Research Article

An Efficient Family of Traub-Steffensen-Type Methods for Solving Systems of Nonlinear Equations

Janak Raj Sharma¹ and Puneet Gupta²

¹ Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab 148 106, India

² Department of Mathematics, Government Ranbir College, Sangrur, Punjab 148 001, India

Correspondence should be addressed to Janak Raj Sharma; jrshira@yahoo.co.in

Received 6 February 2014; Accepted 11 June 2014; Published 2 July 2014

Academic Editor: Zhangxin Chen

Copyright © 2014 J. R. Sharma and P. Gupta. 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.

Based on Traub-Steffensen method, we present a derivative free three-step family of sixth-order methods for solving systems of nonlinear equations. The local convergence order of the family is determined using first-order divided difference operator for functions of several variables and direct computation by Taylor's expansion. Computational efficiency is discussed, and a comparison between the efficiencies of the proposed techniques with the existing ones is made. Numerical tests are performed to compare the methods of the proposed family with the existing methods and to confirm the theoretical results. It is shown that the new family is especially efficient in solving large systems.

1. Introduction

The problem of finding solution of the system of nonlinear equations $F(x) = 0$, where $F : D \rightarrow D$, D is an open convex domain in R^n , by iterative methods is an important and challenging task in numerical analysis and many applied scientific branches. One of the basic procedures for solving nonlinear equations is the quadratically convergent Newton method (see [1, 2]):

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}), \quad k = 0, 1, 2, \dots, \quad (1)$$

where $[F'(x)]^{-1}$ is the inverse of the first Fréchet derivative $F'(x)$ of the function $F(x)$.

In many practical situations, it is preferable to avoid the calculation of derivative $F'(x)$ of the function $F(x)$. In such situations, it is preferable to use only the computed values of $F(x)$ and to approximate $F'(x)$ by employing the values of $F(x)$ at suitable points. For example, a basic derivative free iterative method is the Traub-Steffensen method [3], which also converges quadratically and follows the scheme

$$x^{(k+1)} = G_{1,2}(x^{(k)}) = x^{(k)} - [w^{(k)}, x^{(k)}; F]^{-1} F(x^{(k)}), \quad (2)$$

where $[w^{(k)}, x^{(k)}; F]^{-1}$ is the inverse of the first-order divided difference $[w^{(k)}, x^{(k)}; F]$ of F and $w^{(k)} = x^{(k)} + \gamma F(x^{(k)})$; γ is an arbitrary nonzero constant. Throughout this paper, $G_{i,p}$ is used to denote the i th iteration function of convergence order p . For $\gamma = 1$, the scheme (2) reduces to the well-known Steffensen method [4].

In recent years, many derivative free higher-order methods of great efficiency are developed for solving scalar equation $f(x) = 0$; see [5–14] and the references therein. For systems of nonlinear equations, however, the construction of efficient higher-order derivative free methods is a difficult task and therefore not many such methods can be found in the literature. Recently, based on Steffensen's scheme, that is, when $\gamma = 1$ in (2), a family of seventh-order methods has been proposed in [13]. Some important members of this family, as shown in [13], are given as follows:

$$\begin{aligned} y^{(k)} &= G_{1,2}(x^{(k)}), \\ z^{(k)} &= G_{1,4}(x^{(k)}, y^{(k)}) \\ &= y^{(k)} - ([y^{(k)}, x^{(k)}; F] + [y^{(k)}, w^{(k)}; F]) \end{aligned}$$

$$\begin{aligned}
& - [w^{(k)}, x^{(k)}; F]^{-1} F(y^{(k)}), \\
x^{(k+1)} &= G_{1,7}(x^{(k)}, y^{(k)}, z^{(k)}) \\
&= z^{(k)} - ([z^{(k)}, x^{(k)}; F] + [z^{(k)}, y^{(k)}; F] \\
&\quad - [y^{(k)}, x^{(k)}; F])^{-1} F(z^{(k)}), \tag{3} \\
y^{(k)} &= G_{1,2}(x^{(k)}), \\
z^{(k)} &= G_{2,4}(x^{(k)}, y^{(k)}) \\
&= y^{(k)} - [y^{(k)}, x^{(k)}; F]^{-1} \\
&\quad \times ([y^{(k)}, x^{(k)}; F] - [y^{(k)}, w^{(k)}; F] \\
&\quad + [w^{(k)}, x^{(k)}; F]) [y^{(k)}, x^{(k)}; F]^{-1} F(y^{(k)}), \tag{4} \\
x^{(k+1)} &= G_{2,7}(x^{(k)}, y^{(k)}, z^{(k)}) \\
&= z^{(k)} - ([z^{(k)}, x^{(k)}; F] + [z^{(k)}, y^{(k)}; F] \\
&\quad - [y^{(k)}, x^{(k)}; F])^{-1} F(z^{(k)}).
\end{aligned}$$

Per iteration, both methods use four functions, five first-order divided differences, and three matrix inversions. The notable feature of these algorithms is their simple design which makes them easily implemented to solve systems of nonlinear equations. Here, the fourth-order method $G_{1,4}(x^{(k)}, y^{(k)})$ is the generalization of the method proposed by Ren et al. in [5] and $G_{2,4}(x^{(k)}, y^{(k)})$ is the generalization of the method by Liu et al. [6].

In this paper, our aim is to develop derivative free iterative methods that may satisfy the basic requirements of generating quality numerical algorithms, that is, the algorithms with (i) high convergence speed, (ii) minimum computational cost, and (iii) simple design. In this way, we here propose a derivative free family of sixth-order methods. The scheme is composed of three steps of which the first two steps consist of any derivative free fourth-order method with the base as the Traub-Steffensen iteration (2) whereas the third step is weighted Traub-Steffensen iteration. The algorithm of the present contribution is as simple as the methods (3) and (4) but with an additional advantage that it possesses high computational efficiency, especially when applied for solving large systems of equations.

The rest of the paper is summarized as follows. The sixth-order scheme with its convergence analysis is presented in Section 2. In Section 3, the computational efficiency of new methods is discussed and is compared with the methods which lie in the same category. Various numerical examples are considered in Section 4 to show the consistent convergence behavior of the methods and to verify the theoretical results. Section 5 contains the concluding remarks.

2. The Method and Its Convergence

Based on the above considerations of a quality numerical algorithm, we begin with the following three-step iteration scheme:

$$\begin{aligned}
y^{(k)} &= G_{1,2}(x^{(k)}), \\
z^{(k)} &= G_4(x^{(k)}, y^{(k)}), \\
x^{(k+1)} &= G_6(x^{(k)}, y^{(k)}, z^{(k)}) \\
&= z^{(k)} - (aI + [w^{(k)}, x^{(k)}; F]^{-1} \\
&\quad \times (b[y^{(k)}, x^{(k)}; F] + c[y^{(k)}, w^{(k)}; F])) \\
&\quad \times [w^{(k)}, x^{(k)}; F]^{-1} F(z^{(k)}), \tag{5}
\end{aligned}$$

where $G_4(x^{(k)}, y^{(k)})$ denotes any derivative free fourth-order scheme and a, b, c are some parameters to be determined.

In order to find the convergence order of scheme (5), we first define divided difference operator for multivariable function F (see [15]). The divided difference operator of F is a mapping $[\cdot, \cdot; F] : D \times D \subset R^n \times R^n \rightarrow L(R^n)$ defined by

$$[x + h, x; F] = \int_0^1 F'(x + th) dt, \quad \forall x, h \in R^n. \tag{6}$$

Expanding $F'(x + th)$ in Taylor series at the point x and then integrating, we have

$$\begin{aligned}
[x + h, x; F] &= \int_0^1 F'(x + th) dt \\
&= F'(x) + \frac{1}{2} F''(x) h + \frac{1}{6} F'''(x) h^2 + O(h^3), \tag{7}
\end{aligned}$$

where $h^i = (h, h, \dots, h)$, $h \in R^n$.

Let $e^{(k)} = x^{(k)} - \alpha$. Assuming that $\Gamma = [F'(\alpha)]^{-1}$ exists and then developing $F(x^{(k)})$ and its first three derivatives in a neighborhood of α , we have

$$\begin{aligned}
F(x^{(k)}) &= F'(\alpha) (e^{(k)} + A_2(e^{(k)})^2 + A_3(e^{(k)})^3 \\
&\quad + A_4(e^{(k)})^4 + A_5(e^{(k)})^5 + O((e^{(k)})^6)), \tag{8}
\end{aligned}$$

$$\begin{aligned}
F'(x^{(k)}) &= F'(\alpha) (I + 2A_2e^{(k)} + 3A_3(e^{(k)})^2 + 4A_4(e^{(k)})^3 \\
&\quad + 5A_5(e^{(k)})^4 + O((e^{(k)})^5)), \tag{9}
\end{aligned}$$

$$F''(x^{(k)}) = F'(\alpha) \left(2A_2 + 6A_3e^{(k)} + 12A_4(e^{(k)})^2 + 20A_5(e^{(k)})^3 + O((e^{(k)})^4) \right), \quad (10)$$

$$F'''(x^{(k)}) = F'(\alpha) \left(6A_3 + 24A_4e^{(k)} + 60A_5(e^{(k)})^2 + O((e^{(k)})^3) \right), \quad (11)$$

where $A_i = (1/i!) \Gamma F^{(i)}(\alpha) \in L_i(R^n, R^n)$ and $(e^{(k)})^i = (e^{(k)}, e^{(k)}, \dots, e^{(k)}) \in R^n$. A_i are symmetric operators that are used later on.

We can now analyze behavior of the scheme (5) through the following theorem.

Theorem 1. *Let the function $F : D \subset R^n \rightarrow R^n$ be sufficiently differentiable in an open neighborhood D of its zero α and $G_4(x^{(k)}, y^{(k)})$ is a fourth-order iteration function which satisfies*

$$e_z^{(k)} = z^{(k)} - \alpha = B_0(e^{(k)})^4 + O((e^{(k)})^5), \quad (12)$$

where $B_0 \in L_4(R^n, R^n)$ and $e^{(k)} = x^{(k)} - \alpha$. If an initial approximation $x^{(0)}$ is sufficiently close to α , then the local order of convergence of method (5) is at least 6 provided $a = 3$, $b = -1$, and $c = -1$.

Proof. Let $e_w^{(k)} = w^{(k)} - \alpha = x^{(k)} + \gamma F(x^{(k)}) - \alpha$. Then, using (8), it follows that

$$e_w^{(k)} = (I + \gamma F'(\alpha)) e^{(k)} + \gamma F'(\alpha) A_2(e^{(k)})^2 + \gamma F'(\alpha) A_3(e^{(k)})^3 + O((e^{(k)})^4). \quad (13)$$

Employing (7) for $x + h = w^{(k)}$, $x = x^{(k)}$, and $h = e_w^{(k)} - e^{(k)}$ and then using (9)–(11), we write

$$\begin{aligned} [w^{(k)}, x^{(k)}; F] &= F'(\alpha) \\ &\times \left[I + A_2(e_w^{(k)} + e^{(k)}) \right. \\ &\quad + A_3((e_w^{(k)})^2 + (e^{(k)})^2 + e_w^{(k)}e^{(k)}) \\ &\quad + A_4((e_w^{(k)})^3 + (e^{(k)})^3 + (e_w^{(k)})^2e^{(k)} \\ &\quad \quad \quad + e_w^{(k)}(e^{(k)})^2) \\ &\quad \left. + O((e^{(k)})^4) \right]. \end{aligned} \quad (14)$$

Expanding in the formal power developments, the inverse of preceding operator can be written as

$$\begin{aligned} [w^{(k)}, x^{(k)}; F]^{-1} &= \left[I - A_2(e_w^{(k)} + e^{(k)}) + (A_2^2 - A_3) \right. \\ &\quad \times ((e_w^{(k)})^2 + (e^{(k)})^2) \\ &\quad + (2A_2^2 - A_3)e_w^{(k)}e^{(k)} \\ &\quad - (A_2^3 - A_3A_2 - A_2A_3 + A_4) \\ &\quad \times ((e_w^{(k)})^3 + (e^{(k)})^3) \\ &\quad - (3A_2^3 - 2A_3A_2 - 2A_2A_3 + A_4) \\ &\quad \times ((e_w^{(k)})^2e^{(k)} + e_w^{(k)}(e^{(k)})^2) \\ &\quad \left. + O((e^{(k)})^4) \right] \Gamma. \end{aligned} \quad (15)$$

Using (8) and (15) to the required terms in the first step of (5), we find that

$$\begin{aligned} e_y^{(k)} &= y^{(k)} - \alpha \\ &= A_2e_w^{(k)}e^{(k)} - (A_2^2 - A_3) \\ &\quad \times ((e_w^{(k)})^2e^{(k)} + e_w^{(k)}(e^{(k)})^2) + O((e^{(k)})^4). \end{aligned} \quad (16)$$

Equation (7) for $x + h = y^{(k)}$, $x = x^{(k)}$, and $h = e_y^{(k)} - e^{(k)}$ yields

$$\begin{aligned} [y^{(k)}, x^{(k)}; F] &= F'(\alpha) \left[I + A_2(e_y^{(k)} + e^{(k)}) \right. \\ &\quad + A_3(e_y^{(k)}e^{(k)} + (e^{(k)})^2) \\ &\quad \left. + A_4(e^{(k)})^3 + O((e^{(k)})^4) \right]. \end{aligned} \quad (17)$$

Similarly, substituting $x + h = y^{(k)}$, $x = w^{(k)}$, and $h = e_y^{(k)} - e_w^{(k)}$ in (7), we obtain

$$\begin{aligned} [y^{(k)}, w^{(k)}; F] &= F'(\alpha) \left[I + A_2(e_y^{(k)} + e_w^{(k)}) \right. \\ &\quad + A_3(e_y^{(k)}e_w^{(k)} + (e_w^{(k)})^2) \\ &\quad \left. + A_4(e_w^{(k)})^3 + O((e^{(k)})^4) \right]. \end{aligned} \quad (18)$$

Then, using (15), (17), and (18) to the required terms, we find that

$$\begin{aligned}
 aI + [w^{(k)}, x^{(k)}; F]^{-1} & \left(b[y^{(k)}, x^{(k)}; F] + c[y^{(k)}, w^{(k)}; F] \right) \\
 &= (a+b+c)I - A_2(b e_w^{(k)} + c e_y^{(k)}) \\
 &+ (A_2^2 - A_3)(b(e_w^{(k)})^2 + c(e^{(k)})^2) \\
 &+ (b+c)(A_2^2 - A_3)e_w^{(k)}e^{(k)} + (b+c)A_2e_y^{(k)} \\
 &+ O((e^{(k)})^3). \tag{19}
 \end{aligned}$$

Postmultiplying (19) by $[w^{(k)}, x^{(k)}; F]^{-1}$ and simplifying

$$\begin{aligned}
 \theta &= \left(aI + [w^{(k)}, x^{(k)}; F]^{-1} \right. \\
 &\quad \times \left(b[y^{(k)}, x^{(k)}; F] + c[y^{(k)}, w^{(k)}; F] \right) \\
 &\quad \times [w^{(k)}, x^{(k)}; F]^{-1} \\
 &= \left[(a+b+c)I - (a+2b+c)A_2e_w^{(k)} \right. \\
 &\quad - (a+b+2c)A_2e^{(k)} \\
 &\quad + ((a+3b+c)A_2^2 - (a+2b+c)A_3)(e_w^{(k)})^2 \\
 &\quad + ((a+b+3c)A_2^2 - (a+b+2c)A_3)(e^{(k)})^2 \\
 &\quad + (a+2(b+c))(2A_2^2 - A_3)e_w^{(k)}e^{(k)} \\
 &\quad \left. + (b+c)A_2e_y^{(k)} + O((e^{(k)})^3) \right] \Gamma, \tag{20}
 \end{aligned}$$

Taylor series of $F(z^{(k)})$ about α yields

$$F(z^{(k)}) = F'(\alpha) \left[e_z^{(k)} + A_2(e_z^{(k)})^2 + O((e_z^{(k)})^3) \right]. \tag{21}$$

Then, using (20) and (21) in the third step of (5), we obtain the error equation as

$$\begin{aligned}
 e^{(k+1)} &= e_z^{(k)} - \theta F'(\alpha) \left[e_z^{(k)} + O((e_z^{(k)})^2) \right] \\
 &= -(a+b+c-1)e_z^{(k)} + (a+2b+c)A_2e_w^{(k)}e_z^{(k)} \\
 &\quad + (a+b+2c)A_2e_z^{(k)}e^{(k)} \\
 &\quad - ((a+3b+c)A_2^2 - (a+2b+c)A_3)(e_w^{(k)})^2 e_z^{(k)}
 \end{aligned}$$

$$\begin{aligned}
 &- ((a+b+3c)A_2^2 - (a+b+2c)A_3) \\
 &\times e_z^{(k)}(e^{(k)})^2 - (a+2(b+c)) \\
 &\times (2A_2^2 - A_3)e_w^{(k)}e_z^{(k)}e^{(k)} \\
 &- (b+c)A_2e_y^{(k)}e_z^{(k)} + O((e^{(k)})^7). \tag{22}
 \end{aligned}$$

In order to find a , b , and c , it will be sufficient to equate the factors $a+b+c-1$, $a+2b+c$, and $a+b+2c$ to 0 and then solving the resulting system of equations

$$a+b+c=1, \quad a+2b+c=0, \quad a+b+2c=0, \tag{23}$$

we obtain $a=3$, $b=-1$, and $c=-1$.

Thus, for this set of values, the above error equation reduces to

$$\begin{aligned}
 e^{(k+1)} &= A_2^2(e_w^{(k)} + e^{(k)})^2 e_z^{(k)} + 2A_2e_y^{(k)}e_z^{(k)} \\
 &\quad - A_3e_w^{(k)}e_z^{(k)}e^{(k)} + O((e^{(k)})^7) \\
 &= B_0((I + \gamma F'(\alpha))(6A_2^2 - A_3) \\
 &\quad + \gamma^2(F'(\alpha))^2 A_2^2)(e^{(k)})^6 + O((e^{(k)})^7), \tag{24}
 \end{aligned}$$

which shows the sixth order of convergence and hence the result follows. $\square$

Finally, the sixth-order family of methods is expressed by

$$\begin{aligned}
 y^{(k)} &= G_{1,2}(x^{(k)}), \\
 z^{(k)} &= G_4(x^{(k)}, y^{(k)}), \\
 x^{(k+1)} &= G_6(x^{(k)}, y^{(k)}, z^{(k)}) \\
 &= z^{(k)} - M(x^{(k)}, w^{(k)}, y^{(k)})F(z^{(k)}), \tag{25}
 \end{aligned}$$

wherein

$$\begin{aligned}
 M(x^{(k)}, w^{(k)}, y^{(k)}) &= \left(3I - [w^{(k)}, x^{(k)}; F]^{-1} \left([y^{(k)}, x^{(k)}; F] \right. \right. \\
 &\quad \left. \left. + [y^{(k)}, w^{(k)}; F] \right) \right) [w^{(k)}, x^{(k)}; F]^{-1}. \tag{26}
 \end{aligned}$$

Thus, the scheme (25) defines a new three-step family of derivative free sixth-order methods with the first two steps as any fourth-order scheme whose base is the Traub-Steffensen method (3). Some simple members of this family are as follows.

Method-I. The first method, which is denoted by $G_{1,6}$, is given by

$$\begin{aligned}
 y^{(k)} &= G_{1,2}(x^{(k)}), \\
 z^{(k)} &= G_{1,4}(x^{(k)}, y^{(k)}), \\
 x^{(k+1)} &= z^{(k)} - M(x^{(k)}, w^{(k)}, y^{(k)})F(z^{(k)}), \tag{27}
 \end{aligned}$$

where $G_{1,4}(x^{(k)}, y^{(k)})$ is the fourth-order method as given in the formula (3). It is clear that this formula uses four functions, three first-order divided differences, and two matrix inversions per iteration.

Method-II. The second method, that we denote by $G_{2,6}$, is given as

$$\begin{aligned} y^{(k)} &= G_{1,2}(x^{(k)}), \\ z^{(k)} &= G_{2,4}(x^{(k)}, y^{(k)}), \\ x^{(k+1)} &= z^{(k)} - M(x^{(k)}, w^{(k)}, y^{(k)}) F(z^{(k)}), \end{aligned} \quad (28)$$

where $G_{2,4}(x^{(k)}, y^{(k)})$ is the fourth-order method shown in (4). This method also requires the same evaluations as in the above method.

3. Computational Efficiency

Here, we estimate the computational efficiency of the proposed methods and compare it with the existing methods. To do this, we will make use of efficiency index, according to which the efficiency of an iterative method is given by $E = \rho^{1/C}$, where ρ is the order of convergence and C is the computational cost per iteration. For a system of n nonlinear equations in n unknowns, the computational cost per iteration is given by (see [16])

$$C(\mu, n, l) = P_0(n)\mu + P(n, l). \quad (29)$$

Here, $P_0(n)$ denotes the number of evaluations of scalar functions used in the evaluation of F and $[x, y; F]$, and $P(n, l)$ denotes the number of products needed per iteration. The divided difference $[x, y; F]$ of F is an $n \times n$ matrix with elements given as (see [17, 18])

$$\begin{aligned} [y, x; F]_{ij} &= (f_i(y_1, \dots, y_{j-1}, y_j, x_{j+1}, \dots, x_n) \\ &\quad - f_i(y_1, \dots, y_{j-1}, x_j, x_{j+1}, \dots, x_n) \\ &\quad + f_i(x_1, \dots, x_{j-1}, y_j, y_{j+1}, \dots, y_n) \\ &\quad - f_i(x_1, \dots, x_{j-1}, x_j, y_{j+1}, \dots, y_n)) \\ &\quad \times ((2(y_j - x_j)))^{-1}, \quad 1 \leq i, j \leq n. \end{aligned} \quad (30)$$

In order to express the value of $C(\mu, n, l)$ in terms of products, a ratio $\mu > 0$ between products and evaluations of scalar functions and a ratio $l \geq 1$ between products and quotients are required.

To compute F in any iterative function, we evaluate n scalar functions $(f_1, f_2, \dots, f_n)$, and if we compute a divided difference $[x, y; F]$, then we evaluate $2n(n-1)$ scalar functions, where $F(x)$ and $F(y)$ are computed separately. We must add n^2 quotients from any divided difference, n^2 products for multiplication of a matrix with a vector or of a matrix by a scalar, and n products for multiplication of a vector by a scalar. In order to compute an inverse linear

operator, we solve a linear system, where we have $n(n-1)(2n-1)/6$ products and $n(n-1)/2$ quotients in the LU decomposition and $n(n-1)$ products and n quotients in the resolution of two triangular linear systems.

The computational efficiency of the present sixth-order methods $G_{1,6}$ and $G_{2,6}$ is compared with the existing fourth-order methods $G_{1,4}$ and $G_{2,4}$ and with the seventh-order methods $G_{1,7}$ and $G_{2,7}$. In addition, we also compare the present methods with each other. Let us denote efficiency indices of $G_{i,p}$ by $E_{i,p}$ and computational cost by $C_{i,p}$. Then, taking into account the above and previous considerations, we have

$$C_{1,4} = (6n^2 - 3n)\mu + \frac{n}{3}(2n^2 + 3n - 5 + 3l(4n + 1)), \quad (31)$$

$$E_{1,4} = 4^{1/C_{1,4}},$$

$$C_{2,4} = (6n^2 - 3n)\mu + \frac{n}{3}(2n^2 + 9n - 8 + 3l(4n + 2)), \quad (32)$$

$$E_{2,4} = 4^{1/C_{2,4}},$$

$$C_{1,6} = (6n^2 - 2n)\mu + \frac{n}{3}(2n^2 + 12n - 8 + 3l(4n + 3)),$$

$$E_{1,6} = 6^{1/C_{1,6}}, \quad (33)$$

$$C_{2,6} = (6n^2 - 2n)\mu + \frac{n}{3}(2n^2 + 18n - 11 + 12l(n + 1)),$$

$$E_{2,6} = 6^{1/C_{2,6}}, \quad (34)$$

$$C_{1,7} = (10n^2 - 6n)\mu + \frac{n}{2}(2n^2 + 3n - 5 + l(13n + 3)),$$

$$E_{1,7} = 7^{1/C_{1,7}}, \quad (35)$$

$$C_{2,7} = (10n^2 - 6n)\mu + \frac{n}{2}(2n^2 + 7n - 7 + l(13n + 5)),$$

$$E_{2,7} = 7^{1/C_{2,7}}. \quad (36)$$

3.1. Comparison between Efficiencies. To compare the computational efficiencies of the iterative methods, say $G_{i,p}$ against $G_{j,q}$, we consider the ratio

$$R_{i,p;j,q} = \frac{\log E_{i,p}}{\log E_{j,q}} = \frac{C_{j,q} \log(p)}{C_{i,p} \log(q)}. \quad (37)$$

It is clear that if $R_{i,p;j,q} > 1$, the iterative method $G_{i,p}$ is more efficient than $G_{j,q}$.

$G_{1,4}$ versus $G_{1,6}$ Case. For this case, the ratio (37) is given by

$$R_{1,6;1,4} = \frac{\log 6}{\log 4} \frac{2n^2 + n(18\mu + 12l + 3) - 9\mu + 3l - 5}{2n^2 + n(18\mu + 12l + 12) - 6\mu + 9l - 8}, \quad (38)$$

which shows that $R_{1,6;1,4} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 9$. Thus, we have $E_{1,6} > E_{1,4}$ for $\mu > 0$, $l \geq 1$, and $n \geq 9$.

$G_{2,4}$ versus $G_{1,6}$ Case. In this case, the ratio (37) takes the following form

$$R_{1,6;2,4} = \frac{\log 6}{\log 4} \frac{2n^2 + n(18\mu + 12l + 9) - 9\mu + 6l - 8}{2n^2 + n(18\mu + 12l + 12) - 6\mu + 9l - 8}. \quad (39)$$

In this case, it is easy to prove that $R_{1,6;2,4} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 2$, which implies that $E_{1,6} > E_{2,4}$.

$G_{1,4}$ versus $G_{2,6}$ Case. The ratio (37) yields

$$R_{2,6;1,4} = \frac{\log 6}{\log 4} \frac{2n^2 + n(18\mu + 12l + 3) - 9\mu + 3l - 5}{2n^2 + n(18\mu + 12l + 18) - 6\mu + 12l - 11}, \quad (40)$$

which shows that $R_{2,6;1,4} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 19$. Thus, we conclude that $E_{2,6} > E_{1,4}$ for $\mu > 0$, $l \geq 1$, and $n \geq 19$.

$G_{2,4}$ versus $G_{2,6}$ Case. In this case, the ratio (37) is given by

$$R_{2,6;2,4} = \frac{\log 6}{\log 4} \frac{2n^2 + n(18\mu + 12l + 9) - 9\mu + 6l - 8}{2n^2 + n(18\mu + 12l + 18) - 6\mu + 12l - 11}. \quad (41)$$

With the same range of μ, l as in the previous case and $n \geq 6$, the ratio $R_{2,6;2,4} > 1$, which implies that $E_{2,6} > E_{2,4}$.

$G_{1,6}$ versus $G_{2,6}$ Case. In this case, it is enough to compare the corresponding values of $C_{1,6}$ and $C_{2,6}$ from (33) and (34). Thus, we find that $E_{1,6} > E_{2,6}$ for $\mu > 0$, $l \geq 1$, and $n \geq 2$.

$G_{1,6}$ versus $G_{1,7}$ Case. In this case, the ratio (37) is given by

$$R_{1,6;1,7} = \frac{\log 6}{\log 7} \frac{3(2n^2 + n(20\mu + 13l + 3) - 12\mu + 3l - 5)}{2(2n^2 + n(18\mu + 12l + 12) - 6\mu + 9l - 8)}. \quad (42)$$

It is easy to show that $R_{1,6;1,7} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 4$, which implies that $E_{1,6} > E_{1,7}$ for this range of values of the parameters (μ, n, l) .

$G_{1,6}$ versus $G_{2,7}$ Case. The ratio (37) is given by

$$R_{1,6;2,7} = \frac{\log 6}{\log 7} \frac{3(2n^2 + n(20\mu + 13l + 7) - 12\mu + 5l - 7)}{2(2n^2 + n(18\mu + 12l + 12) - 6\mu + 9l - 8)}. \quad (43)$$

With the same range of μ, l as in the previous cases and $n \geq 2$, we have $R_{1,6;2,7} > 1$, which implies that $E_{1,6} > E_{2,7}$.

$G_{2,6}$ versus $G_{1,7}$ Case. The ratio (37) yields

$$R_{2,6;1,7} = \frac{\log 6}{\log 7} \frac{3(2n^2 + n(20\mu + 13l + 3) - 12\mu + 3l - 5)}{2(2n^2 + n(18\mu + 12l + 18) - 6\mu + 12l - 11)}, \quad (44)$$

which shows that $R_{2,6;1,7} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 11$, so it follows that $E_{2,6} > E_{1,7}$.

$G_{2,6}$ versus $G_{2,7}$ Case. For this case, the ratio (37) is given by

$$R_{2,6;2,7} = \frac{\log 6}{\log 7} \frac{3(2n^2 + n(20\mu + 13l + 7) - 12\mu + 5l - 7)}{2(2n^2 + n(18\mu + 12l + 18) - 6\mu + 12l - 11)}. \quad (45)$$

In this case, also it is not difficult to prove that $R_{2,6;2,7} > 1$ for $\mu > 0$, $l \geq 1$, and $n \geq 5$, which implies that $E_{2,6} > E_{2,7}$.

We summarize the above results in the following theorem.

Theorem 2. For $\mu > 0$ and $l \geq 1$, we have the following:

- (i) $E_{1,6} > E_{1,4}$ for $n \geq 9$;
- (ii) $E_{2,6} > E_{1,4}$ for $n \geq 19$;
- (iii) $E_{2,6} > E_{2,4}$ for $n \geq 6$;
- (iv) $E_{1,6} > E_{1,7}$ for $n \geq 4$;
- (v) $E_{2,6} > E_{1,7}$ for $n \geq 11$;
- (vi) $E_{2,6} > E_{2,7}$ for $n \geq 5$;
- (vii) $\{E_{1,6} > E_{2,4}, E_{1,6} > E_{2,6}, E_{1,6} > E_{2,7}\}$ for $n \geq 2$.

Otherwise, the comparison depends on μ, l , and n .

4. Numerical Results

In this section, some numerical problems are considered to illustrate the convergence behavior and computational efficiency of the proposed methods. The performance is compared with the existing methods $G_{1,4}$, $G_{2,4}$, $G_{1,7}$, and $G_{2,7}$. All computations are performed using the programming package *Mathematica* [19] using multiple-precision arithmetic with 4096 digits. For every method, we analyze the number of iterations (k) needed to converge to the solution such that $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-200}$. In numerical results, we also include the CPU time utilized in the execution of program which is computed by the *Mathematica* command "TimeUsed[]". In order to verify the theoretical order of convergence, we calculate the computational order of convergence (ρ_c) using the formula

$$\rho_c = \frac{\log(\|F(x^{(k)})\| / \|F(x^{(k-1)})\|)}{\log(\|F(x^{(k-1)})\| / \|F(x^{(k-2)})\|)}, \quad (46)$$

(see [20]) taking into consideration the last three approximations in the iterative process.

To connect the analysis of computational efficiency with numerical examples, the definition of the computational cost (29) is applied, according to which an estimation of the factor μ is claimed. For this, we express the cost of the evaluation of the elementary functions in terms of products, which depends on the computer, the software, and the arithmetics used (see [21, 22]). In Table 1, the elapsed CPU time (measured in milliseconds) in the computation of elementary functions and an estimation of the cost of the elementary functions in product units are displayed. The programs are

TABLE 1: CPU time and estimation of computational cost of the elementary functions, where $x = \sqrt{3} - 1$ and $y = \sqrt{5}$.

Functions	$x * y$	x/y	$\sqrt{x}$	e^x	$\ln x$	$\sin x$	$\cos x$	$\cos^{-1}x$	$\tan^{-1}x$	x^y
CPU time	0.0466	0.1305	0.0606	3.7746	3.6348	4.7532	4.7534	7.7356	7.5492	8.2948
Cost	1	2.8	1.3	81	78	102	102	166	162	178

performed in the processor, Intel (R) Core (TM) i5-480M CPU @ 2.67 GHz (64-bit Machine) Microsoft Windows 7 Home Basic 2009, and are compiled by *Mathematica* 7.0 using multiple-precision arithmetic. It can be observed from Table 1 that, for this hardware and the software, the computational cost of quotient with respect to product is $l = 2.8$.

The present methods $G_{1,6}$ and $G_{2,6}$ are tested by using the values $-0.01, 0.01$, and 0.5 for the parameter γ . The following problems are chosen for numerical tests.

Problem 1. Considering the system of two equations,

$$\begin{aligned} x_1^2 - x_2 + 1 &= 0, \\ x_1 - \cos\left(\frac{\pi x_2}{2}\right) &= 0. \end{aligned} \quad (47)$$

In this problem, $(n, \mu) = (2, 52)$ are the values used in (31)–(36) for calculating computational costs and efficiency indices of the considered methods. The initial approximation chosen is $x^{(0)} = \{0.25, 0.5\}^T$ and the solution is $\alpha = \{0, 1\}^T$.

Problem 2. Consider the system of three equations:

$$\begin{aligned} 10x_1 + \sin(x_1 + x_2) - 1 &= 0, \\ 8x_2 - \cos^2(x_3 - x_2) - 1 &= 0, \\ 12x_3 + \sin x_3 - 1 &= 0, \end{aligned} \quad (48)$$

with initial value $x^{(0)} = \{0.8, 0.5, 0.125\}^T$ towards the solution

$$\alpha = \{0.0689783491726666 \dots, 0.2464424186091830 \dots, 0.0769289119875370 \dots\}^T. \quad (49)$$

For this problem, $(n, \mu) = (3, 103.33)$ are used in (31)–(36) to calculate computational costs and efficiency indices.

Problem 3. Next, consider the following boundary value problem (see [23]):

$$y'' + y^3 = 0, \quad y(0) = 0, \quad y(1) = 1. \quad (50)$$

Assume the following partitioning of the interval $[0, 1]$:

$$\begin{aligned} u_0 = 0 < u_1 < u_2 < \dots < u_{m-1} < u_m = 1, \\ u_{j+1} = u_j + h, \quad h = \frac{1}{m}. \end{aligned} \quad (51)$$

Let us define $y_0 = y(u_0) = 0, y_1 = y(u_1), \dots, y_{m-1} = y(u_{m-1}), y_m = y(u_m) = 1$. If we discretize the problem by using the numerical formula for second derivative,

$$y_k'' = \frac{y_{k-1} - 2y_k + y_{k+1}}{h^2}, \quad k = 1, 2, 3, \dots, m-1, \quad (52)$$

we obtain a system of $m - 1$ nonlinear equations in $m - 1$ variables:

$$y_{k-1} - 2y_k + y_{k+1} + h^2 y_k^3 = 0, \quad k = 1, 2, 3, \dots, m-1. \quad (53)$$

In particular, we solve this problem for $m = 5$ so that $n = 4$ by selecting $y^{(0)} = \{0.5, 0.5, 0.5, 0.5\}^T$ as the initial value. The solution of this problem is

$$\begin{aligned} \alpha = \{0.21054188948074775 \dots, 0.42071046387616439 \dots, \\ 0.62790045371805633 \dots, \\ 0.82518822786851363 \dots\}^T \end{aligned} \quad (54)$$

and concrete values of the parameters are $(n, \mu) = (4, 4)$.

Problem 4. Consider the system of fifteen equations (see [16]):

$$\sum_{j=1, j \neq i}^{15} x_j - e^{-x_i} = 0, \quad 1 \leq i \leq 15. \quad (55)$$

In this problem, the concrete values of the parameters (n, μ) are $(15, 81)$. The initial approximation assumed is $x^{(0)} = \{1, 1, \dots, 1\}^T$ and the solution of this problem is

$$\begin{aligned} \alpha = \{0.066812203179582582 \dots, \\ 0.066812203179582582 \dots, \\ \dots, \\ 0.066812203179582582 \dots\}^T. \end{aligned} \quad (56)$$

Problem 5. Consider the system of fifty equations:

$$\begin{aligned} x_i^2 x_{i+1} - 1 &= 0, \quad (i = 1, 2, \dots, 49) \\ x_{50}^2 x_1 - 1 &= 0. \end{aligned} \quad (57)$$

In this problem, $(n, \mu) = (50, 2)$ are the values used in (31)–(36) for calculating computational costs and efficiency indices. The initial approximation assumed is $x^{(0)} = \{1.5, 1.5, 1.5, \dots, 1.5\}^T$ for obtaining the solution $\alpha = \{1, 1, 1, \dots, 1\}^T$.

Problem 6. Lastly, consider the nonlinear and nondifferentiable integral equation of mixed Hammerstein type (see [24]):

$$x(s) = 1 + \frac{1}{2} \int_0^1 G(s, t) (|x(t)| + x(t)^2) dt, \quad (58)$$

TABLE 2: Comparison of the performance of methods.

Methods	k	ρ_c	$C_{i,p}$	$E_{i,p}$	e-time
Problem 1					
$G_{1,4}$	7	4.000013	992.4	1.0013979	0.24518
$G_{2,4}$	7	4.000004	1004.0	1.0013817	0.24754
$G_{1,6}(\gamma = -.01)$	5	5.999999	1117.6	1.0016045	0.19709
$G_{1,6}(\gamma = .01)$	5	5.999999	1117.6	1.0016045	0.19854
$G_{1,6}(\gamma = .5)$	5	6.000000	1117.6	1.0016045	0.19773
$G_{2,6}(\gamma = -.01)$	5	6.000000	1129.2	1.0015880	0.21427
$G_{2,6}(\gamma = .01)$	5	6.000000	1129.2	1.0015880	0.21564
$G_{2,6}(\gamma = .5)$	5	6.000001	1129.2	1.0015880	0.21845
$G_{1,7}$	5	7.120115	1546.2	1.0012593	0.24891
$G_{2,7}$	5	7.078494	1557.8	1.0012499	0.25246
Problem 2					
$G_{1,4}$	5	4.000000	4781.05	1.0002899	1.31891
$G_{2,4}$	5	4.000000	4804.45	1.0002886	1.32373
$G_{1,6}(\gamma = -.01)$	4	6.000000	5131.84	1.0003492	1.10755
$G_{1,6}(\gamma = .01)$	4	6.000000	5131.84	1.0003492	1.11075
$G_{1,6}(\gamma = .5)$	4	5.999930	5131.84	1.0003492	1.11752
$G_{2,6}(\gamma = -.01)$	4	6.000000	5155.24	1.0003476	1.11191
$G_{2,6}(\gamma = .01)$	4	6.000000	5155.24	1.0003476	1.11618
$G_{2,6}(\gamma = .5)$	4	5.999906	5155.24	1.0003476	1.11455
$G_{1,7}$	4	7.000000	7649.20	1.0002544	1.69473
$G_{2,7}$	4	7.000000	7672.60	1.0002537	1.70182
Problem 3					
$G_{1,4}$	5	4.000123	578.4	1.0023996	0.17903
$G_{2,4}$	5	4.000035	617.6	1.0022472	0.18164
$G_{1,6}(\gamma = -.01)$	4	5.999970	660.8	1.0027152	0.13473
$G_{1,6}(\gamma = .01)$	4	5.999968	660.8	1.0027152	0.12900
$G_{1,6}(\gamma = .5)$	4	5.999887	660.8	1.0027152	0.12627
$G_{2,6}(\gamma = -.01)$	4	5.999957	700.0	1.0025629	0.14182
$G_{2,6}(\gamma = .01)$	4	5.999955	700.0	1.0025629	0.14082
$G_{2,6}(\gamma = .5)$	4	5.999870	700.0	1.0025629	0.14464
$G_{1,7}$	4	6.999331	930.0	1.0020946	0.19327
$G_{2,7}$	4	6.999216	969.2	1.0020098	0.19891
Problem 4					
$G_{1,4}$	5	4.000000	110717.0	1.00001252	6.67681
$G_{2,4}$	5	4.000000	111194.0	1.00001246	7.69175
$G_{1,6}(\gamma = -.01)$	4	6.000000	112676.0	1.00001590	6.16291
$G_{1,6}(\gamma = .01)$	4	6.000000	112676.0	1.00001590	5.74189
$G_{1,6}(\gamma = .5)$	4	6.000000	112676.0	1.00001590	6.03452
$G_{2,6}(\gamma = -.01)$	4	6.000000	113153.0	1.00001584	6.53969
$G_{2,6}(\gamma = .01)$	4	6.000000	113153.0	1.00001584	6.62572
$G_{2,6}(\gamma = .5)$	4	6.000000	113153.0	1.00001584	6.72312
$G_{1,7}$	4	7.000000	182793.0	1.00001065	7.29228
$G_{2,7}$	4	7.000000	183270.0	1.00001062	7.95691
Problem 5					
$G_{1,4}$	7	4.000000	143590.0	1.0000097	8.12436
$G_{2,4}$	7	4.000000	148680.0	1.0000093	8.32287
$G_{1,6}(\gamma = -.01)$	5	6.000000	151420.0	1.0000118	6.08475

TABLE 2: Continued.

Methods	k	ρ_c	$C_{i,p}$	$E_{i,p}$	e-time
$G_{1,6}(\gamma = .01)$	5	6.000000	151420.0	1.0000118	5.99152
$G_{1,6}(\gamma = .5)$	5	6.000000	151420.0	1.0000118	5.89734
$G_{2,6}(\gamma = -.01)$	5	6.000000	156510.0	1.0000114	6.61833
$G_{2,6}(\gamma = .01)$	5	6.000000	156510.0	1.0000114	6.45834
$G_{2,6}(\gamma = .5)$	5	6.000000	156510.0	1.0000114	6.66583
$G_{1,7}$	5	7.000000	223735.0	1.0000087	7.84567
$G_{2,7}$	5	7.000000	228825.0	1.0000085	7.89437
Problem 6					
$G_{1,4}$	5	4.000000	5091.200	1.0002723	2.02945
$G_{2,4}$	5	4.000000	5233.600	1.0002649	2.13573
$G_{1,6}(\gamma = -.01)$	4	6.000000	5408.000	1.0003314	1.96338
$G_{1,6}(\gamma = .01)$	4	6.000000	5408.000	1.0003314	1.83227
$G_{1,6}(\gamma = .5)$	4	6.000000	5408.000	1.0003314	1.82236
$G_{2,6}(\gamma = -.01)$	4	6.000000	5550.400	1.0003229	1.79673
$G_{2,6}(\gamma = .01)$	4	6.000000	5550.400	1.0003229	1.80255
$G_{2,6}(\gamma = .5)$	4	6.000000	5550.400	1.0003229	1.90609
$G_{1,7}$	4	8.000000	8298.400	1.0002345	2.34291
$G_{2,7}$	4	8.000000	8440.800	1.0002306	2.48182

where $x \in C[0, 1]$; $s, t \in [0, 1]$ and the kernel G is given as follows:

$$G(s, t) = \begin{cases} (1-s)t, & t \leq s, \\ s(1-t), & s \leq t. \end{cases} \quad (59)$$

We transform the above equation into a finite-dimensional problem by using Gauss-Legendre quadrature formula given as

$$\int_0^1 f(t) dt \approx \sum_{j=1}^n \bar{\omega}_j f(t_j), \quad (60)$$

where the abscissas t_j and the weights $\bar{\omega}_j$ are determined for $n = 8$ by Gauss-Legendre quadrature formula. Denoting the approximation of $x(t_i)$ by x_i ($i = 1, 2, \dots, 8$), we obtain the system of nonlinear equations:

$$2x_i - 2 - \sum_{j=1}^8 a_{ij} (|x_j| + x_j^2) = 0, \quad (61)$$

where

$$a_{ij} = \begin{cases} \bar{\omega}_j t_j (1 - t_i) & \text{if } j \leq i, \\ \bar{\omega}_j t_i (1 - t_j) & \text{if } i < j. \end{cases} \quad (62)$$

In this problem, $(n, \mu) = (8, 10)$ are the values used in (31)–(36) for calculating computational costs and efficiency indices. The initial approximation assumed is $x^{(0)} = \{1, 1, 1, \dots, 1\}^T$ for obtaining the solution

$$\alpha = \{1.0115010875012980 \dots, 1.0546781130247093 \dots, \\ 1.1098925075633296 \dots, 1.1481439774759322, \dots, \\ 1.1481439774759322 \dots, 1.1098925075633296 \dots, \\ 1.0546781130247093 \dots, 1.0115010875012980 \dots\}^T. \quad (63)$$

Table 2 shows the numerical results obtained for the considered problems by various methods. Displayed in this table are the number of iterations (k), the computational order of convergence (ρ_c), the computational costs ($C_{i,p}$) in terms of products, the computational efficiencies ($E_{i,p}$), and the mean elapsed CPU time (e-time). Computational cost and efficiency are calculated according to the corresponding expressions given by (31)–(36) by using the values of parameters n and μ as calculated in each problem, while taking $l = 2.8$ in each case. The mean elapsed CPU time is calculated by taking the mean of 50 performances of the program, where we use $\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\| < 10^{-200}$ as the stopping criterion in single performance of the program.

From the numerical results, we can observe that like the existing methods the present methods show consistent convergence behavior. It is seen that some methods do not preserve the theoretical order of convergence, especially when applied for solving some typical type of nonlinear

systems. This can be observed in the last problem of non-differentiable mixed Hammerstein integral equation where the seventh-order methods $G_{1,7}$ and $G_{2,7}$ yield the eighth order of convergence. However, for the present methods, the computational order of convergence overwhelmingly supports the theoretical order of convergence. Comparison of the numerical values of computational efficiencies exhibited in the second last column of Table 2 verifies the theoretical results of Theorem 2. As we know, the computational efficiency is proportional to the reciprocal value of the total CPU time necessary to complete running iterative process. This means that the method with high efficiency utilizes less CPU time than the method with low efficiency. The truthfulness of this fact can be judged from the numerical values of computational efficiency and elapsed CPU time displayed in the last two columns of Table 2, which are in complete agreement according to the notion.

5. Concluding Remarks

In the foregoing study, we have proposed iterative methods with the sixth order of convergence for solving systems of nonlinear equations. The schemes are totally derivative free and therefore particularly suited to those problems in which derivatives require lengthy computation. A development of first-order divided difference operator for functions of several variables and direct computation by Taylor's expansion are used to prove the local convergence order of new methods. Comparison of efficiencies of the new schemes with the existing schemes is shown. It is observed that the present methods have an edge over similar existing methods, especially when applied for solving large systems of equations. Six numerical examples have been presented and the relevant performances are compared with the existing methods. Computational results have confirmed the robust and efficient character of the proposed techniques. Similar numerical experimentations have been carried out for a number of problems and results are found to be on a par with those presented here.

We conclude the paper with the remark that in many numerical applications multiprecision in computations is required. The results of numerical experiments justify that the high-order efficient methods associated with a multiprecision arithmetic floating point are very useful, because they yield a clear reduction in the number of iterations to achieve the required solution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.
- [2] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, SIAM, Philadelphia, Pa, USA, 2003.
- [3] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New Jersey, NJ, USA, 1964.
- [4] J. F. Steffensen, "Remarks on iteration," *Skandinavski Aktuarsko Tidsskrift*, vol. 16, pp. 64–72, 1933.
- [5] H. Ren, Q. Wu, and W. Bi, "A class of two-step Steffensen type methods with fourth-order convergence," *Applied Mathematics and Computation*, vol. 209, no. 2, pp. 206–210, 2009.
- [6] Z. Liu, Q. Zheng, and P. Zhao, "A variant of Steffensen's method of fourth-order convergence and its applications," *Applied Mathematics and Computation*, vol. 216, no. 7, pp. 1978–1983, 2010.
- [7] M. S. Petković, S. Ilić, and J. Džunić, "Derivative free two-point methods with and without memory for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 1887–1895, 2010.
- [8] M. S. Petković and L. D. Petković, "Families of optimal multipoint methods for solving nonlinear equations: a survey," *Applicable Analysis and Discrete Mathematics*, vol. 4, no. 1, pp. 1–22, 2010.
- [9] M. S. Petković, J. Džunić, and L. D. Petković, "A family of two-point methods with memory for solving nonlinear equations," *Applicable Analysis and Discrete Mathematics*, vol. 5, no. 2, pp. 298–317, 2011.
- [10] Q. Zheng, J. Li, and F. Huang, "An optimal Steffensen-type family for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9592–9597, 2011.
- [11] J. Džunić and M. S. Petković, "On generalized multipoint root-solvers with memory," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2909–2920, 2012.
- [12] J. R. Sharma, R. K. Guha, and P. Gupta, "Some efficient derivative free methods with memory for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 2, pp. 699–707, 2012.
- [13] X. Wang and T. Zhang, "A family of Steffensen type methods with seventh-order convergence," *Numerical Algorithms*, vol. 62, no. 3, pp. 429–444, 2013.
- [14] M. S. Petković, B. Neta, L. D. Petković, and J. Džunić, *Multipoint Methods for Solving Nonlinear Equations*, Elsevier, Boston, Mass, USA, 2013.
- [15] M. Grau-Sánchez, À. Grau, and M. Noguera, "Frozen divided difference scheme for solving systems of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 6, pp. 1739–1743, 2011.
- [16] M. Grau-Sánchez and M. Noguera, "A technique to choose the most efficient method between secant method and some variants," *Applied Mathematics and Computation*, vol. 218, no. 11, pp. 6415–6426, 2012.
- [17] M. Grau-Sánchez, À. Grau, and M. Noguera, "On the computational efficiency index and some iterative methods for solving systems of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 236, no. 6, pp. 1259–1266, 2011.
- [18] F. A. Potra and V. Ptak, *Nondiscrete Induction and Iterative Processes*, Pitman, Boston, Mass, USA, 1984.
- [19] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.
- [20] L. O. Jay, "A note on Q-order of convergence," *BIT Numerical Mathematics*, vol. 41, no. 2, pp. 422–429, 2001.
- [21] L. Fousse, G. Hanrot, V. Lefevre, P. Pelissier, and P. Zimmermann, "MPFR: a multiple-precision binary floating-point library with correct rounding," *ACM Transactions on Mathematical Software*, vol. 33, no. 2, article 13, 15 pages, 2007.

- [22] <http://www.mpfr.org/mpfr-2.1.0/timings.html>.
- [23] M. Grau-Sánchez, J. M. Peris, and J. M. Gutiérrez, “Accelerated iterative methods for finding solutions of a system of nonlinear equations,” *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1815–1823, 2007.
- [24] J. A. Ezquerro, M. Grau-Sánchez, and M. Hernández, “Solving non-differentiable equations by a new one-point iterative method with memory,” *Journal of Complexity*, vol. 28, no. 1, pp. 48–58, 2012.