En enhanced matrix-free method via double step length 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*
[2]*Department of Mathematical Sciences, Bayero University, Kano, Nigeria*
**Corresponding author E-mail:abubakarsaddiqu@gmail.com*

Abstract

In this paper, a new approach for solving system of nonlinear equations is present , that is based on approximating the Jacobian into a diagonal matrix by means of acceleration parameter. This method reduced the two step length parameters into a single step length and employs a backtracking process to obtain a suitable step length. Furthermore, this method is a matrix-free and so it gives an advantage to solve large-scale nonlinear system of equations. Under appropriate conditions, we show that the proposed method is globally convergent. The preliminary numerical results show that the method is practically effective.

Keywords: Acceleration parameter; Convergent rate; Line search; Matrix-free; Newton method.

1. Introduction

Consider the the system of nonlinear equations:

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map, F is assume to satisfy the following assumptions:
Assumption 1.
(1) There exists $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.
(2) F is continuously differentiable mapping in a neighborhood of x^*.
The renowned method for finding the solution to (1) is the Newton's method. The method is simple to implement, and generates an iterative sequences$\{x_k\}$ from a given initial guess x_0 in a neighborhood of x^* via

$$x_{k+1} = x_k - \left(F'(x_k)\right)^{-1} F(x_k), \tag{2}$$

where $k = 0, 1, 2, ...$ and $F'(x_k)$ is the Jacobian matrix.

The attractive features of this method are easy implementation and rapid convergence [15]. However, the Newton's method requires the computation of Jacobian matrix, which demands the first-order derivative of the system. The computation of some functions derivative are costly in practice, sometimes they are not even available or could not be obtained exactly. In this case Newton's method cannot be applied directly [17, 22, 23]. To overcome such difficulty, the simple modification on the Newton method is fixed Newton method [18] for the determination of solution of x^* is given by

$$x_{k+1} = x_k - \left(F'(x_0)\right)^{-1} F(x_k), \tag{3}$$

where $k = 0, 1, 2, ...$ The method avoids the computation and storing of the Jacobian in each iteration(except at k=0), but still it requires solving the system of *n* linear equations which consumes more CPU time as the system's dimension increases [18].
A quasi-Newton's method is another variant of Newton-type methods and replace the Jacobian or its inverse with an approximation which can be updated at each iteration [13], and its updating scheme is given by

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \tag{4}$$

where B_k is the approximation of Jacobian at x_k. The main idea behind quasi-Newton method is to eliminate the evaluation cost of the Jacobian matrix [13, 19, 25].

It is vital to mention that due to the well known shortcomings of Newton method a double step length has been proposed by [14] and the iterative procedure is given as:

$$x_{k+1} = x_k + \alpha_k b_k + \beta_k c_k, \tag{5}$$

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 b_k and c_k are search directions respectively.

It is very important to state that the transformation of double step length methods was used in unconstrained optimization problem. They are particularly efficient due to their convergence properties, simple implementation, and low storage requirement [20]. Nevertheless, the study of transformation of double step length methods for solving system of nonlinear equations is very scanty, so

for this reason we are motivated to write this paper.

We are interested in approximating the Jacobian with diagonal matrix via

$$F'(x_k) \approx \gamma_k I, \qquad (6)$$

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

$$f(x) = \frac{1}{2} ||F(x)||^2. \qquad (7)$$

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

$$min f(x), \quad x \in \mathbb{R}^n. \qquad (8)$$

The double direction method has been proposed in [1], using multi-step iterative information and curve search to generate new iterative points. However, a multi-step algorithm for minimization of a non differentiable function is presented in [1]. Moreover, a double direction method for solving unconstrained optimization problem is proposed in [12]. Recently, a double step size method for solving unconstrained optimization problem is proposed in [14]. There are several procedure for the choice of the search direction [2, 4, 5, 6, 7, 8, 9, 11, 24].

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]. The requirement of the line search is to sufficiently decrease the function values along the ray $x_k + \alpha_k d_k$, $\alpha_k > 0$.

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 Results

In this section we present the reduction of the two step lengths α_k and β_k in (5) into a single step length. This reduction is made possible by an additional assumption:

$$\alpha_k + \beta_k = 1. \qquad (9)$$

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 (5) to be defined as:

$$b_k = -\gamma_k^{-1} F(x_k), \qquad (10)$$

where $\gamma_k > 0$ is an acceleration parameter.

$$c_k = -F(x_k). \qquad (11)$$

So by putting (9), (10) and (11) into (5) we have the iterate

$$x_{k+1} = x_k - \alpha_k \left(\gamma_k^{-1} + \frac{1}{\alpha_k} - 1 \right) F(x_k). \qquad (12)$$

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

$$F(x_{k+1}) \approx F(x_k) + F'(\xi)(x_{k+1} - x_k) \qquad (13)$$

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

$$\xi = x_k + \delta(x_{k+1} - x_k), \quad 0 \leqslant \delta \leqslant 1. \qquad (14)$$

putting in mind that the distance between x_k and x_{k+1} is small enough, we can take $\delta = 1$ in (14) and get $\xi = x_{k+1}$. Thus we assume

$$F'(\xi) \approx \gamma_{k+1} I. \qquad (15)$$

Now from (13) and (15) its not difficult to verify that:

$$F(x_{k+1}) - F(x_k) = \gamma_{k+1}(x_{k+1} - x_k). \qquad (16)$$

Taking $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$, we have

$$y_k = \gamma_{k+1} s_k, \qquad (17)$$

by multiplying y_k^T to the both side of (17) the acceleration parameter yields:

$$\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}. \qquad (18)$$

now, from (12) we can easily show that, our direction is

$$d_k = - \left(\gamma_k^{-1} + \frac{1}{\alpha_k} - 1 \right) F(x_k), \qquad (19)$$

then using (12) and (19) we have the general scheme as:

$$x_{k+1} = x_k + \alpha_k d_k. \qquad (20)$$

We use the back tracking type 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 η_k be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty, \qquad (21)$$

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1 ||\alpha_k F(x_k)||^2 - \omega_2 ||\alpha_k d_k||^2 + \eta_k f(x_k) \qquad (22)$$

Let i_k is the smallest non negative integer i such that (22) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

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

Algorithm 1(EMFD)
STEP 1: Given x_0, $\gamma_0 = 1$, , $\alpha_0 > 0$, $\varepsilon = 10^{-4}$, set $k = 0$.
STEP 2: Compute $F(x_k)$.
STEP 3: If $||F(x_k)|| \leq \varepsilon$ then stop, else goto STEP 4.
STEP 4: Compute search direction d_k (using (19)).
STEP 5: Compute step length α_k (using (22)).
STEP 6: Set $x_{k+1} = x_k + \alpha_k d_k$.
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 2.

3. Convergence Analysis

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

$$\Omega = \{x | \|F(x)\| \le \sqrt{e^\eta} \|F(x_0)\|\}. \tag{23}$$

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

Assumption 2.
(i) In some neighborhood N of Ω the Jacobian of F is bounded and positive definite on N. i.e there exists a positive constants $M > m > 0$ such that

$$\|F'(x)\| \le M \quad \forall x \in N, \tag{24}$$

and

$$m\|d\|^2 \le d^T F'(x)d \quad \forall x \in N, d \in \mathbb{R}^n. \tag{25}$$

.

From the level set we have:

$$\|F(x)\| \le m_1 \quad \forall x \in \Omega. \tag{26}$$

Remarks:
Assumption 2 imply that there exist a constants $M > m > 0$ such that

$$m\|d\| \le \|F'(x)d\| \le M\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{27}$$

$$\frac{1}{m}\|d\| \le \|F'(x)^{-1}d\| \le \frac{1}{M}\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{28}$$

$$m\|x - y\| \le \|F(x) - F(y)\| \le M\|x - y\| \quad \forall x, y \in N. \tag{29}$$

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

$$m\|x - x^*\| \le \|F(x)\| \le \|\|F(x) - F(x^*)\| \le M\|x - x^*\|,$$

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

Lemma 1. Let the sequence $\{x_k\}$ is generated by algorithm 1. Then $\{x_k\} \subset \Omega$.
proof. From (22) we have for all k

$$f(x_{k+1}) \le (1 + \eta_k)f(x_k)$$

$$\vdots$$

$$\le f(x_0)\prod_{i=0}^{k}(1 + \eta_i)$$

$$\le f(x_0)\left(\frac{1}{k+1}\sum_{i=0}^{k}(1 + \eta_i)\right)^{k+1}$$

$$\le f(x_0)\left(1 + \frac{1}{k+1}\sum_{i=0}^{k}\eta_i\right)^{k+1}$$

$$\le f(x_0)\left(1 + \frac{\eta}{k+1}\right)^{k+1}$$

$$\le e^\eta f(x_0)$$

Thus we have,
$$\|F(x_{k+1})\| \le \sqrt{e^\eta}\|F(x_0)\|.$$

Lemma 2.(see[13]) Suppose that assumption 2 holds $\{x_k\}$ is generated by algorithm 1. Then there exists a constant $m_2 > 0$ such that for all k

$$y_k^T s_k \ge m_2\|s_k\|^2. \tag{30}$$

Lemma 3. Suppose that assumption 2 holds and $\{x_k\}$ is generated by algorithm 1. Then we have

$$\lim_{k\to\infty}\|\alpha_k d_k\| = \lim_{k\to\infty}\|s_k\| = 0, \tag{31}$$

and

$$\lim_{k\to\infty}\|\alpha_k F(x_k)\| = 0. \tag{32}$$

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

$$\omega_2\|\alpha_k d_k\|^2 \le \omega_1\|\alpha_k F(x_k)\|^2 + \omega_2\|\alpha_k d_k\|^2$$
$$\le \|F(x_k)\|^2 - \|F(x_{k+1})\|^2 + \eta_k\|F(x_k)\|^2 \tag{33}$$

by summing the above inequality, we have

$$\omega_2\sum_{i=0}^{k}\|\alpha_i d_i\|^2 \le \sum_{i=0}^{k}\left(\|F(x_i)\|^2 - \|F(x_{i+1})\|^2\right) + \sum_{i=0}^{k}\eta_i\|F(x_i)\|^2$$

$$= \|F(x_0)\|^2 - \|F(x_{k+1})\|^2 + \sum_{i=0}^{k}\eta_i\|F(x_i)\|^2$$

$$\le \|F(x_0)\|^2 + m_1^2\sum_{i=0}^{k}\eta_i$$

$$\le \|F(x_0)\|^2 + m_1^2\sum_{i=0}^{\infty}\eta_i, \tag{34}$$

so from (26) and fact that $\{\eta_k\}$ satisfies (21) then the series $\sum_{i=0}^{\infty}\|\alpha_i d_i\|^2$ is convergent. This implies (31). By similar way we can prove that (32) holds.

Lemma 4. Suppose that assumption 2 holds and $\{x_k\}$ is generated by algorithm 1. Then there exist a constant $m_3 > 0$ such that for all $k > 0$,

$$\|d_k\| \le m_3. \tag{35}$$

proof. from (26) and (29) we have

$$\|d_k\| = \left\|-\left(\frac{1}{\gamma_k} + \frac{1}{\alpha_k} - 1\right)F(x_k)\right\|$$

$$\le \left\|\left(\frac{1}{\gamma_k}\right)F(x_k)\right\| + \left\|\left(\frac{1}{\alpha_k} - 1\right)F(x_k)\right\|$$

$$= \left\|\left(\frac{y_k^T s_k}{\|y_k\|^2}\right)F(x_k)\right\| + \left(\frac{1}{\alpha_k} - 1\right)\|F(x_k)\|$$

$$\le \frac{\|y_k\|\|s_k\|}{\|y_k\|^2}\|F(x_k)\| + \left(\frac{1}{\alpha_k} - 1\right)\|F(x_k)\|$$

$$\le \frac{\|s_k\|}{m\|s_k\|}\|F(x_k)\| + \left(\frac{1}{\alpha_k} - 1\right)\|F(x_k)\|$$

$$\le \frac{m_1}{m} + \left(\frac{1}{\alpha_k} - 1\right)m_1. \tag{36}$$

Taking $m_3 = \dfrac{m_1}{m} + \left(\dfrac{1}{\alpha_k} - 1\right) m_1$, we have (35). We can deduce that for all k (35) 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 6. Suppose that assumption 2 holds, $\{x_k\}$ is generated by algorithm 1. Assume further for all $k > 0$,

$$\alpha_k \geq c \frac{|F(x_k)d_k|}{\|d_k\|^2}, \tag{37}$$

where c is some positive constant. Then

$$\lim_{k \to \infty} \|F(x_k)\| = 0. \tag{38}$$

Proof. From lemma 5 we have (35). Therefore by (31) and the boundedness of $\{\|d_k\|\}$, we have

$$\lim_{k \to \infty} \alpha_k \|d_k\|^2 = 0, \tag{39}$$

combining (32) and (34) we have

$$\lim_{k \to \infty} |F(x_k)^T d_k| = 0. \tag{40}$$

On the other hand from (19) we have,

$$F(x_k)^T d_k = -\left(\frac{1}{\gamma_k} + \frac{1}{\alpha_k} - 1\right) \|F(x_k)\|^2, \tag{41}$$

$$\begin{aligned} \|F(x_k)\|^2 &= -\frac{F(x_k)^T d_k}{\left(\frac{1}{\gamma_k} + \frac{1}{\alpha_k} - 1\right)} \\ &\leq \frac{|F(x_k)^T d_k|}{\left|\frac{1}{\gamma_k} + \frac{1}{\alpha_k} - 1\right|}, \end{aligned} \tag{42}$$

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 (42) we have,

$$\|F(x_k)\|^2 \leq \frac{|F(x_k)^T d_k|}{\left|\frac{m}{M} + \frac{1}{\alpha_k} - 1\right|}. \tag{43}$$

Thus

$$0 \leq \|F(x_k)\|^2 \leq \frac{|F(x_k)^T d_k|}{\left|\frac{m}{M} + \frac{1}{\alpha_k} - 1\right|} \longrightarrow 0. \tag{44}$$

Therefore

$$\lim_{k \to \infty} \|F(x_k)\| = 0. \tag{45}$$

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].

(i) An enhanced Matrix-free via double step length approach (EMFD) stands for our method and we set the following:
$\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^4}$.

(ii) A derivative-free CG (DFCG) is the method proposed in [2] and we have the following:
$\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^4}$.

The 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 $\|F(x_k)\| \leq 10^{-4}$. We tested the two methods on ten test problems with different initial points and dimension (n values). problems 1-7 are from [2] and problem 8 was arbitrarily constructed by us, while problems 9 and 10 are from [21].

Problem 1:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (e_1^x - 1, ..., e_n^x - 1)^T. \quad x_0 = (0.5, 0 \tag{46}$$

Problem 2:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (sinx_1 - 1, ..., sinx_n - 1)^T. \quad x_0 = (1 \tag{47}$$

Problem 3:

$$\begin{aligned} F_1(x) &= x_1(x_1^2 + x_2^2) - 1, \\ F_i(x) &= x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2), \\ F_n(x) &= x_n(x_{n-1}^2 + x_n^2). \\ i &= 2, 3, ..., n-1. \\ x_0 &= (0.01, 0.01, ..., 0.01)^T. \end{aligned} \tag{48}$$

Problem 4:

$$\begin{aligned} F_{3i-2}(x) &= x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1, \\ F_{3i-1}(x) &= x_{3i-2}x_{3i-2}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2, \\ F_{3i}(x) &= e^{-x_{3i-2}} - e^{-x_{3i-1}}. \\ i &= 1, ..., \frac{n}{3}. \\ x_0 &= (0.1, 0.1, ..., 0.1)^T. \end{aligned} \tag{49}$$

Problem 5:

$$\begin{aligned} F_i(x) &= (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2. \\ i &= 1, 2, ..., n. \\ x_0 &= (0.7, 0.7, ..., 0.7)^T. \end{aligned} \tag{50}$$

Problem 6:

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

(51)

Problem 7:

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

(52)

Problem 8:

$$F_i(x) = 0.i(1 - x_i)^2 - e^{-x_i^2},$$
$$F_n(x) = \frac{n}{10}(1 - e^{-x_n^2}).$$
$$i = 1, 2, ..., n - 1.$$
$$x_0 = (0.5, 0.5, ..., 0.5)^T.$$

(53)

Problem 9:

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

(54)

problem 10:

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

(55)

The numerical results of the two(2) methods are reported in tables 1 and 2, 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 and 2, we can easily perceive that all these methods attempted to solve the system of nonlinear equations (1), but the efficiency and effectiveness of our proposed algorithm was clear for it solve where DFCG fails. For instance see problem 6. In particular, the EMFD method considerably out performs the DFCG for almost all the tested problems, as it has the least number of iterations and the CPU time, which are much less than DFCG method. This is due to the computation of step length in each iteration as well as the approximation of the Jacobian through 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 Mor*é* [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.

Table 1: The Numerical Results for EMFD and DFCG on problems 1 to 10

Problems	Dim	EMFD			DFCG		
		Iter	Time(s)	$\|F(x_k)\|$	Iter	Time(s)	$\|F(x_k)\|$
1	10	13	0.058072	7.55E-04	33	0.137884	9.74E-05
	100	15	0.082624	4.92E-04	38	0.182246	9.55E-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	10	9	0.061803	4.26E-04	49	0.18529	4.08E-05
	100	10	0.082281	8.71E-04	60	0.291577	8.65E-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	10	13	0.006141	8.75E-04	52	0.021726	9.57E-05
	100	13	0.009908	7.68E-04	52	0.021726	9.57E-05
	1000	13	0.035568	8.72E-04	54	0.105493	8.83E-05
	2000	13	0.063945	9.35E-04	54	0.176152	8.43E-05
	3000	13	0.078942	9.75E-04	62	0.237935	9.52E-05
	50000	14	0.824716	7.36E-04	55	3.550896	7.53E-05
4	10	10	0.008541	9.11E-04	47	0.018898	8.07E-05
	100	11	0.010594	4.45E-04	66	0.034618	9.72E-05
	1000	12	0.032936	3.62E-04	60	0.072484	8.25E-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	10	7	0.00348	2.86E-04	431	0.176174	9.54E-07
	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
	5000	8	0.061381	6.36E-04	431	4.354077	2.13E-05
	10000	8	0.10864	8.99E-04	431	8.684382	3.02E-05

Table 1: Continued.

Problems	Dim	EMFD			DFCG		
		Iter	Time(s)	$\|F(x_k)\|$	Iter	Time(s)	$\|F(x_k)\|$
6	10	5	0.003168	1.13E-04	279	-	-
	100	5	0.004443	3.57E-04	279	-	-
	1000	6	0.020648	1.12E-04	261	-	-
	5000	6	0.062225	2.51E-04	279	-	-
	10000	6	0.119242	3.55E-04	279	-	-
7	10	4	0.005876	2.15E-04	5	0.036751	5.23E-06
	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
	5000	5	1.063497	1.60E-04	6	1.910818	3.28E-08
	10000	5	2.899741	2.25E-04	6	5.167084	4.64E-08
8	10	5	0.00258	9.09E-04	14	0.005212	5.80E-05
	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
	5000	11	0.091527	9.17E-04	23	0.138946	6.18E-05
	10000	10	0.132313	4.25E-04	36	0.466298	1.29E-06
9	10	4	0.001606	2.45E-06	6	0.003028	3.53E-11
	100	4	0.002407	7.74E-06	6	0.004275	1.12E-10
	1000	4	0.005958	2.45E-05	6	0.014771	3.53E-10
	5000	4	0.023534	5.47E-05	6	0.050129	7.90E-10
	10000	4	0.043643	7.74E-05	6	0.076038	1.12E-09
10	10	6	0.002224	1.73E-05	13	0.005993	4.42E-05
	100	3	0.002276	6.53E-05	5	0.00527	2.78E-06
	1000	2	0.006544	3.19E-05	4	0.018547	4.78E-06
	5000	2	0.025431	6.68E-06	3	0.04724	8.67E-05
	10000	2	0.044978	3.36E-06	3	0.074872	4.32E-05
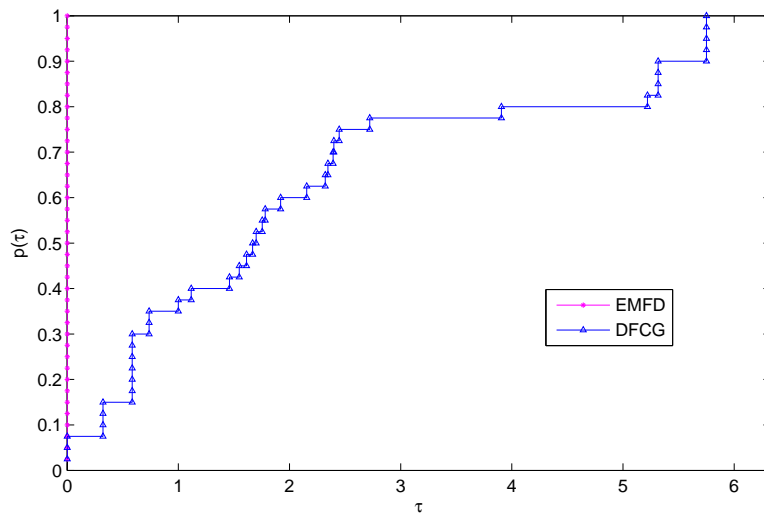
Figure 1: Performance profile of EMFD and DFCG methods with respect to the number of iteration for the problems 1-10
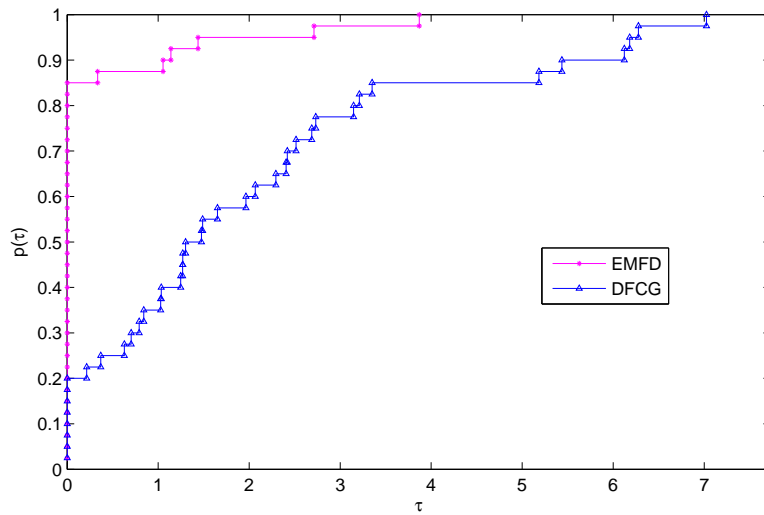
Figure 2: Performance profile of EMFD and DFCG methods with respect to the CPU time (in second) for the problems 1-10

5. Conclusion

In this paper we present an enhanced matrix-free via double step length approach (EMFD) method for solving large-scale system 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 backtracking type line search, and the numerical results show that our method is very efficient.

References

[1] N.I Dbaruranovic-milicic, A multi step curve search algorithm in non-linear optimization, *Yugoslav Journal of operation research*, 18 (2008) 47-52.

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

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

[4] sHI, Z-J," Convergence of line search method for unconstrained optimization , *Appl. Math. computation*, 157 (2004) 393-405.

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

[6] Sun, W.Yuang, Y.-X, Optimization theory and methods, *Nonlinear programming, Springer New york*, 2006.

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

[8] M.Y Waziri, W.J Leong, M.A Hassan, M.Monsi, A new Newton's method with diagonal Jacobian approximation for system of nonlinear equations, *Journal of mathematics and statistics*, 6 (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, 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, P.S Stanimirovic, Accelerated double direction method for solving unconstrained optimization problems, *Mathematical problem Eng.*, 2014.

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

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

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

[16] E.Dolan, J.More, Benchmarking optimization software with performance profiles, *Journal of Mathematical program*, 91 (2002) 201-213.

[17] Decker, D.W, C.T, Broyden's methods for a class of problems having singular Jacobian at root, *SIAM Journal of Numerical Analysis*, 17 (1985) 566-574.

[18] K. Natasa, L.Zorna, Newton-Like method with modification of rigtht-hand vector, *Journal of mathematics of Computation*, 71 (2001) 237-250.

[19] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Journal of mathematics of Computation*, 19 (1965) 577-593.

[20] Predrag S. Stanimirovic, Gradimir V.Milovanovic, Milena J. Petrovic, Natasa Z. Kontrec, A transformation of accelerated double step size method for unconstrained optimization, *Journal of mathematical problems in engineering*, 2015 8.

[21] Qin-Rong Yana, Xiao-Zhen Penga, 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.

[22] M.Y. Waziri, W.J. Leong, M.A Hassan, M. Monsi, Jacobian computation-free Newton method for systems of Non-Linear equations *Journal of numumerical Mathematics and stochastic*, 2 (2010) 54-63.

[23] Mohammed Waziri Yusuf, Leong Wah June, Malik Abu Hassan, Jacobian-Free Diagonal Newton Method for Solving Nonlinear Systems with Singular Jacobian, *Malaysian Journal of Mathematical Science*, 5 (2011), 241-255.

[24] 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, *Journal of mathematics and mathematical science*, 2013.

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