An Efficient Three-step Iterative Methods Based on Bernstein Quadrature Formula for Solving Nonlinear Equations

Huda J. Saeed, Noori Y. Abdul-Hassan

Department of Mathematics, College of Education for Pure Sciences, University of Basrah, Basrah, Iraq.

*Corresponding author E-mail: nooriyasir65@gmail.com

Doi:10.29072/basjs.202130

Abstract

In this study, we suggest and analyze two new one-parameter families of an efficient iterative methods free from higher derivatives for solving nonlinear equations based on Newton theorem of calculus and Bernstein quadrature formula, Bernoulli polynomial basis, Taylor’s expansion and some numerical techniques. We prove that the new iterative methods reach orders of convergence ten with six and eight with four functional evaluations per iteration, which implies that the efficiency index of the new iterative methods is $(10)^{1/6} \approx 1.4678$ and $(8)^{1/4} \approx 1.6818$ respectively. Numerical examples are provided to show the efficiency and performance of our iterative methods, compare to Newton’s method and other relevant methods.
1. Introduction:

A frequently occurring and most important problem in mathematics, science and engineering is how to find the solution of nonlinear equations which can be expressed in general as follows

\[ f(x) = 0, \quad (1) \]

where \( f : D \subset \mathbb{R} \rightarrow \mathbb{R} \) is a scalar function on an open interval \( D \).

Since the numerical analysis is to devise algorithms that give quick and accurate answers to mathematical problems for scientists and engineers, nowadays using computers. Therefore, numerically iterative methods are often the only choice for solving this general problem.

The Newton’s method is one of the famous classical iterative methods to find the root of equation (1). The iterative scheme is given by

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

which it has quadratic convergence, [1]. In the recent past, much attention has been given to developed several iterative methods for solving the nonlinear equations. Many of iterative methods have been obtained by using different techniques such as Taylor expansion, decomposition, homotopy, variational iteration, geometric methods and quadrature formulas also, we know that quadrature formula plays an important role in the evaluation of the numerical integrals.

The first study of quadrature formula was by S. Weerakoon and T.G.I. Fernando in 2000, studied new variant of Newton's method based on trapezoidal instead of a rectangle and they got new two-step iterative method. It has third-order convergence, [2], defined by

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

\[ x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n)+f'(y_n)}, \quad n = 0, 1, ... \]

By improving Newton’s method say, V. I. Hasanov et al. in 2002, modified Newton’s method by approximate the definite integral in quadrature rule by using Simpson’s formula and they obtained a new two-step iterative method with third-order convergence, [3]. G. Nedzhibov in
2002, gave several classes of two-step iterative methods by using different quadrature rules, [4].
M. Frontini and E. Sormani in 2003, extended the results of the iterative methods in [2], and got
new two-step iterative method of order three independent of the integration formula, [5]. A. Y.
Ozban et al. in 2004, presented some new two-step variant of Newton’s method based on harmonic
mean and midpoint integration rule, [6]. H.H.H. Homeier in 2005, modified the iterative method
in [2] by using Newton’s theorem for the invers function and he got new classes of iterative
methods with cubic convergence, [7]. M. A. Noor in 2007, suggested new two-step iterative
methods also by using quadrature formula, [8]. L. Liu and X. Wang in 2010, proposed new three-
step iteration scheme by using the method of weight functions, [9]. M. A. Noor et al. in 2010,
suggested and analyzed some new iterative methods for solving the nonlinear equations using the
decomposition technique coupled with the system of equations, [10]. X. Wang and L. Liu in 2010,
derived two new three-step iterative methods based on Newton’s method and modified
Ostrowski’s method with an eighth-order convergence for solving the simple roots of nonlinear
Presented a new three-step family of eighth-order methods obtained an eight-order convergence
based on Ostrowski’s method, [12]. J. Jayakumar in 2013, proposed a generalization of two-step
Simpson-Newton's method where Simpson's integration rule is applied for approximating the
definite integral in quadrature formula, [13]. J. R. Sharma and H. Arora in 2014, presented a family
of three-point iterative methods for solving nonlinear equations, [14]. O. Oghovese and E. O. John
in 2014, introduced new two-step family of iterative method based on composite trapezoid rule
and fundamental theorem of calculus, [15]. O. Oghovese and E. O. John in 2014, proposed a new
three steps iterative method of order six for solving nonlinear equations, [16]. A.A. Al-Harbi and
I.A. Al-Subaihi in 2015, a new family of three-step optimal eighth-order iterative methods are
presented, [17]. M. Saqib and M. Iqbal in 2017, used quadrature rule to approximate the definite
integral by rectangle integral rule and midpoint integral rule and they obtained new two-step
iterative methods, [18]. R. Thukral in 2018, proposed new three-step Simpson's type method
requires the same number of evaluations of the function as classical method but of fifth order
convergence, [19]. U. k. Qureshi in 2019, Suggested a new iterative method of order two which is
derived from quadrature formula by approximate the definite integral by using composite
trapezoidal rule and some numerical techniques, [20]. G. Sana et al. in 2020, introduced two new
three-step iterative schemes by applied quadrature formula and decomposition approach, [21]. B.
Neta in 2021, developed a derivative-free method with memory based on Traub’s method as the
first step, [22]. C. Zalinescu in 2021, introduced several methods for comparing two convergent iterative processes for the same problem, [23]. G. Sana et al. in 2021, suggested and analyzed some new q-iterative methods by using the q-analogue of the Taylor’s series and the coupled system technique, [24].

In this paper, we present new families of iterative methods for solving equation (1) by using Bernstein integration formula to approximate the definite integral in the quadrature rule and we find that some of well-known iterative methods can be deduced as special cases from the proposed iterative methods. We approximate the higher derivatives in the new three-step iterative methods to reduce the number of functions needed in each iteration to update the efficiency index. Also, we introduce some numerical examples that confirm the theoretical results allow us to compare these methods with Newton’s method and with other relevant methods. Moreover, we introduce the graphical analysis for the uphold of numerical results.

2. Preliminaries

Offers some basic definitions, theorem and lemma that we need in our work.

**Definition 2.1, [25]**: A sequence of iterates \( \{x_n\} \) is said to converge to the root \( \alpha \in \mathbb{R} \) if

\[
\lim_{n \to \infty} |x_n - \alpha| = 0.
\]

If \( x_n, x_{n-1}, \ldots, x_{n-m+1} \) are \( m \) approximates to a root, then we write an iteration method in the form

\[
x_{n+1} = \varphi(x_n, x_{n-1}, \ldots, x_{n-m+1}),
\]

where we have written the equation (1) in the equivalent form

\[
x = \varphi(x)
\]

The function \( \varphi \) is called the iteration function. For \( m = 1 \), we get the one-point iteration method

\[
x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \ldots
\]
If \( \varphi(x) \) is continuous in the interval \([a, b]\) that contains the root and \(|\varphi'(x)| \leq c < 1\) in this interval, then for any choice of \(x_0 \in [a, b]\), the sequence of iterates \(\{x_n\}\) obtained from (4) converges to the root of \(x = \varphi(x)\) or \(f(x) = 0\).

Thus, for any iterative method of the form (3) or (4), we need the iteration function \(\varphi(x)\) and one or more initial approximations to the root.

In practical applications, it is not always possible to find \(\alpha\) exactly. We therefore attempt to obtain an approximate root \(x_{n+1}\) such that

\[
|f(x_{n+1})| < \varepsilon \tag{5}
\]

and/or

\[
|x_{n+1} - x_n| < \varepsilon \tag{6}
\]

where \(x_n\) and \(x_{n+1}\) are two consecutive iterates and \(\varepsilon\) is the prescribed error tolerance.

**Definition 2.2, [2]:** Let \(f: D \subset \mathbb{R} \to \mathbb{R}\) is a scalar function on an open interval \(D\) with a simple root \(\alpha\) of the nonlinear equation. An iterative method is said to have an integer order of convergence \(p\) if it produces the sequence \(\{x_n\}\) of real numbers such that

\[
\lim_{x \to \alpha} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = A \neq 0,
\]

for some \(A \neq 0\) and \(p \geq 1\), then \(p\) is said to be the order of convergence of the sequence, and \(A\) is known as the asymptotic error constant.

or equivalently

\[
x_{n+1} - \alpha = A(x_n - \alpha)^p + O((x_n - \alpha)^{p+1})
\]

**Notation 2.1, [2]:** Let \(e_n = x_n - \alpha\) is the error in the \(n^{th}\) iteration. The equation

\[
e_{n+1} = c e_n^p + O(e_{n+1}^{p+1})
\]

is called the error equation for the method, \(p\) being the order of convergence.
Definition 2.3, [13,2]: Let $\alpha$ be a root of the nonlinear equation and suppose that $x_{n-1}, x_n$ and $x_{n+1}$ are three successive iterations closer to the root $\alpha$. Then, the computational order of convergence (COC) denoted by $\rho$ can be computed using the formula

$$\rho = \frac{ln|e_{n+1}/e_n|}{ln|e_n/e_{n-1}|}.$$ 

Definition 2.4, [19]: The efficiency of a method is measured by the index

$$E.I = \frac{1}{p\omega},$$

where $p$ is the order of convergence and $\omega$ is the total number of function evaluations per iteration.

Theorem 2.1, [17]: Let $\psi_1(x), \psi_2(x), \ldots, \psi_r(x)$ be iterative functions with the orders $s_1, s_2, \ldots, s_r$, respectively. Then the composition of iterative functions

$$\psi(x) = \psi_1(x)(\psi_2(x)(\ldots(\psi_r(x))\ldots))$$

defines the iterative method of the order $s_1s_2\ldots s_r$.

Corollary 2.1, [26,27]: For a continuous function $f(x)$ on $[0, 1]$, we have

$$\int_a^b f'(x) \, dx \approx B_m(f', x) = \frac{b - a}{m + 1} \sum_{k=0}^{m} f'(a + (b - a) \frac{k}{m}).$$

3. Construct of New Iterative Methods

In this section, we construct new Newton-type iterative methods and their modifications based on Newton’s theorem of calculus and Bernstein quadrature formula.

Let $\alpha \in D$ be a simple root of equation (1) and $x_0$ is initial guess sufficiently close to $\alpha$.

Consider Newton’s theorem of calculus, defined by

$$f(x) = f(x_0) + \int_{x_0}^{x} f'(\lambda) \, d\lambda$$

(7)

If we approximate the definite integral in equation (7), by using Corollary 2.1, we have

$$f(x) = f(x_0) + \frac{(x - x_0)}{m + 1} \sum_{k=0}^{m} f'(x_0 + (x - x_0) \frac{k}{m})$$

(8)
From equation (1) and solving equation (8) for \( x \), we obtain

\[
x = x_0 - \frac{(m+1)f(x_0)}{\sum_{k=0}^{m} f'(x_0 + (x-x_0)\frac{k}{m})}
\]

(9)

Since the equation (9) is implicit we can overcome this by approximate \((x - x_0)\) in the right-hand side by \(-\beta \frac{f(x_0)}{f'(x_0)}\), we obtain

\[
x = x_0 - \frac{(m+1)f(x_0)}{\sum_{k=0}^{m} f'(x_0 - \beta \frac{k}{m} f(x_0) f'(x_0))}
\]

(10)

Also, we can get from equation (10) by using Taylor expansion of \( f'(x_0 - \beta \frac{k}{m} f(x_0) f'(x_0)) \), and neglecting the terms of the third order and above, we have

\[
x = x_0 - \frac{(m+1)f(x_0)f'(x_0)}{\sum_{k=0}^{m} (f'(x_0))^2 - \beta \frac{k}{m} f(x_0)f''(x_0)}
\]

(11)

Now, by using equations (10) and (11), we can suggest the following new one-step, two-step and three-step one-parameter family of iterative methods for solving nonlinear equation (1), respectively.

**Algorithm 2.1:** For a given \( x_0 \), compute the approximate solution \( x_{n+1} \) by the following iterative method

\[
x_{n+1} = x_n - \frac{(m+1)f(x_n)}{\sum_{k=0}^{m} f'(x_n - \beta \frac{k}{m} f(x_n) f'(x_n))}, \quad n = 0, 1, ...
\]

when \( \beta = 0 \) and \( m = 1 \), we get Newton’s method.

**Algorithm 2.2:** For a given \( x_0 \), compute the approximate solution \( x_{n+1} \) by the following iterative method

\[
x_{n+1} = x_n - \frac{(m+1)f(x_n)f'(x_n)}{\sum_{k=0}^{m} (f'(x_n))^2 - \beta \frac{k}{m} f(x_n)f''(x_n)}}, \quad n = 0, 1, ...
\]

when \( \beta = 0 \) and \( m = 1 \), we get Newton’s method and also, when \( \beta = 1 \) and \( m = 1 \), we get Halley’s method in [28].
Algorithm 2.3: For a given $x_0$, compute the approximate solution $x_{n+1}$ by the following iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}.$$  

$$x_{n+1} = y_n - \frac{(m+1)f(y_n)}{\sum_{k=0}^{m} f'(y_{n+1})} \quad n = 0, 1, ...$$

Algorithm 2.4: For a given $x_0$, compute the approximate solution $x_{n+1}$ by the following iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}.$$ 

$$x_{n+1} = y_n - \frac{(m+1)f'(y_n)'}{\sum_{k=0}^{m} (f'(y_{n+1}))^2 - \beta \frac{k}{m} f(y_{n+1}) f'(y_{n+1})} \quad n = 0, 1, ...$$

Algorithm 2.5: For a given $x_0$, compute the approximate solution $x_{n+1}$ by the following iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}.$$

$$z_n = y_n - \frac{(m+1)f(y_n)'(y_n)}{\sum_{k=0}^{m} f'(y_{n+1})}.$$ 

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \quad n = 0, 1, ...$$

Algorithm 2.6: For a given $x_0$, compute the approximate solution $x_{n+1}$ by the following iterative method

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}.$$ 

$$z_n = y_n - \frac{(m+1)f'(y_n)'}{\sum_{k=0}^{m} (f'(y_{n+1}))^2 - \beta \frac{k}{m} f(y_{n+1}) f'(y_{n+1})}.$$ 

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \quad n = 0, 1, ...$$
Since the purpose of our research is to obtain an efficient three-step iterative methods of higher order of convergence for solving equation (1), so we depend on Algorithms (3.5) and (3.6). Furthermore, to update the efficiency index of our iterative methods so, we approximate the derivatives $f'(y_n), f''(y_n)$ and $f'(z_n)$ respectively, to reduce the number of functional evaluations needed in each step of iteration by using an orthogonal polynomial as basis. This idea is very important and plays a significant part in developing many iterative methods. Now we look for an approximation of $f'(y_n), f''(y_n)$ and $f'(z_n)$ respectively.

Consider the function

$$Q(t) = \sum_{j=0}^{r} a_j B_j(t - \xi_n)$$  \hspace{1cm} (12)

where $\xi_n \in \{x_n, y_n, z_n\}$, $a_j, j = 0, 1, 2, ..., r$ are unknowns to be found, and $B_j, j = 0, 1, 2, ..., r$ are forms Bernoulli basis polynomial, defined by

$$B_0(t - \xi_n) = 1, \quad B_1(t - \xi_n) = (t - \xi_n) - \frac{1}{2}, \quad B_2(t - \xi_n) = (t - \xi_n)^2 - (t - \xi_n) + \frac{1}{6}$$

and

$$B_3(t - \xi_n) = (t - \xi_n)^3 - \frac{3}{2} (t - \xi_n)^2 + \frac{1}{2} (t - \xi_n), \quad \xi_n \in \{x_n, y_n, z_n\}.$$

To approximate $f'(y_n)$ we construct a Bernoulli interpolation polynomial, that meets the interpolation conditions

$$f(x_n) = Q(x_n), \quad f'(x_n) = Q'(x_n) \quad \text{and} \quad f(y_n) = Q(y_n).$$

Here, take $r = 2$ and $\xi_n = y_n$ and from equation (12), then $Q(t)$ can be written as:

$$Q(t) = a_0 B_0(t - y_n) + a_1 B_1(t - y_n) + a_2 B_2(t - y_n).$$

Applying the interpolation conditions above on $Q(t)$, we get

$$f(x_n) = a_0 + a_1 \left( (x_n - y_n) - \frac{1}{2} \right) + a_2 \left( (x_n - y_n)^2 - (x_n - y_n) + \frac{1}{6} \right),$$

$$f(y_n) = a_0 - \frac{1}{2} a_1 + \frac{1}{6} a_2,$$

$$f'(x_n) = a_1 + a_2 (2(x_n - y_n) - 1).$$
Solving the system above of three linear equations of three unknowns, we obtain

\[ a_0 = f(x_n) - \left( (x_n - y_n) - \frac{1}{2} \right) a_1 - \left( (x_n - y_n)^2 - (x_n - y_n) + \frac{1}{6} \right) a_2, \]

\[ a_1 = \left( \frac{2}{x_n - y_n} - \frac{1}{(x_n - y_n)^2} \right) f(x_n) + \left( \frac{1}{(x_n - y_n)^2} - \frac{2}{x_n - y_n} \right) f(y_n) + \left( \frac{1}{x_n - y_n} - 1 \right) f'(x_n), \text{ and} \]

\[ a_2 = \frac{1}{(x_n - y_n)^2} (f(y_n) - f(x_n)) + \frac{1}{x_n - y_n} f'(x_n). \]

After substituting the values of \( a_1 \) and \( a_2 \) in equation \( f'(y_n) = a_1 - a_2 \), we get

\[ f'(y_n) = \frac{2}{x_n - y_n} (f(x_n) - f(y_n)) - f'(x_n) := H_1(x_n, y_n) \tag{13} \]

Also, to approximate \( f''(y_n) \) we construct a Bernoulli interpolation polynomial, that meets the interpolation conditions

\[ f(x_n) = Q(x_n), f(y_n) = Q(y_n) \text{ and } f'(y_n) = Q'(y_n). \]

Take \( r = 2 \) and \( \xi_n = y_n \) and from equation (12), then \( Q(t) \) can be written as:

\[ Q(t) = a_0 B_0(t - y_n) + a_1 B_1(t - y_n) + a_2 B_2(t - y_n). \]

Again, applying the interpolation conditions above on \( Q(t) \), we get

\[ f(x_n) = a_0 + a_1 \left( (x_n - y_n) - \frac{1}{2} \right) + a_2 \left( (x_n - y_n)^2 - (x_n - y_n) + \frac{1}{6} \right), \]

\[ f(y_n) = a_0 - \frac{1}{2} a_1 + \frac{1}{6} a_2, \]

\[ f'(y_n) = a_1 - a_2. \]

Then by solving the system above of three linear equations of three unknowns, we obtain

\[ a_0 = f(x_n) - \left( (x_n - y_n) - \frac{1}{2} \right) a_1 - \left( (x_n - y_n)^2 - (x_n - y_n) + \frac{1}{6} \right) a_2, \]

\[ a_1 = -\frac{1}{(x_n - y_n)^2} (f(y_n) - f(x_n)) + (1 - (x_n - y_n)) a_2, \text{ and} \]

\[ a_2 = \frac{1}{(x_n - y_n)^2} f(x_n) - \frac{1}{(x_n - y_n)^2} f(y_n) - \frac{1}{(x_n - y_n)} f'(y_n). \]
After substituting the values of $a_2$ in equation $f''(y_n) = 2a_2$, we get
\[
f''(y_n) = \frac{2}{(x_n-y_n)^2} (f(x_n) - f(y_n)) - \frac{2}{(x_n-y_n)^2} H_1(x_n, y_n) := H_2(x_n, y_n). \tag{14}
\]

Finally, to approximate $f'(z_n)$ we construct a Bernoulli interpolation polynomial, that meets the interpolation conditions
\[
f(x_n) = Q(x_n), f(y_n) = Q(y_n), f(z_n) = Q(z_n) \text{ and } f'(x_n) = Q'(x_n).
\]
And take, $r = 3$ and $\xi_n = z_n$ and from equation (12), then $Q(t)$ can be written as:
\[
Q(t) = a_0 B_0(t - z_n) + a_1 B_1(t - z_n) + a_2 B_2(t - z_n) + a_3 B_3(t - z_n).
\]
Also, by applying the interpolation conditions above on equation $Q(t)$, we get
\[
f(x_n) = a_0 + a_1 \left((x_n - z_n) - \frac{1}{2}\right) + a_2 \left((x_n - z_n)^2 - (x_n - z_n) + \frac{1}{6}\right) + a_3 \left((x_n - z_n)^3 - \frac{3}{2} (x_n - z_n)^2 + \frac{1}{2} (x_n - z_n)\right),
\]
\[
f(y_n) = a_0 + a_1 \left((y_n - z_n) - \frac{1}{2}\right) + a_2 \left((y_n - z_n)^2 - (y_n - z_n) + \frac{1}{6}\right) + a_3 \left((y_n - z_n)^3 - \frac{3}{2} (y_n - z_n)^2 + \frac{1}{2} (y_n - z_n)\right),
\]
\[
f(z_n) = a_0 - \frac{1}{2} a_1 + \frac{1}{6} a_2,
\]
\[
f'(x_n) = a_1 + a_2 (2(x_n - z_n) - 1) + a_3 \left(3(x_n - z_n)^2 - 3(x_n - z_n) + \frac{1}{2}\right).
\]
Solving the system above of four linear equations of four unknowns, we obtain
\[
a_0 = f(x_n) - \left(a - \frac{1}{2}\right) a_1 - \left(a^2 - a + \frac{1}{6}\right) a_2 - \left(a^3 - \frac{3}{2} a^2 + \frac{1}{2} a\right) a_3,
\]
\[
a_1 = \frac{1}{b-a} (f(y_n) - f(x_n)) - (b + a - 1) a_2 - \left(b^2 + ab + a^2 - \frac{3}{2} (b + a) + \frac{1}{2}\right) a_3,
\]
\[
a_2 = \frac{1}{ab} (f(z_n) - f(x_n)) + \frac{1}{b(b-a)} (f(y_n) - f(x_n)) - \left(b + a - \frac{3}{2}\right) a_3, \text{ and}
\]
\[
a_3 = -\frac{1}{a(b-a)} f'(x_n) - \frac{1}{a^2 b} (f(z_n) - f(x_n)) + \frac{1}{b(b-a)^2} (f(y_n) - f(x_n)).
\]
where \((x_n - z_n) = a\) and \((y_n - z_n) = b\).

After substituting the values of \(a_1, a_2\) and \(a_3\) in equation \(f'(z_n) = a_1 - a_2 + \frac{1}{2} a_3\), we get

\[
f'(z_n) = 2 \left( \frac{f(x_n) - f(\bar{x}_n)}{x_n - \bar{x}_n} - \frac{f(y_n) - f(\bar{x}_n)}{y_n - \bar{x}_n} \right) + \frac{f(y_n) - f(\bar{x}_n)}{y_n - \bar{x}_n} + \frac{y_n - z_n}{y_n - x_n} \left( \frac{f(y_n) - f(x_n)}{y_n - x_n} - f'(x_n) \right) := H_3(x_n, y_n, z_n)
\]

Therefore, we suggest new three-step one-parameter families of iterative methods free from second derivative for solving nonlinear equation (1) as follows:

**Algorithm 3.7:** For a given \(x_0\), compute the approximate solution \(x_{n+1}\) by the following iterative method

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}.
\]

\[
z_n = y_n - \frac{(m+1)f(y_n)}{\sum_{k=0}^{m} \beta^k f(y_n)}.
\]

\[
x_{n+1} = z_n - \frac{f(z_n)}{H_3(x_n, y_n, z_n)} \cdot n = 0, 1, ...
\]

**Algorithm 3.8:** For a given \(x_0\), compute the approximate solution \(x_{n+1}\) by the following iterative method

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}.
\]

\[
z_n = y_n - \frac{(m+1)f(y_n)H_3(x_n, y_n)}{\sum_{k=0}^{m} (H_3(x_n, y_n))^2 - \beta^k m f(y_n)H_2(x_n, y_n)}.
\]

\[
x_{n+1} = z_n - \frac{f(z_n)}{H_3(x_n, y_n, z_n)} \cdot n = 0, 1, ...
\]

**4. Analysis of Convergence**

In the following Theorems, we establish the convergence of the present Algorithms (3.7) and (3.8) respectively, when \(m = 1\).
Theorem 3.1: Let $\alpha \in D$, be a simple root of a sufficiently differentiable function $f: D \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval $D$. If $x_0$ is sufficiently close to $\alpha$, then the Algorithm 3.7 has tenth-order convergence when $\beta = 1$, while its convergence of eighth order for any $\beta \in \mathbb{R} - \{1\}$.

Proof: Let $\alpha$ be a simple root of $f(x) = 0$. (Since, $f(\alpha) = 0$ and $f'(\alpha) \neq 0$).

Expanding $f(x_n)$ and $f'(x_n)$ by using Taylor expansion about $\alpha$, we get

$$f(x_n) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + O(e_n^4)]$$

(16)

Where $c_s = \frac{f^{(s)}(\alpha)}{s!f'(\alpha)}$, $s = 2, 3, \ldots$ & $e_n = x_n - \alpha$. From (23), we have

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)]$$

(17)

Dividing equation (16) by (17), we get

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2e_n^2 + 2(c_2^2 - c_3)e_n^3 + (-3c_4 + 7c_2c_3 - 4c_2^3)e_n^4 + \ldots$$

(18)

Also, we need to compute

$$y_n = \alpha + c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 + \ldots$$

(19)

Expanding $f(y_n)$ and $f'(y_n)$ about $\alpha$ and using (19) we have

$$f(y_n) = f'(\alpha)[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 5c_2^3)e_n^4 + \ldots]$$

(20)

$$f'(y_n) = f'(\alpha)[1 + 2c_2^2e_n^2 + 4(c_2c_3 - c_2^3)e_n^3 + (6c_2c_4 - 11c_2^2c_3 + 8c_2^3)e_n^4 + \ldots]$$

(21)

$$H_1(x_n, y_n) = f'(\alpha)[1 + (2c_2^2 - c_3)e_n^2 + (6c_2c_3 - 2c_4 - 4c_2^3)e_n^3 + \ldots]$$

(22)

$$f'(y_n - \beta \frac{f(y_n)}{H_1(x_n, y_n)}) = f'(\alpha)[1 - 2c_2^2(\beta - 1)e_n^2 + 4c_2(c_2^2 - c_3)(\beta - 1)e_n^3 + \ldots]$$

(23)

$$z_n = \alpha - c_2^2(\beta - 1)e_n^4 + 4c_2^2(c_2^2 - c_3)(\beta - 1)e_n^5 + \ldots$$

(24)

Expanding $f(z_n)$ about $\alpha$ and using (24) we have

$$f(z_n) = f'(\alpha)[-c_2^2(\beta - 1)e_n^4 + 4c_2^2(c_2^2 - c_3)(\beta - 1)e_n^5 + \ldots]$$

(25)
\[ H_3(x_n, y_n, z_n) = f'(\alpha)[1 + (-2\beta c_2^4 + 2c_2^4 + c_2c_4)e_n^4 + (8(\beta - 1)c_2^5 - 8c_3c_2^3(\beta - 1) - 2c_2^5 c_4 + 2c_2c_5 + 2c_3c_4)e_n^5 + \ldots] \]  

Dividing equation (25) by (26), we get

\[ \frac{f(x_n)}{H_3(x_n, y_n, z_n)} = f'(\alpha)[-c_2^3(\beta - 1)e_n^4 + 4c_2^2(c_2^2 - c_3)(\beta - 1)e_n^5 + \ldots] \]  

From equations (18), (20), (21), (23) and (27) we obtain

\[ x_{n+1} = \alpha + c_2^4(\beta - 1)(c_2^3(\beta - 1) - c_4)e_n^8 - 8c_2^3(\beta - 1)(\beta - 1)c_2^5 - c_3c_2^3(\beta - 1) - c_2^2 c_4 + \left(\frac{3}{4}\right) c_2 c_5 + \left(\frac{3}{4}\right) c_3 c_4 \right) e_n^9 + 2c_2^4 \left(\beta^3 + 15\beta^2 - 34\beta + 18\right)c_2^7 - \left(\frac{2}{3}\right) c_3 \left(\beta^2 + \left(\frac{43}{2}\right) c_3 c_4 + c_6 \right) e_n^{10} + O(e_n^{11}) \]  

Implying that

\[ e_{n+1} = c_2^4(\beta - 1)(c_2^3(\beta - 1) - c_4)e_n^8 + \cdots + O(e_n^{11}) \]  

When \( \beta = 1 \) we have

\[ e_{n+1} = c_2^4 \left(c_4 c_2^4 - \frac{1}{2} c_3 c_4 c_2^2 \right) e_n^{10} + O(e_n^{11}) \]  

Hence, Algorithm 3.7 has at least tenth-order convergence.

**Theorem 3.2:** Let \( \alpha \in D \), be a simple root of a sufficiently differentiable function \( f: D \subset \mathbb{R} \to \mathbb{R} \) for an open interval \( D \). If \( x_0 \) is sufficiently close to \( \alpha \), then the Algorithm 3.8 has at least eighth order of convergence for any \( \beta \in \mathbb{R} \).

**Proof:** With the same assumptions of the previous theorem, we have

\[ H_2(x_n, y_n) = 2c_2 + 4c_3 e_n + (2c_2c_3 + 6c_4)e_n^2 + \ldots \]  

\[ z_n = \alpha - (c_2^2(\beta - 1) + c_3)c_2e_n^4 + ((4\beta - 4)c_2^4 + (-6\beta + 8)c_2^2c_3 - 2c_2c_4 - 2c_3^2)e_n^5 + \ldots \]
Expanding \( f(z_n) \) about \( \alpha \) and using (29) we have

\[
f(z_n) = f'(\alpha)[-c_2^2(\beta - 1) + c_3)c_2 e_n^4 + ((4\beta - 4)c_2^4 + (-6\beta + 8)c_3^2 c_3 - 2c_2 c_4 - 2c_3^2)e_n^5 + ...]
\]  

(33)

\[
H_3(x_n, y_n, z_n) = f'(\alpha)\left[1 - 2c_2((\beta - 1)c_2^3 + c_2 c_3 - \frac{1}{2}c_4)e_n^4 + (8(\beta - 1)c_2^5 + (-12\beta + 16)c_3^2 c_2^2 - 2c_2 c_4 - 2c_3^2)e_n^5 + ...ight]
\]  

(34)

Dividing equation (30) by (31), we get

\[
\frac{f(x_n)}{H_3(x_n, y_n, z_n)} = f'(\alpha)[-c_2((\beta - 1)c_2^2 + c_3)e_n^4 + (4(\beta - 1)c_2^4 + (-6\beta + 8)c_3^2 c_2^2 - 2c_2 c_4 - 2c_3^2)e_n^5 + ...]
\]  

(35)

Substituting equations (18), (20), (22), (31) and (35) in Algorithm 3.8 we get

\[
x_{n+1} = \alpha + ((\beta - 1)c_2^2 + c_3)((\beta - 1)c_2^3 + c_2 c_3 - c_4)c_2^2 e_n^8 + O(e_n^9)
\]  

(36)

Implying that

\[
e_{n+1} = ((\beta - 1)c_2^2 + c_3)((\beta - 1)c_2^3 + c_2 c_3 - c_4)c_2^2 e_n^8 + O(e_n^9)
\]  

(37)

When \( \beta = 1 \) we have

\[
e_{n+1} = (c_2 c_3 - c_4)c_3 c_2^2 e_n^8 + O(e_n^9)
\]  

(38)

Hence, Algorithm 3.8 has at least eighth-order convergence.

5. Numerical Examples

In this section, we apply new three-step iterative methods that defined in Algorithms (3.7) and (3.8), to solve several nonlinear equations and make the comparison of newly established iteration methods with classical Newton’s method [1], S. Weerakoon et al method [2], and with some existing optimal eighth order methods. For example, R. Thukral method [29], L. Liu et al. method [9], X. Wang et al. methods [11], A. Cordero et al. method [12] and one of the methods by A.A. Al-Harbi [17]. The methods are given as follows:
Newton’s method (NM):

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

S. Weerakoon et al. method (WFM):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

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

R. Thukral method (TM):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

\[ z_n = x_n - \frac{(f(x_n))^2 + (f(y_n))^2}{(f(x_n) - f(y_n))f'(x_n)}. \]

\[ x_{n+1} = z_n - \frac{f(x_n)}{f'(x_n)} \left[ \left( \frac{1 + \mu_i^2}{1 - \mu_i} \right)^2 - 2(\mu_i)^2 - 6(\mu_i)^3 + \frac{f(x_n)}{f(y_n)} + \frac{4f(z_n)}{f(x_n)} \right], \quad n = 0, 1, \ldots \]

where \( \mu_i = \frac{f(y_n)}{f'(x_n)}. \)

L. Liu et al. method (LWM):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

\[ z_n = y_n - \frac{f(x_n)}{f(x_n) - 2f(y_n)} \frac{f(y_n)}{f'(x_n)}, \]

\[ x_{n+1} = z_n - \frac{f(x_n)}{f'(x_n)} \left[ \left( \frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)} \right)^2 + \frac{f(x_n)}{f(y_n) - \mu f(z_n)} + \frac{4f(z_n)}{f(x_n) + \beta f(z_n)} \right], \quad n = 0, 1, \ldots \]

where \( \beta = \mu = 1. \)

X. Wang et al. methods (BM8 and BM8-2):

(BM8):
\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \]

\[ z_n = y_n - \frac{f(y_n)}{f'(y_n)}. \]

\[ x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}: \quad n = 0, 1, \ldots \]

(BM8-2):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \]

\[ z_n = y_n - \frac{f(y_n)}{2f(x_n)} \cdot \frac{f'(y_n)}{2f(x_n)} = \frac{f(y_n)}{2f(x_n)}. \]

\[ x_{n+1} = z_n - \frac{f(z_n)}{2f(x_n) + f(y_n) - 2f(x_n) + f(y_n)}: \quad n = 0, 1, \ldots \]

A. Cordero et al. method (CTM):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \]

\[ z_n = x_n - \frac{f(x_n) - f(y_n)}{f(x_n) - 2f(y_n)} f(x_n). \]

\[ x_{n+1} = v_n - \frac{f(z_n)}{f'(z_n)} \cdot \frac{3(\beta_2 + \beta_3)(v_n - z_n)}{\beta_1 (v_n - z_n) + \beta_2 (y_n - x_n) + \beta_3 (z_n - x_n)}: \quad n = 0, 1, \ldots \]

where \( \beta_1 = 0, \beta_2 = 1 \) and \( \beta_3 = 0 \) and

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

A. A. Al-Harbi et al. method (ASM):

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}. \]

\[ z_n = x_n - \frac{f(y_n) - f(x_n)}{2f(y_n) - f(x_n)} f(x_n). \]

\[ x_{n+1} = z_n - \left\{ (1 + t_n^2 + 2t_n^3 + \mu t_n^4) + (-1 + \beta t_2) + (1 + 2t_3 + \gamma t_3^2) \right\} \frac{f(z_n)}{f(y_n, z_n)}. \]
\[ n = 0, 1, ... \]

where \( \mu = 1, \beta = 0 \) and \( \gamma = -2 \) and \( t_1 = \frac{f(y_n)}{f'(x_n)}, t_2 = \frac{f(z_n)}{f'(y_n)} \) and \( t_3 = \frac{f(z_n)}{f'(x_n)} \).

For writing programs, we use Maple 2016 program with 1000-digit floating point arithmetic (Digits: = 1000). We use the stopping criteria \( |x_{n+1} - x_n| < \varepsilon \) and \( |f(x_{n+1})| < \varepsilon \), where \( \varepsilon = 10^{-15} \), for computer programs. Different test functions and their approximate root \( \alpha \) found up to the 28th decimal places are given in Table 1. Table 2 shows a comparison between the various iterative methods depending on the number of iterations (IT), the values of \( |x_{n+1} - x_n| \) and \( |f(x_{n+1})| \) and computational order of convergence (COC). Figures (2.1- 2.8) show the graphical analysis for the uphold of numerical results.

**Table 1:** The test functions and their root \( \alpha \)

<table>
<thead>
<tr>
<th>( f(x) )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ( f_1(x) = x^3 + 4x^2 - 15 )</td>
<td>1.6319808055660635175221064455</td>
</tr>
<tr>
<td>2 ( f_2(x) = (x - 1)^3 - 1 )</td>
<td>2.0000000000000000000000000000</td>
</tr>
<tr>
<td>3 ( f_3(x) = x^3 - e^{-x} )</td>
<td>0.7728829591492101128487486048</td>
</tr>
<tr>
<td>4 ( f_4(x) = (1 + \cos x)(e^x - 2) )</td>
<td>0.6931471805599453095377829940</td>
</tr>
<tr>
<td>5 ( f_5(x) = \sin x - \frac{x}{2} )</td>
<td>1.8954942670339809471440357380</td>
</tr>
<tr>
<td>6 ( f_6(x) = \sin^{-1}(x^2 - 1) - \frac{x}{2} + 1 )</td>
<td>0.5948109683983691775226562351</td>
</tr>
<tr>
<td>7 ( f_7(x) = (\sin x)^2 - x^2 + 1 )</td>
<td>2.0000019101432763850749104202</td>
</tr>
</tbody>
</table>
Table 2. Comparison of our iterative methods with Newton’s method and relevant various methods

<p>| $f(x)$ | $x_0$ | Methods | IT | $|x_{n+1} - x_n|$ | $|f(x_{n+1})|$ | COC |
|--------|------|---------|----|----------------|----------------|-----|
| $f_1$  | 1.9  | NM      | 5  | 4.7627717e-16  | 2.0179551e-30  | 2.0000070 |
|        |      | WFM     | 4  | 3.5538378e-26  | 1.9121725e-76  | 2.9997779 |
|        |      | TM      | 3  | 7.11868510e-22 | 6.7158442e-106 | 4.8688477 |
|        |      | LWM     | 3  | 1.2777913e-56  | 2.2612305e-448 | 7.9411776 |
|        |      | BM8     | 3  | 3.8860695e-63  | 2.6388788e-501 | 7.9592052 |
|        |      | BM8-2   | 3  | 2.1273196e-65  | 1.1467258e-519 | 7.9633384 |
|        |      | CTM     | 3  | 7.8460648e-62  | 1.0680800e-490 | 7.9548448 |
|        |      | ASM     | 3  | 7.71083310e-59 | 2.2975939e-466 | 7.9419739 |
|        |      | NBM     | 3  | 4.8981129e-143 | 0.0             | 11.9628371 |
|        |      | NBM1    | 3  | 1.4736548e-72  | 7.9792114e-578 | 7.9792114 |
| $f_2$  | 1.8  | NM      | 6  | 3.0908727e-21  | 2.8660481e-41  | 2.0000004 |
|        |      | WFM     | 4  | 4.8567907e-17  | 4.0097399e-49  | 3.0037564 |
|        |      | TM      | 4  | 1.2301362e-47  | 4.5069838e-234 | 5.0041787 |
|        |      | LWM     | 3  | 4.8376766e-34  | 4.5997183e-266 | 8.1771117 |
|        |      | BM8     | 3  | 9.5534938e-42  | 2.0817044e-328 | 8.1048138 |
|        |      | BM8-2   | 3  | 2.5437487e-45  | 2.3373966e-357 | 8.08560810 |
|        |      | CTM     | 3  | 2.8011765e-40  | 1.3478179e-316 | 8.1326889 |
|        |      | ASM     | 3  | 2.0337759e-35  | 2.7318796e-277 | 8.1892898 |
|        |      | NBM     | 3  | 1.3599031e-95  | 0.0             | 12.0908695 |
|        |      | NBM1    | 3  | 4.4756985e-53  | 5.3674172e-420 | 8.0212447 |
| $f_3$  | 3.5  | NM      | 9  | 6.5470578e-20  | 8.9491765e-39  | 2.0000016 |
|        |      | WFM     | 6  | 1.9993598e-16  | 1.9761660e-47  | 2.9966706 |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TM</td>
<td>5</td>
<td>5.6254001e-47</td>
<td>4.5689177e-231</td>
<td>4.99626010</td>
<td></td>
</tr>
<tr>
<td>LWM</td>
<td>4</td>
<td>4.4125419e-29</td>
<td>3.9927632e-227</td>
<td>7.7982095</td>
<td></td>
</tr>
<tr>
<td>BM8</td>
<td>4</td>
<td>3.97083810e-39</td>
<td>8.1562120e-308</td>
<td>7.8800229</td>
<td></td>
</tr>
<tr>
<td>BM8-2</td>
<td>4</td>
<td>7.1398043e-47</td>
<td>1.7077867e-370</td>
<td>7.9632156</td>
<td></td>
</tr>
<tr>
<td>CTM</td>
<td>4</td>
<td>7.1286254e-40</td>
<td>3.7426175e-314</td>
<td>7.9329018</td>
<td></td>
</tr>
<tr>
<td>ASM</td>
<td>4</td>
<td>1.4220098e-33</td>
<td>3.6783273e-263</td>
<td>7.8247620</td>
<td></td>
</tr>
<tr>
<td>NBM</td>
<td>4</td>
<td>3.7317121e-125</td>
<td>0.0</td>
<td>9.8812164</td>
<td></td>
</tr>
<tr>
<td>NBM1</td>
<td>4</td>
<td>1.8366324e-95</td>
<td>5.3992020e-759</td>
<td>7.9967565</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_4$</td>
<td>0.9</td>
<td>NM</td>
<td>5</td>
<td>1.9230803e-29</td>
<td>1.8170046e-58</td>
</tr>
<tr>
<td>WFM</td>
<td>4</td>
<td>1.8175039e-31</td>
<td>2.0473423e-93</td>
<td>2.9999130</td>
<td></td>
</tr>
<tr>
<td>TM</td>
<td>3</td>
<td>1.8803857e-36</td>
<td>9.27421034e-181</td>
<td>4.7726412</td>
<td></td>
</tr>
<tr>
<td>LWM</td>
<td>3</td>
<td>5.4674051e-82</td>
<td>5.2221645e-653</td>
<td>7.8238898</td>
<td></td>
</tr>
<tr>
<td>BM8</td>
<td>3</td>
<td>3.6612034e-118</td>
<td>1.13663377e-945</td>
<td>7.8148441</td>
<td></td>
</tr>
<tr>
<td>BM8-2</td>
<td>3</td>
<td>5.0785010e-89</td>
<td>2.4383002e-710</td>
<td>7.8705503</td>
<td></td>
</tr>
<tr>
<td>CTM</td>
<td>3</td>
<td>2.4828066e-72</td>
<td>1.1379207e-575</td>
<td>7.9379761</td>
<td></td>
</tr>
<tr>
<td>ASM</td>
<td>3</td>
<td>6.8395375e-89</td>
<td>6.3437628e-709</td>
<td>7.8293396</td>
<td></td>
</tr>
<tr>
<td>NBM</td>
<td>3</td>
<td>1.2206835e-145</td>
<td>0.0</td>
<td>9.8772847</td>
<td></td>
</tr>
<tr>
<td>NBM1</td>
<td>3</td>
<td>5.1313716e-89</td>
<td>2.6481682e-710</td>
<td>7.8706139</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_5$</td>
<td>2.0</td>
<td>NM</td>
<td>5</td>
<td>1.7664965e-20</td>
<td>1.4787271e-40</td>
</tr>
<tr>
<td>WFM</td>
<td>4</td>
<td>6.9218142e-35</td>
<td>8.2109603e-104</td>
<td>2.9999649</td>
<td></td>
</tr>
<tr>
<td>TM</td>
<td>3</td>
<td>1.1448151e-27</td>
<td>1.1529071e-135</td>
<td>4.9292808</td>
<td></td>
</tr>
<tr>
<td>LWM</td>
<td>3</td>
<td>1.0543348e-69</td>
<td>4.6254912e-553</td>
<td>7.9618721</td>
<td></td>
</tr>
<tr>
<td>BM8</td>
<td>3</td>
<td>1.8860406e-80</td>
<td>2.8462341e-640</td>
<td>7.9718217</td>
<td></td>
</tr>
<tr>
<td>BM8-2</td>
<td>3</td>
<td>7.4215447e-80</td>
<td>1.8459664e-635</td>
<td>7.9733717</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CTM</td>
<td>3</td>
<td>1.0683158e-75</td>
<td>1.0583247e-601</td>
<td>7.9677030</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>3</td>
<td>8.5875488e-73</td>
<td>4.0972561e-578</td>
<td>7.9630357</td>
</tr>
<tr>
<td></td>
<td>NBM</td>
<td>3</td>
<td>5.6305868e-131</td>
<td>4.0e-1000</td>
<td>9.9793032</td>
</tr>
<tr>
<td></td>
<td>NBM1</td>
<td>3</td>
<td>1.5961638e-100</td>
<td>7.9898120e-803</td>
<td>7.9512710</td>
</tr>
<tr>
<td>(f_6)</td>
<td>0.3</td>
<td>NM</td>
<td>5</td>
<td>2.3916069e-19</td>
<td>1.6107744e-38</td>
</tr>
<tr>
<td></td>
<td>WFM</td>
<td>4</td>
<td>1.5267255e-25</td>
<td>7.2841117e-76</td>
<td>2.9997002</td>
</tr>
<tr>
<td></td>
<td>TM</td>
<td>3</td>
<td>7.0798766e-27</td>
<td>8.7741676e-134</td>
<td>5.05203810</td>
</tr>
<tr>
<td></td>
<td>LWM</td>
<td>3</td>
<td>1.3240055e-59</td>
<td>5.6416104e-474</td>
<td>7.9104307</td>
</tr>
<tr>
<td></td>
<td>BM8</td>
<td>3</td>
<td>6.1558807e-77</td>
<td>2.0560733e-614</td>
<td>7.9112957</td>
</tr>
<tr>
<td></td>
<td>BM8-2</td>
<td>3</td>
<td>6.6289185e-66</td>
<td>5.7572363e-525</td>
<td>7.9018207</td>
</tr>
<tr>
<td></td>
<td>CTM</td>
<td>3</td>
<td>5.3278165e-61</td>
<td>1.9435254e-485</td>
<td>7.9366747</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>3</td>
<td>5.0701651e-72</td>
<td>9.5018919e-575</td>
<td>7.9343293</td>
</tr>
<tr>
<td></td>
<td>NBM</td>
<td>3</td>
<td>3.5116305e-108</td>
<td>3.0e-1000</td>
<td>9.9351292</td>
</tr>
<tr>
<td></td>
<td>NBM1</td>
<td>3</td>
<td>1.1274703e-65</td>
<td>4.7650905e-523</td>
<td>7.8942633</td>
</tr>
<tr>
<td>(f_7)</td>
<td>1.5</td>
<td>NM</td>
<td>5</td>
<td>4.4131593e-19</td>
<td>3.7884542e-37</td>
</tr>
<tr>
<td></td>
<td>WFM</td>
<td>4</td>
<td>1.8211104e-31</td>
<td>9.8628863e-93</td>
<td>2.9999346</td>
</tr>
<tr>
<td></td>
<td>TM</td>
<td>3</td>
<td>9.9397361e-26</td>
<td>5.1884889e-125</td>
<td>4.9257651</td>
</tr>
<tr>
<td></td>
<td>LWM</td>
<td>3</td>
<td>2.3530759e-66</td>
<td>3.5837479e-525</td>
<td>7.9648201</td>
</tr>
<tr>
<td></td>
<td>BM8</td>
<td>3</td>
<td>1.8248762e-74</td>
<td>5.5373256e-591</td>
<td>7.9724181</td>
</tr>
<tr>
<td></td>
<td>BM8-2</td>
<td>3</td>
<td>6.3807720e-77</td>
<td>6.2926145e-611</td>
<td>7.9760834</td>
</tr>
<tr>
<td></td>
<td>CTM</td>
<td>3</td>
<td>3.8163135e-72</td>
<td>3.6529135e-572</td>
<td>7.9717647</td>
</tr>
<tr>
<td></td>
<td>ASM</td>
<td>3</td>
<td>5.14822910e-69</td>
<td>9.8364499e-547</td>
<td>7.9641333</td>
</tr>
<tr>
<td></td>
<td>NBM</td>
<td>3</td>
<td>6.4745225e-122</td>
<td>0.0</td>
<td>9.9818428</td>
</tr>
<tr>
<td></td>
<td>NBM1</td>
<td>3</td>
<td>2.2780807e-86</td>
<td>1.8964177e-687</td>
<td>7.9659951</td>
</tr>
</tbody>
</table>
Figure 1: Log of residuals of problem 1

Figure 2: Log of residuals of problem 2
Figure 3: Log of residuals of problem 3

Figure 4: Log of residuals of problem 4
Figure 5: Log of residuals of problem 5

Figure 6: Log of residuals of problem 6
Figure 7: Log of residuals of problem 7

Figure 8: Comparison between methods and efficiency indices
5. Conclusions

In this research, new predictor-corrector iterative methods have been proposed for solving nonlinear equations denoted by (NBM) and (NBM1), respectively. Our new iterative methods have the advantage of evaluating only the first derivative of $f(x)$. Numerical results that we got show the convergence order of (NBM) and (NBM1) methods is ten and eight respectively, which is higher than many existing methods. Also, the number of iterations of (NBM) and (NBM1) methods is better than the classical Newton's method, S. Weerakoon et al. method and equal with other existing methods. The efficiency index of both new iterative methods is much better from the classical Newton's method, S. Weerakoon et al. method and (BM8) method and the efficiency index of (NBM1) method is equal with all the existing methods but, the efficiency index of (NBM) method is less. However, the drawback in the efficiency index of (NBM) method is compensated by increase in accuracy. Moreover, the proposed (NBM) and (NBM1) methods have large computational order of convergence (COC) than all the existing methods which sign that our newly proposed iterative methods are well-matched to inspect the roots.

References


طرائق تكرارية كفؤة من ثلاث خطوات تعتمد على صيغة برنشتاين التربيعية لحل المعادلات غير الخطية

هدى جبار سعيد
نوري ياسر عبد الحسن
قسم الرياضيات، كلية التربية للعلوم الصرفة، جامعة البصرة، البصرة، العراق

المتخلص
في هذه الدراسة، نقترح ونحلل عائلتين جديدتين من الطرائق التكرارية الكفؤة مع معلم معتمد على نظرية نيوتن في حساب التفاضل والتكامل وصيغة برنشتاين التربيعية وأساسيات متعددة حدد برنولي وتايلور وتايلور وبعض التقنيات العددية. ثبت أن الطرائق التكرارية الجديدة تصل إلى رتبة تقارب عشرة مع ستة دوال لكل تكرار وثمانية مع أربع دوال لكل تكرار مما يعني أن مؤشر كفاءة الطرائق التكرارية الجديدة هو 1.4678 و1.6818 على التوالي. تقديم أمثلة عددية لإظهار كفاءة وأداء طرائقنا التكرارية، مقارنة بطريقة نيوتن وغيرها من الطرائق ذات الصلة.