Research Article

Another Simple Way of Deriving Several Iterative Functions to Solve Nonlinear Equations

Ramandeep Behl, V. Kanwar, and Kapil K. Sharma

1 Department of Mathematics, Panjab University, Chandigarh 160 014, India
2 University Institute of Engineering and Technology, Panjab University, Chandigarh 160 014, India

Correspondence should be addressed to V. Kanwar, vmithil@yahoo.co.in

Received 25 June 2012; Revised 20 September 2012; Accepted 4 October 2012

Academic Editor: Alicia Cordero

Copyright © 2012 Ramandeep Behl et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present another simple way of deriving several iterative methods for solving nonlinear equations numerically. The presented approach of deriving these methods is based on exponentially fitted osculating straight line. These methods are modifications of Newton’s method. Also, we obtain well-known methods as special cases, for example, Halley’s method, super-Halley method, Ostrowski’s square-root method, Chebyshev’s method, and so forth. Furthermore, a simple linear combination of two third-order multipoint iterative methods is used for designing new optimal methods of order four.

1. Introduction

Various problems arising in mathematical and engineering science can be formulated in terms of nonlinear equation of the form

\[ f(x) = 0. \quad (1.1) \]

To solve (1.1), we can use iterative methods such as Newton’s method [1–19] and its variants, namely, Halley’s method [1–3, 5, 6, 8, 9], Euler’s method (irrational Halley’s method) [1, 3], Chebyshev’s method [1, 2], super-Halley method [2, 4] as Ostrowski’s square-root method [5, 6], and so forth, available in the literature.

Among these iterative methods, Newton’s method is probably the best known and most widely used algorithm for solving such problems. It converges quadratically to a simple
root and linearly to a multiple root. Its geometric construction consists in considering the straight line

\[ y = ax + b \]  

(1.2)

and then determining the unknowns \( a \) and \( b \) by imposing the tangency conditions

\[ y(x_n) = f(x_n), \quad y'(x_n) = f'(x_n), \]  

(1.3)

thereby obtaining the tangent line

\[ y(x) = f(x_n) + f'(x_n)(x - x_n), \]  

(1.4)

to the graph of \( f(x) \) at \((x_n, f(x_n))\).

The point of intersection of this tangent line with \( x \)-axis gives the celebrated Newton’s method

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

(1.5)

Newton’s method for multiple roots appears in the work of Schröder [19], which is given as

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

(1.6)

This method has a second-order convergence, including the case of multiple roots. It may be obtained by applying Newton’s method to the function \( u_f(x) = f(x)/f'(x) \), which has simple roots in each multiple root of \( f(x) \). The well-known third-order methods which entail the evaluation of \( f''(x) \) are close relatives of Newton’s method and can be obtained by admitting geometric derivation [1, 2, 5] from the different quadratic curves, for example, parabola, hyperbola, circle or ellipse, and so forth.

The purpose of the present work is to provide some alternative derivations to the existing third-order methods through an exponentially fitted osculating straight line. Some other new formulas are also presented. Here, we will make use of symbolic computation in the programming package Mathematica 7 to derive the error equations of various iterative methods.

Before starting the development of iterative scheme, we would like to introduce some basic definitions.

**Definition 1.1.** A sequence of iterations \( \{x_n \mid n \geq 0\} \) are said to converge with order \( P \geq 1 \) to a point \( r \) if

\[ |r - x_{n+1}| \leq C|r - x_n|^P, \quad n \geq 0, \]  

(1.7)

for some \( C > 0 \). If \( P = 1 \), the sequence is said to converge linearly to \( r \). In that case, we require \( C < 1 \); the constant \( C \) is called the rate of linear convergence of \( x_n \) to \( r \).
Definition 1.2. Let $e_n = x_n - r$ be the error in the $n$th iteration; one calls the relation
\[ e_{n+1} = Ce_n + O\left(e_n^{r+1}\right), \] (1.8)
as the error equation. If we can obtain the error equation for any iterative method, then the value of $P$ is its order of convergence.

Definition 1.3. Let $d$ be the number of new pieces of information required by a method. A piece of information typically is any evaluation of a function or one of its derivatives. The efficiency of the method is measured by the concept of efficiency index [6] and is defined by
\[ E = P^{1/d}, \] (1.9)
where $P$ is the order of the method.

2. Development of the Methods

Two equivalent derivations for the third-order iterative methods through exponentially fitted osculating straight line are presented below.

Case I. Consider an exponentially fitted osculating straight line in the following form:
\[ y(x) = e^{p(x-x_n)} A(x-x_n) + B, \] (2.1)
where $p \in \mathbb{R}$, $|p| < \infty$, and $A$ and $B$ are arbitrary constants. These constants will be determined by using the tangency conditions at the point $x = x_n$.

If an exponentially fitted osculating straight line given by (2.1) is tangent to the graph of the equation in question, that is, $f(x) = 0$ at $x = x_n$, then we have
\[ y^{(k)}(x_n) = f^{(k)}(x_n), \quad k = 0, 1, 2. \] (2.2)
Therefore, we obtain
\[ A = f'(x_n) - pf(x_n), \quad B = f(x_n) \] (2.3)
and a quadratic equation in $p$ as follows:
\[ p^2 f(x_n) - 2pf'(x_n) + f''(x_n) = 0. \] (2.4)
Suppose that the straight line (2.1) meets the $x$-axis at $x = x_{n+1}$, then
\[ y(x_{n+1}) = 0, \] (2.5)
and it follows from (2.1) that

\[ A(x_{n+1} - x_n) + B = 0. \]  

(2.6)

From (2.6) and if \( A \neq 0 \), we get

\[ x_{n+1} = x_n - \frac{B}{A}, \]  

(2.7)

or

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

(2.8)

This is the well-known one-parameter family of Newton’s method [2]. This family converges quadratically under the condition \( f'(x_n) - pf(x_n) \neq 0 \), while \( f'(x_n) = 0 \) is permitted in some points. For \( p = 0 \), we obtain Newton’s method. The error equation of Scheme (2.8) is given by

\[ e_{n+1} = (p - c_2)e_n^2 + O(e_n^4), \]  

(2.9)

where \( e_n = x_n - r, \ c_k = (1/k!)(f^{(k)}(r)/f'(r)), \ k = 2, 3, \ldots, \) and \( x = r \) is the root of nonlinear (1.1). In order to obtain the quadratic convergence, the entity in the denominator should be the largest in magnitude. It is straightforward to see from the above error equation (2.9) that for \( p = c_2 = (f'(x_n)/2f(x_n)) \), we obtain the well-known third-order Halley’s method.

If we apply the well-known Newton’s method (1.5) to the modified function \( u_f(x) = f(x_n)/[f'(x_n) - pf(x_n)] \), we get another iterative method as

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

(2.10)

This is a new one-parameter modified family of Schröder’s method [19] for an equation having multiple roots of multiplicity \( m > 1 \) unknown. It is interesting to note that by ignoring the term \( p \), method (2.10) reduces to Schröder’s method. It is easy to verify that this method is also an order two method, including the case of multiple zeros. Theorem 2.1 indicates that what choice on the disposable parameter \( p \) in family (2.10), the order of convergence will reach at least the second and third order, respectively.

**Theorem 2.1.** Let \( f : I \subseteq \mathbb{R} \to \mathbb{R} \) have at least three continuous derivatives defined on an open interval \( I \), enclosing a simple zero of \( f(x) \) (say \( x = r \in I \)). Assume that initial guess \( x = x_0 \) is sufficiently close to \( r \) and \( f'(x_n) - pf(x_n) \neq 0 \) in \( I \). Then an iteration scheme defined by formula (2.10) has at least a second order convergence and will have a third-order convergence when \( p = c_2 \). It satisfies the following error equation

\[ e_{n+1} = (p - c_2)e_n^2 + 2(c_2^2 - 4c_2)e_n^4 + O(e_n^6). \]  

(2.11)

where \( p \in \mathbb{R} \) is a free disposable parameter, \( e_n = x_n - r, \ and \ c_k = (1/k!)(f^{(k)}(r)/f'(r)), \ k = 2, 3, \ldots \)
Proof. Let \( x = r \) be a simple zero of \( f(x) \). Expanding \( f(x_n) \) and \( f'(x_n) \) about \( x = r \) by the Taylor’s series expansion, we have

\[
f(x_n) = f'(r)
\left( e_n + c_2 e_n^2 + c_3 e_n^3 \right) + O\left( e_n^4 \right),
\]
\[
f'(x_n) = f'(r)
\left( 1 + 2c_2 e_n + 3c_3 e_n^2 \right) + O\left( e_n^3 \right).
\]

(2.12)

respectively.

Furthermore, we have

\[
f''(x_n) = f'(r)
\left( 2c_2 + 6c_3 e_n \right) + O\left( e_n^2 \right).
\]

(2.13)

From (2.12) and (2.13), we have

\[
\frac{f(x_n)(f'(x_n) - pf(x_n))}{f^2(x_n) - f(x_n)f''(x_n)} = e_n - (p - c_2)e_n^2 - \left( 2c_2^2 - 4c_3 \right)e_n^3 + O\left( e_n^4 \right).
\]

(2.14)

Finally, the using above equation (2.14) in our proposed scheme (2.10), we get

\[
e_{n+1} = e_n - \frac{f(x_n)(f'(x_n) - pf(x_n))}{f^2(x_n) - f(x_n)f''(x_n)}
= e_n - \left( e_n - (p - c_2)e_n^2 - \left( 2c_2^2 - 4c_3 \right)e_n^3 \right) + O\left( e_n^4 \right)
= (p - c_2)e_n^2 + \left( 2c_2^2 - 4c_3 \right)e_n^3 + O\left( e_n^4 \right).
\]

(2.15)

This reveals that the one-point family of methods (2.10) reaches at least second order of convergence by using only three functional evaluations (i.e., \( f(x_n), f'(x_n), \) and \( f''(x_n) \)) per full iteration. It is straightforward to see that family (2.10) reaches the third order of convergence when \( p = c_2 \). This completes the proof of Theorem 2.1. □

Now we wish to construct different third-order iterative methods, which are dependent on the different values of \( p \) and are given as below.

Special Cases

(i) When \( |p| \ll 1 \), then \( p^2 \) can be neglected in (2.4), and we get

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

(2.16)

(a) Inserting this value of \( p \) in (2.8), we get

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

(2.17)
This is the well-known third-order Halley’s method [1–3, 5, 6, 8, 9]. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{2}{c_2} - c_3 \right) e_n^3 + O\left( e_n^4 \right) \]  

(2.18)

(b) Inserting this value of \( p \) in (2.10), we get

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

(2.19)

This is the well-known third-order super-Halley method [1, 2, 4]. It satisfies the following error equation:

\[ e_{n+1} = -c_2 e_n^4 + O\left( e_n^5 \right). \]  

(2.20)

(ii) When we take \( p \) as an implicit function, then the values of \( p \) are based on the idea of successive approximations. Therefore, we get another value of \( p \) from (2.4) as

\[ p = \frac{f''(x_n)}{2f'(x_n) - pf(x_n)}. \]  

(2.21)

Here, the function \( f''(x_n)/(2f'(x_n) - pf(x_n)) \) which occurs on the right cannot be computed till \( p \) is known. To get round off the difficulty, we substitute the value of \( p = f''(x_n)/2f'(x_n) \) from a previously obtained value in (2.16). Therefore, the new modified value of \( p \) is

\[ p = \frac{2f'(x_n)f''(x_n)}{4f^2(x_n) - pf(x_n)f'(x_n)}. \]  

(2.22)

(a) Inserting this value of \( p \) in (2.8), we get

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{4f^2(x_n) - f(x_n)f'(x_n)}{4f^2(x_n) - 3f(x_n)f'(x_n)} \right] \]  

(2.23)

This is a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{c_7 - 2c_5}{2} \right) e_n^3 + O\left( e_n^4 \right). \]  

(2.24)

(b) Again inserting this value of \( p \) in (2.10), we get

\[ x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{4f^2(x_n) - 3f(x_n)f'(x_n)} \left[ \frac{4f^2(x_n) - f(x_n)f''(x_n)}{4f^2(x_n) - f(x_n)f''(x_n)} \right] \]  

(2.25)
This is again a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{c_2^2 - 2c_1c_3}{2} \right) e_n^3 + O(e_n^4). \]  

(iii) Similarly, from (2.4), one can get another value of \( p \) as

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

Again inserting the previously obtained value of \( p = f'(x_n)/2f'(x_n) \) in the right-hand side of (2.16), we get another modified value of \( p \) as

\[ p = \frac{f'(x_n) \left( 4f^2(x_n) + f(x_n)f'(x_n) \right)}{8f^3(x_n)}. \]

(a) Inserting this value of \( p \) in (2.8), we get

\[ x_{n+1} = x_n - \frac{8f(x_n)f^3(x_n)}{8f^3(x_n) - \left( 4f^2(x_n) + f(x_n)f'(x_n) \right) \left( f(x_n)f''(x_n) \right)}. \]

This is a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{c_2^2 - 2c_1c_3}{2} \right) e_n^3 + O(e_n^4). \]  

(b) Again inserting this value of \( p \) in (2.10), we obtain

\[ x_{n+1} = x_n - \frac{8f^4(x_n) - f(x_n)f''(x_n) \left( 4f^2(x_n) + f(x_n)f'(x_n) \right)}{8f^2(x_n) \left( f^2(x_n) - f(x_n)f''(x_n) \right)}. \]

This is again a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{c_2^2 - 2c_1c_3}{2} \right) e_n^3 + O(e_n^4). \]
(iv) Now we solve the quadratic equation (2.4) for general values of \( p \). Hence, we get

\[
P = \frac{f'(x_n) \pm \sqrt{f'^2(x_n) - f(x_n)f''(x_n)}}{f(x_n)}.
\]  

(2.33)

(a) Inserting these values of \( p \) either in (2.4) or (2.10), we get the well-known third-order Ostrowski’s square-root method [5] as

\[
x_{n+1} = x_n \pm \frac{f(x_n)}{\sqrt{f'^2(x_n) - f(x_n)f''(x_n)}}.
\]  

(2.34)

It satisfies the following error equation:

\[
e_{n+1} = \left( \frac{c_3 - 2c_2}{2} \right) e_n^3 + O\left(e_n^4\right).
\]  

(2.35)

(v) When we rationalize the numerator of (2.33), we get other values of \( p \) as

\[
P = \frac{f''(x_n)}{f'(x_n) \pm \sqrt{f'^2(x_n) - f(x_n)f''(x_n)}}.
\]  

(2.36)

Inserting these values of \( p \) either in (2.8) or (2.10), we get

\[
x_{n+1} = x_n \pm \frac{f(x_n)\sqrt{f'^2(x_n) - f(x_n)f''(x_n)} - f(x_n)f''(x_n)}{f'^2(x_n) \pm f'(x_n)\sqrt{f'^2(x_n) - f(x_n)f''(x_n)} - f(x_n)f''(x_n)}.
\]  

(2.37)

This is a new third-order iterative method. It satisfies the following error equation:

\[
e_{n+1} = \left( \frac{c_3 - 2c_2}{2} \right) e_n^3 + O\left(e_n^4\right).
\]  

(2.38)

Case II. In the second case, we have now considered an exponentially fitted osculating straight line in the following form:

\[
y(x) = e^{\theta_{y(x)}}[A_1(x - x_n) + A_2],
\]  

(2.39)
where $p \in \mathbb{R}$, $|p| < \infty$, and $A_1$ and $A_2$ are arbitrary constants. Adopting the same procedure as above, we get

$$A_1 = f'(x_n) - pf(x_n) f(x_n), \quad A_2 = f(x_n). \tag{2.40}$$

$$p^2 f(x_n) f'^2(x_n) - p \left[ f(x_n) f''(x_n) + 2f'(x_n) f''(x_n) \right] = 0. \tag{2.41}$$

From (2.40) and (2.41), we get another family of iterative methods given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)(1 - pf(x_n))}. \tag{2.42}$$

This is another new one-parameter family of Newton’s method. In order to obtain quadratic convergence, the entity in the denominator should be largest in magnitude. Again note that for $p = 0$, we obtain Newton’s method.

Now we apply the well-known Newton’s method to the modified function $u_j(x) = f(x_n)/(f'(x_n)(1 - pf(x_n)))$, and we will obtain another iterative method as

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)(1 - pf(x_n))}{f'(x_n)(1 - pf(x_n)) + pf^2(x_n)f''(x_n)}. \tag{2.43}$$

This is another new one-parameter modified family of Schröder’s method for an equation having multiple roots of multiplicity $m > 1$ unknown. It is interesting to note that by ignoring the term $p$, (2.43) reduces to Schröder’s method. It is easy to verify that this method is also an order two method, including in the case of multiple zeros. Theorems 2.2 and 2.3 indicate that what choice on the disposable parameter $p$ in families (2.42) and (2.43), the order of convergence will reach at least the second order.

**Theorem 2.2.** Let $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ have at least three continuous derivatives defined on an open interval $I$, enclosing a simple zero of $f(x)$ (say $x = r \in I$). Assume that initial guess $x = x_0$ is sufficiently close to $r$ and $f'(x_n)(1 - pf(x_n)) \neq 0$ in $I$. Then the family of iterative methods defined by (2.42) has at least a second-order convergence and will have a third-order convergence when $p = f''(x_n)/2$. It satisfies the following error equation:

$$e_{n+1} = (-pf'(r) + c_2)e_n^2 - \left( p^2 f''(r) - 2c_2 + 2c_3 \right)e_n^3 + O(e_n^4). \tag{2.44}$$

**Theorem 2.3.** Let $f : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$ have at least three continuous derivatives defined on an open interval $I$, enclosing a simple zero of $f(x)$ (say $x = r \in I$). Assume that initial guess $x = x_0$ is sufficiently close to $r$ and $f'(x_n)(1 - pf(x_n)) \neq 0$ in $I$. Then the family of iterative methods defined by (2.43) has at least a second-order convergence and will have a third-order convergence when $p = f''(x_n)/2$. It satisfies the following error equation:

$$e_{n+1} = (pf'(r) - c_2)e_n^2 + 2\left( 2pf'(r) - c_2 - 2c_3 \right)e_n^3 + O(e_n^4). \tag{2.45}$$
Proof. The proofs of these theorems are similar to the proof of Theorem 2.1. Hence, these are omitted here. □

Now we wish to construct different third-order iterative methods, which are dependent on the different values of $p$ and are given as follows.

**Special Cases**

(i) When $|p| \ll 1$, then $p^2$ can be neglected in (2.41), and we get

$$
 p = \frac{f'(x_n)}{2f^2(x_n) + f(x_n)f''(x_n)}.
$$

(a) Inserting this value of $p$ in (2.42), we get

$$
 x_{n+1} = x_n - \frac{f(x_n) - f'(x_n)f''(x_n)}{f'(x_n) - \frac{f'(x_n)f''(x_n)}{2f^3(x_n)}}.
$$

This is the well-known cubically convergent Chebyshev’s method [1–4, 9]. It satisfies the following error equation:

$$
 e_{n+1} = \left(2c_2^2 - c_3\right)e_n^3 + O(e_n^4).
$$

(b) Again inserting this value of $p$ in (2.43), we get

$$
 x_{n+1} = x_n + \frac{2f(x_n)f'(x_n)}{2f^2(x_n) - f(x_n)f''(x_n)}.
$$

This is the already derived well-known cubically convergent Halley’s method.

(ii) Adopting the same procedure as in (2.21), we get another value of $p$ from (2.41) as

$$
 p = \frac{f''(x_n)}{f^2(x_n)[2 - pf(x_n)] + f(x_n)f''(x_n)}.
$$

Now we substitute $p = \frac{f''(x_n)}{f^2(x_n)[2 - pf(x_n)] + f(x_n)f''(x_n)}$ to get another modified value of $p$ from (2.46) as

$$
 p = \frac{f'(x_n)\left(2f^2(x_n) + f(x_n)f''(x_n)\right)}{4f^4(x_n) + 3f^2(x_n)f(x_n)f''(x_n) + f'(x_n)f^3(x_n)}.
$$

(a) Inserting this value of $p$ in (2.43), we get

$$
 x_{n+1} = x_n - \frac{f(x_n) + \frac{3f(x_n)f'(x_n)f''(x_n) + \frac{2f'(x_n)f(x_n)f''(x_n)}{2f^2(x_n) + f(x_n)f''(x_n)}}{4f^4(x_n) + f^2(x_n)f(x_n)f''(x_n)}}.\]
This is a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{3c_1^2 - 2c_2}{2} \right) e_n^2 + O(e_n^3). \]  

(2.53)

(b) Again inserting this value of \( p \) in (2.43), we get

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{4f^2(x_n) + f(x_n)f''(x_n)}{4f^2(x_n) - f(x_n)f''(x_n)} \right]. \]  

(2.54)

This is again a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{3c_1^2 - 2c_2}{2} \right) e_n^3 + O(e_n^4). \]  

(2.55)

(iii) From (2.41), one can get another value of \( p \) as

\[ p = \frac{p\left(f(x_n)f''(x_n) + 2f'^2(x_n)\right) - f''(x_n)}{p f'^2(x_n) f(x_n)}. \]  

(2.56)

Inserting this previously obtained value of \( p = \left( f'2(x_n) f''(x_n) + f(x_n)f''(x_n) \right) / \left( 4f^2(x_n) - f(x_n)f''(x_n) \right) \) in (2.51), we get another modified value of \( p \) as

\[ p = \frac{f^2(x_n)f''(x_n) - f^2(x_n)f(x_n) + f(x_n)f''(x_n)}{2f^4(x_n) + f^2(x_n)f(x_n)f''(x_n)}. \]  

(2.57)

(a) Inserting this value of \( p \) in (2.42), we get

\[ x_{n+1} = x_n - \frac{f(x_n)f'(x_n) \left( 2f^2(x_n) + f(x_n)f''(x_n) \right)}{2f^4(x_n) + f^2(x_n)f(x_n)f''(x_n) - f^2(x_n)f''(x_n)}. \]  

(2.58)

This is a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{1 - 2c_2}{2} \right) e_n^3 + O(e_n^4). \]  

(2.59)

(b) Again inserting this value of \( p \) in (2.43), we get

\[ x_{n+1} = x_n - \frac{2f^2(x_n)f(x_n) + f'(x_n)f''(x_n) \left( f^2(x_n) - f''(x_n) \right)}{2f^4(x_n) - f^4(x_n)f(x_n)f''(x_n) - f^2(x_n)f'^3(x_n) + f^3(x_n)f''(x_n)}. \]  

(2.60)
This is again a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{6c_1^2 - 1 - 2c_2}{2} \right) e_n^3 + O(e_n^4). \]  

(iv) Now we solve the quadratic equation (2.41) for the general value of \( p \), and we get other values of \( p \) as

\[ p = \frac{\{2f'(x_n) + f(x_n)f''(x_n)\} \pm \sqrt{[2f'(x_n) + f(x_n)f''(x_n)]^2 - 4f(x_n)f''(x_n)f'''(x_n)}}{2f'(x_n)f(x_n)}. \]  

By inserting the above values of \( p \) either in (2.42) or (2.43), we get

\[ x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{f(x_n)f''(x_n) \pm \sqrt{4f'(x_n)f''(x_n) + f''(x_n)f'''(x_n)}}. \]  

This is a new third-order iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{3c_1^2 + 2c_2}{2} \right) e_n^3 + O(e_n^4). \]  

3. Third-Order Multipoint Iterative Methods and Their Error Equations

The practical difficulty associated with the above-mentioned cubically convergent methods may be the evaluation of the second-order derivative. Recently, some new variants of Newton’s method free from second-order derivative have been developed in [3, 10, 11] and the references cited there in by the discretization of the second-order derivative or by the predictor-corrector approach or by considering different quadrature formulae for the computation of integral arising from Newton’s theorem. These multipoint iteration methods calculate new approximations to a zero of a function by sampling \( f(x) \) and possibly its derivatives for a number of values of the independent variable at each step.

Here, we also intend to develop new third-order multipoint methods free from the second-order derivative. The main idea of proposed methods lies in the discretization of the second-order derivative involved in the above-mentioned methods.

Expanding the function \( f(x_n - f(x_n)/f'(x_n)) \) about the point \( x = x_n \) by Taylor’s expansion, we have

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

(3.1)
Therefore, we obtain
\[
\frac{f''(x_n)}{f'(x_n)} = \frac{2f^2(x_n)f(y_n)}{f^2(x_n)},
\] (3.2)

where \( y_n = x_n - \frac{f(x_n)}{f'(x_n)} \).
Using this approximate value of \( f''(x_n) \) into the previously obtained formulæ, we get different multipoint iterative methods free from second-order derivative.

**Special Cases**

(i) Inserting this approximate value of \( f''(x_n) \) (from (3.2)) either in (2.17) or (2.19), we get
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left( \frac{f(y_n) - f(x_n)}{f(x_n) - f(y_n)} \right).
\] (3.3)
This is the well-known third-order Newton-Secant method [3]. It satisfies the following error equation:
\[
\varepsilon_{n+1} = c_1^3 \varepsilon_n^3 + 3c_2 (c_1 - c_2) \varepsilon_n^4 + O(\varepsilon_n^5). \] (3.4)

(ii) Inserting this approximate value of \( f''(x_n) \) (from (3.2)) in (2.23), we get
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left( \frac{f(y_n) - 2f(x_n)}{3f(y_n) - 2f(x_n)} \right).
\] (3.5)
This is a new third-order multipoint iterative method having the error equation
\[
\varepsilon_{n+1} = \left( \frac{c_1^3}{2} \right) \varepsilon_n^3 + \left( \frac{4c_2 c_3 - 3c_2^2}{4} \right) \varepsilon_n^4 + O(\varepsilon_n^5). \] (3.6)

(iii) Inserting this approximate value of \( f''(x_n) \) (from (3.2)) in (2.25), we get
\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left( \frac{f(y_n) - 2f(x_n)}{f(x_n) - 2f(y_n)} \right).
\] (3.7)
This is a new third-order multipoint iterative method having the error equation
\[
\varepsilon_{n+1} = \left( \frac{c_1^3}{2} \right) \varepsilon_n^3 + \left( \frac{4c_2 c_3 - 5c_2^2}{4} \right) \varepsilon_n^4 + O(\varepsilon_n^5). \] (3.8)
(iv) Inserting this approximate value of $f''(x_n)$ (from (3.2)) in (2.29), we get

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

This is a new third-order multipoint iterative method having the error equation

$$e_{n+1} = \left( \frac{c_2}{2} \right) e_n^3 + \left( \frac{2c_2c_1 - c_1}{2} \right) e_n^4 + O(e_n^5).$$

(v) Inserting this approximate value of $f''(x_n)$ (from (3.2)) in (2.31), we get

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

This is a new third-order multipoint iterative method having the error equation

$$e_{n+1} = \left( \frac{c_2}{2} \right) e_n^3 + \left( \frac{2c_2c_1 - c_1}{2} \right) e_n^4 + O(e_n^5).$$

(vi) Inserting this approximate value of $f''(x_n)$ (from (3.2)) in (2.47), we get

$$x_{n+1} = x_n - \frac{f(x_n) + f(y_n)}{f'(x_n)}. $$

This is the well-known third-order Potra-Pták’s method [12]. It satisfies the following error equation:

$$e_{n+1} = 2c_2^2 e_n^3 + \left( 7c_2c_1 - 9c_1^2 \right) e_n^4 + O(e_n^5).$$

(vii) Inserting this approximate value of $f''(x_n)$ (from (3.2)) in (2.52), we get

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

This is a new third-order multipoint iterative method having the error equation

$$e_{n+1} = \left( \frac{3c_1}{2} \right) e_n^3 + \left( \frac{20c_1c_2 - 21c_2^2}{4} \right) e_n^4 + O(e_n^5).$$
(viii) Inserting this approximate value of \( f'(x_n) \) (from (3.2)) in equation (2.54), we get

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

(3.17)

This is a new third-order multipoint iterative method having the error equation

\[
e_{n+1} = \left( \frac{3c_1^2}{2} \right) e_n^4 + \left( \frac{20c_2c_3 - 23c_1^3}{4} \right) e_n^4 + O(e_n^5).
\]

(3.18)

(ix) Inserting this approximate value of \( f''(x_n) \) (from (3.2)) in (2.58), we get

\[
x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)[f(x_n) + f(y_n)]}{6f^2(x_n)[f'(x_n) - 2f'(y_n)] + f^4(x_n)}.
\]

(3.19)

This is a new third-order multipoint iterative method having the error equation

\[
e_{n+1} = \frac{1}{2} e_n^4 + c_2 (c_1^2 - c_1 - 1) e_n^4 + O(e_n^5).
\]

(3.20)

(x) Inserting this approximate value of \( f''(x_n) \) (from (3.2)) in (2.63), we get

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ f'(x_n) \left[ f^4(x_n) + 2f^2(x_n)f^2(x_n) - 2f^2(y_n) \right] \right].
\]

(3.21)

This is a new third-order multipoint iterative method having the error equation

\[
e_{n+1} = \left( \frac{6c_1^2 - 1}{2} \right) e_n^4 + 11c_2 (c_3 - c_2 - 1) e_n^4 + O(e_n^5).
\]

(3.22)

4. Optimal Fourth-Order Multipoint Iterative Methods and Their Error Equations

Now we intend to develop new fourth-order optimal multipoint iterative methods [10, 11, 15–17] for solving nonlinear equations numerically. These multipoint iterative methods are of great practical importance since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. In the case of these multipoint methods, Kung and Traub [13] conjectured that the order of convergence of any multipoint method without memory, consuming \( n \) function evaluations per iteration, cannot exceed the bound \( 2n-1 \) (called optimal order). Multipoint methods with this property are called optimal methods. Traub-Ostrowski’s method [3], Jarratt’s method [14], King’s method [11], a family of Traub-Ostrowski’s method [10], and so forth are famous optimal fourth
order methods, because they require three function evaluations per step. Traub-Ostrowski’s method, Jarratt’s method, and King’s family are the most efficient fourth-order multipoint iterative methods till date. Nowadays, obtaining new optimal methods of order four is still important, because they have very high efficiency index. For this, we will take linear combination of the Newton-Secant method and the newly developed third-order multipoint iterative methods. Let us denote the Newton-Secant method by $N$ and the methods namely (3.5) to (3.17) by $T$, respectively, therefore, taking the linear combination of $N$ (the Newton-Secant method) and $T$ (newly developed multipoint methods) as follows:

\[
\phi(x_n) = aN + \beta T \quad \text{where } a, \beta \in \mathbb{R}.
\]  

(4.1)

For some particular values of $a$ and $\beta$, we get many new fourth-order optimal multipoint iterative methods as follows.

(i) When we take $T$ as a method (3.5) and $(a, \beta) = (-1, 2)$ in (4.1), we get

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

(4.2)

This fourth-order optimal multipoint method is independently derived by Behzad Ghanbari [18]. It satisfies the following error equation

\[
\epsilon_{n+1} = \left(\frac{3\epsilon_n^3 - 2\epsilon_n^2}{2}\right)\epsilon_n + O(\epsilon_n^4). 
\]  

(4.3)

(ii) When we take $T$ as method (3.7) and $(a, \beta) = (-1, 2)$ in (4.1), we get

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)[4f^2(y_n) + 2f^2(x_n) - 5f(x_n)f(y_n)]}{[f(y_n) - 2f(x_n)][f(x_n) - 2f(y_n)][f(y_n) - f(x_n)]}.
\]  

(4.4)

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

\[
\epsilon_{n+1} = \left(\frac{\epsilon_n^3 - 2\epsilon_n^2}{2}\right)\epsilon_n + O(\epsilon_n^4). 
\]  

(4.5)
Program Code in Mathematica 7 for the Order of Convergence of Method (4.4)

In [1]: \( fx = f1r(\epsilon + c\epsilon^2 + c\epsilon^3 + c\epsilon^4) \).
In [2]: \( fdx = f1r(1 + 2\epsilon c + 3\epsilon c^2 + 4\epsilon c^3 + 5\epsilon c^4) \).
In [3]: \( u1 = \text{Series} \left[ \frac{fx}{fdx}, \{\epsilon, 0, 4\} \right] // \text{Simplify} \).
In [4]: \( y = \epsilon - u1 \).
In [5]: \( fy = f1r\left(y + c\epsilon^2 + c\epsilon^3 + c\epsilon^4\right) \).
In [6]: \( u2 = f(x_n) * \left(4f^2(y_n) + 2f^2(x_n) - 5f(x_n)f(y_n)\right) \).
In [7]: \( u3 = (f(y_n) - 2f(x_n))(f(x_n) - 2f(y_n))(f(y_n) - f(x_n)) \).
In [8]: \( u4 = \text{Series} \left[ \frac{y^2}{y^3}, \{\epsilon, 0, 4\} \right] \).
In [9]: \( e1 = \epsilon - u4 // \text{Simplify} \).
Out [10]: \( \left(\frac{c^4}{2} - 2c\epsilon\right) \epsilon^4 + O\left(\epsilon^5\right) \).

(iii) When we take \( T \) as method (3.9) and \((\alpha, \beta) = (-1, 2)\) in (4.1), we get

\[
x_{n+1} = x_n - f(x_n) \left[ \frac{f(x_n) \left(2f^2(x_n) - 2f(x_n)f(y_n) + f^2(y_n)\right)}{2f^3(x_n) - 4f^2(x_n)f(y_n) + f(x_n)f^2(y_n) + f^3(y_n)} \right].
\]  

(4.7)

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

\[
ed_{n+1} = \left(2c^2 - c\epsilon\right)e_n^4 + O\left(\epsilon^5\right).
\]  

(4.8)

(iv) When we take \( T \) as method (3.11) and \((\alpha, \beta) = (-1, 2)\) in (4.1), we get

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

(4.9)
This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

\[ e_{n+1} = -c_2c_3e_n^d + O\left(e_n^3\right). \]  

(v) When we take \( T \) as method (3.13) and \((\alpha, \beta) = (-2, 1)\) in (4.1), we get

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

This is a particular case of quadratically convergent King’s family [11] of multipoint iterative method for \( \gamma = 1 \). It satisfies the following error equation:

\[ e_{n+1} = -\left(3c_2^2 - c_2c_3\right)e_n^d + O\left(e_n^2\right). \]  

(vi) When we take \( T \) as a method (3.15) and \((\alpha, \beta) = (3, -2)\) in (4.1), we get

\[ x_{n+1} = x_n - \frac{2f^3(x_n) + f^2(x_n)f(y_n) + 2f(x_n)f^2(y_n) + 4f^2(y_n)}{f'(x_n)[f(x_n) - f(y_n)][2f(x_n) + f(y_n)]}. \]  

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left(3c_2^3 - c_2c_3\right)e_n^d + O\left(e_n^2\right). \]  

(vii) When we take \( T \) as a method (3.17) and \((\alpha, \beta) = (3, -2)\) in (4.1), we get

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

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

\[ e_{n+1} = \left(\frac{5c_2^3 - 2c_2c_3}{2}\right)e_n^d + O\left(e_n^2\right). \]  

(viii) Using the approximate value of \( f''(x_n) \) (from (3.2)) in (2.19), we get

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

\[ e_{n+1} = \left(\frac{5c_2^3 - 2c_2c_3}{2}\right)e_n^d + O\left(e_n^2\right). \]
This is the well-known Traub-Ostrowski's [3] fourth-order optimal multipoint iterative method. It satisfies the following error equation:

$$e_{n+1} = \left(c_1^3 - c_2c_3\right)e_n + O\left(e_n^3\right).$$

(4.18)

Some Other Formulae

In a similar fashion, let us denote Potra-Pták’s method by \( P \) and the methods, namely, (3.5), (3.9), (3.19) to (3.21), respectively, by \( T \), taking the linear combination of \( P \) (Potra-Pták’s method) and \( T \) (newly developed multipoint methods) as follows:

$$ \psi(x_n) = \alpha P + \beta T \quad \text{where} \quad \alpha, \beta \in \mathbb{R}. $$

(4.19)

For some particular values of \( \alpha \) and \( \beta \), we get many new other fourth-order optimal multipoint iterative methods as follows.

(ix) When we take \( T \) as method (3.9) and \( (\alpha, \beta) = (-1, 4) \) in (4.19), we get

$$ x_{n+1} = x_n - \frac{6f(x_n) + 3f(x_n)f'(y_n) + f'(y_n)}{3f'(x_n)\left\{ 2f'(x_n) - 2f(x_n)f'(y_n) - f'(y_n) \right\}}. $$

(4.20)

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

$$ e_{n+1} = \left(\frac{7}{3}c_1^3 - c_2c_3\right)e_n + O\left(e_n^3\right). $$

(4.21)

(x) When we take method (3.5) \( \times 3 \) – method (3.15), we get

$$ x_{n+1} = x_n - \frac{f(x_n)\left\{ 4f'(x_n) - 4f(x_n)f'(y_n) + 3f'(y_n) \right\}}{f'(x_n)\left\{ 4f'(x_n) - 8f(x_n)f'(y_n) + 3f'(y_n) \right\}}. $$

(4.22)

This is a new fourth-order optimal multipoint iterative method. It satisfies the following error equation:

$$ e_{n+1} = \left(\frac{7}{4}c_1^3 - c_2c_3\right)e_n + O\left(e_n^3\right). $$

(4.23)

Similarly, we can obtain many other new optimal multipoint fourth-order iterative methods for solving nonlinear equations numerically.

It is straightforward to see that per step these methods require three evaluations of function, namely, two evaluations of \( f(x) \) and one of its first-order derivative \( f'(x) \). In order to obtain an assessment of the efficiency of our methods, we shall make use of the efficiency index defined by (1.9). For our proposed third-order multipoint iterative methods, we find...
Table 1: Test problems.

<table>
<thead>
<tr>
<th>f(x)</th>
<th>[a,b]</th>
<th>Root (r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_1(x) = \sin x - \frac{x}{2}</td>
<td>[1.5, 2]</td>
<td>1.895494222640991</td>
</tr>
<tr>
<td>f_2(x) = \cos x - x</td>
<td>[0, 2]</td>
<td>0.739085137844086</td>
</tr>
<tr>
<td>f_3(x) = x^7 - 10</td>
<td>[2.4]</td>
<td>2.154434680938721</td>
</tr>
<tr>
<td>f_4(x) = 10x^e^x - 1</td>
<td>[1, 2]</td>
<td>1.697930687168898</td>
</tr>
<tr>
<td>f_5(x) = (x - 1)^3 - 1</td>
<td>[1.5, 3.5]</td>
<td>2.000000000000000</td>
</tr>
<tr>
<td>f_6(x) = \tan^{-1} x - x + 1</td>
<td>[1.5, 3]</td>
<td>2.13267713546753</td>
</tr>
<tr>
<td>f_7(x) = x^4 - x^3 + 11x - 7 = 0</td>
<td>[0, 1]</td>
<td>0.645023941993713</td>
</tr>
<tr>
<td>f_8(x) = x^3 + 17x = 0</td>
<td>[-0.5, 1]</td>
<td>0</td>
</tr>
<tr>
<td>f_9(x) = x^3 - \cos x + 2 = 0</td>
<td>[-2, -1]</td>
<td>-1.17257977180481</td>
</tr>
<tr>
<td>f_{10}(x) = e^{10x^{1.2}x^3 - 30} - 1 = 0</td>
<td>[2.9, 3.5]</td>
<td>3</td>
</tr>
</tbody>
</table>

P = 3 and D = 3 to get E = \sqrt{3} \approx 1.442 which is better than E = \sqrt{2} = 1.414, the efficiency index of the Newton's method. For the quadratically convergent multipoint iterative methods, we find P = 4 and D = 3 to get E = \sqrt{4} \approx 1.587 which is better than those of most of the third-order methods E \approx 1.442 and Newton's method E \approx 1.414.

5. Numerical Experiments

In this section, we shall check the effectiveness of the new optimal methods. We employ the present methods, namely, (4.2), (4.9), method (4.15), (4.22) respectively, to solve the following nonlinear equations given in Table 1. We compare them with the methods, namely, Newton's method (NM), Traub-Ostrowski's method (also known as Ostrowski's method) (4.17) (TOM), Jarratt's method (JM), and King's method (KM) for γ = 1/2 and γ = 1 respectively. We have also shown the comparison of all methods mentioned above in Table 2. Computations have been performed using C++ in double-precision arithmetic. We use ε = 10^{-15} as a tolerable error. The following stopping criteria are used for computer programs:

(i) |x_{n+1} - x_n| < ε,
(ii) |f(x_{n+1})| < ε.

6. Conclusion

In this paper, we have presented another simple and elegant way of deriving different iterative functions to solve nonlinear equations numerically. This study represents several formulae of third and fourth order and has a well-known geometric derivation. Multipoint iterative methods belong to the class of the most powerful methods since they overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. The most important class of multipoint methods are optimal methods, which attain the convergence order 2n - 1 using n function evaluations per iteration. Therefore, fourth-order multipoint iterative methods are the main findings of the present paper in terms of speed and efficiency index. According to Kung-Traub conjecture, these different methods presented in this paper have the maximal efficiency index because only three function values are needed per step. The numerical results presented in Table 2, overwhelmingly, support that these different methods are equally competent to Traub-Ostrowski’s method, Jarratt’s method, and
### Table 2: Total number of iterations to approximate the zero of a function and the total number of function evaluations for various multipoint iterative methods.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Initial guess</th>
<th>NM</th>
<th>TOM</th>
<th>JM</th>
<th>KM (y = 1/2)</th>
<th>KM (y = 1)</th>
<th>Method (4.4)</th>
<th>Method (4.9)</th>
<th>Method (4.15)</th>
<th>Method (4.22)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x)$</td>
<td>1.5 (4.8) (2.6) (3.9) (3.9) (4.12) (2.6) (2.6) (3.9) (3.9)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (3.9) (3.9)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>4 (5.10) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1 (4.8) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (6.12) (3.9) (3.9) (4.12) (5.15) (3.9) (4.12) (4.12) (4.12)</td>
<td>3.5 (6.12) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>3 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>0 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
</tr>
<tr>
<td>$f_2(x)$</td>
<td>0 (4.8) (2.6) (3.9) (3.9) (3.9) (2.6) (2.6) (3.9) (3.9)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>4 (5.10) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1 (4.8) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (6.12) (3.9) (3.9) (4.12) (5.15) (3.9) (4.12) (4.12) (4.12)</td>
<td>3.5 (6.12) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>3 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>0 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
</tr>
<tr>
<td>$f_3(x)$</td>
<td>0 (4.8) (2.6) (3.9) (3.9) (3.9) (2.6) (2.6) (3.9) (3.9)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>2 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>4 (5.10) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1 (4.8) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (6.12) (3.9) (3.9) (4.12) (5.15) (3.9) (4.12) (4.12) (4.12)</td>
<td>3.5 (6.12) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9) (3.9)</td>
<td>1.5 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>3 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
<td>0 (3.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6) (2.6)</td>
</tr>
</tbody>
</table>

King’s family. By using the same idea, one can obtain other iterative processes by considering different exponentially fitted osculating curves.

### Acknowledgments

The authors would like to record their sincerest thanks to the Editor, Professor Alicia Cordero, and anonymous reviewers for their constructive suggestion and remarks which have considerably contributed to the readability of this paper. R. Behl acknowledges further the financial support of CSIR, New Delhi, India.

### References


Simply constructed family of a Ostrowski's method with optimal order of convergence

V. Kanwar a,*, Ramandeep Behl b, Kapil K. Sharma b

a University Institute of Engineering and Technology, Panjab University, Chandigarh- 160 014, India
b Department of Mathematics, Panjab University, Chandigarh-160 014, India

ABSTRACT

In this paper, we propose a simple modification over Chun’s method for constructing iterative methods with at least cubic convergence [5]. Using iteration formulas of order two, we now obtain several new interesting families of cubically or quartically convergent iterative methods. The fourth-order family of Ostrowski’s method is the main finding of the present work. Per iteration, this family of Ostrowski’s method requires two evaluations of the function and one evaluation of its first-order derivative. Therefore, the efficiency index of this Ostrowski’s family is $E = \sqrt[4]{\frac{1}{3}} \approx 1.87$, which is better than those of most third-order iterative methods $E = \sqrt[3]{\frac{1}{2}} \approx 1.44$ and Newton’s method $E = \sqrt{2} \approx 1.41$. The performance of Ostrowski’s family is compared with its closest competitors, namely Ostrowski’s method, Jarratt’s method and King’s family in a series of numerical experiments.

1. Introduction

Various problems arising in the diverse disciplines of science, engineering and nature can be described by nonlinear equations of the form

$$f(x) = 0.$$  \hspace{1cm} (1.1)

The best known and most widely used algorithm for solving such problems is the classical Newton’s method [1-4], which is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, 3, \ldots \hspace{1cm} (1.2)$$

Many researchers have developed modifications of Newton’s method or Newton-like methods [5-12] in a number of ways to improve the local order of convergence of Newton’s method at the expense of additional evaluations of the functions and/or derivatives, mostly at the point iterated by the method. All these modifications are targeted at increasing the local order of convergence with the view of increasing their efficiency index [2].

Kung and Traub [6] have conjectured that multipoint iteration methods without memory based on $n$ function evaluations have optimal order of convergence $2n-1$. The famous Ostrowski’s method [1-8] and Jarratt’s method [7] are examples of fourth-order multipoint methods without memory. These methods are the most efficient fourth-order multipoint iterative methods.

* Corresponding author.
E-mail address: vimithi@yahoo.co.in (V. Kanwar).
methods known to date. Another well-known example of a fourth-order multipoint method with the same number of function evaluations is King’s family \[8\]. This family is defined as:

\[ x_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)} + \frac{y}{2}f''(w_n), \]

where \( w_n = x_n - \frac{f(x_n)}{f'(x_n)} \) and \( y \in \mathbb{R} \).

The research of finding iterative methods with optimal fourth-order convergence, not requiring the computation of a second-order derivative, is important and interesting from the practical point of view. In this work, we contribute further to the development of the theory of iteration processes and derive many families of new third- and fourth-order multipoint iterative methods. Ostrowski’s family of methods requires two evaluations of the function \( f(x) \) and one of its derivatives \( f'(x) \) per iteration and the efficiency index is the same as that of Ostrowski’s method. These methods are obtained by introducing quadratically convergent methods, a secant line and a parabola while moving along the curve to solve nonlinear equations numerically, and the approach for deriving the formula is a different one.

2. Basic definitions

**Definition 1.** Let \( f(x) \) be a real valued function with a simple root \( r \) and let \( \{x_n\}_{n=0}^\infty \) be a sequence of real numbers that converges to \( r \). Then, we say that the order of convergence of the sequence is \( p \), if there exists a \( p \in \mathbb{R}^+ \) such that

\[
\lim_{n \to \infty} \frac{r - x_{n+1}}{(r - x_n)^p} = C \neq 0,
\]

where \( C \) is known as the asymptotic error constant. If \( p = 1, 2 \) or \( 3 \), the sequence \( \{x_n\} \) is said to have linear, quadratic or cubic convergence, respectively.

**Definition 2.** Let \( e_n = x_n - r \) be the error in the \( n \)th iteration. We call the relation

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

the error equation. If we can obtain the error equation for any iterative method, then the value of \( p \) is its order of convergence.

**Definition 3.** Let \( d \) be the number of new pieces of information (function or its derivatives) required by an iterative method per step. Then the efficiency of the method may be measured by the efficiency index introduced by Ostrowski [2] and defined as:

\[
E = p^\frac{1}{3}.
\]

3. Development of the methods

Consider a nonlinear equation

\[
f(x) = 0,
\]

whose one or more roots are to be found. Let \( x = r \) be a simple root of the nonlinear equation (3.1) and \( x = x_0 \) be the initial guess to the required root.

Here we also intend to construct new families of iteration functions from available iteration functions of order two based on a geometric observation. In the sequel, whenever we mention that an iteration function \( \phi \) is of order \( p \), it means that the corresponding iterative method defined by \( x_{n+1} = \phi(x_n) \) is of convergence order \( p \), that is, the error \( |r - x_{n+1}| \) is proportional to \( |r - x_n|^p \) as \( n \to \infty \). We will indicate here that \( \phi \) is an iteration function whose order is \( p \) by writing \( \phi \in \mathcal{P}^p \).

Our proposed scheme (similar to Chun) to develop new families of iteration functions is constructed geometrically as follows:

Let \( \beta \) be a fixed parameter with \( 0 < \beta < 1 \) and let \( \phi(x_0) \in \mathcal{P}^2 \) be an iteration function of order two.

Let

\[
y = f(x),
\]

represents the graph of the function \( f(x) \). Assume that two points, namely \( \left( x_0, \frac{\phi(x_0) + f(x_0)}{2} \right) \) and \( \left( x_0, \phi(\phi(x_0)) \right) \), lie on the same graph of the function \( y = f(x) \). Then the approximated line of the function \( f(x) \) passing through the above mentioned points is given by

\[
y - \beta f(\phi(x_0)) = \frac{2\beta f(\phi(x_0)) - f(x_0)}{\phi(x_0) - x_0} (x - \phi(x_0)).
\]

Draw a parabola with vertex at \( (\phi(x_0), 0) \) and axis parallel to the \( y \)-axis on the graph of the same function (3.2). The equation of this parabola is given by

\[
y = a(x - \phi(x_0))^2,
\]

where
where $a$ is the scaling parameter. The parabola (3.4) widens as $a$ approaches zero and narrows as $|a|$ becomes large. The intersection of a line (3.3) with the parabola (3.4) is obtained by setting them equal to each other since each equals $y$. Therefore, we end up with a quadratic equation given by

$$\alpha(x - \phi(x_0))^2 - \frac{2\beta f(\phi(x_0)) - f(x_0)}{\phi(x_0) - x_0}(x - \phi(x_0)) - \beta f(\phi(x_0)) = 0. \quad (3.5)$$

Solving this quadratic equation for $(x - \phi(x_0))$, and after some simplification, we get the first approximation to the required root as

$$x = \phi(x_0) + \frac{2\beta f(\phi(x_0))}{\beta f(\phi(x_0)) - f(x_0)} \pm \sqrt{\left(\frac{2\beta f(\phi(x_0)) - f(x_0)}{\beta f(\phi(x_0)) - x_0}\right)^2 + 4\alpha \beta^2 f(\phi(x_0))}$$

$$\cdot \frac{2\alpha}{2\alpha}. \quad (3.6)$$

This can further be rewritten in the equivalent form (by rationalizing the numerator) as

$$x = \phi(x_0) - \frac{2\beta f(\phi(x_0))}{\beta f(\phi(x_0)) - f(x_0)} \pm \sqrt{\left(\frac{2\beta f(\phi(x_0)) - f(x_0)}{\beta f(\phi(x_0)) - x_0}\right)^2 + 4\alpha \beta^2 f(\phi(x_0))}$$

$$\cdot \frac{2\alpha}{2\alpha}. \quad (3.7)$$

in which the sign should be chosen so as to make the denominator largest in magnitude.

Now consider the factor

$$\frac{4\alpha \beta^2 f(\phi(x_0))}{\beta f(\phi(x_0)) - f(x_0)} \leq 1. \quad (3.8)$$

Since the scaling parameter $\alpha$ appears in the numerator of (3.8), it is clear that there exists some real values of $\alpha$ such that

$$\left|\frac{4\alpha \beta^2 f(\phi(x_0))}{\beta f(\phi(x_0)) - f(x_0)}\right|^2 < 1. \quad (3.9)$$

holds.

With this assumption, the binomial theorem is applicable in Eq. (3.7) and one can get the following formula free from the square root term as

$$x = \phi(x_0) - \frac{\beta f(\phi(x_0))(x_0 - \phi(x_0))(f(x_0) - 2\beta f(\phi(x_0)))}{(f(x_0) - 2\beta f(\phi(x_0)))^2 + \alpha \beta^2 f(\phi(x_0))(x_0 - \phi(x_0))^2}.$$

(3.10)

Now repeating this process until the parabola becomes x-axis, the general formula for successive approximation is given by

$$\psi(x_{n+1}) = x_{n+1} = \phi(x_n) - \frac{\beta f(\phi(x_n))(x_n - \phi(x_n))(f(x_n) - 2\beta f(\phi(x_n)))}{(f(x_n) - 2\beta f(\phi(x_n)))^2 + \alpha \beta^2 f(\phi(x_n))(x_n - \phi(x_n))^2}.$$

(3.11)

The family of iteration functions $\psi(x_{n+1})$ constructed in this manner again has order of convergence equal to at least three when $\beta = 1$. This is proved by the next theorem. A similar approach for deriving the family of Secant-like methods has been used previously by Kanwar et al. [13].

4. Order of convergence

**Theorem 4.1.** Let $r$ be a simple zero of $f(x)$ and $\phi(x)$ be an iteration function with $\phi \in \mathbb{C}$, such that $\phi'''(r)$ is continuous in a neighborhood of $r$. Let

$$\psi(x_{n+1}) = \phi(x_n) - \frac{\beta f(\phi(x_n))(x_n - \phi(x_n))(f(x_n) - 2\beta f(\phi(x_n)))}{(f(x_n) - 2\beta f(\phi(x_n)))^2 + \alpha \beta^2 f(\phi(x_n))(x_n - \phi(x_n))^2}.$$

(4.1)

Then $\phi \in \mathbb{C}$ for some $p$ with $p \geq 3$ when $\beta = 1$. Furthermore, if $\phi''(r) = 0$ or $\phi'''(r) = 0$, then $\phi \in \mathbb{C}$ for some $p$ with $p \geq 4$.

**Proof.** Let $r$ be a simple zero of $f(x)$. Since $\phi(x)$ is an iteration function of order two, then we have $\phi(r) = r$ and $\phi'(r) = 0$. Expanding $f(x_n)$ and $\phi(x_n)$ about $r$ by a Taylor series expansion, we have

$$f(x_n) = f'(r)(x_n - r) + O(e^2_n).$$

(4.2)
and
\[ \phi(x_n) = r + \frac{\phi''(r)}{2} x_n^2 + \frac{1}{6} \phi'''(r) x_n^3 + O(x_n^4), \] 
(4.3)
respectively, where \( e_n = x_n - r \) and \( e_k = \frac{\phi^{(k)}(r)}{k!}, \) \( k = 2, 3, \ldots. \)
Furthermore
\[ x_n - \phi(x_n) = e_n - \frac{\phi''(r)}{2} e_n^2 - \frac{1}{6} \phi'''(r) e_n^3 + O(e_n^4), \]
(4.4)
and in combination with the Taylor series expansion of \( f(\phi(x_n)) \) about \( r, \) we have
\[ f(\phi(x_n)) = f'(r) \left( \frac{\phi''(r)}{2} e_n^2 + \frac{1}{6} \phi'''(r) e_n^3 + O(e_n^4) \right) \]
and
\[ f(x_n) - 2f(\phi(x_n)) = f'(r) \left( e_n + \left( c_2 - \frac{\beta \phi''(r)}{3} \right) e_n^2 + \left( c_3 - \frac{\beta \phi'''(r)}{3} \right) e_n^3 \right) + O(e_n^4). \]
(4.5)
Using (4.3)-(4.6), we get
\[ \frac{f(\phi(x_n))}{f'(r) \phi''(r)}(x_n - \phi(x_n)) = \frac{f'(r)}{2} \left( \frac{6\beta - 3}{12} \phi''(r) + 2\phi'''(r) - 6\phi''(r) c_2 \right) e_n^3 \]
(4.7)
which further implies that
\[ e_{n+1} = e_n - \frac{\beta}{2} \phi''(r) e_n^2 + \frac{1}{12} \left( 2\phi'''(r) - \beta (6\beta - 3) \phi''(r) + 2\phi'''(r) - 6\phi''(r) c_2 \right) e_n^3. \]
(4.8)
Therefore, when \( \beta = 1, \) we have the following error equation:
\[ e_{n+1} = e_n - \frac{\phi''(r)}{2} e_n^2 + \frac{1}{12} \left( 2\phi'''(r) - \beta (6\beta - 3) \phi''(r) + 2\phi'''(r) - 6\phi''(r) c_2 \right) e_n^3. \]
(4.9)
This means that the iteration function \( \varphi \) defined by (4.1) is of order at least three. Furthermore, when \( \phi''(r) = 0 \) or \( \phi'''(r) = \frac{\phi''(r)}{2}, \) we can observe from (4.10) that \( \varphi \) is of order at least four. This completes the proof. \( \square \)

It should be noted that the result of Theorem 4.1 is independent of the structure of the iteration function of order two involved. Therefore, this is a further modification of Chun’s basic tool for deriving the families of higher order iterative methods. This idea can further be extended for the case of multiple roots if the quadratically convergent methods for the multiple roots are taken into consideration.
5. Examples

Example 5.1. Consider the Newton’s scheme defined by \( \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \). In this case, \( \psi(x_{n+1}) \) defined by (4.1) becomes

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

This is the new family of Ostrowski’s method and also has order of convergence four [1-5]. This exactly agrees with the result predicted by Theorem 4.1, since \( \epsilon_n = \frac{C_1}{f'(r)} \) and the error equation of the above family when \( \beta = 1 \) is

\[
e_n = \left( c_2 + \frac{\alpha}{f'(r)} \right) e_n^3 - c_2 e_n^3 = O(e_n^3).
\]

Example 5.2. Consider the Stirling’s scheme of order two defined by \( \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \). In this case, \( \psi(x_{n+1}) \) defined by (4.1) becomes

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

By Theorem 4.1, the order of convergence of the iteration function \( \psi(x_{n+1}) \) defined by (5.3) is at least three. From an elementary computation for the coefficient of \( e_n^3 \) in the error equation (4.10), we have the following error equation:

\[
e_n = \left( c_2 + \frac{\alpha}{f'(r)} \right) e_n^3 + O(e_n^3).
\]

Example 5.3. Consider the second-order scheme defined in [9,10] by \( \phi(x_n) = x_n - \frac{f(x_n)}{f(x_n) + f'(x_n)} \). In this case the iteration scheme \( \psi(x_{n+1}) \) defined by (4.1) becomes

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

The order of convergence of this iteration function \( \psi(x_{n+1}) \) is three and has the following error equation:

\[
e_n = -\left( 1 + c_2 \right) e_n^3 + O(e_n^3).
\]

Example 5.4. Consider the Steffensen’s scheme defined by \( \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \), where \( g(x_n) = \frac{f(x_n) - f(y_n)}{f'(x_n)} \). Then the iteration scheme defined by (4.1) becomes

\[
\psi(x_{n+1}) = x_n - \frac{f'(x_n)}{f(x_n) + f'(x_n)} - \frac{f'(x_n)}{f(x_n) + f'(x_n)} \left[ 1 + \frac{\beta f \left( y_{n+1} - \frac{f(y_{n+1})}{f'(y_{n+1})} \right) \left\{ f(y_{n+1}) - 2 \beta f (y_{n+1}) \right\}^2 + \alpha \beta f^2(y_{n+1}) f'(y_{n+1}) \right].
\]

where \( y_{n+1} = x_n - \frac{f(x_n) - f(y_n)}{f'(x_n) - f'(y_n)} \).

This scheme is again cubically convergent and has the following error equation:

\[
e_n = -\frac{1}{4} \left( c_2 + f'(r) \right) e_n^3 + O(e_n^3).
\]

Example 5.5. Consider the Mamta et al. scheme [10] defined by \( \phi(x_n) = x_n - \frac{f(x_n) f'(x_n)}{f'(x_n) + f'(x_n)} \), in this case we have \( \phi''(r) = \frac{C_2}{f'(r)} \).

Hence, by Theorem 4.1, the iteration function \( \psi(x_{n+1}) \) defined by (4.1) becomes
\[
\psi(x_{n+1}) = x_n - \frac{f(x_n) f'(x_n)}{f'(x_n) + f''(x_n)} \\
\times \left[ 1 + \frac{f'(y_{n+1})(f(x_n) - 2f(y_{n+1}))}{f'(x_n) + f''(x_n)} \right]
\]

(5.9)

where \( y_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \).

Taking a particular value of \( \beta = 1 \), we get the following error equation of the above method as:

\[
\varepsilon_{n+1} = c_2 \left( c_2^2 - c_3 + \frac{\alpha c_2}{f'(r)} - 1 \right) \varepsilon_n^3 + O(\varepsilon_n^4).
\]

(5.10)

It should be pointed out that the obtained family of methods (5.1), (5.3), (5.5), (5.7) and (5.9) converge cubically or quartically even though per iteration they require two evaluations of \( f(x) \) and one of \( f'(x) \) or three evaluations of \( f(x) \) and none of \( f'(x) \). In a similar fashion, we can continuously construct other families of iterative methods with at least cubic convergence by making use of quadratically convergent iterative methods as long as they are available.

Similarly, if we take two points \( (\phi(x_0), f(\phi(x_0))) \) and \( \left( \frac{y_{n+1} + x_n}{2}, \frac{f(y_{n+1}) + f(x_n)}{2} \right) \) on the graph of the function \( y = f(x) \), and using the same concept as mentioned above, we get

\[
\psi(x_{n+1}) = \phi(x_n) - \frac{f(\phi(x_n))(x_n - \phi(x_n)) - f(\phi(x_n))}{(x_n - \phi(x_n))^2 + \alpha f(\phi(x_n))(x_n - \phi(x_n))^2}.
\]

(5.11)

This family has the following error equation:

\[
\varepsilon_{n+1} = \frac{1}{2} \frac{\phi'(r)}{c_2^2} \varepsilon_n^3 + O(\varepsilon_n^4).
\]

(5.12)

If we take \( \phi(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} \) as the Newton's iterate, then we get the cubically convergent family of the Newton-Secant method given by

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

(5.13)

The error equation for this family is given by

\[
\varepsilon_{n+1} = c_2^2 \varepsilon_n^3 + O(\varepsilon_n^4).
\]

(5.14)

It is straightforward to see that per step these methods require three functional evaluations, viz. two evaluations of \( f(x) \) and one of \( f'(x) \) or three evaluations of \( f(x) \) and none of \( f'(x) \). In order to obtain an assessment of the efficiency of our methods, we shall make use of the efficiency index defined by Eq. (2.3). For our proposed iteration schemes, namely (5.3), (5.5) and (5.7), we find \( p = 3 \) and \( d = 3 \), yielding \( E = \sqrt{3} \approx 1.442 \), which is better than \( E = \sqrt{2} \approx 1.414 \), the efficiency index of the Newton's method. For the family of methods, namely (5.1) and (5.8), we find \( p = 4 \) and \( d = 3 \), yielding \( E = \sqrt{4} \approx 1.587 \), which is better than those of most third order methods \( E \approx 1.442 \) and Newton's method \( E \approx 1.414 \).

6. Numerical experiments

In this section, we shall present the numerical results obtained by employing various methods, namely Newton's method (NM), Traub–Ostrowski's method (also known as Ostrowski's method) (TOM), Jarratt's method (JM), modified Traub–Ostrowski's method (MTOM) (5.2) for \( \alpha = -1.0, \alpha = 0.5, \alpha = 1.0, \alpha = 10, \) and King's method (KM) (1.3) for \( y = 0.5, \gamma = 1 \) respectively to solve the nonlinear equations given in Table 1. We also show the comparison of all methods mentioned above in Table 2; computations were performed using C++ in double precision arithmetic. We use \( \varepsilon = 10^{-15} \) as the tolerable error. The following stopping criteria are used for the computer programs:

(i) \( |x_{n+1} - x_n| < \varepsilon \), (ii) \( |f(x_{n+1})| < \varepsilon \).

7. Conclusions

In this paper, we have modified Chun's scheme for constructing iterative methods of order three or higher to solve nonlinear equations numerically. Now we have obtained a wide class of general methods which are without memory and include three functional evaluations per iteration. A fourth-order family of Ostrowski's method is the main finding of the present contribution in terms of speed and efficiency index. According to the Kung–Traub conjecture, the family of Ostrowski’s method and the family (5.9) have the maximal efficiency index because only three function values are needed per step. The numerical results presented in Table 2 overwhelmingly support that the new family of Ostrowski's method
Table 1
Test problems.

<table>
<thead>
<tr>
<th>No.</th>
<th>Problem</th>
<th>([a, b])</th>
<th>Initial guess</th>
<th>Root(r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(x^4 - 4x^2 = 0)</td>
<td>([0.5, 2])</td>
<td>0.5</td>
<td>2.0</td>
</tr>
<tr>
<td>2.</td>
<td>(x^3 + 4x^2 - 10 = 0)</td>
<td>([1, 2])</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td>3.</td>
<td>(\cos x - x = 0)</td>
<td>([0, 2])</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.</td>
<td>(x^2 - \sqrt{x} - 2 = 0)</td>
<td>([0.1])</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5.</td>
<td>(x^2 - \sin x + 3e^x + 5 = 0)</td>
<td>([-1.5, -0.5])</td>
<td>-1.5</td>
<td>-1.5</td>
</tr>
<tr>
<td>6.</td>
<td>(\sin^2 x - x^2 + 1 = 0)</td>
<td>([-1.3])</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7.</td>
<td>(e^{x^2 + x} - 1 = 0)</td>
<td>([2.9, 3.5])</td>
<td>2.9</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Table 2
Total number of iterations to approximate the zero of a function, total number of function evaluations for various multipoint iterative methods.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Initial guess</th>
<th>NM</th>
<th>TOM</th>
<th>JM</th>
<th>MTOM ((\alpha = 0.5))</th>
<th>MTOM ((\alpha = 1))</th>
<th>MTOM ((\alpha = 10))</th>
<th>KM ((\alpha = 0.5))</th>
<th>KM ((\alpha = 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>(4, 8)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(3.9)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>(5, 10)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>(4, 8)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>(4, 8)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>5</td>
<td>2.0</td>
<td>(3, 6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>6</td>
<td>0.0</td>
<td>(3, 6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>(3, 6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
<td>(2.6)</td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>(5, 10)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
</tr>
<tr>
<td>9</td>
<td>1.0</td>
<td>(5, 10)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
</tr>
<tr>
<td>10</td>
<td>2.9</td>
<td>(6, 18)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
<td>(3.9)</td>
</tr>
<tr>
<td>11</td>
<td>3.5</td>
<td>(11, 22)</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(6, 18)</td>
<td>(6, 18)</td>
<td></td>
</tr>
</tbody>
</table>

(D above stands for divergent).

is equally competent to Ostrowski’s method, Jarratt’s method and the King’s family. Further, we have also determined that the family of Ostrowski’s method gives a very good approximation to the required root when |\(\alpha\)| (the scaling parameter) is small. This is because, for small values of \(\alpha\), the parabola widens along the horizontal direction. This means that our next approximation will move faster towards the desired root. For large values of \(\alpha\) (provided that the inequality (3.9) holds), the formula still works but takes a greater number of iterations as compared to smaller values of \(\alpha\). This idea can be further extended to the case of multiple roots.

Acknowledgments

We would like to record our sincerest thanks to the anonymous reviewers for their constructive suggestion and remarks which have considerably contributed to the readability of this paper. Ramandeep Behl further acknowledges the financial support of CSIR, New Delhi, India.

References

International Journal of Computer Mathematics

Optimal equi-scaled families of Jarratt's method

Ramandeep Behl a, V. Kanwar b & Kapil K. Sharma a

a Department of Mathematics, Panjab University, Chandigarh-160 014, India
b University Institute of Engineering and Technology, Panjab University, Chandigarh-160 014, India


To cite this article: Ramandeep Behl, V. Kanwar & Kapil K. Sharma (2013): Optimal equi-scaled families of Jarratt's method, International Journal of Computer Mathematics, 90:2, 408-422

To link to this article: http://dx.doi.org/10.1080/00207160.2012.719078

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Optimal equi-scaled families of Jarratt’s method

Ramandeep Behla, V. Kanwar and Kapil K. Sharma

Department of Mathematics, Panjab University, Chandigarh-160 014, India; University Institute of Engineering and Technology, Panjab University, Chandigarh-160 014, India

(Received 8 June 2012; accepted 31 July 2012)

In this paper, we present many new fourth-order optimal families of Jarratt’s method and Ostrowski’s method for computing simple roots of nonlinear equations numerically. The proposed families of Jarratt’s method having the same scaling factor of functions as that of Jarratt’s method (i.e. quadratic scaling factor of functions in the numerator and denominator of the correction factor) are the main finding of this paper. It is observed that the body structures of our proposed families of Jarratt’s method are simpler than those of the original families of Jarratt’s method. The efficiency of these methods is tested on a number of relevant numerical problems. Furthermore, numerical examples suggest that each member of the proposed families can be competitive to other similar robust methods available in the literature.

Keywords: nonlinear equations; simple roots; Halley’s method; Schröder’s method; Jarratt’s method; Ostrowski’s method; optimal order of convergence; efficiency index

2010 AMS Subject Classifications: 65H05; 65H99

1. Introduction

Finding the simple roots of a nonlinear equation \( f(x) = 0 \) is a common and important problem in applied sciences. Analytical methods for solving such equations are almost non-existent and therefore it is only possible to obtain approximate solutions by relying on numerical methods based on iterative procedures. One of the best known one-point optimal methods is classical Newton’s method. Many methods that improve the convergence rate of Newton’s method have been developed in [1–17] and the references cited therein. For example, some third-order and fourth-order modifications of Newton’s method are Chebyshev’s method [5,6,16], Halley’s method [5,6,15,16], super-Halley method [2,10], Weerakoon and Fernando method [17], Potra-Pták’s method [1], etc. Kanwar and Tomar [5,6] have developed some new families of third-order methods in which \( f'(x) = 0 \) is permitted at some points. Kumar et al. [13] have developed some third-order families of multipoint iterative methods based on power means for finding multiple roots of nonlinear equations. Peteković et al. [8] have proposed some modifications of Newton-like interval method for the simultaneous inclusion of all simple complex zeros of a polynomial using Jarratt’s method. Very recently, Kim [8] has developed a new triparametric family of three-step optimal eighth-order iterative methods free from second derivatives by introducing seven free disposable parameters.

*Corresponding author. Email: vimithil@yahoo.co.in

© 2013 Taylor & Francis
As the order of an iterative method increases, so does the number of functional evaluations per step. The efficiency index \( p/d \) gives a measure of the balance between these quantities, according to the formula \( p/d \), where \( p \) is the order of convergence of the method and \( d \) the number of functional evaluations per step. According to the Kung–Traub conjecture [11], the order of convergence of any multipoint method cannot exceed the bound \( 2^{n-1} \), called the optimal order. Thus, the optimal order for a method with three functional evaluations per step would be four. Ostrowski’s method [7,12], Jarratt’s method [3,4] and King’s method [9] are the most efficient fourth-order methods known to date. The most striking feature of these multipoint methods without memory is that they have the quadratic scaling factor of functions in the numerator and denominator of the correction factor. Nowadays, obtaining new optimal methods of order four with quadratic scaling factor in the correction factor, not requiring the computation of a second-order derivative, is a very important and interesting task from the practical point of view, because their corresponding efficiency index, 1.5874, is very competitive.

The contents of this paper unfold the material in what follows. Section 2 presents a brief look at the existing families of Jarratt’s method, where it is followed by Section 3 wherein our main contribution lies. We develop the families of Jarratt’s method having the same scaling factor of functions as that of Jarratt’s method. Some new families of Ostrowski’s method are also proposed. Section 4 includes a numerical comparison between proposed methods without memory and the existing robust methods such as Jarratt’s method and Ostrowski’s method. Finally, the concluding remarks of the paper are drawn in Section 5.

2. Brief literature review

Ostrowski [12] was the first mathematician to discover the optimal fourth-order multipoint iterative method without memory having the quadratic scaling factor of functions in the correction factor. Ostrowski’s method requires two function evaluations and one of its derivative per full iteration. After that in 1966, Jarratt [3,4] had given the optimal families of fourth-order multipoint methods requiring three functional evaluations per full iteration. Of these proposed families, only two methods became popular because they had a quadratic scaling factor of functions in the correction factor as well as simple body structures, which are given by

\[
y_n = x_n - \frac{2 f(x_n)}{3 f'(x_n)},
\]

and

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

\[
 y_n = x_n - \frac{2 f(x_n)}{3 f'(x_n)},
\]

and

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{9 f'(x_n) - 5 f'(y_n)}{10 f'(x_n) - 6 f'(y_n)} \right].
\]

The rest of the methods have a cubic scaling factor of functions in their correction factors. Again in 1975, King [9] had proposed an optimal family of fourth-order multipoint methods having the same scaling factor of functions in the correction factor as that of Ostrowski’s method and Jarratt’s method. King had also shown that Ostrowski’s method was a special case of his family.
With the advancement of computer algebra, we have developed [7] the optimal families of fourth-order methods without caring about the scaling factor of functions in the correction factors. But Ostrowski’s method [7,12], Jarratt’s method [3] and King’s method [9] are the most efficient fourth-order methods known to date. They became popular because of their efficiency and the quadratic scaling factor of functions in their correction factors. But till today Jarratt’s method and Ostrowski’s method do not have the families having a quadratic scaling factor of functions in their correction factors.

With this aim, we intend to develop the families of Jarratt’s method having the same scaling factor as that of Jarratt’s method. Our proposed families of iterative methods can easily be derived by taking the arithmetic mean of discretized (free from second-order derivative) Schröder’s method [14] and Halley’s method [16]. The proposed families are equally competent with similar existing classical methods available in the literature.

3. Construction of novel techniques without memory

3.1 Case I: new optimal families of Jarratt’s method

The well-known Schröder method [14] for multiple zero and Halley method [16] for simple zero are given by

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

and

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

respectively.

We now intend to develop new optimal families of Jarratt’s method having the quadratic scaling factor of functions in the correction factor. For this, we take the arithmetic mean of Equations (3) and (4) to get

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

Now, consider a Newton-type iterative method

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

where \( a \neq 0 \in \mathbb{R} \).

\[ f'(x_n - a \frac{f(x_n)}{f'(x_n)}) \approx f'(x_n) - a \frac{f(x_n)f''(x_n)}{f'(x_n)}; \]

therefore, we obtain

\[ f''(x_n) \approx \frac{f'(x_n)f''(x_n) - f'(y_n)}{af(x_n)}. \]
Using this approximate value of $f''(x_n)$ in method (5), we get
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[
x_{n+1} = x_n - a \left[ \frac{f(x_n)}{(a-1)f'(x_n) + f'(y_n)} + \frac{2f(x_n)}{(2a-1)f'(x_n) + f'(y_n)} \right]. \tag{8}
\]
This method has quadratic convergence and satisfies the following error equation:
\[
\epsilon_{n+1} = -\frac{c_2^2}{2} \epsilon_n^2 + \frac{1}{4} (6c_2^2 + (9a - 10)c_3) \epsilon_n^3 + O(\epsilon_n^4).
\]
But according to the Kung–Traub conjecture [11], method (8) is not an optimal method because it has second-order convergence and requires three functional evaluations per full iteration. Therefore, to build our optimal family of Jarratt’s method, we take six free disposable parameters (similar to Kim [8] to develop optimal eighth-order family of methods). Therefore, we consider
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[
x_{n+1} = x_n - a \left[ \frac{a_1 f(x_n)}{(a-1)f'(x_n) + a_2 f'(y_n)} + \frac{2a_1 f(x_n)}{(2a-1)a_2 f'(x_n) + a_3 f'(y_n)} \right], \tag{9}
\]
where $a_1, a_2, a_3, a_4$ and $a_5$ are disposable parameters such that the order of convergence reaches the optimal level four without using any more functional evaluations. Theorem 1 indicates under what choices on the disposable parameters in Equation (9) does the order of convergence reach the optimal level four.

**Theorem 1** Let a sufficiently smooth function $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ have a simple zero $r$ in the open interval $D$. Then, the family of iterative methods defined by Equation (9) has fourth-order convergence when
\[
a = \frac{2}{3},
\]
\[
a_1 = \frac{(a_4 - a_5)^3}{(a_4 + 3a_5)(a_4^2 + 2a_4a_5 + 3a_5^2)}, \quad a_4 \neq -5a_5,
\]
\[
a_2 = \frac{a_4 + a_5}{a_4 + 5a_5},
\]
\[
a_3 = \frac{(a_4 + 3a_5)^3}{4(a_4^2 + 2a_4a_5 + 3a_5^2)},
\]
where $a_4, a_5 \in \mathbb{R}$ but $a_4$ is equal to neither $a_5$ nor $-3a_5$ (otherwise these families of methods have third order of convergence) and satisfies the following error equation:
\[
\epsilon_{n+1} = \left( \frac{9(a_4^2 - 6a_4a_5 - 11a_5^2)c_1^3 + (a_4^3 + 2a_4a_5 - 3a_5^2)(c_1 - 9c_2c_3)}{9(a_4^2 + 2a_4a_5 - 3a_5^2)} \right) \epsilon_n^4 + O(\epsilon_n^5). \tag{11}
\]
where $\epsilon_n = x_n - r$ and $c_k = (1/k!)f^{(k)}(r)/f'(r)$, $k = 2, 3, \ldots$. 
Proof Let \( x = r \) be a simple zero of \( f(x) \). Expanding \( f(x_n) \) and \( f'(x_n) \) about \( x = r \) by Taylor’s series expansion, we have

\[
f(x_n) = f(r)(e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4) + O(e_n^5) \tag{12}
\]

and

\[
f'(x_n) = f'(r)(1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4) + O(e_n^5) \tag{13}
\]

respectively. From Equations (12) and (13), we have

\[
\frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 + (2c_2^2 - c_1)e_n^3 + (7c_2 c_3 - 4c_3^3 - 3c_4) e_n^4 + O(e_n^5). \tag{14}
\]

and in combination with Taylor’s series expansion of \( f'(x_n - a(f(x_n)/f'(x_n))) \) about \( x = r \), we have

\[
f'(y_n) = f'(x_n - a f(x_n)/f'(x_n)) = f'(r)[1 - 2(a - 1)c_2 e_n + (2a c_2^2 + 3(a - 1)^2 c_3) e_n^3
\]
\[+ (-4a c_2^3 + 2(5a - 3a^3)c_2 c_3 - 4(a - 1)^3 c_4) e_n^4] + O(e_n^5). \tag{15}
\]

Furthermore, we have

\[
\frac{a_2 f(x_n)}{(a - 1)f'(x_n) + a_2 f'(x_n)} = \frac{a_1}{(a + a_2 - 1)} e_n + \frac{a_1(1 - a + a_2(2a - 1))c_2}{(a + a_2 - 1)^2} e_n^2
\]
\[+ \frac{a_1}{(a + a_2 - 1)^3} [2((a - 1)^2 - a_2(4a^2 - 6a + 2))
\]
\[+ a_2^2(2a^2 - 4a + 1))c_2^2
\]
\[- (2(a - 1)^2 + a_2^2(3a^2 - 6a + 2)
\]
\[+ a_2(3a^3 - 9a^2 + 10a - 4))c_3] e_n^3 + O(e_n^4), \tag{16}
\]

and

\[
\frac{2a_2 f(x_n)}{(2a - 1)a_2 f'(x_n) + a_2 f'(y_n)} = \frac{2a_3}{(2a - 1)a_4 + a_5} e_n - \frac{2a_3(2a - 1)(a_4 - a_5)c_2}{((2a - 1)a_4 + a_5)^2} e_n^2
\]
\[+ \frac{2a_3}{((2a - 1)a_4 + a_5)^3} [2((2a - 1)^2(c_3 - c_2^2)a_2^2
\]
\[+ a_2^2((3a^2 - 6a + 2)c_3 - (4a^2 - 8a + 2)c_2^2)
\]
\[+ (2a - 1)a_4 a_5((8a - 4)c_3^2)
\]
\[+ (3a^3 - 6a + 4)c_3] e_n^3 + O(e_n^4), \tag{17}
\]

Using Equations (16) and (17) in scheme (9), we have the following error equation:

\[
e_{n+1} = \left( 1 - \frac{aa_1}{2(a + a_2 - 1)} - \frac{aa_3}{(2a - 1)a_4 + a_5} \right) e_n
\]
\[+ \frac{a}{2} \left( \frac{a_1(1 - a + a_2(2a - 1))}{(a + a_2 - 1)^2} - \frac{2a_3(2a - 1)(a_4 - a_5)}{((2a - 1)a_4 + a_5)^2} \right) c_2 e_n^2
\]
\[- \frac{a}{2} (2Ac_2^2 - Bc_3) e_n^3 + O(e_n^4), \tag{18}
\]

\[
\]
where

\[ A = \frac{a_1((a-1)^2 + a_2(2a^2 - 4a + 1) - a_2(4a^2 - 6a + 2))}{(a + a_2 - 1)^3} + \frac{2a_2((2a-1)^2a_2^2 - 2(2a-1)^2a_2a_5 + (2a^2 - 4a + 1)a_2^2)}{(2a - 1)a_4 + a_3)^3} \]

\[ B = \frac{a_1(2(a-1) + a_2(3a^2 - 6a + 2))}{(a + a_2 - 1)^2} + \frac{4(2a - 1)a_3a_4 + 2a_3a_5(3a^2 - 6a + 2)}{(2a - 1)a_4 + a_3)^2} \]

For obtaining an iterative method of order four, the coefficients of \(e_n\), \(e_n^2\) and \(e_n^3\) in the error equation (18) must be zero simultaneously. After simplifying Equation (18), we have the following equations involving \(a\), \(a_1\), \(a_2\), \(a_3\), \(a_4\) and \(a_5\):

\[ a_1 = \frac{2(a + a_2 - 1)}{a} \left( 1 - \frac{aa_3}{(2a - 1)a_4 + a_5)^2} \right), \]

\[ a_1(1 - a + a_2(2a - 1)) \left( \frac{2a_3(2a - 1)(a_4 - a_5)}{(2a - 1)a_4 + a_5)^2} \right), \]

\[ a_1((a-1)^2 + a_2^2(2a^2 - 4a + 1) - a_2(4a^2 - 6a + 2)) \left( (a + a_2 - 1)^3 \right), \]

\[ = \frac{-2a_3((2a - 1)^2a_2^2 - 2(2a - 1)^2a_2a_5 + (2a^2 - 4a + 1)a_2^2)}{(2a - 1)a_4 + a_5)^4} \]

\[ a_1(2(a - 1) + a_2(3a^2 - 6a + 2)) \left( (a + a_2 - 1)^2 \right), \]

\[ = \frac{-4a_3(2a - 1)a_4 + 2a_3a_5(3a^2 - 6a + 2)}{(2a - 1)a_4 + a_5)^2} \]

Solving the above equations for \(a\), \(a_1\), \(a_2\) and \(a_3\), we get

\[ a = \frac{2}{3}, \]

\[ a_1 = \frac{(a_4 - a_5)^3}{(a_4 + 5a_5)(a_4^2 + 2a_4a_5 + 5a_5^2)}, \quad a_4 \neq -5a_5, \]

\[ a_2 = \frac{a_4 + a_5}{a_4 + 5a_5}, \]

\[ a_3 = \frac{(a_4 + 3a_5)^3}{4(a_4^2 + 2a_4a_5 + 5a_5^2)}, \]

where \(a_4, a_5 \in \mathbb{R}\), but \(a_4\) is equal to neither \(-a_5\) nor \(-3a_5\).

Finally, we get the following new optimal families of Jarratt’s method given by

\[ y_n = x_n - \frac{2f(x_n)}{3f'(x_n)}, \]

\[ x_{n+1} = x_n - \frac{f(x_n)(a_2^2 - 22a_5a_5 - 27a_2^2f'(x_n) + 3(a_2^2 + 10a_5a_5 + 5a_2^2)f''(y_n))}{2(2a_2f'(x_n) + 3a_2f'(y_n))(3(a_4 + a_5)f'(y_n) - (a_4 + 5a_5)f''(x_n))}. \]

It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{9(a_2^2 - 6a_5a_5 - 11a_2^2)c_1^3 + (a_2^2 + 2a_5a_5 - 3a_2^2)k_1}{9(a_2^2 + 2a_5a_5 - 3a_2^2)k_1} \right) e_n^4 + O(e_n^5). \]

This reveals that the general two-step class of methods (21) reaches the optimal order of convergence four using only three functional evaluations per full iteration. Furthermore, the general
class of methods (21) has the same scaling factor of functions as that of Jarratt’s method. This is the beauty of this family. This completes the proof of Theorem 1.

3.1.1 Special cases of formula (21)

For different specific values of $a_4$ and $a_5$, the following various optimal multipoint methods can be derived from formula (21):

(i) For $a_5 = 0$ and $a_4 \neq 0$, family (21) reads as

\[
\begin{align*}
y_n &= x_n - \frac{2 f(x_n)}{3 f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{2 f'(x_n)} \left[ \frac{3 f'(y_n) + f'(x_n)}{3 f'(y_n) - f'(x_n)} \right].
\end{align*}
\]

This is a well-known Jarratt method. One can also get this method by substituting $a_5 = -a_4$, provided that the rest of the conditions of Theorem 1 are satisfied. It satisfies the following error equation:

\[ e_{n+1} = \left( c_2^3 - c_2 c_3 + \frac{c_4}{9} \right) e_n^4 + O(e_n^5). \]

(ii) For $a_4 = 0$ and $a_5 \neq 0$, family (21) reads as

\[
\begin{align*}
y_n &= x_n - \frac{2 f(x_n)}{3 f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{2 f'(y_n)} \left[ \frac{9 f'(x_n) - 5 f'(y_n)}{5 f'(x_n) - 3 f'(y_n)} \right].
\end{align*}
\]

This is another well-known fourth-order method given by Jarratt [3] and it satisfies the following error equation:

\[ e_{n+1} = \frac{1}{9} \left( 33 c_2^3 - 9 c_2 c_3 + c_4 \right) e_n^4 + O(e_n^5). \]

(iii) For $a_5 = 1$, family (21) reads as

\[
\begin{align*}
y_n &= x_n - \frac{2 f(x_n)}{3 f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)(a_4^2 - 22 a_4 - 27)f'(x_n) + 3(a_4^2 + 10 a_4 + 5)f'(y_n)}{2(a_4 f'(x_n) + 3 f'(y_n))(3(a_4 + 1)f'(y_n) - (a_4 + 5)f'(x_n))}.
\end{align*}
\]

This is a new optimal family of fourth-order methods. It satisfies the following error equation:

\[ e_{n+1} = \left( \frac{9(a_4^2 - 6 a_4 - 11)c_2^3 + (a_4^2 + 2 a_4 - 3)(c_4 - 9 c_2 c_3)}{9(a_4^2 + 2 a_4 - 3)} \right) e_n^4 + O(e_n^5). \]
3.1.2 Sub-case of family (24)

(a) For \(a_4 = 5\), family (24) reads as

\[
y_n = x_n - \frac{2}{3} f(x_n),
\]

\[
x_{n+1} = x_n - \frac{4f(x_n)(7f'(x_n) - 15f'(y_n))}{(5f'(x_n) - 9f'(y_n))(5f'(x_n) + 3f'(y_n))}.
\]

This is a new fourth-order optimal method and it satisfies the following error equation:

\[
\varepsilon_{n+1} = \left( -\frac{c_3^2}{3} - c_2 c_3 + \frac{c_4}{9} \right) \varepsilon_n^4 + O(\varepsilon_n^5).
\]

(iv) For \(a_4 = 1\), family (21) reads as

\[
y_n = x_n - \frac{2}{3} f(x_n),
\]

\[
x_{n+1} = x_n - \frac{f(x_n)(27a_5^2 + 22a_5 - 1)f'(x_n) - 3(5a_5^2 + 10a_5 + 1)f'(y_n))}{2(f'(x_n) + 3a_5f'(y_n))(5a_5 + 1)f'(x_n) - 3(a_5 + 1)f'(y_n))}.
\]

This is a new optimal family of fourth-order methods. It satisfies the following error equation:

\[
\varepsilon_{n+1} = \left( \frac{9(11a_5^2 + 6a_5 - 1)c_3^2 + (3a_5^2 - 2a_5 - 1)(c_4 - 9c_2 c_3)}{9(3a_5^2 - 2a_5 - 1)} \right) \varepsilon_n^4 + O(\varepsilon_n^5).
\]

3.1.3 Sub-cases of family (26)

(a) For \(a_5 = \frac{1}{3}\), family (26) reads as

\[
y_n = x_n - \frac{2}{3} f(x_n),
\]

\[
x_{n+1} = x_n - \frac{f(x_n)(7f'(x_n) - 11f'(y_n))}{2(2f'(x_n) - 3f'(y_n))(f'(x_n) + f'(y_n))}.
\]

This is a new fourth-order optimal method. It satisfies the following error equation:

\[
\varepsilon_{n+1} = \frac{1}{9} \left( \frac{15c_2^3 + 9c_2 c_3 - c_4}{15c_2^3 - c_4} \right) \varepsilon_n^4 + O(\varepsilon_n^5).
\]

(b) For \(a_5 = -2\), family (26) reads as

\[
y_n = x_n - \frac{2}{3} f(x_n),
\]

\[
x_{n+1} = x_n - \frac{f(x_n)f'(y_n) - 21f'(x_n))}{6f'^2(x_n) - 38f'(x_n)f'(y_n) + 12f'^2(y_n)}.
\]

This is again a new fourth-order optimal method. It satisfies the following error equation:

\[
\varepsilon_{n+1} = \left( \frac{31}{15} c_2^3 - c_2 c_3 + \frac{c_4}{9} \right) \varepsilon_n^4 + O(\varepsilon_n^5).
\]

Note that family (21) can produce many more new optimal multipoint families of Jarratt’s method for simple roots by fixing one of the disposable parameters, that is, either \(a_4\) or \(a_5\).
The beauty of these families is that they have the same scaling factor of functions as that of Jarratt’s method.

3.2 Case II: new optimal families of Ostrowski’s method

We now intend to develop new optimal families of Ostrowski’s method having the cubic scaling factor of functions in the correction factor. For this, we consider a well-known Newton method

\[ y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (29) \]

Now, we expand the function \( f(y_n) = f(x_n - (f(x_n)/f'(x_n))) \) about the point \( x = x_n \) by Taylor’s series expansion, and we have

\[ f \left( x_n - \frac{f(x_n)}{f'(x_n)} \right) \approx \frac{f^3(x_n)f''(x_n)}{2f'(x_n)^2}; \]

therefore, we obtain

\[ f''(x_n) \approx \frac{2f^2(x_n)f(y_n)}{f'(x_n)^2}. \quad (30) \]

Using this approximate value of \( f''(x_n) \) in method (5), we get

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

\[ x_{n+1} = x_n - \frac{1}{2} \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)}{f(x_n) - 2f(y_n)} + \frac{f(x_n)}{f(x_n) - f(y_n)} \right]. \quad (31) \]

This method has quadratic convergence and satisfies the following error equation:

\[ e_{n+1} = -\frac{c_3^2}{2} e_n^2 + \left( \frac{3}{2} c_2^2 - c_3 \right) e_n^3 + O(e_n^4). \]

Again according to the Kung–Traub conjecture [11], method (31) is not an optimal method because it has second-order convergence and requires three evaluations of function per full iteration. Therefore, to build our optimal families of Ostrowski’s method, we take five free disposable parameters. Therefore, we consider

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

\[ x_{n+1} = x_n - \frac{1}{2} \frac{f(x_n)}{f'(x_n)} \left[ \frac{b_1 f(x_n)}{b_2 f(x_n) - 2f(y_n)} + \frac{b_2 f(x_n)}{b_3 f(x_n) - b_4 f(y_n)} \right], \quad (32) \]

where \( b_1, b_2, b_3, b_4 \) and \( b_5 \) are disposable parameters such that the order of convergence reaches the optimal level four without using any more functional evaluations. Theorem 2 indicates under what conditions on these disposable parameters in Equation (32) does the order of convergence reach the optimal level four.
Theorem 2. Let a sufficiently smooth function $f : D \subseteq \mathbb{R} \to \mathbb{R}$ have a simple zero $r$ in the open interval $D$. Then, the family of iterative methods defined by Equation (32) has fourth-order convergence when

$$b_1 = \frac{4(b_4 - b_5)}{4b_4^2 - 6b_5^2 + 4b_3^2 - b_5^2},$$
$$b_2 = \frac{2(b_4 - b_5)}{2b_4 - b_5}, \quad b_5 \neq 2b_4,$$
$$b_3 = \frac{2b_4^2}{2b_4^2 - 2b_3b_5 + b_5^2},$$

where $b_4, b_5 \in \mathbb{R}$ but choose $b_4$ and $b_5$ such that neither $b_4 = 0$ nor $b_5 = b_5$.

Proof. The proof of this theorem is similar to the proof of Theorem 1. Hence, it is omitted here.

It is straightforward to see from Theorem 2 that by using some specific values of the disposable parameters $b_1, b_2, b_3, b_4$ and $b_5$, one can get the modified families of iterative methods. Using the above values in scheme (32), we get the following optimal families of methods given by

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{(b_4^2 + b_3b_5 - b_5^2)f(x_n)f(y_n) - b_3(b_4 - b_5)f^2(x_n)}{(b_4f(x_n) - b_5f(y_n))(2b_4 - b_5f(y_n)) - (b_4 - b_5)f(x_n)} \right].$$

It satisfies the following error equation:

$$e_{n+1} = \left( \frac{(b_4^2 - 3b_3b_5 + b_5^2)c_2^3 - b_3(b_4 - b_5)c_2^3}{b_4(b_4 - b_5)} \right) e_n^4 + O(e_n^5),$$

where $e_n = x_n - r$ and $c_k = (1/k!)(f^{(k)}(r))/f'(r)), k = 2, 3, \ldots$. 

3.2.1 Special cases of formula (34)

For different specific values of $b_4$ and $b_5$, the following various optimal multipoint methods can be derived from formula (34):

(i) For $b_5 = 0$ and $b_4 \neq 0$, family (34) reads as

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

This is a famous Ostrowski method and satisfies the following error equation:

$$e_{n+1} = (c_2^3 - c_2c_3)e_n^4 + O(e_n^5).$$
(ii) For $b_4 = 1$, family (34) reads as
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]
\[x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{(b_2^2 - b_3 - 1)f(x_n)f(y_n) + (1 - b_5)f^2(x_n)}{f'(x_n) - b_5 f(y_n))(b_5 - 2)f(y_n) + (1 - b_5)f(x_n))} \right]. \tag{37}
\]

This is a new fourth-order optimal family and satisfies the following error equation:
\[
e_{n+1} = \left( \frac{(b_2^2 - 3b_5 + 1)c_2^2 - (1 - b_5)c_2 c_3}{(1 - b_5)} \right) e_n^4 + O(e_n^5).
\]

3.2.2 Sub-cases of family (37)

(a) For $b_5 = \frac{1}{10}$, family (37) reads as
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}.
\]
\[x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)(90f(x_n) - 109f(y_n))}{90f^2(x_n) - 199f(x_n)f(y_n) + 192f(y_n)} \right]. \tag{38}
\]

This is a new fourth-order method and satisfies the following error equation:
\[
e_{n+1} = \left( \frac{71}{90} c_2^2 - c_2 c_3 \right) e_n^4 + O(e_n^5).
\]

(b) For $b_5 = -1$, family (37) reads as
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}.
\]
\[x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)(2f(x_n) + f(y_n))}{2f^2(x_n) - f(x_n)f(y_n) - 3f^2(y_n)} \right]. \tag{39}
\]

This is a new fourth-order method and satisfies the following error equation:
\[
e_{n+1} = \left( \frac{5}{2} c_2^2 - c_2 c_3 \right) e_n^4 + O(e_n^5).
\]

(iii) For $b_4 = 10$, family (34) reads as
\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}.
\]
\[x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{10(10 - b_3)f^2(x_n) + (b_2^2 - 10b_5 - 100)f(x_n)f(y_n)}{(b_5 f(y_n) - 10f(x_n))(b_5 - 10f(x_n) - (b_5 - 20)f(y_n))} \right]. \tag{40}
\]

This is a new fourth-order optimal family and satisfies the following error equation:
\[
e_{n+1} = \left( \frac{(b_2^2 - 30b_5 + 100)c_2^2 + 10(b_5 - 10)c_2 c_3}{10(10 - b_5)} \right) e_n^4 + O(e_n^5).
\]
3.2.3 Sub-case of family (40)

(a) For $b_5 = \frac{1}{10}$, family (40) reads as

\[
\begin{align*}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)(9900f(x_n) - 10099f(y_n))}{9900f^2(x_n) - 19999f(x_n)f(y_n) + 199f^2(y_n)} \right]. 
\end{align*}
\]

This is a new fourth-order method and satisfies the following error equation:

\[
e_{n+1} = \left( -\frac{9701}{9900} c_1 - c_2 c_3 \right) e_n^4 + O(e_n^5).
\]

(iv) For $b_5 = 1$, family (44) reads as

\[
\begin{align*}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{(b_4^2 + b_4 - 1)f(x_n)f(y_n) - b_4(b_4 - 1)f^2(x_n)}{(b_4f(x_n) - f(y_n))(2b_4 - 1)f(y_n) - (b_4 - 1)f(x_n))} \right]. 
\end{align*}
\]

This is a new fourth-order optimal family and satisfies the following error equation:

\[
e_{n+1} = \left( \frac{(b_4^2 - 3b_4 + 1)c_1^3 - b_4(b_4 - 1)c_2 c_3}{b_4(b_4 - 1)} \right) e_n^4 + O(e_n^5).
\]

3.2.4 Sub-cases of family (42)

(a) For $b_4 = \frac{49}{100}$, family (42) reads as

\[
\begin{align*}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)(2499f(x_n) - 2699f(y_n))}{2499f^2(x_n) - 5198f(x_n)f(y_n) + 200f^2(y_n)} \right]. 
\end{align*}
\]

This is a new fourth-order method and satisfies the following error equation:

\[
e_{n+1} = \left( \frac{2299}{2499} c_1^3 - c_2 c_3 \right) e_n^4 + O(e_n^5).
\]

(b) For $b_4 = 2$, family (42) reads as

\[
\begin{align*}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \left[ \frac{f(x_n)(2f(x_n) - 5f(y_n))}{2f^2(x_n) - 7f(x_n)f(y_n) + 3f^2(y_n)} \right]. 
\end{align*}
\]

This is a new fourth-order method and it satisfies the following error equation:

\[
e_{n+1} = -\left( \frac{c_1^3}{2} + c_2 c_3 \right) e_n^4 + O(e_n^5).
\]
Table 1. Test problems.

<table>
<thead>
<tr>
<th>No.</th>
<th>Problem</th>
<th>([a, b])</th>
<th>Root(r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(x^3 + 17x = 0)</td>
<td>([-0.5, 1])</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>(\tan^{-1}(x^2 - x) = 0)</td>
<td>([0.7, 1.6])</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>(e^{-x} + \cos x = 0)</td>
<td>([1, 2.5])</td>
<td>1.746139526367188</td>
</tr>
<tr>
<td>4.</td>
<td>(\log(x^2 + x + 2) - (x - 1) = 0)</td>
<td>([-1, 0])</td>
<td>-0.438204487909823</td>
</tr>
<tr>
<td>5.</td>
<td>(\sin x - \frac{1}{2} = 0)</td>
<td>([1.5, 2])</td>
<td>1.894942222640991</td>
</tr>
<tr>
<td>6.</td>
<td>(e^{x^2 + 3x - 1} = 0)</td>
<td>([2.9, 3.5])</td>
<td>3</td>
</tr>
<tr>
<td>7.</td>
<td>(x^3 - 10 = 0)</td>
<td>([2, 3])</td>
<td>2.15443680938721</td>
</tr>
<tr>
<td>8.</td>
<td>(\tan^{-1} x - x + 1 = 0)</td>
<td>([1.5, 3])</td>
<td>2.132267713546753</td>
</tr>
<tr>
<td>9.</td>
<td>(e^x - \cos x + 2 = 0)</td>
<td>([-2, -1])</td>
<td>-1.17257977180481</td>
</tr>
<tr>
<td>10.</td>
<td>(x^3 - x^2 + 11x - 7 = 0)</td>
<td>([0, 1])</td>
<td>0.645023941993713</td>
</tr>
</tbody>
</table>

Notes: JM, Jarratt’s method; OM, Ostrowski’s method.

Table 2. Total number of iterations to approximate the zero of a function and total number of function evaluations for various multipoint iterative methods.

<table>
<thead>
<tr>
<th>Problem</th>
<th>JM1</th>
<th>JM2</th>
<th>JM3</th>
<th>MOM1</th>
<th>MOM2</th>
<th>MOM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial guess</td>
<td>JM</td>
<td></td>
<td></td>
<td>JM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a_4 = 5) and (a_5 = 1) and (a_6 = 1)</td>
<td>(a_4 = 1) and (a_5 = \frac{1}{2}) and (a_6 = -2)</td>
<td>OM</td>
<td>(b_1 = 4) and (b_2 = 10) and (b_3 = \frac{67}{10})</td>
<td>(b_1 = \frac{57}{10}) and (b_2 = \frac{12}{11}) and (b_3 = 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>-0.5</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>-0.5</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>2.9</td>
<td>(6, 18)</td>
<td>(6, 18)</td>
<td>(6, 18)</td>
<td>(6, 18)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>(11, 33)</td>
<td>(11, 33)</td>
<td>(11, 33)</td>
<td>(11, 33)</td>
<td>(5, 15)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>-2</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(5, 15)</td>
<td>(3, 9)</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(4, 12)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(2, 6)</td>
<td>(2, 6)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
<td>(3, 9)</td>
</tr>
</tbody>
</table>

Notes: JM, Jarratt’s method; OM, Ostrowski’s method.

It is straightforward to see that per step these proposed methods require three functional evaluations, viz. two evaluations of \(f(x)\) and one of \(f'(x)\) or two evaluations of \(f(x)\) and one of \(f'(x)\). In order to obtain an assessment of the efficiency index of our proposed methods, we make use of the efficiency index [16]. For our proposed iteration schemes, we find \(p = 4\) and \(d = 3\), yielding \(E = \sqrt[4]{4} \approx 1.587\), which is better than those of most third-order methods \(E \approx 1.442\) and Newton’s method \(E \approx 1.414\). Therefore, these new families of methods reach the optimal order of convergence four.
4. Numerical experiments

In this section, we check the effectiveness of the new optimal methods. We employ the present methods (25), (27), (28), (38), (41) and (43) denoted by JM1, JM2, JM3, MOM1, MOM2 and MOM3, respectively, to solve nonlinear equations given in Table 1. We compare them with Jarratt’s method and Ostrowski’s method. We present the comparison of all the methods mentioned above in Table 2. Computations have been performed using C++ in double precision arithmetic. We use $\varepsilon = 10^{-15}$ as tolerable error. The following stopping criteria are used for computer programs:

(i) $|x_{n+1} - x_n| < \varepsilon$,
(ii) $|f(x_{n+1})| < \varepsilon$.

5. Conclusions

In this paper, we contribute further to the development of the theory of iteration processes and propose several families of Jarratt’s method and Ostrowski’s method. We obtained a wide general class of Jarratt’s families without memory and having the same scaling factor of functions as that of Jarratt’s method, Ostrowski’s method and King’s method. The proposed methods have been tested on a number of examples using different initial guesses. The results given in Table 2 overwhelmingly support that the members of the families are equally competent with the most efficient methods available in the literature. Each iteration of these methods requires one evaluation of the function and two evaluations of its first derivative or two evaluations of the function and one evaluation of its first-order derivative so that their efficiency indices are 1.587. However, the proposed methods are the variants of Newton’s method and do not work when $f'(x) = 0$.

Finally, we conclude that the methods presented in this paper are equally competitive with other recognized efficient methods, namely, Jarratt’s method, Ostrowski’s method and King’s methods. As future work, we are currently trying to develop the equi-scaling families of Ostrowski’s method for simple roots of nonlinear equations.

Acknowledgements

We express our gratitude to the Editor for his constructive suggestions and remarks which have considerably contributed to the readability of this paper. Ramandeep Behl further acknowledges CSIR, New Delhi, India, for the financial support.

References


