studies in Computational Mathematics , ed. Chui, C.K., Wuytack, L., Elsevier, New York, 2007.

[4] W. Bi, H. Ren, Q. Wu, Three-step iterative methods with eighth-order convergence for solving nonlinear equations, J. Comput. Appl. Math. , 225, 105-112, 2009.

[5] C. Chun, Y. Ham, Some sixth-order variants of Ostrowski root-finding methods, Appl. Math. Comput. , 193, 389-394, 2003.

[6] C. Chun, Construction of Newton-like iteration methods for solving nonlineare quations, Numer. Math. , 104, 297-315, 2006.

[7] V. Kanwar, S.K. Tomar, Modified families of Newton, Halley and Chebyshev methods, Appl. Math. Comput. , 192, 20-26, 2007.

[8] J. Kou, Some variants of Cauchy's method with acceleratedfourth-order convergence, J. Comput. Appl. Math. , 213, 71-78, 2008.

[9] B. Neta, On a family of multipoint methods for non-linear equations, Int. J. Comput. Math. , 9, 353-361, 1981.

[10] N. Osada, Chebyshev-Halley methods for analytic functions, J. Comput. Appl. Math. , 216, 585-599, 2008.

[11] H. Ren, Q. Wu, W. Bi, New variants of Jarratts method with sixth-order convergence, Numer. Algorithms , 52, 585-603, 2009.

[12] J.R. Sharma, R.K. Guha, A family of modified Ostrowski methods with accelerated sixth order convergence, Appl. Math. Comput. , 190, 111-115, 2007.

[13] J.F. Traub, Iterative Methods for the Solution of Equations, Chelsea Publishing Company, New York, 1977.