An Efficient Continued Fraction Algorithm for Nonlinear Optimization and Its Computer Implementation

Daxin Zhu[a], Xiaodong Wang[a,b,*]
[a]Faculty of Mathematics & Computer Science, Quanzhou Normal University, China
Email: dex@qztc.edu.cn
[b] Faculty of Mathematics & Computer Science, Fuzhou University, China
Email: wangxiaodong@qztc.edu.cn

Abstract— **Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences. There has been much attention to develop iterative methods for solving nonlinear equations in these years. New algorithms and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace. One of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Among wide various of papers have been published in the recent years, there are some progress about multi-step methods. These multi-step methods have been suggested by combining the well-known Newton's method with other methods.**

In this work, we develop a simple yet practical algorithm for solving nonlinear optimization problems by solving nonlinear equations with a good local convergence. The algorithm uses a continued fraction interpolation 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.

Index Terms— **optimization, algorithms, nonlinear equations, convergence rate, continued fraction**

I. INTRODUCTION

OPTIMIZATION is a mature and very successful area of applied mathematics. We often use the methods of optimization to find the best element in a set. There are different fields of the optimization theory: linear, integer, optimal control, semi-infinite programming, etc. In these fields, several directions such as optimality conditions, duality theory, sensitivity analysis, and numerical methods are often studied.

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 exist numerous methods for solving the above equation [6], [9], [11]–[13].

There are many applications of optimization methods, including engineering, management sciences, computer science, economics and statistics. Optimization problems arise in facility location, structure design, experiment design, radiation therapy, asset valuation, image reconstruction and among others.

Though there exists many iterative methods but still the Newton's 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 [1], [2], [4], [7], [8], [10].

The theory of continued fractions is a well-developed branch of mathematics. The subject of continued fractions has received considerable attention in the areas of mathematics as well as computer science. We consider a root finding problem for a nonlinear equation (1). In recent years many iterative methods have been proposed [16]–[18]. These methods are mostly based on the well-known Newton's method. When an initial approximation is not properly chosen or the function $f(x)$ behaves pathologically near the root, however, approximate roots obtained by the iterative methods tend to converge very slowly or even fail to converge. In addition, derivatives of $f(x)$ are necessary in most higher order iterative methods. To overcome the necessity for derivatives of $f(x)$ and the problem of choosing an initial approximation, some simple iterative methods have recently been introduced. Convergence orders of these methods were shown to be quadratic, but yet the convergence rate is more or less slow in some particular cases.

Manuscript received August 10, 2012; revised January 2, 2013; accepted April 16, 2013. © 2005 IEEE.

*Corresponding author. This work was supported in part by the Natural Science Foundation of Fujian (Grant No.2013J0101), and the Haixi Project of Fujian (Grant No.A099).

doi:10.4304/jcp.9.4.883-890

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 numeric experiments indicate that the scheme of the algorithm even for three points converges very fast.

The organization of the paper is as follows.

In the following 6 sections we describe the algorithms and our computing experience with the algorithms for solving the nonlinear optimization problems. Before we talk about the design ideas of our new algorithm, there is some necessary background to go through. We provide most of this information mainly for reference purposes. Some necessary preliminaries and notations are introduced in section 2. In section 3 we describe a new iterative algorithm for solving the nonlinear optimization problems and their correctness and complexities. The convergence properties of the presented algorithm are discussed in section 4. The convergence rate, the most important part of the algorithm, is discussed in section 5. In section 6 we give a computational study of the presented algorithms which demonstrates that the achieved results are not only of theoretical interest, but also that the techniques developed may actually lead to practical algorithms. Some concluding remarks are in section 7.

A preliminary conference version of this paper was presented at International Conference on System Modeling and Optimization (ICSMO 2012) [16]. In this paper the correctness, complexities and the convergence rate are proved rigorously, but not just stated in intuitively. More experiment details are described in this version of the paper.

II. Preliminaries

A nonlinear optimization model consists of the optimization of a function subject to constraints, where any of the functions may be nonlinear. This is the most general type of model, including other types of models as special cases. Nonlinear optimization models arise often in science and engineering.

Consider the unconstrained optimization problem

$$\min_{x \in R^n} f(x) \tag{2}$$

where $f(x)$ is a continuously differentiable function. We know that if a point x^* is a local minimum of the problem, then

$$\nabla f(x^*) = 0 \tag{3}$$

If the function $f(x)$ has a simple form, we can find all local minima by solving this equation. But if the description of the function $f(x)$ is complex, then the approach by the necessary conditions of optimality may fail, especially in the case when $f(x)$ is defined algorithmically. In this situation, an iterative method is often employed to solve the problem. The iterative method constructs a sequence of points $x_k, k = 0, 1, 2, ...$, so that it converges to a solution of the problem. The points x_k are considered as approximations of a solution to the problem. In practical applications, we stop the calculations at some iteration k and we accept x_k as a sufficiently good approximation of a solution. It is clearly the first thing to be addressed, although the condition is neither necessary nor sufficient for a convergent iterative method.

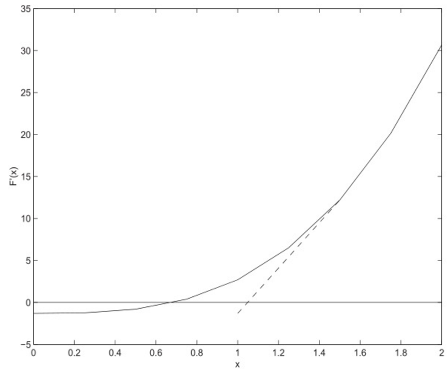

Figure 1. Geometric explanation of Newton's method

A wide class of problems which arises in various disciplines can be formulated by the nonlinear equations of the form $f(x) = 0$.

The Newton's method [13], [15] is one of the best iterative methods for solving a single nonlinear equation by using the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4}$$

which converges quadratically in some neighborhood of x^*. It is well known that Newton's method has a problem that the condition $f'(x) \neq 0$ in a neighborhood of root x^* is severe for its convergence.

The algorithm can be viewed as using the tangent to the curve of $f'(x)$ to predict where $f(x)$ itself becomes zero (see Fig. 1).

The iterations of the method generate x_{k+1} as a stationary point of the quadratic function defined by the values of $f(x_k)$, $f'(x_k)$ and $f''(x_k)$. Another method is a direct search iterative method based on locating the minimum of the quadratic defined by values of f at three points x_k, x_{k-1}, x_{k-2} and a gradient approach could minimize the local quadratic approximation of $f(x_{k-1})$, $f'(x_{k-1})$ and $f(x_k)$. If a minimum x_{k+1} is found close enough to x^*, then the iteration terminates; otherwise x_{k+1} is used to generate a new quadratic model to predict a new minimum.

To choose a direction d_k and then to make a step in this direction, we set $x_{k+1} = x_k + \tau_k d_k$ with some step size coefficient $\tau_k > 0$. This process is illustrated in Fig. 2. After this step, a new direction d_{k+1} can be chosen at the point x_{k+1}, and the method continues.

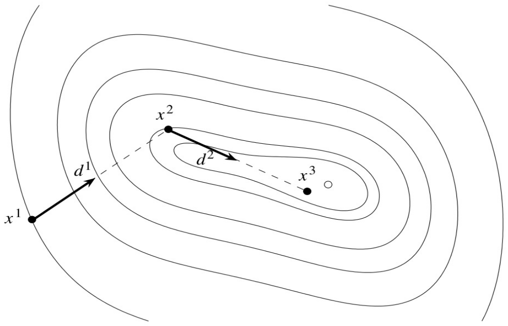

Figure 2. Moving directions

The cases where the interpolated quadratic has negative curvature and thus does not have a minimum must be treated in the practical implementation of this basic idea. To locate a group of points which implies a suitable quadratic model, the bracketing algorithm is often used. Another method is based on the repeated location of the minimum of a cubic polynomial fitted either to values of f at four or two points. Although the method is often fast, it also must avoid the search attracted to a maximum of the interpolating polynomial.

Many iterative methods with cubic convergence have been developed in recent years. However, these modified Newton's methods also suffer the problem which restricts their applications. The research of the new methods free from the restriction of derivatives are required for practical applications.

Continued fractions have a long history and applications. The origin of the continued fractions may be placed in the age of Euclid's algorithm for the greatest common divisor [14], [15]. Continued fractions experience a revival nowadays thanks to their applications in high speed computer arithmetics. Faster division and multiplication, fast and precise evaluation of functions, precise representation of transcendental numbers are the advantages of continued fractions in computer arithmetics.

We can give some basic definitions and results here. The integer part of a real number x is denoted as $\lfloor x \rfloor$.

Definition 1: $(a_n)_{n\in\mathbb{N}}$, the continued fraction expansion of a real number x can be obtained by the following algorithm

$$x_0 = x, \quad a_n = \lfloor x_n \rfloor, \quad x_{n+1} = \begin{cases} \frac{1}{x_n - a_n} & \text{if } x_n \notin \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases}$$

Where $a_0 \in \mathbb{Z}$ and $a_n \in \mathbb{N}$. The algorithm producing the continued fraction is closely related to the Euclidean algorithm for computing the greatest common divisor of two integers. It is thus readily seen that if the number x is rational, the algorithm eventually produces zeroes. There exists $N \in \mathbb{N}$ such that $a_n = 0$ for all $n > N$, and thus

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_{N-1} + \cfrac{1}{a_N}}}}} \tag{5}$$

We write $x = [a_0, \ldots, a_N]$. On the other hand, if we want to find an expression with $a_0 \in \mathbb{Z}$ and otherwise $a_n \in \mathbb{N} \setminus \{0\}$, the continued fraction $[a_0, \ldots, a_N]$ and

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_{N-1} + \cfrac{1}{(a_N - 1) + \cfrac{1}{1}}}}}}$$

If the number x is irrational, then the sequence of the so called *convergent*

$$a_0, \; a_0 + \frac{1}{a_1}, \; \ldots, a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}}} \tag{6}$$

converges to x for $n \to \infty$. On the other hand, every sequence of rational numbers with $a_0 \in \mathbb{Z}$ and $a_n \in \mathbb{N} \setminus \{0\}$ converges to an irrational number. We write $x = [a_0, \ldots, a_n, \ldots]$.

Let $x \in \mathbb{R} \setminus \mathbb{Q}$ and let $\frac{p_n}{q_n}$ be its n-th convergent (where p_n and q_n are coprime) and let $\frac{p}{q}$ with $p, q \in \mathbb{Z}$ be distinct from $\frac{p_n}{q_n}$ and such that $0 < q \leq q_n$. Then

$$\left| x - \frac{p_n}{q_n} \right| < \left| x - \frac{p}{q} \right|.$$

How well the continued fractions approximate irrational numbers is also to be treated.

Let $x \in \mathbb{R} \setminus \mathbb{Q}$ and let $\frac{p_n}{q_n}$ be its n-th convergent (where p_n and q_n are coprime). Then either

$$\left| x - \frac{p_n}{q_n} \right| < \frac{1}{2q_n^2} \text{ or } \left| x - \frac{p_{n+1}}{q_{n+1}} \right| < \frac{1}{2q_{n+1}^2}.$$

Continued fractions can get very close to irrational numbers in a certain way.

Let $x \in \mathbb{R} \setminus \mathbb{Q}$ and let $\frac{p}{q}$ with $p, q \in \mathbb{Z}$ satisfy $|x - \frac{p}{q}| < \frac{1}{2q^2}$. Then $\frac{p}{q}$ is a convergent of x.

The convergent of continued fractions is closely related to the so-called *continuant* $K_n(x_1, \ldots, x_n)$.

Let $x_0 \in \mathbb{R}, \; x_i > 0, \; i \in \mathbb{N}$. Then it holds

$$x_0 + \cfrac{1}{x_1 + \cfrac{1}{\ddots + \cfrac{1}{x_n}}} = \frac{K_{n+1}(x_0, x_1, \ldots, x_n)}{K_n(x_1, \ldots, x_n)}$$

where the polynomial $K_n(x_1, \ldots, x_n)$ is given by the recurrence relation $K_{-1} = 0$, $K_0 = 1$ and for $n \geq 1$ $K_n(x_1, \ldots, x_n) = K_{n-2}(x_1, \ldots, x_{n-2}) + x_n K_{n-1}(x_1, \ldots, x_{n-1})$.

For every $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathbb{R}$, we have

$$K_n(x_1, \ldots, x_n) = K_n(x_n, \ldots, x_1).$$

Let $[a_0, \ldots, a_n, \ldots]$ be the continued fraction of an irrational number x. Then its n-th convergent $\frac{p_n}{q_n}$ satisfies

$$p_n = K_{n+1}(a_0, \ldots, a_n), \quad q_n = K_n(x_1, \ldots, x_n).$$

III. The Design of the Algorithm

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

To find the real roots of y, The following scheme of n points based on the continued fraction interpolation of $y = f'(x)$ can be constructed as follows.

$$\psi(y) = a_0 + \cfrac{y - y_{n-1}}{a_1 + \cfrac{y - y_{n-2}}{\ddots \cfrac{y - y_2}{a_{n-2} + \cfrac{y - y_1}{a_{n-1}}}}} \tag{7}$$

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

$$\begin{cases} \Phi_0(y) = x \\ \Phi_k(y_1, \cdots, y_k, y) = \\ \dfrac{y - y_k}{\Phi_{k-1}(y_1, \cdots, y_{k-1}, y) - \Phi_{k-1}(y_1, \cdots, y_k)}, \\ k = 1, 2, \cdots \end{cases} \tag{8}$$

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

$$\begin{cases} a_0 = x_{n-1} = \Phi_0(y_{n-1}) \\ a_i = \Phi_i(y_{n-1}, \cdots, y_{n-i-1}), \\ i = 1, 2, \cdots, n-1 \end{cases} \tag{9}$$

It follows that

$$\psi(0) = a_0 - \cfrac{y_{n-1}}{a_1 - \cfrac{y_{n-2}}{\ddots \cfrac{y_2}{a_{n-2} - \cfrac{y_1}{a_{n-1}}}}} \tag{10}$$

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

Algorithm III.1: CONTINUED-FRACTION(f)

input $x_0, \cdots, x_{n-1}, \varepsilon$
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 (9).

IV. The Convergence

Suppose $f \in C^{(1)}[a,b], x \in [a,b]$.
If for $x_i \in [a,b], 0 \leq i \leq k$, the limit

$$\lim_{x_0, \cdots, x_k \to x} \Phi_k(f'(x_0), \cdots, f'(x_k)) = \Phi_k(x)$$

exists, then the function f is kth inverse differentiable and $\Phi_k(x)$ is called the kth inverse derivative of f at x.

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

$T^{(n)}[a,b] =$
$\{f \mid f$ kth inverse differentiable in $[a,b]$,
$\Phi_k(x^*) \neq 0, 1 \leq k \leq n\}$

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

$$x_{k+1} = a_0 - \cfrac{y_k}{a_1 - \cfrac{y_{k-1}}{\ddots \cfrac{y_{k-n+3}}{a_{n-2} - \cfrac{y_{k-n+2}}{a_{n-1}}}}} \tag{11}$$

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

The $n+1$ point inverse continued fraction interpolation of $y = f'(x)$ at $x^*, x_{k-n+1}, \cdots, x_k$ can be computed as follows

$$\psi_1(y) = a'_0 + \cfrac{y - y_k}{a'_1 + \cfrac{y - y_{k-1}}{\ddots \cfrac{y - y_{k-n+3}}{a'_{n-2} + \cfrac{y - y_{k-n+2}}{a'_{n-1} + \cfrac{y - y_{k-n+1}}{a'_n}}}}} \tag{12}$$

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

It follows that

$$\psi_1(0) = a_0 - \cfrac{y_k}{a_1 - \cfrac{y_{k-1}}{\ddots \cfrac{y_{k-n+2}}{a_{n-1} - \cfrac{y_{k-n+1}}{a'_n}}}} = x^* \quad (13)$$

Denote $x^* = \frac{P_n}{Q_n}$, and $x_{k+1} = \frac{P_{n-1}}{Q_{n-1}}$. 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

$$|x_{k+1} - x^*| = \prod_{i=1}^{n} |x_{k-n+i} - x^*| \left|\frac{\prod_{i=1}^{n} \Phi_1(y_{k-n+i}, 0)}{Q_{n-1}Q_n}\right| \quad (14)$$

where Q_{n-1} and Q_n cab be calculated recursively as follows

$$\begin{cases} Q_{-1} = 0 \\ Q_0 = 1 \\ Q_i = a_i Q_{i-1} - y_{k-i+1} Q_{i-2}, \quad i \geq 1 \end{cases} \quad (15)$$

$$\begin{cases} a_i = \Phi_i(y_k, \cdots, y_{k-i}) & 1 \leq i < n \\ a_n = \Phi_n(y_k, \cdots, y_{k-n+1}, 0) \end{cases} \quad (16)$$

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

Proof.

It follows from $f \in T^n[a,b]$ that

$$\Phi_k(x^*) \neq 0, 1 \leq k \leq n.$$

Therefore, there must be $\delta_1 > 0$, $m_1 > 0$ and $M_1 > \frac{1}{2}$ such that when

$$\{x_{k-n+1}, \cdots, x_k\} \subset o(x^*, \delta_1),$$

$$0 < m_1 \leq |\Phi_i(y_k, \cdots, y_{k-i})| = |a_i| \leq M_1,$$

and

$$|y_{k-i+1}| \leq \frac{m_1}{2}\left(\frac{m_1}{4M_1}\right)^n, 1 \leq i \leq n.$$

It can be proved by induction that

$$\left(\frac{m_1}{2}\right)^i \leq |Q_i| \leq (2M_1)^i, 1 \leq i \leq n \quad (17)$$

For the case of $i = 1$ we have,

$$\left(\frac{m_1}{2}\right) \leq m_1 \leq |Q_1| \leq M_1 \leq 2M_1.$$

If the formula (17) is true for the case of $i \leq R$. In the case of $i = R + 1$, we have,

$$\begin{aligned} &Q_{R+1} = |a_{R+1}Q_R - y_{k-R}Q_{R-1}| \\ &\leq |a_{R+1}||Q_R| + |y_{k-R}||Q_{R-1}| \\ &\leq M_1(2M_1)^R + M_1(2M_1)^{R-1} \leq (2M_1)^{R+1} \end{aligned}$$

and

$$\begin{aligned} &Q_{R+1} \geq |a_{R+1}||Q_R| - |y_{k-R}||Q_{R-1}| \\ &\geq m_1\left(\tfrac{m_1}{2}\right)^R - \tfrac{m_1}{2}\left(\tfrac{m_1}{aM_1}\right)^R (2M_1)^{R-1} \\ &\geq m_1\left(\tfrac{m_1}{2}\right)^R - \tfrac{m_1}{2}\left(\tfrac{m_1}{aM_1}\right)^R (2M_1)^R \\ &= \left(\tfrac{m_1}{2}\right)^{R+1} \end{aligned}$$

By induction we know that the formula (17) is true for $k = 1, \cdots, n$.

Therefore,

$$\begin{aligned} &|Q_{n-1}Q_n| \geq \left(\tfrac{m_1}{2}\right)^{n-1}\left(\tfrac{m_1}{2}\right)^n \\ &= \left(\tfrac{m_1}{2}\right)^{2n-1} \triangleq \varepsilon > 0 \end{aligned}$$

■

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^*.

Proof.

It follows from Theorem 1 that there exist $\epsilon > 0$ and $\delta_1 > 0$ and $M_1 > 1/2$ such that when

$$\{x_0, \cdots, x_{n-1}\} \subset o(x^*, \delta_1),$$

$|\Phi_1(y_i, 0)| \leq M_1, 0 \leq i < n$, and $|Q_{n-1}Q_n| \geq \varepsilon$.

If we set $M = \frac{M_1^n}{\epsilon} + 1$, then it follows from formula (14) that when $\{x_0, \cdots, x_{n-1}\} \subset o(x^*, \delta_1)$,

$$|x_n - x^*| \leq M \prod_{i=0}^{n-1} |x_i - x^*|.$$

Take $0 < \delta < \min\{\delta_1, \frac{1}{\sqrt[n-1]{M}}\}$, we then have when $\{x_0, \cdots, x_{n-1}\} \subset o(x^*, \delta)$,

$$|x_n - x^*| \leq M \prod_{i=0}^{n-1} |x_i - x^*| \leq M\delta^n < \delta,$$

and thus $x_n \in o(x^*, \delta)$.

It follows by induction that

$$\{x_k\}_1^\infty \subset o(x^*, \delta) \subset o(x^*, \delta_1).$$

On the other hand, it follows from Theorem 1 that

$$|x_{k+1} - x^*| \leq M \prod_{i=1}^{n} |x_{k-i+1} - x^*|, k \geq n \quad (18)$$

Thus

$$|x_{k+1} - x^*| \leq M\delta^{n-1}|x_k - x^*| \leq (M\delta^{n-1})^k, k \geq n.$$

It follows from $M\delta^{n-1} < 1$ that $\lim_{k\to\infty} |x_k - x^*| = 0$. This means the algorithm converges.■

V. The Convergence Rate of the Algorithm

Theorem 3: The equation

$$f_n(x) = x^n - x^{n-1} - \cdots - x - 1 = 0, (n > 1)$$

has a unique positive root τ_n, and

$$1 < \tau_n < \tau_{n+1} < 2, \lim_{n\to\infty} \tau_n = 2.$$

Proof.

Since $f_n(1) = 1 - n < 0$ and $f_n(2) = 1 > 0$, there exists $\tau_n \in (1, 2)$ such that $f_n(\tau_n) = 0$. By Descartes' sign-rule we know that the number of positive roots of the equation $f_n(x) = 0$ is $p \leq 1$ due to the change of signs of the coefficients of $f_n(x)$.

In view of the fact that τ_n is a positive root of $f_n(x) = 0$, we know $p = 1$. This means $f_n(x) = 0$ has a unique positive root τ_n, and $1 < \tau_n < 2$.

Since

$$f_{n+1}(\tau_n) = \tau_n f_n(\tau_n) - 1 = -1 < 0$$

and $f_{n+1}(2) = 1 > 0$, $f_{n+1}(x) = 0$ has a root in $(\tau_n, 2)$, i.e.

$$1 < \tau_n < \tau_{n+1} < 2 \tag{19}$$

Therefore the sequence $\{\tau_n\}$ is monotone increasing and bounded above.

Let $\lim_{n\to\infty} \tau_n = \tau$. Then $\tau \leq 2$. Now, we will show $\tau = 2$.

If $\tau < 2$, then for a sufficiently large n we have

$$\begin{aligned} f_n(\tau) &= \tau^n - \tau^{n-1} - \cdots - \tau - 1 \\ &= \tau^n - \frac{\tau^n - 1}{\tau - 1} \\ &= \frac{(\tau-2)\tau^n + 1}{\tau - 1} < 0 \end{aligned}$$

We have already had $f_n(2) = 1 > 0$. Thus, $\tau_n \in (\tau, 2)$. That means$\tau_n > \tau$. This contradicts the fact that τ_n is monotone increasing and tends to τ.

Therefore $\tau = 2$.■

Theorem 4: Let ω be an iteration procedure and let $C(\omega, x^*)$ be the set of the sequences which converges to x^* and is produced by ω. M is a positive constant. If for any $\{x_k\} \in C(\omega, x^*)$, there is a positive integer $k_0 \geq n$ such that

$$\|x_{k+1} - x^*\| \leq M \prod_{i=1}^{n} \|x_{k-i+1} - x^*\|$$

when $k \geq k_0$, then $O_R(\omega, x^*) \geq \tau_n$, where τ_n is the unique positive root of the equation

$$f_n(x) = x^n - x^{n-1} - \cdots - x - 1 = 0.$$

Proof.

Set $\epsilon_k = \|x_k - x^*\|$.

Then for all $k \geq k_0$, $\epsilon_{k+1} \leq M\epsilon_k \cdots \epsilon_{k-n+1}$.

Set $\eta_k = M^{\frac{1}{n-1}} \varepsilon_k$. Then,

$$\eta_{k+1} \leq \eta_k \eta_{k-1} \cdots \eta_{k-n+1} \tag{20}$$

Since $\epsilon_k \to 0$, there exists a positive integer k' and a positive number η such that for all $k \geq k'$, $\eta_k \leq \eta < 1$. Now, we will show by induction that

$$\eta_{k'+i} \leq \eta^{\mu_i}, i = 1, 2, \cdots \tag{21}$$

where μ_i are defined by the following difference equation:

$$\begin{cases} \mu_1 = \mu_2 = \cdots = \mu_n = 1 \\ \mu_{i+1} = \mu_i + \mu_{i-1} + \cdots + \mu_{i-n+1}, \\ i = n, n+1, \cdots \end{cases} \tag{22}$$

For $i = 1, 2, \cdots, n$, the formula (21) is trivial. Let the formula (21) is true for $n \leq m \leq i$. Then when $i = m+1$, it follows from the formula (20) that

$$\eta_{k'+m+1} \leq \eta_{k'+m} \cdots \eta_{k'+m-n+1} \leq \eta^{\mu_m + \cdots + \mu_{m-n+1}}$$

By induction we know that the formula (21) holds.

Now, we will show μ_i satisfy

$$\mu_i \geq \alpha\tau_n^i, i = 1, 2, \cdots \tag{23}$$

where $\alpha = \tau_n^{-n}$.

In fact, since $\tau_n > 1$, the formula (23) is obviously true for $i = 1, 2, \cdots, n$. Suppose the formula (23) holds when $n \leq m \leq i$. Then when $i = m + 1$,

$$\begin{aligned} \mu_{m+1} &= \mu_m + \mu_{m-1} + \cdots + \mu_{m-n+1} \\ &\geq \alpha(\tau_n^m + \tau_n^{m-1} + \cdots + \tau_n^{m-n+1}) \\ &= \alpha\tau_n^{m-n+1}(\tau_n^{n-1} + \tau_n^{n-2} + \cdots + \tau_n + 1) \\ &= \alpha\tau_n^{m-n+1}\tau_n^n = \alpha\tau_n^{m+1} \end{aligned}$$

By induction we know that the formula (23) holds. Therefore,

$$\varepsilon_{k'+i} = M^{\frac{1}{1-n}} \eta_{k'+i} \leq M^{\frac{1}{1-n}} \eta^{\alpha\tau_n^i}$$

It follows that

$$R_{\tau_n}\{x_k\} = \limsup_{i\to\infty} \varepsilon_{k'+i}^{\frac{1}{\tau_n^{k'+i}}} \leq \eta^{\frac{\alpha}{\tau_n k'}} < 1$$

Thus for any $\varepsilon > 0$, $R_{\tau_n - \epsilon}\{x_k\} = 0$.

Therefore, $O_R(\omega, x^*) \geq \tau_n$. ■

Theorem 5: Let ω be the iteration procedure formed by the inverse continued fraction interpolation algorithm 2.1 and $f(x)$ satisfy the conditions of Theorem 2. Then $O_R(\omega, x^*) \geq \tau_n$, where τ_n is the unique positive root of the equation

$$f_n(x) = x^n - x^{n-1} - \cdots - x - 1 = 0.$$

Proof.

It follows from Theorem 2 that when $\{x_0, \cdots, x_{n-1}\} \subseteq o(x^*, \delta)$, the sequence $\{x_k\}$ produced by the algorithm 2.1 converges to x^*. From the proof of Theorem 2 we know that (18) holds. Therefore, $\{x_k\}$ satisfies all conditions of Theorem 4. By Theorem 4 we conclude that $O_R(\omega, x^*) \geq \tau_n$. ■

VI. Experiments and Applications

If the convergence order η of an iterative method is given as

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\eta} = c > 0$$

then the computational order of convergence is approximated as follows

TABLE I.
COMPARISON OF THE PRESENTED METHOD AND NEWTON'S METHOD

Equations	Iterations k	Presented method	Newton's method
$f_1(x) = 0$	3	1.3×10^{-5}	4.3×10^{-1}
	4	5.1×10^{-13}	4.2×10^{-3}
	5	5.5×10^{-24}	7.5×10^{-4}
	6	1.2×10^{-49}	2.5×10^{-7}
	7	2.7×10^{-97}	2.3×10^{-13}
	8	3.3×10^{-196}	1.7×10^{-28}
	9	2.3×10^{-390}	1.3×10^{-55}
$f_2(x) = 0$	4	8.9×10^{-9}	7.5×10^{1}
	5	1.7×10^{-16}	2.6×10^{1}
	6	1.5×10^{-33}	9.7×10^{0}
	7	3.6×10^{-66}	3.3×10^{0}
	8	6.8×10^{-133}	9.3×10^{-1}
	9	8.7×10^{-625}	1.6×10^{-1}
	15	-	5.3×10^{-75}
$f_3(x) = 0$	4	8.6×10^{-7}	1.5×10^{-3}
	5	7.7×10^{-14}	4.2×10^{-4}
	6	4.8×10^{-27}	1.2×10^{-4}
	7	2.3×10^{-54}	2.5×10^{-5}
	8	5.8×10^{-108}	6.3×10^{-6}
	9	3.5×10^{-216}	1.5×10^{-6}
	90	-	2.6×10^{-56}

$$\rho = \frac{\ln |(x_{n+1} - \gamma)/(x_n - \gamma)|}{\ln |(x_n - \gamma)/(x_{n-1} - \gamma)|}$$

In this section, we consider the equations on each given interval and initial points as below for numerical performance experiment of our new algorithm.

$$\begin{array}{ll} f_1(x) = x \sin x + \cos x - 0.6, & x_0 = 0 \\ f_2(x) = 1 + (x-2)e^{-x}, & x_0 = 0 \\ f_3(x) = e^{\sin x} - x - 1, & x_0 = 1 \end{array} \tag{24}$$

All the computations are carried out using the C++ programming language. Many scientific computations demand a very high degree of numerical precision [13], [15]. For convergence, it is required that the distance of two consecutive approximations $|x_{n+1} - x_n|$ be less than ϵ. And, the absolute value of the function $|f(x_n)|$ also referred to as residual be less than ϵ.

In the numerical implementation for the examples above, a stopping criterion $|f(x_n)| < \epsilon$ for a small number $\epsilon = 10^{-500}$ within a maximum number of iterations $k_{max} = 100$ is used. $x0 = b$, the right endpoint of the given interval is taken as the initial approximation for the Newton's method. The numerical results of the values $|f(x_k)|$ for the approximations obtained by the presented method and the Newton's method are presented in Table 1. The sign "-" in the Table 1 indicates that the corresponding iteration is stopped as the approximation $|f(x_k)|$ satisfies the criterion $|f(x_k)| < \epsilon$. Our new method is superior to the Newton's method as shown in Table 1. The convergence rate of the Newton's method seems to be very slow for the equations $f_i(x) = 0, i = 2, 3$.

We have also applied our new algorithm to a project of optimal design of the noumenon structure of the largełscale electrostatic precipitator. The noumenon structure of a largełscale electrostatic precipitator is shown in Fig. 3.

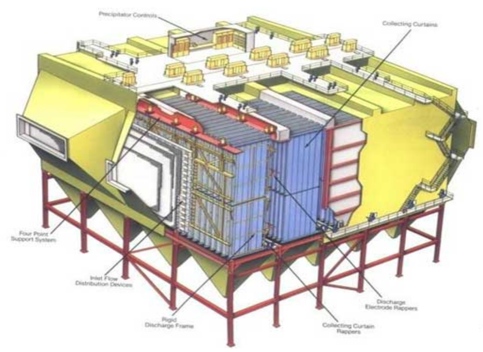

Figure 3. An electrostatic precipitator

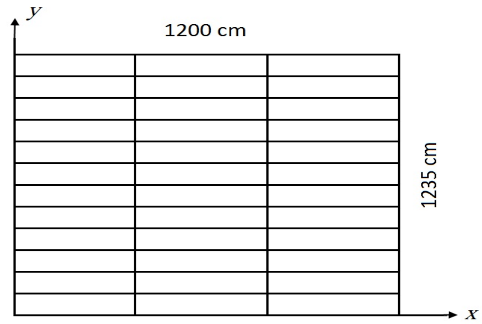

Figure 4. The structure and size of the wall

There are five parts of the structure, including the collecting electrode system, corona electrode system, the housing system, smoke box discharging systems and storage systems. The goal of our project is to optimized the noumenon structure of the electrostatic precipitator, to get a more rational structure, lighten the weight, save the material and reduce the costs.

Our new algorithm is used in the structural optimization of the wall of an electrostatic precipitator with good results. The wall of the electrostatic precipitator is made up of A_3 steel plates and steel bars. Its size is shown in Fig. 4. Its main loads are the negative pressures vertically distributed over the wall. The goal is to find a minimum weight design subjected to the stress constraints $\sigma_i \leq \overline{\sigma}$ and the deflection constraints $w_i \leq \overline{w}$. If we divide the wall along the Y axis properly into n points, then we can set a boolean variable x_i to each divided points. If $x_i = 1$ we then put a bar parallel to the X axis at the point i. Otherwise, if $x_i = 0$ then no bar at the point i. The design variables can be expressed as a binary n-vector $X = (x_1, \cdots, x_n)$, and the problem can be formulated as a nonlinear programming problem as follows.

$$\begin{array}{l} \min Z(X) = \|X\| = \sum_{i=1}^{n} x_i \\ \texttt{s.t.} \left\{ \begin{array}{lll} \sigma_i - \overline{\sigma} \leq 0 & , & i = 1, \cdots, m \\ w_i - \overline{w} \leq 0 & , & i = 1, \cdots, m \\ x_1 = x_n = 1 & , & \\ x_i \in \{0, 1\} & , & i = 1, \cdots, n \end{array} \right. \end{array} \tag{25}$$

Our presented algorithm is applied successfully as a local line search algorithm to solve the above nonlinear programming problem. The final optimal design makes an originally unfeasible design become a feasible design and reduce the total weights by 47.8%.

VII. Concluding Remarks

New algorithms and theoretical techniques for solving nonlinear equations have been developed and their applications have been expanding at an astonishing rate in all directions during the last few decades. The constantly increasing emphasis on the interdisciplinary nature of the field is the most striking trends in optimization. In all areas of applied mathematics, engineering, medicine, economics and other sciences, optimization has been a basic tool nowadays. In these areas, to develop iterative methods for solving nonlinear equations has become more and more important. There are some progress about multi step methods in the recent years. These multi step methods have been developed by combining the Newton's method with other optimization methods.

In this paper, we have proposed an efficient algorithm for solving the nonlinear equations. The presented new algorithm has a fast convergence rate. In general, the scheme of $n = 3$ with a convergence rate $\tau_3 = 1.8393$ or $n = 4$ with a convergence rate $\tau_4 = 1.9276$ converges already very fast.

The computational experiments demonstrate that the achieved results are not only of theoretical interest, but also that the techniques developed may actually lead to considerably faster algorithms.

Acknowledgement

The authors acknowledge the financial support of Natural Science Foundation of Fujian under Grant No.2013J0101, and the Haixi Project of Fujian under Grant No.A099. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] S. Amat, C. Bermudez, S. Busquier, S. Plaza, On two families of high order Newton type methods, *Appl. Math. Lett.*, vol. 25, no.12, pp. 2209-2217, 2012.

[2] S. Amat, S. Busquier, A. Grau, M. Grau-Sanchez, Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications, *Appl. Math. Comput.*, vol. 219, no.15, pp. 7954-7963 , 2013.

[3] Michael Bartholomew-Biggs, *Nonlinear Optimization with Financial Applications*, Kluwer Academic Publishers, New York, 2005.

[4] J. Dzunic, M.S. Petkovic, L.D. Petkovic, Three-point methods with and without memory for solving nonlinear equations, *Appl. Math. Comput.*, vol. 218, no.9, pp. 4917-4927, 2012.

[5] J.A. Ezquerro, M. Grau-Sanchez, A. Grau, M.A. Hernandez, M. Noguera, N. Romero, On iterative methods with accelerated convergence for solving systems of nonlinear equations, *J. Optim. Theory Appl.*, vol. 151, no.1, pp. 163-174, 2011.

[6] M.A. Hernandez, N. Romero, On a characterization of some Newton-like methods of R-order at least three, *J. Comput. Appl. Math.*, vol. 183, no.1, pp. 53-66, 2005.

[7] J.A. Ezquerro, M.A. Hernandez, An improvement of the region of accessibility of Chebyshev's method from Newton's method, *Math. Comput.*, vol. 78, no.1, pp. 1613-1627, 2009.

[8] Fang L., A cubically convergent iterative method for solving nonlinear equations, *Adv. Appl. Math. Sci.*, vol. 10, no. 2, pp. 117-119, 2011.

[9] M. Grau-Sanchez, A. Grau, M. Noguera, Frozen divided difference scheme for solving systems of nonlinear equations, *J. Comput. Appl. Math.*, vol. 235, no. 6, pp. 1739-1743 2011.

[10] M. Grau-Sanchez, A. Grau, M. Noguera, On the computational efficiency index and some iterative methods for solving systems of nonlinear equations,*J. Comput. Appl. Math.*, vol. 236, no. 6, pp. 1259-1266, 2011.

[11] Y.I. Kim, Y.H. Geum, A cubic-order variant of Newton's method for finding multiple roots of nonlinear equations, *Comput. Math. Appl.*, vol. 62, no. 4, pp. 1634-1640, 2011.

[12] M.S. Petkovic, J. Dzunic, B. Neta, Interpolatory multipoint methods with memory for solving nonlinear equations, *Appl. Math. Comput.*, vol. 218, no. 6, pp. 2533-2541, 2011.

[13] L.B. Rall, *Computational Solution of Nonlinear Operator Equations*, Robert Krieger Publishing Company, Inc., California, 1979.

[14] F. Sha, X. Tan, A class of iterative methods of third order for solving nonlinear equations, *Int J. Nonlinear Sci.*, vol. 11, no. 2, pp. 165-167, 2011.

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

[16] X. Wang, L. Wang, A Fast Algorithm for Solving Nonlinear Equations, *International Conference on System Modeling and Optimization (ICSMO 2012)*, pp. 6-9, 2012.

[17] T. Yamashita, H. Yabe, T. Tanabe, A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, *Math. Program. Ser. A*, vol. 102, no. 1, pp. 111-151, 2005.

[18] B. I. Yun, Solving nonlinear equations by a new derivative free iterative method, *Appl. Math. Comput.*, vol. 217, no. 1, pp. 5768-5773, 2011.