Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 751975, 15 pages
doi:10.1155/2012/751975

Research Article

On a New Method for Computing the Numerical Solution of Systems of Nonlinear Equations

H. Montazeri,[1] F. Soleymani,[2] S. Shateyi,[3] and S. S. Motsa[4]

[1] *Department of Mathematics, Islamic Azad University, Sirjan Branch, Sirjan, Iran*
[2] *Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran*
[3] *Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa*
[4] *School of Mathematical Sciences, University of KwaZulu-Natal, Private Bag X01, Pietermaritzburg 3209, South Africa*

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

Received 22 June 2012; Revised 17 August 2012; Accepted 27 August 2012

Academic Editor: Changbum Chun

We consider a system of nonlinear equations $F(\mathbf{x}) = 0$. A new iterative method for solving this problem numerically is suggested. The analytical discussions of the method are provided to reveal its sixth order of convergence. A discussion on the efficiency index of the contribution with comparison to the other iterative methods is also given. Finally, numerical tests illustrate the theoretical aspects using the programming package Mathematica.

1. Preliminaries

In this work, we consider the following system of nonlinear equations:

$$F(\mathbf{x}) = 0, \tag{1.1}$$

wherein the function F is $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))^T$ and the functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $\ldots$, $f_n(\mathbf{x})$ are the coordinate functions, [1].

There are many approaches to solve the system (1.1). One of the best iterative methods to challenge this problem is Newton's method. This method starts with an initial guess $\mathbf{x}^{(0)}$ and after k updates and by considering a stopping criterion satisfies (1.1).

To find an update per full cycle of such fixed point methods, the linear system(s) involved in the process should be solved by direct or indirect methods. In terms of computational point of view, when dealing with large-scale systems arising from the discretization of nonlinear PDEs or integral equations, solving the system of linear equations by a direct method such as LU decomposition is not that easy. Hence, it seems reasonable to solve the linearized system approximately using iterative methods such as GMRES. In this way, one of such approaches may be categorized as an inexact method [2]. For example, Newton's method to solve (1.1) can be written in the form below:

$$\begin{gathered}\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}, \quad k = 0, 1, 2, \ldots, \\ F'\left(\mathbf{x}^{(k)}\right)\mathbf{s}^{(k)} = -F\left(\mathbf{x}^{(k)}\right),\end{gathered} \tag{1.2}$$

and it could be seen in the inexact form by considering $\eta^{(k)}$ which satisfies

$$\left\| F'\left(\mathbf{x}^{(k)}\right)\mathbf{s}^{(k)} + F\left(\mathbf{x}^{(k)}\right) \right\| \leq \eta^{(k)} \left\| F\left(\mathbf{x}^{(k)}\right) \right\|, \tag{1.3}$$

with

$$\eta^{(k)} \in [0, 1), \tag{1.4}$$

as the forcing term [3].

A benefit of Newton's method is that the obtained sequence converges quadratically if the initial estimate is close to the exact solution. However, there are some downsides of the method. One is the selection of the initial guess. A good initial guess can lead to convergence in a couple of steps. To improve the initial guess, several ideas can be found in the literature; see, for example, [4–6].

In what follows, we assume that $F(\mathbf{x})$ is a smooth function of $\mathbf{x}$ in the open convex set $D \subseteq \mathbb{R}^n$. There are plenty of other solvers to tackle the problem (1.1); see, for example, [7–10]. Among such methods, the third-order iterative methods like the Halley and Chebyshev methods [11] are considered less practically from a computational point of view because they need to compute the expensive second-order Frechet derivatives in contrast to the quadratically convergent method (1.2), in which the first-order Frechet derivative needs to be calculated.

Let us now review some of the most current methods for solving (1.1). In 2012, Sharma et al. in [12] presented the following quadratically convergent method:

$$\begin{gathered}\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \frac{2}{3}F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right), \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{1}{2}\left[-I + \frac{9}{4}F'\left(\mathbf{y}^{(k)}\right)^{-1}F'\left(\mathbf{x}^{(k)}\right)\right. \\ \left. + \frac{3}{4}F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right)\right]F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right).\end{gathered} \tag{1.5}$$

The fourth-order method of Soleymani (the first two steps of relation (11) in [13]) for systems of nonlinear equations is as follows:

$$\begin{gathered}
\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \frac{2}{3}F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right), \\
\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left[I - \frac{3}{8}\left(I - \left(F'\left(\mathbf{y}^{(k)}\right)^{-1}F'\left(\mathbf{x}^{(k)}\right)\right)^{2}\right)\right]F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right).
\end{gathered} \tag{1.6}$$

As could be seen, (1.5) and (1.6) include the evaluations $F(\mathbf{x}^{(k)})$, $F'(\mathbf{x}^{(k)})$, and $F'(\mathbf{y}^{(k)})$.

The primary goal of the present paper is to achieve both high order of convergence and low computational load in order to solve (1.1) with a special attention to the matter of efficiency index. Hence, we propose a new iterative method with sixth-order convergence to find both real and complex solutions. The new proposed method needs not the evaluation of the second-order Frechet derivative.

The rest of this paper is prepared as follows. In Section 2, the construction of the novel scheme is offered. Section 3 includes its analysis of convergence behavior, and it shows the suggested method has sixth order. Section 4 contains a discussion on the efficiency of the iterative method. This is followed by Section 5 where some numerical tests will be furnished to illustrate the accuracy and efficiency of the method. Section 6 ends the paper, where a conclusion of the study is given.

2. The Proposed Method

This section contains the new method of this paper. We aim at having an iteration method to have high order of convergence with an acceptable efficiency index. Hence, in order to reach the sixth order of convergence without imposing the computation of further Frechet derivatives, we consider a three-step structure using a Jarratt-type method as the predictor, while the corrector step would be designed as if no new Frechet derivative has been used. Thus, we should use $F'(\mathbf{x}^{(k)})$ and $F'(\mathbf{y}^{(k)})$ in the third step, and hence we could suggest the following iteration method:

$$\begin{gathered}
\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \frac{2}{3}F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right), \\
\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \left[\frac{23}{8}I - 3F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right) + \frac{9}{8}\left(F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right)\right)^{2}\right]F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{x}^{(k)}\right), \\
\mathbf{x}^{(k+1)} = \mathbf{z}^{(k)} - \left[\frac{5}{2}I - \frac{3}{2}F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right)\right]F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{z}^{(k)}\right).
\end{gathered} \tag{2.1}$$

Per computing step of the new method (2.1), for not large-scale problems, we may use the LU decomposition to prevent the computation of the matrix inversion which is costly. Simplifying method (2.1) for the sake of implementation yields in

$$\begin{aligned}
\mathbf{y}^{(k)} &= \mathbf{x}^{(k)} - \frac{2}{3}V^{(k)}, \\
\mathbf{z}^{(k)} &= \mathbf{x}^{(k)} - \left[\frac{23}{8}I - 3M^{(k)} + \frac{9}{8}\left(M^{(k)}\right)^2\right]V^{(k)}, \\
\mathbf{x}^{(k+1)} &= \mathbf{z}^{(k)} - \left[\frac{5}{2}I - \frac{3}{2}M^{(k)}\right]W^{(k)},
\end{aligned} \tag{2.2}$$

wherein $F'(\mathbf{x}^{(k)})V^{(k)} = F(\mathbf{x}^{(k)})$, $F'(\mathbf{x}^{(k)})M^{(k)} = F'(\mathbf{y}^{(k)})$, and $F'(\mathbf{x}^{(k)})W^{(k)} = F(\mathbf{z}^{(k)})$.

Now the implementation of (2.2) depends on the involved linear algebra problems. For large-scale problems, one may apply the GMRES iterative solver which is well known for its efficiency for large sparse linear systems.

Remark 2.1. The interesting point in (2.2) is that three linear systems should be solved per computing step, but all have the same coefficient matrix. Hence, one LU factorization per full cycle is needed, which reduces the computational load of the method when implementing.

We are about to prove the convergence behavior of the proposed method (2.2) using n-dimensional Taylor expansion. This is illustrated in the next section. An important remark on the error equation in this case comes next.

Remark 2.2. In the next section, $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$ is the *error* in the kth iteration and $e^{(k+1)} = Le^{(k)^p} + O(e^{(k)^{p+1}})$ is the *error equation*, where L is a p-linear function, that is, $L \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n, \ldots, \mathbb{R}^n)$ and p is the order of convergence. Observe that $e^{(k)^p} = (e^{(k)}, e^{(k)}, \ldots, e^{(k)})$.

3. Convergence Analysis

Let us assess the analytical convergence rate of method (2.2) in what follows.

Theorem 3.1. *Let $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be sufficiently Frechet differentiable at each point of an open convex neighborhood D of $\mathbf{x}^* \in \mathbb{R}^n$, that is a 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 $\{\mathbf{x}^{(k)}\}_{k\geq 0}$ obtained using the iterative method* (2.2) *converges to $\mathbf{x}^*$ with convergence rate six, and the error equation reads*

$$e^{(k+1)} = \frac{1}{9}\left(6C_2^2 - C_3\right)\left(45C_2^3 - 9C_3C_2 + C_4\right)e^{(k)^6} + O\left(e^{(k)^7}\right). \tag{3.1}$$

Proof. Let $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be sufficiently Frechet differentiable in D. By using the notation introduced in [14], the qth derivative of F at $u \in \mathbb{R}^n$, $q \geq 1$, is the q-linear function $F^{(q)}(u) : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}^n$ such that $F^{(q)}(u)(v_1, \ldots, v_q) \in \mathbb{R}^n$. 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 expansion can be applied and we have

$$F(\mathbf{x}^* + h) = F'(\mathbf{x}^*)\left[h + \sum_{q=2}^{p-1} C_q h^q\right] + O(h^p), \tag{3.2}$$

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 $F^{(q)}(\mathbf{x}^*) \in \mathcal{L}(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$ and $[F'(\mathbf{x}^*)]^{-1} \in \mathcal{L}(\mathbb{R}^n)$. In addition, we can express F' as

$$F'(\mathbf{x}^* + h) = F'(\mathbf{x}^*)\left[I + \sum_{q=2}^{p-1} qC_q h^{q-1}\right] + O(h^p), \tag{3.3}$$

wherein I is the identity matrix of the same order to the Jacobian matrix. Note that $qC_q h^{q-1} \in \mathcal{L}(\mathbb{R}^n)$. From (3.2) and (3.3) we obtain

$$F\left(\mathbf{x}^{(k)}\right) = F'(\mathbf{x}^*)\left[e^{(k)} + C_2 e^{(k)^2} + C_3 e^{(k)^3} + C_4 e^{(k)^4} + C_5 e^{(k)^5} + C_6 e^{(k)^6}\right] + O\left(e^{(k)^7}\right), \tag{3.4}$$

$$F'\left(\mathbf{x}^{(k)}\right) = F'(\mathbf{x}^*)\left[I + 2C_2 e^{(k)} + 3C_3 e^{(k)^2} + 4C_4 e^{(k)^3} + 5C_5 e^{(k)^4} + 6C_6 e^{(k)^5}\right] + O\left(e^{(k)^6}\right), \tag{3.5}$$

where $C_k = (1/k!)[F'(\mathbf{x}^*)]^{-1}F^{(k)}(\mathbf{x}^*)$, $k = 2, 3, \ldots$, and $e^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$. From (3.5), we have

$$\left[F'\left(\mathbf{x}^{(k)}\right)\right]^{-1} = \left[I + X_1 e^{(k)} + X_2 e^{(k)^2} + X_3 e^{(k)^3} + \cdots\right][F'(\mathbf{x}^*)]^{-1} + O\left(e^{(k)^6}\right), \tag{3.6}$$

where $X_1 = -2C_2$, $X_2 = 4C_2^2 - 3C_3$, $X_3 = -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4, \ldots$. Note that based on Remark 2.2. $e^{(k)^p}$ is a singular matrix, not a vector. Then,

$$\begin{aligned}\left[F'\left(\mathbf{x}^{(k)}\right)\right]^{-1}F\left(\mathbf{x}^{(k)}\right) = {} & e^{(k)} - C_2 e^{(k)^2} + 2\left(C_2^2 - C_3\right)e^{(k)^3} + \left(-4C_2^3 + 4C_2C_3 + 3C_3C_2 - 3C_4\right)e^{(k)^4} \\ & + \left(8C_2^4 - 20C_2^2C_3 + 6C_3^2 + 10C_2C_4 - 4C_5\right)e^{(k)^5} \\ & + \left(-16C_2^5 + 52C_2^3C_3 - 33C_2C_3^2 - 28C_2^2C_4 + 17C_3C_4 + 13C_2C_5 - 5C_6\right)e^{(k)^6} \\ & + O\left(e^{(k)^7}\right),\end{aligned} \tag{3.7}$$

and the expression for $y^{(k)}$ is

$$\begin{aligned}\mathbf{y}^{(k)} = {} & \mathbf{x}^* + \frac{1}{3}e^{(k)} + \frac{2}{3}C_2 e^{(k)^2} - \frac{4}{3}\left(C_2^2 - C_3\right)e^{(k)^3} \\ & + \left(2C_4 - \frac{8}{3}C_2C_3 - 2C_3C_2 + 8C_2^3\right)e^{(k)^4} + O\left(e^{(k)^5}\right).\end{aligned} \tag{3.8}$$

The Taylor expansion of the Jacobian matrix $F'(\mathbf{y}^{(k)})$ is

$$\begin{aligned} F'\left(\mathbf{y}^{(k)}\right) &= F'(\mathbf{x}^*)\left[I + 2C_2\left(\mathbf{y}^{(k)} - \mathbf{x}^*\right) + 3C_3\left(\mathbf{y}^{(k)} - \mathbf{x}^*\right)^2\right. \\ &\qquad \left. + 4C_4\left(\mathbf{y}^{(k)} - \mathbf{x}^*\right)^3 5C_5\left(\mathbf{y}^{(k)} - \mathbf{x}^*\right)^4\right] + O\left(e^{(k)^5}\right) \\ &= F'(\mathbf{x}^*)\left[I + N_1 e^{(k)} + N_2 e^{(k)^2} + N_3 e^{(k)^3}\right] + O\left(e^{(k)^4}\right), \end{aligned} \tag{3.9}$$

where $N_1 = (2/3)C_2$, $N_2 = (4/3)C_2^2 + (1/3)C_3$, and $N_3 = -(8/3)C_2(C_2^2 - C_3) + (4/3)C_3C_2 + (4/27)C_4$. Therefore,

$$\begin{aligned} &\frac{23}{8}I - 3F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right) + \frac{9}{8}\left(F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right)\right)^2 \\ &\quad = I + C_2 e^{(k)} + \left(-C_2^2 + 2C_3\right)e^{(k)^2} + \left(-4C_2^3 - 2C_2C_3 + \frac{26}{9}C_4\right)e^{(k)^3} + O\left(e^{(k)^4}\right). \end{aligned} \tag{3.10}$$

From an analogous reasoning as in (3.9), we obtain

$$\begin{aligned} \mathbf{z}^{(k)} - \mathbf{x}^* = {} & \left(5C_2^3 - C_2C_3 + \frac{1}{9}C_4\right)e^{(k)^4} + \left(-36C_2^4 + 32C_2^2C_3 - 2C_3^2 - \frac{20}{9}C_2C_4 + \frac{8}{27}C_5\right)e^{(k)^5} \\ & + \frac{2}{27}\left(2295C_2^5 - 3537C_2^3C_3 + 633C_2^2C_4 - 99C_3C_4\right. \\ & \qquad \left. + 9C_2\left(99C_3^2 - 5C_5\right)7C_6\right)e^{(k)^6} + O\left(e^{(k)^7}\right). \end{aligned} \tag{3.11}$$

Hence, taking into account (3.11), it will be easy to write the Taylor series of $F(\mathbf{z}^{(k)})$ as follows:

$$\begin{aligned} F\left(\mathbf{z}^{(k)}\right) = {} & \frac{1}{9}\left(45C_2^3 - 9C_2C_3 + C_4\right)F'(\mathbf{x}^*)e^{(k)^4} \\ & - \frac{2}{27}\left(86C_2^4 - 144C_2^2C_3 + 27C_3^2 + 30C_2C_4\right. \\ & \qquad \left. - 4C_5\right)F'(\mathbf{x}^*)e^{(k)^5} + O\left(e^{(k)^6}\right). \end{aligned} \tag{3.12}$$

We now should find the Taylor series at the third step of (2.2), thus using (3.6) and (3.12), we have

$$\begin{aligned} F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{z}^{(k)}\right) = {} & \left(5C_2^3 - C_2C_3 + \frac{1}{9}C_4\right)e^{(k)^4} \\ & + \left(-46C_2^4 + 34C_2^2C_3 - 2C_3^2 - \frac{22}{9}C_2C_4 + \frac{8}{27}C_5\right)e^{(k)^5} \\ & + \left(262C_2^5 - 345C_2^3C_3 + 73C_2C_3^2 + \frac{466}{9}C_2^2C_4 - \frac{23}{3}C_3C_4 \right. \\ & \left. - \frac{106}{27}C_2C_5 + \frac{14}{27}C_6\right)e^{(k)^6} + O\left(e^{(k)^7}\right). \end{aligned} \tag{3.13}$$

By using (3.13), we could have

$$\begin{aligned} & \left[\frac{5}{2}I - \frac{3}{2}F'\left(\mathbf{x}^{(k)}\right)^{-1}F'\left(\mathbf{y}^{(k)}\right)\right]F'\left(\mathbf{x}^{(k)}\right)^{-1}F\left(\mathbf{z}^{(k)}\right) \\ & \quad = \left(5C_2^3 - C_2C_3 + \frac{C_4}{9}\right)e^{(k)^4} + \left(-36C_2^4 + 32C_2^2C_3 - 2C_3^2 - \frac{20}{9}C_2C_4 + \frac{8}{27}C_5\right)e^{(k)^5} \\ & \quad + \left(140C_2^5 - 251C_2^3C_3 + 65C_2C_3^2 + \frac{416}{9}C_2^2C_4 - \frac{65}{9}C_3C_4 - \frac{10}{3}C_2C_5 + \frac{14}{27}C_6\right)e^{(k)^6} + O\left(e^{(k)^7}\right). \end{aligned} \tag{3.14}$$

Combining the error equation terms (3.14) and (3.11) in the iteration method (2.2) will end in the final error equation (3.1), which shows that the new method has sixth order of convergence for solving systems of nonlinear equations. □

4. Concerning the Efficiency Index

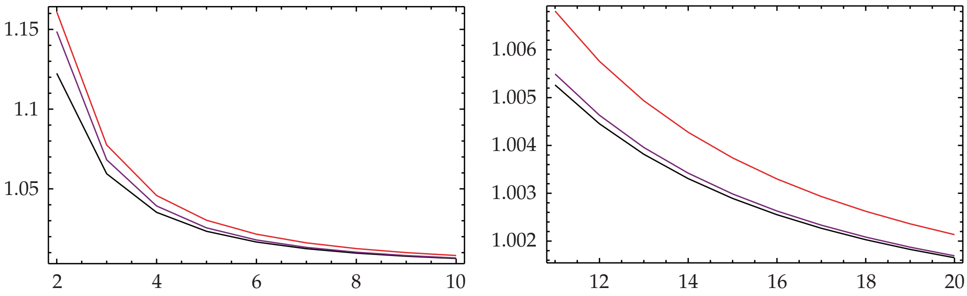
Now we assess the efficiency index of the new iterative method in contrast to the existing methods for systems of nonlinear equations. In the iterative method (2.2), three linear systems based on LU decompositions are needed to obtain the sixth order of convergence. The point is that for applying to large-scale problems one may solve the linear systems by iterative solvers and the number of linear systems should be in harmony with the convergence rate. For example, the method of Sharma et al. (1.5) requires three different linear systems for large sparse nonlinear systems, that is, the same with the new method (2.2), but its convergence rate is only 4, which clearly shows that method (2.2) is better than (1.5).

Now let us invite the number of functional evaluations to obtain the classical efficiency index for different methods. The iterative method (2.2) has the following computational cost (without the index k): n evaluations of scalar functions for $F(\mathbf{x})$, n evaluations of scalar functions $F(\mathbf{z})$, n^2 evaluations of scalar functions for the Jacobian $F'(\mathbf{x})$, and n^2 evaluations of scalar functions for the Jacobian $F'(\mathbf{y})$.

We now provide the comparison of the classical efficiency indices for methods (1.2), (1.5), and (1.6) alongside the new proposed method (2.2). The plot of the index of efficiencies according to the definition of the efficiency index of an iterative method, which is given by

Table 1: Comparison of efficiency indices for different methods.

Iterative methods	(1.2)	(1.5)	(1.6)	(2.2)
Number of steps	1	2	2	3
Rate of convergence	2	4	4	6
Number of functional evaluations	$n + n^2$	$n + 2n^2$	$n + 2n^2$	$2n + 2n^2$
The classical efficiency index	$2^{1/(n+n^2)}$	$4^{1/(n+2n^2)}$	$4^{1/(n+2n^2)}$	$6^{1/(2n+2n^2)}$
Number of LU factorizations	1	2	2	1
Cost of LU factorizations (based on flops)	$2n^3/3$	$4n^3/3$	$4n^3/3$	$2n^3/3$
Cost of linear systems (based on flops)	$(2n^3/3) + 2n^2$	$(10n^3/3) + 2n^2$	$(7n^3/3) + 2n^2$	$(5n^3/3) + 4n^2$
Flops-like efficiency index	$2^{1/((2n^3/3)+3n^2+n)}$	$4^{1/((10n^3/3)+4n^2+n)}$	$4^{1/((7n^3/3)+4n^2+n)}$	$6^{1/((5n^3/3)+6n^2+2n)}$



(a)

(b)

Figure 1: The plot of the (traditional) efficiency indices for different methods ((a), for $n = 2, \ldots, 10$ and (b), for $n = 11, \ldots, 20$).

$E = p^{1/C}$, 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, is given in Figure 1. A comparison over the number of functional evaluations of some iterative methods is also illustrated in Table 1. Note that both (1.5) and (1.6) in this way have the same classical efficiency index.

In Figures 1 and 2, the colors yellow, black, purple, and red stand for methods (1.6), (1.2), (1.5), and (2.2), respectively. It is clearly obvious that the new method (2.2) for any $n \geq 2$ has dominance with respect to the traditional efficiency indices on the other well-known and recent methods.

As was positively pointed out by the second referee, taking into account only the number of evaluations for scalar functions cannot be the effecting factor for evaluating the efficiency of nonlinear solvers. The number of scalar products, matrix products, decomposition LU of the first derivative, and the resolution of the triangular linear systems are of great importance in assessing the real efficiency of such schemes. Some extensive discussions on this matter can be found in the recent literature [15, 16].

To achieve this goal, we in what follows consider a different way. Let us count the number of matrix quotients, products, summations, and subtractions along with the cost of

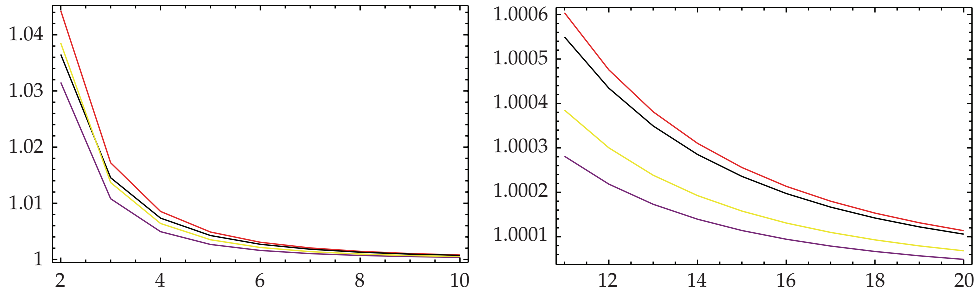


Figure 2: The plot of the flops-like efficiency indices for different methods ((a), for $n = 2, \ldots, 10$ and (b), for $n = 11, \ldots, 20$).

Table 2: Results of comparisons for different methods in Example 5.1.

Iterative methods	(1.2)	(1.5)	(1.6)	(2.2)
Number of iterations	13	7	7	6
Residual norm	3.03×10^{-108}	2.44×10^{-162}	0	6.54×10^{-137}
Number of functional evaluations	3120	3255	3255	2830
Time	1.59	1.76	1.79	1.51

solving two triangular systems, that is, based on flops (the real cost of solving systems of linear equations). In this case, we remark that the flops for obtaining the LU factorization are $2n^3/3$, and to solve two triangular system, the flops would be $2n^2$. Note that if the right-hand side is a matrix, the cost (flops) of the two triangular systems is $2n^3$, or roughly n^3 as considered in this paper. Table 1 also reveals the comparisons of flops and the flops-like efficiency index. Note that to the best of the authors' knowledge, such an index has not been given in any other work. Results of this are reported in Table 1 and Figure 2 as well. It is observed that the new scheme again competes all the recent or well-known iterations when comparing the computational efficiency indices.

5. Numerical Testing

We employ here the second-order method of Newton (1.2), the fourth-order scheme of Sharma et al. (1.5), the fourth-order scheme of Soleymani (1.6), and the proposed sixth-order iterative method (2.2) to compare the numerical results obtained from these methods in solving test nonlinear systems. In this section, the residual norm along with the number of iterations in Mathematica 8 [17] is reported in Tables 2-3. The computer specifications are Microsoft Windows XP Intel(R), Pentium(R) 4 CPU, 3.20 GHz with 4 GB of RAM. In numerical comparisons, we have chosen the floating point arithmetic to be 256 using the command $SetAccuarcy[expr, 256]$ in the written codes. The stopping criterion (for the first two examples) is the residual norm of the multivariate function to be less than $1 \cdot E - 150$, that is, $\|F(\mathbf{x}^{(k)})\| \leq 10^{-150}$. Note that for some iterative methods if their residual norms at the specified iteration exceed the bound of 10^{-200}, we will consider that such an approximation is the exact solution and denoted the residual norm by 0 in some cells of the tables. We consider the test problems as follows.

Example 5.1. As the first test, we take into account the following hard system of 15 nonlinear equations with 15 unknowns having a complex solution to reveal the capability of the new scheme in finding (n-dimensional) complex zeros:

$$\begin{gathered}
5x_1 - 2x_2 + 8x_3^{x_4} - 5x_6^3 + 2x_7^{x_{10}} - x_9^2 + x_{11}^{x_2} - 10x_{13} + x_{14}^2 - 2x_{15}^2 = 0, \\
5^{x_1} + 2 + 3x_2^3 + 7x_3^4 - 2x_6^3 + 2x_9^{x_{10}} + 2x_{12}^{x_{11}} - 10x_{13}x_{14} - x_{15}^2 = 0, \\
x_1^2 + 2x_2 + 2x_3^{x_4} - 5x_6^3 - x_5x_6x_7x_8x_9x_{10} + x_{11} - x_{13} + x_{14} + x_{15}^3 = 0, \\
2x_1^{x_5} - x_6^{x_2} - 3x_4 + 2x_5 - 7x_8^2 - 2x_{10}^{x_{12}} + x_{13} + x_{15}^2 - 10x_{14}^2 = 0, \\
2x_{10}^{x_1} + 2^{x_2} + 3{x_3}^2 - 5{x_5}^3 - 2x_6 + 2x_8^{x_9} + x_{13}^2 + 10x_{14} - x_{15}^2 = 0, \\
10x_1^2 + x_2 + x_3 - 5x_6^3 - 4^{x_9} - 2x_8 - 4x_{10} + 2x_{12}^2 - x_{13}^{x_{14}} + 2x_{15}^2 = 0, \\
-2x_1^2 - 2x_4 + 10x_3^2 - 100x_5^4x_6x_9x_{10} + 3x_{12} - 2x_{13}^{x_{14}} + 10x_{15}^3 = 0, \\
2x_4^{x_2} + 2x_2 - 5x_1^3 + 2x_7 - 2x_8^{x_{10}} - 5x_{11}^2 + x_{12} - 2x_{14} + 10x_{15}^3 = 0, \\
x_2 - x_2 + x_1^{10} + 3x_1^3 - 15x_5^2 + x_4x_7 + x_8 + x_9 - 2x_{10} - x_{11}x_{12}^3 + x_{15} = 0, \\
10x_1 + x_3^2 + x_4^2 - 5x_5^2 + 10x_6^{x_8} + 2x_9 - x_7 - 2x_2^2 + x_{12} - 2x_{13}^{x_{14}} + 2x_{15}^3 = 0, \\
x_1x_2 + 10x_3^{x_4} - 5x_5 - 100x_6 - 2x_{10}^{x_8} - 10x_9 + x_{11}^{x_{12}} - 2x_{13} - 4x_{14}^2 + x_{15} = 0, \\
2x_4 + x_2^3 + 7x_3^4 - 20x_6^3 + 2x_9^{x_{10}} - 101x_{12}^{x_{11}} - 3x_{13} - 10x_{14} - x_{15}^2 = 0, \\
2x_1^2 - x_7 + 2 + x_4^3 + 7x_3^4 - 20x_6^3 + x_9^{x_{10}} + 2x_{12}^{x_{11}} - x_{13}x_{14} - x_{15}^2 = 0, \\
30x_1 - x_5 + 2 + x_3^3 + 7x_2^2 - 2x_6 + 2x_9^{x_{10}} + 10x_{12} - 20x_{13}x_{14} - 3x_{15}^2 = 0, \\
-x_1 + 2 + 7x_3^{x_4} - x_2^3 + x_9^{x_{10}} + 3x_{12}^{x_{10}} - 2x_{13}x_{14} - 2x_{15}^2 = 0,
\end{gathered} \tag{5.1}$$

where its complex solution up to 10 decimal places is as follows: $\mathbf{x}^* \approx (1.981336305 + 0.983441720i,\ 3.409384824 - 0.764284796i,\ 1.813731796 - 0.637692532i,\ 3.491727320 + 0.872252620i,\ 6.550770690 - 0.907243589i,\ 1.336919493 - 1.019442606i,\ 79.096785866 + 48.25743733i,\ 3.082975105 + 0.835126126i,\ 5.320491462 - 1.520266411i,\ 0.000020217 + 0.000010961i,\ 0.013114096 + 0.08934367440i,\ 13.79912588 - 26.64001284i,\ 1.144969346 + 2.175231550i,\ -2.699288335 - 6.949621654i,\ -3.302042978 - 0.005478294i)^T$.

In this test problem, the approximate solution up to 2 decimal places can be constructed based on a line search method for obtaining robust initial value as follows: $\mathbf{x}^* \approx (1.98+0.98I, 3.40-0.76I, 1.81-0.63I, 3.49+0.87I, 6.55-0.90I, 1.33-1.01I, 79.10+48.26I, 3.08+0.83I, 5.32-1.52I, 0.02+0.01I, 0.00+0.1I, 13.80-26.64I, 1.14+2.17I, -2.69-6.94I, -3.30+0.00I)^T$. The results for this test problem are given in Table 2.

Results in Table 2 show that the new scheme can be considered for complex solutions of hard nonlinear systems as well. In this test, we have used the LU decomposition, when dealing with linear systems.

Table 3: Results of comparisons for different methods in Example 5.2 when $n = 99$.

Iterative methods	(1.2)	(1.5)	(1.6)	(2.2)
Number of iterations	9	4	4	4
Residual norm	0	1.57×10^{-101}	7.63×10^{-112}	0
Number of functional evaluations	89100	78804	78804	79200
Time	1.12	1.58	2.64	1.20

Example 5.2. In order to tackle with large-scale nonlinear systems, we have included this example in this work:

$$\begin{gathered} x_i x_{i+1} - 1 = 0, \qquad i = 1, 2, \ldots, n-1, \\ x_n x_1 - 1 = 0, \end{gathered} \tag{5.2}$$

where its solution is the vector $\mathbf{x}^* = (1, \ldots, 1)^T$ for odd n, and its first Frechet derivative has the following sparse pattern:

$$J(\mathbf{x}) = \begin{pmatrix} x_2 & x_1 & 0 & \cdots & 0 \\ 0 & x_3 & x_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & x_{n-2} & x_{n-1} \\ x_n & 0 & 0 & 0 & x_1 \end{pmatrix}. \tag{5.3}$$

We report the numerical results for solving Example 5.2 in Table 3 based on the initial value `x0 = Table [2., {i, 1, 99}]`. The case for 99×99 is considered. In Table 3, the new scheme is good in terms of the obtained residual norm in a few number of iterations.

Throughout the paper, the computational time has been reported by the command `AbsoluteTiming[]`. The mean of 5 implementations is listed as the time. In the following example, we consider the application of such nonlinear solvers when challenging hard nonlinear PDEs or nonlinear integral equations, some of such applications have also been given in [18–20].

Example 5.3. Consider the following nonlinear system of Volterra integral equations:

$$\begin{aligned} y_1(x) &= f_1(x) + \int_0^x \left[y_1(s) y_2(s) \right] ds, \\ y_2(x) &= f_2(x) + \int_0^x \left[y_1^2(s) y_2^2(s) \right] ds, \end{aligned} \tag{5.4}$$

where $f_1(x) = \cos(x) - (1/2)\sin^2(x)$, $f_2(x) = \sin(x) - x$. In order to solve this nonlinear problem, there are many methods. Recently, Samadi and Tohidi [21] showed that the spectral method is a much more reliable treatment in solving such problems. In fact, they discussed that traditional solvers will require very fine grid point which may cause obtaining large-scale (ill-conditioned) nonlinear systems of equations after implementation to find the final

Table 4: Results of comparisons for different methods in Example 5.3 when $n = 8$.

Iterative methods	(1.2)	(1.5)	(1.6)	(2.2)
Number of iterations	8	5	5	4
Residual norm	2.47×10^{-7}	3.40×10^{-16}	1.26×10^{-16}	4.47×10^{-10}
Number of functional evaluations	576	680	680	576
Time	0.03	0.03	0.03	0.015

solution. As a remedy, they presented a stable Legendre collocation method for solving systems of Volterra integral equations (SVIEs) of the second kind. Hence, now by applying the same procedure in [21] and by considering $N = 3$, and 5 digits at the first part of the process, the following nonlinear system of eight equations (5.5) will be attained where its solutions vector is $\{0.9975\ldots, 0.946\ldots, 0.783\ldots, 0.597\ldots, 0.0693\ldots, 0.324\ldots, 0.62\ldots, 0.80\ldots\}$.

Numerical results for this example are reported in Table 4. We again used 256-digit floating point in our calculations but the stopping criterion is the residual norm to be accurate for seven decimal places at least, that is to say, $\|F(\mathbf{x}^{(k)})\| \leq 10^{-7}$. The starting values are chosen as $(-10, \ldots, -10)^T$. In this case, we only report the time of solving the system (5.4) by the iterative methods and do not include the computational time of spectral method to first discretize (5.4).

The aim of the above example was twofold, first to show the clear reduction in number of steps in solving practical problems and also to reveal the fact that the proposed iterative method could be applied on *inexact* systems as well. By the word *inexact*, we mean the coefficients of the nonlinear systems are not integer and they are some number with floating arithmetic in essence, which itself may cause some round-off errors. We observe from Tables 2–4 that not only the order of convergence but also the number of new functional evaluations and operations is important in order to obtain new efficient iterative methods to solve nonlinear systems of equations.

$$
\begin{aligned}
&-0.99518 + x_1 - 0.11056x_1x_5 + 0.035818x_2x_5 - 0.017053x_3x_5 + 0.0048022x_4x_5 + 0.035818x_1x_6 \\
&-0.014033x_2x_6 + 0.0067323x_3x_6 - 0.0018999x_4x_6 - 0.017053x_1x_7 + 0.0067323x_2x_7 \\
&-0.0032313x_3x_7 + 0.00091202x_4x_7 + 0.0048022x_1x_8 - 0.0018999x_2x_8 + 0.00091202x_3x_8 \\
&-0.00025742x_4x_8, \\
&-0.89354 + x_2 - 0.17166x_1x_5 - 0.015764x_2x_5 - 0.0015117x_3x_5 + 0.0007561x_4x_5 - 0.015764x_1x_6 \\
&-0.1751x_2x_6 + 0.037366x_3x_6 - 0.0095897x_4x_6 - 0.0015117x_1x_7 + 0.037366x_2x_7 - 0.010818x_3x_7 \\
&+0.0028453x_4x_7 + 0.0007561x_1x_8 - 0.0095897x_2x_8 + 0.0028453x_3x_8 - 0.00075075x_4x_8, \\
&-0.59102 + x_3 - 0.17325x_1x_5 - 0.0028122x_2x_5 + 0.0095642x_3x_5 - 0.00075441x_4x_5 \\
&-0.0028122x_1x_6 - 0.31532x_2x_6 - 0.03736x_3x_6 + 0.0015064x_4x_6 + 0.0095642x_1x_7 \\
&-0.03736x_2x_7 - 0.15105x_3x_7 + 0.015772x_4x_7 - 0.00075441x_1x_8 + 0.0015064x_2x_8 \\
&+0.015772x_3x_8 - 0.0023316x_4x_8,
\end{aligned}
$$

$$
\begin{aligned}
&-0.27581 + x_4 - 0.17375x_1x_5 - 0.00089146x_2x_5 + 0.0018833x_3x_5 - 0.0048034x_4x_5\\
&-0.00089146x_1x_6 - 0.32288x_2x_6 - 0.0067382x_3x_6 + 0.017042x_4x_6 + 0.0018833x_1x_7\\
&-0.0067382x_2x_7 - 0.31209x_3x_7 - 0.035814x_4x_7 - 0.0048034x_1x_8 + 0.017042x_2x_8\\
&-0.035814x_3x_8 - 0.063427x_4x_8,\\
&0.00006 - 0.11056x_1^2 + 0.071636x_1x_2 - 0.014033x_2^2 - 0.034105x_1x_3 + 0.013465x_2x_3 - 0.0032313x_3^2\\
&+0.0096044x_1x_4 - 0.0037998x_2x_4 + 0.001824x_3x_4 - 0.00025742x_4^2 + x_5 - 0.11056x_5^2\\
&+0.071636x_5x_6 - 0.014033x_6^2 - 0.034105x_5x_7 + 0.013465x_6x_7 - 0.0032313x_7^2 + 0.0096044x_5x_8\\
&-0.0037998x_6x_8 + 0.001824x_7x_8 - 0.00025742x_8^2,\\
&0.00596 - 0.17166x_1^2 - 0.031527x_1x_2 - 0.1751x_2^2 - 0.0030234x_1x_3 + 0.074732x_2x_3 - 0.010818x_3^2\\
&+0.0015122x_1x_4 - 0.019179x_2x_4 + 0.0056905x_3x_4 - 0.00075075x_4^2 - 0.17166x_5^2 + x_6\\
&-0.031527x_5x_6 - 0.1751x_6^2 - 0.0030234x_5x_7 + 0.074732x_6x_7 - 0.010818x_7^2 + 0.0015122x_5x_8\\
&-0.019179x_6x_8 + 0.0056905x_7x_8 - 0.00075075x_8^2,\\
&0.04901 - 0.17325x_1^2 - 0.0056243x_1x_2 - 0.31532x_2^2 + 0.019128x_1x_3 - 0.074719x_2x_3 - 0.15105x_3^2\\
&-0.0015088x_1x_4 + 0.0030128x_2x_4 + 0.031544x_3x_4 - 0.0023316x_4^2 - 0.17325x_5^2 - 0.0056243x_5x_6\\
&-0.31532x_6^2 + x_7 + 0.019128x_5x_7 - 0.074719x_6x_7 - 0.15105x_7^2 - 0.0015088x_5x_8 + 0.0030128x_6x_8\\
&+0.031544x_7x_8 - 0.0023316x_8^2,\\
&0.12861 - 0.17375x_1^2 - 0.0017829x_1x_2 - 0.32288x_2^2 + 0.0037666x_1x_3 - 0.013476x_2x_3 - 0.31209x_3^2\\
&-0.0096067x_1x_4 + 0.034085x_2x_4 - 0.071628x_3x_4 - 0.063427x_4^2 - 0.17375x_5^2 - 0.0017829x_5x_6\\
&-0.32288x_6^2 + 0.0037666x_5x_7 - 0.013476x_6x_7 - 0.31209x_7^2 + x_8 - 0.0096067x_5x_8 + 0.034085x_6x_8\\
&-0.071628x_7x_8 - 0.063427x_8^2.
\end{aligned}
\tag{5.5}
$$

6. Conclusions

In this paper, an efficient iterative method for finding the real and complex solutions of nonlinear systems has been presented. We have supported the proposed iteration by a mathematical proof through the n-dimensional Taylor expansion. This let us analytically find the sixth order of convergence. Per computing step, the method is free from second-order Frechet derivative.

A complete discussion on the efficiency index of the new scheme was given. Nevertheless, the efficiency index is not the only aspect to take into account: the number of operations per iteration is also important; hence, we have given the comparisons of efficiencies based on the flops and functional evaluations. Some different numerical tests have been used to compare the consistency and stability of the proposed iteration in contrast to the existing methods. The numerical results obtained in Section 5 reverified the theoretical aspects of the paper. In fact, numerical tests have been performed, which not only illustrate the method practically but also serve to check the validity of theoretical results we had derived.

Future studies could focus on two aspects, one to extend the order of convergence alongside the computational efficiency and second to present some hybrid algorithm based on convergence guaranteed methods (at the beginning of the iterations) to ensure being in the trust region to apply the high-order methods.

Acknowledgments

The authors thank the reviewers for offering some helpful comments on an earlier version of this work. The work of the third author is financially supported through a Grant by the University of Venda, South Africa.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] H. Binous, “Solution of a system of nonlinear equations using the fixed point method,” 2006, http://library.wolfram.com/infocenter/MathSource/6611/.

[8] A. Margaris and K. Goulianas, “Finding all roots of 2 × 2 nonlinear algebraic systems using backpropagation neural networks,” *Neural Computing and Applications*, vol. 21, no. 5, pp. 891–904, 2012.

[9] E. Turan and A. Ecder, “Set reduction in nonlinear equations,” . In press, http://arxiv.org/abs/1203.3059v1.

[10] B.-C. Shin, M. T. Darvishi, and C.-H. Kim, “A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems,” *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3190–3198, 2010.

[11] D. K. R. Babajee, M. Z. Dauhoo, M. T. Darvishi, and A. Barati, “A note on the local convergence of iterative methods based on Adomian decomposition method and 3-node quadrature rule,” *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 452–458, 2008.

[12] J. R. Sharma, R. K. Guha, and R. Sharma, “An efficient fourth order weighted-Newton method for systems of nonlinear equations,” *Numerical Algorithms*. In press.

[13] F. Soleymani, “Regarding the accuracy of optimal eighth-order methods,” *Mathematical and Computer Modelling*, vol. 53, no. 5-6, pp. 1351–1357, 2011.

[14] 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.

[15] M. Grau-Sánchez, A. 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.

[16] J. A. Ezquerro, M. Grau-Sánchez, A. Grau, M. A. Hernández, M. Noguera, and N. Romero, “On iterative methods with accelerated convergence for solving systems of nonlinear equations,” *Journal of Optimization Theory and Applications*, vol. 151, no. 1, pp. 163–174, 2011.

[17] M. Trott, *The Mathematica GuideBook for Numerics*, Springer, New York, NY, USA, 2006.

[18] A. H. Bhrawy, E. Tohidi, and F. Soleymani, “A new Bernoulli matrix method for solving high-order linear and nonlinear Fredholm integro-differential equations with piecewise intervals,” *Applied Mathematics and Computation*, vol. 219, no. 2, pp. 482–497, 2012.

[19] M. T. Darvishi, “Some three-step iterative methods free from second order derivative for finding solutions of systems of nonlinear equations,” *International Journal of Pure and Applied Mathematics*, vol. 57, no. 4, pp. 557–573, 2009.

[20] M. Y. Waziri, W. J. Leong, M. A. Hassan, and M. Monsi, “A low memory solver for integral equations of Chandrasekhar type in the radiative transfer problems,” *Mathematical Problems in Engineering*, vol. 2011, Article ID 467017, 12 pages, 2011.

[21] O. R. N. Samadi and E. Tohidi, “The spectral method for solving systems of Volterra integral equations,” *Journal of Applied Mathematics and Computing*, vol. 40, no. 1-2, pp. 477–497, 2012.