J. Ramanujan Math. Soc. **33,** No.1 (2018) 75–89

An improved derivative-free method via double direction approach for solving systems of nonlinear equations

Abubakar S. Halilu[1] and M. Y. Waziri[2]

[1]*Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Jigawa, Nigeria*
e-mail: abubakarsaddiqu@gmail.com

[2]*Department of Mathematical Sciences, Bayero University Kano, Nigeria*
e-mail: mywaziri.mth@buk.edu.ng

Communicated by: Prof. Sujatha

Received: December 10, 2016

Abstract. We present some double direction method for solving large-scale system of nonlinear equations. This method used the special form of iteration by introducing the two direction vectors in different ways. The approximation to the Jacobian matrix is done by sufficiently constructed diagonal matrix via acceleration parameter as well as derivative-free line search procedure. The proposed method is proved to be globally convergent under mild condition. Finally, numerical comparison using a large scale benchmark test problems show that the proposed method is very promising.

1. Introduction

Consider the system of nonlinear equations:

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map.

The most popular schemes for solving (1) are based on successive linearization [4,11,13], where the search direction d_k is obtained by 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.

It is vital to mention that double direction has been proposed by [2] and the iterative procedure is given as:

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

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.

It is very important to state that the derivative-free double direction methods is severally used in unconstrained optimization problem. They are particularly efficient due to their convergence properties, simple implementation, and low storage requirement [12]. However, the study of derivative-free double direction methods for solving system of nonlinear equations is scanty, and this is what motivated us to write this paper.

We are interested in approximating the Jacobian with diagonal matrix via

$$F'(x_k) \approx \gamma_k I,$$

where I is an identity matrix.

Furthermore (1) can come from an unconstrained optimization problem, a saddle point, and equality constrained problem [11]. Let f be a norm function defined by

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

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

$$\min f(x), \quad x \in \mathbb{R}^n. \tag{5}$$

The double direction method has been proposed in [2], using multi-step iterative information and curve search to generate new iterative points. Nevertheless a multi-step algorithm for minimization of a non differentiable function is also presented in [1]. Recently [12] presented a double direction to solve unconstrained optimization problem.

There are several procedure for the choice of the search direction [1,4–6,8–11].

In steepest descent method the direction d_k is defined by $d_k = -F(x_k)$ [15]. 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

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

where β_k is a CG-parameter. See [5–7].

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 [3,5,7,10,12,13], 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 present the the two directions vectors (3). 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 b_k and c_k in (3) to be defined as:

$$b_k = -\gamma_k^{-1} F(x_k) \tag{6}$$

$$c_k = -F(x_k), \tag{7}$$

where $\gamma_k \in \mathbb{R}$ is an acceleration parameter.

By putting (6) and (7) in to (3) we obtained

$$x_{k+1} = x_k - (\alpha_k + \alpha_k^2 \gamma_k)\gamma_k^{-1} F(x_k), \tag{8}$$

from (8) we can easily show that, our direction is :

$$d_k = -\gamma_k^{-1} F(x_k), \tag{9}$$

then using (8) and (9) we have the general scheme as:

$$x_{k+1} = x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k. \tag{10}$$

The presented method has a norm descent property, whose global convergence will be given under suitable conditions. Numerical result show that the method is very interesting.

We proceed to obtained the acceleration parameter by using Taylor's expansion of the first order and gives the following approximation:

$$F(x_{k+1}) \approx F(x_k) + F'(\xi)(x_{k+1} - x_k) \tag{11}$$

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

$$\xi = x_k + \delta(x_{k+1} - x_k) = x_k - \delta\alpha_k d_k, \quad 0 \leqslant \delta \leqslant 1. \tag{12}$$

putting in mind that the distance between x_k and x_{k+1} is small enough, we can take $\delta = 1$ in (12) and get $\xi = x_{k+1}$. Thus we assume

$$F'(\xi) \approx \gamma_{k+1} I. \tag{13}$$

Now from (11) and (13) one can easily see that:

$$F(x_{k+1}) - F(x_k) = \gamma_{k+1}(x_{k+1} - x_k), \tag{14}$$

where $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$, so

$$y_k = \gamma_{k+1} s_k \tag{15}$$

by multiplying y_k^T to the both side of (15) we obtained the acceleration parameter as:

$$\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}. \tag{16}$$

We use the derivative-free line search proposed in [11] 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

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty \tag{17}$$

$$f(x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k) - f(x_k) \leq -\omega_1 \|\alpha_k F(x_k)\|^2 - \omega_2 \|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{18}$$

Let i_k is the smallest non negative integer i such that (18) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

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

Algorithm 1 (IDFDD).

Step 1: Given x_0, $\gamma_0 = 0.01$, $\epsilon = 10^{-4}$, 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 (18)).
Step 6: Set $x_{k+1} = x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k$.
Step 7: Compute $F(x_{k+1})$.
Step 8: Determine $\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}$.
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 (IDFDD). To begin with, let us defined the level set

$$\Omega = \{x \mid \|F(x)\| \leq \|F(x_0)\|\}. \tag{19}$$

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

$$\|F'(x)\| \leq M \quad \forall x \in N, \tag{20}$$

and

$$m\|d\|^2 \leq d^T F'(x) d \quad \forall x \in N, d \in \mathbb{R}^n. \tag{21}$$

From the level set we have:

$$\|F(x)\| \leq m_1 \quad \forall x \in \Omega. \tag{22}$$

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

$$m\|d\| \leq \|F'(x)d\| \leq M\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{23}$$

$$m\|x - y\| \leq \|F(x) - F(y)\| \leq M\|x - y\| \quad \forall x, y \in N. \tag{24}$$

In particular $\forall x \in N$ we have

$$m\|x - x^*\| \leq \|F(x)\| = \|F(x) - F(x^*)\| \leq M\|x - x^*\|, \tag{25}$$

where x^* stands for the unique solution of (1) in N.

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

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

$$\|(F'(x_k) - \gamma_k I)d_k\| \leq \epsilon\|F(x_k)\| \tag{26}$$

where $\epsilon \in (0, 1)$ is a small quantity [13].

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.*

$$\nabla f(x_k)^T d_k < 0. \tag{27}$$

Proof. from (9), we have

$$\begin{aligned}\nabla f(x_k)^T d_k &= F(x_k)^T F'(x_k) d_k \\ &= F(x_k)^T [(F'(x_k) - \gamma_k I) d_k - F(x_k)] \\ &= F(x_k)^T ((F'(x_K) - \gamma_k I) d_k - \|F(x_k)\|^2, \end{aligned} \tag{28}$$

by chauchy-schwaz we have,

$$\begin{aligned}\nabla f(x_k)^T d_k &\leq \|F(x_k)\| \|((F'(x_k) - \gamma_k I) d_k\| - \|F(x_k)\|^2 \\ &\leq -(1-\epsilon)\|F(x_k)\|^2. \end{aligned} \tag{29}$$

Hence for $\epsilon \in (0, 1)$ this lemma is true.

By the above lemma, 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.

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

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

Lemma 3 (see [13]). *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*

$$y_k^T s_k \geq m\|s_k\|^2. \tag{30}$$

Proof. By mean-value theorem and (21) 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 complete.

Using $y_k^T s_k \geq m_2\|s_k\|^2 > 0$, γ_{k+1} is always generated by the update formula (16), and we can deduce that $\gamma_{k+1} I$ inherits the positive definiteness of $\gamma_k I$. By the above lemma and (24), we obtained

$$\frac{y_k^T s_k}{\|s_k\|} \geq m, \qquad \frac{\|y_k\|}{y_k^T s_k} \leq \frac{M^2}{m} \tag{31}$$

□

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

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0, \tag{32}$$

and

$$\lim_{k\to\infty} \|\alpha_k F(x_k)\| = 0. \tag{33}$$

Proof. By (18) we have for all $k > 0$

$$\begin{aligned}\omega_2\|\alpha_k d_k\|^2 &\leq \omega_1\|\alpha_k F(x_k)\|^2 + \omega_2\|\alpha_k d_k\|^2\\ &\leq \|F(x_k)\|^2 - \|F(x_{k+1})\|^2 + \eta_k\|F(x_k)\|^2.\end{aligned} \tag{34}$$

By summing the above inequality, we have

$$\begin{aligned}\omega_2\sum_{i=0}^{k}\|\alpha_i d_i\|^2 &\leq \sum_{i=0}^{k}(\|F(x_i)\|^2 - \|F(x_{i+1})\|^2) + \sum_{i=0}^{k}\eta_i\|F(x_i)\|^2\\ &= \|F(x_0)\|^2 - \|F(x_{k+1})\|^2 + \sum_{i=0}^{k}\eta_i\|F(x_i)\|^2\\ &\leq \|F(x_0)\|^2 + \|F(x_0)\|^2\sum_{i=0}^{k}\eta_i\\ &\leq \|F(x_0)\|^2 + \|F(x_0)\|^2\sum_{i=0}^{\infty}\eta_i.\end{aligned} \tag{35}$$

So from the level set and fact that $\{\eta_k\}$ satisfies (17) then the series $\sum_{i=0}^{\infty}\|\alpha_i d_i\|^2$ is convergent. This implies (32). By similar arguments as above but with $\omega_1\|\alpha_k F(x_k)\|^2$ on the left hand side, we obtain (33). □

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

$$\|d_k\| \leq m_3. \tag{36}$$

Proof. from (24) we have,

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

$$\leq \frac{\|F(x_k)\|M\|s_{k-1}\|}{m^2\|s_{k-1}\|}$$
$$\leq \frac{\|F(x_k)\|M}{m^2}$$
$$\leq \frac{\|F(x_0)\|M}{m^2}. \tag{37}$$

Taking $m_3 = \frac{\|F(x_0)\|M}{m^2}$, we have (36).

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

Theorem 6. *Suppose that assumption 1 holds,$\{x_k\}$ is generated by algorithm 1. Assume further for all $k > 0$,*

$$\alpha_k \geq c\frac{|F(x_k)^T d_k|}{\|d_k\|^2}, \tag{38}$$

where c is some positive constant. Then

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{39}$$

Proof. From lemma 5 we have (36). Therefore by (32) and the boundedness of $\{\|d_k\|\}$, we have

$$\lim_{k\to\infty} \alpha_k\|d_k\|^2 = 0, \tag{40}$$

from (38) and (40) we have

$$\lim_{k\to\infty} |F(x_k)^T d_k| = 0. \tag{41}$$

on the other hand from (9) we have,

$$F(x_k)^T d_k = -\gamma_k^{-1}\|F(x_k)\|^2, \tag{42}$$
$$\|F(x_k)\|^2 = \| - F(x_k)^T d_k\gamma_k\|$$
$$\leq |F(x_k)^T d_k\|\gamma_k|. \tag{43}$$

But

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

Then,

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

so from (43) we have,

$$\|F(x_k)\|^2 \leq |F(x_k)^T d_k| \left(\frac{m}{M}\right). \tag{44}$$

Thus,

$$0 \leq \|F(x_k)\|^2 \leq |F(x_k)^T d_k| \left(\frac{m}{M}\right) \longrightarrow 0. \tag{45}$$

Therefore,

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{46}$$

The proof is completed. □

4. Numerical results

In this section, the performance of our method for solving the system of nonlinear equations is compared with An inexact PRP conjugate gradient method for symmetric nonlinear equations (IPRP) [18] was reported.

IDFDD stand for our method and we set the following:

$$\omega_1 = \omega_2 = 10^{-4}, r = 0.2 \quad \text{and} \quad \eta_k = \frac{1}{(k+1)^2}.$$

An inexact PRP conjugate gradient method (IPRP) is the method proposed by [18] and we have the following:

$$\omega_1 = \omega_2 = 10^{-4}, \alpha_0 = 0.01, r = 0.2 \quad \text{and} \quad \eta_k = \frac{1}{(k+1)^2}.$$

The codes, was written in Matlab 7.9.0 (R2009b) and run on a computer 2.00 GHz CPU processor and 3 GB RAM memory. We stopped the iteration if the total number of iterations exceeds 1000 or $\|F(x_k)\| \leq 10^{-4}$. We tested the two methods on ten test problems with different initial points and dimension (n values). Problems 1–7 are from [7] and problem 8 was arbitrarily constructed by us, while Problems 9 and 10 are from [17].

Problem 1.

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (e_1^x - 1, \ldots, e_n^x - 1)^T. \quad x_0 = (0.5, 0.5, \ldots, 0.5)^T.$$

Problem 2.

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (\sin x_1 - 1, \ldots, \sin x_n - 1)^T. \quad x_0 = (1, 1, \ldots, 1)^T.$$

Problem 3.

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

Problem 4.

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

Problem 5.

$$\begin{aligned} F_i(x) &= (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2. \\ i &= 1, 2, \ldots, n. \\ x_0 &= (0.7, 0.7, \ldots, 0.7)^T. \end{aligned}$$

Problem 6.

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

Problem 7.

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

Problem 8.

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

Problem 9.

$$
\begin{aligned}
F_i(x) &= 2x_i - \sin|x_i|,\\
i &= 1, 2, \ldots, n.\\
x_0 &= (-0.1, -0.1, \ldots, -0.1)^T.
\end{aligned}
$$

Problem 10.

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

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 [16]. 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.

The numerical results of the two (2) methods are reported in Table 1, where "NI" and "$\|F(x_k)\|$" stand for the total number of all iterations and the norm of the residual at the stopping point respectively. The symbol "—"

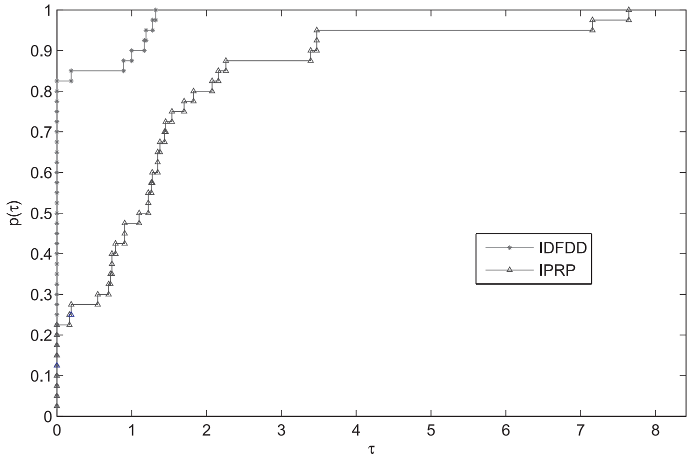

Figure 1. Performance profile of IDFDD and IPRP methods with respect to the number of iteration for the Problems 1–10

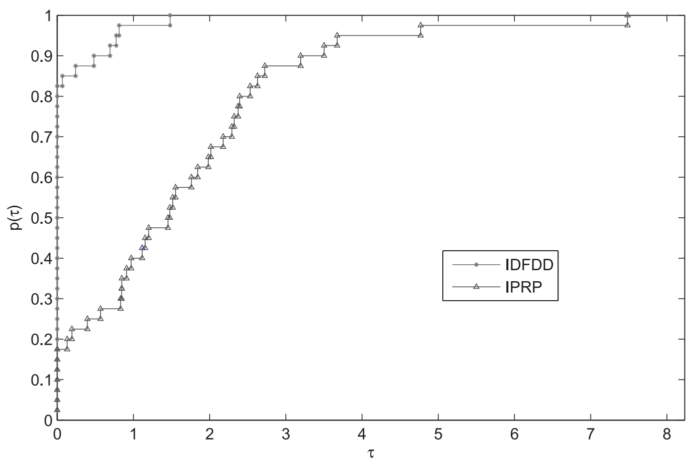

Figure 2. Performance profile of IDFDD and IPRP methods with respect to the CPU time (in second) for the Problems 1–10

indicate a failure at a point. In Table 1, the numerical results indicate that the proposed method, i.e. IDFDD has minimum number of iterations and CPU time, compared to inexact PRP respectively. Except for Problem 9 and 10 where the number of iteration in IDFDD is less than that of inexact IPRP. We can easily see that our claim is fully justified from the tables, that is, less CPU time and number of iterations for each test problem with the exception of Problem 9 and 10. Furthermore, on average, our $\|F(x_k)\|$ is too small which signifies that the solution obtained is the true approximation of the exact solution compared to IPRP.

Table 1. The numerical results for IDFDD and IPRP on Problems 1 to 10

		IDFDD			IPRP		
Problems	Dimension	NI	CPU time	$\|F(x_k)\|$	NI	CPU time	$\|F(x_k)\|$
1	10	13	0.088757	9.03E-05	31	0.141133	8.18E-05
	100	16	0.090698	8.37E-05	46	0.270288	7.74E-05
	1000	18	0.665361	7.55E-05	56	2.988004	9.44E-05
	2000	20	2.369097	2.30E-05	63	11.296114	9.98E-05
2	10	11	0.05542	6.44E-05	39	0.196596	9.66E-05
	100	12	0.077602	8.14E-05	58	0.372076	7.98E-05
	1000	13	0.497451	3.81E-05	59	3.574642	9.37E-05
	2000	13	1.624743	4.01E-05	67	13.693516	9.92E-05
3	10	17	0.006714	7.88E-05	57	0.04027	9.13E-05
	100	16	0.009807	8.27E-05	49	0.048546	9.62E-05
	1000	18	0.039922	8.57E-06	78	0.12197	8.87E-05
	10000	16	0.181285	5.93E-05	61	0.858538	7.90E-05
4	10	7	0.018487	6.51E-05	24	2.64E-02	4.19E-05
	100	8	0.009104	1.67E-05	26	0.040968	3.25E-05
	1000	8	0.030184	5.30E-05	28	0.095478	2.39E-05
	10000	9	0.156268	5.32E-05	28	0.523166	7.56E-05
5	10	6	0.006879	3.08E-06	19	0.011114	7.02E-05
	100	6	0.006364	9.75E-06	21	0.016205	6.88E-05
	1000	6	0.011127	3.08E-05	23	0.048532	6.74E-05
	10000	6	0.096672	9.75E-05	25	0.321242	6.60E-05
6	10	6	0.004391	2.27E-06	15	0.012938	2.97E-06
	100	6	0.007188	7.17E-06	15	0.0102	9.38E-06
	1000	6	0.028501	2.27E-05	15	0.054797	2.97E-05
	10000	6	0.141485	7.17E-05	15	0.201566	9.38E-05
7	10	5	0.010662	1.63E-06	7	0.014107	1.01E-05
	100	5	0.004181	6.20E-06	8	0.043108	2.90E-05
	1000	5	0.012702	2.00E-05	9	0.303001	6.16E-06
	10000	5	0.067249	6.33E-05	9	6.864108	5.15E-05
8	10	6	0.01237	6.14E-05	—	—	—
	100	8	0.006167	8.89E-05	—	—	—
	1000	11	0.033589	4.86E-05	—	—	—
	10000	13	0.172049	6.86E-05	—	—	—
9	10	5	0.00999	7.03E-05	7	0.003152	1.04E-07
	100	6	0.004478	8.79E-05	7	0.004767	3.27E-07
	1000	7	0.019501	8.34E-06	7	0.017545	1.04E-06
	10000	7	0.088119	2.64E-05	7	0.101793	3.27E-06
10	10	8	0.004317	1.83E-05	84	0.041959	8.25E-05
	100	5	0.006049	3.74E-05	5	0.00475	7.89E-05
	1000	5	0.028385	2.01E-06	4	0.018542	1.81E-05
	10000	5	0.123436	4.32E-06	4	0.087698	7.26E-07

5. Conclusion

In this paper we present an improved derivative-free via double direction approach (IDFDD) method for solving system of nonlinear equations and compare its performance with that of inexact PRP (IPRP) method for symmetric nonlinear equations [18] by doing some numerical experiments. We however proved the global convergence of our proposed method by using a backtracking type line search, and the numerical results show that our method is very efficient.

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** (2010) 11–24.

[2] N. I. Dbaruranovic-milicic, A multi step curve search algorithm in nonlinear optimization, *Yugoslav Journal of Operation Research*, **18** (2008) 47–52.

[3] N. Andrei, An accelerated gradient descent algorithm with backtracking for unconstrained optimization, *Numerical Algorithm*, **42** (2006) 63–73.

[4] J. E. Dennis, Jr. and R. B. Schnabel, Numerical method for unconstrained optimization and non-linear equations, Practice Hall, Englewood cliffs, NJ, USA (1983).

[5] 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)** (2005) 277–289.

[6] L. Zang, W. Zhou and D. H. Li, A descent modified Polak-Pibiere-Polyak conjugate gradient method and its global convergence, *IMA Journal of Numerical Analysis*, **164(1)** (2006) 227–572.

[7] M. Y. Waziri and Jamilu Sabiu, A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equations, *Journal of Mathematics and Mathematical Sciences*, **8** (2015).

[8] 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)**, (2010) 246–252.

[9] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal of Optimization*, **7** (1997) 26–33.

[10] M. Raydan, On Barzilai and Borwein choice of step length for the gradient method, *IMA Journal of Numerical Analysis*, **13** (1993) 321–326.

[11] 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** (1999) 152–172.

[12] M. J. Petrovic and P. S. Stanimirovic, Accelerated double direction method for solving unconstrained optimization problems, *Mathematical Problem Eng.* (2014).

[13] Gongli Yuan* and Xiwen Lu, A new backtracking inexact BFGS method for symmetric nonlinear equations, *Journal of Computers and Mathematics with Applications*, **55** (2008) 116–129.

[14] P. S. Stanimirovic and M. B. Miladinovic, Accelerated gradient descent methods with line search, *Journal of Numerical Algorithm*, **13** (1993) 321–326.

[15] M. J. Petrovic, An accelerated double step size model in unconstrained optimization, *Journal of Mathematics and Computation*, **250** (2015) 309–319.

[16] E. Dolan and J. More, Benchmarking opyimization software with performance profiles, *Journal of Mathematical Program*, **91** (2002) 201–213.

[17] 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** (2010) 649–657.

[18] W. Zhou and D. Shen, An inexact PRP conjugate gradientmethod for symmetric nonlinear equations, *Numerical Functional Analysis and Optimization*, **35** (2014) 370–388.