Chapter

# A Shamanskii-Like Accelerated Scheme for Nonlinear Systems of Equations

*Ibrahim Mohammed Sulaiman, Mustafa Mamat and Umar Audu Omesa*

## Abstract

Newton-type methods with diagonal update to the Jacobian matrix are regarded as one most efficient and low memory scheme for system of nonlinear equations. One of the main advantages of these methods is solving nonlinear system of equations having singular Fréchet derivative at the root. In this chapter, we present a Jacobian approximation to the Shamanskii method, to obtain a convergent and accelerated scheme for systems of nonlinear equations. Precisely, we will focus on the efficiency of our proposed method and compare the performance with other existing methods. Numerical examples illustrate the efficiency and the theoretical analysis of the proposed methods.

**Keywords:** Newton method, Shamanskii method, diagonal updating scheme, nonlinear equations, Jacobian matrix

## 1. Introduction

A large aspect of scientific and management problems is often formulated by obtaining the values of $x$ of which the function evaluation of that variable is equal to zero [1]. The above description can be represented mathematically by the following system of nonlinear equations:

$$\begin{aligned} f_1(x_1, x_2, ..., x_n) &= 0 \\ f_2(x_1, x_2, ..., x_n) &= 0 \\ \vdots \quad \vdots \quad \vdots &= \quad \vdots \\ f_n(x_1, x_2, ..., x_n) &= 0 \end{aligned} \tag{1}$$

where $x_1, x_2, ..., x_n \in \mathbb{R}^n$ are vectors and $f_i$ is nonlinear functions for $i = 1, 2, ..., n$. The above system of equations (1) can be written as

$$F(x) = 0 \tag{2}$$

where $F : R^n \to R^n$ is continuously differentiable in an open neighborhood of the solution $x^*$. These systems are seen as natural description of observed phenomenon of numerous real-life problems whose solutions are seen as an important goal in

mathematical study. Recently, this area has been studied extensively [2, 3]. The most powerful techniques for handling nonlinear systems of equations are to linearize the equations and proceed to iterate on the linearized set of equations until an accurate solution is obtained [4]. This can be achieved by obtaining the derivative or gradient of the equations. Various scholars stress that the derivatives should be obtained analytically rather than using numerical approach. However, this is usually not always convenient and, in most cases, not even possible as equations may be generated simply by a computer algorithm [2]. For one variable problem, system of nonlinear equation defined in (2) represents a function $F : R \to R$ where $f$ is continuous in the interval $f \in [a, b]$.

**Definition 1:** Consider a system of equations $f_1, f_2, \ldots, f_n$; the solution of this system in one variable, two variables, and $n$ variable is referred to as a point $(a_1, a_2, \ldots, a_n) \in R^n$ such that $f_1(a_1, a_2, \ldots, a_n) = f_2(a_1, a_2, \ldots, a_n) = \ldots = f_n(a_1, a_2, \ldots, a_n) = 0$.

In general, the problem to be considered is that for some function $f(x)$, one wishes to evaluate the derivative at some points $x$, i.e.,

$$\text{Given } f(x), \text{ Evaluate; deriv} = \frac{df}{dx}$$

This often used to represent an instantaneous change of the function at some given points [5].

**Definition 2:** For a function $f(x)$ that is smooth, then there exists, at any point $x$, a vector of first-order partial derivative or gradient vector:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = g(x).$$

The Taylor's series expansion of the function $f(x)$ about point $x_0$ is an ideal starting point for this discussion [1].

**Definition 3:** Let $f$ be a differentiable function; the Taylor's $f(x)$ around a point $a$ is the infinite sum:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots$$

However, continuous differentiable vector valued function does not satisfy the mean value theorem (MVT), an essential tool in calculus [6]. Hence, academics suggested the use of the following theorem to replace the mean valued theorem.

**Theorem 1:** Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable in an open convex set $D \subset \mathbb{R}^n$. For any $x, x + s \in D$

$$F(x+s) - F(x) = \int_0^1 J(x+ts)s\,dt \equiv \int_x^{x+s} F'(z)\,dz$$

**Definition 4:** Suppose $F : R^n \rightarrow R^n$ is continuously differentiable at the point $x \in R^n$ and each component function $f_1, f_2, \ldots, f_m$ is also continuously differentiable at $x$; then the derivative of $F\,x$ is defined as

$$J_x(F) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{pmatrix}$$

Most of the algorithms employ for obtaining the solution of Eq. (1) centered on approximating the Jacobian matrix which often provides a linear map $T(x) : R^n \rightarrow R^n$ defined by Eq. (3)

$$T(x) = J_x F(x)\ \forall\, x \in R^n \tag{3}$$

Also, if $F$ is differentiable at point $x_*$, then the affine function $A(x) = f(x_*) + J_*(x - x_*)$ is a good approximation to $F(x)$ near $x = x_*$ in such a way that

$$\lim_{x \rightarrow x_*} \frac{\|F(x) - F(x_*) - J_*(x - x_*)\|}{\|x - x_*\|} \tag{4}$$

where $\|x - x_*\| = \sqrt{(x_1 - x_*)^2 + (x_2 - x_*)^2 + \ldots + (x_n - x_*)^2}$.

If all the given component functions $f_1, f_2, \ldots, f_m$ of $J_x(F)$ are continuous, then we say the function $F$ is differentiable.

The most famous method for solving nonlinear systems of equations $F(x) = 0$ is the Newton method which generates a sequence $\{x_k\}$ from any given initial point $x_0$ via the following:

$$x_{k+1} = x_k - F'(x_k)^{-1} F(x_k) \tag{5}$$

where $F'(x_k)$ is the Jacobian for $F(x_k)$. The above sequence Eq. (5) is said to converge quadratically to the solution $x^*$ if $x_0$ is sufficiently near the solution point and the Jacobian $F'(x_k)$ is nonsingular [7, 8]. This convergent rate makes the method outstanding among other numerical methods. However, Jacobian evaluation and solving the linear system for the step $s(x_n) = -F'(x_n)^{-1} F(x_n)$ are expensive and time-consuming [9]. This led to the study of different variants of Newton methods for systems of nonlinear equations. One of the simplest and low-cost variants of the Newton method that almost entirely evades derivate evaluation at every iteration is the chord method. This scheme computes the Jacobian matrix $F'(x_0)$ once throughout the iteration process for finite dimensional problem as presented in Eq. (6),

$$x_{k+1} = x_k - F'(x_0)^{-1} F(x_k) \tag{6}$$

The rate of convergence is linear and improves as the initial point gets better. Suppose $x_0$ is sufficiently chosen near solution point $x_*$ and $F(x^*)$ is nonsingular; then, for some $K_c > 0$, we have

$$\|x_{n+1} - x^*\| \leq K_c \|x_0 - x^*\| \|x_n - x^*\| \tag{7}$$

The convergence theorems and proof of Eq. (7) can be referred to [9, 10]. Motivated by the excellent convergence of Newton method and low cost of Jacobian evaluation of chord method, a method due originally to Shamanskii [11, 12] that lies between Newton method and chord method was proposed and has been analyzed in Kelly [9, 13–15]. Other variants of Newton methods with different Jacobian approximation schemes include [9, 14, 16–18]. However, most of these methods require the computation and storage of the full or approximate Jacobian, which become very difficult and time-consuming as the dimension of systems increases [10, 19].

It would be worthwhile to construct a derivative-free approach and analyze with existing techniques [20–22]. The aim of this work is to derive a diagonal matrix for the approximate Jacobian of Shamanskii method by means of variational techniques. The expectation would be to reduce computational cost, storage, and CPU time of evaluating any problem. The proposed method works efficiently by combining the good convergence rate of Shamanskii method and the derivate free approach employed, and the results are very encouraging. The next section presents the Shamanskii method for nonlinear systems of equations.

## 2. Shamanskii method

It is known that the Newton method defined in Eq. (2) converges quadratically to $x^*$ when the initial guess is sufficiently close to the root [7, 10, 19]. The major concern about this method is the evaluation and storage of the Jacobian matrix at every iteration [23]. A scheme that almost completely overcomes this is the chord method. This method factored the Jacobian matrix only once in the case of a finite dimensional problem, thereby reducing the evaluation cost of each iteration as in Eq. (3) and thereby degrading the convergence rate to linear [10].

Motivated by this, a method due originally to Shamanskii [11] was developed and analyzed by [7, 13, 14, 16, 24]. Starting with an initial approximation $x_0$, this method uses the multiple pseudo-Newton approach as described below:

$$x_{k+\frac{1}{2}} = x_k - F'(x_k)^{-1}F(x_k) \tag{8}$$

$$x_{k+1} = x_{k+\frac{1}{2}} - F'(x_k)^{-1}F\left(x_{k+\frac{1}{2}}\right) \tag{9}$$

after little simplification, we have

$$x_{k+1} = x_k - F'(x_k)^{-1}\left[F(x_k) + F\left(x_k - F'(x_k)^{-1}F(x_k)\right)\right] \tag{10}$$

This method converges superlinearly with $q$-order of at least $t+1$ when the initial approximation $x_0$ is sufficiently chosen near the solution point $x^*$ and $F'(x^*)$ is nonsingular. This implies that there exists $K_s > 0$, such that

$$\|x_{n+1} - x^*\| \leq K_s \|x_n - x^*\|^{t+1} \tag{11}$$

Combining Eq. (7) and the quadratic convergence of Newton method produces the convergence rate of the Shamanskii method as in Eq. (8). Thus, the balance is between the reduced evaluation cost of Fréchet derivative and Jacobian computation for Shamanskii method and Newton method rapid convergence. This low-cost derivative evaluation as well as the rapid convergence rate of several methods including the Shamanskii method has been studied and analyzed in [13, 15]. From the analysis, the researchers conclude that that Shamanskii method has shown

superior performance compared to Newton method in terms of efficiency whenever work is measured in terms of function evaluations [9]. Also, if the value of $t$ is sufficiently chosen, then, as the dimension increases, the performance of the Shamanskii method improves and thus reduces the limit of complexity of factoring the approximate Jacobian for two pseudo-Newton iterations [14]. Please refer to [15] for the proof of the convergence theorem described below.

**Theorem 2** [15]: Let $F : D \subset R^n \rightarrow R^n$ conform hypotheses H1(2), H2, and H3. Then the solution point $x^*$ is a point of attraction of the Shamanskii iteration, i.e., Eq. (10), and this method possesses at least cubic order of convergence.

## 3. Diagonal updating scheme for solving nonlinear systems

Evaluation or inversion of the Jacobian matrix at every iteration or after few iterations does not seem relevant even though the computational cost has generally been reduced as in Shamanskii method [14, 25–28]. As a matter of fact, it can be easily shown that by adding a diagonal updating scheme to a method, we would have a new low memory iterative approach which would approximate the Jacobian $F'(x_k)$ into nonsingular diagonal matrix that can be updated in every iteration [29–31]. Indeed, using the Shamanskii procedure, the proposed method avoids the main complexity of the Newton-type methods by reusing the evaluated Jacobian during the iteration process. This is the basic idea of the Shamanskii-like method which is described as follows.

Given an initial approximation $x_0$, we compute Eq. (2) to obtain the Jacobian $F'(x_k)$ and present a diagonal approximation to the Jacobian say $D_k$ as follows:

$$F'(x_{k+1}) \approx D_{k+1} \tag{12}$$

Suppose $s_k = x_{k+1} - x_k$ and $y_k = F(x_{k+1}) - F(x_k)$; by mean value theorem (MVT), we have

$$D_{k+1}s_k \approx y_k \tag{13}$$

Substituting Eq. (12) in Eq. (13), we have

$$F'(x_k)s_k \approx y_k \tag{14}$$

Since $D_{k+1}$ is the update of diagonal matrix $D_k$, let us assume $D_{k+1}$ satisfy the weak secant equation:

$$s_k^T D_{k+1} s_k = s_k^T y_k \tag{15}$$

which would be used to minimize the deviation between $D_{k+1}$ and $D_k$ under some norms. The updated formula for $D_k$ follows after the theorem below:

**Theorem 3:** Suppose $D_{k+1}$ is the update of the diagonal matrix $D_k$ and $\Delta_k = D_{k+1} - D_k, s_k \neq 0$. Consider the problem

$$\min \frac{1}{2}\|\Delta_k\|_F^2 \tag{16}$$

such that Eq. (15) holds and $\|.\|_F$ denotes the Frobenius norm. From Eq. (16), we have the following solution also regarded as the optimal solution:

$$\Delta_k = \frac{s_k^T y_k - s_k^T D_{k+1} s_k}{tr\left(\Omega_k^2\right)} \Omega_k \tag{17}$$

where $\Omega_k = \text{diag}\left(\left(s_k^{(1)}\right)^2, , \left(s_k^{(2)}\right)^2, ..., , \left(s_k^{(n)}\right)^2\right)$, $\sum_{i=1}^{n}\left(s_k^{(i)}\right)^4 = tr\left(\Omega_k^2\right)$, and $Tr$ is the trace operation.

**Proof:** It is known that the objective function and the constraint of Eq. (16) are convex; thus, we intend to use its Lagrangian function to obtain the unique solution as follows:

$$\mathrm{L}(\Delta_k, \mu) = \frac{1}{2}\|\Delta_k\|_F^2 + \mu\left(s_k^T\Delta_k s_k - s_k^T y_k - s_k^T D_{k+1} s_k\right) \tag{18}$$

where $\mu$ is the corresponding Lagrangian multiplier. Simplifying Eq. (18), we have

$$\mu = \frac{s_k^T y_k - s_k^T D_{k+1} s_k}{\sum_{i=1}^{n}\left(s_k^{(i)}\right)^4} \tag{19}$$

and

$$\Delta_k^{(i)} = \frac{s_k^T y_k - s_k^T D_{k+1} s_k}{\sum_{i=1}^{n}\left(s_k^{(i)}\right)^4}\left(s_k^{(i)}\right)^2 \quad \forall\, i = 1, 2, .., n \tag{20}$$

Also, for diagonal matrix $D_k$, the element of the diagonal component is given as $D_k^{(i)}$, and the $i^{th}$ component of the vector $s_k$ is $s_k^{(i)}$. Then $\Omega_k = \text{diag}\left(\left(s_k^{(1)}\right)^2, , \left(s_k^{(2)}\right)^2, ..., , \left(s_k^{(n)}\right)^2\right)$, and $\sum_{i=1}^{n}\left(s_k^{(i)}\right)^4 = tr\left(\Omega_k^2\right)$. To complete the proof, we rewrite Eq. (20) as follows:

$$\Delta_k = \frac{\left(s_k^T y_k - s_k^T D_{k+1} s_k\right)}{tr\left(\Omega_k^2\right)}\Omega_k. \tag{21}$$

This completes the proof. ∎

Now, from the above description of the theorem, we deduce that the best possible diagonal update $\mathrm{D}_{k+1}$ is as follows:

$$\mathrm{D}_{k+1} = \mathrm{D}_k + \frac{\left(s_k^T y_k - s_k^T D_{k+1} s_k\right)}{tr\left(\Omega_k^2\right)}\Omega_k \tag{22}$$

However, for possibly small $\|s_k\|$ and $tr\Omega_k$, we need to define a condition that would be applied for such cases. To this end, we require that $\|s_k\| \geq s_1$ for some chosen small $s_1 > 0$. Otherwise, we set the updated diagonal $\mathrm{D}_{k+1} = \mathrm{D}_k$ where $\mathrm{D}_{k+1}$ is defined as

$$\mathrm{D}_{k+1} = \begin{cases} \mathrm{D}_k + \dfrac{\left(s_k^T y_k - s_k^T D_{k+1} s_k\right)}{tr\left(\Omega_k^2\right)}\Omega_k;\ \|s_k\| \geq \epsilon_1 \\ D_k; \textit{Otherwise} \end{cases} \tag{23}$$

Thus, the proposed accelerated method is described as follows:

$$x_{k+1} = x_k - D_k^{-1}\left[F(x_k) + F\left(x_k - D_k^{-1}F(x_k)\right)\right] \tag{24}$$

The performance of this proposed method would be tested on well-known benchmark problems employed by researchers on existing methods. This would be

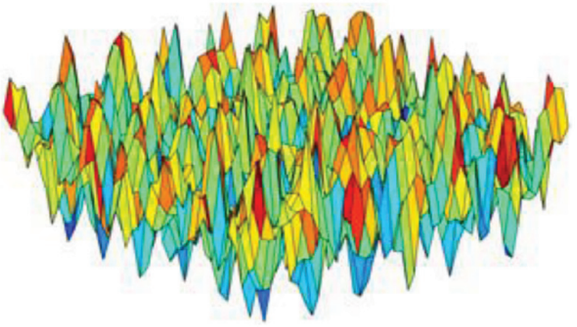

**Figure 1.**
*Functions with a huge number of significant local optima.*

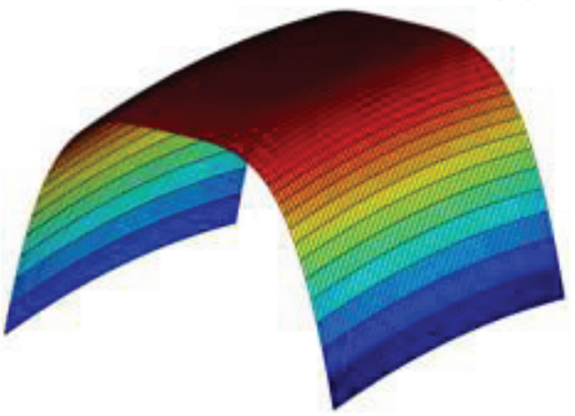

**Figure 2.**
*Functions with significant null space.*

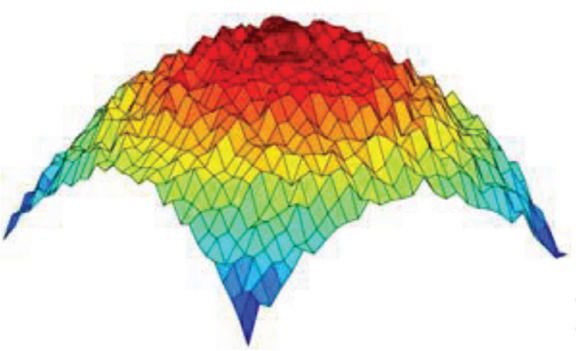

**Figure 3.**
*Essentially unimodal function.*

carried out using a designed computer code for its algorithm. The problems could be artificial or real-life problems. The artificial problems check the performance of any algorithm in situation such as point of singularity, function with many solutions, and null space effect as presented in **Figures 1–3** [7, 32].

While the real-life problems emerge from fields such as chemistry, engineering, management, etc., the real-life problems often contain large data or complex algebraic expression which makes it difficult to solve.

## 4. Numerical results

This section demonstrates the proposed method and illustrates its advantages on some benchmark problems with dimensions ranging from 25 to 1,000 variables.

These include problems with restrictions such as singular Jacobian or problems with only one point of singularity. To evaluate the performance of the proposed diagonal updating Shamanskii method (DUSM), we employ some tools by Dolan and Moré [33] and compare the performance with two classical Newton-type methods based on the number of iterations and CPU time in seconds. The methods include:

1. The Newton method (NM)

2. The Shamanskii method (SM)

These tools are used to represent the efficiency, robustness, and numerical comparisons of different algorithms. Suppose there exist $n_s$ solvers and $n_p$ problems; for each problem $p$ and solver $s$, they define:

$t_{p,s}$ = computing time needed to solve a problem by solver (the number of iteration or CPU time)

Requiring a baseline for comparisons, they compared the performance on problem $p$ by solver $s$ with the best performance by any solver for this problem using the performance ratio:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$

We suppose that parameter $r_m \geq r_{p,s}$ for all $p, s$ is chosen and $r_{p,s} = r_M$ if and only if solver $s$ does not solve problem $p$. The performance of solvers $s$ on any given problem might be of interest, but because we would prefer obtaining the overall assessment of the performance of the solver, then it was defined as

$$p_s(t) = \frac{1}{n_p} size\{p \in P : r_{p,s} \leq t\}.$$

Thus, $p_s(t)$ was the probability for solver $s \in S$ that a performance ratio $r_{p,s}$ was within a factor $t \in R$ of the best possible ratio. Then, function $p_s$ was the cumulative distribution function for the performance ratio. The performance profile $p_s : R \to [0,1]$ for a solver was nondecreasing, piecewise, and continuous from right. The value of $p_s(1)$ is the probability that the solver will win over the rest of the solvers. In general, a solver with high value of $p(\tau)$ or at the top right of the figure is preferable or represents the best solver.

All problems considered in this research are solved using MATLAB (R 2015a) subroutine programming [37]. This was run on an Intel® Core™ i5-2410M CPU @ 2.30 GHz processor, 4GB for RAM memory and Windows 7 Professional operating system. The termination condition is set as

$$\|s_k\| + \|F(x_k)\| \leq 10^{-6}$$

and the program has been designed to terminate whenever:

- The number of iterations exceeds 500, and no point of $x_k$ satisfies the termination condition.

- The CPU time in seconds reaches 500.

- Insufficient memory to initiate the run.

At the point of failure due to any of the above conditions as in the tabulated results, it is assumed the number of iteration and CPU time is zero and thus that point has been denoted by "$*$." The following are the details of the standard test problems, the initial points used, and the exact solutions for systems of nonlinear equations.

**Problem 1** [31]: System of $n$ nonlinear equations

$$F_i(x) = \left(1 - x_i^2\right) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2$$
$$i = 1, 2, 3, ..., n, \qquad x_0 = (0.3, 0, 3, ..., 0.3)$$

**Problem 2** [34]: Systems of $n$ nonlinear equations

$$F_i(x) = x_i^2 - \cos(x_i - 1)$$
$$i = 1, 2, 3, ..., n, \qquad x_0 = (0.2, 0.2, ..., 0.2)$$

**Problem 3** [31]: Structured exponential function

$$F_i(x) = e^{x_i} - 1$$
$$F_n(x) = x_n - 0.1x_n^2$$
$$i = 1, 2, 3, ..., n, \qquad x_0 = (0.05, 0.05, ..., 0.05)$$

| Problem | Dim | NM | | DUSM | | SM | |
|---|---|---|---|---|---|---|---|
| | | NI | CPU | NI | CPU | NI | CPU |
| 1 | 25 | 13 | 0.016102 | 8 | 0.034777 | 13 | 0.015999 |
| 2 | 25 | 6 | 0.009522 | 7 | 0.028231 | 7 | 0.010412 |
| 3 | 25 | * | * | 4 | 0.023766 | * | * |
| 4 | 25 | 16 | 0.019679 | 17 | 0.077072 | 22 | 0.022889 |
| 5 | 25 | 4 | 0.006605 | 16 | 0.061750 | 4 | 0.005761 |
| 1 | 50 | 13 | 0.032998 | 8 | 0.090310 | 13 | 0.032271 |
| 2 | 50 | 10 | 0.022134 | 7 | 0.089785 | 7 | 0.017036 |
| 3 | 50 | 4 | 0.010350 | 4 | 0.052899 | 4 | 0.010238 |
| 4 | 50 | 30 | 0.054640 | 17 | 0.228077 | 23 | 0.041569 |
| 5 | 50 | 4 | 0.012361 | 16 | 0.201262 | 4 | 0.010735 |
| 1 | 100 | 13 | 0.073565 | 8 | 0.339333 | 13 | 0.066363 |
| 2 | 100 | 10 | 0.054075 | 7 | 0.292001 | 7 | 0.044512 |
| 3 | 100 | * | * | 4 | 0.175300 | * | * |
| 4 | 100 | 15 | 0.075073 | 18 | 0.770165 | 25 | 0.118170 |
| 5 | 100 | 4 | 0.029221 | 17 | 0.755556 | 4 | 0.023154 |
| 1 | 1000 | 13 | 1.868606 | 8 | 27.171776 | 13 | 2.042222 |
| 2 | 1000 | 10 | 1.444533 | 7 | 24.295632 | 7 | 1.045329 |
| 3 | 1000 | * | * | 4 | 27.1250 | * | * |
| 4 | 1000 | 52 | 6.757533 | 19 | 63.981376 | 39 | 5.138997 |
| 5 | 1000 | 4 | 0.610145 | 18 | 62.364143 | 4 | 0.612590 |

**Table 1.**
*Numerical comparison of NM, DUSM, and SM.*

**Problem 4** [35]: Extended trigonometric of Byeong-Chun

$$F_i(x) = \cos\left(x_i^2 - 1\right) - 1$$
$$i = 1, 2, 3, ..., n, \qquad x_0 = (0.06, 0.06, ..., 0.06)$$

**Problem 5** [36]: Extended spare system of Byeong

$$F_i(x) = x_i^2 - x_i - 2$$
$$i = 1, 2, 3, ..., n, \qquad x_0 = (1.1, 11.1, ..., 1.1)$$

**Table 1** shows the number of iterations (NI) and CPU time for Newton method (NM), Shamanskii method (SM), and the proposed diagonal updating method (DUSM), respectively. The performance of these methods was analyzed via storage locations and execution time. It can be observed that the proposed DUSM was able to solve the test problems perfectly, while NM and SM fail at some points due to the matrix being singular to working precision. This shows that the diagonal scheme employed has provided an option in the case of singularity, thereby reducing the computational cost of the classical Newton-type methods.

## 5. Conclusion

This chapter proposes a diagonal updating formula for systems of nonlinear equations which attributes to reduction in Jacobian evaluation cost. By computational experiments, we reach the conclusion that the proposed scheme is reliable and efficient and reduces Jacobian computational cost during the iteration process. Meanwhile, the proposed scheme is superior compared to the result of the classical and existing numerical methods for solving systems of equations.

## Author details

Ibrahim Mohammed Sulaiman*, Mustafa Mamat and Umar Audu Omesa
Universiti Sultan Zainal Abidin, Kuala Terengganu, Malaysia

*Address all correspondence to: sulaimanib@unisza.edu.my

## References

[1] Rainer K. Numerical Analysis. New York: Springer Science+Business Media, LLC; 1998

[2] Sulaiman IM. New iterative methods for solving fuzzy and dual fuzzy nonlinear equations [PhD thesis]. Malaysia: Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin; 2018

[3] Wenyu S, Ya-Xiang Y. Optimization Theory and Methods, Springer Optimization and Its Applications. Boston, MA: Springer; 2006

[4] John RH. Numerical Methods for Nonlinear Engineering Models. Netherlands: Springer; 2009

[5] Burden RL, Faires JD. Numerical Analysis. 8th ed. USA: Thomson; 2005

[6] Wright SJ, Nocedal J. Numerical Optimization. 2nd ed. Berlin, Germany: Springer; 1999

[7] Dennis JE Jr, Schnabel RB. Numerical Method for Unconstrained Optimization and Nonlinear Equations. Houston, Texas: SIAM; 1996

[8] Griva I, Nash SG, Sofer A. Linear and Nonlinear Optimization. 2nd ed. Philadelphia: SIAM; 2009

[9] Kelley CT. A Shamanskii-like acceleration scheme for nonlinear equations at singular roots. Mathematics of Computation. 1986;**47**:609-623

[10] Kelley CT. Iterative Methods for Linear and Nonlinear Equations. Philadelphia, PA: SIAM; 1995

[11] Shamanskii VE. A modification of Newton's method. Ukrains'kyi Matematychnyi Zhurnal. 1967;**19**:133-138

[12] Shamanskii VE. On a realization of Newton's method on electronic computers. Ukrains'kyi Matematychnyi Zhurnal. 1966;**18**(6):135-140

[13] Traub JF. Iterative Methods for the Solution of Equations. Englewood Cliffs: Prentice-Hall; 1964

[14] Kchouk B, Dussault J. The Chebyshev–Shamanskii method for solving systems of nonlinear equations. Journal of Optimization Theory and Applications. 2013;**157**:148-167

[15] Ortega JM, Rheinboldt WC. Iterative Solution of Nonlinear Equations in Several Variables. New York: Academic Press; 1970

[16] Brent RP. Some efficient algorithms for solving systems of nonlinear equations. Journal of Nucleic Acids. 1973;**10**(2):327-344

[17] Waziri MY, Leong WJ, Mamat M, Moyi AU. Two-step derivative-free diagonally Newton's method for large-scale nonlinear equations. World Applied Sciences Journal. 2013;**21**:86-94. DOI: 10.5829/idosi.wasj.2013.21.am.2045

[18] Broyden CG. A class of methods for solving nonlinear simultaneous equations. Mathematics of Computation. 1965;**19**(92):577-593

[19] Chong EKP, Zak SH. An Introduction to Optimization, Wiley Series in Discrete Mathematics and Optimization. New York: John Wiley and Sons; 2013

[20] Jose LH, Eulalia M, Juan RM. Modified Newton's method for systems of nonlinear equations with singular Jacobian. Journal of Computational and Applied Mathematics. 2009;**224**:77-83

[21] Leong WJ, Hassan MA, Waziri MY. A matrix-free quasi-Newton method for solving large-scale nonlinear systems.

Computational and Applied Mathematics. 2011;**625**:2354-2363

[22] Natasa K, Zorana L. Newton-like methods with modification of the right-hand side vector. Mathematics of Computation. 2002;**71**:237-250

[23] Waziri MY, Leong WJ, Hassan MA, Monsi M. A new Newton's method with diagonal Jacobian approximation for systems of nonlinear equations. Journal of Mathematics and Statistics. 2010;**6** (3):246-252

[24] Lampariello F, Sciandrone M. Global convergence technique for the Newton method with periodic Hessian evaluation. Journal of Optimization Theory and Applications. 2001;**111**(2): 341-358

[25] Sulaiman IM, Mamat M, Nurnadiah Z, Puspa LG. Solving dual fuzzy nonlinear equations via Shamanskii method. International Journal of Engineering & Technology. 2018;**7** (3.28):89-91

[26] Ypma TJ. Historical development of the Newton-Raphson method. SIAM Review. 1995;**37**(4):531-551

[27] Hao L, Qin N. Incomplete Jacobian Newton method for nonlinear equation. Computers and Mathematics with Applications. 2008;**56**(1):218-227

[28] Chency E, Kincaid D. Numerical Mathematics and Computing. Asia: Nelson Education, Cengage Learning; 2012

[29] Sulaiman IM, Mamat M, Afendee MM, Waziri MY. Diagonal updating Shamanskii-like method for solving singular fuzzy nonlinear equation. Far East Journal of Mathematical Sciences. 2017;**103**(10):1619-1629

[30] Waziri MY, Leong WJ, Hassan MA, Monsi M. Jacobian-free Newton's method for systems of nonlinear equations. Journal of Numerical Mathematics and Stochastics. 2010;**2**(1): 54-63

[31] Waziri MY, Abdulmajid Z. An improved diagonal Jacobian approximation via a quasi-Cauchy condition for solving large-scale systems of nonlinear equations. Journal of Applied Mathematics. 2013;**3**:1-6

[32] Andrei N. An unconstrained optimization test functions collection. Advanced Modeling and Optimization. 2008;**10**:147-161

[33] Dolan E, Moré JJ. Benchmarking optimization software with performance profiles. Mathematical Programming. 2002;**91**(2):201-213

[34] Hafiz MA, Muhammad SMB. An efficient two-step iterative method for solving system of nonlinear equation. Journal of Mathematics Research. 2012; **4**(4):28-34

[35] Shin BC, Darvishi M, Kim CH. A comparison of the Newton-Krylov method with high order newton-like methods to solve nonlinear systems. Applied Mathematics and Computation. 2010;**217**(7):3190-3198

[36] Mamat M, Muhammad K, Waziri MY. Trapezoidal Broyden's method for systems of nonlinear equations. Applied Mathematical Sciences. 2014;**8**(6):54-63

[37] Sulaiman I, Mamat M, Waziri MY, Umar AO, et al. Journal of Advanced Research in Modelling and Simulations. 2018;**1**(1):13-18