European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 07, 2020

Double Direction And Step Length Method For Solving System Of Nonlinear Equations

Abubakar Sani Halilu[1], Arunava Majumder[2*], Mohammed Yusuf Waziri[3] and Habibu Abdullahi[4]

[1,2,4]*Department of Mathematics, School of Chemical Engineering and Physical Sciences, Lovely Professional University, Phagwara, India -144411.*

[1,4]*Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria.*

[3]*Department of Mathematical Sciences, Bayero University, Kano, Nigeria.*

[*]*Corresponding Author: Arunava Majumder, Email: am.arunavamajumder@gmail.com*

Abstract: Due to the fact that single direction and step length methods for solving some iterative methods has a single correction which might fail during the iterative process. For this and other possible reasons a double direction and step length method for solving system of nonlinear equations is presented. The method used a specific technique with two direction vectors and two step length parameters. The approximation to the Jacobian matrix is achieved by a properly derived acceleration parameter. Moreover, the normed descent line search procedure is employed in this work in order to obtain the optimal step lengths. The numerical experiment shown in this paper, depicts the efficiency of the proposed method. Mathematics Subject Classification: 65H11, 65K05, 65H12, 65H18

Keywords: Acceleration parameter, Double direction, Double step length, Global convergence, Jacobian matrix.

1 Introduction

Nonlinear systems of equations form a family of problems that are equivalent to un- constrained optimization problems and they often occur in the science and technology sectors. Researchers have been looking at various examples in this field in recent years. The standard nonlinear system of equations is represented by

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

where $F : \mathrm{R}^n \to \mathrm{R}^n$ is nonlinear map and is assumed to satisfy the following assumptions:

Assumption 1

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

(2) F is continuously differentiable mapping in a neighborhood of x^*.

Throughout this paper, the space R^n denote the $n-$dimensional real space, $\|\cdot\|$ is the Euclidean norm and $Fk = F(xk)$.

The studies of such mappings are practically applied in many scientific fields, like in the systems of economic equilibrium [1-8] and chemical equilibrium [9-13]. It has practical application in Chandrasekhar H-equation that arises in the theory of radioactive heat transfer

is a nonlinear integral equation [14-18] as well. Some iterative methods for solving these problems include Newton and quasi-Newton methods [3, 4, 19]. Newton's method is very welcome because of its nice properties such as rapid con- vergence rate from a reasonably good starting point and the decreasing of the function value [17] and it's iterative scheme is given by

$xk+1 = xk + sk, \quad sk = \alpha k dk, \quad k = 0, 1, ..., \qquad (2)$

where αk is a step length, $xk+1$ represents a new iterate, xk is the previous iterate, while dk is the search direction to be determined by solving the following linear system of equations,

$Fk + Fk' dk = 0, \qquad (3)$

where $F'(xk)$ is the Jacobian matrix of $F(xk)$ at xk.

Despite the appealing characteristics of the Newton methods, the derivative F' will be computed at each iteration, so they are not ideal for solving large-scale problems. Due to this shortcoming, the double step length method has been proposed in [16] and the iterative procedure is given as:

$$xk+1 = xk + \alpha k dk + \beta k ck, \qquad (4)$$

is considered in this work. Here $xk+1$ represents a new iterative point, xk is the previous iterative point, αk and βk denote the step lengths, while dk and ck generate search directions [16, 20].

Suppose f be a merit function defined by

$$f(x) = ||F(x)||^2 \qquad (5)$$

Then (1) is the first-order necessary condition for the unconstrained optimization problem [11], where F is the gradient mapping of some function $f : \mathrm{R}^n \rightarrow \mathrm{R}$,

$Min\ f(x), x \in \mathrm{R}^n.$

The idea of double direction approach is presented by Duranovic-Milicic, in [2], via

$$xk+1 = xk + \alpha_k dk + \alpha_k^2 bk, \qquad (6)$$

where, bk and dk are search directions respectively. The method in [2] used multi-step iterative scheme and curve search in order to generate new iterates. Nonetheless, Dura- novic et.al [6] also proposed a multi-step algorithm for minimizing a non-differentiable function using double direction approach. Motivated by the work in [6], Petrovic and Stanimirovic, [13] proposed a double direction method for solving unconstrained op- timization problems. In their work, an approximate Hessian matrix is obtained via acceleration parameter γk i.e.

$$\nabla^2 f(xk) \approx \gamma k I, \qquad (7)$$

where $\nabla^2 f(xk)$ is the Hessian matrix and $f : \mathrm{R}^n \rightarrow \mathrm{R}$. The interesting feature of the scheme in [13], is that, the two derivative-free directions were presented in the scheme, where, the first direction approximated the Hessian with diagonal matrix through accel- eration parameter while the second one is the steepest decent direction. This gives the method an opportunity to solve the large-scale problems. Since the study of double direction methods for solving system of nonlinear equations is very rare in the literature, this motivated Halilu and Waziri [18] to used the scheme in [13] and proposed a derivative-free method via double direction approach for

solving system of nonlinear equations. In their work, the Jacobian matrix is approximated is via acceleration parameter $\gamma k > 0$ i.e.

$$Fk' \approx \gamma kI, \quad (8)$$

where I is an identity matrix. However, In [8], Abdullahi et al. modified the idea in [18] in solving conjugate gradient approach, the method converged globally using the nonmonotone line search proposed by Li and Fukushima in [11].

The rationale behind double direction method is that, there are two correction in the scheme (6), if one correction fails during iterative process then the second one will correct the system. In order to improve the performance of the scheme in [13] numerically, Petrovic, presented the accelerated double step size model for unconstrained optimization [16] using the iterative scheme in (4). The numerical results indicated that the method in [16, 20], performed better than the double direction method [13], because of the contribution of double step length scheme used in [16]. In [7], Halilu and Waziri, also incorporated the idea of in (4), and proposed the transformed double step length method for solving system of nonlinear equations to enhance the numerical performance of double direction method in [18].

The numerical results showed that, the method in [7], converged faster than [18]. Furthermore, Motivated by the work in [7], recently, Halilu and Waziri presented inexact double step length method for solving system of nonlinear equations [9].

Extensive numerical experiments showed that the method performed quite well by comparing it with the existing method in the literature. The attractive feature of the method in [9] is that, it has double step length and single direction. Therefore, motivated this idea, we aimed at developing a matrix-free method with double direction and step length for solving system of nonlinear equations, without computing the Jacobian matrix with less number of iterations and CPU time than the method in [9].

There are some known procedures for driving the search directions [4-6, 8, 15]. Theoretically exact optimal step length generally cannot be found in practical compu- tation, and it is also expensive to find almost exact step length. Therefore the most frequently used algorithm in practice is the inexact line search [10, 13, 14, 21-24]. A fun- damental need of the line search is to establish $\|F(xk+1)\| \leq \|F(xk)\|$.

Author's Name	IDSL method	MCGD Method	Derivative-free	Matrix-free	Double Direction	Double step lengtht
Dennis and Schnabel [17]	no	No	no	no	no	no
Li and Fukushima [11]	no	no	yes	no	no	no
Yuan and Xiwen [14]	no	no	yes	no	no	no
Waziri et al	no	no	yes	no	no	no
Duranovic [2]	no	no	yes	yes	yes	no
Halilu and Waziri [7]	no	no	yes	yes	yes	no
Duranovic and Filipovic [6]	no	no	yes	yes	yes	no
Halilu and Waziri [18]	no	no	yes	yes	yes	no
Petrovic and Stanimirovic [13]	no	no	yes	yes	yes	no
petrovic [16]	no	no	yes	yes	yes	yes
Abdullahi et al. [8]	no	yes	yes	yes	yes	no
Halilu and Waziri [9]	yes	no	yes	yes	no	yes
This article	yes	yes	yes	yes	yes	yes

Table 1: Authors contribution Table

From Table 1, IDSL and MCGD are methods in [9] and [8] respectively.

The rest of this paper is organized as follows. In section 2, the algorithm of the proposed method is presented. In Sections 3 some numerical results is reported and conclusion is made in section 4.

2 Main Result

In this section we intend to compute the two step lengths αk and βk in (4) using inexact line search procedure. We suggest a new directions ck and dk in (4) to be defined as:

$$dk = -\gamma k^{-1} Fk, \tag{9}$$

$$ck = -Fk, \tag{10}$$

so by putting (9) and (10) in to (4) we obtained

$$xk+1 = xk - (\alpha k + \beta k \gamma k)\gamma k^{-1} Fk. \tag{11}$$

Now, from Taylor's expansion of the first order the approximation of $F(xk+1)$ can be brought as follows:

$$Fk+1 \approx Fk + F'(\delta)(xk+1 - xk) \tag{12}$$

where the parameter $\delta \in [xk, xk+1]$,

$$\delta = xk + \lambda(xk+1 - xk) = (\alpha k + \beta k \gamma k)\gamma k^{-1} dk \qquad 0 \leq \lambda \leq 1. \tag{13}$$

By taking $\lambda = 1$ in (13) we get $\xi = xk+1$. Therefore we have

$$F'(\delta) \approx \gamma k+1 I. \tag{14}$$

This approximation implies

$$yk = \gamma k+1 sk, \tag{15}$$
$$(\alpha k+\beta k\, \gamma k)\, yk^T dk$$

where, $yk = Fk+1 - Fk$, $sk = (\alpha k + \beta k \gamma k)dk$. By multiplying both side of (15) by yk^T, the relation allows us to compute the parameter $\gamma k+1$ in the following way:

$$\gamma k+1 = \qquad yk^T yk \,. \tag{16}$$

So from (9) and (11) we have the general scheme as:

$$xk+1 = xk + (\alpha k + \beta k \gamma k)dk. \tag{17}$$

Now, the nonmonotone line search used in [1, 11, 15] is the best choice to compute the step lengths αk and βk.

Let $\omega 1 > 0$, $\omega 2 > 0$ and $q, r \in (0, 1)$ be constants and let $\{\eta_k\}$ be a positive sequence such that

$$\sum_k^{\infty} \eta_k < \eta < \infty \tag{18}$$

$$f(xk + (\alpha k + \beta k\gamma k)dk) - f(xk) \leq -\omega 1\|(\alpha k + \beta k\gamma k)Fk\|^2 - \omega 2\|(\alpha k + \beta k\gamma k)dk\|^2 + \eta_k f(xk). \quad (19)$$

Let ik be the smallest non negative integer i such that (19) holds for $\alpha = r^i$ and $\beta = q^i$. Let $\alpha k = r^{ik}$ and $\beta k = q^{ik}$.

Algorithm 1(DDSL).

STEP 1: Given $x0$, $\gamma 0 = 1$, $\epsilon = 10^{-4}$, set $k = 0$.

STEP 2: Compute $F(xk)$ and test a stopping criterion. If yes, then stop; otherwise continue with Step 3.

STEP 3: Compute search direction $dk = -\gamma_k^{-1}Fk$.

STEP 4: Compute step the lengths αk and βk (using (19)).

STEP 5: Set $xk+1 = xk + \lambda kdk$, where $\lambda k = \alpha k + \beta k\gamma k$.

STEP 6: Compute $Fk+1$.

STEP 7: Determine $\gamma k+1$ (using (16)).

STEP 8: Consider k=k+1, and go to STEP 3.

3 Numerical Results

In this section, some numerical results is provided to show the effectiveness of the proposed method by comparing it with the following existing methods in the literature.

- An inexact double step length method for solving systems of nonlinear equations (IDSL)[9].
- A modified conjugate gradient method via a double direction approach for solvinglarge-scale symmetric nonlinear equations (MCGD) [8].

For the three algorithms the following parameters are set $\omega 1 = \omega 2 = 10^{-4}$, $r = 0.2$ and $\eta_k = 1/(k+1)$. The computer codes used were written in Matlab 9.4.0 (R2018a) and run on a personal computer equipped with a 1.80 GHz CPU processor and 8 GB RAM. The algorithms were implemented with the same line search (19) in the experiments. The iteration is set to stop for all the methods if $\|Fk\| \leq 10^{-4}$ or when the iterations exceed 1000 but no point of xk satisfying the stopping criterion is obtained. We use the symbol '-' to represents failure due to; (i) Memory requirement (ii) Number of iterations exceed 1000. To show the extensive numerical experiments of DDSL, IDSL and MCGD methods, we have tried these methods on the previous four Benchmark test problems with different initial points and dimension (n values) between 100 to 10,000.

Table 2: Initial points

Initial Points (IP)	Values
X1	$(0.5, 0.5, \ldots, 0.5)^T$
X2	$(0.2, 0.2, \ldots, 0.2)^T$
X3	$(1.5, 1.5, \ldots, 1.5)^T$
X4	$(0.4, 0.4, \ldots, 0.4)^T$
X5	$(0, 1/2, 2/3\ldots, 1\text{-}(1/n))^T$
X6	$(-0.25, 0.25, \ldots, (-1)^n 0.25)^T$
X7	$(1, ½, 1/3, \ldots, 1/n)^T$

Problem 1[7]

$F_1 = x_1 - e^{\cos\left(\frac{x_1+x_2}{n+1}\right)},$

$F_1 = x_1 - e^{\cos\left(\frac{x_{i-1}+x_i+x_{i+1}}{n+1}\right)},$

$$F_1 = x_1 - e^{\cos\left(\frac{x_{n-1}+x_n}{n+1}\right)}, i = 2,3, \ldots . n-1.$$

Problem 2[9]

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

Problem 3[12]

$F_i(x) = x_i - 3x_i\left(\frac{sinx_i}{3} - 0.66\right) + 2, i = 1,2, \ldots , n.$

Problem 4: The discretized Chandrasekhar's H-equation (the problem of integral equation arising in radioactive heat transfer).

$$F_i = x_i - \left(1 - \frac{c}{2n}\sum_{j=1}^{n}\frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1}, i = 1,2, \ldots , n, j = 1,2, \ldots , n,$$

with $c \in [0,1)$ and $\mu = \frac{i-0.5}{n}$. (In our experiment we take c=0.1).

Table 3: Numerical results of Problem 1

		DDSL			IDSL			MCGD		
		NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖
100	x1	21	0.020308	7.38E-05	35	0.019647		8.38E-05	51	0.037003 9.97E-05
	x2	21	0.007911	8.38E-05	35	0.01576	9.51E-05	52	0.030388	8.59E-05
	x3	20	0.004392	7.36E-05	33	0.007892		9.37E-05	49	0.020116 9.49E-05
	x4	21	0.00692	7.71E-05	35	0.008354		8.76E-05	52	0.021197 7.91E-05
	x5	21	0.009167	5.89E-05	34	0.013246		9.57E-05	51	0.023046 7.96E-05
	x6	21	0.006058	9.88E-05	36	0.00776	7.85E-05	51	0.019354	9.84E-05
	x7	21	0.0058	8.88E-05	36	0.007382		7.06E-05	52	0.021039 9.11E-05
1000	x1	23	0.008289	7.48E-05	38	0.020307		9.11E-05	98	0.082055 7.76E-05
	x2	23	0.011367	8.49E-05	39	0.015225		7.24E-05	98	0.071271 8.81E-05
	x3	22	0.011744	7.47E-05	37	0.016177		7.15E-05	97	0.112219 9.72E-05
	x4	23	0.010517	7.82E-05	38	0.014206		9.53E-05	98	0.077551 8.11E-05
	x5	23	0.014227	5.82E-05	38	0.015695		7.09E-05	98	0.069258 7.95E-05
	x6	24	0.009698	5.51E-05	39	0.016115		8.54E-05	98	0.075943 7.89E-05
	x7	23	0.011454	9.15E-05	39	0.015542	7.8E-05	97	0.091594	9.49E-05
2000	x1	24	0.015516	5.82E-05	39	0.024235		9.02E-05	160	0.171997 8.33E-05
	x2	24	0.017208	6.61E-05	40	0.028406		7.17E-05	159	0.168376 9.45E-05
	x3	23	0.016528	5.81E-05	38	0.023067		7.08E-05	160	0.168809 7.92E-05
	x4	24	0.016009	6.09E-05	39	0.02537	9.43E-05	159	0.166845	8.7E-05
	x5	23	0.017041	8.22E-05	39	0.025231		7.01E-05	159	0.170209 8.51E-05
	x6	24	0.013357	7.79E-05	40	0.024802		8.45E-05	159	0.179851 8.47E-05
	x7	24	0.016955	7.13E-05	40	0.025583		7.73E-05	160	0.166983 7.74E-05
50000	x1	26	0.284344	8.81E-05	44	0.468508		7.58E-05	-	- -
	x2	27	0.297186	5.5E-05	44	0.469222		8.61E-05	-	- -
	x3	25	0.268456	8.8E-05	42	0.450222	8.5E-05	-	-	-
	x4	26	0.278951	9.21E-05	44	0.468853		7.92E-05	-	- -
	x5	26	0.286123	6.82E-05	43	0.460054		8.39E-05	-	- -
	x6	27	0.292626	6.48E-05	45	0.513792	7.1E-05	-	-	-
	x7	27	0.287828	5.94E-05	44	0.47603	9.29E-05	-	-	-
100000	x1	27	0.582064	6.85E-05	45	0.96428	7.51E-05	-	-	-
	x2	27	0.586279	7.78E-05	45	0.9641	8.52E-05	-	-	-
	x3	26	0.554868	6.84E-05	43	0.924496		8.41E-05	-	- -
	x4	27	0.614813	7.16E-05	45	0.954237		7.84E-05	-	- -
	x5	26	0.667372	9.65E-05	44	0.939732		8.31E-05	-	- -
	x6	27	0.587419	9.17E-05	46	0.98069	7.03E-05	-	-	-
	x7	27	0.573314	8.4E-05	45	0.95427	9.2E-05	-	-	-

Table 4: Numerical results of Problem 2

		DDSL			IDSL			MCGD		
		NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖
100	x1	9	0.00233	4.5E-05	31	0.00538	7.62E-05	11	0.009309	4.95E-05
	x2	8	0.001822	2.76E-05	22	0.003583	8.07E-05	13	0.008163	3.11E-05
	x3	12	0.00499	2.51E-05	36	0.005543	9.41E-05	14	0.005189	5.43E-05
	x4	9	0.001933	6.01E-05	27	0.00419.	7.41E-05	12	0.004766	2.93E-05
	x5	19	0.004243	7.44E-05	27	0.004135	9.82E-05	50	0.016974	9.23E-05
	x6	10	0.002106	8.64E-05	32	0.005313	7.9E-05	15	0.008504	7.52E-05
	x7	22	0.004414	8.6E-05	37	0.004701	8.35E-05	56	0.017513	9.64E-05
1000	x1	10	0.002664	2.85E-05	34	0.009687	8.26E-05	12	0.008843	4.3E-05
	x2	8	0.002357	8.72E-05	25	0.009012	8.75E-05	13	0.00942	9.84E-05
	x3	12	0.003225	7.93E-05	40	0.012323	7.14E-05	15	0.011108	5.02E-05
	x4	10	0.00314	3.8E-05	30	0.008506	8.03E-05	12	0.008777	9.25E-05
	x5	19	0.00455	6.69E-05	27	0.007473	8.83E-05	53	0.025709	9.88E-05
	x6	11	0.003263	5.46E-05	35	0.009546	8.57E-05	17	0.012442	7.54E-05
	x7	23	0.005515	6.66E-05	37	0.009939	8.08E-05	59	0.030565	8.76E-05
2000	x1	10	0.004971	4.03E-05	35	0.012794	8.18E-05	12	0.011999	6.08E-05
	x2	9	0.003319	2.47E-05	26	0.010414	8.66E-05	14	0.012521	4.76E-05
	x3	13	0.008255	2.24E-05	41	0.016502	7.07E-05	15	0.014285	7.1E-05
	x4	10	0.003766	5.38E-05	31	0.011927	7.95E-05	13	0.013266	4.5E-05
	x5	19	0.008212	6.25E-05	27	0.008205	8.66E-05	53	0.036777	9.74E-05
	x6	11	0.004325	7.73E-05	36	0.013046	8.48E-05	18	0.017248	8.41E-05
	x7	23	0.008695	6.91E-05	37	0.013568	9.88E-05	59	0.048304	9.88E-05
50000	x1	11	0.083511	4.03E-05	39	0.247138	9.82E-05	14	0.230363	7.67E-05
	x2	10	0.068093	2.47E-05	31	0.201364	7.28E-05	16	0.261177	9.09E-05
	x3	14	0.094492	2.24E-05	45	0.287329	8.49E-05	18	0.312119	7.95E-05
	x4	11	0.076223	5.38E-05	35	0.225824	9.55E-05	15	0.249004	8.72E-05
	x5	18	0.12158	7.42E-05	27	0.179076	8.44E-05	54	0.745097	8.09E-05
	x6	12	0.083701	7.73E-05	41	0.270302	7.13E-05	19	0.320412	6.85E-05
	x7	23	0.15932	8.25E-05	41	0.261467	9.1E-05	60	0.851692	8.05E-05
100000	x1	11	0.156349	5.69E-05	40	0.495755	9.72E-05	16	0.524545	7.82E-05
	x2	10	0.129462	3.49E-05	32	0.395405	7.21E-05	18	0.600452	1.92E-05
	x3	14	0.183824	3.17E-05	46	0.574214	8.4E-05	19	0.615717	1.67E-05
	x4	11	0.147754.	7.6E-05	36	0.445583	9.45E-05	16	0.524639	9.65E-05
	x5	18	0.23887	6.83E-05	27	0.337565	8.43E-05	55	1.490285	8.22E-05
	x6	13	0.172011	2.19E-05	42	0.516052	7.06E-05	19	0.62121	9.68E-05
	x7	23	0.289727.	8.6E-05	42	0.518661	8.86E-05	60	1.693632	8.09E-05

Table 5: Numerical results of Problem 3

		DDSL			IDSL			MCGD		
		NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖
100	x1	9	0.00364	5.15E-05	31	0.005018	7.07E-05	10	0.007256	7.35E-05
	x2	9	0.004069	4.72E-05	19	0.004239	9.34E-05	10	0.007981	3.89E-05
	x3	9	0.001819	2.78E-05	35	0.007662	7.2E-05	12	0.004759	6.57E-05
	x4	9	0.001963	5.1E-05	29	0.004049	9.18E-05	10	0.006381	3.93E-05
	x5	9	0.002251	4.46E-05	33	0.005449	9.49E-05	15	0.010947	3.85E-05
	x6	9	0.003677	2.49E-05	29	0.003919	7.12E-05	10	0.004153	8.96E-05
	x7	9	0.002855	4.17E-05	29	0.005077	7.32E-05	6	0.003945	5.8E-05
1000	x1	10	0.003675	3.25E-05	34	0.013140	7.67E-05	12	0.012441	8.02E-05
	x2	10	0.003423	2.98E-05	23	0.010396	7.09E-05	11	0.013115	6.68E-05
	x3	9	0.003907	8.79E-05	38	0.009726	7.81E-05	15	0.011736	3.61E-05
	x4	10	0.003717	3.21E-05	32	0.011048	9.96E-05	11	0.012016	6.22E-05
	x5	10	0.006543	2.75E-05	37	0.012118	7.51E-05	12	0.009357	8.4E-05
	x6	9	0.003037	7.87E-05	32	0.011658	7.73E-05	14	0.013979	4.38E-05
	x7	10	0.0033	2.52E-05	31	0.011087	8.8E-05	6	0.012169	6.44E-05
2000	x1	10	0.008353	4.59E-05	35	0.015900	7.59E-05	13	0.017096	8.92E-05
	x2	10	0.004442	4.21E-05	24	0.010313	7.02E-05	11	0.015081	9.44E-05
	x3	10	0.004428	2.48E-05	39	0.014924	7.73E-05	15	0.019097	5.1E-05
	x4	10	0.006238	4.54E-05	33	0.012296	9.86E-05	11	0.020361	8.79E-05
	x5	10	0.008252	3.88E-05	38	0.016809	7.45E-05	14	0.017525	8.65E-05
	x6	10	0.006497	2.22E-05	33	0.012928	7.65E-05	14	0.017757	6.19E-05
	x7	10	0.007916	3.56E-05	32	0.014994	8.57E-05	8	0.012718	3.29E-05
50000	x1	11	0.085827	4.58E-05	39	0.278389	9.11E-05	14	0.275254	7.75E-05
	x2	11	0.089163.	4.2E-05	28	0.201013	8.43E-05	14	0.268456	6.53E-05
	x3	11	0.08644	2.47E-05	43	0.303875	9.29E-05	16	0.311202	9.81E-05
	x4	11	0.084851	4.53E-05	38	0.271259	8.28E-05	14	0.267068	4.81E-05
	x5	11	0.086475	3.85E-05	42	0.309995	8.98E-05	16	0.309297	4.84E-05
	x6	11	0.083897	2.21E-05	37	0.264575	9.18E-05	16	0.308864	9.91E-05
	x7	11	0.086029	3.53E-05	37	0.264988	7.1E-05	15	0.287802	3.74E-05
100000	x1	11	0.161917	6.47E-05	40	0.566583	9.02E-05	15	0.566334	3.17E-05
	x2	11	0.160816	5.94E-05	29	0.398891	8.35E-05	14	0.525767	9.24E-05
	x3	11	0.157884	3.49E-05	44	0.599958	9.19E-05	17	0.643667	7.86E-05
	x4	11	0.162063	6.4E-05	39	0.547434	8.2E-05	14	0.528784	6.8E-05
	x5	11	0.163694	5.45E-05	43	0.589709	8.89E-05	16	0.598712	6.85E-05
	x6	11	0.163389	3.13E-05	38	0.527089	9.09E-05	18	0.676396	1.81E-05
	x7	11	0.16761	4.99E-05	38	0.52088.	7.03E-05	15	0.550873	7.98E-05

Table 6: Numerical results of Problem 4

		DDSL			IDSL			MCGD		
		NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖	NI	TIME	‖F (xk) ‖
100	x1	13	0.021336	9.69E-05	30	0.008626	8.26E-05	900	0.246065	5.42E-05
	x2	12	0.007279	5.1E-05	27	0.007261	9.63E-05	426	0.189515	5.28E-05
	x3	15	0.008706	7.97E-05	33	0.006674	8.5E-05	910	0.253295	8.12E-05
	x4	13	0.00653	6.18E-05	29	0.006529	9.44E-05	985	0.253201	5.24E-05
	x5	18	0.011575	5.52E-05	34	0.00944	9.26E-05	730	0.193783	9.11E-05
	x6	12	0.003037	7.99E-05	28	0.007349	8.43E-05	655	0.180629	5.61E-05
	x7	16	0.007131	8.73E-05	27	0.00672	8.44E-05	58	0.021541	8.6E-05
1000	x1	17	0.008372	9.57E-05	33	0.013016		9.02E-05	- -	-
	x2	16	0.007382	5.04E-05	31	0.014239		7.36E-05	- -	-
	x3	19	0.009179	7.87E-05	36	0.014404		9.29E-05	- -	-
	x4	17	0.010357	6.11E-05	33	0.013243		7.21E-05	- -	-
	x5	19	0.00713	4.02E-05	35	0.017575		8.96E-05	- -	-
	x6	16	0.006828	7.9E-05	31	0.012037		9.21E-05	- -	-
	x7	16	0.007204	8.95E-05	27	0.011008		8.43E-05	- -	-
2000	x1	19	0.011844	3.63E-05	34	0.02008	9.01E-05	-	-	-
	x2	17	0.010978	6.37E-05	32	0.018178		7.33E-05	- -	-
	x3	20	0.012297	9.93E-05	37	0.020296		9.27E-05	- -	-
	x4	18	0.011337	7.71E-05	34	0.018731		7.18E-05	- -	-
	x5	20	0.013027	4.44E-05	36	0.020001		8.82E-05	- -	-
	x6	17	0.010128	9.96E-05	32	0.01747	9.19E-05	-	-	-
	x7	16	0.010714	8.97E-05	27	0.015713		8.43E-05	- -	-
50000	x1	24	0.261458	5.03E-05	39	0.416273		9.61E-05	- -	-
	x2	22	0.237088	7.36E-05	37	0.4047	7.7E-05	-	-	-
	x3	25	0.274684	9.7E-05	42	0.447796		9.92E-05	- -	-
	x4	23	0.25077	8.31E-05	39	0.460268		7.53E-05	- -	-
	x5	25	0.274757	5.75E-05	41	0.454622	9.4E-05	-	-	-
	x6	22	0.240391	9.72E-05	37	0.403697		9.83E-05	- -	-
	x7	16	0.174784	8.99E-05	27	0.292641		8.43E-05	- -	-
100000	x1	24	0.499683	8.94E-05	40	0.815062	9.9E-05	-	-	-
	x2	23	0.497906	6.3E-05	38	0.794383		8.02E-05	- -	-
	x3	26	0.534909	8.05E-05	44	0.905854		7.04E-05	- -	-
	x4	24	0.493499	7.01E-05	40	0.820122		7.86E-05	- -	-
	x5	25	0.529229	9.9E-05	42	0.859169	9.7E-05	-	-	-
	x6	23	0.48736	8.07E-05	39	0.799749		6.98E-05	- -	-
	x7	16	0.341215	8.99E-05	27	0.555021		8.43E-05	- -	-

The numerical results of the three methods are reported in Tables (3-6), where ”NI” and ”Time” stand for the total number of all iterations and the CPU time in seconds, respectively, while ‖F (xk)‖ is the norm of the residual at the stopping point. From Tables (3-6), We can quickly note that both of these approaches are attempting to solve nonlinear equation systems (1), but the improved efficiency and efficacy of our proposed algorithm was obvious because it solves where MCGD fails This is quite obvious, for example, with the problems 1 and 4. The DDSL approach significantly outperforms the IDSL and MCGD methods substantially for nearly all the test problems examined, because it has the fewer number of iterations and CPU time, which are less than that of IDSL and MCGD methods. This is apparently due to the computation of double direction and step length in each iteration of the DDSL method. We generate Figures (1-2) using the Dolan and Mor*e'* [20-22] performance profile in order to show the performance of each of the three methods. This is done by plotting the fraction p(τ) of the problems for which each method is within τ of the smallest number of iterations and CPU time respectively. As can be seen in Figures 1 and 2, the curves corresponding to the DDSL method remain above the other curves representing the IDSL and MCGD methods as for the number of iteration and CPU time. This indicates that the proposed method outperforms the IDFDD and IDSL in terms of fewer number of iteration and is therefore the most efficient method. Finally, from the results in Tables (3-6) it is obviously that our approach is very successful in solving large-scale nonlinear problems.

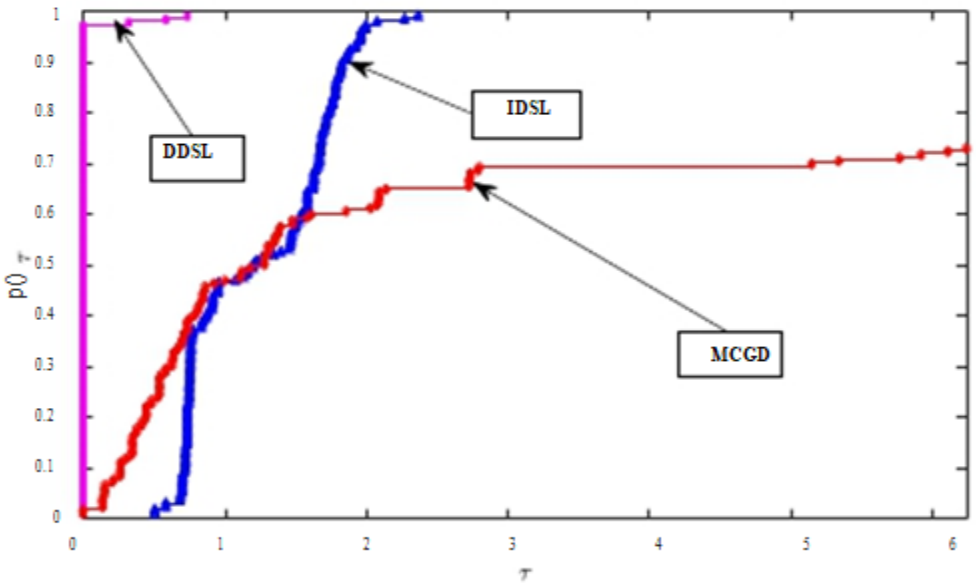

Figure 1: Performance profile with respect to the number of iterations

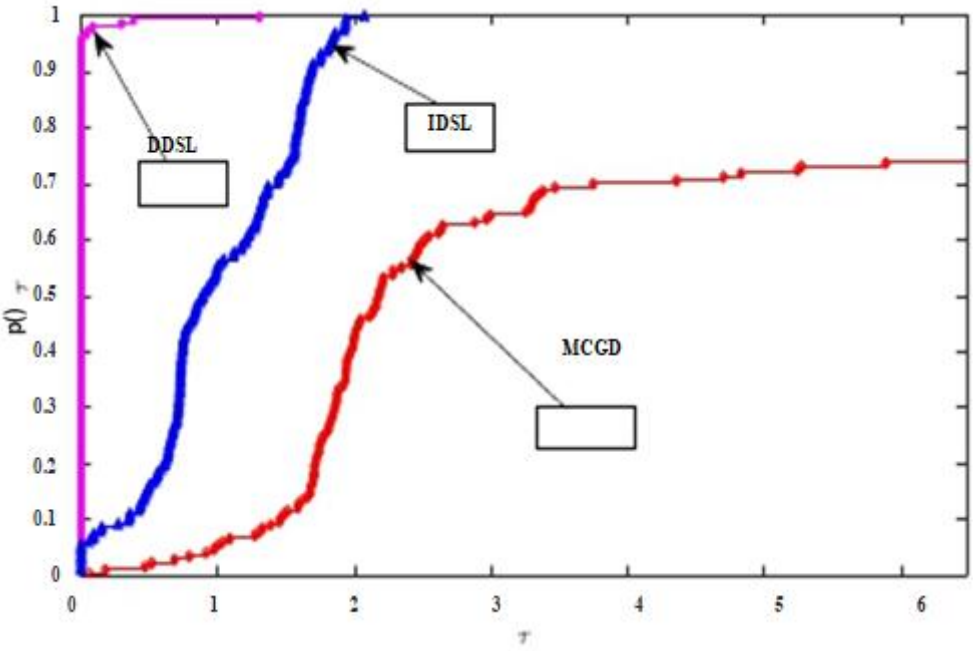

Figure 2: Performance profile with respect to the CPU time (in second)

4 Conclusion

In this paper, a double direction and step length method for solving system of nonlinear equations is presented. This was achieved by modifying the method in [9] as well as approximating the Jacobian matrix via acceleration parameter. The proposed method is completely derivative-free iterative method. Numerical comparisons have been made using a set of large-scale test problems. Moreover, Table (3-6) and Figure (1-2), showed that the proposed method is practically quite efficient because it has the least num- ber of iteration compared to IDSL and MCGD methods. In addition, the proposed method is successfully solve the discretized Chandrasekhar H-equation problem arising in the heat transfer. In the future research, the idea proposed in this scheme will be ap- plied to solve the monotone nonlinear equations with application in compressive sensing.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] 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, 2017; 6(4): 147-156.
[2] N.I. Duranovic-milicic, ” A multi step curve search algorithm in nonlinear optimization”, Yugoslav Journal of operation research, 2008; 18: 47-52.
[3] M.Y. Waziri, W.J. Leong, and M.A. Hassan, ”Jacobian-Free Diagonal Newtons Method for Solving Nonlinear Systems with Singular Jacobian”. Malaysian Journal of Mathematical Science 2011; 5: 241-255.

[4] N. Andrei, ” An accelerated gradient descent algorithm with backtracking for unconstrained optimization”, Numerical algorithm, 2006; 42: 63-73.
[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, 2005: 164(1): 277-289. [6] N.I Duranovic-milicic and M. Gardasevic-Filipovic,” A multi step curve search algo- rithm in nonlinear convex case”, Facta universitatis series:mathematics and informat- ics, 2010; 25: 11-24. [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, 2017; 9(1): 20-23.
[8] H. Abdullahi, A.S. Halilu, and Waziri M. Y. ”A Modified Conjugate Gradient Method via a Double Direction Approach for solving large-scale Symmetric Nonlinear Systems”,Journal of Numerical Mathematics and Stochastics, 2018; 10(1): 32-44.
[9] A.S. Halilu and M.Y. Waziri, ”Inexact Double Step Length Method for Solving Systems of Nonlinear Equations. Stat., Optim. Inf. Comput.”, 2020; 8: 165174.
[10] M.Raydan, ” On Barzilai and Borwein choice of step length for the gradient method”, IMA Journal of Numerical Analysis, 13: 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 Anal- ysis, 1999; 37: 152-172.
[12] M.Y. Waziri and J. Sabiu,” A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equations”, Journal of mathematics and mathematical sciences, 2015, 1-8.
[13] M.J. petrovic, P.S. Stanimirovic, ”Accelerated double direction method for solving unconstrained optimization problems”, Mathematical problem Eng., 2014.
[14] Gongli Yuan*, Xiwen Lu,” A new backtracking inexact BFGS method for symmetric nonlinear equations”, Journal of computers and mathematics with applications, 2008; 55: 116-129.
[15] Halilu, A.S., Waziri M.Y., and I.yusuf, ”Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations”, Yugoslav Journal of Operations Research. DOI:https://doi.org/10.2298/YJOR160515005H.
[16] M.J. petrovic, ”An accelerated double step size model in unconstrained optimiza- tion”, Journal of mathematics and computation, 2015; 250: 309-319.
[17] J.E. Dennis, Jr and R.B. Schnabel, ”Numerical method for unconstrained optimiza- tion and non-linear equations”, practice Hall, Englewood cliffs, NJ, USA, 1983.
[18] A.S. Halilu and M.Y. Waziri, ”An improved derivative-free method via double direc- tion approach for solving systems of nonlinear equations”, Journal of the Ramanujan Mathematical Society, 2018; 33: 75-89.
[19] Q.R. Yana, X.Z. Penga, D.H. Li, A globally convergent derivative-free method for solving large-scale nonlinear monotone equations. Journal of Computational and Applied Mathematics, 2010; 234: 649-657.
[20] E.Dolan and J.Mor´*e*, ”Benchmarking optimization software with performance pro- files”, Journal of Mathematical programming, 2002; 91(2): 201-213.
[21] M.Y. Waziri, W.J. Leong, M.A. Hassan, and M. Monsi, ”A low memory solver for integral equations of chandrasekhar type in the radiative transfer problems” 2011; 1-12.

[22] M. Sun, M.Y. Tian, Y.J. Wang, ”Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization”, Discrete Dyn. Nat. Soc. Article ID 4745759, 2019; 114.
[23] Kant, N., Wani, M. A., & Kumar, A. (2012). Self-focusing of Hermite–Gaussian laser beams in plasma under plasma density ramp. *Optics Communications*, *285*(21-22), 4483-4487.
[24] K. Meintjes, A.P.Morgan, ”A methodology for solving chemical equilibrium sys- tems”’ Appl. Math. Comput. 1987; 22: 333361.