Research Article

A Class of Steffensen-Type Iterative Methods for Nonlinear Systems

F. Soleymani,[1] M. Sharifi,[1] S. Shateyi,[2] and F. Khaksar Haghani[1]

[1] *Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran*
[2] *Department of Mathematics, University of Venda, Private Bag Box X5050, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

A class of iterative methods without restriction on the computation of Fréchet derivatives including multisteps for solving systems of nonlinear equations is presented. By considering a frozen Jacobian, we provide a class of m-step methods with order of convergence $m + 1$. A new method named as Steffensen-Schulz scheme is also contributed. Numerical tests and comparisons with the existing methods are included.

1. Introduction

In this paper, we take into account the following system of nonlinear equations:

$$
\begin{aligned}
f_1 (x_1, x_2, \ldots, x_N) &= 0, \\
f_2 (x_1, x_2, \ldots, x_N) &= 0, \\
&\vdots \\
f_N (x_1, x_2, \ldots, x_N) &= 0,
\end{aligned}
\tag{1}
$$

wherein each function f_i maps a vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ of the N-dimensional space $\mathbb{R}^N$ into the real line $\mathbb{R}$. This system contains N nonlinear equations with N unknowns and could be expressed in a more simplified form by defining a function F mapping $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x}))^T$. Hence, the nonlinear system (1) can be written in the form $F(\mathbf{x}) = 0$, where the functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_N(\mathbf{x})$ are the coordinate functions of F; see for more details [1] and also for application issues the work [2].

We assume that $F(\mathbf{x})$ is a smooth function of $\mathbf{x}$ in the open convex set $D \subseteq \mathbb{R}^N$. There is a strong incentive to use derivative information as well as function values in order to solve traditionally the system (1). The most famous solver for such a problem is Newton's iteration [1], which is defined as

$$
\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - F'\left(\mathbf{x}^{(n)}\right)^{-1} F\left(\mathbf{x}^{(n)}\right), \quad n = 0, 1, 2, \ldots. \tag{2}
$$

In this iterative method, we use the $N \times N$ Jacobian matrix, that is, $F'(\mathbf{x})$, with entries $F'(\mathbf{x})_{jk} = \partial_{x_k} f_j(\mathbf{x})$. To be more precise, there are plenty of solvers to tackle the problem (1) or its scalar case, such as those in [3–6]. Among such methods, the third-order iterative methods like the Halley and Chebyshev methods [7] are considered less practically from a computational point of view because they need to compute the expensive second-order Fréchet derivatives in contrast to the quadratically convergent method (2), in which the first-order Fréchet derivative needs to be calculated. As a matter of fact, for the considered problem (1), the first Fréchet derivative is a matrix with N^2 entries, while the second-order Fréchet derivative has N^3 entries (without considering the symmetry). On this account, it needs a large amount of operations in order to evaluate the second derivative for each iteration.

To avoid using the Jacobian whose computation is time-consuming for large scale systems, in Chapter 7 of

[8], authors presented the definition of divided difference in N-dimensional space as an $N \times N$ matrix with elements

$$
\begin{aligned}
[x, y; f]_{i,j} = & \left(f_i \left(x_1, \ldots, x_j, y_{j+1}, \ldots, y_N \right) \right. \\
& \left. - f_i \left(x_1, \ldots, x_{j-1}, y_j, \ldots, y_N \right) \right) \\
& \times \left(x_j - y_j \right)^{-1},
\end{aligned}
\tag{3}
$$

to prevent computing the first-order Fréchet derivative. Note that (3) is a component-to-component definition. Traub in the pioneer book [1] introduced another tool named as $J(x, H)$ to estimate the Jacobian matrix and to derive Steffensen's method for nonlinear systems as follows:

$$
\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - J\left(\mathbf{x}^{(n)}, H^{(n)} \right)^{-1} F\left(\mathbf{x}^{(n)} \right), \quad n = 0, 1, 2, \ldots,
\tag{4}
$$

wherein

$$
\begin{aligned}
J\left(\mathbf{x}^{(n)}, H^{(n)} \right) = & \left(F\left(\mathbf{x}^{(n)} + H^{(n)} e^1 \right) - F\left(\mathbf{x}^{(n)} \right), \ldots, \right. \\
& \left. F\left(\mathbf{x}^{(n)} + H^{(n)} e^N \right) - F\left(\mathbf{x}^{(n)} \right) \right) H^{(n)^{-1}},
\end{aligned}
\tag{5}
$$

with $H^{(n)} = \mathrm{diag}(f_1(\mathbf{x}^{(n)}), \ldots, f_N(\mathbf{x}^{(n)}))$.

The primary aim of the present study is to achieve high rate of convergence in order to solve (1) using (5). Hence, we first propose a new iterative method with fourth-order convergence to find both real and complex solutions. The new method does not even need the evaluation of one first-order Fréchet derivative, let alone the higher-order ones. Next, by considering a frozen Jacobian matrix, we suggest a general m-step class of iterative methods with arbitrary order of convergence and higher computational efficiency index.

The rest of this paper is prepared as follows. In Section 2, the construction of a scheme is offered. It also includes the analysis of convergence and shows that the suggested method has fourth order. In Section 3, we will extend the new method to present a multistep class of iterations. Section 4 contains a discussion about the implementation and the efficiency of the iterative methods. Section 5 contains a contribution of this study by introducing Steffensen-Schulz iterative method for the first time. This is followed by Section 6 where some numerical tests will be furnished to illustrate the accuracy and efficiency of the proposed approach. Section 7 ends the paper where short conclusions of the study by pointing out future research aspects are given.

2. Derivation

In [9], authors presented the following iterative method:

$$
\mathbf{y}^{(n)} = \mathbf{x}^{(n)} - \left[\mathbf{x}^{(n)}, \mathbf{w}^{(n)}; F \right]^{-1} F\left(\mathbf{x}^{(n)} \right), \quad n = 0, 1, 2, \ldots,
$$

$$
\begin{aligned}
\mathbf{x}^{(n+1)} = & \mathbf{y}^{(n)} - \left[\mathbf{x}^{(n)}, \mathbf{y}^{(n)}; F \right]^{-1} \\
& \times \left(\left[\mathbf{x}^{(n)}, \mathbf{y}^{(n)}; F \right] - \left[\mathbf{y}^{(n)}, \mathbf{w}^{(n)}; F \right] \right. \\
& \left. + \left[\mathbf{x}^{(n)}, \mathbf{w}^{(n)}; F \right] \right) \\
& \times \left[\mathbf{x}^{(n)}, \mathbf{y}^{(n)}; F \right]^{-1} F\left(\mathbf{y}^{(n)} \right),
\end{aligned}
\tag{6}
$$

wherein $\mathbf{w}^{(n)} = \mathbf{x}^{(n)} + F(\mathbf{x}^{(n)})$ and it possesses the fourth-order convergence to approximate the simple solution of the system (1). As could be seen, it includes the evaluations $F(\mathbf{x}^{(n)}), F(\mathbf{y}^{(n)}), F(\mathbf{w}^{(n)})$ and three divided difference operators of order one based on (5), to possess the fourth-order convergence. This process is costly for challenging systems of nonlinear equations.

Now, in order to reach the fourth order of convergence without imposing the computation of even first-order Fréchet derivative or three divided difference operators, we consider a three-step structure with the same correcting factor as follows:

$$
\begin{aligned}
\mathbf{y}^{(n)} &= \mathbf{x}^{(n)} - M^{(n)^{-1}} F\left(\mathbf{x}^{(n)} \right), \\
\mathbf{z}^{(n)} &= \mathbf{y}^{(n)} - M^{(n)^{-1}} F\left(\mathbf{y}^{(n)} \right), \\
\mathbf{x}^{(n+1)} &= \mathbf{z}^{(n)} - M^{(n)^{-1}} F\left(\mathbf{z}^{(n)} \right),
\end{aligned}
\tag{7}
$$

wherein we could use the component-by-component approximation (3) [10]:

$$
M^{(n)} = \left[\mathbf{x}^{(n)}, \mathbf{w}^{(n)}; F \right],
\tag{8}
$$

or the estimation introduced by Traub (5) as follows:

$$
M^{(n)} = J\left(\mathbf{x}^{(n)}, H^{(n)} \right).
\tag{9}
$$

Note that (8) and (9) are not equal. Per computing step, the method (6) requires computing F at three different points without the computation of the Fréchet derivatives which are costly for large scale problems.

The following theorem will be demonstrated by means of the N-dimensional Taylor expansion of the functions using the estimation (8) in (7). We here include some of the basic notions which are important in the proof. Let $F : D \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ be sufficiently Fréchet differentiable in D. By using the notation introduced in [11], the qth derivative of F at $u \in \mathbb{R}^N$, $q \geq 1$, is the q-linear function $F^{(q)}(u) : \mathbb{R}^N \times \cdots \times \mathbb{R}^N \rightarrow \mathbb{R}^N$ such that $F^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^N$. Thus, we have the following:

(1) $F^{(q)}(u)(v_1, \ldots, v_{q-1}, \cdot) \in \mathscr{L}(\mathbb{R}^N)$,

(2) $F^{(q)}(u)(v_{\sigma(1)}, \ldots, v_{\sigma(q)}) = F^{(q)}(u)(v_1, \ldots, v_q)$, for all permutation σ of $\{1, 2, \ldots, q\}$.

Hence, we take into account $F^{(q)}(u)(v_1, \ldots, v_q) = F^{(q)}(u)v_1, \ldots, v_q$, and also $F^{(q)}(u)v^{q-1}F^{(p)}v^p = F^{(q)}(u)F^{(p)}(u)v^{q+p-1}$. It is well known that, for $\mathbf{x}^* + h \in \mathbb{R}^N$ lying in a neighborhood of a solution $\mathbf{x}^*$ of the nonlinear system $F(\mathbf{x}) = 0$, Taylor's expansion might be written as follows (assuming that the Jacobian matrix $F'(\mathbf{x}^*)$ is nonsingular): $F(\mathbf{x}^* + h) = F'(\mathbf{x}^*)[h + \sum_{q=2}^{p-1} C_q h^q] + O(h^p)$, where $C_q = (1/q!)[F'(\mathbf{x}^*)]^{-1}F^{(q)}(\mathbf{x}^*)$, $q \geq 2$. We observe that $C_q h^q \in \mathbb{R}^N$ since $F^{(q)}(\mathbf{x}^*) \in \mathscr{L}(\mathbb{R}^N \times \cdots \times \mathbb{R}^N, \mathbb{R}^N)$ and $[F'(\mathbf{x}^*)]^{-1} \in \mathscr{L}(\mathbb{R}^N)$.

In addition, we can express F' as $F'(\mathbf{x}^* + h) = F'(\mathbf{x}^*)[I + \sum_{q=2}^{p-1} qC_q h^{q-1}] + O(h^p)$, wherein I is the identity matrix of the same order to the Jacobian matrix. Therefore, $qC_q h^{q-1} \in \mathscr{L}(\mathbb{R}^N)$. We also in the sequel denote $e^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}^*$ as the error in the nth iteration. The equation

$$e^{(n+1)} = Le^{(n)p} + O\left(e^{(n)p+1}\right), \tag{10}$$

where L is a p-linear function $L \in \mathscr{L}(\mathbb{R}^N \times \cdots \times \mathbb{R}^N, \mathbb{R}^N)$, is called the *error equation* and p is the *order of convergence*.

Remark 1. $e^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}^*$ which is the *error* in the nth iteration is a vector and $e^{(n)p} = (e^{(n)}, e^{(n)}, \ldots, e^{(n)})$ would be a matrix.

Theorem 2. *Let $F : D \subseteq \mathbb{R}^N \to \mathbb{R}^N$ be sufficiently Fréchet differentiable at each point of an open convex neighborhood D of $\mathbf{x}^* \in \mathbb{R}^N$, that is, a simple solution of the system $F(\mathbf{x}) = 0$. Let one suppose that $F'(\mathbf{x})$ is continuous and nonsingular in $\mathbf{x}^*$. Then the sequence $\{\mathbf{x}^{(n)}\}_{n \geq 0}$ obtained using the iterative method (7) using (8) converges to $\mathbf{x}^*$ with convergence rate 4, and the error equation reads*

$$e^{(n+1)} = C_2^3 \left(I + F'(\mathbf{x}^*)\right)\left(2I + F'(\mathbf{x}^*)\right)^2 e^{(n)4} + O\left(e^{(n)5}\right). \tag{11}$$

Proof. Similar notation and terminology as in [11], yields to $F(\mathbf{x}^{(n)}) = F'(\mathbf{x}^*)[e^{(n)} + C_2 e^{(n)2} + C_3 e^{(n)3} + C_4 e^{(n)4}] + O(e^{(n)5})$, and, by taking into consideration $b^{(n)} = e^{(n)} + F(e^{(n)})$, we write

$$F\left(\mathbf{w}^{(n)}\right) = F'(\mathbf{x}^*)\left[b^{(n)} + C_2 b^{(n)2} + C_3 b^{(n)3} + C_4 b^{(n)4}\right] + O\left(b^{(n)5}\right), \tag{12}$$

where $C_k = (1/k!)[F'(\mathbf{x}^*)]^{-1}F^{(n)}(\mathbf{x}^*)$, $k = 2, 3, \ldots$. Then,

$$\left[\mathbf{x}^{(n)}, \mathbf{w}^{(n)}; F\right]^{-1} F\left(\mathbf{x}^{(n)}\right) = e^{(n)1} - C_2\left(I + F'(\mathbf{x}^*)\right)e^{(n)2}$$
$$- \left(C_3\left(I + F'(\mathbf{x}^*)\right)\left(2I + F'(\mathbf{x}^*)\right)\right.$$
$$\left. -C_2^2\left(2I + F'(\mathbf{x}^*)\left(2I + F'(\mathbf{x}^*)\right)\right)\right)e^{(n)3}$$
$$+ O\left(e^{(n)4}\right), \tag{13}$$

and subsequently the expression for $\mathbf{y}^{(n)}$ would be

$$\mathbf{y}^{(n)} - \mathbf{x}^* = C_2\left(I + F'(\mathbf{x}^*)\right)e^{(n)2} + \cdots + O\left(e^{(n)4}\right). \tag{14}$$

The Taylor expansion in the second step using (14) yields

$$\mathbf{z}^{(n)} - \mathbf{x}^* = C_2^2\left(I + F'(\mathbf{x}^*)\right)\left(2I + F'(\mathbf{x}^*)\right)e^{(n)3} + \cdots + O\left(e^{(n)5}\right). \tag{15}$$

Therefore,

$$F\left(\mathbf{z}^{(n)}\right) = F'(\mathbf{x}^*)\left[C_2^2\left(I + F'(\mathbf{x}^*)\right)\left(2I + F'(\mathbf{x}^*)\right)e^{(n)3}\right. + \cdots + O\left(e^{(n)5}\right)\right]. \tag{16}$$

Next, from an analogous reasoning as in (15) and (16), we obtain the error equation (11). Consequently, taking into account (11), it can be concluded that the order of convergence of the proposed method is four. □

Our most important class of methods is to derive a class of iterations free from derivatives using the estimation (5). For example, the method (7) using (9) results in

$$\mathbf{y}^{(n)} = \mathbf{x}^{(n)} - J\left(\mathbf{x}^{(n)}, H^{(n)}\right)^{-1} F\left(\mathbf{x}^{(n)}\right),$$
$$\mathbf{z}^{(n)} = \mathbf{y}^{(n)} - J\left(\mathbf{x}^{(n)}, H^{(n)}\right)^{-1} F\left(\mathbf{y}^{(n)}\right), \tag{17}$$
$$\mathbf{x}^{(n+1)} = \mathbf{z}^{(n)} - J\left(\mathbf{x}^{(n)}, H^{(n)}\right)^{-1} F\left(\mathbf{z}^{(n)}\right).$$

Note that it could be shown, in a similar way to the previous theorem, that (17) possesses fourth order of convergence. The implementation of (17) depends on the involved linear algebra problems. An interesting point in the new method (17) is that the LU decomposition of J needs to be done only once, and it could effectively be used three times per computing step to increase the rate of convergence without imposing much computational burden.

3. An m-Step Class

This section presents a general class of multistep iteration methods. In fact, the new scheme (17) can simply be improved by considering the Jacobian matrix $J(\mathbf{x}^{(n)}, H^{(n)})$ to be frozen. In such a way, we are able to propose a general m-step multipoint class of iterative methods in the following structure:

$$\vartheta_1^{(n)} = \mathbf{x}^{(n)} - \varpi_1^{(n)},$$
$$\vartheta_2^{(n)} = \vartheta_1^{(n)} - \varpi_2^{(n)},$$
$$\vdots \tag{18}$$
$$\mathbf{x}^{(n+1)} = \vartheta_m^{(n)} = \vartheta_{m-1}^{(n)} - \varpi_m^{(n)},$$
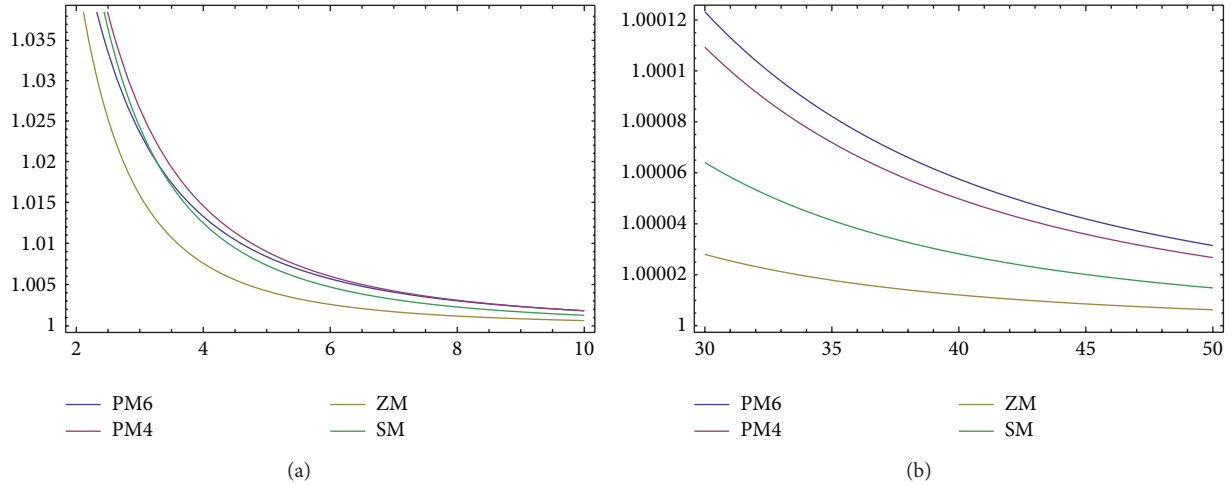
FIGURE 1: The log plot of the efficiency indices for different methods (a) when $N = 4, \ldots, 10$ and (b) when $N = 30, \ldots, 50$.

wherein $\vartheta_i^{(n)}$ is used in the linear system $J(\mathbf{x}^{(n)}, H^{(n)})\varpi_i^{(n)} = F(\vartheta_{i-1}^{(n)})$, $i = 1, \ldots, m$. We remark that, in this structure, the LU factorization of the Jacobian matrix would be computed only once. This reduces the computational load of the linear algebra problems in implementing (18).

In the iterative process (18) each added step will impose one more N-dimensional function, whose cost is N while the convergence order will be improved to $1 + O(m - 1)$, wherein $O(m - 1)$ is the order of the previous substeps. Considering the well-known mathematical induction, it would be easy to deduce the following Theorem for (18).

Theorem 3. *Using the same conditions as in Theorem 2, the m-step iterative process* (18) *has the local convergence order $m + 1$ using $m + 1$ evaluations of the function F and one first-order divided difference operator per full iterations.*

Proof. The proof of this theorem is based on mathematical induction and is straightforward. □

As an example, the five-step sixth-order method from the new class has the following structure:

$$
\begin{aligned}
\vartheta_1^{(n)} &= \mathbf{x}^{(n)} - \varpi_1^{(n)}, \\
\vartheta_2^{(n)} &= \vartheta_1^{(n)} - \varpi_2^{(n)}, \\
\vartheta_3^{(n)} &= \vartheta_2^{(n)} - \varpi_3^{(n)}, \\
\vartheta_4^{(n)} &= \vartheta_3^{(n)} - \varpi_4^{(n)}, \\
\mathbf{x}^{(n+1)} &= \vartheta_5^{(n)} = \vartheta_4^{(n)} - \varpi_5^{(n)}.
\end{aligned}
\tag{19}
$$

4. Complexity

In the iterative method (17), one may solve one linear system of equations per computing step, with three right-hand side vectors $F(\mathbf{x}^{(n)})$, $F(\mathbf{y}^{(n)})$, and $F(\mathbf{z}^{(n)})$, and a similar procedure must be applied for (19). In such a case, one could compute a factorization of the matrix and use it repeatedly. It is known that the cost (number of products/quotients) of solving the associated linear system by LU decomposition is $(1/3)N^3 + N^2 - (1/3)N$ (including the LU factorization and two triangular systems), where N is the size of the system. Moreover, if one has k systems with the same matrix, then the final cost is $(1/3)N^3 + kN^2 - (1/3)N$.

The computational cost for (17) is as follows: N evaluations of scalar functions for $F(\mathbf{x})$, N evaluations of scalar functions $F(\mathbf{y})$, and N evaluations of scalar functions for $F(\mathbf{w})$ and $N^2 - N$ (since we have computed above $F(\mathbf{x})$ and $F(\mathbf{w})$ before) for the estimation (5).

We provide a comparison of efficiency indices for the methods (2) denoted by NM, (4) denoted by SM, (6) denoted by ZM, and the new methods (17) denoted by PM4 and (19) denoted by PM6 based on the computational efficiency index which is also known as operational-efficiency index in the literature [11].

The log plot of the efficiencies according to the definition of this index for an iterative method, which is given by $E = p^{1/C}$, where p is the order of convergence and C stands for the total computational cost per iteration in terms of the number of functional evaluations and the number of products/quotients in finding the LU decomposition and two triangular systems, is given in Figure 1. It is clearly obvious that the new method (19) for higher N has dominance with respect to the other well-known methods.

Note that $E_{\mathrm{NM}} = E_{\mathrm{SM}} = 2^{3/N(2+6N+N^2)}$, $E_{\mathrm{ZM}} = 4^{3/N(-2+15N+5N^2)}$, and for the proposed methods $E_{\mathrm{PM4}} = 4^{3/N(8+12N+N^2)}$ and $E_{\mathrm{PM6}} = 6^{3/N(14+18N+N^2)}$.

Although the new scheme does not have the drawbacks of NM or ZM in terms of computing Fréchet derivatives or three different divided difference operators, a significant focus should be allocated for large scale nonlinear systems. In fact, for large scale sparse nonlinear systems of equations, we have two main shortcomings, first putting numeric values into the Jacobian matrix for Newton-type methods and second the
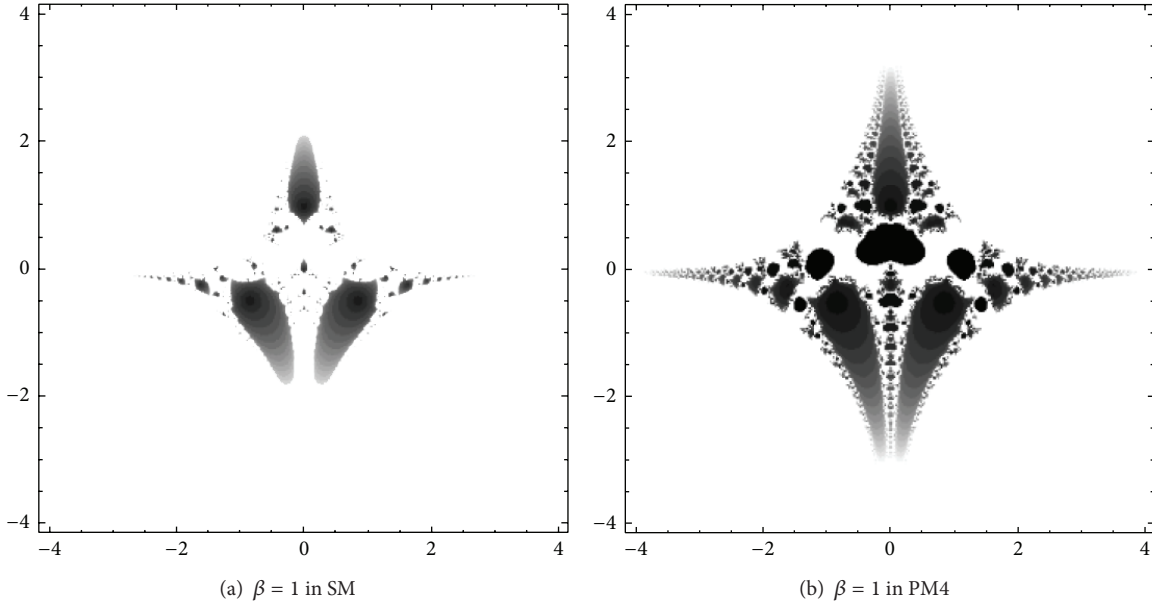
(a) $\beta = 1$ in SM

(b) $\beta = 1$ in PM4

FIGURE 2: Attraction basins of the polynomial $z^3 - 1 = 0$ in the complex plane (shaded according to the number of iterations).

LU decomposition, which is time-consuming. The suggested method in this work does not need the computations of the Fréchet derivatives and, due to this, it resolves the first problem. For solving the second problem, the new scheme should be coded in the inexact form (see, e.g., [12, 13]) using very efficient solvers for the solution of linear systems of equations such as GMRES.

5. A Multiplication-Rich Scheme

A contribution of this study lies in a simple trick that could be used in this section to produce derivative-free schemes which are inversion-free as well. As a matter of fact, the general class (18) is free from Fréchet derivative per computing step and this makes it nice but we could use a simple trick to make the process inversion-free. That is to say, the computation of the inverse of J in (18) is tough since it could be a dense matrix. Although we use the LU decomposition to proceed per cycle, we could apply an iterative inverse finder to avoid the computation of inverse (or solving a system) by imposing further matrix-matrix multiplications.

Such a procedure could be done using the well-known Schulz-type methods (see, e.g., [14]). Here, we apply the Schulz inverse finder for this purpose, for one step of the matrix iterative method (18) as follows:

$$P_n = \frac{2}{\sigma_1^2 + \sigma_r^2} J_n^*,$$

$$P_{n+1} = P_n \left(2I - J_n P_n \right),$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - P_{n+1} F \left(\mathbf{x}^{(n)} \right),$$

(20)

wherein $J_n = J(\mathbf{x}^{(n)}, H^{(n)}) = (F(\mathbf{x}^{(n)} + H^{(n)}e^1) - F(\mathbf{x}^{(n)}), \dots, F(\mathbf{x}^{(n)} + H^{(n)}e^N) - F(\mathbf{x}^{(n)}))H^{(n)^{-1}}$ and σ_1, σ_r are the largest

and smallest singular values of J. Unfortunately, convergence of this iterative scheme happens for initial matrices *so* close to the zeros and it might not be quadratic. In fact, the Steffensen method (SM) and its variants obtained by the class (18), for instance, the approach (19), should become inversion-free easily and efficiently.

We now illustrate the basins of attraction for SM2 and PM4 in the complex square of $[-4, 4] \times [-4, 4]$, when the stopping criterion is $|x_{n+1} - x_n| \leq 10^{-6}$. Hopefully, the convergence radius for tackling nonlinear problems can become broadened by introducing the free nonzero parameter β, which would be a constant array in the multivariate case [9]. Introducing β yields the following form of Steffensen's method:

$$H^{(n)} = \text{diag} \left(\beta f_1 \left(\mathbf{x}^{(n)} \right), \dots, \beta f_N \left(\mathbf{x}^{(n)} \right) \right),$$

$$J \left(\mathbf{x}^{(n)}, \beta H^{(n)} \right)$$

$$= \left(F \left(\mathbf{x}^{(n)} + H^{(n)} e^1 \right) - F \left(\mathbf{x}^{(n)} \right), \dots, \right.$$

$$\left. F \left(\mathbf{x}^{(n)} + H^{(n)} e^N \right) - F \left(\mathbf{x}^{(n)} \right) \right) H^{(n)^{-1}},$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - J \left(\mathbf{x}^{(n)}, \beta H^{(n)} \right)^{-1} F \left(\mathbf{x}^{(n)} \right).$$

(21)

Figure 2(a) shows the SM without imposing small values for β, that is, with $\beta = 1$, and Figure 2(b) is the basins of attraction of PM4. Figure 3 shows the basins of attraction of SM and PM4, but with small values of β.

It is clear that choosing small value for β results in larger basins of attraction [15]. Hence, we revise this idea

(a) $\beta = 0.0001$ in SM
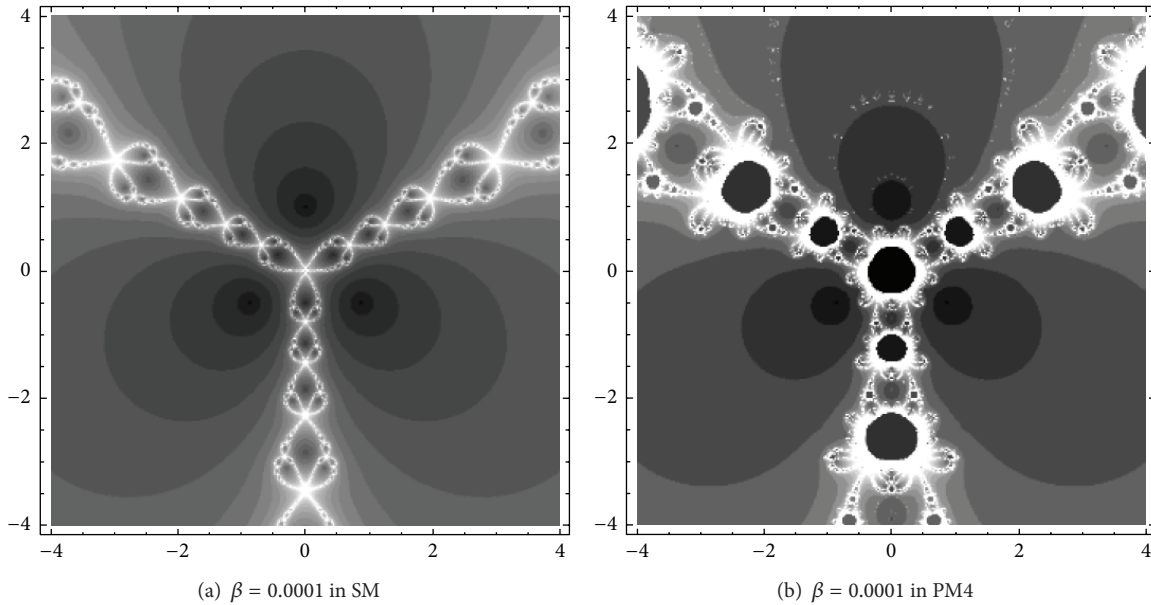

(b) $\beta = 0.0001$ in PM4

Figure 3: Attraction basins of the polynomial $z^3 - 1 = 0$ in the complex plane (shaded according to the number of iterations).

and propose an inversion-free method for solving nonlinear systems of equation in what follows:

choose a small value for β,

$$
\begin{aligned}
&\text{compute } \mathbf{x}^{(1)} \text{ by (21) and set } T_1 = J\left(\mathbf{x}^{(0)}, \beta H^{(0)}\right)^{-1}, \\
&T_{n+1} = T_n\left(2I - J_n T_n\right), \quad n = 1, 2, \ldots, \\
&\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - T_{n+1} F\left(\mathbf{x}^{(n)}\right),
\end{aligned}
\tag{22}
$$

wherein $J_n = J(\mathbf{x}^{(n)}, \beta H^{(n)})$.

We name this multiplication-rich method as the Steffensen-Schulz iteration since it is derivative-free and inversion-free at the same time. It only requires one matrix inversion in the whole process of computing $\mathbf{x}^{(1)}$.

6. Numerical Testing

We employ here the second-order method of Newton (2), the second-order scheme of Steffensen (4), the fourth-order scheme of Zheng et al. (6), and the proposed fourth-order derivative-free method (17) to compare the numerical results obtained from these methods in solving test nonlinear systems. We also put (22) into test for Test Problems 1 and 4.

Test 1. As the first problem, we take into account the following hard system of 10 nonlinear equations with 10 unknowns:

$$
5 \exp\left(x_1 - 2\right) x_2 + 8 x_3^{x_4} - 5 x_6^3 + 2 x_7^{x_{10}} - x_9 = 0,
$$

$$
5 \tan\left(x_1 + 2\right) + x_2^3 + 7 x_3^4 - 2 \sin\left(x_6\right)^3 + \cos\left(x_9^{x_{10}}\right) = 0,
$$

$$
x_1^2 + \tan\left(x_2\right) + 2 x_3^{x_4} - 5 x_6^3 - x_5 x_6 x_7 x_8 x_9 x_{10} = 0,
$$

$$
2 \tan\left(x_1^2\right) + 2^{x_2} + x_3^2 - 5 x_5^3 - x_6 + x_8^{\cos(x_9)} = 0,
$$

$$
10 x_1^2 + \cos\left(x_2\right) + x_3^2 - 5 x_6^3 - 4^{x_9} - 2 x_8 - x_{10} = 0,
$$

$$
\arccos\left(x_1^2\right) \sin\left(x_2\right) + x_3^2 - 2 x_5^4 x_6 x_9 x_{10} = 0,
$$

$$
x_1 x_2^{x_7} + x_3^5 - 5 x_5^3 + x_7 - x_8^{x_{10}} = 0,
$$

$$
x_4 \sin\left(x_2\right) + x_3 - 15 x_5^2 + x_7 + \arccos\left(x_8 + x_9 - 10 x_{10}\right) = 0,
$$

$$
10 x_1 + x_3^2 - 5 x_5^2 + 10 x_6^{x_8} + 2 x_9 - \sin\left(x_7\right) = 0,
$$

$$
x_1 \sin\left(x_2\right) - 5 x_6 - 2 x_{10}^{x_8} - 10 x_9 + x_{10} = 0.
\tag{23}
$$

In this test problem, the approximate solution up to 5 decimal places is the following vector: $\mathbf{x}^* \approx (1.88885 + 0.20069I, 0.57690 - 2.01025I, 1.003311 - 0.271000I, 2.94243 + 0.83281I, 0.841597 - 0.133319I, -0.471176 + 0.882220I, 0.123992 + 0.141636I, 1.58763 - 0.37199I, 2.55259 + 0.18419I, -2.06453 + 1.58241I)^T$.

Test 2. We consider the following nonlinear system:

$$
\begin{aligned}
x_i x_{i+1} - 1 &= 0, \quad i = 1, 2, \ldots, N - 1, \\
x_N x_1 - 1 &= 0,
\end{aligned}
\tag{24}
$$

where its solution is the vector $\mathbf{x}^* = (1, \ldots, 1)^T$ for odd N.

Test 3. We consider the following large scale nonlinear system:

$$
\begin{aligned}
\left(x_i x_{i+1}\right)^2 - 3 &= 0, \quad i = 1, 2, \ldots, N - 1, \\
x_N\left(x_1\right)^2 - 1 &= 0,
\end{aligned}
\tag{25}
$$

TABLE 1: Results of comparisons for different methods in Test 1 using $\mathbf{x}^{(0)} = (1.88 + 0.2I, 0.57 - 2.01I, 1.00 - 0.27I, 2.94 + 0.83I, 0.84 - 0.13I, -0.47 + 0.88I, 0.12 + 0.14I, 1.58 - 0.37I, 2.55 + 0.18I, -2.06 + 1.58I)^T$.

Iterative methods	NM	SM	ZM	PM4
Number of iterations	7	9	4	5
The residual norm	3.73×10^{-164}	1.49×10^{-153}	4.14×10^{-149}	9.37×10^{-173}
COC	2.01	1.99	3.99	4.00
The elapsed time	0.35	1.00	1.23	0.37

TABLE 2: Results of comparisons for different methods in Test 2 using $\mathbf{x}^{(0)} = (2, \ldots, 2)$ and $N = 99$.

Iterative methods	NM	SM	ZM	PM4
Number of iterations	8	8	5	5
The residual norm	2.86×10^{-121}	2.86×10^{-121}	0	0
COC	2.00	2.00	4.00	4.00
The elapsed time	0.70	0.79	2.82	0.65

TABLE 3: Results of comparisons for different methods in Test 3 using $\mathbf{x}^{(0)} = (2, \ldots, 2)$ and $N = 200$.

Iterative methods	NM	SM	ZM	PM4
Number of iterations	9	17	5	7
The residual norm	2.56×10^{-110}	1.24×10^{-126}	5.66×10^{-75}	2.13×10^{-107}
COC	1.97	2.00	2.69	3.97
The elapsed time	5.95	8.59	23.23	4.95

where its solution is the vector $\mathbf{x}^* = (0.57735, 3.0000, 0.57735, 3.0000, \ldots, 0.57735, 3.0000)^T$.

We report the numerical results for solving Tests 1–3 in Tables 1–3 based on the initial values. Note that an important aspect in implementing iterative method for solving nonlinear systems is to find a robust initial guess to guarantee the convergence. Some discussions regarding this are given in [16, 17].

The residual norm along with the number of iterations and computational time using the command Timing in Mathematica 8 [18] is reported in Tables 1–3. The computer specifications are Microsoft Windows XP Intel(R), Pentium(R) 4 CPU, and 3.20 GHz with 4 GB of RAM.

An efficient way to observe the behavior of the order of convergence is to use the local computational order of convergence (COC) that can be defined by $\rho \approx \ln(\| F(\mathbf{x}^{(n+1)}) \| / \| F(\mathbf{x}^{(n)}) \|)/\ln(\| F(\mathbf{x}^{(n)}) \| / \| F(\mathbf{x}^{(n-1)}) \|)$, in the N-dimensional case. We have used this index in the numerical comparisons listed in Tables 1–3 for each different method to illustrate their numerical orders.

In numerical comparisons, we have chosen the fixed point arithmetic to be 200 using the command SetAccuarcy[expr,200] in the written codes. Note that, for some iterative methods, their residual norm at the specified iteration will exceed the bound of 10^{-200}; thus, we consider such approximations as the exact solution and denoted the residual norm by 0 in some cells of the tables.

Results in Table 1 show that the new scheme can be considered for complex solutions of hard nonlinear systems.

In this test, due to the fact that the dimension of the nonlinear system is low, we have used the LU decomposition to prevent the computation of 3 linear systems. Computational time reported in Table 1 verifies the fact that derivative-free methods with few number of divided difference operators are reliable. Note that Figure 4(a) reveals the residual fall of (22) for solving Test 1. It also reveals a quadratic convergence in the obtained residuals per full cycle.

In order to tackle large scale nonlinear systems, we have included Tests 2 and 3 in this work. As could be seen from Tables 2 and 3, the cases for 99×99 and 200×200 are considered, respectively. It is obvious that for large scale nonlinear systems we have two difficulties in implementation; first the LU decomposition takes time to be attained and second finding the numeric values of the Jacobian matrix in the Newton-type methods is costly. However, to have a fair comparison, we have not defined the pattern of the Jacobian in the computations and used LU decomposition in solving the considered linear systems.

Test 4. Consider the mixed Hammerstein integral equation [8]:

$$x(s) = 1 + \frac{1}{5} \int_0^1 G(s, t) x(t)^3 dt, \tag{26}$$

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

$$G(s, t) = \begin{cases} (1 - s)t, & t \le s, \\ s(1 - t), & t > s. \end{cases} \tag{27}$$

(a) Convergence history of (22) for Test 1

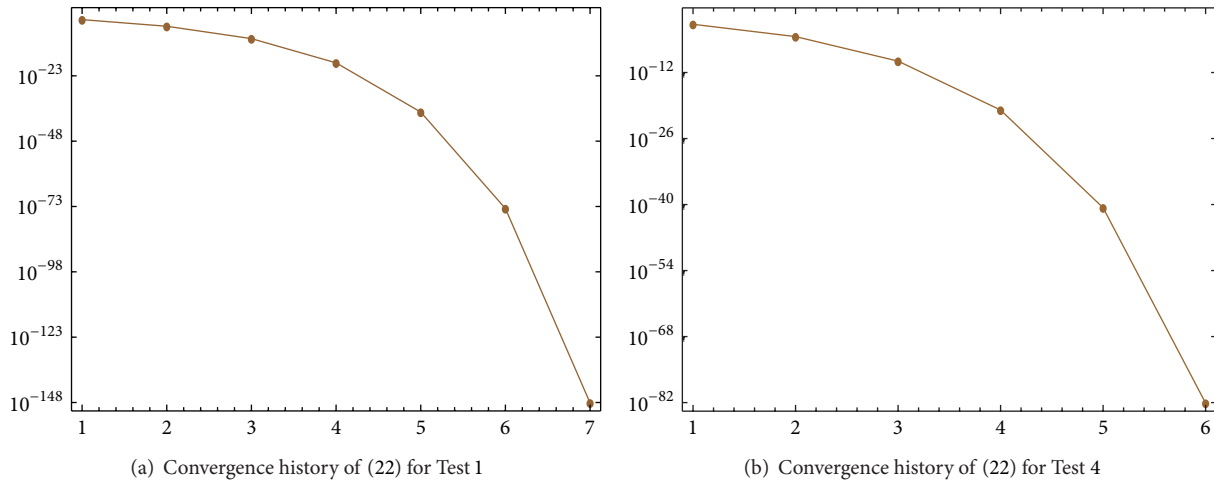(b) Convergence history of (22) for Test 4

FIGURE 4: Using $\beta = 0.0001$ in (22) so as to solve different tests.

In order to solve this nonlinear integral equation, we transform the above equation into a finite-dimensional problem by using Gauss-Legendre quadrature formula given as $\int_0^1 f(t)dt \approx \Sigma_{j=1}^t w_j f(t_j)$, where the abscissas t_j and the weights w_j are determined for $t = 50$ by Gauss-Legendre quadrature formula. Denoting the approximation of $x(t_i)$ by x_i ($i = 1, 2, \ldots, t$), we obtain the system of nonlinear equations

$$5x_i - 5 - \sum_{j=1}^t a_{ij} x_j^3 = 0, \qquad (28)$$

where, for $i = 1, 2, \ldots, t$, we have

$$a_{ij} = \begin{cases} w_j t_j \left(1 - t_i\right), & \text{if } j \leq i, \\ w_j t_i \left(1 - t_j\right), & \text{if } i < j, \end{cases} \qquad (29)$$

wherein the abscissas t_j and the weights w_j are known.

Using the initial approximation $\mathbf{x}^{(0)} = (0.5, \ldots, 0.5)^T$, we apply the proposed method (22) which is multiplication-rich to find the final solution vector of the nonlinear integral equation (28). Figure 4(b) puts on show the residual fall for solving the nonlinear integral equation (26) using (22) when $t = 50$ is the size of the nonlinear system of equations.

From numerical results in this section, it is clear that the accuracy in successive iterations increases, showing stable nature of the methods. Also, like the existing methods, the presented method shows consistent convergence behavior. From the calculation of computational order of convergence, it is also verified that order of convergence is preserved.

7. Concluding Summary

We have presented a class of iterative methods for finding the solution of nonlinear systems. The construction of the suggested scheme let us achieve high convergence order by avoiding the computation of the Jacobian matrix, whose evaluation takes time for large scale nonlinear systems.

Some different numerical tests have been used to compare the consistency and stability of the proposed iterations in contrast to the existing methods. The numerical results obtained in Section 6 reverified the theoretical aspects of the paper. We have also revealed that the methods can efficiently be used for complex zeros as well.

Further modifications could be done to make the method hybrid so as to have a trust region. In sum, we can conclude that the novel iterative methods have an acceptable performance in solving systems of nonlinear equations.

Conflict of Interests

The authors declare that they do not have any conflict of interests in their submitted paper.

Acknowledgment

References

[1] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, New York, NY, USA, 1964.

[2] E. S. Alaidarous, M. Z. Ullah, F. Ahmad, and A. S. Al-Fhaid, "An efficient higher-order quasilinearization method for solving nonlinear BVPs," *Journal of Applied Mathematics*, vol. 2013, Article ID 259371, 11 pages, 2013.

[3] Q. Zheng, F. Huang, X. Guo, and X. Feng, "Doubly-accelerated Steffensen's methods with memory and their applications on solving nonlinear ODEs," *Journal of Computational Analysis and Applications*, vol. 15, no. 5, pp. 886–891, 2013.

[4] F. Soleymani and F. Soleimani, "Novel computational derivative-free methods for simple roots," *Fixed Point Theory*, vol. 13, no. 1, pp. 247–258, 2012.

[5] F. Soleymani, "On a novel optimal quartically class of methods," *Far East Journal of Mathematical Sciences*, vol. 58, no. 2, pp. 199–206, 2011.

[6] W.-X. Wang, Y.-L. Shang, W.-G. Sun, and Y. Zhang, "Finding the roots of system of nonlinear equations by a novel filled function method," *Abstract and Applied Analysis*, vol. 2011, Article ID 209083, 9 pages, 2011.

[7] M. T. Darvishi and B.-C. Shin, "High-order Newton-Krylov methods to solve systems of nonlinear equations," *Journal of the Korean Society for Industrial and Applied Mathematics*, vol. 15, no. 1, pp. 19–30, 2011.

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.

[9] Q. Zheng, P. Zhao, and F. Huang, "A family of fourth-order Steffensen-type methods with the applications on solving nonlinear ODEs," *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8196–8203, 2011.

[10] M. Grau-Sánchez, M. Noguera, and S. Amat, "On the approximation of derivatives using divided difference operators preserving the local convergence order of iterative methods," *Journal of Computational and Applied Mathematics*, vol. 237, no. 1, pp. 363–372, 2013.

[11] 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.

[12] Z.-Z. Bai and H.-B. An, "A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations," *Applied Numerical Mathematics*, vol. 57, no. 3, pp. 235–252, 2007.

[13] S. C. Eisenstat and H. F. Walker, "Globally convergent inexact Newton methods," *SIAM Journal on Optimization*, vol. 4, no. 2, pp. 393–422, 1994.

[14] F. Soleymani and P. S. Stanimirovic, "A higher order iterative method for computing the Drazin inverse," *The Scientific World Journal*, vol. 2013, Article ID 708647, 11 pages, 2013.

[15] A. Cordero, F. Soleymani, J. R. Torregrosa, and S. Shateyi, "Basins of attraction for various Steffensen-type methods," *Journal of Applied Mathematics*, vol. 2014, Article ID 539707, 17 pages, 2014.

[16] F. Toutounian, J. Saberi-Nadjafi, and S. H. Taheri, "A hybrid of the Newton-GMRES and electromagnetic meta-heuristic methods for solving systems of nonlinear equations," *Journal of Mathematical Modelling and Algorithms*, vol. 8, no. 4, pp. 425–443, 2009.

[17] S. Wagon, *Mathematica in Action*, Springer, New York, NY, USA, 3rd edition, 2010.

[18] M. Trott, *The Mathematica Guide-Book for Numerics*, Springer, New York, NY, USA, 2006.