Journal of the Nigerian Mathematical Society
Vol. 39, Issue 2, pp. 287-301, 2020

SOLVING SYSTEM OF NONLINEAR EQUATIONS USING IMPROVED DOUBLE DIRECTION METHOD

A. S. HALILU[1] AND M. Y. WAZIRI

ABSTRACT.The fundamental reason behind double direction approach is that, there are two corrections in the scheme. If one correction fails during iterative process then the other one will correct the system. Therefore, this research aims to present a derivative-free method for solving large-scale system of nonlinear equations via double direction approach. The acceleration parameter used in this approach approximated the Jacobian matrix in order to form a derivative-free method by reducing two direction presented in double direction scheme into a single one. under mild conditions, the proposed method is proved to be globally convergent using derivative-free line search. Numerical results recorded in this paper using a set of large-scale test problems show that the proposed approach is successful for solving large-scale problems.

Keywords and phrases: Acceleration parameter, Double direction; Global Convergence.
2010 Mathematical Subject Classification:65H11, 65K05, 65H12, 65H18

1. INTRODUCTION

Due to the fact that most problems arising from engineering, biology, mathematics, physics, and many other branches of science, are naturally nonlinear in nature. In this paper following system of nonlinear equations is to be considered.

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map. Throughout this paper, the space $\mathbb{R}^n$ denote the $n-$dimensional real space equipped with the Euclidean norm $\|\cdot\|$.

The most popular schemes for solving (1) are based on successive linearization [3, 4, 5], where the search direction d_k is obtained by

Received by the editors April 29, 2018; Revised July 14, 2020 ; Accepted: July 14, 2020
www.nigerianmathematicalsociety.org; Journal available online at https://ojs.ictp.it/jnms/
[1]Corresponding author

solving the following linear equation:

$$F(x_k) + F'(x_k)d_k = 0, \tag{2}$$

where, $F'(x_k)$ is the Jacobian matrix of $F(x_k)$ at x_k or an approximation of it.

The well-known approach for solving (1) is the Newton's method [5, 10]. The method has some nice properties, such as a fast convergence rate from a reasonably good starting point [5]. Despite the appealing characteristics of the Newton methods, Jacobian matrix will be computed at each iteration, so they are not ideal for solving large-scale problems. Due to the shortcomings of Newton method, the double direction method has been proposed in [2] and the iterative procedure is given as:

$$x_{k+1} = x_k + \alpha_k d_k + \alpha_k^2 c_k, \tag{3}$$

where x_{k+1} represents a new iterative point, x_k is the previous iteration, and α_k denotes the step length, while b_k and c_k are search directions respectively. In addition, (1) can be obtained from an unconstrained optimization problem [14]. Let f be a norm function defined by

$$f(x) = \frac{1}{2}\|F(x)\|^2. \tag{4}$$

The nonlinear equations problem (1) is analogous to the following global optimization problem

$$minf(x), \quad x \in \mathbb{R}^n, \quad \text{where}, f : \mathbb{R}^n \to \mathbb{R}.$$

The idea of double direction approach is presented by Duranovic-Milicic, in [2], using multi-step iterative scheme to generate new iterate. Notwithstanding, Duranovic et.al [1] also proposed a multi-step algorithm for minimizing a non-differentiable function using double direction approach. The approach in [1], Motivated Petrovic and Stanimirovic, [15] and proposed a double direction method for solving unconstrained optimization problems. In their work, an approximation to Hessian matrix is obtained via acceleration parameter $\gamma_k > 0$ i.e., $\bigtriangledown^2 f(x_k) \approx \gamma_k I$, where, I is an identity matrix. The attractive feature of the scheme in [15], is that, the two directions presented are derivative-free, in the first direction, the Hessian matrix is approximated with diagonal matrix via acceleration parameter and the second one is presented as steepest decent direction. For this reason, the method is opportuned to solve the large-scale problems. Nonetheless, the study of derivative-free double direction methods for solving system of nonlinear equations is very rare in the literature, this motivated Halilu and Waziri [8]

and used the scheme in (3), and proposed a derivative-free method via double direction approach for solving system of nonlinear equations. The Jacobian matrix is approximated via acceleration parameter $\gamma_k > 0$ i.e., $F'_k \approx \gamma_k I$, where, I is an identity matrix. The study of double direction is further improved by Habibu et al. [7] and solve conjugate gradient method for solving symmetric nonlinear equations. In their scheme, the two direction are presented as conjugate gradient and steepest descent direction respectively. Despite, the good convergence properties of the methods in [8, 7], but it is numerical performance defined weaker due to the present of steepest descent direction in the schemes. Therefore, motivated by this idea, we aimed at developing a matrix-free method with line search for solving system of nonlinear equations, without computing the Jacobian matrix with less number of iterations and CPU time that is globally convergent. This is made possible by making the two directions in (3) to a single direction, and make the numerical comparison with a single direction method existing in the literature.

There are several procedure for the choice of the search direction [1, 5, 12, 13, 14]. In steepest descent method the direction d_k is defined by $d_k = -F(x_k)$ [18]. The conjugate gradient direction for solving system of nonlinear equations has received a good attention and take an appropriate progress, where the direction d_k is defined by

$$d_k = \begin{cases} -F(x_k), & if \quad k = 0 \\ -F(x_k) + \beta_k d_{k-1}, & if \quad k \geq 1 \end{cases}$$

where β_k is a CG-parameter.See[6, 7, 9].

The step length α_k can also be computed either exact or in exact. It is very expensive to find exact step length in practical computation. Therefore the most frequently used line search in practice is inexact line search [6, 8, 9, 16, 20, 22],which sufficiently decrease the function values along the ray $x_k + \alpha_k d_k$, $\alpha_k > 0$. i.e $\|F(x_k + \alpha_k d_k)\| \leq \|F(x_k)\|$.

We organized the paper as follows; In the next section, we present the proposed method, convergence results are presented in section 3. Some numerical results are reported in section 4. Finally we made conclusions in section 5.

2. MAIN RESULT

In this section, we propose to reduce the two directions (3) into a single one. In order to incorporate more information of the iterates at each iteration and to improve good direction towards the solution, we suggest a new directions d_k and c_k in (3) to be defined as:

$$d_k = -\gamma_k^{-1} F(x_k) = c_k, \tag{5}$$

where, $\gamma_k \in \mathbb{R}$ is an acceleration parameter.
By putting (5) into (3) we obtained

$$x_{k+1} = x_k - (\alpha_k + \alpha_k^2)\gamma_k^{-1} F(x_k), \tag{6}$$

from (6) we can easily show that, our direction is :

$$d_k = -\gamma_k^{-1} F(x_k). \tag{7}$$

The acceleration parameter, γ_k is obtained by applying Taylor's expansion of the first order as follows:

$$F(x_{k+1}) \approx F(x_k) + F'(\xi)(x_{k+1} - x_k), \tag{8}$$

where the parameter ξ fulfills the conditions $\xi \in [x_k, x_{k+1}]$,

$$\xi = x_k + \delta(x_{k+1} - x_k) = x_k - \delta(\alpha_k + \alpha_k^2) d_k, \quad 0 \leq \delta \leq 1. \tag{9}$$

putting in mind that the distance between x_k and x_{k+1} is small enough, Taking $\delta = 1$ in (9), we get $\xi = x_{k+1}$. So we are interested to approximate Jacobian via

$$F'(\xi) \approx \gamma_{k+1} I. \tag{10}$$

Now from (8) and (10) its not difficult to verify:

$$y_k = \gamma_{k+1} s_k \tag{11}$$

where, $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$.
By multiplying y_k^T to the both side of (11) we obtained the acceleration parameter as:

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2) y_k^T d_k}, \tag{12}$$

From (6) and (7) we have the general scheme as:

$$x_{k+1} = x_k + (\alpha_k + \alpha_k^2) d_k. \tag{13}$$

We use the derivative-free line used in [11, 14] in order to compute our step length α_k.

Let $\omega_1 > 0$, $\omega_2 > 0$ and $r \in (0,1)$ be constants and let $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty. \tag{14}$$

Let $\alpha_k = r^{i_k}$, i_k is the smallest non negative integer i that satisfies

$$f(x_k+(\alpha_k+\alpha_k^2)d_k)-f(x_k) \leq -\omega_1\|\alpha_k F(x_k)\|^2-\omega_2\|\alpha_k d_k\|^2+\eta_k f(x_k). \tag{15}$$

Now we describe the algorithm of the proposed method as follows:

Algorithm 1(SSIDD)

STEP 1: Given x_0, $\gamma_0 = 1$, $\epsilon = 10^{-4}$, set $k = 0$.
STEP 2: Compute $F(x_k)$.
STEP 3: If $\|F(x_k)\| \leq \epsilon$ then stop, else goto STEP 4.
STEP 4: Compute search direction $d_k = -\gamma_k^{-1}F(x_k)$.
STEP 5: Compute step length α_k(using (15)).
STEP 6: Set $x_{k+1} = x_k + (\alpha_k + \alpha_k^2)d_k$.
STEP 7: Compute $F(x_{k+1})$.
STEP 8: Update $\gamma_{k+1} = \dfrac{y_k^T y_k}{(\alpha_k + \alpha_k^2)y_k^T d_k}$.
STEP 9: Set $k = k + 1$, and go to STEP 3.

3. CONVERGENCE ANALYSIS

In this section we present the global convergence of our method (SSIDD). To begin with, let us defined the level set

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

In order to analyze the convergence of algorithm 1 we need the following assumption:

Assumption 1.

(1) There exists $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.
(2) F is continuously differentiable in some neighborhood say N of x^* containing Ω.
(3) The Jacobian of F is bounded and positive definite on N, i.e., there exists a positive constants $M > m > 0$ such that

$$\|F'(x)\| \leq M \quad \forall x \in N, \tag{17}$$

and

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

From the level set we have:

$$\|F(x)\| \leq m_1 \quad \forall x \in \Omega. \tag{19}$$

Remarks:
Assumption 1 implies that there exists a constants $M > m > 0$ such that

$$m\|d\| \leq \|F'(x)d\| \leq M\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{20}$$

$$m\|x-y\| \leq \|F(x)-F(y)\| \leq M\|x-y\| \quad \forall x, y \in N. \tag{21}$$

Since $\gamma_k I$ approximates $F'(x_k)$ along direction s_k, now we consider the following assumption.

Assumption 2.
$\gamma_k I$ is a good approximation to $F'(x_k)$, i.e.,

$$\|(F'(x_k) - \gamma_k I)d_k\| \leq \epsilon\|F(x_k)\| \tag{22}$$

where $\epsilon \in (0,1)$ is a small quantity [17].

Lemma 1. Suppose assumption 2 holds and let $\{x_k\}$ be generated by algorithm 1. Then d_k is a descent direction for $f(x_k)$ at x_k i.e.,

$$f(x_k)^T d_k < 0. \tag{23}$$

Proof. From (7), we have

$$\begin{aligned} f(x_k)^T d_k &= F(x_k)^T F'(x_k) d_k \\ &= F(x_k)^T[(F'(x_k) - \gamma_k I)d_k - F(x_k)] \\ &= F(x_k)^T((F'(x_K) - \gamma_k I)d_k - \|F(x_k)\|^2, \end{aligned} \tag{24}$$

by Chauchy-Schwarz we have,

$$\begin{aligned} f(x_k)^T d_k &\leq \|F(x_k)\|\|((F'(x_k) - \gamma_k I)d_k\| - \|F(x_k)\|^2 \\ &\leq -(1-\epsilon)\|F(x_k)\|^2. \end{aligned} \tag{25}$$

Hence for $\epsilon \in (0,1)$ this lemma is true.
From lemma 1, we can deduce that the norm function $f(x_k)$ is a descent along d_k , which means that $\|F(x_{k+1})\| \leq \|F(x_k)\|$ is true.

Lemma 2. Let assumption 2 hold and $\{x_k\}$ be generated by algorithm 1. Then $\{x_k\} \subset \Omega$.

Proof. By lemma 1, we have $\|F(x_{k+1})\| \leq \|F(x_k)\|$. Moreover, we have for all k.

$$\|F(x_{k+1})\| \leq \|F(x_k)\| \leq \|F(x_{k-1})\| \leq \ldots \leq \|F(x_0)\|.$$

This implies that $\{x_k\} \subset \Omega$.

Lemma 3. Suppose that assumption 1 holds and $\{x_k\}$ be generated by algorithm 1. Then there exists a constant $m > 0$ such that for all k

$$y_k^T s_k \geq m\|s_k\|^2. \tag{26}$$

Proof. By mean-value theorem and (18) we have,
$y_k^T s_k = s_k^T(F(x_{k+1}) - F(x_k)) = s_k^T F'(\zeta) s_k \geq m\|s_k\|^2$,

where $\xi = x_k + \zeta(x_{k+1} - x_k)$, $\zeta \in (0,1)$. The proof is completed. Using $y_k^T s_k \geq m\|s_k\|^2 > 0$, γ_{k+1} is always generated by the update formula (12), and we can deduce that $\gamma_{k+1}I$ inherits the positive definiteness of $\gamma_k I$. By the above lemma and (21), we obtained

$$\frac{y_k^T s_k}{\|s_k\|^2} \geq m, \qquad \frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{M^2}{m}. \tag{27}$$

Lemma 4. Suppose that assumption 1 holds and $\{x_k\}$ is generated by algorithm 1. Then we have

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0, \tag{28}$$

and

$$\lim_{k\to\infty} \|\alpha_k F(x_k)\| = 0. \tag{29}$$

Proof. From (15) we have for all $k > 0$

$$\begin{aligned} \omega_2\|\alpha_k d_k\|^2 &\leq \omega_1\|\alpha_k F(x_k)\|^2 + \omega_2\|\alpha_k d_k\|^2 \\ &\leq \|F(x_k)\|^2 - \|F(x_{k+1})\|^2 + \eta_k\|F(x_k)\|^2. \end{aligned} \tag{30}$$

By summing the above inequality, we have

$$\begin{aligned} \omega_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 &\leq \sum_{i=0}^{k} \left(\|F(x_i)\|^2 - \|F(x_{i+1})\|^2\right) + \sum_{i=0}^{k} \eta_i\|F(x_i)\|^2 \\ &= \|F(x_0)\|^2 - \|F(x_{k+1})\|^2 + \sum_{i=0}^{k} \eta_i\|F(x_i)\|^2 \\ &\leq \|F(x_0)\|^2 + \|F(x_0)\|^2 \sum_{i=0}^{k} \eta_i \\ &\leq \|F(x_0)\|^2 + \|F(x_0)\|^2 \sum_{i=0}^{\infty} \eta_i. \\ &\leq \|F(x_0)\|^2(1+\eta). \end{aligned} \tag{31}$$

From the level set and fact that $\{\eta_k\}$ satisfies (14), the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ is convergent. This implies (28). By similar argument, we can prove that (29) holds.

Lemma 5. Suppose assumption 1 holds and let $\{x_k\}$ be generated by algorithm 1. Then there exist a constant $m_3 > 0$ such that for all $k > 0$,

$$\|d_k\| \leq m_3. \tag{32}$$

Proof. From the level set, (7) and (21) we have,

$$\begin{aligned}\|d_k\| &= \left\| -\frac{y_{k-1}^T s_{k-1} F(x_k)}{\|y_{k-1}\|^2} \right\| \\ &\leq \frac{\|F(x_k)\|\|s_{k-1}\|\|y_{k-1}\|}{m^2\|s_{k-1}\|^2} \\ &\leq \frac{\|F(x_k)\|M\|s_{k-1}\|}{m^2\|s_{k-1}\|} \\ &\leq \frac{\|F(x_k)\|M}{m^2} \\ &\leq \frac{\|F(x_0)\|M}{m^2}.\end{aligned} \tag{33}$$

Taking $m_3 = \frac{\|F(x_0)\|M}{m^2}$, we have (32). We can deduce that for all k (32) hold.

Now we are going to establish the following global convergence theorem to show that under some suitable conditions, there exist an accumulation point of $\{x_k\}$ which is a solution of problem (1).

Theorem 1. Suppose that assumption 1 holds,$\{x_k\}$ is generated by algorithm 1. Assume further for all $k > 0$,

$$\alpha_k \geq c\frac{|F(x_k)^T d_k|}{\|d_k\|^2}, \tag{34}$$

where c is some positive constant. Then

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{35}$$

Proof. From lemma 5 we have (32). Therefore by (28) and the boundedness of $\{\|d_k\|\}$, we have

$$\lim_{k\to\infty} \alpha_k \|d_k\|^2 = 0, \tag{36}$$

from (34) and (36) we have

$$\lim_{k\to\infty} |F(x_k)^T d_k| = 0. \tag{37}$$

On the other hand from (7) we have,

$$F(x_k)^T d_k = -\gamma_k^{-1}\|F(x_k)\|^2, \tag{38}$$

$$\begin{aligned}\|F(x_k)\|^2 &= \| - F(x_k)^T d_k \gamma_k\| \\ &\leq |F(x_k)^T d_k||\gamma_k|.\end{aligned} \tag{39}$$

But

$$\gamma_k = \frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \geq \frac{m\|s_{k-1}\|^2}{y_{k-1}^T s_{k-1}}.$$

So,

$$|\gamma_k| \geq \frac{\|y_{k-1}\|^2}{\|y_{k-1}\|\|s_{k-1}\|} \geq \frac{m\|s_{k-1}\|^2}{\|y_{k-1}\|\|s_{k-1}\|} \geq \frac{m\|s_{k-1}\|}{M\|s_{k-1}\|} \geq \frac{m}{M},$$

from (39) we have,

$$\|F(x_k)\|^2 \leq |F(x_k)^T d_k| \left(\frac{m}{M}\right). \tag{40}$$

As a result,

$$0 \leq \|F(x_k)\|^2 \leq |F(x_k)^T d_k| \left(\frac{m}{M}\right) \longrightarrow 0. \tag{41}$$

Therefore,

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{42}$$

The proof is completed.

4. NUMERICAL RESULTS

In this section, some numerical results are reported in order to show the effectiveness of the proposed method by comparing it with the following existing method in the literature.

- An inexact PRP conjugate gradient method for symmetric nonlinear equations (IPRP) [23].

The codes, were written in Matlab 7.9.0 (R2009b) and run on a computer 2.00 GHz CPU processor and 3 GB RAM memory. However, two algorithms were implemented with the same line search (15) in the experiments, and the following parameters are set: $\omega_1 = \omega_2 = 10^{-4}$, $r = 0.2$ and $\eta_k = \dfrac{1}{(k+1)^2}$. The iteration is set to stop for all the methods if $\|F(x_k)\| \leq 10^{-4}$ or when the iterations exceed 1000 but no point of x_k satisfying the stopping criterion is obtained. We use the symbol '-' to represent failure due to; (i) Memory requirement (ii) Number of iterations exceed 1000. To show the extensive numerical experiments of SSIDD and IPRP methods, we used ten test problems with different initial points and dimension (n values) between 10 to 10,000. problems 1-7 are from [9] and problem 8 was arbitrarily constructed by us, while problems 9 and 10 are from [21].

Problem 1:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (e_1^x - 1, ..., e_n^x - 1)^T.$$

$$x_0 = (0.5, 0.5, ..., 0.5)^T.$$

Problem 2:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (\sin x_1 - 1, ..., \sin x_n - 1)^T.$$

$$x_0 = (1, 1, ..., 1)^T.$$

Problem 3:

$$F_1(x) = x_1(x_1^2 + x_2^2) - 1,$$
$$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2),$$
$$F_n(x) = x_n(x_{n-1}^2 + x_n^2).\ i = 2, 3, ..., n-1.$$
$$x_0 = (0.01, 0.01, ..., 0.01)^T.$$

Problem 4:

$$F_{3i-2}(x) = x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1,$$
$$F_{3i-1}(x) = x_{3i-2}x_{3i-2}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,$$
$$F_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}.\ i = 1, ..., \frac{n}{3}.$$
$$x_0 = (0.4, 0.4, ..., 0.4)^T.$$

Problem 5:

$$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2.$$
$$i = 1, 2, ..., n.$$
$$x_0 = (0.7, 0.7, ..., 0.7)^T.$$

Problem 6:

$$F_1(x) = x_1^2 - 3x_1 + 1 + cos(x_1 - x_2),$$
$$F_i(x) = x_1^2 - 3x_i + 1 + cos(x_i - x_{i-1}).\ i = 1, 2, ..., n.$$
$$x_0 = (0.4, 0.4, ..., 0.4)^T.$$

Problem 7:

$$F_i(x) = x_i - 0.1x_{i+1}^2,$$
$$F_n(x) = x_n - 0.1x_1^2.\ i = 1, 2, ..., n-1.$$
$$x_0 = (1, 1, ..., 1)^T.$$

Problem 8:

$$F_i(x) = 0.i(1 - x_i)^2 - e^{-x_i^2},$$
$$F_n(x) = \frac{n}{10}(1 - e^{-x_n^2}).\ i = 1, 2, ..., n-1.$$
$$x_0 = (-0.1, -0.1, ..., -0.1)^T.$$

Problem 9:

$$F_i(x) = 2x_i - sin|x_i|,\ i = 1, 2, ..., n.$$
$$x_0 = (-0.1, -0.1, ..., -0.1)^T.$$

Problem 10:

$$F_1 = x_1 - e^{cos\left(\frac{x_1+x_2}{n+1}\right)}$$
$$F_i = x_i - e^{cos\left(\frac{x_{i-1}+x_i+x_{i+1}}{n+1}\right)}$$
$$F_n = x_n - e^{cos\left(\frac{x_{n-1}+x_n}{n+1}\right)}\ i = 2, 3, ..., n-1.$$
$$x_0 = (-2, -2, ..., -2)^T.$$

The numerical results of the two methods are shown in Table 1, where, "NI" and "TIME" represent the total number of iterations and the CPU time (in seconds) respectively, and $\|F(x_k)\|$ represents the value of the residual at the stopping point. From Table 1, the numerical results indicated that the proposed method, i.e., SSIDD method has performed better than IPRP method for it has minimum number of iterations and CPU time than IPRP method. Except for problem 9 and 10, where the number of iteration in SSIDD method is less than that of IPRP method. Furthermore,

TABLE 1. The numerical results for SSIDD and IPRP methods on problems 1 to 10

		SSIDD			IPRP		
Problems	Dimension	NI	TIME	$\|F(x_k)\|$	NI	TIME	$\|F(x_k)\|$
1	10	18	0.041624	3.51E-05	31	0.141133	8.18E-05
	100	19	0.075251	8.59E-05	46	0.270288	7.74E-05
	1000	22	0.568489	8.30E-05	56	2.988004	9.44E-05
	2000	23	1.952655	3.00E-05	63	11.296114	9.98E-05
2	10	12	0.068755	2.75E-05	39	0.196596	9.66E-05
	100	13	0.060112	7.00E-05	58	0.372076	7.98E-05
	1000	14	0.390139	4.77E-05	59	3.574642	9.37E-05
	2000	14	1.207898	5.88E-05	67	13.693516	9.92E-05
3	10	21	0.008905	6.66E-05	57	0.040270	9.13E-05
	100	21	0.009361	6.61E-05	49	0.048546	9.62E-05
	1000	22	0.041587	4.40E-05	78	0.121970	8.87E-05
	10000	21	0.175196	7.40E-05	61	0.858538	7.90E-05
4	10	10	0.006528	2.00E-05	24	0.026414	4.19E-05
	100	10	0.006197	6.62E-05	26	0.040968	3.25E-05
	1000	11	0.034254	4.06E-05	28	0.095478	2.39E-05
	10000	12	0.132343	2.46E-05	28	0.523166	7.56E-05
5	10	13	0.010148	1.93E-05	19	0.011114	7.02E-05
	100	13	0.007288	6.10E-05	21	0.016205	6.88E-05
	1000	14	0.026965	3.67E-05	23	0.048532	6.74E-05
	10000	15	0.117319	2.20E-05	25	0.321242	6.60E-05
6	10	7	0.020930	4.29E-05	15	0.012938	2.97E-06
	100	8	0.004707	2.58E-05	15	0.010200	9.38E-06
	1000	8	0.030770	8.15E-05	15	0.054797	2.97E-05
	10000	9	0.102759	4.89E-05	15	0.201566	9.38E-05
7	10	7	0.024790	2.07E-05	7	0.014107	1.01E-05
	100	7	0.022932	5.59E-05	8	0.043108	2.90E-05
	1000	8	0.204210	3.29E-05	9	0.303001	6.16E-06
	10000	9	3.816874	1.97E-05	9	6.864108	5.15E-05
8	10	5	0.001114	7.39E-05	—	—	—
	100	7	0.006096	7.41E-05	—	—	—
	1000	9	0.024428	7.76E-05	—	—	—
	10000	9	0.018475	7.76E-05	—	—	—
9	10	13	0.008806	6.08E-05	7	0.003152	1.04E-07
	100	14	0.003623	6.07E-05	7	0.004767	3.27E-07
	1000	16	0.030029	6.23E-05	7	0.017545	1.04E-06
	10000	17	0.106776	6.22E-05	7	0.101793	3.27E-06
10	10	8	0.001538	9.87E-05	84	0.041959	8.25E-05
	100	8	0.004155	8.74E-05	5	0.00475	7.89E-05
	1000	9	0.025882	4.86E-05	4	0.018542	1.81E-05
	10000	10	0.103763	2.90E-05	4	0.087698	7.26E-07

SSIDD method solved problem 8 where IPRP method fails completely. This is evidently due to contribution of the proposed acceleration parameter that approximated the Jacobian matrix at each iteration.

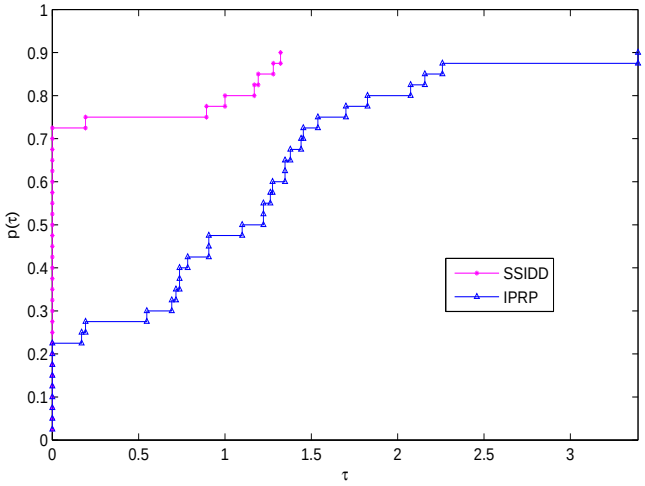

FIGURE 1. Performance profile of SSIDD and IPRP methods with respect to the number of iteration for the problems 1-10.

Figures (1-2) show the performance of our method relative to the number of iterations and CPU time, which were evaluated using the profiles of Dolan and Moré [19]. That is, for each method, we plot the fraction $P(\tau)$ of the problems for which the method is within a factor τ of the best time. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time.

5. CONCLUSION

In this paper, solving system of nonlinear equations using improved double direction approach is presented. This was achieved by approximating the Jacobian matrix via acceleration parameter. The proposed method is completely derivative-free iterative method that is globally convergent under certain appropriate conditions. Numerical comparisons of SSIDD and IPRP methods have been made using a set of large-scale test problems. Furthermore, Table 1 and Figure (1-2), showed that the proposed method is practically quite efficient because it has the least number of iteration compared to IPRP method. In the future research, the double direction scheme will also be modified to solve the monotone nonlinear equations with convex constraints.

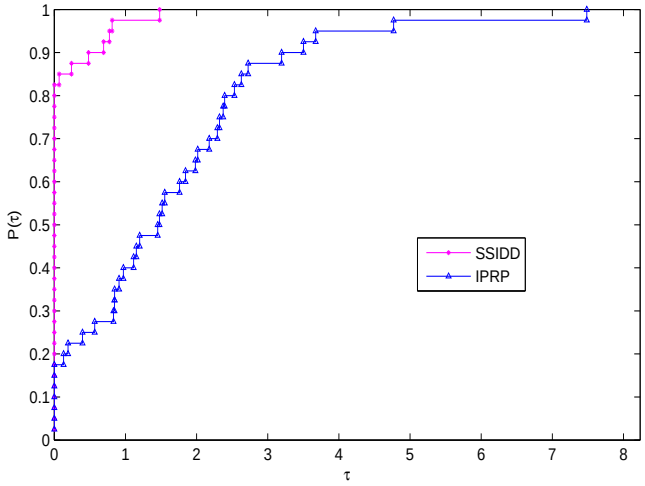

FIGURE 2. Performance profile of SSIDD and IPRP methods with respect to the CPU time (in second) for the problems 1-10.

ACKNOWLEDGEMENTS

The author would like to thank the anonymous referee whose comments improved the original version of this manuscript.

REFERENCES

[1] N. I. Dbaruranovic-milicic and M. Gardasevic-Filipovic, *A multi step curve search algorithm in nonlinear convex case*, Facta universitatis series:mathematics and informatics **25** 11-24, 2010.

[2] N. I Dbaruranovic-milicic,*A multi step curve search algorithm in nonlinear optimization*, Yugoslav Journal of operation research **18** 47-52, 2008.

[3] A. S. Halilu, M. Y., Waziri, and Y. B. Musa, *Inexact double step length method for solving systems of nonlinear equations*, Stat., Optim. Inf. Comput.**8** 165174, 2020.

[4] 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*, Yugoslav Journal of Operations Research. doi.org/10.2298/YJOR160515005H

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

[6] L. Zang, W. Zhou and D. H. Li,*Global convergence of modified Fletcher-Reeves conjugate gradient method with Armijo-type line search*, Numerische mathematik, **164**(1) 277-289, 2005.

[7] 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) 32-44, 2018.

[8] A. S. Halilu and M. Y. Waziri, *An improved derivative-free method via double direction approach for solving systems of nonlinear equations*, Journal of the Ramanujan Mathl. Soc. **33** 75-89, 2018.

[9] M. Y. Waziri and J. Sabiu, *A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equations*, Journal of mathematics and mathematical sciences, 1-8, 2015.

[10] M. Y. Waziri, W. J. Leong, M. A. Hassan and M. Monsi, *A new Newton's method with diagonal Jacobian approximation for system of nonlinear equations*, Journal of mathematics and statistics, **6**(3) 246-252, 2010.

[11] M. Y. Waziri, H. U. Muhammad, A. S. Halilu, K. Ahmed, *Modified matrix-free methods for solving system of nonlinear equations*, Optimization, 1-20. 2020. DOI: 10.1080/02331934.2020.1778689

[12] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal of optimization, **7** 26-33, 1997.

[13] A. S. Halilu, and M. Y. Waziri, *Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations*, Int. Journal app Math Res. **6** 147-156, 2017.

[14] D. Li and M. Fukushima, *A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equation*, SIAM Journal on numerical Analysis, **37** 152-172, 1999.

[15] M. J. petrovic, P. S. Stanimirovic,*Accelerated double direction method for solving unconstrained optimization problems*, Mathematical problem Eng., 2014.

[16] A. S. Halilu and M. Y. Waziri, *A transformed double step length method for solving large-scale systems of nonlinear equations*, Journal of Num. Math. Stoch. **9** 20-32, 2017.

[17] Gongli Yuan*, Xiwen Lu, *A new backtracking inexact BFGS method for symmetric nonlinear equations*, Journal of computers and mathematics with applications, **55** 116-129, 2008.

[18] M. J. petrovic,*An accelerated double step size model in unconstrained optimization*, Journal of mathematics and computation, **250** 309-319, 2015.

[19] E. Dolan and J. Moré. *'Benchmarking opyimization software with performance profiles*, Journal of Mathematical program, **91** 201-213, 2002.

[20] 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*, Journal of Num. Math. Stoch. **9** 80-94, 2017.

[21] Qin-Rong Yana, Xiao-Zhen Penga and Dong-Hui Li,*A globally convergent derivative-free method for solving large-scale nonlinear monotone equations*, Journal of Computational and Applied Mathematics, **234** 649-657,2010.

[22] A. S. Halilu, M. K. Dauda, M. Y. Waziri, and M. Mamat, *Aderivative-free decent method via acceleration parameter for solving systems of nonlinear equations*, Open Journal sci.tech. **2**(3) 14, 2019.

[23] W. Zhou and D. Shen, *An inexact PRP conjugate gradient method for symmetric nonlinear equations*, Numerical Functional Analysis and Optimization, **35** 370388, 2014.

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, SULE LAMIDO UNIVERSITY, KAFIN HAUSA, NIGERIA
E-mail addresses: abubakars.halilu@slu.edu.ng, ashalilu.mth@gmail.com

DEPARTMENT OF MATHEMATICAL SCIENCES, BAYERO UNIVERSITY, KANO, NIGERIA
E-mail addresses: mywaziri.mth@buk.edu.ng, mywaziri@gmail.com