Available online at http://scik.org
J. Math. Comput. Sci. 7 (2017), No. 3, 606-624
ISSN: 1927-5307

A DOUBLE DIRECTION CONJUGATE GRADIENT METHOD FOR SOLVING LARGE-SCALE SYSTEM OF NONLINEAR EQUATIONS

H. ABDULLAH[1,*], M. Y. WAZIRI[2], AND M. K. DAUDA[3]

[1]Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria

[2]Department of Mathematical Sciences, Bayero University, Kano, Nigeria

[3]Department of Mathematics and Statistics, Kaduna Polytechnic, Kaduna, Nigeria

Abstract. This paper presents a method for solving nonlinear system of equations via double direction approach. We consider the first direction to be steepest descent direction while the other direction is the proposed CG direction. Derivative-free line search is used to obtain the step length α_k. The global convergence of the proposed algorithm is established under suitable conditions. Numerical results show that the proposed method is efficient for large scale problems.

Keywords: derivative-free line search; double direction; nonlinear equations; conjugate gradient method.

2010 AMS Subject Classification: 65H10, 65K05.

1. Introduction

Consider the following nonlinear system of equations:

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

[*]Corresponding author

E-mail address: habibtaura32@gmail.com

Received September 1, 2016; Published May 1, 2017

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear, continuous mapping and is assumed to satisfy the following assumptions:

Assumption 1

(A1) There exists $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.

(A2) F is continuously differentiable mapping.

(A3) The Jacobian, $F^{'}(x_k)$ is symmetric.

Solving nonlinear system of equations is very important part in Mathematics and has a wide range of applications in various aspect of applied sciences, technology and industry. Many examples from all of these branches have been considered in recent years [1, 2, 3].

One of the most attractive factor of each numerical algorithm for solving system of nonlinear equations is how the procedure deals with large scale problem. The effectiveness or otherwise of the methods depends solely on step length, α_k, and search direction d_k. There are several procedure for the choice of the search direction (see[4, 5, 33]). Likewise, α_k can be computed either exact or inexact. The most frequently used line search in practice is inexact line search [5, 10, 24] which is chosen in such a way that the function values along the ray $x_k + \alpha_k d_k$, $\alpha_k > 0$ decreases ie.,

$$||F(x_k + \alpha_k d_k)|| < ||F(x_k)||. \tag{2}$$

In this work, we adopt a derivative-free line search described in [6] which is based on Li and Fukushima[8] to obtained the optimal step length. Large number of efficient solver for large scale symmetric nonlinear system of equations have been proposed in the last decades, the most popular ones are due to Li and Fukushima [8] in which a Gauss-Newton based BFGS method is developed, the global and superlinear convergence were established. It's performance is improved by Gu et.al [9], were a norm decent BFGS method is designed. Since then, norm descent type BFGS method especially with trust region approaches are presented in the literature and have shown the efficiency experimentally [11, 12]. However, all these methods require matrix storage location, solving *nxn* linear system and hence not suitable for large scale system.

The emergence of conjugate gradient method for solving symmetric nonlinear system of equations is a welcome development. Nonlinear conjugate gradient came into existence in the year 1964 [13], since then the work on CG became prominent in the literature. Different CG parameter β_k, corresponds to different CG direction. We refer to survey paper [13] for a summary of the derivative free Quasi-Newton methods for solving nonlinear system of equations. In 2006, [14] presented a CG method for solving unconstrained optimization problems which was modified in 2011 see [15] to solve symmetric nonlinear system of equations. Further-more, in 2015, [7] presented a CG method for solving symmetric system of nonlinear equations without computing Jacobian via special structure of the underlying function.

It's important to mention here that double direction iteration for unconstrained optimization has been presented in the literature by many authors [16, 17, 18] and has the iterative procedure given by:

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

where x_{k+1} represents a new iterative point, x_k is the previous iterative point, α_k is the step length, while c_k and d_k are the search directions respectively.

However, double direction methods for solving system of nonlinear equations are very scanty, that is what motivated us to have this paper. We assume the first direction ie., c_k, to be the steepest descent direction and the other direction ie, d_k to be a hybrid CG direction.

Problem (1) can be converted to the following global optimization problem

$$min f(x), \quad x \in \mathbb{R}^n, \tag{4}$$

with function f defined by

$$f(x) = \frac{1}{2}||F(x)||^2, \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}$

Motivated by [18], we propose our scheme for solving system of nonlinear equations with two directions vectors. We organized the rest of the paper as follows. In the next section, we present the proposed method. Section 3 presents convergence results. Numerical experiments are presented in section 4, and finally conclusion is given in section 5.

2. Derivation of the Method

In this section we present a new CG parameter β_k. This made possible by combining the direction presented in [19] with classical Newton direction. However, the direction proposed in [19] is defined as:

$$d_k = \begin{cases} -F(x_k) & if \quad k = 0 \\ -F(x_k) + \beta_k^{PRP} d_{k-1} - v_k y_k & if \quad k \geq 1 \end{cases} \tag{6}$$

where $\beta_k^{PRP} = \frac{F^T(x_k)y(x_{k-1})}{||F(x_{k-1})||^2}$, $v_k = \frac{F^T(x_k)d_{k-1}}{||F(x_{k-1})||^2}$, $y_k = F(x_k) - F(x_{k-1})$.

The Newton's direction, d_k, given by:

$$d_k = -J(x_k)^{-1}F(x_k), \tag{7}$$

where $J(x_k)$ is the Jacobian matrix of $F(x_k)$. By combining (6) and (7), we have the following expression for d_k:

$$-J^{-1}(x_k)F(x_k) = -F(x_k) + \beta_k d_{k-1} - v_k y_k, \tag{8}$$

multiplying both sides of (8) by $J(x_k)$ leads to

$$-J(x_k)J^{-1}(x_k)F(x_k) = -J(x_k)F(x_k) + J(x_k)\beta_k d_{k-1} - v_k J(x_k) y_k, \tag{9}$$

$$-F(x_k) = -J(x_k)F(x_k) + J(x_k)\beta_k d_{k-1} - v_k J(x_k) y_k. \tag{10}$$

To ensure good approximation, we multiply both sides of (10) by s_k^T to obtain:

$$-s_k^T F(x_k) = -s_k^T J(x_k)F(x_k) + s_k^T J(x_k)\beta_k d_{k-1} - v_k s_k^T J(x_k) y_k. \tag{11}$$

From the secant condition we have,

$$J(x_k)s_k = y_k, \tag{12}$$

$$s_k^T J^T(x_k) = y_k^T, \tag{13}$$

$$s_k^T J(x_k) = y_k^T \tag{14}$$

by assuming that $J(x_k)$ is symmetric. Substituting (14) into (11), we obtain

$$-s_k^T F(x_k) = -y_k^T F(x_k) + y_k^T \beta_k d_{k-1} - v_k y_k^T y_k. \tag{15}$$

After little linear algebra, we presents the new CG parameter as:

$$\beta_k^* = \frac{(y_k - s_k)^T F(x_k) + v_k ||y_k||^2}{y^T d_{k-1}}. \tag{16}$$

Having derived the CG parameter β_k^*, in (16) then using (6), we presents our proposed direction d_k, as

$$d_k = \begin{cases} -F(x_k) & if \quad k = 0, \\ -F(x_k) + \beta_k^* d_{k-1} - v_k y_k & if \quad k \geq 1, \end{cases} \tag{17}$$

where β_k^* is defined in (16).

We use a derivative-free line search described in [6] to compute the step length $\alpha_k > 0$

Let ω_1, $\omega_2 > 0$, $r \in (0,1)$ be a constant and let η_k be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty, \tag{18}$$

and

$$\|F(x_k - \alpha_k F(_k) + \alpha_k^2 d_k)\|^2 - \|F(x_k)\|^2 \leq -\omega_1 \|\alpha_k F(x_k)\|^2 - \omega_2 \|\alpha_k d_k\|^2 + \eta_k. \tag{19}$$

Let i_k be the smallest nonnegative integer i such that (19) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

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

Algorithm 1(DDLS)

STEP 1: Given x_o, $\varepsilon = 10^{-4}$, set $d_0 = -F(x_0)$ and $k = 0$.

STEP 2: Compute $F(x_k)$.

STEP 3: If $\|F(x_k)\| \leq \varepsilon$, then stop, else go to STEP 4.

STEP 4: Compute step length α_k (by (19)).

STEP 5: Set $x_{k+1} = x_k - \alpha_k F(_k) + \alpha_k^2 d_k$.

STEP 6: Compute $F(x_{k+1})$.

STEP 7: Compute β_k^* (using (16)).

STEP 8: Update d_{k+1} (using (6)).

STEP 9: Set $k = k+1$, and go to STEP 3.

3. Convergence Analysis

In this section we present the global convergence of our proposed method (DDLS). To begin with, let Ω be the level set define by

$$\Omega = \{x | \|F(x)\| \leq \sqrt{\|F(x_0)\|^2 + \eta}, \text{ where,} \tag{20}$$

η is a positive constant such that (18) is satisfied. Here, we can see that the level set Ω is bounded. In order to analyze the global convergence of (DDLS) algorithm, we need the following assumptions:

Assumption 2

(i)In some neighborhood N of Ω the nonlinear function F is Lipschitz continuous ie., there exists a positive constant $L > 0$, such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \tag{21}$$

for all $x, y \in N$. From the level set, there exists a positive constant $M_1 > 0$, such that

$$\|F(x)\| \leq M_1, \tag{22}$$

for all $x \in \Omega$

Lemma 3.1: Let $\{x_n\}$ be a sequence generated by (DDLS) algorithm. Then $\{x_n\} \subset \Omega$.

proof: From (19) we have for all k,

$$\|F(x_k - \alpha_k F(_k) + \alpha_k^2 d_k)\|^2 \leq \|F(x_k)\|^2 + \eta_k$$

$$\vdots$$

$$\leq \|F(x_0)\| + \sum_{i=0}^{k} \eta_i \leq \eta < \infty$$

Thus we have,

$$\|F(x_{k+1})\| \leq \sqrt{\|F(x_0)\|^2 + \eta}. \tag{23}$$

Then we can see that from (23)

$\{x_n\} \subset \Omega$.

Lemma3.2: Suppose that the above assumption holds and $\{x_k\}$ is generated by DDLS algorithm, then we have

$$\lim_{k\to\infty} \|\alpha_k d_k\|^2 = 0 \tag{24}$$

and

$$\lim_{k\to\infty} \|\alpha_k F(x_k)\|^2 = 0 \tag{25}$$

proof: By (19) we have for all $k > 0$

$$\begin{aligned} \omega_2\|\alpha_k d_k\|^2 &\leq \omega_1\|\alpha_K F(x_k)\|^2 + \omega_2\|\alpha_k d_k\|^2 \\ &\leq \|F(x_k)\|^2 - \|F(x_{k+1})\|^2\| + \eta_k \end{aligned} \tag{26}$$

by summing the above k inequality, we obtain

$$\begin{aligned}\omega_2\sum_{i=0}^{k}\|\alpha_i d_i\|^2 &\le \sum_{i=0}^{k}\left(\|F(x_i)\|^2-\|F(x_{i+1})\|^2\right)+\sum_{i=0}^{k}\eta_i\\ &\le \|F(x_0)\|^2-\|F(x_{k+1})\|^2+\sum_{i=0}^{k}\eta_i \qquad (27)\\ &\le \|F(x_0)\|^2+\sum_{i=0}^{k}\eta_i\le \|F(x_0)\|^2+\sum_{i=0}^{\infty}\eta.\end{aligned}$$

So from (22) and fact that $\{\eta_k\}$ satisfies (18) the series $\sum_{i=0}^{\infty}\|\alpha_i d_i\|^2$ is convergent. This implies (24).

By similar way we can prove (24) holds. The following lemma shows that the search direction d_k is bounded when the current point x_k is bounded away from solution (1)

Lemma 3.3; Suppose that assumption 2 holds, and let $\{x_k\}$ is generated by DDLS algorithm, suppose there is a constant $\varepsilon > 0$ such that for all k,

$$\|F(x_k)\| \ge \varepsilon \qquad (28)$$

then there exist a constant $M > 0$ such that for all k,

$$\|d_k\| \le M. \qquad (29)$$

Proof: using (21), (24), and (25) we have

$$\begin{aligned}\|y_k\| = \|F(x_k)-F(x_{k-1})\| \le L\|x_{k+1}-x_k\| &= L\|\alpha_k^2 d_k-\alpha_k F(x_k)\| \qquad (30)\\ &\le L(\|\alpha_k^2 d_k\|+\|\alpha_k F(x_k)\|)\to 0.\end{aligned}$$

And furthermore,

$$\|(y_k-s_k)^T F(x_k)\| = \|y_k-s_k\|\|F(x_k)\| \le M_1(\|(y_k\|+\|s_k\|), \qquad (31)$$

which goes to zero by (22) and (30).

Hence

$$|\beta_k^*| \le \frac{\|y_k-s_k\|\|F(x_k)\|+|v_k|\|y_k\|^2}{|y^T d_{k-1}|}\to 0, \qquad (32)$$

by (30) and (31).

This implies that there exist a constant $\rho \in (0,1)$ such that for sufficiently large k,

$$|\beta_k^*| \leq \rho. \tag{33}$$

By using

$$\|d_k\| \leq \|F(x_k)\| + |\beta_k^*|\|d_{k-1}\| - |v_k|\|y_k\|, \tag{34}$$

and setting

$$M_3 = \max\{\|d_1\|, \|d_2\|, ..., \|d(k_0)\|, \frac{M_1}{1-\varepsilon_0}\}, \tag{35}$$

we can deduce that for all k, (29) holds, ie., $\|d_k\|$ is uniformly bounded.

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 the solution of the problem (1).

Theorem 3.4: Suppose that assumption 1 holds, $\{x_k\}$ is generated by DDLS algorithm. Assume further that for all $k > 0$,

$$\alpha_k \geq c \frac{|F^T(x_k)d_k|}{\|d_k\|^2}, \tag{36}$$

where c is some positive constant. Then

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

Proof: Suppose that the condition does not hold. Then there exists a constant $\varepsilon > 0$ such that for all k (28) holds. Moreover, from lemma 3.3, we have (29) holds. Therefore by (24) and the boundedness of $\{\|d_k\|\}$,

we have

$$\lim_{k\to\infty} \alpha_k \|d_k\|^2 = 0, \tag{38}$$

which combine with (36) to yields

(39) $$\lim_{k\to\infty} |F_k^T d_k| = 0.$$

On the other hand from (17), we have
$F^T(x_k)d_k = -\|F(x_k)\| + \beta_k^* F^T(x_k)d_{k-1} - F^T(x_k)v_k y_{k-1}$
which can be written as

(40) $$\|F(x_k)\| \leq |F^T(x_k)d_k| + |\beta_k^*| \|F(x_k)\| \|d_{k-1}\| + \|F(x_k)\| \|v_k\| \|y_{k-1}\|$$

So that by equation (22), (29), (30), (33) and taking the limit of the above inequality, we have

(41) $$\lim_{k\to\infty} \|F(x_k)\| = 0,$$

which contradicts equation (28) and hence the proof is completed

4. Numerical Results

In this section, we compare the performance of our method ie., Double Direction Method for solving large system of nonlinear equations (DDLS) with that of Inexact PRP conjugate gradient method for solving symmetric nonlinear equations (IPRP) [20].
,

Throughout this paper ;
DDLS stands for our method (Double Direction Method for solving large system of nonlinear equations) and IPRP stands for Inexact PRP conjugate gradient method for solving symmetric nonlinear equations. And we set the following parameters for DDLS and IPRP respectively:

$\omega_1 = \omega_2 = 10^{-4}$, $r = 0.3$ and $\eta_k = \frac{1}{(k+1)^3}$.
$\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$, $r = 0.3$ and $\eta_k = \frac{1}{(k+1)^3}$.

The codes were written in Matlab (R2013a) and run on a personal computer 2.10 GHZ CPU processor and 2.00 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 from different sources with dimension between 1000 and 100,000 with different initial points which is not restricted to a point that is too close to the solution as suggested by Hillstrom [29]. We believe that this approach, will add to the complexity of the computer programming, which would lead to high CPU time. These initial points will also allow us to test the global convergence properties and the robustness of our method at the same time.

Further more, in table 2 we also report the behavior of the DDLS algorithm for problems 1, 2 and 5 with some different initial points, to illustrate the global convergence. For these problems, the solution vector is $x^* = (1,...,1)^T$. The chosen initial points are $x^0 = (-9\times 10^{-7},...,-9\times 10^{-7})^T$, $x^1 = (0,...,0)^T$, $x^2 = -x^0$ and $x^3 = (10,...,10)^T$ which are wider enough to test the global convergence.

Problem 1 [23] :

$$F_i(x) = x_i x_{i+1} - 1,$$

$$F_i(x) = x_n x_i - 1,$$

$$i = 1,2,...,n.$$

$$x_0 = (0.1, 0.1, ..., 0.1)^T.$$

Problem 2 [24] :

$$F_i(x) = x_i^2 - 1,$$

$$i = 2,3,...,n,$$

$$x_0 = (-0.1, -0.1, ..., -0.1)^T.$$

Problem 3 [23] :

$$F_i(x) = \cos(x_i - 1) + x_i - 1,$$

$$i = 2,3,...,n,$$

$$x_0 = (5,5,...,5)^T.$$

Problem 4 [25] :

$$F_i(x) = x(_i)^2 - cos(x_i - 1),$$

$$i = 1, 2, ..., n,$$

$$x_0 = (10, 10, ..., 10)^T.$$

Problem 5 [25] :

$$F_i(x) = cos(x_i^2 - 1) - 1,$$

$$i = 1, 2, ..., n,$$

$$x_0 = (-0.001, -0.001, ..., -0.001)^T.$$

Problem 6 [25] :

$$F_i(x) = (\sin(x_i)\cos(x_i))^2 x_i - (\cos(x_i) - x_i - 1)x_i,$$

$$i = 1, 2, ..., n,$$

$$x_0 = (5, 5, ..., 5)^T.$$

Problem 7 [25] :

$$F_i(x) = cos(x_i) - 1,$$

$$i = 1, 2, ..., n,$$

$$x_0 = (-1.5, -1.5, ..., -1.5)^T.$$

Problem 8 [26] :

$$F_i(x) = \sin((x_i)^2 \sin(x_i)) - (x_i)^4 + \sin((x_i)^2),$$

$$i = 1, 2, ..., n,$$

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

Problem 9 [27] :

$$F_i(x) = \exp(x_i^2 - 1) - \cos(1 - x_i),$$

$$i = 1, 2, ..., n,$$

$$x_0 = (2.5, 2.5, ..., 2.5)^T.$$

Problem 10 [26]:

$$F_1(x) = (\sin(x_1 - x_2) - 4\exp(2 - x_2) + 2x_1,$$

$$F_i(x) = \sin(2 - x_i) - 4\exp(x_i - 2) + 2x_i + \cos(2 - x_i) - \exp(2 - x_i),$$

$$i = 1, 2, ..., n,$$

$$x_0 = (-0.55, -0.55, ..., -0.55)^T.$$

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 More [22]. 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 tables 1, where "NI" and "Time" stand for the total number of iterations and the CPU time in seconds, respectively, while $\|F(x_k)\|$ is the norm of the residual at the stopping point. We claim that the method fails, and use the symbol "-" when some of the following hold :

(a) the number of iterations is greater than or equal to 1000; or

(b) the number of backtracking at some line search is greater than or equal to 20.

From tables 1, we can easily see that all the two methods attempted to solve the large scale system of nonlinear equations. In particular, the DDSL method considerably out performs the IPRP method because it solved all the tested problems while the IPRP method fails to solve some problems(ie., problems 3,5,6,7). In addition, DDLS method has the least number of iterations as well as the CPU time as both figure(1-2) and table 1 indicated, this is due to the contribution of the added direction in each iteration which help in better approximation at the iterate point.

TABLE 1. Numerical result for DDLS and IPRP methods Problems 1-10

		DDLS			IPRP		
Problems	Dimensions	NI	Time	$\|F(x_k)\|$	NI	Time	$\|F(x_k)\|$
1	1000	16	0.05585	6.34E-05	19	0.01125	3.10E-04
	10000	17	0.10615	8.61E-05	19	0.09086	9.81E-04
	100000	19	1.04116	5.03E-05	21	1.57928	6.20E-04
2	1000	13	0.03558	5.30E-05	15	0.00689	6.49E-04
	10000	11	0.03536	8.08E-05	17	0.06407	4.11E-04
	100000	13	0.46454	4.72E-05	19	0.91622	2.60E-04
3	1000	33	0.07841	7.75E-05	-	-	-
	10000	34	0.15752	8.60E-05	-	-	-
	100000	36	2.04949	8.00E-05	-	-	-
4	1000	11	0.0078	7.33E-05	15	0.00978	2.69E-04
	10000	12	0.0612	9.02E-05	15	0.08671	8.51E-04
	100000	14	0.82151	5.40E-05	17	1.10984	9.62E-05
5	1000	81	0.03345	9.85E-05	512	0.23933	9.98E-04
	10000	93	0.29034	9.31E-05	-	-	-
	100000	104	4.08782	9.79E-05	-	-	-
6	1000	87	0.15636	9.86E-05	-	-	-
	10000	99	0.89807	9.37E-05	-	-	-
	100000	110	11.6707	9.87E-05	-	-	-
7	1000	87	0.08012	9.71E-05	-	-	-
	10000	99	0.30255	9.25E-05	-	-	-
	100000	110	4.32452	9.77E-05	-	-	-
8	1000	25	0.10782	7.13E-05	53	0.10721	9.74E-04
	10000	27	0.39058	9.43E-05	59	0.83149	9.07E-04
	100000	30	4.50478	8.07E-05	65	9.55386	8.44E-04
9	1000	14	0.01109	9.03E-05	13	0.02364	2.37E-04
	10000	16	0.10521	5.27E-05	13	0.17952	7.48E-04
	100000	17	1.25541	7.15E-05	15	2.1842	4.73E-04
10	1000	23	0.02933	6.14E-05	22	0.02688	8.54E-04
	10000	23	0.26057	8.03E-05	19	0.20725	9.34E-04
	100000	24	3.06528	7.22E-05	18	2.20971	9.29E-04

TABLE 2. DDLS Algorithm for problems 1, 2 and 5 for different initial points

			DDLS		
Problems	Dimensions	Initial point	NI	Time	$\|F(x_k)\|$
1	1000	x^0	16	0.006913	5.69E-05
	10000	x^0	17	0.059945	7.73E-05
	100000	x^0	19	0.9285	4.51E-05
1	1000	x^1	16	0.006795	5.69E-05
	10000	x^1	17	0.064209	7.73E-05
	100000	x^1	19	0.904196	4.51E-05
1	1000	x^2	16	0.006763	5.69E-05
	10000	x^2	17	0.059095	7.73E-05
	100000	x^2	19	0.932831	4.51E-05
1	1000	x^3	13	0.006344	8.32E-05
	10000	x^3	15	0.069158	4.85E-05
	100000	x^3	16	0.839786	6.59E-05
2	1000	x^0	16	0.005609	5.69E-05
	10000	x^0	17	0.047292	7.73E-05
	100000	x^0	19	0.645581	4.51E-05
2	1000	x^1	16	0.005246	5.69E-05
	10000	x^1	17	0.041803	7.73E-05
	100000	x^1	19	0.65083	4.51E-05
2	1000	x^2	16	0.005231	5.69E-05
	10000	x^2	17	0.048022	7.73E-05
	100000	x^2	19	0.633379	4.51E-05
2	1000	x^3	13	0.004807	8.32E-05
	10000	x^3	15	0.04832	4.85E-05
	100000	x^3	16	0.696073	6.59E-05
5	1000	x^0	81	0.035934	9.87E-05
	10000	x^0	93	0.29387	9.33E-05
	100000	x^0	104	4.109458	9.82E-05
5	1000	x^1	81	0.032387	9.88E-05
	10000	x^1	93	0.296111	9.33E-05
	100000	x^1	104	4.125491	9.82E-05
5	1000	x^2	81	0.032562	9.88E-05
	10000	x^2	93	0.287072	9.33E-05
	100000	x^2	104	4.162607	9.82E-05
5	1000	x^3	38	0.019406	9.91E-05
	10000	x^3	49	0.223457	9.09E-05
	100000	x^3	59	2.755678	9.88E-05

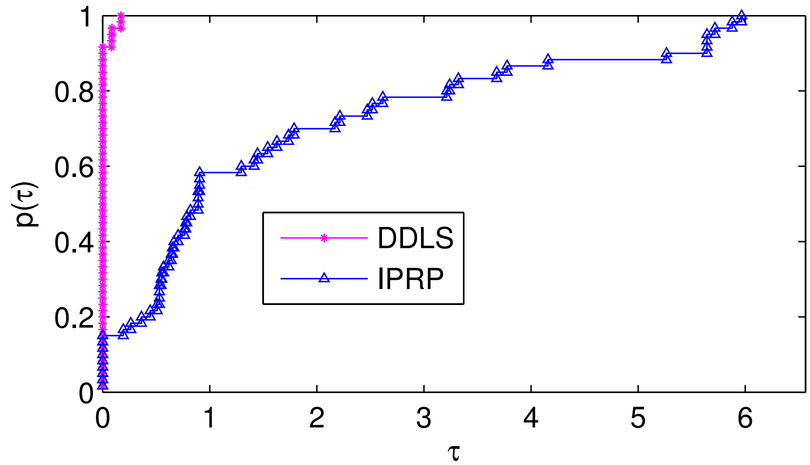

FIGURE 1. Performance profile of DDLS and IPRP methods with respect to the number of iteration for the problems 1-10

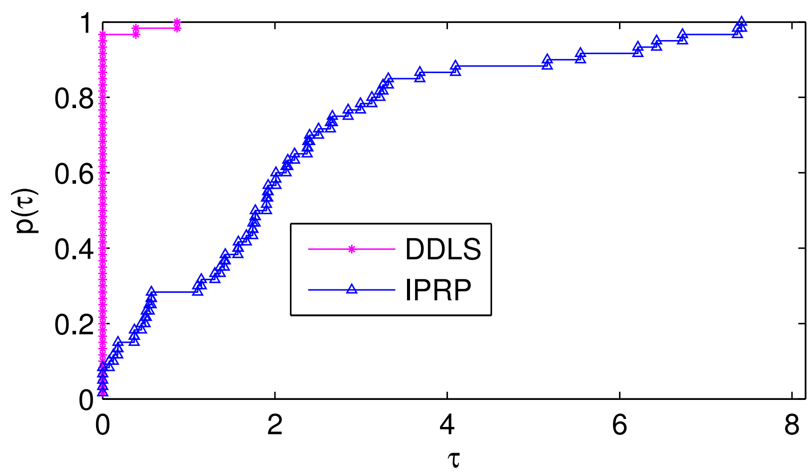

FIGURE 2. Performance profile of DDLS and IPRP methods with respect to the CPU time (in second) for the problems 1-10

5. Conclusion

In this paper we present a double direction iterative scheme for solving large-scale system of nonlinear equations and compare its performance with that of Inexact PRP (IPRP) method for symmetric nonlinear equations [20]. We observe, from Table 1, that the DDLS algorithm is a robust option to solve large-scale system nonlinear system of equations. We also observe from Table 2 the global behavior of the DDLS algorithm for a typical problems, although it requires more iterations when the initial guess is further a way from the solution. In addition, we proved the global convergence of our proposed method using a non derivative-free type line search described in [6]. We choose initial points far away from the solution to see the robustness and global convergence of our method. The numerical result shows that double direction iterations has significant influence towards the convergence of system of nonlinear equations especially large scale system.

Conflict of Interests

The authors declare that there is no conflict of interests.

REFERENCES

[1] M. Ahookhosh, K. Amini, Two derivative-free projection approaches for system of large scale nonlinear monotone equation, Numer. Algor. 64 (2013), 21-42.

[2] Bouaricha A., Schnabel, R.B. Tensor methods for large sparse systems of nonlinear equations, Math. Program. 82 (1998), 377-400.

[3] Xiao, Y., Wu, Chunjie, W, Soon-Yi, Norm descent conjugate gradient methods for solving symmetric nomlinear equations, J. Glob. Optim. 62(2015),751-762.

[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] G. YU, A derivative-free method for solving large-scale nonlinear systems of equations, J. Ind. Manag. Optim. 6 (1) (2010), 149-160.

[7] M.Y Waziri and Jamilu Sabiu, A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equations, J. Math. Math. Sci. 2015 (2015), Article ID 961487, 8 pages.

[8] D. H. 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.

[9] G.Z. GU, D.-H. Li, L. Qi and S.Z. Zhou, Descent directions of quasi-Newton methods for symmetric nonlinear equations, SIAM J. Numer. Anal. 40 (5) (2002), 1763-1774.

[10] M. Raydan, On Barzilai and Borwein choice of step length for the gradient method, IMA J. Numer. Anal. 13 (1993), 321-326.

[11] G.Yuan and X. Lu, A new backtracking inexact BFGS method for symmetric nonlinear equations, Comput. Math. Appl., 55 (2008), 11-129.

[12] G.Yuan and X. Lu and Z.Wei, BFGS trust-region method for symmetric nonlinear equations, J. Comput. Appl. Math., 230 (2009), 44-58.

[13] W.W.Hager and Zhang, A survey of nonlinear conjugate gradient methods, Pac. J. Optim. 2 (1) (2006), 35-58.

[14] Li Zhang, Weijun Zhou, and Dong-Hui Li, A descent modified Polak-Ribiere-Polyak conjugate gradient method and its global convergence, IMA J. Numer. Anal., 26 (4) (2006), 629-640.

[15] Li D.H., Wang, X.L., A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations, Numer. Algebra Control Optim. 1 (1) (2011), 71-82.

[16] N. I Dbaruranovic-milicic and M. Gardasevic-Filipovic, A multi step curve search algorithm in nonlinear convex case, Facta Univ., Ser. Math. Inf. 25 (2010), 11-24.

[17] N. I Dbaruranovic-Milicic, A multi step curve search algorithm in nonlinear optimization, Yugoslav J. Oper. Res. 18 (2008), 47-52.

[18] M. J Petrovic, P.S Stanimirovic, Accelerated double direction method for solving unconstrained optimization problems, Math. Probl. Eng., 2014 (2014), Article ID 965104, 8 pages.

[19] L. Zhang W.J.Zhou, and D. H -Li, A descent modified Polak-Ribirie-Polyak conjugate gradient method and its global convergence, IMA J. Numer. Anal. 26 (2011), 629-640.

[20] Zhou and D.Shen, An inexact PRP Conjogate gradient method for symmetric nonlinear equations, Numer. Funct. Anal. Optim. 35 (3) (2014), 370-388.

[21] Musa Y.B., A family of BFGS methods for solving symmetric nonlinear system of equations, (2015).

[22] E. Dolan and J. More, Benchmarking optimization software with performance profiles, J. Math. Program, 91 (2002), 201-213.

[23] A. B. Abubakar, ON Improved Broyden-Type Method For Systems of Nonlinear Equations, M.Sc. Thesis BUk. 5154. (2014).

[24] M.A Hafiz and Mahmud S. Bahgat, An Efficient two-step Iterative Method For Solving System of Nonlinear Equations, J. Math. Res. 5 (2012), 3456-3476.

[25] G. Yuan et.al, A nonmonotone line search method for symmetric nonlinear equations, Intell. Control Autom. 1 (1) (2010), 28-35.

[26] Y. H. Dai, A nonmonotone conjugate gradient algorithm for unconstrained optimization, J. Syst. Sci. Complex, 15 (2002), 139-145.

[27] La Cruz, W., Martinez, J.M., Raydan, M., Spetral residual method without gradient information for solving large-scale nonlinear systems of equations, Math. Comput. 75 (255) (2006), 1429-1448.

[28] M.Y Waziri, Z.A Majid, An enhanced matrix-free secant method via predictor-corrector modified line search strategies for solving systems of nonlinear equations, Int. J. Math. Math. Sci. 2013 (2013), Article ID 814587, 6 pages

[29] K.E. Hilstrom, A Simulation tese approach to the evalution of nonlinear optimization algorithms, A.C.M. Trans. Math. Soft, 3 (1997),305-315.

[30] M.Y Waziri, Leong, W.J, Hassan, M.A. and Monsi, M, Jacobian computation-free Newton method for systems of Non-Linear equations, J. Num. Math. Stochastic. 2 (2010), 54-63.

[31] Mohammed Waziri Yusuf, Leong Wah June and Malik Abu Hassan, Jacobian-Free Diagonal Newton's Method for Solving Nonlinear Systems with Singular Jacobian, Malaysian J. Math. Sci. 5 (2011), 241-255.

[32] M.Y Waziri,Z.A Majid, An enhanced matrix-free secant mehod via predictor-corrector modified line search strategies for solving systems of nonlinear equations, Int. J. Math. Math. Sci.2013 (2013), Article ID 814587.

[33] H. Mohammed, M.Y Waziri, On Broyden-like update via some quadratures for solving nonlinear systems of equations, Turkish J. Math. 39 (2015), 335-345.