A family of composite fourth-order iterative methods for solving nonlinear equations

Changbum Chun

School of Liberal Arts, Korea University of Technology and Education, Cheonan City, Chungnam 330-708, Republic of Korea

Abstract

In this paper, we present a family of new fourth-order iterative methods for solving nonlinear equations. Per iteration the methods consisting of the family require only two evaluations of the function and one evaluation of its derivative. Several numerical examples are given to illustrate the efficiency and performance of some of the presented methods. © 2006 Elsevier Inc. All rights reserved.

Keywords: Newton’s method; Iterative methods; Nonlinear equations; Order of convergence

1. Introduction

In this paper, we consider iterative methods to find a simple root \( a \), i.e., \( f(a) = 0 \) and \( f'(a) \neq 0 \), of a nonlinear equation \( f(x) = 0 \).

Newton’s method is the well-known iterative method for finding \( a \) by using

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},
\]

that converges quadratically in some neighborhood of \( a \).

There exist many iterative methods improving Newton’s method for solving nonlinear equations. However, many of those iterative methods depend on the second or higher derivatives in computing process which make their practical application restricted strictly. As a result, Newton’s method is frequently and alternatively used to solve nonlinear equations because of higher computational efficiency.

In recent years, there has been some progress on iterative methods improving Newton’s method with cubic convergence that do not require the computation of second derivatives for solving nonlinear equations, see [1–9] and the reference therein. Furthermore, in [10–12] several iterative fourth-order methods which are free from second derivatives and that do require only three evaluations of both the function and its derivatives are proposed.

E-mail address: cbchun@kut.ac.kr

0096-3003/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2006.09.009

Motivated and inspired by the research going on in this area, in this paper, we suggest and analyze a family of new fourth-order iterative methods. The new methods do not require the computation of second or higher derivatives; besides, they require only two evaluations of the function and one of its derivative. We show that the iterative methods so obtained have all order of convergence four, so that they have better efficiency than Newton's method. Several numerical examples are given to show the efficiency and the performance of the methods presented in this contribution.

2. Families of iterative methods

In the sequel, whenever we mention that an iteration function $\phi$ is of order $p$, it means that the corresponding iterative method defined by $x_{n+1} = \phi(x_n)$ is of convergence order $p$, that is, the error $|x - x_{n+1}|$ is proportional to $|x - x_n|^p$ as $n \rightarrow \infty$. We refer to [13] for further details about the order of an iteration function.

Let $\phi$, $\zeta$ and $\eta$ be iteration functions of order three. Now, we consider the function $\Phi$ defined in the following linear combination form

$$\Phi(x) = x - \theta_1[x - \phi(x)] - \theta_2[x - \zeta(x)] - \theta_3[x - \eta(x)],$$

(2)

where $\theta_i \in \mathbb{R}$, $i = 1, 2, 3$, $\theta_1 + \theta_2 + \theta_3 = 1$. Obviously, when $\theta_1 = 1$, $\theta_2 = \theta_3 = 0$ and $\theta_2 = 1$, $\theta_1 = \theta_3 = 0$, (2) reduces to $\phi(x)$ and $\zeta(x)$, respectively, while it becomes $\eta(x)$ if $\theta_3 = 1$, $\theta_1 = \theta_2 = 0$.

We now give a convergence analysis of the iterative method defined by (2). We let $e_n = x_n - x$ and $c_k = (1/k!)f^{(k)}(x)/f'(x)$.

**Theorem 2.1.** Let $x \in I$ be a simple zero of a sufficiently differentiable function $f: I \rightarrow \mathbb{R}$ for an open interval $I$. Let $\theta_i, i = 1, 2, 3$ be nonzero real numbers with $\theta_1 + \theta_2 + \theta_3 = 1$, and $\phi$, $\zeta$ and $\eta$ be iteration functions of order three. Then the iteration function defined by (2) is of order at least three, and the iterative method defined by $x_{n+1} = \Phi(x_n)$ then satisfies the error equation

$$e_{n+1} = \frac{1}{6} [\theta_1 \phi^{(3)}(x) + \theta_2 \zeta^{(3)}(x) + \theta_3 \eta^{(3)}(x)]e_n^3 + O(e_n^4).$$

(3)

Furthermore, the iteration function defined by (2) is of order at least four for each triple $(\theta_1, \theta_2, \theta_3)$ making the coefficient of $e_n^3$ in (3) zero, and the iterative method defined by $x_{n+1} = \Phi(x_n)$ then satisfies the error equation

$$e_{n+1} = \frac{1}{24} [\theta_1 \phi^{(4)}(x) + \theta_2 \zeta^{(4)}(x) + \theta_3 \eta^{(4)}(x)]e_n^4 + O(e_n^5).$$

(4)

**Proof.** Since $\phi$, $\zeta$ and $\eta$ are iteration functions of order three, we have $\phi(x) = \zeta(x) = \eta(x) = x$, $\phi'(x) = \phi''(x) = \zeta'(x) = \zeta''(x) = \eta'(x) = \eta''(x) = 0$. Using Taylor expansion, we obtain

$$\phi(x) = x + \phi_3(x - x)^3 + \phi_4(x - x)^4 + O((x - x)^5),$$

(5)

$$\zeta(x) = x + \zeta_3(x - x)^3 + \zeta_4(x - x)^4 + O((x - x)^5),$$

(6)

$$\eta(x) = x + \eta_3(x - x)^3 + \eta_4(x - x)^4 + O((x - x)^5),$$

(7)

where $\phi_k = \frac{1}{3!}\phi^{(3)}(x)$, $\zeta_k = \frac{1}{3!}\zeta^{(3)}(x)$, and $\eta_k = \frac{1}{3!}\eta^{(3)}(x)$, $k = 3, 4$.

Since $\theta_1 + \theta_2 + \theta_3 = 1$, substituting (5)–(7) into (2) yields

$$\Phi(x) = x + [\theta_1 \phi_3 + \theta_2 \zeta_3 + \theta_3 \eta_3](x - x)^3 + [\theta_1 \phi_4 + \theta_2 \zeta_4 + \theta_3 \eta_4](x - x)^4 + O((x - x)^5).$$

(8)

This shows that the iteration function defined by (2) is of order at least three, and the error equation defined by (3) follows.

Equating the coefficient of $(x - x)^3$ in Eq. (8) to zero and solving the resulting equation for $\theta_1$, $\theta_2$ and $\theta_2$ under the condition $\theta_1 + \theta_2 + \theta_3 = 1$, we can find its solutions $(\theta_1, \theta_2, \theta_3)$. With each triple $(\theta_1, \theta_2, \theta_3)$, the associated iteration function defined by (2) is of order at least four, and the iterative method defined by $x_{n+1} = \Phi(x_n)$ then satisfies the error equation

\[ e_{n+1} = \frac{1}{24} \left[ \theta_1 \phi^{(4)}(x) + \theta_2 \psi^{(4)}(x) + \theta_3 \eta^{(4)}(x) \right] e_n^4 + O(e_n^5). \]  

This completes the proof. \( \square \)

To construct the fourth-order iterative method via Theorem 2.1, we consider only the following third-order iteration functions \( \phi, \zeta \) and \( \eta \) defined by

\[ \phi(x) = x - \frac{f(x) + f(y)}{f'(x)}, \]  

which is Potra–Pták’s iteration function [1],

\[ \zeta(x) = x - \frac{f^2(x)}{f'(x)[f(x) - f(y)]}, \]  

where \( y = x - f(x)/f'(x) \), which is Newton–Steffensen iteration function [5], and

\[ \eta(x) = x - \left[ \frac{f(x)}{f'(x)} + \frac{f(x)f(y)}{f^2(x) + f^2(y)} \right], \]  

where \( y = x - f(x)/f'(x) \), which is the iteration function obtained in [14], respectively.

By the help of Maple, we have

\[ \phi^{(3)}(x) = 12c_2^3, \quad \phi^{(4)}(x) = -216c_2^3 + 168c_2c_3, \]  

\[ \zeta^{(3)}(x) = 6c_2^2, \quad \zeta^{(4)}(x) = 72(c_2c_3 - c_2^3), \]  

\[ \eta^{(3)}(x) = 12c_2^2, \quad \eta^{(4)}(x) = 24c_2 - 216c_2^3 + 168c_2c_3, \]  

By Theorem 2.1, we need to solve the system of equations

\[ \begin{cases} \theta_1 + \theta_2 + \theta_3 = 1, \\ \phi^{(3)}(x)\theta_1 + \zeta^{(3)}(x)\theta_2 + \eta^{(3)}(x)\theta_3 = 0, \end{cases} \]  

for \( \theta_1, \theta_2 \) and \( \theta_3 \) to construct fourth-order iteration functions via (2).

In the case that \( \phi = \phi_1, \zeta = \phi_2 \) and \( \eta = \phi_3 \) in (2), the system of equations

\[ \begin{cases} \theta_1 + \theta_2 + \theta_3 = 1, \\ 12c_2^3\theta_1 + 6c_2^2\theta_2 + 12c_2\theta_3 = 0, \end{cases} \]  

to obtain

\[ \begin{align*} 
\theta_1 &= -1 - \beta, \\
\theta_2 &= 2, \\
\theta_3 &= \beta, 
\end{align*} \]  

where \( \beta \in \mathbb{R} \). In this case, the iteration function defined by (2) gives us a family of infinitely many new fourth-order iterative methods

\[ x_{n+1} = x_n + (1 + \beta) \frac{f(x_n) + f(y_n)}{f'(x_n)} - 2 \frac{f^2(x_n)}{f'(x_n)[f(x_n) - f(y_n)]} - \beta \left[ \frac{f(x_n)}{f'(x_n)} + \frac{f(x_n)f(y_n)}{f^2(x_n) + f^2(y_n)} \right], \]  

where \( y_n = x_n - f(x_n)/f'(x_n) \).

The fourth-order convergence of these new methods can be also ascertained by making use of Maple. By the help of Maple, we have the error equation for (18)

\[ e_{n+1} = (\beta c_2 + 3c_2^3 - c_2c_3)e_n^4 + O(e_n^5), \]

which proves the fourth-order convergence for any real number \( \beta \), agreeing with the theory as asserted in Theorem 2.1.

Formula (18) includes, as particular cases, the following ones:

For \( \beta = 0 \), we obtain a fourth-order method:
\[ x_{n+1} = x_n - \frac{f^2(x_n) + f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)]}, \]  

(20)

which was obtained by Jisheng et al. [12].

For \( \beta = -1 \), we obtain a new fourth-order method:

\[ x_{n+1} = x_n + \frac{f(x_n) + f(y_n)}{f'(x_n)} + 2 \left( \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)}{f'(x_n)[f(x_n) - f(y_n)]} \right), \]

(21)

For \( \beta = -2 \), we obtain another new fourth-order method:

\[ x_{n+1} = x_n - \frac{f'(x_n) + f'(y_n)}{f''(x_n)} + 2 \left( \frac{f(x_n)}{f'(x_n)} - \frac{f^2(x_n)}{f'(x_n)[f(x_n) - f(y_n)]} \right), \]

(22)

which can be considered as an improved Potra–Pták’s method.

We can continuously apply formula (18) to derive as many new quartically convergent iterative methods, as the methods (20)–(22) above. It should be noted that the methods defined by (18) are of fourth-order even though they require but two evaluations of the function and one of its derivative. If we consider the definition of efficiency index [15] as 

\[ p^n, \]

where \( p \) is the order of the method and \( m \) is the number of functional evaluations per iteration required by the method, we have that all of the methods obtained from formula (18) have the efficiency index equal to \( 4^\frac{1}{3} \approx 1.587 \), which is better than the one of Newton’s method \( \sqrt{2} \approx 1.414 \).

3. Numerical examples and conclusions

All computations were done using MAPLE using 64 digit floating point arithmetics (Digits:=64). We accept an approximate solution rather than the exact root, depending on the precision (\( \epsilon \)) of the computer. We use the following stopping criteria for computer programs: (i) \( |x_{n+1} - x_n| < \epsilon \), (ii) \( |f(x_{n+1})| < \epsilon \), and so, when the stopping criterion is satisfied, \( x_{n+1} \) is taken as the exact root \( x \) computed. For numerical illustrations in this section we used the fixed stopping criterion \( \epsilon = 10^{-15} \).

We present some numerical test results for various quartically convergent iterative schemes in Table 1. Compared were the Newton method (NM), the Jarratt method [10] (JM) defined by

\[ x_{n+1} = x_n - \left( 1 - \frac{3}{2} \frac{f''(y_n) - f''(x_n)}{f''(y_n) - f''(x_n)} \right) \frac{f(x_n)}{f'(x_n)}, \]

Table 1

<table>
<thead>
<tr>
<th>( f(x) )</th>
<th>IT</th>
<th>NFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x_0) = -0.3 )</td>
<td>55</td>
<td>NM 110</td>
</tr>
<tr>
<td>( f_1(x_0) = 1 )</td>
<td>6</td>
<td>JM 12</td>
</tr>
<tr>
<td>( f_2(x_0) = 0 )</td>
<td>5</td>
<td>KM 10</td>
</tr>
<tr>
<td>( f_2(x_0) = 1 )</td>
<td>5</td>
<td>CM1 10</td>
</tr>
<tr>
<td>( f_3(x_0) = -1 )</td>
<td>6</td>
<td>CM2 10</td>
</tr>
<tr>
<td>( f_3(x_0) = -2 )</td>
<td>5</td>
<td>NM 10</td>
</tr>
<tr>
<td>( f_4(x_0) = 0 )</td>
<td>6</td>
<td>JM 12</td>
</tr>
<tr>
<td>( f_4(x_0) = 1 )</td>
<td>5</td>
<td>KM 10</td>
</tr>
<tr>
<td>( f_5(x_0) = 3 )</td>
<td>7</td>
<td>CM1 12</td>
</tr>
<tr>
<td>( f_5(x_0) = 4 )</td>
<td>8</td>
<td>CM2 12</td>
</tr>
<tr>
<td>( f_6(x_0) = 2 )</td>
<td>9</td>
<td>NM 12</td>
</tr>
<tr>
<td>( f_6(x_0) = 3.5 )</td>
<td>11</td>
<td>JM 13</td>
</tr>
<tr>
<td>( f_7(x_0) = 1 )</td>
<td>7</td>
<td>KM 12</td>
</tr>
<tr>
<td>( f_7(x_0) = 2 )</td>
<td>6</td>
<td>CM1 12</td>
</tr>
</tbody>
</table>

where \( yn = x_n - \frac{f(x_n)}{f'(x_n)} \), the method of Kou et al. (20) (KM), and the methods (21) (CM1) and (22) (CM2) introduced in the present contribution. We used the same test functions as Kou et al. [12] and display the approximate zero \( x_n \) found up to the 28th decimal places.

\[
\begin{align*}
f_1(x) &= x^3 + 4x^2 - 10, \quad x_n = 1.3652300134140968457608068290, \\
f_2(x) &= x^2 - e^x - 3x + 2, \quad x_n = 0.25753028543986076045536730494, \\
f_3(x) &= xe^{-x} - \sin^2x + 3 \cos x + 5, \quad x_n = -1.2076478271309189270094167584, \\
f_4(x) &= \frac{\sin(x)e^x + \ln(x^2 + 1)}{x}, \quad x_n = 0, \\
f_5(x) &= (x - 1)^3 - 2, \quad x_n = 2.2599210498948731647672106073, \\
f_6(x) &= (x + 2)e^x - 1, \quad x_n = 0.4428540100238858314132800000, \\
f_7(x) &= \sin^2(x) - x^2 + 1, \quad x_n = 1.404491648215341226035086178.
\end{align*}
\]

As convergence criterion, it was required that the distance of two consecutive approximations for the zero was less than \( 10^{-15} \). Displayed in Table 1 are the number of iterations to approximate the zero (IT) and the number of functional evaluations (NFE) counted as the sum of the number of evaluations of the function itself plus the number of evaluations of the derivative.

The computational results presented in Table 1 shows that in almost all of cases, the presented methods converge more rapidly than Newton’s method and require the less number of functional evaluations. This means that the new methods have better efficiency in computing process than Newton’s method as the compared other methods, and furthermore, the formula (18) produces the fourth-order methods that can compete with Newton’s method. For most of the functions we tested, the obtained methods behave at least equal performance compared to the other known methods of the same order.

Among the methods compared for numerical tests the Jarrat method seems in almost all of cases to require the least number of functional evaluations. However, it should be noted that per iteration the methods (KM), (CM1) and (CM2) do require two evaluations of the function and one of its first derivative, whereas the Jarratt method does require one evaluations of the function and two of its first derivative, costing more expensive computation.

4. Conclusion

In this work we presented an approach which can be used to constructing family of fourth-order iterative methods that do not require the computation of second or higher derivatives; besides the resulting methods do require only two evaluations of the function and one of its first derivative. Some of the obtained methods were also compared in their performance and efficiency to various other iteration methods of the same order, and it was observed that they demonstrate at least equal behavior. In this paper, we have constructed one such family only, and more families could be constructed using the presented theory, which will be exploited in detail in a forthcoming paper.

References


