STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **8**, March 2020, pp 165–174.
Published online in International Academic Press (www.IAPress.org)

Inexact Double Step Length Method for Solving Systems of Nonlinear Equations

Abubakar Sani Halilu [1,*], Mohammed Yusuf Waziri [2], Yau Balarabe Musa [1]

[1] *Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria*
[2] *Department of Mathematical Sciences, Bayero University, Kano, Nigeria*

Abstract In this paper, a single direction with double step length method for solving systems of nonlinear equations is presented. Main idea used in the algorithm is to approximate the Jacobian via acceleration parameter. Furthermore, the two step lengths are calculated using inexact line search procedure. This method is matrix-free, and so is advantageous when solving large-scale problems. The proposed method is proven to be globally convergent under appropriate conditions. The preliminary numerical results reported in this paper using a large-scale benchmark test problems show that the proposed method is practically quite effective.

Keywords Acceleration Parameter, Double Step Length, Global Convergent, Inexact line Search

AMS 2010 subject classifications 65H11, 65K05, 65H12, 65H18

DOI: 10.19139/soic-2310-5070-532

1. Introduction

Consider the systems of nonlinear equations:

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map.
Among various methods for solving nonlinear equations (1), Newton's method is quite welcome due to its nice properties such as the rapid convergence rate, the decreasing of the function value sequence[14]. The iterative formula of a Newton method is given by

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, ..., \tag{2}$$

where, α_k is a step length to be computed by a line search technique [4, 5, 7], x_{k+1} represents a new iterative point, x_k is the previous iteration, while d_k is the search direction to be calculated by solving the following linear system of equations,

$$F'(x_k)d_k = -F(x_k), \tag{3}$$

where $F'(x_k)$ is the Jacobian matrix of $F(x_k)$ at x_k.
A basic requirement of the line search is to sufficiently decrease the function values i.e. to establish

$$\|F(x_k + \alpha_k d_k)\| \le \|F(x_k)\|. \tag{4}$$

[*] Correspondence to: Abubakar Sani Halilu (Email: abubakarsani.halilu@jsu.edu.ng). Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Moreover (1) can come from an unconstrained optimization problem, a saddle point, and equality constrained problem [10]. Let f be a norm function defined by

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

where, $||\cdot||$ to stand for the Euclidian norm. Then the nonlinear equations problem (1) is equivalent to the following global optimization problem

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

and condition (4) is equivalent to

$$f(x_k + \alpha_k d_k) \leq f(x_k) \tag{6}$$

Furthermore, the search direction d_k is generally required to satisfy the descent condition

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

It is vital to mention that due to the well known shortcomings of Newton method, a double step length has been proposed in [13, 16] and the iterative procedure is given as:

$$x_{k+1} = x_k + \alpha_k c_k + \beta_k d_k, \tag{7}$$

r where x_{k+1} represents a new iterative point, x_k is the previous iterative point, α_k and β_k denote the step lengths, while c_k and d_k are search directions respectively. The step lengths α_k and β_k can also be computed either exact or in exact. The ideal line search rule is the exact one [1] that satisfies

$$f(x_k + \alpha_k d_k) = \min_{\alpha>0} f(x_k + \alpha d_k). \tag{8}$$

In fact, the exact step length is difficult or even impossible to seek in practical computation. Therefore the most frequently used line search in practice is inexact line search [3, 9, 11]. Brown and Saad [5] proposed the following line search rule to obtain the step length α_k

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \sigma \alpha_k \nabla f(x_k)^T d_k, \tag{9}$$

where $\sigma \in (0, 1)$ and the search direction d_k can be obtained in several ways (see[2, 6, 7, 8] and references therein). From the technique in (9), it is easy to see that the Jacobian matrix must be computed at every iteration, which will increase the computing difficulty, especially for the large-scale problems or when the matrix is expensive to calculate. Considering these points, a new backtracking inexact technique is presented by Yuan* and Lu [12] in order to obtain the step length α_k:

$$\|F(x_k + \alpha_k d_k)\|^2 \leq \|F(x_k)\|^2 + \delta \alpha_k^2 F(x_k)^T d_k, \tag{10}$$

where $\delta \in (0, 1)$. The global convergence and the superlinear convergence of this method is established. The numerical results showed that the line search technique (10) is more effective than the normal methods.

In this article, we introduce the derivative-free line search proposed by Li and Fukushima [10] in order to compute our step lengths α_k and β_k.

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

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty, \tag{11}$$

$$\begin{aligned} f(x_k + (\alpha_k + \beta_k)d_k) - f(x_k) \leq & -\omega_1 \|(\alpha_k + \beta_k)F(x_k)\|^2 \\ & -\omega_2 \|(\alpha_k + \beta_k)d_k\|^2 + \eta_k f(x_k). \end{aligned} \tag{12}$$

Let i_k be the smallest non negative integer i such that (12) holds for $\alpha = r^i$ and $\beta = q^i$. Let $\alpha_k = r^{i_k}$ and $\beta_k = q^{i_k}$.
An iterative method that generates a sequence $\{x_k\}$ satisfying (4) or (6) is called a norm descent method [16]. If d_k is a descent direction of f at x_k, then inequality (6) holds for all $\alpha_k > 0$ sufficiently small. Accordingly, the related iterative method is a norm descent method. In particular, Newton's method with line search is norm descent. For a quasi-Newton method, however, d_k may not be a descent direction of f at x_k even if B_k is symmetric and positive definite. To globalize a quasi-Newton method, Li and Fukushima [10] proposed an approximately norm descent line search technique and established global and superlinear convergence of a Gauss-Newton based BFGS method for solving symmetric nonlinear equations. The method in [10] is not norm descent. In addition, the global convergence theorem is established under the assumption that $F'(x_k)$ is uniformly nonsingular.
The drawback of the technique (3) is the need to compute the Jacobian matrix $F'(x_k)$ at every iteration, which will increase the computing difficulty, due to the first-order derivative of the system because sometimes they are not even available or could not be obtained exactly [14] especially for the large-scale problems. In this case Newton's method cannot be applied directly [7]. Therefore, motivated by[13] the purpose of this article is to develop a derivative-free method with decent direction for solving system of nonlinear equations via

$$F'(x_k) \approx \gamma_k I, \tag{13}$$

Where I is an identity matrix. The proposed method has a norm descent property without computing the Jacobian matrix with less number of iterations and CPU time that is globally convergent.
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. Derivation of the Method

In this section we compute the two step lengths α_k and β_k in (7) using inexact line search procedure. This is made possible by making the two directions in (7) to be equal, i.e $c_k = d_k$. In order to incorporate more information of the iterates at each iteration and to improve good direction towards the solution, we suggest a new direction to be defined as:

$$d_k = -\gamma_k^{-1} F(x_k), \tag{14}$$

where $\gamma_k > 0$ is an acceleration parameter. By putting (14) in to (7) we obtained

$$x_{k+1} = x_k - (\alpha_k + \beta_k)\gamma_k^{-1} F(x_k). \tag{15}$$

We now proceed to obtain the proposed acceleration parameter. Therefore, we start from

$$F(x_{k+1}) \approx F(x_k) + F'(\xi)(x_{k+1} - x_k) \tag{16}$$

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

$$\xi = x_k + \delta(x_{k+1} - x_k) = x_k - \delta(\alpha_k + \beta_k)\gamma_k^{-1} F(x_k) \quad 0 \leqslant \delta \leqslant 1. \tag{17}$$

Bearing in mind that the distance between x_k and x_{k+1} is small enough, we can take $\delta = 1$ in (17) and get $\xi = x_{k+1}$. Thus we have

$$F'(\xi) \approx \gamma_{k+1} I. \tag{18}$$

Now from (16) and (18) its not difficult to verify that:

$$F(x_{k+1}) - F(x_k) = -\gamma_{k+1}(\alpha_k + \beta_k)\gamma_k^{-1} F(x_k). \tag{19}$$

Taking $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = -(\alpha_k + \beta_k)\gamma_k^{-1}F(x_k)$, and multiplying y_k^T to the both side of (19) the acceleration parameter yields:

$$\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}. \tag{20}$$

From (14) and (15) we have the general scheme as:

$$x_{k+1} = x_k + (\alpha_k + \beta_k)d_k. \tag{21}$$

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

Algorithm 1(IDS).

```
STEP 1: Given $x_0$, $\gamma_0 = 1$, $\epsilon = 10^{-3}$, set $k = 0$.
STEP 2: Compute $F(x_k)$.
STEP 3: If$\|F(x_k)\| \leq \epsilon$, then stop, else goto next STEP.
STEP 4: Compute search direction $d_k = -\gamma_k^{-1}F(x_k)$.
STEP 5: Compute step the lengths $\alpha_k$ and $\beta_k$ (using (12)).
STEP 8: Set $x_{k+1} = x_k + (\alpha_k + \beta_k)d_k$.
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 (IDS). To begin with, let us defined the level set

$$\Omega = \{x | \|F(x)\| \leq \|F(x_0)\|\}. \tag{22}$$

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

$$\|F'(x)\| \leq M \quad \forall x \in N, \tag{23}$$

and

$$m\|d\|^2 \leq d^T F'(x)d \quad \forall x \in N, d \in \mathbb{R}^n. \tag{24}$$

From the level set we have:

$$\|F(x)\| \leq m_1 \quad \forall x \in \Omega. \tag{25}$$

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

$$m\|d\| \leq \|F'(x)d\| \leq M\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{26}$$

$$m\|x - y\| \leq \|F(x) - F(y)\| \leq M\|x - y\| \quad \forall x, y \in N. \tag{27}$$

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

$$m\|x - x^*\| \leq \|F(x)\| = \|F(x) - F(x^*)\| \leq M\|x - x^*\|, \tag{28}$$

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

$$\|(F'(x_k) - \gamma_k I)d_k\| \leq \epsilon\|F(x_k)\| \tag{29}$$

where $\epsilon \in (0, 1)$ is a small quantity [12].

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

$$\nabla f(x_k)^T d_k < 0. \tag{30}$$

proof. from (14), we have

$$\begin{aligned} \nabla f(x_k)^T d_k &= F(x_k)^T F'(x_k) d_k \\ &= F(x_k)^T [(F'(x_k) - \gamma_k I)d_k - F(x_k)] \\ &= F(x_k)^T ((F'(x_k) - \gamma_k I)d_k - \|F(x_k)\|^2, \end{aligned} \tag{31}$$

by chauchy swatz we have,

$$\begin{aligned} \nabla f(x_k)^T d_k &\leq \|F(x_k)\|\|((F'(x_k) - \gamma_k I)d_k\| - \|F(x_k)\|^2 \\ &\leq -(1 - \epsilon)\|F(x_k)\|^2. \end{aligned} \tag{32}$$

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

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

This implies that $\{x_k\} \subset \Omega$.
Lemma 3.(see[12]) Suppose that assumption 1 holds $\{x_k\}$ is generated by algorithm 1. Then there exists a constant $m > 0$ such that for all k.

$$y_k^T s_k \geq m\|s_k\|^2. \tag{33}$$

Lemma 4.(see[7]) Suppose that assumption 1 holds and $\{x_k\}$ is generated by algorithm 1. Then we have

$$\lim_{k\to\infty} \|\alpha_k d_k\| = 0, \tag{34}$$

and

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

Lemma 5. Suppose that assumption 1 holds and $\{x_k\}$ is generated by algorithm 1. Then there exists some positive constants m_2 such that for all $k > 0$,

$$\|d_k\| \leq m_2, \tag{36}$$

proof. from (27), we have

$$\begin{aligned} \|d_k\| &= \left\| -\frac{y_{k-1}^T s_{k-1} F(x_k)}{\|y_{k-1}\|^2} \right\| \\ &\le \frac{\|s_{k-1}\| \|y_{k-1}\| \|F(x_k)\|}{m^2 \|s_{k-1}\|^2} \\ &\le \frac{M\|s_{k-1}\| \|F(x_k)\|}{m^2 \|s_{k-1}\|} \\ &\le \frac{M\|F(x_k)\|}{m^2} \\ &\le \frac{M\|F(x_0)\|}{m^2}. \end{aligned} \tag{37}$$

Taking $m_2 = \frac{M\|F(x_0)\|}{m^2}$, we have (36).
The proof is completed.

We can deduce that for all k (36) 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 and $\{x_k\}$ is generated by algorithm 1. Assume further for all $k > 0$,

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

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 (34) 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 (14) we have,

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

$$\begin{aligned} \|F(x_k)\|^2 &= \| - F(x_k)^T d_k \gamma_k \| \\ &\le |F(x_k)^T d_k| |\gamma_k|. \end{aligned} \tag{43}$$

but

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

then

$$|\gamma_k^{-1}| \ge \frac{m}{M^2},$$

so from (43) we have,

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

Thus

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

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 non linear equation (1) is compared with a derivative-free CG method and its global convergence for solving symmetric nonlinear equations [2]. It should be noted here that:

(i) Inexact Double Step length Method (IDS) stands for our method and we set the following:
$\omega_1 = \omega_2 = 10^{-4}$, $r = 0.2$ and $\eta_k = \dfrac{1}{(k+1)^4}$.
(ii) A derivative-free CG (DFCG) is the method proposed by [2] and we set the following:
$\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$ $r = 0.2$ and $\eta_k = \dfrac{1}{(k+1)^4}$.
The employed computational codes was written in Matlab 7.9.0 (R2009b) and run on a personal computer 2.00 GHz CPU processor and 3 GB RAM memory. We stopped the iteration if the total number of iterations exceeds 1000 or $\|F(x_k)\| \leq 10^{-3}$. We have tried the two methods on eight test problems with different initial points and dimension (n values). problems 1-7 are from [2] and problem 8 was arbitrarily constructed by us.

Problem 1:

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

Problem 2:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (sinx_1 - 1, ..., sinx_n - 1)^T.$$

Problem 3:

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

Problem 4:

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

Problem 5:

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

Problem 6:

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

Problem 7:

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

Problem 8:

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

Table 1. The Numerical Results for IDS and DFCG on problems 1 to 8, where e=Ones(n,1)

				IDS			DFCG	
Problems	x_0	dim	iter	Time(s)	$\|F(x_k)\|$	iter	Time(s)	$\|F(x_k)\|$
1	0.5*e	100	13	0.058072	7.55E-04	33	0.137884	9.74E-05
		1000	18	0.717315	3.62E-04	53	2.285821	8.72E-05
		2000	13	1.818927	4.03E-04	54	7.791001	8.10E-05
2	e	100	9	0.061803	4.26E-04	49	0.18529	4.08E-05
		1000	11	0.480761	4.77E-04	63	2.874518	9.31E-05
		2000	11	1.517515	5.57E-04	61	9.321487	9.30E-05
3	0.01*e	100	13	0.006141	8.75E-04	52	0.021726	9.57E-05
		1000	13	0.035568	8.72E-04	54	0.105493	8.83E-05
		50000	14	0.824716	7.36E-04	55	3.550896	7.53E-05
4	0.1*e	100	10	0.008541	9.11E-04	47	0.018898	8.07E-05
		5000	12	0.108037	8.09E-04	57	0.308569	9.39E-05
		10000	13	0.178396	1.22E-04	58	0.637811	6.51E-05
5	0.7*e	100	7	0.004733	9.04E-04	431	0.31315	3.02E-06
		1000	8	0.019697	2.84E-04	431	0.996263	9.54E-06
		10000	8	0.10864	8.99E-04	431	8.684382	3.02E-05
6	0.4*e	100	5	0.004443	3.57E-04	279	0.141663	-
		1000	6	0.020648	1.12E-04	261	0.641637	-
		10000	6	0.119242	3.55E-04	279	6.450851	-
7	e	100	4	0.022792	3.14E-04	5	0.017619	2.35E-05
		1000	4	0.154512	7.43E-04	5	0.206744	7.52E-05
		10000	5	2.899741	2.25E-04	6	5.167084	4.64E-08
8	0.5*e	100	5	0.003375	7.96E-04	13	0.008095	6.11E-05
		1000	7	0.017057	9.89E-04	27	0.057786	6.03E-05
		10000	10	0.132313	4.25E-04	36	0.466298	1.29E-06

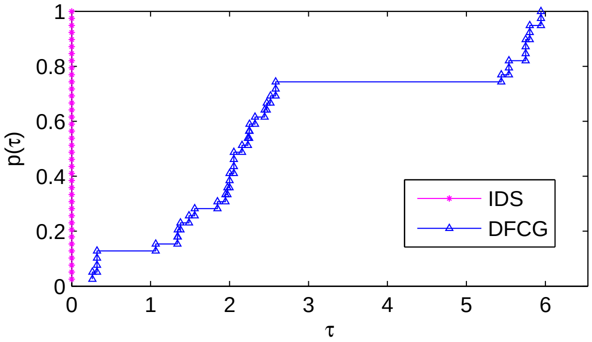

Figure 1. Performance profile of IDS and DFCG methods with respect to the number of iteration for the problems 1-8

The numerical results of the two methods are reported in Tables 1,where "iter" and "Time" stand for the total number of all iterations and the CPU time in seconds, respectively, while $\|F(x_k)\|$ is the norm of the residual at the stopping point. From Tables 1, we can easily observe that both of these methods attempt to solve the systems of nonlinear equations (1), but the better efficiency and effectiveness of our proposed algorithm was clear for it solves where DFCG fails. This is quite evident for instance with problem 6. In particular, the IDS method considerably outperforms the DFCG for almost all the tested problems, as it has the least number of iterations and CPU time, which are even much less than the CPU for the DFCG method. This is apparently due to the computation of double step length in each iteration of the IDS as well as the approximation of the Jacobian through the acceleration parameter.

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 [15]. 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.

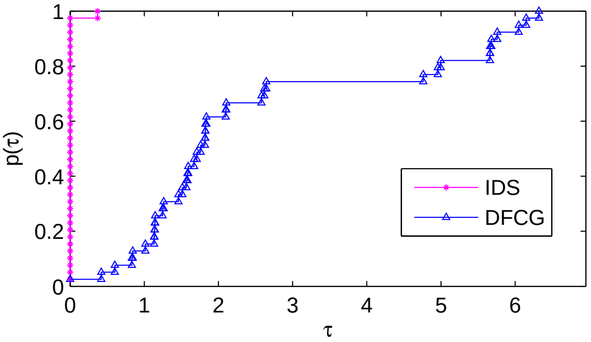

Figure 2. Performance profile of IDS and DFCG methods with respect to the CPU time (in second) for the problems 1-8

5. Conclusion

In this paper we present an Inexact Double Step length (IDS) method for solving systems of nonlinear equations and compare its performance with that of a derivative-free conjugate gradient (DFCG) method for symmetric nonlinear equations [2] by doing some numerical experiments. We however proved the global convergence of our proposed method by using a derivative-free line search, and the numerical results show that our method is very efficient.

REFERENCES

1. Zhen-Jun Shi and Jie Shen, *Convergence of nonmonotone line search method*, Journal of Computational and Applied Mathematics, vol.193, pp. 397-412, 2006.
2. M.Y Waziri and Jamilu Sabiu, *A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equation*, Journal of Mathematics and Mathematical Sciences, vol. 2015, pp. 8, 2015.
3. W. Cheng and Q. Hu, *A Nonmonotone Spectral Gradient Method for l_1-regularized Least Squares*, Statistics, Optimization and Information Computing, vol.4, pp.265-277, 2016.
4. M.Y Waziri and A.S. Halilu, *An improved derivative-free method via double direction approach for solving systems of nonlinear equations*, Journal of the Ramanujan Mathematical Society, vol. 33, no. 1, pp. 75-89, 2018.
5. P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton-Krylov algorithms search*, SIAM Journal on Optimization, vol.4, no.2, pp.297-330, 1994.
6. M. Y. Waziri, K. Ahmed and J. Sabiu, *A Dai-Liao conjugate gradient method via modified secant equation for system of nonlinear equations*, Arabian Journal of Mathematics, pp.15, 2019.
7. A.S Halilu and M.Y Waziri, *A transformed double steplength method for for solving large-scale systems of nonlinear equations*, Journal of Numerical Mathematics and Stochastics, vol.9, no. 1, pp. 20–23, 2017.
8. M. Y. Waziri, K. Ahmed and J. Sabiu, *A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations*, Applied Mathematics and Computation, vol.361, pp. 645-660, 2019.
9. M.Raydan, *On Barzilai and Borwein choice of step length for the gradient method*, IMA Journal of Numerical Analysis, vol.13, pp. 321–326.
10. D.Li and M.Fukushima, *A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equations*, SIAM Journal on numerical Analysis, vol.37, pp. 152–172, 1999.
11. I. A. Osinuga and S. O. Yusuff, *Quadrature based Broyden-like method for systems of nonlinear equations*, Statistics, Optimization and Information Computing, vol.6, pp. 130-138, 2018.
12. Gongli Yuan*, Xiwen Lu, *A new backtracking inexact BFGS method for symmetric nonlinear equations*, Journal of Computers and Mathematics with Applications, vol.55, pp. 116–129, 2008.
13. M.J petrovic, *An accelerated double step size model in unconstrained optimization*, Journal of Mathematics and Computation, vol.250, pp. 309–319, 2015.
14. J.E Dennis, Jr and R.B Schnabel, *Numerical method for unconstrained optimization and non-linear equations*, practice Hall, Englewood cliffs, NJ, USA, 1983.
15. E.Dolan and J.Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, vol. 91, no. 2, pp. 201–213, 2002.
16. A.S Halilu and M.Y Waziri, *Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations*, International Journal of Applied Mathematical Research, vol.6, no. 4, pp. 147–156, 2017.