Hindawi Publishing Corporation
Advances in Numerical Analysis
Volume 2013, Article ID 687382, 7 pages
http://dx.doi.org/10.1155/2013/687382

Research Article

A Two-Point Newton Method Suitable for Nonconvergent Cases and with Super-Quadratic Convergence

Ababu Teklemariam Tiruneh, W. N. Ndlela, and S. J. Nkambule

Department of Environmental Health Science, University of Swaziland, P.O. Box 369, Mbabane H100, Swaziland

Correspondence should be addressed to Ababu Teklemariam Tiruneh; ababute@yahoo.com

Received 25 October 2012; Revised 25 December 2012; Accepted 17 February 2013

Academic Editor: Michele Benzi

An iterative formula based on Newton's method alone is presented for the iterative solutions of equations that ensures convergence in cases where the traditional Newton Method may fail to converge to the desired root. In addition, the method has super-quadratic convergence of order 2.414 (i.e., $1 + \sqrt{2}$). Newton method is said to fail in certain cases leading to oscillation, divergence to increasingly large number, or offshooting away to another root further from the desired domain or offshooting to an invalid domain where the function may not be defined. In addition when the derivative at the iteration point is zero, Newton method stalls. In most of these cases, hybrids of several methods such as Newton, bisection, and secant methods are suggested as substitute methods and Newton method is essentially blended with other methods or altogether abandoned. This paper argues that a solution is still possible in most of these cases by the application of Newton method alone without resorting to other methods and with the same computational effort (two functional evaluations per iteration) like the traditional Newton method. In addition, the proposed modified formula based on Newton method has better convergence characteristics than the traditional Newton method.

1. Introduction

Iterative procedures for solutions of equations are routinely employed in many science and engineering problems. Starting with the classical Newton methods, a number of methods for finding roots of equations have come to exist, each of which has its own advantages and limitations. The Newton method of root finding is based on the iterative formula:

$$x_{k+1} = x_k - \frac{y(x_k)}{y'(x_k)}. \tag{1}$$

Newton's method displays a faster quadratic convergence near the root while it requires evaluation of the function and its derivative at each step of the iteration. However, when the derivative evaluated is zero, Newton method stalls. For low values of the derivative, the Newton iteration offshoots away from the current point of iteration and may possibly converge to a root far away from the intended domain. For certain forms of equations, Newton method diverges or oscillates and fails to converge to the desired root. In addition, the convergence of Newton method can be slow near roots of multiplicity although modifications can be made to increase the rate of convergence [1].

Modifications of the Newton method with higher order convergence have been proposed that require also evaluation of a function and its derivatives. An example of such methods is a third order convergence method by Weerakoon and Fernando [2] that requires evaluation of one function and two first derivatives. A fourth order iterative method, according to Traub [3] also requires evaluation of one function and two derivative evaluations. Grau-Sánchez and Díaz-Barrero [4] gave a compositing of function evaluation at a point and its derivative to improve the convergence of Newton's method from 2 to 4. Recently other methods of fifth, sixth, seventh, and higher order convergence have been proposed [5–11]. While higher order Newton methods ensure faster convergence, their stability for certain equation forms may have the same problem similar to that of the traditional Newton method.

The secant method does not require evaluation of derivatives. However, the rate of convergence is about 1.618, the

convergence may be a problem for some forms of equations, and the secant method may fail to converge in those cases. Muller's method is an extension of the secant method to a quadratic polynomial [12]. It requires three functional evaluations to start with but continues with one function evaluation afterwards. The method does not require derivatives and the rate of convergence near the root is superlinear, that is, about 1.84. However, Muller's method can converge to a complex root from an initial real number [13]. Muller's method also requires the three points to be distinct. If two of the points coincide, the method degenerates to secant method.

Hybrids of methods are also used to provide stability and ensure convergence to a desired root. For example, Newton or secant methods can be combined with bisection to bracket the root by a small interval so that a good initial guess is available for applying Newton method with quadratic convergence. Dekker's method [14] combines bisection method with that of secant method. The method starts by bracketing the root between two initial points that have functional values opposite in sign. The secant estimate of x is compared with the bisection of the interval, the one estimate resulting in small interval with the point of smaller magnitude functional value is chosen, and the iteration continues. Brent's method [15] is a root finding algorithm that combines root bracketing, bisection, and inverse quadratic interpolation. It is a modification of Dekker's method to avoid slow convergence when the difference between consecutive estimates of x is arbitrarily small. In such cases bisection is used for the next root estimate.

The Leap-frogging Newton method [16] uses the Newton method as an intermediate step for the estimation of the root followed by the application of the secant method using the initial point and the intermediate point found by Newton method. The method has cubic convergence and works in certain pathological cases where Newton method may fail. However, being dependent on Newton method for the intermediate step, the method may suffer from the same drawbacks of using the traditional Newton method mentioned above.

2. Method Development

It will be shown that the iterative formula for the two-point method will take the form

$$x_{k+1} = x_{k-1} - \frac{(x_{k-1} - x_k)}{1 - (y_k/y_{k-1})\left(\left((y_k - y_{k-1})/(x_k - x_{k-1})\right)/y'_k\right)}. \tag{2}$$

The method starts by selecting two points lying on a curve $y = f(x)$, namely (x_0, y_0) and (x_1, y_1). A line connecting the two points is drawn from the point (x_0, y_0) to the point (x_1, y_1) as shown in Figure 1. A new variable m (the cotangent of the angle α_1 between this line and the vertical) is defined so that

$$m_1 = \cot(\alpha_1) = \frac{y_0 - y_1}{x_0 - x_1}. \tag{3}$$

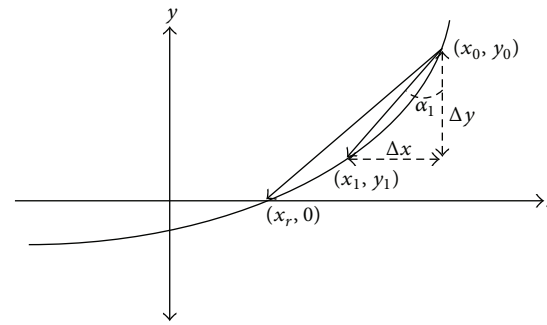

Figure 1: The graph of $y = f(x)$ on which the angle α is defined between the two points.

In general, for any other point (x, y) lying on the curve, the variable m is defined so that:

$$m = \cot(\alpha) = \frac{y_0 - y}{x_0 - x}. \tag{4}$$

Next, Newton method of finding roots will be applied taking m as the independent variable and y as the dependent variable. Denoting that m_r is the estimate of the root for which $y(m) = 0$ from the Newton method with the corresponding value of x_r and applying Newton method gives

$$m_r = m_1 - \frac{y_1}{(dy/dm_1)}. \tag{5}$$

Substituting for m_r and m_1 the equivalent expressions in terms of x and y values, namely,

$$m_r = \frac{y_0 - 0}{x_0 - x_r}, \qquad m_1 = \frac{y_0 - y_1}{x_0 - x_1} \tag{6}$$

into (5) above gives the following expression:

$$\frac{y_0}{x_0 - x_r} = \left(\frac{y_0 - y_1}{x_0 - x_1}\right) - \frac{y_1}{(dy/dm_1)}. \tag{7}$$

The derivative dy/dm is evaluated from $dy/dx = y'$ and dm/dx using the formula

$$\frac{dy}{dm} = \frac{dy/dx}{dm/dx} = \frac{y'}{dm/dx}. \tag{8}$$

The evaluation of the derivative dm/dx in turn gives

$$\frac{dm}{dx} = \frac{d}{dx}\left(\frac{y_0 - y}{x_0 - x}\right) = \frac{(y_0 - y) - (x_0 - x)\,y'}{(x_0 - x)^2}. \tag{9}$$

Therefore, dy/dm can now be written in terms of x, y and y' as follows:

$$\frac{dy}{dm} = \frac{y'(x_0 - x)^2}{(y_0 - y) - (x_0 - x)\,y'}. \tag{10}$$

Substituting the above expression for dy/dm in the equation

$$\frac{y_0}{x_0 - x_r} = \left(\frac{y_0 - y_1}{x_0 - x_1}\right) - \frac{y_1}{(dy/dm_1)} \tag{11}$$

gives

$$\frac{y_0}{x_0 - x_r} = \left(\frac{y_0 - y_1}{x_0 - x_1}\right) - \left[\frac{y_1}{\left(\left(y'(x_0 - x)^2\right)/\left((y_0 - y) - (x_0 - x)\,y'\right)\right)}\right]. \tag{12}$$

Solving for the root estimate x_r and further rearranging results in

$$x_r = x_0 - \frac{(x_0 - x_1)}{1 - (y_0/y_1)\left(((y_1 - y_0)/(x_1 - x_0))/y_1'\right)}. \tag{13}$$

Continuing the iteration using the above formula, for the kth step of the iteration, the $k+1$th estimate of the root will take the form

$$x_{k+1} = x_{k-1} - \frac{(x_{k-1} - x_k)}{1 - (y_k/y_{k-1})\left(((y_k - y_{k-1})/(x_k - x_{k-1}))/y_k'\right)}. \tag{14}$$

Denoting $(y_k - y_{k-1})/(x_k - x_{k-1})$ by $(\Delta y/\Delta x)_k$ will result in the expression

$$x_{k+1} = x_{k-1} - \frac{(x_{k-1} - x_k)}{1 - (y_k/y_{k-1})\left((\Delta y/\Delta x)_k/y_k'\right)}. \tag{15}$$

3. Proof of Super-Quadratic Convergence

Recalling the iteration formula of (14),

$$x_{k+1} = x_{k-1} - \frac{(x_{k-1} - x_k)}{1 - (y_k/y_{k-1})\left(((y_k - y_{k-1})/(x_k - x_{k-1}))/y_k'\right)}. \tag{16}$$

Defining the error at the kth iteration to be $e_k = x_k - r$, where r is the root of the equation desired, the errors at the $k-1$th and kth iteration are also defined similarly as follows:

$$e_{k-1} = x_{k-1} - r, \qquad ek_{+1} = x_{k+1} - r. \tag{17}$$

The iteration formula in (14) can now be rewritten in terms of the error terms as follows:

$$e_{k+1} = e_{k-1} - \frac{(e_{k-1} - e_k)}{1 - (y_k/y_{k-1})\left(((y_k - y_{k-1})/(e_k - e_{k-1}))/y_k'\right)}. \tag{18}$$

Expanding the y_{k-1}, y_k and y_k', terms about the root r using Taylor series expansion,

$$y_k = (y_r = 0) + e_k y_r' + \frac{e_k^2 y_r''}{2} + \frac{e_k^3 y_r'''}{6} + O(e_k)^4, \qquad y_k = C_1 e_k + C_2 e_k^2 + C_3 e_k^3 + O(e_k)^4, \tag{19}$$

where $C_n = y_r^n/n!$ for $n = 1, 2, 3, \ldots$.

The Taylor series expansions of y_{k-1} and y_k' about $x = r$ are formulated similarly as follows:

$$y_{k-1} = C_1 e_{k-1} + C_2 e_{k-1}^2 + C_3 e_{k-1}^3 + O(e_{k-1})^4, \qquad y_k' = C_1 + C_2 e_k + C_3 e_k^2 + C_4 e_k^3 + O(e_k)^4. \tag{20}$$

The algebraic expression of (18), after substituting the previous Taylor series forms, was simplified using MATLAB program. After, the fourth order error terms in e_k and e_{k-1} were discarded and the error terms of the denominator were also discarded compared to the dominant term of C_1^2, the resulting expression will be as follows:

$$e_{k+1} = e_{k-1} - \frac{(e_{k-1} - e_k)}{1 - (y_k/y_{k-1})\left(((y_k - y_{k-1})/(e_k - e_{k-1}))/y_k'\right)} = \left(\frac{C_2^2 - C_1 C_3}{C_1^2}\right) e_k^2 e_{k-1}. \tag{21}$$

Therefore, the error sequence has the simplified expression

$$e_{k+1} = \left(\frac{C_2^2 - C_1 C_3}{C_1^2}\right) e_k^2 e_{k-1}. \tag{22}$$

Defining positive real terms S_k and S_{k-1} so that

$$S_k = \frac{|e_{k+1}|}{|e_k^\alpha|}, \qquad S_{k-1} = \frac{|e_k|}{|e_{k-1}^\alpha|}, \qquad |e_{k+1}| = (S_k S_{k-1}^\alpha)\left|e_{k-1}^{\alpha^2}\right|, \qquad \frac{|e_{k+1}|}{|e_k^2|\,|e_{k-1}|} = \left|\frac{C_2^2 - C_1 C_3}{C_1^2}\right| = \left(S_k S_{k-1}^{\alpha-2}\right)\left|e_{k-1}^{\alpha^2 - 2\alpha - 1}\right|. \tag{23}$$

Near the root, the above expression approaches the constant terms in C. Therefore, the power of the error term $\alpha^2 - 2\alpha - 1$ shall approach zero:

$$\alpha^2 - 2\alpha - 1 = 0, \qquad \alpha = \frac{2 \pm \sqrt{4+4}}{2} = 1 \pm \sqrt{2} \approx 2.414. \tag{24}$$

The positive solution of α is the one for which the error sequence converges to zero. Therefore, the iteration formula of (14) near the root has a convergence of order 2.414 which is super-quadratic.

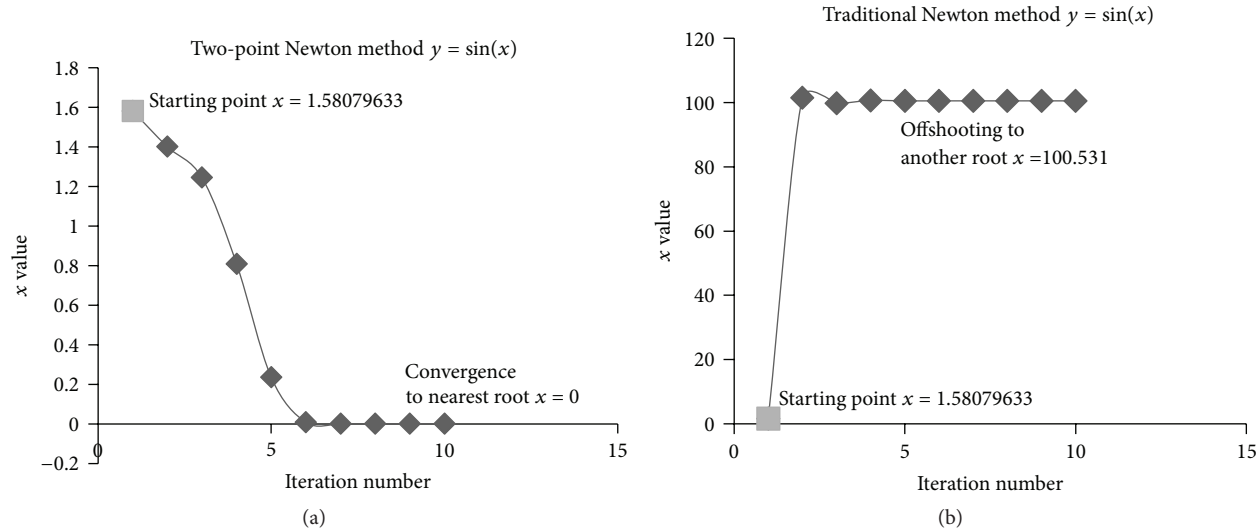

Figure 2: Stability of the proposed two-point Newton method near points of zero derivatives.

4. Stability of the Proposed Method

The proposed method shows stability near points where application of the traditional Newton method may result in oscillation, divergence, or offshooting away from the desired root or offshooting to a possibly invalid domain where the function $y = f(x)$ may not be defined. This stability characteristic may be studied by examining the iteration formula again which takes the following form:

$$x_{k+1} = \left(1 - \frac{1}{r}\right) x_{k-1} + \left(\frac{1}{r}\right) x_k, \tag{25}$$

where the variable r is defined as

$$r = 1 - \left(\frac{y_k}{y_{k-1}}\right)\left(\frac{(y_k - y_{k-1})/(x_k - x_{k-1})}{y'_k}\right). \tag{26}$$

The estimate for the root at the $k + 1$th iteration, x_{k+1}, therefore, can be taken as the weighted sum of the x_{k-1} and x_k values at the $k - 1$th and kth iteration, respectively. The weighing factors are $(1 - 1/r)$ for x_{k-1} and $(1/r)$ for x_k. When the method converges to the root, the value of r approaches unity. This is found by examining the behavior of r near the root and noting that y_k approaches 0 near the root; that is,

$$\begin{aligned}\lim_{k\to\infty}(r) &= 1 - \left(\frac{0}{y_{k-1}}\right)\left(\frac{(y_k - y_{k-1})/(x_k - x_{k-1})}{y'_k}\right) \\ &= 1 - 0 = 1.\end{aligned} \tag{27}$$

The iterative formula will, near the root, approach the following expression:

$$\begin{aligned}\lim_{k\to\infty}(x_{k+1}) &= \left(1 - \frac{1}{1}\right) x_{k-1} + \left(\frac{1}{1}\right) x_k \\ &= (0)\, x_{k-1} + (1)\, x_k.\end{aligned} \tag{28}$$

The iteration, therefore, moves away from x_{k-1} with a weighing factor approaching zero and gives weighing factor of 1 to x_k.

Near points where the derivative of the function may approach zero, the r value approaches infinity and the root estimate weighs heavily in favor of x_{k-1} rather than x_k. This can be seen from the following limit evaluation of r:

$$\begin{aligned}\lim_{dy/dx\to 0}(r) &= 1 - \left(\frac{y_k}{y_{k-1}}\right)\left(\frac{(y_k - y_{k-1})/(x_k - x_{k-1})}{0}\right) \\ &= 1 \pm \infty = \pm\infty, \\ \lim_{dy/dx\to 0}(x_{k+1}) &= \left(1 - \frac{1}{\infty}\right) x_{k-1} + \left(\frac{1}{\infty}\right) x_k \\ &= (1)\, x_{k-1} + (0)\, x_k.\end{aligned} \tag{29}$$

The iteration, therefore, moves away from x_k with a weighing factor approaching zero and gives weighing factor of near to 1 to x_{k-1}.

Near points where the derivative of the function may approach zero, Newton method typically displays oscillation or offshooting behavior. The proposed two-point modified Newton procedure however shows stability without displaying oscillation and offshooting tendencies. For example, for the function $y = \sin(x)$, the derivative of y (i.e., $dy/dx = \cos(x)$) is zero at $x = \pi/2$. Starting Newton method near this point, say $x = (\pi/2 + 0.01)$, will result in offshooting of the iteration to a different domain.

As shown in Figure 2, for a starting point $x = 1.58079633$ where the function $y = \sin(x)$ has near zero derivative, the traditional Newton method of shoots to a root further from the nearest root of $x = 0$. The Newton method as such converges to $x = 100.531$. On the other hand, the proposed two-point Newton method shows stability and converges to the nearest root of $x = 0$.

Another example of the stability of the proposed method is when Newton method diverges instead of converging to the desired root. Figures 3 and 4 show comparison of the iterative values of x for the proposed method and the traditional

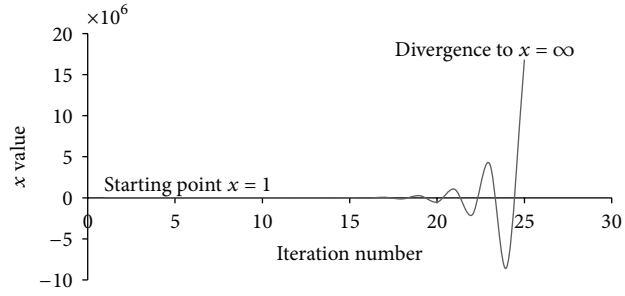

Figure 3: Application of Newton method to $y = x^{1/3}$ leading to oscillating divergence.

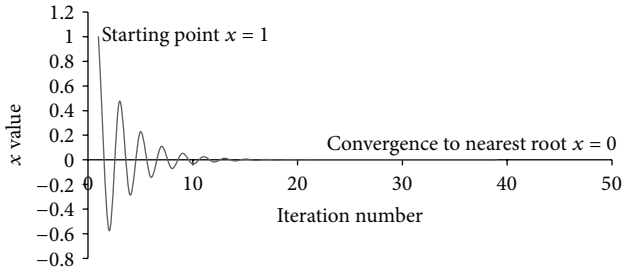

Figure 4: Application of the proposed two-point Newton method to $y = x^{1/3}$ leading to oscillating convergence.

Newton method for the function $y = x^{1/3}$ which is often used to illustrate the pathological condition with respect to application of Newton's method which leads to an oscillating divergence to infinity. By contrast the proposed method displays an oscillating convergence to the desired root albeit with slower rate of convergence of order 1.0.

5. Application Examples

Equations used to test efficiency of root finding methods are used here to evaluate the number of iterations required to reach to a specified level of convergence (Table 1). The stopping criterion used for the iteration process is given by

$$|x_k - x_{k-1}| + |y_k| < 10^{-15}. \tag{30}$$

The rate of convergence towards the root $x = r$ for each step of the iteration is evaluated using the formula

$$\alpha_k = \frac{\mathrm{Log}\left(|e_{k+1}|\right)}{\mathrm{Log}\left(|e_k|\right)} = \frac{\mathrm{Log}\left(|x_{k+1} - r|\right)}{\mathrm{Log}\left(|x_k - r|\right)}. \tag{31}$$

Table 1 shows comparison of the proposed two-point Newton method with the Newton and secant methods for a number of equations used to test efficiency of root finding methods elsewhere. A super-quadratic convergence with which the proposed method converges to the root is mostly evident with α_k values being close to 2.414 during most of the iterations. It can also be seen from Table 1 that a less number of iterations are required to reach convergence for the proposed method than those required for Newton and secant methods. For example, for equations with multiple roots such as $y = (x-1)^6 - 1$, shown in Table 1, Newton method displays linear convergence while the proposed method converges super-quadratically with an order of 2.414. Similar order of convergence is also observed for the equation $y = e^{x^2+7x-30} - 1$.

5.1. Examples Where the Proposed Method Works While Newton Method Fails. The advantage of the use of the proposed two-point Newton method is shown for cases where the Newton method and in several cases also the secant method fail to converge to the root. Table 2 shows the results of the iteration for several examples of equations. In all of the examples listed, Newton method fails to converge whereas the proposed two-point Newton method converges. Secant method also fails to converge in several of the examples cited in Table 2. For example, in the case of $y = \mathrm{Log}(x)$, Newton method starting with $x = 3$ fails with the second iteration because the estimated x value is a negative number whose logarithm is undefined. For the case $y = \arctan(x)$ Newton method always diverges to increasingly large number for any starting x value while the proposed two-point Newton method converges super-quadratically near the root. For $y = x^{1/3}$, Newton method also diverges whereby each iteration gives an estimate of the root which is twice the previous value and with alternate signs leading to oscillating divergence. On the other hand application of the proposed two-point modification of Newton method leads to oscillating convergence at somehow reduced rate of convergence (of order 1 as explained above).

For the equation $y = 10xe^{-x^2} - 1$, both of Newton and secant methods continue to diverge to increasingly large x values while the proposed two-point Newton method shows stability of convergence. For the polynomials of different degrees cited in the Table 2, Newton and secant methods display oscillation for the starting values shown in the Table 2 while the proposed two-point Newton method displays stability and super-quadratic convergence for iteration near the root.

6. Conclusion

A numerical procedure of root finding using two-point modification of Newton method has been presented. It is proved that the method has a super-quadratic convergence of order about 2.414. The method is based on application of Newton iteration formula by taking as the independent variable the cotangent of the angle between the line connecting the two successive points of iteration with the vertical and as the dependent variable the given function $y = f(x)$. The resulting iteration formula for root estimation is shown to be the weighted sum of the estimates of the two previous iterations with a weighing factor that penalizes the iteration point having undesirable characteristics such as a near zero derivative. For example, near a point where the derivative is zero, the weighing factor for that point will be near zero effectively moving the iteration away from that undesirable point.

Table 1: Comparison of results of iterations of the two-point Newton method with Newton and secant methods.

Function	Root	Starting point	Secant method	Newton method	Two-point Newton method
$y = x^3 + 4x^2 - 10$	1.365230013414100	0.5	10	8	6
		1	9	6	5
$y = [\sin(x)]^2 - x^2 + 1$	−1.404491648215340	−1	9	7	5
		−3	9	7	6
$y = (x-2)(x+2)^4$	−2.0000000000000	−3	168	119	83
		1.4	116	81	60
$y = (x-1)^6 - 1$	2.00000000000000	1.5	25	17	8
		2.5	12	8	6
		3.5	16	11	8
$y = \sin(x) \cdot e^x + \ln(x^2+1)$	−0.603231971557215	−0.8	8	7	5
		−0.65	8	5	4
$y = e^{x^2+7x-30} - 1$	3.000000000000000	4	29	20	14
		4.5	39	28	18
$y = x - 3\ln(x)$	1.857183860207840	2	8	5	4
		0.5	11	8	5

Table 2: Results of application of the proposed method for cases in which Newton or secant method fails to converge.

Function	Root	Comparison of number of iterations required			
		Starting point	Secant method	Newton method	Two-point Newton method
$y = -x^4 + 3x^2 + 2$	1.887207676120680	1	11	Oscillates	7
		0.5	23	Oscillates	6
$y = \log(x)$	1.000000000000000	3.0	Fails	Fails	5
$y = \arctan(x)$	0.0000000000000	3.0	Diverges	Diverges	6
		−3.0	Diverges	Diverges	6
$y = x^5 - x + 1$	−1.167303978261420	2.0	Oscillates	Oscillates	12
		3.0	14	Oscillates	15
$y = 0.5x^3 - 6x^2 + 21.5x - 22$	4.00000000000000	3.0	10	Oscillates	7
		5.0	8	Oscillates	6
$y = x^{1/3}$	0.00000000000000	1.0	Oscillates	Diverges	101
		−1.0	Oscillates	Diverges	101
$y = 10xe^{-x^2} - 1$	1.679630610428450	3.0	Diverges	Diverges	8
	0.101025848315685	−1.0	Diverges	Diverges	11

Application examples have been given to demonstrate that the proposed method requires fewer number of iterations for convergence to a root than the traditional Newton and secant methods. The method offers a particular advantage for cases where the traditional Newton method and its variants of various order convergence may not converge. A number of examples are given where the proposed method converges to a root in a stable manner without oscillation, divergence, or offshooting from the desired domain of the root, whereas the Newton method and in several of the cases also the secant method display undesirable behaviors which prevent convergence to the desired root.

In terms of computational effort, the proposed method requires one function evaluation and one derivative evaluation at each step of the iteration except for the first step where two functional evaluations are required. The proposed method, therefore, requires essentially the same number of functional evaluations as the traditional Newton method while offering a super-quadratic convergence of order 2.414.

References

[1] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, 5th edition, 1994.

[2] S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000.

[3] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1964.

[4] M. Grau-Sánchez and J. L. Díaz-Barrero, "A technique to composite a modified Newton's method for solving nonlinear equations," *Annals of the University of Bucharest*, vol. 2, no. 1, pp. 53–61, 2011.

[5] J. R. Sharma and R. K. Guha, "A family of modified Ostrowski methods with accelerated sixth order convergence," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 111–115, 2007.

[6] C. Chun, "Some improvements of Jarratt's method with sixth-order convergence," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1432–1437, 2007.

[7] J. Kou and X. Wang, "Sixth-order variants of Chebyshev-Halley methods for solving non-linear equations," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1839–1843, 2007.

[8] J. Kou, "On Chebyshev-Halley methods with sixth-order convergence for solving non-linear equations," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 126–131, 2007.

[9] J. Kou and Y. Li, "An improvement of the Jarratt method," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1816–1821, 2007.

[10] J. Kou, Y. Li, and X. Wang, "Some modifications of Newton's method with fifth-order convergence," *Journal of Computational and Applied Mathematics*, vol. 209, no. 2, pp. 146–152, 2007.

[11] S. K. Parhi and D. K. Gupta, "A sixth order method for nonlinear equations," *Applied Mathematics and Computation*, vol. 203, no. 1, pp. 50–55, 2008.

[12] D. E. Muller, "A method for solving algebraic equations using an automatic computer," *Mathematical Tables and Other Aids to Computation*, vol. 10, pp. 208–215, 1956.

[13] W. R. Mekwi, *Iterative methods for roots of polynomials [M.S. thesis]*, University of Oxford, 2001.

[14] T. J. Dekker, "Finding a zero by means of successive linear interpolation," in *Constructive Aspects of the Fundamental Theorem of Algebra*, B. Dejon and P. Henrici, Eds., Wiley-Interscience, London, UK, 1969.

[15] R. P. Brent, *Algorithms for Minimization without Derivatives*, chapter 4, Prentice-Hall, Englewood Cliffs, NJ, USA, 1973.

[16] A. B. Kasturiarachi, "Leap-frogging Newton's method," *International Journal of Mathematical Education in Science and Technology*, vol. 33, no. 4, pp. 521–527, 2002.