A combination method for solving nonlinear equations

B.P. Silalahi¹, R. Laila² and I.S. Sitanggang³

¹,²Department of Mathematics, Bogor Agriculture University, Bogor, INDONESIA
³Department of Computer Science, Bogor Agriculture University, Bogor, INDONESIA

Emails: bibparuhum@gmail.com, lailarahmah.ella@gmail.com

Abstract. This paper discusses methods for finding solutions of nonlinear equations: the Newton method, the Halley method and the combination of the Newton method, the Newton inverse method and the Halley method. Computational results in terms of the accuracy, the number of iterations and the running time for solving some given problems are presented.

1. Introduction

In this paper we will discuss methods for finding solutions of nonlinear equations. The Newton method is one of the best methods to determine the root solution of nonlinear equations (Sánchez 2009). In its development the Newton method is also used to find the optimum point of an optimization problems (Silalahi 2014). The Newton method is one of the best techniques to solve nonlinear equations and optimization problems. This method is very easy to implement and often converges quickly, when the iteration begins quite close to the desired root (Kumar 2012).

Some researchers have been looking for the most effective and efficient methods for solving nonlinear equation problems. Some modified Newton methods have been developed by some researchers including Werakoon and Fernando (2000). They modified the Newton method using the trapezoidal rule to produce Newton trapezoidal method. This method has cubic convergence which is better than the Newton method. The results of Werakoon and Fernando have sparked a lot of researches on the Newton method. The study was carried out to obtain a better method for searching root values of nonlinear equation problems. Subsequent researchers are Frontini (2003) and Ozban (2004). They use arithmetic Newton method and harmonics Newton method, both of these methods also produce cubic convergence. Subsequent research conducted by Homeier (2005) that modifying Newton method by using the inverse function and also produces cubic convergence. Then Noor (2006) modified the Halley method with the basic concepts of the Newton method and produce a new development of the Halley method with better performance.

Based on the researches that has been done, this research will use the combination of the the Newton method, the Newton inverse method combined and the Halley method to obtain better iteration, running time and convergent to the exact value.

2. Combination and Formulation of the Newton Method, the Newton Inverse Method and the Halley Method

The Newton method, the Newton inverse method and the Halley method have the same characteristics, they are the iterative methods. For finding a root of a function, the Newton method requires an initial guess point \( x_0 \). Newton method does not require a large cost for searching the root of a function, so that in the combination: the formula of the Newton method is placed in the first step. In general the formula of the Newton method for finding the value of the root of a function is shown in (1), (Luenberger and Ye 2008),

\[
x_n^* = x - \frac{f(x)}{f'(x)}
\]
with \( x = x_0 \) is initial guess point, \( f(x) \) is a function value by substituting the value of \( x \), and \( f'(x) \) is the first derivative function value at \( x \).

After determined the initial method, the next step is to add the Newton inverse method formula in the second step of finding the root of a function. According to Homeier (2005), the Newton inverse method produces fewer number of iterations in the search of the root of a function. The result point \( x_n^* \) in (1) is used as a value that is going to be substituted to find the root value in the second step using the Newton inverse formula as in (2), (Homeier 2005),

\[
\bar{x}_n = x_0 - \frac{f(x_0)}{f'(x_0)}
\]

with \( f'(x_n^*) \) states the first derivative function value at \( x_n^* \) which is obtained from the calculation of (1).

The Halley method contains the second derivative and thus requires much time in the calculation process. Therefore, this method is placed in third step for finding a solution approach of a given function. The general form of the formula Halley method is as follows (Noor et al. 2006),

\[
x_{n+1} = \bar{x}_n - \frac{2f(\bar{x}_n)f'(\bar{x}_n)}{2f'(x_0)f'(\bar{x}_n) - f(x_0)f''(x_0)}
\]

with \( f(\bar{x}_n) \) states a function value at \( \bar{x}_n \), \( f'(\bar{x}_n) \) states the first derivative function value at \( \bar{x}_n \), and \( f''(\bar{x}_n) \) states the second derivative function value at \( \bar{x}_n \).

These three combination methods are applied in each iteration. The result of the combination and formulation methods then used to create an algorithm: combination of the Newton Method, the Newton Inverse Method and the Halley Method (NIH algorithm) as follows:

**NIH algorithm (Newton Invers Halley algorithm)**

1. Step 1. Given the initial function \( f(x) \)
2. Step 2. Given the initial guess point \( x_0 \in \mathbb{R}^n \) and the limits of tolerance \( 0 < \varepsilon < 1 \)
3. Step 3. Given a maximum point of iteration
4. Step 4. Calculate \( x_n^* = x_0 - \frac{f(x_0)}{f'(x_0)} \)
   
   Calculate \( \bar{x}_n = x_0 - \frac{f(x_0)}{f'(x_0)} \left( 1 + \frac{1}{f'(x_n^*)} \right) \)
   
   Calculate \( x_{n+1} = \bar{x}_n - \frac{2f(\bar{x}_n)f'(\bar{x}_n)}{2f'(x_0)f'(\bar{x}_n) - f(x_0)f''(x_0)} \)
5. Step 5. Calculate \( b = x_{n+1} - x_n \)
6. Step 6. If \( b < \varepsilon \) or iteration reaches the maximum point, go to step 4.

3. The functions for Testing

The functions that will be used for data test of computing, with \( \alpha \) is the root of each function described below (Werakoon and Fernando 2000):

\[
\begin{align*}
  f_1(x) &= x^3 + 4x^2 - 10 & \alpha &= 1.36523001341448 \\
  f_2(x) &= \sin^2(x) - x^2 + 1 & \alpha &= 1.40449164821621 \\
  f_3(x) &= x^4 - e^x - 3x + 2 & \alpha &= 0.257530285439771 \\
  f_4(x) &= \cos(x) - x & \alpha &= 0.739085133217458 \\
  f_5(x) &= (x - 1)^3 - 1 & \alpha &= 2.0 \\
  f_6(x) &= x^3 - 10 & \alpha &= 2.15443469003367 \\
  f_7(x) &= x \exp(x^2) - \sin^2(x) + 3 \cos(x) + 5 & \alpha &= -1.20764782713013 \\
  f_8(x) &= x^2 \sin^2(x) + \exp[x^2 \cos(x) \sin(x)] - 28 & \alpha &= 4.82458931731526 \\
  f_9(x) &= \exp(x^2 + 7x - 30) - 1 & \alpha &= 3.0 \\
\end{align*}
\]

4. Results

This research was carried out by using hardware and software as follows: personal computer with the following specifications: Processor Intel (R) Pentium (R) CPU B970@2.30Ghz, RAM 2 GB, mouse and keyboard. The software used is as follows: Operating system Windows 7 Ultimate 32-bit, Microsoft Excel 2013 and a mathematical software.

Computational test is conducted by comparing the numerical results of only the Newton method, only the Halley method and the combination method NIH. The functions \( f_n(x) \) is used for
testing by different initial values. In this test the tolerance $\varepsilon$ is $10^{-4}$.

4.1 The number of iterations
The computational results for the number of iterations of each method with tolerance $\varepsilon = 10^{-4}$ are shown in Table 1. Based on the computational results, the proposed NIH method is better than the Newton method and the Newton inverse method. The total number of iterations for all experiments is 69 which is less than two other method greatly. The starting point $x_0$ and the tolerance also affect the number of iterations. If we use a starting point or initial point that is close enough to the root value, then the iteration process becomes smaller.

4.2 The running time
The running time of each method with tolerance $\varepsilon = 10^{-4}$ are shown in Table 2. The computational results show that the NIH method has the smallest total number of the running time. But we still need further exploration, because for some functions the Newton method is the best.

| $f(x)$ | $x_0$ | Newton | Halley | NIH |
|--------|-------|--------|--------|-----|
| $f_1$  | -0.5  | 130    | 74     | 7   |
|        | 1     | 4      | 4      | 3   |
|        | 2     | 4      | 4      | 3   |
|        | -0.3  | 53     | 53     | 11  |
| $f_2$  | 1     | 5      | 5      | 3   |
|        | 3     | 5      | 5      | 3   |
| $f_3$  | 2     | 2      | 2      | 2   |
|        | 3     | 5      | 4      | 3   |
| $f_4$  | 1     | 3      | 44     | 3   |
|        | 1.7   | 4      | 4      | 3   |
|        | -0.3  | 5      | 5      | 3   |
| $f_5$  | 3.5   | 6      | 4      | 3   |
|        | 2.5   | 5      | 4      | 3   |
| $f_6$  | 1.5   | 5      | 4      | 3   |
| $f_7$  | -2    | 7      | 5      | 4   |
| $f_8$  | 5     | 8      | 7      | 4   |
| $f_9$  | 3.5   | 11     | 5      | 4   |
|        | 3.25  | 7      | 4      | 4   |
| Total  | 269   | 237    | 69     |     |

4.3 The accuracy
The computational results for accuracy (the root value) of each method with tolerance $\varepsilon = 10^{-4}$ are shown in Table 3. Table 3 shows that almost all of the three methods find the expected value of the root of all the given functions. The root values obtained from computational test using a tolerance $\varepsilon = 10^{-4}$ are not different significantly when compared with the root value obtained from Weerakoon and Fernando. But for $f_8$ there are differences between root value of Werakoon and Fernando (4.8246) with the Newton method (3.4374), the Halley Method (3.4374), and the NIH method (4.6221). From table, it can be seen that the root value of the NIH method is closer to the real root value compared to the Newton method and the Halley Method. The NIH Method seem is better than Newton method and the Halley Method for finding a solution of “difficult functions” such as $f_8$. 
Table 2. Comparison of the running time of each method with tolerance $\varepsilon = 10^{-4}$

| $f(x)$ | $x_0$ | Running time (ms) |   |   |   |
|-------|-------|-------------------|---|---|---|
|       |       | Newton | Halley | NIH |
| $f_1$ | -0.5  | 0.1204 | 0.0822 | 0.0280 |
|       | 1     | 0.0049 | 0.0072 | 0.0096 |
|       | 2 | 0.0049 | 0.0059 | 0.0100 |
|       | -0.3 | 0.0438 | 0.0705 | 0.0193 |
| $f_2$ | 1 | 0.0063 | 0.0107 | 0.0107 |
|       | 3 | 0.0064 | 0.0108 | 0.0160 |
| $f_3$ | 2 | 0.0031 | 0.0064 | 0.0117 |
|       | 3 | 0.0070 | 0.0122 | 0.0172 |
| $f_4$ | 1 | 0.0013 | 0.0067 | 0.0088 |
|       | 1.7 | 0.0044 | 0.0087 | 0.0094 |
|       | -0.3 | 0.0053 | 0.0087 | 0.0128 |
| $f_5$ | 3.5 | 0.0066 | 0.0089 | 0.0131 |
|       | 2.5 | 0.0055 | 0.0088 | 0.0090 |
| $f_6$ | 1.5 | 0.0054 | 0.0074 | 0.0088 |
| $f_7$ | -2 | 0.0108 | 0.0143 | 0.0208 |
| $f_8$ | 5 | 0.0149 | 0.0246 | 0.0242 |
| $f_9$ | 3.5 | 0.0146 | 0.0182 | 0.0207 |
|       | 3.25 | 0.0092 | 0.0151 | 0.0170 |
| Total |     | 0.2748 | 0.3273 | 0.2671 |

Table 3. Comparison of the root value of each method for tolerance $\varepsilon = 10^{-4}$

| $f(x)$ | $x_0$ | root value |   |   |   |
|-------|-------|------------|---|---|---|
|       |       | Newton | Halley | NIH |
| $f_1$ | 1.3652 | 1.3652 | 1.3652 | 1.3652 |
|       | 1.3652 | 1.3652 | 1.3652 |
|       | 1.3652 | 1.3652 | 1.3652 |
|       | 1.3652 | 1.3652 | 1.3652 |
| $f_2$ | 1.4045 | 1.4044 | 1.4044 | 1.4044 |
|       | 1.4044 | 1.4044 | 1.4044 |
| $f_3$ | 0.2575 | 0.2575 | 0.2575 | 0.2575 |
|       | 0.2575 | 0.2575 | 0.2575 |
| $f_4$ | 0.7391 | 0.7391 | 0.7391 | 0.7391 |
|       | 0.7391 | 0.7391 | 0.7391 |
| $f_5$ | 2.0000 | 2.0000 | 2.0000 | 2.0000 |
|       | 2.0000 | 2.0000 | 2.0000 |
| $f_6$ | 2.1544 | 2.1544 | 2.1544 | 2.1544 |
| $f_7$ | -1.2076 | -1.2076 | -1.2076 | -1.2076 |
| $f_8$ | 4.8246 | 3.4374 | 3.4374 | 4.6221 |
| $f_9$ | 3.0000 | 3.0000 | 3.0000 | 3.0000 |
|       | 3.0000 | 3.0000 | 3.0000 |
|       | 3.25 | 3.0000 | 3.0000 | 3.0000 |
4. Conclusions
The results showed that a combination of the Newton method, the Newton inverse method and the Halley method (NIH) can be used for finding the root solutions of the given nonlinear functions. Based on the computational results, the combination method in general has better performance in the terms of the number of iterations compared to the Newton method and the Halley method. The computational results show that the combination method has the smallest total number of the running time, but we still need further exploration, because for some functions the Newton method is the best. In terms of accuracy, the NIH method obtains results closer to the roots than other methods especially for the functions that are quite difficult.

Acknowledgment
This work was supported by the directorate general of research strengthening and the development, the ministries of research, technology, and higher education republic of Indonesia (No. 079/SP2H/LT/DRPM/II/2016).

References
[1] Frontini M, Sormain E. 2003. Some Variant of Newton’s Method with Third-Order Convergence. Applied Mathematics and Computation 140: 419-426.
[2] Homeier HHH. 2005. On Newton-type Methods with Cubic Convergence. Journal of Computational and Applied Mathematic 176: 425-432.
[3] Kumar M, Singh AK, Srivastava A. 2012. Various Newton-type Iterative Methods for Solving Nonlinear Equations. Journal of the Egyptian Mathematic Society 21,334-339.
[4] Luenberger DG, Ye Y. 2008. Linear and Nonlinear Programing Third Edition. Stanford, California: Springer.
[5] Noor MA, Khan WA, Hussain A. 2006. A new modified Halley method without second derivatives for nonlinear equation. Applied Mathematics and Computation. 189: 1268–1273
[6] Ozban AY. 2004. Some New Variants of Newton’s Methods. Applied Mathematic Letters 17: 677-682.
[7] Sánchez MG. 2009. Improving Order and Efficiency: Composition with a Modified Newton’s Method. Journal of Computational and Applied Mathematics 231, 592-597.
[8] Silalahi BP. 2014. Sharper analysis of upper bound for the iteration complexity of an interior-point method using primal-dual full-Newton step algorithm. Far East Journal of Mathematical Sciences 95(1): 69-80.
[9] Weerakoon S, Fernando T.G.I. 2000. A Variant of Newton’s Method with Accelerated Third-Order Convergence. Applied Mathematic Letters 13: 87-93.