Research Article

On Some Efficient Techniques 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 148106, India

² Department of Mathematics, Government Ranbir College, Sangrur, Punjab 148001, India

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

Received 18 June 2013; Revised 3 September 2013; Accepted 4 September 2013

Academic Editor: Zhangxing Chen

Copyright © 2013 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.

We present iterative methods of convergence order three, five, and six for solving systems of nonlinear equations. Third-order method is composed of two steps, namely, Newton iteration as the first step and weighted-Newton iteration as the second step. Fifth and sixth-order methods are composed of three steps of which the first two steps are same as that of the third-order method whereas the third is again a weighted-Newton step. Computational efficiency in its general form is discussed and a comparison between the efficiencies of proposed techniques with existing ones is made. The performance is tested through numerical examples. Moreover, theoretical results concerning order of convergence and computational efficiency are verified in the examples. It is shown that the present methods have an edge over similar existing methods, particularly when applied to large systems of equations.

1. Introduction

Solving the system $F(x) = 0$ of nonlinear equations is a common and important problem in various disciplines of science and engineering [1–4]. This problem is precisely stated as follows: For a given nonlinear function $F(x) : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, where $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^t$ and $x = (x_1, x_2, \dots, x_n)^t$, to find a vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^t$ such that $F(\alpha) = 0$. The solution vector α can be obtained as a fixed point of some function $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by means of fixed point iteration

$$x^{(k+1)} = G(x^{(k)}), \quad k = 0, 1, \dots \quad (1)$$

One of the basic procedures for solving systems of nonlinear equations is the classical Newton's method [4, 5] which converges quadratically under the conditions that the function F is continuously differentiable and a good initial approximation $x^{(0)}$ is given. It is defined by

$$x^{(k+1)} = G_2(x^{(k)}) = x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \quad k = 0, 1, 2, \dots, \quad (2)$$

where $F'(x)^{-1}$ is the inverse of first Fréchet derivative $F'(x)$ of the function $F(x)$. Note that this method uses one function,

one first derivative, and one matrix inversion evaluations per iteration.

In order to improve the order of convergence of Newton's method, many modifications have been proposed in literature, for example, see [6–18] and references therein. For a system of n equations in n unknowns, the first Fréchet derivative is a matrix with n^2 evaluations whereas the second Fréchet derivative has $n^2(n+1)/2$ evaluations. Thus, the methods such as those developed in [6–8] with second derivative are considered less efficient from a computational point of view.

In quest of efficient methods without using second Fréchet derivative, a variety of third and higher order methods have been proposed in recent years. For example, Frontini and Sormani in [9] and Homeier in [10] developed some third order methods each requiring the evaluations of one function, two first-order derivatives, and two matrix inversions per iteration. Darvishi and Barati [11] presented a fourth-order method which uses two functions, three first derivatives, and two matrix inversions. Cordero and Torregrosa [12] developed two variants of Newton's method with third-order convergence; one of the variants requires the evaluations of one function, three first derivatives, and two

matrix inversions whereas, the other requires one function, two first derivatives, and two matrix inversions. Noor and Waseem [13] have developed two third-order methods of which one method requires one function, two first derivatives, and two matrix inversions, and the other requires one function, three first derivatives, and two matrix inversions. Cordero et al. [14] presented fourth and fifth-order methods requiring two functions, two first derivatives, and one matrix inversion and three functions, two first derivatives, and one matrix inversion, respectively. Cordero et al. [15] developed a sixth-order method which requires two functions, two first derivatives, and two matrix inversions. Grau-Sánchez et al. [16] proposed the methods with third, fourth, and fifth-order convergence; third-order method with one function, two first derivatives, and two matrix inversions, fourth method with three functions, one first derivative, and one matrix inversion, and fifth-order method with two functions, two first derivatives, and two matrix inversions. Grau-Sánchez et al. [17] have also developed fourth and sixth-order methods requiring the evaluations of one function, one divided difference, one first derivative, and two matrix inversions for the fourth-order method and two functions, one divided difference, one first derivative, and two matrix inversions for the sixth-order method. Sharma et al. [18] proposed a fourth-order method which requires one function, two first derivatives, and two matrix inversions per iteration.

In this paper, our aim is to develop second derivative free iterative methods that may satisfy the basic requirements of generating the quality numerical algorithms, that is, the algorithms which have (i) high convergence speed, (ii) minimum computational cost, and (iii) simple structure. Taking into account these considerations, we here devise the methods of third, fifth, and sixth-order of convergence. Third order method is composed of two steps namely, Newton's and weighted-Newton steps, and per iteration it requires the evaluations of one function, two first derivatives, and one matrix inversion. Fifth and sixth-order methods are composed of three steps of which the first step is Newton's step and the last two are weighted-Newton steps. Per iteration, both of these schemes require the evaluations of two functions, two first derivatives, and one matrix inversion.

We summarize the contents of the paper as follows. In Section 2, third-order scheme is developed and its convergence analysis is studied. In Section 3, fifth-order scheme is developed and its convergence analysis is studied. Sixth-order scheme with its convergence analysis is presented in Section 4. In Section 5, 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 6 to show the consistent convergence behavior of the methods and to verify the theoretical results. Section 7 contains the concluding remarks.

2. Development of Third-Order Method

Since our objective is to develop a method which accelerates the convergence of Newton's method with minimum number

of evaluations. Thus we consider two-step iterative scheme of the type:

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - \left(a_1 I + a_2 F'(x^{(k)})^{-1} F'(y^{(k)}) \right) \\ &\quad \times F'(x^{(k)})^{-1} F(x^{(k)}), \end{aligned} \quad (3)$$

where a_1 and a_2 are some parameters and I denotes the $n \times n$ identity matrix. In order to explore the convergence property of (3), we recall the following result of Taylor's expansion on vector functions (see [2]).

Lemma 1. *Letting $F : D \subseteq R^n \rightarrow R^n$ be a p time Fréchet differentiable in a convex set $D \subseteq R^n$, then for any $x, h \in R^n$, the following expression holds:*

$$\begin{aligned} F(x+h) &= F(x) + F'(x)h + \frac{1}{2!}F''(x)h^2 \\ &\quad + \frac{1}{3!}F'''(x)h^3 + \dots + \frac{1}{p!}F^{(p-1)}(x)h^{p-1} + R_p, \end{aligned} \quad (4)$$

where

$$\begin{aligned} \|R_p\| &\leq \frac{1}{p!} \sup_{0 \leq t \leq 1} \|F^{(p)}(x+th)\| \|h\|^p, \\ h^p &= (h, h, \dots, h). \end{aligned} \quad (5)$$

The Taylor's expansion (4) for $F(x)$ about $x^{(k)}$ is

$$\begin{aligned} F(x) &= F(x^{(k)}) + F'(x^{(k)})(x - x^{(k)}) + \frac{1}{2!}F''(x^{(k)}) \\ &\quad \times (x - x^{(k)})^2 + \frac{1}{3!}F'''(x^{(k)})(x - x^{(k)})^3 \\ &\quad + \frac{1}{4!}F^{(iv)}(x^{(k)})(x - x^{(k)})^4 + O(\|x - x^{(k)}\|^5). \end{aligned} \quad (6)$$

Let $e^{(k)} = x^{(k)} - \alpha$. Then setting $x = \alpha$ and using $F(\alpha) = 0$ in (6), we obtain

$$\begin{aligned} F(x^{(k)}) &= F'(\alpha) \left[e^{(k)} + A_2(e^{(k)})^2 + A_3(e^{(k)})^3 \right. \\ &\quad \left. + A_4(e^{(k)})^4 + O((e^{(k)})^5) \right], \end{aligned} \quad (7)$$

where $A_i = (1/i!) \Gamma F^{(i)}(\alpha) \in L_i(R^n, R^n)$, $\Gamma = F'(\alpha)^{-1}$ and $(e^{(k)})^i = (e^{(k)}, e^{(k)}, \dots, e^{(k)})$, $e^{(k)} \in R^n$.

Also,

$$\begin{aligned} F'(x^{(k)}) &= F'(\alpha) \left[I + 2A_2e^{(k)} + 3A_3(e^{(k)})^2 \right. \\ &\quad \left. + 4A_4(e^{(k)})^3 + O((e^{(k)})^4) \right]. \end{aligned} \quad (8)$$

Then,

$$F'(x^{(k)})^{-1} = [D(e^{(k)})^{-1} + O((e^{(k)})^4)] \Gamma, \quad (9)$$

where $D(e^{(k)}) = I + 2A_2e^{(k)} + 3A_3(e^{(k)})^2 + 4A_4(e^{(k)})^3$.

Inverse of $D(e^{(k)})$ is given by

$$D(e^{(k)})^{-1} = I + X_1e^{(k)} + X_2(e^{(k)})^2 + X_3(e^{(k)})^3, \quad (10)$$

where X_i ($i = 1$ to 3) are such that they satisfy the definition

$$D(e^{(k)})D(e^{(k)})^{-1} = D(e^{(k)})^{-1}D(e^{(k)}) = I. \quad (11)$$

Solving the above system, we have

$$\begin{aligned} X_1 &= -2A_2, \\ X_2 &= 4A_2^2 - 3A_3, \\ X_3 &= -(8A_2^3 - 6A_2A_3 - 6A_3A_2 + 4A_4). \end{aligned} \quad (12)$$

From the above expressions, we find that

$$\begin{aligned} F'(x^{(k)})^{-1} &= [I - 2A_2e^{(k)} + (4A_2^2 - 3A_3)(e^{(k)})^2 \\ &\quad - (8A_2^3 - 6A_2A_3 - 6A_3A_2 + 4A_4) \\ &\quad \times (e^{(k)})^3 + O((e^{(k)})^4)] \Gamma. \end{aligned} \quad (13)$$

From (7) and (13), it follows that

$$\begin{aligned} F'(x^{(k)})^{-1}F(x^{(k)}) &= e^{(k)} - A_2(e^{(k)})^2 + 2(A_2^2 - A_3) \\ &\quad \times (e^{(k)})^3 + O((e^{(k)})^4). \end{aligned} \quad (14)$$

Letting $\bar{e}^{(k)} = y^{(k)} - \alpha$, then using (14) in the first step of (3), we have

$$\bar{e}^{(k)} = A_2(e^{(k)})^2 - 2(A_2^2 - A_3)(e^{(k)})^3 + O((e^{(k)})^4). \quad (15)$$

Expanding $F'(y^{(k)})$ about α , we obtain

$$F'(y^{(k)}) = F'(\alpha) [I + 2A_2\bar{e}^{(k)} + 3A_3(\bar{e}^{(k)})^2 + O((\bar{e}^{(k)})^3)]. \quad (16)$$

Then,

$$\begin{aligned} F'(x^{(k)})^{-1}F'(y^{(k)}) &= I - 2A_2(e^{(k)} - \bar{e}^{(k)}) + (4A_2^2 - 3A_3) \\ &\quad \times (e^{(k)})^2 - 4A_2^2e^{(k)}\bar{e}^{(k)} \\ &\quad - (8A_2^3 - 6A_2A_3 - 6A_3A_2 + 4A_4) \\ &\quad \times (e^{(k)})^3 + O((e^{(k)})^4). \end{aligned} \quad (17)$$

Substituting (14) and (17) in the second step of (3), we obtain the error equation as

$$\begin{aligned} e^{(k+1)} &= (1 - a_1 - a_2)e^{(k)} + (a_1 + 3a_2)A_2(e^{(k)})^2 - 2a_2A_2\bar{e}^{(k)} \\ &\quad \times e^{(k)} - (2(a_1 + 4a_2)A_2^2 - (2a_1 + 5a_2)A_3) \\ &\quad \times (e^{(k)})^3 + O((e^{(k)})^4). \end{aligned} \quad (18)$$

Our aim is to find values of the parameters a_1 and a_2 in such a way that the proposed iterative scheme (3) may produce order of convergence as high as possible. This happens when $a_1 = 3/2$ and $a_2 = -1/2$. Then, the error equation (18) on using (15) becomes

$$e^{(k+1)} = \frac{1}{2}(4A_2^2 + A_3)(e^{(k)})^3 + O((e^{(k)})^4). \quad (19)$$

According to the above analysis the following theorem can be formulated.

Theorem 2. *Let the function $F : D \subseteq R^n \rightarrow R^n$ be sufficiently differentiable in a neighborhood D of its zero α . If an initial approximation $x^{(0)}$ is sufficiently close to α , then the local order of convergence of the iterative scheme (3) is at least 3, provided $a_1 = 3/2$ and $a_2 = -1/2$.*

Thus, the proposed scheme (3) finally can be written as

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} &= G_{1,3} = x^{(k)} - \frac{1}{2} \left(3I - F'(x^{(k)})^{-1}F'(y^{(k)}) \right) \\ &\quad \times F'(x^{(k)})^{-1}F(x^{(k)}). \end{aligned} \quad (20)$$

Clearly this formula uses one function, two derivatives, and one matrix inversion evaluations per iteration.

3. The Fifth-Order Method

Based on the two-step scheme (20), we propose the following three-step scheme:

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \frac{1}{2} \left(3I - F'(x^{(k)})^{-1}F'(y^{(k)}) \right) \\ &\quad \times F'(x^{(k)})^{-1}F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - \left(b_1I + b_2F'(x^{(k)})^{-1}F'(y^{(k)}) \right) \\ &\quad \times F'(x^{(k)})^{-1}F(z^{(k)}), \end{aligned} \quad (21)$$

where b_1 and b_2 are some parameters that are to be determined.

Now to calculate the local order of convergence, let $\tilde{e}^{(k)} = z^{(k)} - \alpha$. In view of (19), error in the second step of (21) is given by

$$\tilde{e}^{(k)} = \frac{1}{2} (4A_2^2 + A_3) (e^{(k)})^3 + O((e^{(k)})^4). \quad (22)$$

Expanding $F(z^{(k)})$ about α ,

$$F(z^{(k)}) = F'(\alpha) [\tilde{e}^{(k)} + O((\tilde{e}^{(k)})^2)]. \quad (23)$$

Using (13), (17), (22), and (23) in the third step of (21), it follows that

$$\begin{aligned} e^{(k+1)} &= (1 - b_1 - b_2) \tilde{e}^{(k)} + 2(b_1 + 2b_2) A_2 e^{(k)} \tilde{e}^{(k)} \\ &\quad - 2b_2 A_2 \tilde{e}^{(k)} \tilde{e}^{(k)} - (4(b_1 + 3b_2) A_2^2 - 3(b_1 + 2b_2) A_3) \\ &\quad \times (e^{(k)})^2 \tilde{e}^{(k)} + O((e^{(k)})^6). \end{aligned} \quad (24)$$

It is clear that for the parameter values $b_1 = 2$ and $b_2 = -1$, the terms containing $(e^{(k)})^3$ and $(e^{(k)})^4$ vanish. Thus, from the expressions of $\tilde{e}^{(k)}$ and $\tilde{e}^{(k)}$ the error equation (24) yields

$$e^{(k+1)} = 3A_2^2 (4A_2^2 + A_3) (e^{(k)})^5 + O((e^{(k)})^6), \quad (25)$$

which shows fifth-order of convergence.

From the above analysis, we can state the following theorem.

Theorem 3. *Let the function $F : D \subseteq R^n \rightarrow R^n$ be sufficiently differentiable in an open neighborhood D of its zero α . If an initial approximation $x^{(0)}$ is sufficiently close to α , then the local order of convergence of method (21) is at least 5, if $b_1 = 2$ and $b_2 = -1$.*

The proposed algorithm (21) now can be written as

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \frac{1}{2} (3I - F'(x^{(k)})^{-1} F'(y^{(k)})) \\ &\quad \times F'(x^{(k)})^{-1} F(x^{(k)}), \\ x^{(k+1)} &= G_{1,5} = z^{(k)} - (2I - F'(x^{(k)})^{-1} F'(y^{(k)})) \\ &\quad \times F'(x^{(k)})^{-1} F(z^{(k)}). \end{aligned} \quad (26)$$

It is clear that this formula uses the evaluations of two functions, two derivatives and only one matrix inversion in all.

4. The Sixth-Order Method

Here, with the first two steps of the scheme (20) we consider the following three-step scheme:

$$\begin{aligned} y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - \frac{1}{2} (3I - F'(x^{(k)})^{-1} F'(y^{(k)})) \\ &\quad \times F'(x^{(k)})^{-1} F(x^{(k)}), \\ x^{(k+1)} &= z^{(k)} - (c_1 I + F'(x^{(k)})^{-1} F'(y^{(k)})) \\ &\quad \times (c_2 I + c_3 F'(x^{(k)})^{-1} F'(y^{(k)})) \\ &\quad \times F'(x^{(k)})^{-1} F(z^{(k)}), \end{aligned} \quad (27)$$

where c_1, c_2 , and c_3 are some arbitrary constants.

In the following lines we obtain the local order of convergence of the above proposed scheme. Using (13), (17), (22), and (23) in the third step of (27), it follows that

$$\begin{aligned} e^{(k+1)} &= (1 - c_1 - c_2 - c_3) \tilde{e}^{(k)} + 2(c_1 + 2c_2 + 3c_3) \\ &\quad \times A_2 e^{(k)} \tilde{e}^{(k)} - 2(c_2 + 2c_3) A_2 \tilde{e}^{(k)} \tilde{e}^{(k)} \\ &\quad - (4(c_1 + 3c_2 + 6c_3) A_2^2 - 3(c_1 + 2c_2 + 3c_3) A_3) \\ &\quad \times (e^{(k)})^2 \tilde{e}^{(k)} + 8(c_2 + 3c_3) A_2^2 e^{(k)} \tilde{e}^{(k)} \tilde{e}^{(k)} \\ &\quad - (c_1 + c_2 + c_3) A_2 (\tilde{e}^{(k)})^2 \\ &\quad + 2(4(c_1 + 4c_2 + 10c_3) A_2^3 - 3(c_1 + 3c_2 + 6c_3) A_2 A_3 \\ &\quad - 3(c_1 + 3c_2 + 6c_3) A_3 A_2 \\ &\quad + 2(c_1 + 2c_2 + 3c_3) A_4) \\ &\quad \times (e^{(k)})^3 \tilde{e}^{(k)} + O((e^{(k)})^7). \end{aligned} \quad (28)$$

Combining (15), (22), and (28), it is easy to prove that for the parameters $c_1 = 7/2$, $c_2 = -4$, and $c_3 = 3/2$, error equation (28) produces the maximum error. For this set of values the above equation reduces to

$$\begin{aligned} e^{(k+1)} &= \frac{1}{2} (18A_2^3 + \frac{1}{2} A_2 A_3 - 3A_3 A_2) \\ &\quad \times (4A_2^2 + A_3) (e^{(k)})^6 + O((e^{(k)})^7), \end{aligned} \quad (29)$$

which shows sixth-order of convergence. Thus, based on the above discussion, we can formulate the following theorem.

Theorem 4. *Let the function $F : D \subseteq R^n \rightarrow R^n$ be sufficiently differentiable in an open neighborhood D of its zero α . If an initial approximation $x^{(0)}$ is sufficiently close to α , then the local order of convergence of method (27) is at least 6, provided $c_1 = 7/2$, $c_2 = -4$ and $c_3 = 3/2$.*

Finally, the proposed scheme (27) is given by

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\
 z^{(k)} &= x^{(k)} - \frac{1}{2} \left(3I - F'(x^{(k)})^{-1} F'(y^{(k)}) \right) \\
 &\quad \times F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= G_{1,6} = z^{(k)} - \left(\frac{7}{2} I - F'(x^{(k)})^{-1} F'(y^{(k)}) \right) \\
 &\quad \times \left(4I - \frac{3}{2} F'(x^{(k)})^{-1} F'(y^{(k)}) \right) \\
 &\quad \times F'(x^{(k)})^{-1} F(z^{(k)}). \tag{30}
 \end{aligned}$$

Like fifth-order scheme, this formula also uses two function evaluations, two derivatives evaluations, and only one matrix inversion in all.

5. Computational Efficiency

To obtain an assessment of the efficiency of the proposed methods we shall 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. In order to do this, we must consider all possible factors which contribute to the total cost of computation. For a system of n nonlinear equations in n unknowns, the computational cost per iteration is given by (see [17])

$$C(\mu_0, \mu_1, n) = P_0(n) \mu_0 + P_1(n) \mu_1 + P(n). \tag{31}$$

Here $P_0(n)$ represents the number of evaluations of scalar functions $(f_1, f_2, \dots, f_n)$ used in the evaluation of F , $P_1(n)$ is the number of evaluations of scalar functions of F' , say $\partial f_i / \partial x_j$, $1 \leq i, j \leq n$, $P(n)$ represents the number of products or quotients needed per iteration, and μ_0 and μ_1 are ratios between products and evaluations required to express the value of $C(\mu_0, \mu_1, n)$ in terms of products. We suppose that a quotient is equivalent to l products.

To compute F in any iterative method, we evaluate n scalar functions, whereas the number of scalar evaluations is n^2 for any new derivative F' . In addition, we must include the amount of computational work required to evaluate inverse of a matrix. Instead of computing the inverse 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. Moreover, we must add n products for the multiplication of a vector by a scalar, while n^2 products for the same operation in case of a matrix.

We compare the computational efficiency of present third order method $G_{1,3}$ with existing third-order methods by Potra-Pták [3], Homeier [10], Noor-Waseem [13], and Cordero-Torregrosa [12]; the fifth-order method $G_{1,5}$ with fifth-order method given by Grau-Sánchez et al. [16]; the

sixth-order method $G_{1,6}$ with sixth-order method developed by Cordero et al. [15]. These methods are expressed as follows.

Method by Potra-Pták ($G_{2,3}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= y^{(k)} - F'(x^{(k)})^{-1} F(y^{(k)}). \tag{32}
 \end{aligned}$$

Homeier's method ($G_{3,3}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - \frac{1}{2} F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= x^{(k)} - F'(y^{(k)})^{-1} F(x^{(k)}). \tag{33}
 \end{aligned}$$

Method by Noor and Waseem ($G_{4,3}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= x^{(k)} - 4 \left[F'(x^{(k)}) + 3F' \left(\frac{x^{(k)} + 2y^{(k)}}{3} \right) \right]^{-1} F(x^{(k)}). \tag{34}
 \end{aligned}$$

Method by Cordero and Torregrosa ($G_{5,3}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= x^{(k)} - 3 \left[2F' \left(\frac{3x^{(k)} + y^{(k)}}{4} \right) \right. \\
 &\quad \left. - F' \left(\frac{x^{(k)} + y^{(k)}}{2} \right) \right. \\
 &\quad \left. + 2F' \left(\frac{x^{(k)} + 3y^{(k)}}{4} \right) \right]^{-1} F(x^{(k)}). \tag{35}
 \end{aligned}$$

Method by Grau-Sánchez et al. ($G_{2,5}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \\
 z^{(k)} &= x^{(k)} - \frac{1}{2} \left[F'(x^{(k)})^{-1} + F'(y^{(k)})^{-1} \right] F(x^{(k)}), \tag{36} \\
 x^{(k+1)} &= z^{(k)} - F'(y^{(k)})^{-1} F(z^{(k)}).
 \end{aligned}$$

Method by Cordero et al. ($G_{2,6}$):

$$\begin{aligned}
 y^{(k)} &= x^{(k)} - \frac{2}{3} F'(x^{(k)})^{-1} F(x^{(k)}), \\
 z^{(k)} &= x^{(k)} - \frac{1}{2} \left[3F'(y^{(k)}) - F'(x^{(k)}) \right]^{-1} \\
 &\quad \times \left(3F'(y^{(k)}) + F'(x^{(k)}) \right) F'(x^{(k)})^{-1} F(x^{(k)}), \\
 x^{(k+1)} &= z^{(k)} - 2 \left[3F'(y^{(k)}) - F'(x^{(k)}) \right]^{-1} F(z^{(k)}). \tag{37}
 \end{aligned}$$

FIGURE 1: Boundary lines in (μ_1, μ_0) -plane for $E_{1,3}$ and $E_{2,3}$. FIGURE 2: Plots for E values of $G_{1,3}$ and $G_{3,3}$ for $n \geq 5$.

In addition, we also compare 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

$$C_{1,3} = n\mu_0 + 2n^2\mu_1 + \frac{n}{6}(2n^2 + 15n + 1 + 3l(n+3)), \quad (38)$$

$$E_{1,3} = 3^{1/C_{1,3}}.$$

$$C_{2,3} = 2n\mu_0 + n^2\mu_1 + \frac{n}{6}(2n^2 + 9n - 11 + 3l(n+3)), \quad (39)$$

$$E_{2,3} = 3^{1/C_{2,3}}.$$

$$C_{3,3} = n\mu_0 + 2n^2\mu_1 + \frac{n}{3}(2n^2 + 3n - 2 + 3l(n+1)), \quad (40)$$

$$E_{3,3} = 3^{1/C_{3,3}}.$$

$$C_{4,3} = n\mu_0 + 2n^2\mu_1 + \frac{n}{3}(2n^2 + 6n + 4 + 3l(n+1)), \quad (41)$$

$$E_{4,3} = 3^{1/C_{4,3}}.$$

$$C_{5,3} = n\mu_0 + 4n^2\mu_1 + \frac{n}{3}(2n^2 + 9n + 13 + 3l(n+1)), \quad (42)$$

$$E_{5,3} = 3^{1/C_{5,3}}.$$

$$C_{1,5} = 2n\mu_0 + 2n^2\mu_1 + \frac{n}{6}(2n^2 + 33n - 5 + 3l(n+7)), \quad (43)$$

$$E_{1,5} = 5^{1/C_{1,5}}.$$

$$C_{2,5} = 2n\mu_0 + 2n^2\mu_1 + \frac{n}{3}(2n^2 + 6n - 5 + 3l(n+2)), \quad (44)$$

$$E_{2,5} = 5^{1/C_{2,5}}.$$

$$C_{1,6} = 2n\mu_0 + 2n^2\mu_1 + \frac{n}{6}(2n^2 + 45n + 1 + 3l(n+9)), \quad (45)$$

$$E_{1,6} = 6^{1/C_{1,6}}.$$

$$C_{2,6} = 2n\mu_0 + 2n^2\mu_1 + \frac{n}{3}(2n^2 + 12n + 1 + 3l(n+2)), \quad (46)$$

$$E_{2,6} = 6^{1/C_{2,6}}.$$

5.1. Comparison between the Efficiencies. To compare the computational efficiencies of the iterative methods $G_{i,p}$, we consider the ratio

$$R_{i,p;j,q} = \frac{\log E_{i,p}}{\log E_{j,q}} = \frac{\log(p) C_{j,q}}{\log(q) C_{i,p}}. \quad (47)$$

It is clear that if $R_{i,p;j,q} > 1$, the iterative method $G_{i,p}$ is more efficient than $G_{j,q}$. Taking into account that the border between two computational efficiencies is given by $R_{i,p;j,q} = 1$, this boundary is given by the equation μ_0 written as a function of μ_1 , n , and l , $(\mu_1, \mu_0) \in (0, +\infty) \times (0, +\infty)$, n is a positive integer ≥ 2 , and $l \geq 1$.

$G_{1,3}$ versus $G_{2,3}$ Case. Here we study the boundary $R_{1,3;2,3} = 1$ expressed by μ_0 as a function of μ_1 and n . This boundary, on using (38) and (39) in (47), is given by

$$\mu_0 = n\mu_1 + n + 2. \quad (48)$$

In order to compare the efficiencies $E_{1,3}$ and $E_{2,3}$ in the (μ_1, μ_0) -plane, we present in Figure 1 some boundary lines L_i ($i = 1, 2, 3, 4$) corresponding to the cases $n = 2, 5, 10, 15$. These boundaries are the lines with positive slopes, where $E_{1,3} > E_{2,3}$ on the above and $E_{2,3} > E_{1,3}$ on the below of each line.

$G_{1,3}$ versus $G_{3,3}$ Case. Using the corresponding values of $C_{1,3}$ and $C_{3,3}$ from (38) and (40) in (47), it follows that $R_{1,3;3,3} > 1$ for all $n \geq 5$ and $l \geq 1$. Thus, we have $E_{1,3} > E_{3,3}$ for all $n \geq 5$ and $l \geq 1$. This is shown graphically in Figure 2 for

a particular set $(\mu_0, \mu_1, l) = (1, 1, 2.8)$. The reason for taking $l = 2.8$ will be made clear in the next section.

$G_{1,3}$ versus $G_{4,3}$ and $G_{1,3}$ versus $G_{5,3}$ Cases. From (47) on using (38), (41) and then (38), (42), it is easy to prove that $R_{1,3;4,3} > 1$ and $R_{1,3;5,3} > 1$ for all $n \geq 2$ and $l \geq 1$. Thus, we conclude that $E_{1,3}$ is higher than $E_{4,3}$ and $E_{5,3}$ for all $n \geq 2$ and $l \geq 1$. This situation is shown graphically in Figure 3 taking the same set of values of (μ_0, μ_1, l) as in the previous case.

$G_{1,3}$ versus $G_{1,5}$ Case. Here we study the boundary $R_{1,5;1,3} = 1$ expressed by μ_0 as a function of μ_1, l , and n . This boundary is given by

$$\mu_0 = \left(2n^2(r-s) + 3n((r-s)(4\mu_1 + l) + 5r - 11s) + 3l(3r - 7s) + r + 5s \right) (6(2s - r))^{-1}, \quad (49)$$

where $r = \ln(5)$ and $s = \ln(3)$. In order to compare the efficiencies $E_{1,3}$ and $E_{1,5}$ in the (μ_1, μ_0) -plane, we present in Figure 4 some boundary lines L_i ($i = 1, 2, 3, 4$) corresponding to the cases $n = 2, 5, 10, 15$ taking $l = 2.8$ in each case. These boundaries are the straight lines with positive slopes, where $E_{1,3} > E_{1,5}$ on the above and $E_{1,5} > E_{1,3}$ on the below of each line.

$G_{1,3}$ versus $G_{1,6}$ Case. In this case the boundary $R_{1,6;1,3} = 1$ is expressed by

$$\mu_0 = \frac{2n^2t + 3n(4t\mu_1 + tl + 5(t - 2s)) + 9l(t - 2s) + t}{6(s - t)}, \quad (50)$$

where $s = \ln(3)$ and $t = \ln(2)$. Similar to the case above, here we also present the boundaries L_i ($i = 1, 2, 3, 4$) using the same set of n and l in (μ_1, μ_0) -plane to compare the efficiencies $E_{1,3}$ and $E_{1,6}$. These boundaries are the lines with positive slopes, where $E_{1,3} > E_{1,6}$ on the above and $E_{1,6} > E_{1,3}$ on the below of each line (see Figure 5).

$G_{1,5}$ versus $G_{1,6}$ Case. For this case the boundary $R_{1,6;1,5} = 1$ is given by

$$\mu_0 = \left(2n^2(u-r) + 3n((u-r)(4\mu_1 + l) + 11u - 15r) + 3l(7u - 9r) - 5u - r \right) (12(r - u))^{-1}, \quad (51)$$

where $r = \ln(5)$ and $u = \ln(6)$. Here we also draw boundaries in (μ_1, μ_0) -plane with the same set of values of n and l . The boundaries L_i are the lines with negative slopes, where $E_{1,6} > E_{1,5}$ on the above and $E_{1,5} > E_{1,6}$ on the below of each line (see Figure 6).

$G_{1,5}$ versus $G_{2,5}$ Case. Substituting the corresponding values of $C_{1,5}$ and $C_{2,5}$ from (43) and (44) in (47), we see that $R_{1,5;2,5} > 1$ for all $n \geq 10$ and $l \geq 1$, which further implies that $E_{1,5} > E_{2,5}$ for all $n \geq 10, l \geq 1$. This result is verified graphically in Figure 7 by using the values $(\mu_0, \mu_1, l) = (1, 1, 2.8)$.

$G_{1,6}$ versus $G_{2,6}$ Case. In this case substitution of the corresponding values of $C_{1,6}$ and $C_{2,6}$ from (45) and (46) in (47) results in $R_{1,6;2,6} > 1$ for all $n \geq 10$ and $l \geq 1$. Thus, we analyze that $E_{1,6} > E_{2,6}$ for all $n \geq 10$ and $l \geq 1$. This is shown

FIGURE 3: Plots for E values of $G_{1,3}$, $G_{4,3}$, and $G_{5,3}$ for $n \geq 2$.

FIGURE 4: Boundary lines in (μ_1, μ_0) -plane for $E_{1,3}$ and $E_{1,5}$.

graphically in Figure 8 by using the same values of (μ_0, μ_1, l) that are used in the previous case.

We summarize the above results in following theorem.

Theorem 5. (i) For all $\mu_0 > 0$, $\mu_1 > 0$, and $l \geq 1$ we have:

- (a) $E_{1,3} > E_{3,3} \forall n \geq 5$,
- (b) $E_{1,3} > E_{4,3}$ and $E_{1,3} > E_{5,3} \forall n \geq 2$,
- (c) $E_{1,5} > E_{2,5}$ and $E_{1,6} > E_{2,6} \forall n \geq 10$.

Otherwise, the comparison depends on n, μ_0, μ_1 , and l .

(ii) For all $n \geq 2$, we have:

- (a) $E_{1,3} > E_{2,3}$ for $\mu_0 > m_1$,
- (b) $E_{1,5} > E_{1,3}$ for $\mu_0 < m_2$,
- (c) $E_{1,6} > E_{1,3}$ for $\mu_0 < m_3$,
- (d) $E_{1,6} > E_{1,5}$ for $\mu_0 > m_4$,

where $m_1 = n\mu_1 + n + 2$, $m_2 = (2n^2(r-s) + 3n((r-s)(4\mu_1 + l) + 5r - 11s) + 3l(3r - 7s) + r + 5s)/6(2s - r)$, $m_3 = (2n^2t + 3n(4t\mu_1 + tl + 5(t - 2s)) + 9l(t - 2s) + t)/6(s - t)$, and $m_4 = (2n^2(u - r) + 3n((u - r)(4\mu_1 + l) + 11u - 15r) + 3l(7u - 9r) - 5u - r)/12(r - u)$.

FIGURE 5: Boundary lines in (μ_1, μ_0) -plane for $E_{1,3}$ and $E_{1,6}$. FIGURE 6: Boundary lines in (μ_1, μ_0) -plane for $E_{1,5}$ and $E_{1,6}$.

6. 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 $G_{2,3}$, $G_{3,3}$, $G_{4,3}$, $G_{5,3}$, $G_{2,5}$, and $G_{2,6}$ that we have introduced in previous section. All computations are performed in 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 CPU time utilized in the execution of program which is computed by the *Mathematica* command "TimeUsed[]". To verify the theoretical order of convergence, we calculate the computational order of convergence (ρ_c) using the following formula [20]:

$$\rho_c = \frac{\log(\|F(x^{(k)})\| / \|F(x^{(k-1)})\|)}{\log(\|F(x^{(k-1)})\| / \|F(x^{(k-2)})\|)}, \quad (52)$$

FIGURE 7: Plots for E values of $G_{1,5}$ and $G_{2,5}$ for $n \geq 10$. FIGURE 8: Plots for E values of $G_{1,6}$ and $G_{2,6}$ for $n \geq 10$.

taking into consideration the last three approximations in the iterative process.

To connect the analysis of computational efficiency with numerical examples, we apply the definition of the computational cost (31), according to which an estimation of the factors μ_0 and μ_1 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, an estimation of the cost of the elementary functions in product units is shown, wherein the running time of one product is measured in milliseconds. For the hardware and the software used in the present numerical work, the computational cost of quotient with respect to product is $l = 2.8$ (see Table 1).

We consider the following problems for numerical tests.

Problem 1. Consider the system of two equations:

$$\begin{aligned} x_1 + e^{x_2} - \cos(x_2) &= 0, \\ 3x_1 - x_2 - \sin(x_2) &= 0, \end{aligned} \quad (53)$$

TABLE 1: Estimation of computational cost of elementary functions computed with Mathematica 7.0 in a processor Intel(R) Core (TM) i5-480M CPU @ 2.67 GHz (64-bit Machine) Microsoft Windows 7 Home Basic 2009, where $x = \sqrt{3} - 1$ and $y = \sqrt{5}$.

Digits	$x * y$	$l = x/y$	$\sqrt{x}$	$\exp(x)$	$\ln(x)$	$\sin(x)$	$\cos(x)$	$\arccos(x)$	$\arctan(x)$	x^y
4096	0.4665 ms	2.8	1.3	81	78	102	102	166	162	178

with initial approximation $x^{(0)} = \{0.1, 0.3\}^t$. The solution is $\alpha = \{0, 0\}^t$. The concrete values of parameters in this case are $(n, \mu_0, \mu_1) = (2, 143, 71.25)$, which we use in (38)–(46) for calculating computational costs and efficiency indices of different methods.

Problem 2. Consider the mixed Hammerstein integral equation (see [2, 17]):

$$x(s) = 1 + \frac{1}{5} \int_0^1 G(s, t) x(t)^3 dt, \quad (54)$$

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

$$G(s, t) = \begin{cases} (1-s)t, & t \leq s, \\ s(1-t), & s \leq t. \end{cases} \quad (55)$$

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

$$\int_0^1 f(t) dt \approx \sum_{j=1}^m \omega_j f(t_j), \quad (56)$$

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

$$x_i - 1 - \frac{1}{5} \sum_{j=1}^2 a_{ij} x_j^3 = 0, \quad (57)$$

where

$$a_{ij} = \begin{cases} \omega_j t_j (1-t_i) & \text{if } j \leq i, \\ \omega_j t_i (1-t_j) & \text{if } i < j. \end{cases} \quad (58)$$

The initial approximation assumed is $x^{(0)} = \{-1, -1\}^t$ and the solution of this problem is $\alpha = \{1.0594598013021009 \dots, 1.0594598013021009 \dots\}^t$. Here the values of (n, μ_0, μ_1) are $(2, 5, 1.5)$.

Problem 3. Consider the system of three equations:

$$\begin{aligned} 10x_1 + \sin(x_1 + x_2) - 1 &= 0, \\ 8x_2 - \cos^2(x_3 - x_2) - 1 &= 0, \\ 12x_3 + \sin x_3 - 1 &= 0, \end{aligned} \quad (59)$$

with initial approximation $x^{(0)} = \{4/5, 1/2, 1/8\}^t$. The solution correct to 16 places of decimal is $\alpha = \{0.0689783491726666, 0.2464424186091830, 0.0769289119875370\}^t$ and the concrete values of parameters are $(n, \mu_0, \mu_1) = (3, 103.3, 45.56)$.

Problem 4. Consider the system of eight equations:

$$\begin{aligned} x_i^2 (x_i + x_{i+1}) - 1.5 &= 0, \quad i = 1, 2, \dots, 7 \\ x_8^2 (x_1 + x_8) - 1.5 &= 0. \end{aligned} \quad (60)$$

The initial approximation chosen is $x^{(0)} = \{1.1, 1.1, \dots, 1.1\}^t$ for the solution $\alpha = \{0.9085602964160698, 0.9085602964160698, \dots, 0.9085602964160698\}^t$. Here the values of (n, μ_0, μ_1) are $(8, 2, 0.5)$.

Problem 5. Consider the system of thirteen equations:

$$\sum_{j=1, j \neq i}^{13} x_j - e^{-x_i} = 0, \quad 1 \leq i \leq 13, \quad (61)$$

with the initial value $x^{(0)} = \{0.2, 0.2, \dots, 0.2\}^t$ towards the solution $\alpha = \{0.0771462076130646, 0.0771462076130646, \dots, 0.0771462076130646\}^t$. For this problem the corresponding values of parameters (n, μ_0, μ_1) are $(13, 81, 6.23)$.

Problem 6. Consider the system of fifteen equations:

$$x_i - \cos\left(2x_i - \sum_{j=1}^{15} x_j\right) = 0, \quad 1 \leq i \leq 15, \quad (62)$$

with initial value $x^{(0)} = \{1.1, 1.1, \dots, 1.1\}^t$. The solution of this problem is $\alpha = \{0.9814737799885131, 0.9814737799885131, \dots, 0.9814737799885131\}^t$ and $(n, \mu_0, \mu_1) = (15, 102, 6.8)$.

Problem 7. Now consider the system of twenty equations:

$$\begin{aligned} x_i^2 x_{i+1} - 1 &= 0, \quad i = 1, 2, \dots, 19, \\ x_{20}^2 x_1 - 1 &= 0, \end{aligned} \quad (63)$$

with initial approximation $x^{(0)} = \{1.2, 1.2, \dots, 1.2\}^t$ and the solution $\alpha = \{1, 1, \dots, 1\}^t$. The concrete values of parameters in this case are $(n, \mu_0, \mu_1) = (20, 2, 0.15)$.

The numbers 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 CPU time (CPUtime) for each method are displayed in Table 2. Computational cost and efficiency are calculated according to the corresponding expressions given by (38)–(46) by using the values of parameters n , μ_0 , and μ_1 as shown at the end of each problem, while taking $l = 2.8$ in each case. The mean CPU time is calculated by taking the mean of 50 performances of the program, where we use

TABLE 2: Performance of methods.

Methods	k	ρ_c	$C_{i,p}$	$E_{i,p}$	CPUtime
Problem 1					
$G_{1,3}$	7	3.0000	883.0	1.001245	0.205573
$G_{2,3}$	7	3.0000	876.0	1.001255	0.192909
$G_{3,3}$	7	3.0000	880.8	1.001248	0.203215
$G_{4,3}$	7	3.0000	888.8	1.001237	0.218364
$G_{5,3}$	7	5.0000	1468.8	1.000748	0.460909
$G_{1,5}$	5	5.0000	1190.2	1.001353	0.176545
$G_{2,5}$	5	5.0000	1174.4	1.001370	0.172728
$G_{1,6}$	5	6.0000	1205.8	1.001487	0.166418
$G_{2,6}$	5	6.0000	1186.4	1.001511	0.162545
Problem 2					
$G_{1,3}$	6	3.0000	49.0	1.022674	0.052327
$G_{2,3}$	6	3.0000	45.0	1.024714	0.050854
$G_{3,3}$	7	3.0000	46.8	1.023752	0.051545
$G_{4,3}$	7	3.0000	54.8	1.020250	0.052173
$G_{5,3}$	7	3.0000	76.8	1.014408	0.102091
$G_{1,5}$	5	5.0000	80.2	1.020271	0.052734
$G_{2,5}$	5	5.0000	64.4	1.025306	0.050727
$G_{1,6}$	5	6.0000	95.8	1.018879	0.053545
$G_{2,6}$	5	6.0000	76.4	1.023730	0.051182
Problem 3					
$G_{1,3}$	6	3.0000	1187.2	1.000926	0.489455
$G_{2,3}$	6	3.0000	1072.0	1.001025	0.398545
$G_{3,3}$	6	3.0000	1188.6	1.000925	0.540818
$G_{4,3}$	6	3.0000	1203.6	1.000913	0.566636
$G_{5,3}$	6	3.0000	2041.7	1.000538	1.023910
$G_{1,5}$	5	5.0000	1537.9	1.001047	0.379364
$G_{2,5}$	5	5.0000	1512.9	1.001064	0.376345
$G_{1,6}$	4	6.0000	1567.3	1.001144	0.365864
$G_{2,6}$	4	6.0000	1536.9	1.001167	0.364636
Problem 4					
$G_{1,3}$	7	3.0000	535.2	1.002055	0.340364
$G_{2,3}$	7	3.0000	439.2	1.002505	0.228364
$G_{3,3}$	7	3.0000	681.6	1.001613	0.370091
$G_{4,3}$	7	3.0000	761.6	1.001444	0.387273
$G_{5,3}$	7	3.0000	913.6	1.001203	0.393000
$G_{1,5}$	5	5.0000	780.0	1.002066	0.237091
$G_{2,5}$	5	5.0000	776.0	1.002076	0.295000
$G_{1,6}$	5	6.0000	938.4	1.001911	0.262727
$G_{2,6}$	5	6.0000	920.0	1.001949	0.240543
Problem 5					
$G_{1,3}$	6	3.0000	4606.9	1.000238	1.660450
$G_{2,3}$	6	3.0000	4412.1	1.000249	1.618180
$G_{3,3}$	6	3.0000	5293.3	1.000208	2.425090
$G_{4,3}$	6	3.0000	5488.3	1.000200	2.437910
$G_{5,3}$	6	3.0000	7802.1	1.000141	3.006550
$G_{1,5}$	4	5.0000	6226.7	1.000259	1.515910
$G_{2,5}$	4	5.0000	6538.7	1.000246	1.799640

TABLE 2: Continued.

Methods	k	ρ_c	$C_{i,p}$	$E_{i,p}$	CPUtime
$G_{1,6}$	4	6.0000	6614.1	1.000271	1.435450
$G_{2,6}$	4	6.0000	6902.7	1.000260	1.510730
Problem 6					
$G_{1,3}$	8	3.0000	6658.0	1.000165	15.81950
$G_{2,3}$	8	3.0000	6403.0	1.000172	12.79912
$G_{3,3}$	8	3.0000	7727.0	1.000142	18.29325
$G_{4,3}$	8	3.0000	7982.0	1.000138	18.29750
$G_{5,3}$	8	3.0000	11312.0	1.000097	28.52323
$G_{1,5}$	6	5.0000	8932.0	1.000180	11.58340
$G_{2,5}$	6	5.0000	9509.0	1.000169	13.87700
$G_{1,6}$	6	6.0000	9439.0	1.000190	11.37760
$G_{2,6}$	6	6.0000	9989.0	1.000179	12.69813
Problem 7					
$G_{1,3}$	7	3.0000	4474.0	1.000246	0.697636
$G_{2,3}$	7	3.0000	4014.0	1.000274	0.592818
$G_{3,3}$	7	3.0000	7056.0	1.000156	0.882182
$G_{4,3}$	7	3.0000	7496.0	1.000147	0.975636
$G_{5,3}$	7	3.0000	8076.0	1.000136	0.978727
$G_{1,5}$	5	5.0000	5806.0	1.000277	0.584010
$G_{2,5}$	5	5.0000	7532.0	1.000214	0.753091
$G_{1,6}$	5	6.0000	6682.0	1.000268	0.643909
$G_{2,6}$	5	6.0000	8372.0	1.000214	0.779636

$(\|x^{(k+1)} - x^{(k)}\| + \|F(x^{(k)})\|) < 10^{-200}$ as the stopping criterion in single performance of the program.

From numerical results, we can observe that like the existing methods the present methods show consistent convergence behavior. It is also clear that the computational order of convergence overwhelmingly supports the theoretical order of convergence. As far as the verification of the results of Theorem 5 is concerned, it is simple to check the theoretical results of statement (i) using the numerical values of the efficiency indices displayed in the second last column of Table 2. However, the results of statement (ii) are not quite obvious to verify. In order to do this, we first find m_1 , m_2 , m_3 , and m_4 using the values of l , n , and μ_1 obtained for each numerical problem and then we compare the efficiencies as per the rules (a)–(d) of statement (ii). The results are displayed in Table 3. From Table 2 we can see that the numerical values of $E_{1,3}$, $E_{2,3}$, $E_{1,5}$, and $E_{1,6}$ also confirm the results as shown in Table 3.

Comparison of numerical results shows that $G_{2,3}$ is more efficient among third-order methods. However, the present third-order method $G_{1,3}$ is efficient among the rest of third-order methods in majority of the problems. The present fifth-order $G_{1,5}$ and sixth-order $G_{1,6}$ methods are more efficient than the existing methods of same and inferior order for larger systems. This behavior can be observed in the numerical results of Problems 5–7. From the results of last two columns in Table 2, one can conclude that the more is the efficiency of a method the lesser is the computing time of that method. This shows that the efficiency results are in complete

TABLE 3: Comparison between $E_{1,3}$, $E_{2,3}$, $E_{1,5}$, and $E_{1,6}$ according to Theorem 5(ii).

Problem	μ_0	m_1	m_2	m_3	m_4	Comparison between efficiencies as per Theorem 5(ii)			
						Case (a)	Case (b)	Case (c)	Case (d)
						$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,6} > E_{1,5}$
1	143	146.50	239.60	460.43	-120.12	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,6} > E_{1,5}$
2	5	7	-2.87	-16.52	19.38	$E_{2,3} > E_{1,3}$	$E_{1,3} > E_{1,5}$	$E_{1,3} > E_{1,6}$	$E_{1,5} > E_{1,6}$
3	103.3	141.68	228.72	436.50	-109.76	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,6} > E_{1,5}$
4	2	14	2.95	-20.19	40.64	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,3} > E_{1,6}$	$E_{1,5} > E_{1,6}$
5	81	95.99	156.09	268.47	-26.96	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,6} > E_{1,5}$
6	102	119.00	204.39	358.45	-46.55	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,6} > E_{1,5}$
7	2	25	71.93	85.29	50.17	$E_{2,3} > E_{1,3}$	$E_{1,5} > E_{1,3}$	$E_{1,6} > E_{1,3}$	$E_{1,5} > E_{1,6}$

agreement with the CPU time utilized in the execution of program.

7. Concluding Remarks

In the foregoing study, we have developed iterative methods of third, fifth and sixth-order of convergence for solving systems of nonlinear equations. The computational efficiency in its general form is discussed. Then a comparison between the efficiencies of proposed methods with existing methods is made. It is proved that the present methods are at least competitive with existing methods of similar nature; in particular, these are especially efficient for larger systems. To illustrate the new techniques, seven numerical examples are presented and completely solved. The performance is compared with some known methods of similar character. The theoretical order of convergence and the analysis of computational efficiency are verified in the considered examples. The numerical results have confirmed the robust and efficient nature of the proposed techniques.

References

- [1] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, New York, NY, USA, 1966.
- [2] J. M. Ortega and W. C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.
- [3] F. A. Potra and V. Pták, *Nondiscrete Induction and Iterative Processes*, Pitman, Boston, Mass, USA, 1984.
- [4] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, SIAM, Philadelphia, Pa, USA, 2003.
- [5] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1964.
- [6] J. M. Gutiérrez and M. A. Hernández, "A family of chebyshev-halley type methods in banach spaces," *Bulletin of the Australian Mathematical Society*, vol. 55, no. 1, pp. 113–130, 1997.
- [7] M. Palacios, "Kepler equation and accelerated Newton method," *Journal of Computational and Applied Mathematics*, vol. 138, no. 2, pp. 335–346, 2002.
- [8] S. Amat, S. Busquier, and J. M. Gutiérrez, "Geometric constructions of iterative functions to solve nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 157, no. 1, pp. 197–205, 2003.
- [9] M. Frontini and E. Sormani, "Third-order methods from quadrature formulae for solving systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 149, no. 3, pp. 771–782, 2004.
- [10] H. H. H. Homeier, "A modified Newton method with cubic convergence: the multivariate case," *Journal of Computational and Applied Mathematics*, vol. 169, no. 1, pp. 161–169, 2004.
- [11] M. T. Darvishi and A. Barati, "A fourth-order method from quadrature formulae to solve systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 257–261, 2007.
- [12] A. Cordero and J. R. Torregrosa, "Variants of Newton's Method using fifth-order quadrature formulas," *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 686–698, 2007.
- [13] M. A. Noor and M. Waseem, "Some iterative methods for solving a system of nonlinear equations," *Computers and Mathematics with Applications*, vol. 57, no. 1, pp. 101–106, 2009.
- [14] A. Cordero, E. Martínez, and J. R. Torregrosa, "Iterative methods of order four and five for systems of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 231, no. 2, pp. 541–551, 2009.
- [15] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "A modified Newton-Jarratt's composition," *Numerical Algorithms*, vol. 55, no. 1, pp. 87–99, 2010.
- [16] M. Grau-Sánchez, Á. 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.
- [17] M. Grau-Sánchez, Á. Grau, and M. Noguera, "Ostrowski type methods for solving systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 218, no. 6, pp. 2377–2385, 2011.
- [18] J. R. Sharma, R. K. Guha, and R. Sharma, "An efficient fourth order weighted-Newton method for systems of nonlinear equations," *Numerical Algorithms*, vol. 62, no. 2, pp. 307–323, 2013.
- [19] S. Wolfram, *The Mathematica Book*, Wolfram Media, Champaign, Ill, USA, 5th edition, 2003.
- [20] M. S. Petković, "Remarks on 'on a general class of multipoint root-finding methods of high computational efficiency,'" *SIAM Journal on Numerical Analysis*, vol. 49, no. 3, pp. 1317–1319, 2011.
- [21] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann, "MPFR: a multiple-precision binary floating-point library with correct rounding," *ACM Transactions on Mathematical Software*, vol. 33, no. 2, Article ID 1236468, p. 15, 2007.
- [22] <http://www.mpf.fr/mpfr-2.1.0/timings.html>.