Advances in the Theory of Nonlinear Analysis and its Applications **5** (2021) No. 4, 568–579.
https://doi.org/10.31197/atnaa.890281
Available online at www.atnaa.org
Research Article

Advances in the Theory of Nonlinear Analysis and its Applications
ISSN: 2587-2648
Peer-Reviewed Scientific Journal

# On efficient matrix-free method via quasi-Newton approach for solving system of nonlinear equations

Muhammad Abdullahi[a], Abubakar Sani Halilu[a], Aliyu Muhammed Awwal[b], Nuttapol Pakkaranang[c]

[a]*Department of Mathematics and Computer Sciences, Sule Lamido University, Kafin Hausa, Nigeria.*
[b]*Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria.*
[c]*Department of Mathematics, Faculty of Science and Technology, Phetchabun Rajabhat University, Phetchabun 67000, Thailand.*

**Abstract**

In this paper, a matrix-free method for solving large-scale system of nonlinear equations is presented. The method is derived via quasi-Newton approach, where the approximation to the Broyden's update is done by constructing diagonal matrix using acceleration parameter. A fascinating feature of the method is that it is a matrix-free, so is suitable for solving large-scale problems. Furthermore, the convergence analysis of the new method is discussed based on some standard condition. Preliminary numerical results on some test problems show that the method is promising.

*Keywords:* Matrix-free, Descent direction, Global convergence, Acceleration parameter.
*2010 MSC:* 65H11, 65K05, 65H12, 65H18.

## 1. Introduction

Many of problems in sciences, engineering and economics can be expressed as optimization problems or nonlinear system of equations, which are usually solved using iterative methods. This paper focuses on the following system

$$F(x) = 0, \tag{1}$$

where $x \in \mathbb{R}^n$ and the nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous. Throughout this paper, the symbol $\mathbb{R}^n$ denotes the $n-$dimensional real space equipped with the Euclidean norm $\|\cdot\|$, $F_k = F(x_k)$ where $x_k \in \mathbb{R}^n$ is the point at certain iteration $k = 1, 2, \ldots$.

*Email addresses:* muhammad.abdullahi@slu.edu.ng (Muhammad Abdullahi), abubakarsani.halilu@slu.edu.ng (Abubakar Sani Halilu), aliyumagsu@gmail.com (Aliyu Muhammed Awwal), nuttapol.pak@pcru.ac.th (Nuttapol Pakkaranang)

*Received :March 3, 2021; Accepted: July 8, 2021; Online: July 10, 2021*

Moreover, the system (1) can be obtained from general unconstrained optimization problems [9]. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a merit function defined by

$$f(x) = \frac{1}{2}\|F(x)\|^2, \quad x \in \mathbb{R}^n. \tag{2}$$

Then the nonlinear equations problem (1) is equivalent to the following unconstrained optimization problem

$$\min f(x), \quad x \in \mathbb{R}^n.$$

The study of such mappings is applied in a variety of scientific areas, including economic and chemical equilibrium systems [37, 39, 38]. Some iterative methods for solving these problems include Newton method [41], the quasi-Newton methods [4, 6, 9], the Levenberg-Marquardt methods [42, 43], the double direction methods [23, 33, 29], The double step length methods [17, 18, 28], and derivative-free methods [34, 45, 46, 44]. But, the famous method used to solve (1) is Newton method that determines the search direction $d_k$ by solving the following linear system of equations,

$$F_k + F_k' d_k = 0, \tag{3}$$

where $F_k'$ is the Jacobian matrix of $F$ at $x_k$. The Newton method is appealing because it converges quadratically from a reasonably good starting point [14]. Despite its excellent convergence property, the method has some shortcomings, which includes storing of Jacobian matrix and solving system of linear equations in every iteration. In order to overcome some of the challenges associated with Newton method, alternatives such as quasi-Newton methods have been developed [4, 6]. These Methods avoid the computation of the exact Jacobian matrix and a matrix which is an approximation the Jacobian matrix or its inverse is used instead. This matrix is there by updated in every iteration. It has been shown that most of the quasi-Newton methods have supperlinear order of convergence [14]. One of the successful quasi-Newton method, known as Broyden's method, generates a sequence of iterates $\{x_k\}$ using

$$x_{k+1} = x_k - B_k^{-1} F_k, \quad k = 0, 1, 2, \dots, \tag{4}$$

where the Broyden matrix $B_k$ is the approximation of the Jacobian matrix, such that the following quasi-Newton equation

$$B_{k+1}(x_{k+1} - x_k) = F_{k+1} - F_k, \tag{5}$$

is satisfied for all $k$. It is important to note that Broyden's method requires the computation and storage of $n \times n$ matrix at every iteration. Therefore, for large-scale problems, this could result to serious memory constraints. Efforts have been made by different researchers to reduce the storage problem associated with quasi-Newton methods. For instance, some modifications of the Broyden's method have been done in the literature in order to reduce its computational cost [5, 8, 11, 12]. These methods are usually referred to as limited memory Broyden methods [12, 16].

As mentioned earlier, the quasi-Newton methods has contributed in overcoming of the shortcomings of Newton's method which is computing Jacobian matrix in every iteration. However, the prize paid by the quasi-Newton method is that only superlinear rate of convergence can be achieved instead of quadratic rate. In order to improve the convergence order of quasi-Newton method, many higher order approaches have been proposed. There is a great deal of literature on the family of derivative-free methods used to solve nonlinear equations. In [16], a family of conjugate gradient methods for solving nonlinear monotone equations has been presented. The advantage of the method is that, the computation of Jacobian matrix is completely avoided throughout the iteration process. Also, a derivative-free methods for nonlinear monotone equations has been proposed in [26] and it hs shown to converged Q-linearly to the solution of the monotone equations based on the assumption that the underlying function is Lipschitz continuous. Recently. some matrix-free methods have been proposed [19, 20, 21, 24, 21, 25].

Motivated by the above contributions, this paper aimed at proposing the derivative-free method for solving large-scale problem (1) that is globally convergent. The remaining part of the paper is organized as follows. In Section 2, we present the algorithms of the proposed method. Convergence analysis is presented in Section 3. Numerical results of the methods are reported in Section 4. Concluding remarks are given in Section 5.

## 2. Main Result

In this section, we present the proposed method for solving large scale system of nonlinear equations. The method is based on approximation of quasi-Newton's update in (4) via

$$B_k \approx \lambda_k I, \tag{6}$$

where $\lambda_k \in \mathbb{R}^n$ and $I$ is an identity matrix.
In order to enhance good direction toward the solution, we suggest new direction $d_k$ to be defined as

$$d_k = -\lambda_k^{-1} F_k, \tag{7}$$

where $\lambda_k \in \mathbb{R}$ is an acceleration parameter to be determined.
Furthermore, the search direction $d_k$ is usually needed to satisfy the descent condition

$$\nabla f(x_k)^T d_k < 0.$$

Now, consider the Broyden's matrix updating formula given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k}, \tag{8}$$

where $s_k = x_{k+1} - x_k$ and $y_k = F_{k+1} - F_k$. Despite the attractive features of this method, it is not suitable for solving the large-scale problems due the matrix storage at each iteration. Motivated by this reason, this work is aim at proposing a new matrix-free method for solving large-scale problems.
Now, from (6) and (8), it can be deduced that

$$\lambda_{k+1} I = \lambda_k I + \frac{(y_k - \lambda_k s_k)s_k^T}{s_k^T s_k}, \tag{9}$$

and by multiplying (8) by $F_k$, we have

$$\lambda_{k+1} F_k = \lambda_k F_k + \frac{(y_k - \lambda_k s_k)s_k^T F_k}{s_k^T s_k}. \tag{10}$$

Again, multiplying (10) by $s_k^T$, we have

$$\lambda_{k+1} s_k^T F_k = \lambda_k s_k^T F_k + \frac{s_k^T(y_k - \lambda_k s_k)s_k^T F_k}{s_k^T s_k}, \tag{11}$$

Dividing (11) by $s_k^T F_k$, where $s_k^T F_k \neq 0$ yields

$$\lambda_{k+1} = \lambda_k + \frac{s_k^T(y_k - \lambda_k s_k)}{s_k^T s_k}. \tag{12}$$

We finally present our iterative scheme as

$$x_{k+1} = x_k + \alpha_k d_k, \tag{13}$$

where $\alpha_k > 0$ is the step length and $d_k$ is the search direction. Moreover, inexact line search proposed in [9] is used in this work to compute the step length $\alpha_k$ as follows.

Given some positive constants $\eta_1, \quad \eta_2 > 0$ and let $h \in (0, 1)$. Suppose that $\{\omega_k\}$ is a sequence of some positive numbers for which

$$\sum_{k=0}^{\infty} \omega_k < \omega < \infty, \tag{14}$$

and

$$f(x_k + \alpha d_k) - f(x_k) \le -\eta_1 \|\alpha F(x_k)\|^2 - \eta_2 \|\alpha d_k\|^2 + \omega_k f(x_k), \tag{15}$$

where $\alpha = h^i$ with $i$ being the least nonnegative integer for which (15) holds. Set $\alpha_k = \alpha$.

**Algorithm 1:** On Efficient Matrix-Free Method Via Quasi-Newton Approach (EMQN)

**Input:** Given $x_0$, $\lambda_0 = 0.01$, $\epsilon = 10^{-4}$, set $k = 0$.
**Step 1:** Compute $F(x_k)$.
**Step 2:** If $\|F_k\| \le \epsilon$ then stop, else go to **Step 3**.
**Step 3:** Compute $d_k = -\lambda_k^{-1} F(x_k)$.
**Step 4:** Compute step length $\alpha_k$ (15).
**Step 5:** Set $x_{k+1} = x_k + \alpha_k d_k$.
**Step 6:** Compute $F_{k+1}$.
**Step 7:** Determine $\lambda_{k+1} = \lambda_k + \frac{s_k^T (y_k - \lambda_k s_k)}{s_k^T s_k}$.
**Step 9:** Set $k = k + 1$, and go to **Step 2**.

**Remark 2.1.** *It can be seen that the parameter $\lambda_{k+1}$ defined by* (12) *is a scalar for all $k$. In addition, the gradient of $F$ is not needed in the implementation of Algorithm 1. With these into consideration, we can conclude that the Algorithm 1 is derivative-free as well as matrix-free. Therefore, Algorithm 1 is suitable for large-scale problems as well as nonsmooth problems. Furthermore, we show in Lemma 3.4 that the search direction generated by Algorithm 1 is sufficiently descent.*

## 3. Convergence Result

In this section, we present the global convergence of our method (EMQN). To start, let the level set be defined as

$$\Omega = \{x | \|F(x)\| \le \|F(x_0)\|\}. \tag{16}$$

**Assumption 3.1.** *We now state the following assumptions to establish the convergence result of EMQN Algorithm .*
*(1) There exists a point $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.*
*(2) $F$ is continuously differentiable in some neighborhood say $A$ of $x^*$ containing $\Omega$.*
*(3) The Jacobian of function $F$ is positive definite bounded on $A$, namely, there exists some positive constants $G > g > 0$ such that*

$$\|F'(x)\| \le G, \quad \forall x \in A, \tag{17}$$

*and*

$$g\|d\|^2 \le d^T F'(x) d, \quad \forall x \in A, d \in \mathbb{R}^n. \tag{18}$$

**Remark 3.2.** *Assumption* (3.1) *implies that there exists a constants $G > g > 0$ such that*

$$g\|d\| \le \|F'(x)d\| \le G\|d\|, \quad \forall x \in A, d \in \mathbb{R}^n. \tag{19}$$

$$g\|x - y\| \le \|F(x) - F(y)\| \le G\|x - y\|, \quad \forall x, y \in A. \tag{20}$$

Since $\lambda_k I$ approximates $F_k'$ along direction $d_k$, let us state the following assumption.

**Assumption 3.3.** $\lambda_k I$ *is a good approximation to $F'(x_k)$, i.e.,*

$$\|(F'(x_k) - \lambda_k I) d_k\| \le \epsilon \|F(x_k)\|, \tag{21}$$

*where $\epsilon \in (0, 1)$ [13].*

**Lemma 3.4.** *Suppose that Assumption* (3.3) *holds and let* $\{x_k\}$ *be generated by EMQN algorithm. Then* $d_k$ *is a descent direction of* $f$ *at* $x_k$ *i.e*

$$\nabla f(x_k)^T d_k < 0. \tag{22}$$

*Proof.* From (7), we have

$$\begin{aligned} \nabla f(x_k)^T d_k &= F_k^T F_k' d_k \\ &= F_k^T[(F_k' - \lambda_k I)d_k - F_k] \\ &= F_k^T(F_k' - \lambda_k I)d_k - \|F_k\|^2, \end{aligned} \tag{23}$$

by Chauchy-Schwarz we have,

$$\begin{aligned} \nabla f(x_k)^T d_k &\le \|F_k\|\|(F_k' - \lambda_k I)d_k\| - \|F_k\|^2 \\ &\le -(1-\epsilon)\|F_k\|^2. \end{aligned} \tag{24}$$

Hence for $\epsilon \in (0,1)$ we have (22).
Since the search direction satisfied the decent condition in (22), it means that the inequality $\|F_{k+1}\| \le \|F_k\|$ holds. □

**Lemma 3.5.** *Suppose that Assumption* (3.3) *holds and* $\{x_k\}$ *be generated by EMQN algorithm. Then* $\{x_k\} \subset \Omega$.

*Proof.* From lemma (3.4) we have $\|F_{k+1}\| \le \|F_k\|$. In addition, for all $k$ we have

$$\|F_{k+1}\| \le \|F_k\| \le \|F_{k-1}\| \le \ldots \le \|F_0\|.$$

This shows that $\{x_k\} \subset \Omega$. □

**Lemma 3.6.** *(see[3]) Suppose that Assumption* (3.1) *holds and* $\{x_k\}$ *be generated by EMQN algorithm. Then there exists a constant* $g > 0$ *such that for all* $k$

$$y_k^T s_k \ge g\|s_k\|^2. \tag{25}$$

**Lemma 3.7.** *Suppose that Assumption* (3.1) *holds and* $\{x_k\}$ *is generated by EMQN algorithm. Then we have*

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0, \tag{26}$$

*and*

$$\lim_{k\to\infty} \|\alpha_k F_k\| = 0. \tag{27}$$

*Proof.* By (15), we have for all $k > 0$,

$$\begin{aligned} \eta_2\|\alpha_k d_k\|^2 &\le \eta_1\|\alpha_k F_k\|^2 + \eta_2\|\alpha_k d_k\|^2 \\ &\le \|F_k\|^2 - \|F_{k+1}\|^2 + \omega_k\|F_k\|^2. \end{aligned} \tag{28}$$

By summing the inequality above, we have

$$\begin{aligned} \eta_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 &\le \sum_{i=0}^{k} \left(\|F_i\|^2 - \|F_{i+1}\|^2\right) + \sum_{i=0}^{k} \omega_i \|F_i\|^2 \\ &= \|F_0\|^2 - \|F_{k+1}\|^2 + \sum_{i=0}^{k} \omega_i \|F_i\|^2 \\ &\le \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{k} \omega_i \\ &\le \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{\infty} \omega_i. \end{aligned} \tag{29}$$

So from the level set and fact that $\{\omega_k\}$ satisfies (14) then the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ is convergent. This implies (26). Following the similar arguments as above but with $\eta_1\|\alpha_k F_k\|^2$ on the left hand side, we obtain (27). □

**Lemma 3.8.** *Suppose that Assumption* (3.1) *holds and let* $\{x_k\}$ *be generated by algorithm 1. Then there exists a constant* $m_3 > 0$ *such that for all* $k > 0$,

$$\|d_k\| \le m_3. \tag{30}$$

*Proof.* From (7) and (25) we have,

$$\begin{aligned}
\|d_k\| &= \| - \lambda_k^{-1} F_k\| \\
&= \left\| -\left(\lambda_{k-1} I + \frac{(y_{k-1} - \lambda_{k-1}s_{k-1})s_{k-1}^T}{s_{k-1}^T s_{k-1}}\right)^{-1} F_k \right\| \\
&= \left\| -\left(\lambda_{k-1} + \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T s_{k-1}} - \lambda_k \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T s_{k-1}}\right)^{-1} F_k \right\| \\
&= \left\| -\frac{\|s_{k-1}\|^2}{y_{k-1}^T s_{k-1}} F_k \right\| \\
&\le \frac{\|s_{k-1}\|^2 \|F_k\|}{g\|s_{k-1}\|^2} \\
&\le \frac{\|F_0\|}{g}.
\end{aligned} \tag{31}$$

Taking $m_3 = \frac{\|F_0\|}{g}$, we have (30). □

**Theorem 3.9.** *Suppose that Assumption* (3.1) *holds and* $\{x_k\}$ *is generated by EMQN Algorithm. We further assume that for all* $k > 0$,

$$\alpha_k \ge h \frac{|F_k^T d_k|}{\|d_k\|^2}, \tag{32}$$

*where* $h > 0$. *Then*

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{33}$$

*Proof.* From Lemma (3.8), we have (30). Therefore by (26) and the boundedness of $\{\|d_k\|\}$, we have

$$\lim_{k\to\infty} \alpha_k \|d_k\|^2 = 0. \tag{34}$$

From (32) and (34), we have

$$\lim_{k\to\infty} |F_k^T d_k| = 0. \tag{35}$$

On the other hand from (7), we have

$$F_k^T d_k = -\lambda_k^{-1}\|F_k\|^2, \tag{36}$$

$$\begin{aligned}
\|F_k\|^2 &= | - F_k^T d_k \lambda_k| \\
&\le |F_k^T d_k||\lambda_k|.
\end{aligned} \tag{37}$$

By using (20), we obtain

$$\lambda_k = \lambda_{k-1} + \frac{y_{k-1}^T s_{k-1}}{\|s_{k-1}\|^2} - \frac{\lambda_{k-1}\|s_{k-1}\|^2}{\|s_{k-1}\|^2} = \frac{y_{k-1}^T s_{k-1}}{\|s_{k-1}\|^2} \le \frac{\|y_{k-1}\|\|s_{k-1}\|}{\|s_{k-1}\|^2} \le G,$$

which means,$|\lambda_k| \leq G$. So from (37), we have

$$\|F_k\|^2 \leq |F_k^T d_k| G. \tag{38}$$

Thus,

$$0 \leq \|F_k\|^2 \leq |F_k^T d_k| G \to 0. \tag{39}$$

Therefore,

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{40}$$

The proof is completed. □

## 4. Numerical results

In this section, some numerical results are presented to demonstrate the efficiency of the proposed method by comparing it with the following existing methods in the literature.

- An improved derivative-free method via double direction approach for solving systems of nonlinear equations (**IDFDD**) [3].
- Classical Broyden's method (**CBM**) for solving system of nonlinear equations.

The three algorithms were implemented using the same line search (15) in the course of the experiments and the following parameters are set: $\eta_1 = \eta_2 = 10^{-4}$, $h = 0.35$ and $\omega_k = \dfrac{1}{(k+1)^2}$. However, for the classical Broyden's method, we set $B_0 = I$, $I$ is an identity matrix.

The computer codes used were written in Matlab 8.3.0.532 (R2014a) and run on a personal computer equipped with a 1.40.00 GHz CPU processor and 4 GB RAM memory. We have tried the three methods on three test problems with different initial points and dimension ($n-$values) between 100 to 10,000. The iteration is set to stop for all the methods if $\|F_k\| \leq 10^{-4}$. The symbol '–' represents failure due to:

(i) Failure to complete execution due to insufficient memory.

(ii) Number of iterations exceed 1000 but no $x_k$ satisfy the stopping criterion.

Table 1: Initial points

| INITIAL GUESS (IP) | VALUES |
|---|---|
| $x_1$ | $(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})^T$ |
| $x_2$ | $(-1.5, -1.5, \ldots, -1.5)^T$ |
| $x_3$ | $(-25, -25, \ldots, -25)^T$ |
| $x_4$ | $(5, 5, \ldots, 5)^T$ |
| $x_5$ | $(14, 14, \ldots, 14)^T$ |

**Problem 1**: [4].

$$F_i(x) = 2x_i - \sin|x_i|, \quad i = 1, 2, \ldots, n.$$

**Problem 2**: [10].

$$F_i(x) = \cos(x_i^2 - 1)^2 - 1, \quad i = 1, 2, \ldots, n.$$

**Problem 3**.

$$F_1(x) = \frac{1}{3}x_1^3 + \frac{1}{2}x_2^2$$
$$F_i(x) = -\frac{1}{2}x_i^2 + \frac{i}{3}x_i^3 + \frac{1}{2}x_{i+1}^2, \quad i = 1, 2, \ldots, n-1,$$
$$F_n(x) = -\frac{1}{2}x_n^2 + \frac{n}{3}x_n^3.$$

The results of the numerical experiments for the IDFDD and CBM methods as well as our proposed method

Table 2: Numerical results of EMQN, CMB and IDFDD methods for problem 1

| | | EMQN | | CBM | | IDFDD | |
|---|---|---|---|---|---|---|---|
| Dimension | Initial Guess | NIT | CPUT | NIT | CPUT | NIT | CPUT |
| 100 | $x_1$ | 28 | 0.142571 | 19 | 0.535173 | 23 | 0.143207 |
| | $x_2$ | 24 | 0.028437 | 8 | 0.099478 | 25 | 0.112461 |
| | $x_3$ | 9 | 0.055164 | 37 | 0.495608 | - | - |
| | $x_4$ | 147 | 0.366485 | 78 | 0.931389 | 366 | 1.349186 |
| | $x_5$ | 9 | 0.044106 | 109 | 1.283275 | - | - |
| 1000 | $x_1$ | 31 | 0.107026 | 18 | 27.2388 | 26 | 0.246356 |
| | $x_2$ | 25 | 0.119676 | 8 | 11.92841 | 28 | 0.344145 |
| | $x_3$ | 9 | 0.086169 | 42 | 91.47828 | - | - |
| | $x_4$ | 126 | 0.812398 | 140 | 242.139 | 397 | 3.385763 |
| | $x_5$ | 9 | 0.070279 | 27 | 41.0478 | - | - |
| 10000 | $x_1$ | 34 | 0.654757 | - | - | 27 | 1.842495 |
| | $x_2$ | 29 | 0.619744 | - | - | 31 | 1.519259 |
| | $x_3$ | 9 | 0.383985 | - | - | - | - |
| | $x_4$ | 146 | 3.49417 | - | - | 428 | 23.4393 |
| | $x_5$ | 9 | 0.384739 | - | - | - | - |

Table 3: Numerical results of EMQN, CMB and IDFDD methods for problem 2

| | | EMQN | | CBM | | IDFDD | |
|---|---|---|---|---|---|---|---|
| Dimension | Initial Guess | NIT | CPUT | NIT | CPUT | NIT | CPUT |
| 100 | $x_1$ | 9 | 0.054347 | 10 | 0.592524 | 4 | 0.091207 |
| | $x_2$ | 6 | 0.026207 | 6 | 0.110625 | - | - |
| | $x_3$ | 10 | 0.01284 | 8 | 0.183785 | 4 | 0.016239 |
| | $x_4$ | 7 | 0.009098 | 10 | 0.196762 | - | - |
| | $x_5$ | 8 | 0.010415 | 10 | 0.44328 | - | - |
| 1000 | $x_1$ | 11 | 0.041969 | 11 | 19.17501 | - | - |
| | $x_2$ | 7 | 0.022809 | 7 | 12.06882 | - | - |
| | $x_3$ | 11 | 0.024054 | 9 | 15.57875 | 4 | 0.095884 |
| | $x_4$ | 8 | 0.032852 | 11 | 19.02109 | - | - |
| | $x_5$ | 9 | 0.051642 | 11 | 19.28136 | - | - |
| 10000 | $x_1$ | 12 | 0.177906 | - | - | - | - |
| | $x_2$ | 8 | 0.271242 | - | - | - | - |
| | $x_3$ | 12 | 0.269865 | - | - | 4 | 0.566027 |
| | $x_4$ | 10 | 0.171702 | - | - | - | - |
| | $x_5$ | 13 | 0.305941 | - | - | - | - |

are reported in Tables 2-4, where NIT and CPUT are respectively stand for the number of iterations and the number of time taken for each method to successfully obtained the solution of each problem. Tables 3-4 indicated that the proposed method EMQN has minimum number of iterations and CPU time, compared to CBM and IDFDD methods, except at Table 2 with initial guesses $x_1$ and $x_4$ for the dimension 100 and $x_1$ and $x_2$ in 1000 dimension, where the number of iteration of CBM method is less than that of EMQN and IDFDD methods. Therefore, EMQN method out performed CBM and IDFDD methods. One can easily observe that our claim is fully justified from the Tables, that is, the proposed method has less CPU time and

Table 4: Numerical results of EMQN, CMB and IDFDD methods for Problem 3

| | | EMQN | | CBM | | IDFDD | |
|---|---|---|---|---|---|---|---|
| Dimension | Initial Guess | NIT | CPUT | NIT | CPUT | NIT | CPUT |
| 100 | $x_1$ | 12 | 0.239779 | 21 | 0.700504 | 27 | 1.230928 |
| | $x_2$ | 11 | 0.226276 | 19 | 0.513159 | 29 | 1.29121 |
| | $x_3$ | 13 | 0.270103 | 21 | 0.656313 | 23 | 1.156257 |
| | $x_4$ | 14 | 0.244887 | 24 | 0.687868 | 28 | 0.968806 |
| | $x_5$ | 14 | 0.226145 | 25 | 0.764494 | 26 | 1.047147 |
| 1000 | $x_1$ | 14 | 0.944261 | 23 | 34.40515 | 35 | 5.13596 |
| | $x_2$ | 11 | 0.892505 | 28 | 76.17269 | 29 | 5.233836 |
| | $x_3$ | 13 | 0.951678 | 21 | 46.46457 | 31 | 5.45977 |
| | $x_4$ | 14 | 0.947066 | 24 | 54.18002 | 32 | 6.261874 |
| | $x_5$ | 14 | 1.036368 | 23 | 48.51827 | 40 | 7.275419 |
| 10000 | $x_1$ | 14 | 88.4897 | - | - | 35 | 647.7165 |
| | $x_2$ | 12 | 102.4096 | - | - | 29 | 526.0935 |
| | $x_3$ | 15 | 99.16974 | - | - | 31 | 524.7104 |
| | $x_4$ | 14 | 75.62304 | | | 32 | 480.4203 |

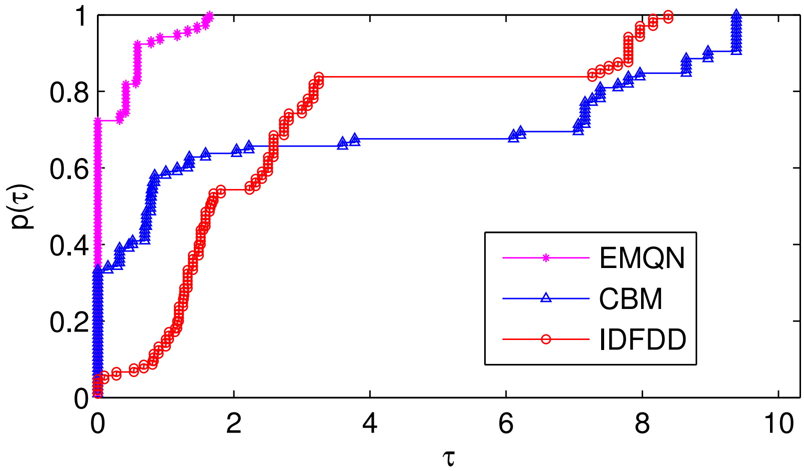


Figure 1: Performance profile of EMQN,CBM and IDFDD methods with respect to the number of iteration for the problems 1-3.

number of iterations for each of the test problems with the exception of Problem 1. Furthermore, on average, the CPU time of the proposed method is the smallest which signifies that our method is fully derivative-free and matrix-free (i.e., no computation of matrix at all).

Figures 1-2 show the summery of the numerical performance of the IDFDD and CBM methods against the proposed method in terms of iterations number and CPU time. The summery is evaluated based on the famous performance profiles developed by Dolan and Moré [4]. This means, for each method, the fraction $P(\tau)$ of the problems for which the method falls within a factor $\tau$ of the best time is plotted. The curve that stays longer on the vertical axis corresponds to the method that solved highest percentage of the test problems considered in a time that was within a factor $\tau$ of the best time.

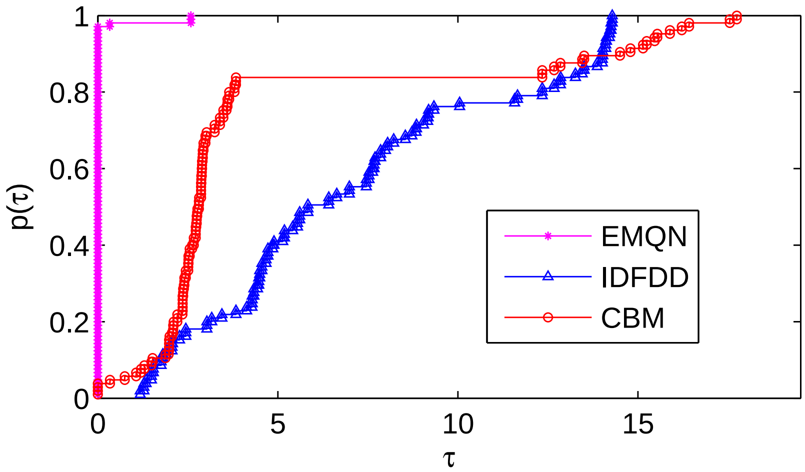


Figure 2: Performance profile of EMQN,CBM and IDFDD methods with respect to the CPU time (in second) for the problems 1-3.

## 5. Conclusion

In this paper, an efficient matrix-free method via quasi-Newton update for handling nonlinear system of equations has been developed. This was achieved by approximating the Broyden's Update via acceleration parameter. The proposed method is completely matrix-free iterative method that is globally convergent under certain appropriate conditions. The efficiency as well as the performance of the proposed method have been compared with that of classical broyden method (CBM) and IDFDD method [3]. Numerical comparisons have been done using a set of large-scale test problems. Moreover, Table 2-4 and Figure 1-2, showed that the proposed method is quite efficient because it has the least number of iteration compared to IDFDD and CBM methods. Future research include using the proposed method to solve nonlinear problems as discussed in [35, 36, 40].

## Acknowledgment

Nuttapol Pakkaranang would like to thank Phetchabun Rajabhat University.

## References

[1] R.P. Agarwal, B. Xu and W. Zhang, Stability of functional equations in single variable, J. Math. Anal. Appl., 288 (2003), 852–869.
[2] J.A. Baker, The stability of certain functional equations, Proc. Amer. Math. Soc., 112 (1991), 729–732.
[3] A.S. Halilu and M.Y. Waziri, An improved derivative-free method via double direction approach for solving systems of nonlinear equations, J. of the Ramanujan Math. Soc., 33 (2018), 75–89.
[4] J.E. Dennis and J.J. More, A characterization of superlinear convergence and its application to quasi-Newton methods, Math. Comp., 28 (1974), 549-560.
[5] M. Mamat, K. Muhammad and M.Y. Waziri, Trapezoidal Broyden's method for solving systems of nonlinear equations, Appl. Math. Sci., 6 (2014), 251–260.
[6] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, Math. Comput., 19 (1965), 577-593.
[7] M.Y. Waziri, Y.M. Kufena, and A.S. Halilu, Derivative-free three-term spectral conjugate gradient method for symmetric nonlinear equations, Thai J. Math., 18 (2020), 1417–1431.
[8] M. Ziani and H.F. Guyomarch, An autoadaptative limited memory Broyden's method to solve systems of nonlinear equations, Appl. Math. Comput., 205 (2008), 205-215.
[9] D. Li and M. Fukushima, A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equation, SIAM J. Numer. Anal., 37 (1999), 152–172.
[10] M.Y. Waziri and L.Y. Uba, Three-step derivative-Free diagonal updating method for solving large-scale systems of nonlinear equations, J. Numer. Math. Stoch., 6 (2014), 73–83.
[11] W. Leong, M.A. Hassan and M.Y. Waziri, A matrix-free quasi-Newton method for solving large-scale nonlinear systems, Comput. Math. Appl., 62 (2011), 2354–2363.
[12] A. Ramli, M.L. Abdullah and M. Mamat, Broyden's method for solving fuzzy nonlinear equations, Adv. Fuzzy Syst., (2010), Art. ID 763270, 6 pages.
[13] A.S. Halilu and M.Y. Waziri, A transformed double step length method for solving large-scale systems of nonlinear equations, J. Numer. Math. Stoch., 9 (2017), 20–32.
[14] C.T. Kelley, Solving nonlinear equations with Newtons method, SIAM, Philadelphia (2003).
[15] J.E. Dennis and R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, SIAM, Philadelphia (1993).
[16] W. Sun and Y.X. Yuan, Optimization theory and methods; Nonlinear programming, Springer, New York (2006).
[17] A.S. Halilu and M.Y Waziri, En enhanced matrix-free method via double step lengthapproach for solving systems of nonlinear equations, International Journal of Applied Mathematical Research, 6(4) (2017), 147–156.
[18] A.S. Halilu,A. Majumder, M.Y. Waziri, H. Abdullahi, Double direction and step length method for solving system of nonlinear equations, Euro. J. Mol. Clinic. Med., 7(7) (2020), 3899–3913.
[19] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya and K. Sitthithakerngkiet, An efficient DY-type spectral conjugate gradient method for system of nonlinear monotone equations with application in signal recovery, AIMS Math., 6 (2021): 8078–8106.
[20] A.M. Awwal, P. Kumam and H. Mohammad, Iterative algorithm with structured diagonal Hessian approximation for solving nonlinear least squares problems, J. Nonlinear Convex Anal., 22(6) (2021), 1173–1188.
[21] A.M. Awwal, P. Kumam, K. Sitthithakerngkiet, A.M. Bakoji, A.S. Halilu and I.M. Sulaiman, Derivative-free method based on DFP updating formula for solving convex constrained nonlinear monotone equations and application, AIMS Math., 6(8) (2021), 8792–8814.
[22] S. Aji, P. Kumam, A.M. Awwal, M.M. Yahaya and W. Kumam, Two hybrid spectral methods with inertial effect for solving system of nonlinear monotone equations with application in robotics, IEEE Access, 9 (2021), 30918–30928.
[23] A.S. Halilu and M.Y. Waziri, Solving systems of nonlinear equations using improved double direction method, J. Nigerian Math. Soc., 39(2) (2020), 287–301.

[24] A.M. Awwal, P. Kumam, L. Wang, S. Huang and W. Kumam, Inertial-based derivative-free method for system of monotone nonlinear equations and application, IEEE Access, 8 (2020) 226921–226930.

[25] A.M. Awwal, L. Wang, P. Kumam, H. Mohammad and W. Watthayu, A projection Hestenes–Stiefel method with spectral parameter for nonlinear monotone equations and signal processing, Math. Comput. Appl., 25(2) (2020), Art. ID 27-28, 29 pages.

[26] Y.B. Zhao and D. Li, Monotonicity of fixed point and normal mapping associated with variational inequality and its application, SIAM J. Optim., 11 (2001), 962–997.

[27] M. Li, H. Liu and Z. Liu, A new family of conjugate gradient methods for unconstrained optimization, J. Appl. Math. Comput., 58 (2018) 219–234.

[28] A.S. Halilu, and M.Y. Waziri, Inexact double step length method for solving systems of nonlinear equations, Stat. Optim. Inf. Comput., 8(1) (2020), 165–174.

[29] H.Abdullahi, A.S. Halilu, and M.Y. Waziri, A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear systems, Journal of Numerical Mathematics and Stochastics, 10(1) (2018), 32–44.

[30] Y.H. Dai and C.X. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved wolfe line search, SIAM J. Optim., 23 (2013), 296–320.

[31] Y.H. Dai and L.Z. Lio, New conjugacy conditions and related nonlinear conjugate gradient methods, Appl. Math. Optim., 43 (2001), 87–101.

[32] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, Pac. J. Optim., 2 (2006), 35–58.

[33] A.S. Halilu, M.Y. Waziri and I. Yusuf, Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations, Yugosl. J. Oper. Res., 30(4) (2020), 399–412.

[34] M.Y. Waziri, K. Ahmad, and A.S. Halilu, Enhanced Dai–Liao conjugate gradient methods for systems of monotone nonlinear equations, SeMA J., 78(1) (2020), 15–51.

[35] L. Li and D. Wu, The convergence of Ishikawa iteration for generalized Φ-contractive mappings, Results in Nonlinear Analysis, 4(1) (2021), 47–56.

[36] P. Lo'lo' and M. Shabibi, Common best proximity points theorems for H-contractive non-self mappings, Advances in the Theory of Nonlinear Analysis and its Application, 5(2) (2021), 173–179.

[37] K. Meintjes, and A.P. Morgan, A methodology for solving chemical equilibrium systems, Appl. Math. Comput., 22 (1987), 333–361.

[38] M. Sun, M.Y. Tian, and Y.J. Wang, Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization, Discrete Dyn. Nat. Soc., Art. ID 4745759, (2019) 1–14.

[39] A.M. Awwal, L. Wang, P. Kumam, and H. Mohammad, A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems, Symmetry, 12(6) (2020), Art. ID 874, 20 pages.

[40] N. Wairojjana, N. Pakkaranang, I. Uddin, P. Kumam and A.M. Awwal, Modified proximal point algorithms involving convex combination technique for solving minimization problems with convergence analysis, Optimization, 69(7–8) (2020), 1655–1680.

[41] J.E. Dennis and R.B Schnabel, Numerical method for unconstrained optimization and non-linear equations, practice Hall, Englewood cliffs, NJ, USA, 1983.

[42] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Indust. Appl. Math., 11 (1963), 431–441.

[43] Y.B. Musa, M.Y. Waziri, and A.S. Halilu, On computing the regularization Parameter for the Levenberg-Marquardt method via the spectral radius approach to solving systems of nonlinear equations, J. Numer. Math. Stoch., 9(1) (2017), 80–94.

[44] A.S. Halilu, M.K. Dauda, M.Y. Waziri, and M. Mamat, A derivative-free decent method via acceleration parameter for solving systems of nonlinear equations, Open J. sci. tech., 2(3) (2019) 1-4.

[45] A.S. Halilu, A. Majumder, M.Y. Waziri, K. Ahmed, Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach, Math. Comput. Simulation, 187 (2021), 520–539.

[46] M.Y. Waziri, H.U. Muhammad, A.S. Halilu, and K. Ahmad, Modified matrix-free methods for solving system of nonlinear equations, (2020) DOI: 10.1080/02331934.2020.1778689.