An efficient two-step iterative method with fifth-order convergence for solving non-linear equations

K. Trachoo, D. Prathumwan* and I. Chaiya*

Abstract. In this paper, we propose a new modification two-step iterative methods for finding simple roots of nonlinear equation in a single variable. The new method is based on Chebychev-Halley method and Newton's method of third-order. Analysis of convergence shows that the new method have fifth-order convergence. The experimental results and comparison confirm that the new method is efficient.

AMS Subject Classification (2020): 41A25, 65D99

Keywords: Root finding, iterative method, Chebychev-Halley method, Newton’s method, nonlinear equation

1. Introduction

In numerical analysis, developing a technique to solve non-linear equations \( f(x) = 0 \) is essential since many non-linear equations arise in many fields such as science, technology and engineering [4]. Many non-linear equations cannot be solved analytically. Therefore, the numerical approach, which find the approximate solution, is essential and wildly use in many real-world problems.

There are many iterative methods which can be used to find a root of \( f(x) = 0 \), such as Newton’s method, and Chebyshev method. The
Newton’s method is given by

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x) \neq 0, \quad n = 0, 1, 2, \ldots \] \tag{1} 

This method converges quadratically. Kou et al. [3] proposed the modification of Newton method with cubically convergence which is of the form

\[ x_{n+1} = x_n - \frac{f(x_n)}{2} \left( \frac{1}{f'(x_n)} + \frac{1}{f'(x_n - f(x_n)/f'(x_n))} \right). \] \tag{2} 

Basto et al. [1] presented a new method which also converges cubically,

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

A number of cubically convergent iterative methods have been presented. One of them is a family of third-order method, namely Chebyshev-Halley method [2] which is given by

\[ x_{n+1} = x_n - \left( 1 + \frac{1}{2} \frac{L_f(x_n)}{1 - \alpha L_f(x_n) f'(x_n)} \right) f(x_n), \] \tag{4} 

where

\[ L_f(x_n) = \frac{f''(x_n)f(x_n)}{[f'(x_n)]^2}. \]

Weerakoon and Fernando [5] proposed a variant modification of Newton’s method with third-order convergence which can be written as

\[ x_{n+1} = x_n - \frac{2f(x_n)}{f''(x_n) + f'(x_n - f(x_n)/f'(x_n))}. \] \tag{5} 

In this work, we improve the Chebyshev-Halley method (4) by using the concept of modification of Newton method with third-order (5). Then, we obtain the two-step iteration method.

The proposed method had fifth-order of convergence. It is suggested that the obtained method is better than the previous methods.
2. The Method

We derive the modification method based on the concept of a variant Newton’s method of third-order (5) and the Chebyshev-Halley method (4).

Assume that \( x^* \) is a simple root of \( f(x) = 0 \) that is \( f(x^*) = 0 \) where \( f \) is a real value function.

Let \( y_n \) be defined by (4) as

\[
y_n = x_n - \left( 1 + \frac{1}{2} \left( \frac{L_f(x_n)}{1 - \alpha L_f(x_n)} \right) \frac{f(x_n)}{f'(x_n)} \right).
\]  

(6)

Then the Taylor’s expansion of \( f'(y_n) \) around \( x_n \) can be written as

\[
f'(y_n) = f'(x_n) + f''(x_n)(y_n - x_n) + \frac{1}{2}f'''(x_n)(y_n - x_n)^2 + \cdots,
\]  

(7)

or we can write (7) as

\[
f'(y_n) \approx f'(x_n) + f''(x_n)(y_n - x_n).
\]  

(8)

Substitute (8) into equation (5), we obtain a new iteration method

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

(9)

where \( y_n \) is defined by (6) and

\[
L_f(x_n) = \frac{f''(x_n)f(x_n)}{[f'(x_n)]^2}.
\]

So (6) and (9) are combined. The modified Newton-Chebyshev-Halley (mNCH) method is the name of the modification method.

3. Analysis of convergence

In this section, we consider the analysis of convergence of the proposed algorithm.
Definition 3.1 [5]. Let $e_n = x_n - x^*$ be the error in the $n^{th}$ iteration, the relation

$$e_{n+1} = c e_n^p + O(e_n^{p+1})$$

is called the error equation. The value of $p$ thus obtained is called the order of convergence.

Theorem 3.2 Assume that the function $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval $D$ has a simple root $x^* \in D$ and $\alpha \in [0,1]$. If $f(x)$ is sufficiently smooth in the neighborhood of the root $x^*$, then the order of convergence of the modified Newton-Chebyshev-Halley methods is five.

Proof. Let $x^*$ be a simple root of

$$f(x) = 0,$$
$$e_n = x_n - x^*,$$
$$d_n = y_n - x^*,$$
$$f'(x^*) \neq 0.$$

Using Taylor’s expansion of $f(x_n)$ about $x^*$, we have

$$f(x_n) = f(x^*) + f'(x^*)(x_n - x^*) + \frac{1}{2!}f''(x^*)(x_n - x^*)^2 + \ldots$$
$$= f(x^*) + f'(x^*)[e_n + e_n^2 c_2 + e_n^3 c_3 + \ldots]$$

(10)

where $c_k = \frac{1}{k!} \frac{f^{(k)}(x^*)}{f'(x^*)}$, for $k = 2, 3, \ldots$.

Since $x^*$ is a simple root of $f(x) = 0$ then $f(x^*) = 0$. So (10) becomes

$$f(x_n) = f'(x^*)[e_n + e_n^2 c_2 + e_n^3 c_3 + \ldots]$$

(11)

Then, the Taylor’s expansion of $f'(x_n)$ and $f''(x_n)$ about $x^*$ can be written
An efficient two-step iterative method

As

\[ f'(x_n) = f'(x^*) + f''(x^*)(x_n - x^*) + \frac{1}{2!} f'''(x^*)(x_n - x^*)^2 + \ldots \]

\[ = f'(x^*)[1 + 2e_n c_2 + 3e_n^2 c_3 + \ldots] \]  
\[ (12) \]

Dividing (11) by (12), we obtain

\[ \frac{f(x_n)}{f'(x_n)} = \frac{f'(x^*)[e_n + e_n^2 c_2 + e_n^3 c_3 + \ldots]}{f'(x^*)[1 + 2e_n c_2 + 3e_n^2 c_3 + \ldots]} \]

\[ = e_n - c_2 e_n^2 + 2c_2 e_n^3 - (9c_2 c_3 - 4c_2^3) e_n^4 + \ldots \]  
\[ (13) \]

By dividing (13) by (12), we get

\[ \frac{f''(x_n)}{f'(x_n)} = 2c_2 - 2(2c_2^2 - 3c_3)e_n - (9c_2^2 - 4c_2^3)e_n^2 \]

\[ - 6(3c_3^2 + 4c_2^2 c_3)e_n^3 + \ldots \]  
\[ (15) \]

After that, we substitute (14) and (15) into

\[ L_f(x_n) = \frac{f''(x_n) f(x_n)}{(f'(x_n))^2} \]

We obtain

\[ L_f(x_n) = \frac{f''(x_n) f(x_n)}{(f'(x_n))^2} = 2c_2 e_n + 6(c_3 - c_2^2)e_n^2 \]

\[ + 8(2c_2^3 - 2c_2 c_3)e_n^3 + \ldots \]  
\[ (16) \]

From (6), (14), (16), \( d_n = y_n - x^* \), and \( x^* = x_n - e_n \), we get

\[ d_n = x_n - \left( 1 + \frac{L_f(x_n)}{2(1 - \alpha L_f(x_n))} \right) \frac{f(x_n)}{f'(x_n)} - (x_n - e_n) \]

\[ = (4c_2^2 - 3c_3 - 2\alpha c_2^2)e_n^3 + 2(7\alpha c_2^3 - 6\alpha c_2 c_3 - 12\alpha^2 c_2^4)e_n^4 + \ldots \]  
\[ (17) \]

Since \( d_n = y_n - x^* \), we have that \( z_n - x_n = d_n - e_n \).

The Taylor’s expansion of \( f(y_n) \) about \( x^* \) can be expressed as

\[ f(y_n) = f(x^*) + f'(x^*)(y_n - x^*) + \frac{1}{2!} f''(x^*)(y_n - x^*)^2 + \ldots \]

\[ = f(x^*) + f'(x^*)[d_n + d_n^2 e_2 + \ldots] \],  
\[ (18) \]
where \( c_k = \frac{1}{k!} \frac{f^{(k)}(x^*)}{f'(x^*)} \), for \( k = 2, 3, \ldots \).

Since \( f(x^*) = 0 \), (18) becomes

\[
f(y_n) = f'(x^*)[d_n + d_n^2c_2 + \ldots].
\]

(19)

Using the Taylor’s expansion of \( f'(y_n) \) about \( x^* \), we have

\[
f'(y_n) = f'(x^*)[1 + 2d_n^2c_2 + \ldots].
\]

(20)

By (13), (17) and \( z_n - x_n = d_n - e_n \), we can write \( f''(x_n)(y_n - x_n) \) as

\[
f''(x_n)(y_n - x_n) = f'(x^*)(-2c_2e_n - 6c_3e_n^2 + 2(4c_2^3 - 3c_2c_3 - 2\alpha c_2^2) e_n^3 + \ldots).
\]

(21)

From (12) and (21), we obtain

\[
f'(x_n) + f''(x_n)(y_n - x_n) = f'(x^*)[1 - 3c_3e_n^2 + 2(4c_2^3 - 3c_2c_3 - 2\alpha c_2^2) e_n^3 + \ldots].
\]

(22)

From (17), (18) and (22), we have

\[
y_n - f'(x_n) + f''(x_n)(y_n - x_n)
\]

\[
= y_n - f'(x^*)[d_n + e_n^2c_2 + \ldots]
\]

\[
= x^* - 3(4c_2^3c_3 - 3c_2^2c_3 - 2\alpha c_2^2c_3)e_n^3 + \ldots.
\]

(23)

Using the Taylor’s expansion of

\[
f'(z_n - f'(x_n) + f''(x_n)(z_n - x_n))
\]

around \( x^* \) and simplifying, we get

\[
f'(z_n - f'(x_n) + f''(x_n)(z_n - x_n))
\]

\[
= f'(x^*)\left(1 - 6(4c_2^3c_3 - 3c_2^2c_3 - 2\alpha c_2^2c_3)e_n^3 + \ldots\right) + \ldots.
\]

(24)
Combining (22) and (24), we have
\[
\begin{align*}
&f'(z_n - f(z_n) + f'(x_n) + f''(x_n)(z_n - x_n) \\
&= f'(x^*) \left[ 2 - 3c_3e_n^2 + 2(4c_2^3 - 3c_2c_3 - 2c_3^3)e_n^3 \\
&\quad - 6(4c_2^3 - 3c_2c_3^2 - 2c_2c_3\alpha)e_n^5 + \ldots \right].
\end{align*}
\] (25)

From (17), (20) and (25), we obtain
\[
e_{n+1} = - \left( 6c_2^2c_3 - 3c_2^2c_3\alpha - \frac{9c_2^2}{2} \right) e_n^5 + O(e_n^6).
\]

So, we can conclude that the modified Newton-Chebyshev-Halley’s method (6) and (9) is the fifth convergent.

4. Numerical Results

In this section, the results of numerical calculation on various different functions, initial point to demonstrate the efficiency of proposed methods are presented. We compare these methods including:

- the classical Newton’s method (NM), Chebyshev method (CM), Halley’s method (HM), Chebyshev-Halley’s method (CHM) and modified Newton-Chebyshev-Halley’s method (mNCH).

All computation are carried out by using Matlab. Moreover, we use the stopping criteria \(|x_{n+1} - x_n| \leq \epsilon\) and \(|f(x)| \leq \epsilon\) where \(\epsilon = 10^{-14}\) for computer program. The various test functions which their approximate zero \(x^*\) and initial point \(x_0\) are as shown below.

Table 1 presents the various iterative methods and the number of iteration in order to find the approximate solution \(x^*\). Also, \(NC\) in this table means that the method does not converge to the root \(x^*\).
\[ f_1(x) = x^3 + 4x^2 - 10, \quad x^* = 1.3652300134140969, \quad x_0 = -0.3, 1, 2 \]
\[ f_2(x) = x^3 + 4x^2 - 15, \quad x^* = 1.63198080, \quad x_0 = -0.3, 1, 2 \]
\[ f_3(x) = x^3 - 10, \quad x^* = 2.1544346906318837, \quad x_0 = 1.7, 2.7 \]
\[ f_4(x) = (x - 1)^3 - 2, \quad x^* = 2.2599210498948, \quad x_0 = 1.85 \]
\[ f_5(x) = x^5 + x - 1000, \quad x^* = 6.30877712997, \quad x_0 = 4 \]
\[ f_6(x) = \sqrt{x} + \frac{1}{x} - 3, \quad x^* = 9.6335955628, \quad x_0 = 9 \]
\[ f_7(x) = x^3 - 2x^2 + x - 1, \quad x^* = 1.754910578, \quad x_0 = 1, 1.02 \]
\[ f_8(x) = x^2 - (2 - x)^3, \quad x^* = 1, \quad x_0 = 1.1 \]
\[ f_9(x) = x^5 + x^4 + 4x^2 - 15, \quad x^* = 1.34742809896304, \quad x_0 = 1.2 \]
\[ f_{10}(x) = x^3 + 1, \quad x^* = -1, \quad x_0 = -1.5 \]
\[ f_{11}(x) = 11x^{11} - 1, \quad x^* = 0.840133097503664, \quad x_0 = 1 \]
\[ f_{12}(x) = x^{10} - 1, \quad x^* = 1, \quad x_0 = 2 \]
\[ f_{13}(x) = x^3 - 3x^2 + x - 2, \quad x^* = 2.893289, \quad x_0 = 2.5 \]
\[ f_{14}(x) = (x - 2)^3 - 1, \quad x^* = 3, \quad x_0 = 3.5, 4.5 \]

Table 1: Comparison between various methods and the number of iterations

<table>
<thead>
<tr>
<th>Functions</th>
<th>( x_0 )</th>
<th>( f_1(x) )</th>
<th>( f_2(x) )</th>
<th>( f_3(x) )</th>
<th>( f_4(x) )</th>
<th>( f_5(x) )</th>
<th>( f_6(x) )</th>
<th>( f_7(x) )</th>
<th>( f_8(x) )</th>
<th>( f_9(x) )</th>
<th>( f_{10}(x) )</th>
<th>( f_{11}(x) )</th>
<th>( f_{12}(x) )</th>
<th>( f_{13}(x) )</th>
<th>( f_{14}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
<td>( x_0 )</td>
</tr>
<tr>
<td>( f_1(x) )</td>
<td>55</td>
<td>31</td>
<td>6</td>
<td>31</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>29</td>
<td>9</td>
<td>29</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>28</td>
<td>9</td>
<td>28</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_2(x) )</td>
<td>28</td>
<td>12</td>
<td>6</td>
<td>27</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3(x) )</td>
<td>7</td>
<td>12</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_4(x) )</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_5(x) )</td>
<td>( NC )</td>
<td>( NC )</td>
<td>5</td>
<td>( NC )</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_6(x) )</td>
<td>256</td>
<td>259</td>
<td>64</td>
<td>6</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_7(x) )</td>
<td>16</td>
<td>4</td>
<td>( NC )</td>
<td>3</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.02</td>
<td>3</td>
<td>55</td>
<td>( NC )</td>
<td>55</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_8(x) )</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_9(x) )</td>
<td>7</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{10}(x) )</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{11}(x) )</td>
<td>11</td>
<td>23</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{12}(x) )</td>
<td>2</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table Contd...
An efficient two-step iterative method

<table>
<thead>
<tr>
<th>Functions</th>
<th>$x_0$</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{13}(x)$</td>
<td>2.5</td>
<td>7 6 6 6 10</td>
</tr>
<tr>
<td>$f_{14}(x)$</td>
<td>3.5 4.5</td>
<td>16 11 4 11 5 28 19 4 19 5</td>
</tr>
</tbody>
</table>

5. Conclusion

We obtained a modified Newton-Chebyshev-Halley’s method for solving non-linear equations. We proved that the proposed method has the fifth-order of convergence by Theorem 3.2. This means that the modification method is more efficient than the well-known Newton’s method. From the numerical results, we confirm that the proposed method has good practical utility.

Acknowledgements. The authors would like to thank the anonymous referee for his/her comments that helped us improve this article. In addition, the authors also thank Maturada Chiewkarn for her help.

References


Department of Mathematics  
Faculty of Science  
Mahasarakham University  
Mahasarakham 44150  
Thailand  
E-mail: kamonchat.t@msu.ac.th

Department of Mathematics  
Faculty of Science  
Khon Kaen University  
Khon Kaen 40002  
Thailand  
E-mail: dinpr@kku.ac.th

Department of Mathematics  
Faculty of Science  
Mahasarakham University  
Mahasarakham 44150  
Thailand  
E-mail: inthira.c@msu.ac.th

(Received: August, 2021; Revised: September, 2021)