Malaysian Journal of Mathematical Sciences 16(2): 329–349 (2022)
https://doi.org/10.47836/mjms.16.2.10

Malaysian Journal of Mathematical Sciences

Journal homepage: https://einspem.upm.edu.my/journal

On the Hybridization of the Double Step Length Method for Solving System of Nonlinear Equations

Halilu, A. S. [1,2,4], Waziri, M. Y.[3,4], Abdullahi, A.[1,2,4], and Majumder, A.*[1]

[1]*Department of Mathematics, Lovely Professional University, India -144411*
[2]*Department of Mathematics, Sule Lamido University, Nigeria*
[3]*Department of Mathematical Sciences, Bayero University, Nigeria*
[4]*Numerical Optimization Research Group, Bayero University, Nigeria*

E-mail: am.arunavamajumder@gmail.com
**Corresponding author*

Received: 15 December 2020
Accepted: 16 February 2022

Abstract

A hybrid derivative-free double step length technique is proposed in this work in order to enhance the numerical results and convergence properties of the double direction and step length scheme. This is accomplished by combining a Picard-Mann hybrid iterative method proposed by Khan [Fix Point Theory and Applications, pp. 1-10, vol.69 (2013)] with the double step length approach. A derivative line search is employed in order to compute the two step lengths. Furthermore, a suitable acceleration parameter is developed to approximate the Jacobian matrix. Under some mild conditions, the proposed method is shown to converge globally. The numerical experiment presented in this paper illustrates the efficiency of the proposed method over some existing methods.

Keywords: acceleration parameter; double direction and step length; global convergence; hybridization; Jacobian matrix.

1 Introduction

Many researchers in the fields of sciences, engineering, and other relevant areas try to achieve results with models in the form of a system of nonlinear equations

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

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

Moreover, (1) can be obtained from an unconstrained optimization problem [9] Let f be a merit function defined by

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

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

$$\min f(a), \quad a \in \mathbb{R}^n. \tag{3}$$

The study of such mappings is used in many scientific fields, such as economic [18] and chemical [10] equilibrium systems. It has practical application in Chandrasekhar H-equation that arises in the theory of radiative heat transfer in nonlinear integral equation [19] as well. Newton [4] and quasi-Newton [20] methods are among the iterative methods used to solve these problems. Their pertaining iterative procedure is given by

$$a_{k+1} = a_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, ..., \tag{4}$$

where α_k is a step length, a_{k+1} represents a new iteration, a_k is the previous iteration, while d_k is the search direction.

Newton's method is very welcome because of its nice properties such as rapid convergence rate from a reasonably good starting and Newton's search direction d_k is given by

$$d_k = -(F'_k)^{-1} F_k, \tag{5}$$

where F'_k is the Jacobian matrix of F_k at a_k. However, in Newton's method, the Jacobian matrix and its inverse are computed at each iteration, which invokes the first-order derivative of the system. It is well known that the computation of some function derivatives is not always available or cannot be obtained precisely in practice. In this case, Newton's method cannot be applied directly. For this reason, quasi-Newton's methods were developed to replace the Jacobian matrix or its inverse with an approximation which can be updated at each iteration [9, 20, 3], and its search direction is given by

$$d_k = -B_k^{-1} F_k,$$

where B_k is $n \times n$ matrix that approximate the Jacobian of F at x_k. In addition, the most outstanding class of quasi-Newton update B_k needs to satisfy the secant equation

$$B_k s_{k-1} = y_{k-1},$$

where, $y_{k-1} = F_k - F_{k-1}$ and $s_{k-1} = a_k - a_{k-1}$. In addition, the search direction d_k is usually required to satisfy the descent condition

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

If d_k is a descent direction of f at a_k, then inequality

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

holds for all $\alpha_k > 0$ sufficiently small.

A suitable step length α_k is needed for the methods to converge globally. The step length α_k can also be computed either exact or in exact. The best line search rule is the exact one [21] that satisfies

$$f(a_k + \alpha_k d_k) = \min_{\alpha > 0} f(a_k + \alpha d_k). \tag{7}$$

By using the iterative procedure (4), the step length requirement is to sufficiently decrease the function value to achieve global convergence. However, in practical computation, determining the exact step length is difficult, if not impossible. As a result, inexact line search is the most commonly used line search in practice. Brown and Saad [2] proposed the following inexact line search rule to obtain the step length α_k

$$f(a_k + \alpha_k d_k) - f(a_k) \leq \sigma \alpha_k \nabla f(a_k)^T d_k, \tag{8}$$

where $\sigma \in (0, 1)$. It is clear from the technique in (8), that the Jacobian matrix must be computed at each iteration, which increases the computing difficulty, especially for large-scale problems or when the matrix is expensive to compute. Consequently, Yuan* and Lu [20] present a new backtracking inexact technique for obtaining the step length α_k as follows:

$$\|F(a_k + \alpha_k d_k)\|^2 \leq \|F(a_k)\|^2 + \delta \alpha_k^2 F(a_k)^T d_k, \tag{9}$$

where $\delta \in (0, 1)$. Another derivative-free line search is proposed by Li and Fukushima in [9], which computes the step length α_k as follows:
Let $\omega_1 > 0$, $\omega_2 > 0$ and $r \in (0, 1)$ be constants and let $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty, \tag{10}$$

$$f(a_k + \alpha_k d_k) - f(a_k) \leq -\omega_1 \|\alpha_k F(a_k)\|^2 - \omega_2 \|\alpha_k d_k\|^2 + \eta_k f(a_k). \tag{11}$$

Let i_k is the smallest non negative integer i such that (11) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

The paper is organized as follows. The literature review is presented in Section 2. Assumptions and Notations are presented in Section 3. In Section 4, the algorithm of the proposed method is presented. In Section 5, some numerical results are reported. The discussion is presented in Section 6, and the conclusion is made in Section 7.

2 Literature Review

Despite the appealing characteristics of the Newton and quasi-Newton's methods, they require the Jacobian computation or its approximation at each iteration. Thus, they are not ideal for solving large-scale problems. Due to this shortcoming, the idea of the double direction approach is presented by Duranovic-Milicic, in [11], via

$$a_{k+1} = a_k + \alpha_k d_k + \alpha_k^2 b_k, \tag{12}$$

where, b_k and d_k are search directions respectively. Nonetheless, Duranović et al. [12] also proposed the algorithm for minimizing non-differentiable function using the double direction approach. Motivated by the work in [12], Petrović and Stanimirović, [15] proposed a double direction method for unconstrained optimization problems. In their work, an approximate Hessian matrix is obtained via acceleration parameter γ_k i.e.

$$\bigtriangledown^2 f(a_k) \approx \gamma_k I, \tag{13}$$

where $\bigtriangledown^2 f(x_k)$ is the Hessian matrix and $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The interesting feature of the scheme in [15] is that the two derivative-free directions were presented in the scheme, where the first direction approximated the Hessian with diagonal matrix through acceleration parameter. In contrast, the second one is a derivative-free direction, allowing the method to solve large-scale problems. However, the study of double direction methods for solving systems of nonlinear equations is infrequent in the literature. This motivated Halilu and Waziri [7] to use the scheme in [15] and proposed a derivative-free method via double direction approach for solving a system of nonlinear equations. In their work, the Jacobian matrix is approximated via acceleration parameter $\gamma_k > 0$, i.e.,

$$F'_k \approx \gamma_k I, \tag{14}$$

where I is an identity matrix. However, In [1], Abdullahi et al. modified the idea in [7] in solving the conjugate gradient approach, the method converged globally using the derivative-free line search proposed in [9]. The rationale behind the double direction method is that there are two corrections in the scheme (12); if one correction fails during the iterative process, then the second one will correct the system. To improve the performance of the scheme in [15] numerically, Petrović, presented the accelerated double step size model for unconstrained optimization [13], and its iterative scheme is given by

$$a_{k+1} = a_k + \alpha_k d_k + \beta_k c_k, \tag{15}$$

is considered in this work. Here α_k and β_k denote the step lengths, and d_k and c_k generate search directions [13].

The numerical results indicated that the method in [13], performed better than the double direction method [15], because of the contribution of the double step length scheme used in [13]. In [6], Halilu and Waziri also incorporated the idea of in (15), and proposed the transformed double step length method for solving system of nonlinear equations to enhance the numerical performance of double direction method in [7]. The numerical results presented in their work showed that the method in [6] converged faster than [7]. Furthermore, Motivated by the work in [6], recently, Halilu and Waziri presented an inexact double step length approach in [8]. The extensive numerical experiments showed that the method performed exceptionally well by comparing it with the existing method in the literature. The attractive feature of the technique in [8] is that it has double step length and single direction.

Petrović et al. [16] improve the convergence properties, and numerical results of the double direction method in [15] by hybridizing the scheme with Picard-Mann hybrid iterative process proposed by Khan in [17]. The Picard-Mann hybrid iterative process is defined as three relations:

$$\begin{cases} a_1 = a \in \mathbb{R}^n, \\ v_k = (1-\eta_k)a_k + \eta_k T(a_k), \\ a_{k+1} = T(v_k), \quad k \in \mathbb{N}, \end{cases} \tag{16}$$

where $T : \Omega \longrightarrow \Omega$ is a mapping defined on nonempty convex subset Ω of a normed space $\mathbf{E}$, a_k and v_k are sequences determined by the iteration (16), and $\{\eta_k\}$ is the sequence of positive numbers in $(0, 1)$. Following that, in [14], the hybridization rule in [17] is applied to the double step length scheme in order to improve the convergence properties of the double step length method in [13].

Based on the literature reviewed above, it can be concluded that using the hybridization process is an excellent way to improve the convergence properties and numerical experiments of some existing methods. Therefore, motivated by the idea presented by Petrović in [14] and the approximation in (14), this paper aims to develop a hybrid double direction and step length method for solving a system of nonlinear equations via

$$F'_k \approx \theta_k^{-1}\gamma_k I,$$

where, θ_k and γ_k are the correction and acceleration parameters, respectively.

The following research questions might be posed by someone.

Why double direction and step length?

The double direction and step length are because most methods for solving nonlinear equations are single-direction methods. One disadvantage of these methods is that they only have one correction in their iterative scheme, so if the correction fails, the system will collapse. The rational behind the double direction and step length approaches are that the scheme contains two corrections. The second will correct the system if one fails during the iterative process.

Table 1: Authors' contribution table.

Author's Name	Derivative-free	Matrix-free	Double Direction	Double step length	Hybridization	System of nonlinear equations
Dennis and Schnabel [4]	no	no	no	no	no	no
Li and Fukushima [9]	yes	no	no	no	no	yes
Yuan and Xiwen [20]	yes	no	no	no	no	yes
Waziri et al. [19]	yes	no	no	no	no	yes
Duranovic [11]	yes	yes	yes	no	no	no
Halilu and Waziri [6]	yes	yes	yes	no	no	yes
Duranovic and Filipovic [12]	yes	yes	yes	no	no	no
Halilu and Waziri [7]	yes	yes	yes	no	no	yes
Petrovic and Stanimirovic [15]	yes	yes	yes	no	no	no
petrovic [13]	yes	yes	yes	yes	no	no
petrovic et al. [16]	yes	yes	yes	no	yes	no
petrovic [14]	yes	yes	yes	yes	yes	no
Abdullahi et al. [1]	yes	yes	yes	no	no	yes
Halilu and Waziri [8]	yes	yes	no	yes	no	yes
This article	yes	yes	yes	yes	yes	yes

The followings are some of the contributions of this paper.

- This paper presents a hybrid derivative-free double step length approach that approximated the Jacobian matrix via the acceleration parameter.
- The new search direction is proposed so that it satisfies the decent condition.
- The correction parameter is derived using the Picard-Mann iterative scheme to improve the convergence properties and numerical experiments of some existing double direction and step length methods.
- There are two corrections in the double direction and step length scheme. Therefore, if one correction fails during the iterative process, the second one will automatically correct the system.

3 Assumptions and Notations

Let us start by defining the level set

$$S = \{a | \|F(a)\| \leq \|F_0\|\}. \tag{17}$$

Assumption 1.

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

(2) F is continuously differentiable in some neighborhood say M of a^* containing S.

(3) The Jacobian of F is bounded and positive definite on M, i.e there exists a positive constants $H > h > 0$ such that

$$\|F'(a)\| \leq H \quad \forall a \in M, \tag{18}$$

and

$$h\|d\|^2 \leq d^T F'(a)d \quad \forall a \in M, d \in \mathbb{R}^n. \tag{19}$$

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

$$h\|d\| \leq \|F'(a)d\| \leq H\|d\| \quad \forall a \in M, d \in \mathbb{R}^n. \tag{20}$$

$$h\|a - b\| \leq \|F(a) - F(b)\| \leq H\|a - b\| \quad \forall a, b \in M. \tag{21}$$

In particular $\forall a \in M$ we have

$$h\|a - a^*\| \leq \|F(a)\| = \|F(a) - F(a^*)\| \leq H\|a - a^*\|, \tag{22}$$

where a^* stands for the unique solution of (1) in M. Since $\theta^{-1}\gamma_k I$ approximates $F'(a_k)$ along direction s_k, another assumption can be contemplated.

Assumption 2.
$\theta^{-1}\gamma_k I$ is a good approximation to F'_k, i.e.,

$$\|(F'_k - \theta^{-1}\gamma_k I)d_k\| \leq \epsilon\|F_k\| \tag{23}$$

where $\epsilon \in (0, 1)$ is a small quantity and $\theta \in (1, 2)$ is a correction parameter.

Notations.

- The space $\mathbb{R}^n$ denote the $n-$dimensional real space.
- $\|\cdot\|$ is the Euclidean norm.
- $F_k = F(a_k)$.
- $s_k = a_{k+1} - a_k$.
- $y_k = F_{k+1} - F_k$.
- $F'_k = F'(a_k)$.

4 Main Result

The computation of two step lengths α_k and β_k, as well as the derivation of the acceleration parameter, is presented in this section. Now, let the two directions, c_k and d_k in (15) to be defined as:

$$d_k = -\gamma_k^{-1} F_k, \tag{24}$$

$$c_k = -F_k, \tag{25}$$

so by putting (24) and (25) in to (15) the following equation is obtained

$$a_{k+1} = a_k - (\alpha_k + \beta_k \gamma_k)\gamma_k^{-1} F_k. \tag{26}$$

Now, from Taylor series expansion of the first order, the approximation of F_{k+1} is presented

$$F_{k+1} \approx F_k + F'(\delta)(a_{k+1} - a_k), \tag{27}$$

where the parameter $\delta \in [a_k, a_{k+1}]$,

$$\delta = a_k + \phi(a_{k+1} - a_k) = a_k + \phi(\alpha_k + \beta_k\gamma_k)d_k \quad 0 \leq \phi \leq 1. \tag{28}$$

By taking $\phi = 1$ in (28), $\delta = a_{k+1}$. Therefore,

$$F'(\delta) \approx \gamma_{k+1} I. \tag{29}$$

This approximation implies

$$y_k = \gamma_{k+1} s_k, \tag{30}$$

where, $y_k = F_{k+1} - F_k$, $s_k = (\alpha_k + \beta_k\gamma_k)d_k$. By multiplying both side of (30) by y_k^T, the acceleration parameter γ_{k+1} can be computed in the following way:

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \beta_k\gamma_k) y_k^T d_k}. \tag{31}$$

To present the hybrid type of derivative-free double direction method, the mapping T in (16) is redefined by $T(v_k) = v_k - (\alpha_k + \beta_k\gamma_k)\gamma_k^{-1}F_k$. By this definition and (16),

$$\begin{cases} a_1 = x \in \mathbb{R}^n, \\ v_k = (1-\eta_k)a_k + \eta_k T(a_k) = a_k - \eta_k(\alpha_k + \beta_k\gamma_k)\gamma_k^{-1}F_k, \\ a_{k+1} = T(v_k) = v_k - (\alpha_k + \beta_k\gamma_k)\gamma_k^{-1}F_k, \quad k \in \mathbb{N}. \end{cases} \tag{32}$$

From the second and third equations in (32) we obtain the second iterative scheme,

$$a_{k+1} = a_k - \theta_k(\alpha_k + \beta_k\gamma_k)\gamma_k^{-1}F_k, \tag{33}$$

where, $\theta_k = (\eta_k + 1)$ and $\eta_k \in (0, 1)$. Define $\eta = \eta_k$, so that $\theta = (\eta + 1) \in (1, 2)$ is a correction parameter. From (33). Therefore, the proposed search direction is defined as:

$$d_k = -\theta\gamma_k^{-1}F_k. \tag{34}$$

Finally, from (33) and (34), the general scheme is given as:

$$a_{k+1} = a_k + (\alpha_k + \beta_k\gamma_k)d_k. \tag{35}$$

Algorithm 1: Hybrid Double direction and step length method(HDDSL).

Input: Given x_0, $\gamma_0 = 1$, $\epsilon = 10^{-4}$, $\omega_1 > 0$, $\omega_2 > 0$ and $q, r \in (0, 1)$, set $k = 0$.
Step 1: Compute F_k.
Step 2: If $\|F_k\| \leq \epsilon$ then stop, else goto step 3.
Step 3: Compute search direction $d_k = -\theta\gamma_k^{-1}F_k$.
Step 4: Set $a_{k+1} = a_k + (\alpha_k + \beta_k\gamma_k)d_k$. Let $\alpha_k = r^{m_k}$ and $\beta_k = q^{m_k}$, with m_k being the smallest nonnegative integer m such that

$$f(a_k + \lambda_k d_k) - f(a_k) \leq -\omega_1\|\lambda_k F_k\|^2 - \omega_2\|\lambda_k d_k\|^2 + \eta_k f(a_k), \tag{36}$$

where $\lambda_k = \alpha_k + \beta_k\gamma_k$ and $\{\eta_k\}$ is a given positive sequence such that

$$\sum_{k=0}^{\infty}\eta_k < \eta < \infty. \tag{37}$$

Step 5: Compute F_{k+1}.
Step 6: Update $\gamma_{k+1} = \dfrac{y_k^T y_k}{(\alpha_k + \beta_k\gamma_k)y_k^T d_k}$.
Step 7: Set $k = k + 1$, and go to step 2.

5 Convergence Analysis

The global convergence of Algorithm 1 (HDDSL) is presented in this section.

Lemma 5.1. *Suppose that Assumption 2 holds and $\{a_k\}$ be generated by Algorithm 1. Then d_k is a descent direction for $f(a_k)$ at a_k i.e*

$$\nabla f(a_k)^T d_k < 0. \tag{38}$$

Proof. From (2), (23) and (34),

$$\begin{aligned} \nabla f(a_k)^T d_k &= F_k^T F_k' d_k \\ &= F_k^T[(F_k' - \theta^{-1}\gamma_k I)d_k - F_k] \\ &= F_k^T(F_k' - \theta^{-1}\gamma_k I)d_k - \|F_k\|^2, \end{aligned} \tag{39}$$

and by Cauchy-Schwarz,

$$\begin{aligned}\nabla f(a_k)^T d_k &\le \|F_k\| \|(F_k' - \theta^{-1}\gamma_k I) d_k\| - \|F_k\|^2 \\ &\le -(1-\epsilon)\|F_k\|^2.\end{aligned} \tag{40}$$

Hence, for $\epsilon \in (0,1)$ this lemma is true. By Lemma 5.1, we can deduce that the norm function $f(a_k)$ is a descent along d_k , which means that $\|F_{k+1}\| \le \|F_k\|$ is true. □

Lemma 5.2. *Suppose that Assumption 2 hold and $\{a_k\}$ be generated by Algorithm 1. Then $\{a_k\} \subset S$.*

Proof. By lemma 5.1, $\|F_{k+1}\| \le \|F_k\|$. Moreover, we have for all k.

$$\|F_{k+1}\| \le \|F_k\| \le \|F_{k-1}\| \le \ldots \le \|F_0\|.$$

This implies that $\{a_k\} \subset S$. □

Lemma 5.3. *Suppose that Assumption 1 holds $\{a_k\}$ is generated by Algorithm 1. Then there exists a constant $m > 0$ such that for all k.*

$$y_k^T s_k \ge h\|s_k\|^2. \tag{41}$$

Proof. By mean-value theorem and (19),

$$y_k^T s_k = s_k^T (F(x_{k+1}) - F(x_k)) = s_k^T F'(\zeta) s_k \ge h\|s_k\|^2,$$

where $\xi = x_k + \zeta(a_{k+1} - a_k)$, $\zeta \in (0,1)$. □

Using $y_k^T s_k \ge h\|s_k\|^2 > 0$, γ_{k+1} is always generated by the update formula (31). Therefore, $\gamma_{k+1} I$ inherits the positive definiteness of $\gamma_k I$. From Lemma 5.3 and (21), the following inequality holds,

$$\frac{y_k^T s_k}{\|s_k\|^2} \ge h, \qquad \frac{\|y_k\|^2}{y_k^T s_k} \le \frac{H^2}{h}. \tag{42}$$

Lemma 5.4. *Suppose that Assumption 1 holds and $\{a_k\}$ is generated by Algorithm 1. Then*

$$\lim_{k\to\infty} \|\lambda_k d_k\| = 0, \tag{43}$$

and

$$\lim_{k\to\infty} \|\lambda_k F_k\| = 0. \tag{44}$$

Proof. By (36) and for all $k > 0$,

$$\begin{aligned}\omega_2\|\lambda_k d_k\|^2 &\le \omega_1\|\lambda_k F_k\|^2 + \omega_2\|\lambda_k d_k\|^2 \\ &\le \|F_k\|^2 - \|F_{k+1}\|^2 + \eta_k\|F_k\|^2.\end{aligned} \tag{45}$$

By summing the above inequality,

$$\begin{aligned}\omega_2 \sum_{i=0}^{k} \|\lambda_i d_i\|^2 &\leq \sum_{i=0}^{k} \left(\|F_i\|^2 - \|F_{i+1}\|^2\right) + \sum_{i=0}^{k} \eta_i \|F_i\|^2 \\ &= \|F_0\|^2 - \|F_{k+1}\|^2 + \sum_{i=0}^{k} \eta_i \|F_i\|^2 \\ &\leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{k} \eta_i \\ &\leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{\infty} \eta_i.\end{aligned} \tag{46}$$

Thus, from level set and fact that $\{\eta_k\}$ satisfies (37), then the series $\sum_{i=0}^{\infty} \|\lambda_i d_i\|^2$ is convergent. This implies (43). By similar arguments as above but with $\omega_1 \|\lambda_k F(x_k)\|^2$ on the left hand side, (44) holds. □

Lemma 5.5. *Suppose that assumption 1 holds and $\{a_k\}$ is generated by Algorithm 1. Then there exists some positive constants m_2 such that for all $k > 0$,*

$$\|d_k\| \leq m_2, \tag{47}$$

Proof. From (21), (31), and (34)

$$\begin{aligned}\|d_k\| &= \left\| -\theta \frac{\lambda_{k-1} y_{k-1}^T d_{k-1}}{y_{k-1}^T y_{k-1}} F_k \right\| \\ &= \left\| -\theta \frac{y_{k-1}^T s_{k-1} F_k}{\|y_{k-1}\|^2} \right\| \\ &\leq \frac{\theta \|F_k\| \|s_{k-1}\| \|y_{k-1}\|}{h^2 \|s_{k-1}\|^2} \\ &\leq \frac{\theta \|F_k\| H \|s_{k-1}\|}{h^2 \|s_{k-1}\|} \\ &\leq \frac{\theta \|F_k\| H}{h^2} \\ &\leq \frac{\theta \|F_0\| H}{h^2}.\end{aligned} \tag{48}$$

Taking $m_2 = \frac{\theta \|F_0\| H}{h^2}$, (47) is obtained. □

Theorem 5.1. *Suppose that Assumption 1 holds and $\{a_k\}$ is generated by Algorithm 1. Assume further for all $k > 0$,*

$$\lambda_k \geq c \frac{|F_k^T d_k|}{\|d_k\|^2}, \tag{49}$$

where $c > 0$. Then,

$$\lim_{k \to \infty} \|F_k\| = 0. \tag{50}$$

Proof. From Lemma 5.5, (47) is obtained. Therefore by (43) and the boundedness of $\{\|d_k\|\}$,

$$\lim_{k\to\infty} \lambda_k \|d_k\|^2 = 0, \tag{51}$$

from (49) and (51), the following condition holds.

$$\lim_{k\to\infty} |F_k^T d_k| = 0. \tag{52}$$

On the other hand from (34),

$$F_k^T d_k = -\theta\gamma_k^{-1}\|F_k\|^2, \tag{53}$$

$$\begin{aligned} \|F_k\|^2 &= |-F_k^T d_k \theta^{-1}\gamma_k| \\ &= \theta^{-1}|\gamma_k||F_k^T d_k|. \end{aligned} \tag{54}$$

But,

$$\gamma_k^{-1} = \frac{\lambda_{k-1} y_{k-1}^T d_{k-1}}{\|y_{k-1}\|^2} = \frac{y_{k-1}^T s_{k-1}}{\|y_{k-1}\|^2} \geq \frac{h\|s_{k-1}\|^2}{\|y_{k-1}\|^2} \geq \frac{h\|s_{k-1}\|^2}{H^2\|s_{k-1}\|^2} = \frac{h}{H^2}.$$

Then,

$$|\gamma_k| \leq \frac{H^2}{h}. \tag{55}$$

However, from (54) and (55),

$$\|F_k\|^2 \leq |F_k^T d_k| \left(\frac{H^2}{\theta h}\right). \tag{56}$$

Therefore,

$$0 \leq \|F_k\|^2 \leq |F_k^T d_k| \left(\frac{H^2}{\theta h}\right) \longrightarrow 0. \tag{57}$$

Therefore

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{58}$$

The proof is completed. □

6 Numerical Results

In this section, some numerical results are 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) [8].
- An improved derivative-free method via double direction approach for solving systems of nonlinear equation(IDFDD) [7].

For the HDDSL Algorithm, the following parameters are set. $\omega_1 = \omega_2 = 10^{-4}$, $r = 0.2$ and $q = 0.3$. $\eta_k = \dfrac{1}{(k+1)^2}$. The parameters of IDSL and IDFDD algorithms are taken from [8] and [7] respectively.

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 (36) in the experiments. The iteration is set to stop for the three methods if $\|F_k\| \leq 10^{-4}$, or when the iterations exceed 1000 but no point of a_k satisfying the stopping criterion is obtained. The symbol '-' represents failure due to; (i) Memory requirement (ii) the Number of iterations exceeding 1000. To show the extensive numerical experiments of HDDSL, IDSL, and IDFDD methods, we have tried these methods on the previous eight Benchmark test problems with different initial points and dimensions (n values) between 100 to 10,000.

Table 2: Initial points.

INITIAL POINTS (IP)	VALUES
a1	$\left(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\right)^T$
a2	$\left(\frac{1}{5}, \frac{1}{5}, ..., \frac{1}{5}\right)^T$
a3	$\left(\frac{3}{2}, \frac{3}{2}, ..., \frac{3}{2}\right)^T$
a4	$\left(\frac{2}{5}, \frac{2}{5}, ..., \frac{2}{5}\right)^T$
a5	$\left(0, \frac{1}{2}, \frac{2}{3}, ..., 1 - \frac{1}{n}\right)^T$
a6	$\left(\frac{1}{4}, \frac{-1}{4}, ..., \frac{(-1)^n}{4}\right)^T$
a7	$\left(1, \frac{1}{2}, \frac{1}{3}, ..., \frac{1}{n}\right)^T$

1. Problem 1 [8]
   $F(a) = Ba + c_1$,
   where, $B = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$, and $c_1 = (e_1^a - 1, ..., e_n^x - 1)^T$.
2. Problem 2 [1]
   $F(a) = Ba + c_2$,
   where, $B = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$, and $c_2 = (\sin a_1 - 1, ..., \sin a_n - 1)^T$.
3. Problem 3[6]
   $F_1 = a_1 - e^{\cos\left(\frac{a_1+a_2}{n+1}\right)}$,
   $F_i = a_i - e^{\cos\left(\frac{a_{i-1}+a_i+a_{i+1}}{n+1}\right)}$,
   $F_n = a_n - e^{\cos\left(\frac{a_{n-1}+a_n}{n+1}\right)}$, $\quad i = 2, 3, ..., n-1$.
4. Problem 4 [8]
   $F_i(a) = (1 - a_i^2) + a_i(1 + a_i a_{n-2} a_{n-1} a_n) - 2, \quad i = 1, 2, ..., n.$

5. Problem 5
$F_i(a) = a_i - 3a_i\left(\frac{\sin a_i}{3} - 0.66\right) + 2, \quad i = 1, 2, ..., n.$

6. Problem 6: The discretized Chandrasehar H-equation (problem of integral equation arising in radiative heat transfer)
$$F_i(a) = x_i - \left(1 - \frac{c}{2n}\sum_{j=1}^{n}\frac{\mu_i a_j}{\mu_i + \mu_j}\right)^{-1}, \quad \text{i=1,2,...,n}, \quad \text{j=1,2,...,n}.$$
with $c \in [0, 1)$ and $\mu = \frac{i-0.5}{n}$. (In our experiment we take $c = 0.1$).

7. Problem 7 [7]
$F_i(a) = 2a_i - \sin|a_i|, \quad i = 1, 2, ..., n.$

8. Problem 8
$F_1(a) = a_1(a_1^2 + a_2^2) - 1,$
$F_i(a) = a_i(a_{i-1}^2 + 2a_i^2 + a_{i+1}^2) - 1$
$F_n(a) = a_n(a_{n-1}^2 + a_n^2). \quad i = 2, 3, ..., n - 1.$

Table 3: Numerical results of Problem 1.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
100	a1	19	0.086046	7.56E-05	21	0.039053	6.79E-05	44	0.119926	8.47E-05
	a2	17	0.09443	6.73E-05	20	0.054056	6.64E-05	38	0.107599	9.01E-05
	a3	23	0.068536	9.07E-05	22	0.056029	8.48E-05	52	0.132793	8.69E-05
	a4	19	0.073777	5.92E-05	23	0.077724	5.24E-05	43	0.124381	8.42E-05
	a5	21	0.078571	5.3E-05	21	0.069383	8.87E-05	47	0.136727	8.71E-05
	a6	18	0.053883	5.31E-05	21	0.067299	5.95E-05	39	0.10678	7.95E-05
	a7	17	0.088059	5.21E-05	20	0.048554	9.03E-05	36	0.100868	9.24E-05
1,000	a1	20	0.427984	5.11E-05	24	0.448418	8.33E-05	46	1.132894	8.26E-05
	a2	17	0.355284	6.2E-05	22	0.383085	9.86E-05	42	0.948192	8.66E-05
	a3	20	0.400495	8.62E-05	24	0.403832	6.94E-05	48	1.062128	9.58E-05
	a4	19	0.413405	9.63E-05	23	0.418959	8.24E-05	45	1.041173	7.73E-05
	a5	21	0.414048	7.36E-05	24	0.426779	7.09E-05	47	1.045997	8.3E-05
	a6	16	0.319878	8.07E-05	23	0.410682	5.48E-05	40	0.892642	9.23E-05
	a7	17	0.336022	5.2E-05	20	0.350069	9.53E-05	36	0.911509	9.3E-05
2,000	a1	20	1.511404	8.28E-05	24	1.512597	7.89E-05	47	3.65381	9.41E-05
	a2	18	1.326404	5.05E-05	23	1.429866	6.19E-05	44	3.521878	7.78E-05
	a3	21	1.499297	4.61E-05	24	1.461959	9.55E-05	51	4.101937	8.91E-05
	a4	19	1.397923	8.53E-05	24	1.432547	7.51E-05	46	3.726025	9.93E-05
	a5	20	1.427108	9.31E-05	24	1.489955	8.92E-05	52	4.742935	8.2E-05
	a6	16	1.119568	8.9E-05	23	1.430295	5.11E-05	42	3.436851	9.3E-05
	a7	17	1.281192	5.2E-05	20	1.232176	9.56E-05	36	3.241174	9.3E-05

Table 4: Numerical results of Problem 2.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
100	a1	18	0.062572	5.69E-05	16	0.043041	8.58E-05	26	0.078548	9.75E-05
	a2	21	0.059264	8.42E-05	14	0.042393	6.73E-05	34	0.098625	7.79E-05
	a3	24	0.074858	5.56E-05	17	0.047558	6.89E-05	37	0.117063	9.59E-05
	a4	19	0.056566	8.93E-05	14	0.041349	5.91E-05	31	0.094199	7.37E-05
	a5	23	0.072131	6.63E-05	16	0.050072	6.89E-05	35	0.143327	8.81E-05
	a6	24	0.065612	5.29E-05	16	0.050092	8.79E-05	36	0.095618	1E-04
	a7	23	0.084185	8.93E-05	15	0.045582	8.53E-05	35	0.097265	8.71E-05
1,000	a1	18	0.390192	5.9E-05	16	0.289141	5.79E-05	28	0.641491	7.86E-05
	a2	22	0.462563	9.25E-05	14	0.244885	9.32E-05	37	0.907733	9.47E-05
	a3	25	0.578908	8.99E-05	18	0.303107	7.86E-05	41	0.962955	8.17E-05
	a4	21	0.484974	6.63E-05	14	0.253508	6.48E-05	34	0.857877	8.92E-05
	a5	24	0.534088	8.86E-05	17	0.304339	6.28E-05	39	0.947689	7.89E-05
	a6	24	0.497849	7.18E-05	16	0.292897	6.32E-05	40	0.950097	8.78E-05
	a7	23	0.500307	8.76E-05	15	0.25549	8.39E-05	39	0.915183	8.07E-05
2,000	a1	20	1.534464	5.69E-05	15	0.938192	8.33E-05	29	2.448305	7.26E-05
	a2	22	1.62065	9.31E-05	15	0.912588	6.04E-05	38	3.268568	9.72E-05
	a3	25	1.996947	6.07E-05	19	1.156874	5.41E-05	42	3.544581	8.37E-05
	a4	21	1.659435	6.3E-05	14	0.883999	7.78E-05	35	3.010445	9.15E-05
	a5	24	1.879494	7.51E-05	17	1.041232	8.36E-05	40	3.329917	8.11E-05
	a6	26	2.000885	7.42E-05	16	0.971261	6.45E-05	41	3.415999	9.01E-05
	a7	24	1.914637	6.15E-05	15	0.951498	9.28E-05	40	3.329539	8.32E-05

Table 5: Numerical results of Problem 3.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
1,000	a1	5	0.005064	2.18E-05	20	0.029283	6.69E-05	88	0.039125	7.72E-05
	a2	5	0.004858	2.48E-05	20	0.016974	7.59E-05	88	0.041215	8.77E-05
	a3	5	0.005357	1.2E-05	19	0.016551	7.35E-05	85	0.041695	9.66E-05
	a4	5	0.004579	2.28E-05	20	0.009005	6.99E-05	88	0.045116	8.07E-05
	a5	5	0.005276	1.7E-05	20	0.008642	5.21E-05	87	0.037374	7.91E-05
	a6	5	0.002724	2.92E-05	20	0.023986	8.95E-05	89	0.044046	7.85E-05
	a7	5	0.005249	2.67E-05	20	0.011291	8.18E-05	88	0.04363	9.44E-05
10,000	a1	5	0.020228	6.93E-05	22	0.069044	5.29E-05	103	0.288239	8.13E-05
	a2	5	0.018162	7.87E-05	22	0.062359	6E-05	103	0.267648	9.23E-05
	a3	5	0.016448	3.81E-05	21	0.080194	5.81E-05	101	0.248799	7.73E-05
	a4	5	0.017735	7.24E-05	22	0.092328	5.53E-05	103	0.294591	8.49E-05
	a5	5	0.018758	5.37E-05	21	0.054243	8.2E-05	102	0.303128	8.29E-05
	a6	5	0.020606	9.28E-05	22	0.082501	7.08E-05	104	0.283596	8.27E-05
	a7	5	0.017448	8.49E-05	22	0.060351	6.48E-05	103	0.281549	9.96E-05
100,000	a1	6	0.143677	1.1E-05	23	0.651922	8.36E-05	107	2.957521	8.57E-05
	a2	6	0.170244	1.24E-05	23	0.742498	9.49E-05	107	2.58559	9.73E-05
	a3	6	0.131867	6.02E-06	22	0.647278	9.19E-05	105	2.523885	8.15E-05
	a4	6	0.136426	1.15E-05	23	0.625249	8.74E-05	107	2.552052	8.96E-05
	a5	6	0.162541	8.49E-06	23	0.652416	6.48E-05	106	2.538057	8.74E-05
	a6	6	0.168431	1.47E-05	24	0.682688	5.59E-05	108	2.578981	8.72E-05
	a7	6	0.172361	1.34E-05	24	0.666882	5.12E-05	108	2.611317	7.99E-05

Table 6: Numerical results of Problem 4.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|\|F_k\|\|$	NI	CT	$\|\|F_k\|\|$	NI	CT	$\|\|F_k\|\|$
1,000	a1	11	0.00601	4.87E-05	18	0.009278	6.45E-05	23	0.012334	8.87E-05
	a2	6	0.005123	2.81E-05	19	0.009113	8.3E-05	17	0.009321	6.99E-05
	a3	12	0.008914	5.84E-05	16	0.006937	9.39E-05	27	0.009924	8.55E-05
	a4	9	0.007446	6.69E-05	17	0.005503	6.44E-05	25	0.009966	7.36E-05
	a5	12	0.004589	6.02E-05	18	0.005172	8.98E-05	62	0.022277	9.21E-05
	a6	6	0.005202	2.77E-05	19	0.007668	5.68E-05	27	0.010335	9.17E-05
	a7	16	0.00982	5.78E-05	25	0.010771	5.29E-05	39	0.015002	7.82E-05
10,000	a1	12	0.039852	3.7E-05	20	0.037669	5.1E-05	26	0.067009	7.35E-05
	a2	6	0.019763	8.87E-05	21	0.042498	6.56E-05	19	0.055433	9.06E-05
	a3	13	0.037047	4.43E-05	18	0.049099	7.42E-05	30	0.073083	7.09E-05
	a4	10	0.032617	5.08E-05	19	0.036196	5.09E-05	27	0.058444	9.54E-05
	a5	10	0.032417	3.99E-05	17	0.042394	8.93E-05	61	0.122019	9.99E-05
	a6	6	0.017417	8.76E-05	20	0.043078	8.98E-05	30	0.065872	7.61E-05
	a7	17	0.066437	2.4E-05	26	0.047149	8.78E-05	40	0.07688	6.35E-05
100,000	a1	13	0.298914	2.81E-05	21	0.351706	8.07E-05	28	0.581697	9.52E-05
	a2	7	0.171232	6.74E-05	23	0.359819	5.19E-05	22	0.461627	7.51E-05
	a3	14	0.286959	3.36E-05	20	0.347148	5.87E-05	32	0.652686	9.18E-05
	a4	11	0.229944	3.85E-05	20	0.332691	8.05E-05	30	0.626244	7.91E-05
	a5	8	0.151097	3.88E-05	17	0.308093	7.34E-05	63	1.24419	9.63E-05
	a6	7	0.144951	6.65E-05	22	0.375993	7.1E-05	32	0.868389	9.85E-05
	a7	17	0.296909	6.33E-05	27	0.44873	8.87E-05	42	0.81673	8.33E-05

Table 7: Numerical results of Problem 5.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|\|F_k\|\|$	NI	CT	$\|\|F_k\|\|$	NI	CT	$\|\|F_k\|\|$
1,000	a1	10	0.008395	9.52E-05	21	0.009242	5.77E-05	30	0.012078	7.69E-05
	a2	10	0.007966	4.72E-05	21	0.007001	5.84E-05	29	0.012126	6.82E-05
	a3	11	0.006258	9.48E-05	19	0.008618	9.67E-05	32	0.018467	7.74E-05
	a4	10	0.006726	7.71E-05	21	0.008077	5.92E-05	30	0.017134	6.53E-05
	a5	11	0.005082	5.2E-05	20	0.007062	8.01E-05	31	0.014759	8.63E-05
	a6	9	0.006061	3.33E-05	20	0.010393	6.83E-05	25	0.012315	8.88E-05
	x7	10	0.006331	2.75E-05	21	0.006859	5.18E-05	28	0.011884	6.56E-05
10,000	a1	11	0.050923	7.24E-05	22	0.048906	9.12E-05	32	0.082389	9.96E-05
	a2	11	0.045862	3.59E-05	22	0.046715	9.23E-05	31	0.094621	8.84E-05
	a3	12	0.052182	7.21E-05	21	0.04662	7.65E-05	35	0.094544	6.42E-05
	a4	11	0.041051	5.86E-05	22	0.047237	9.36E-05	32	0.100286	8.45E-05
	a5	12	0.059072	3.98E-05	22	0.048057	6.29E-05	34	0.096634	7.19E-05
	a6	10	0.042105	2.53E-05	22	0.048966	5.4E-05	28	0.085704	7.37E-05
	a7	10	0.037627	8.07E-05	22	0.053777	8.16E-05	30	0.08902	8.15E-05
100,000	a1	12	0.340436	5.5E-05	24	0.402003	7.21E-05	35	0.697165	8.26E-05
	a2	12	0.296061	2.73E-05	24	0.41094	7.3E-05	34	0.811713	7.33E-05
	a3	13	0.292467	5.48E-05	23	0.39072	6.05E-05	37	0.745469	8.32E-05
	a4	12	0.315122	4.46E-05	24	0.413037	7.4E-05	35	0.695419	7.01E-05
	a5	13	0.26992	3.03E-05	23	0.388943	9.94E-05	36	0.757766	9.32E-05
	a6	10	0.222806	8E-05	23	0.381519	8.54E-05	30	0.624813	9.54E-05
	a7	11	0.255413	6.08E-05	24	0.521528	6.45E-05	33	0.702426	6.72E-05

Table 8: Numerical results of Problem 6.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
1,000	a1	9	0.007258	5.17E-05	17	0.01007	5.46E-05	–	0.288954	0.223257
	a2	9	0.004237	6.88E-05	15	0.009534	8.81E-05	–	0.327166	0.280343
	a3	10	0.009703	6.05E-05	18	0.012275	8.25E-05	–	0.310184	0.223573
	a4	9	0.007728	6.77E-05	16	0.011595	8.81E-05	–	0.346968	0.238486
	a5	9	0.007035	6.05E-05	18	0.010225	5.79E-05	–	0.37145	0.266009
	a6	9	0.007174	7.18E-05	16	0.007221	5.46E-05	–	0.278931	0.263308
	a7	7	0.005738	5.72E-05	14	0.005254	7.83E-05	124	0.047107	8.6E-05
10,000	a1	5	0.020753	8.68E-05	19	0.055901	4.98E-05	–	2.582012	0.019957
	a2	6	0.022018	7.78E-05	17	0.050239	9.03E-05	–	2.595433	0.02505
	a3	7	0.026238	7.06E-05	20	0.059733	8.28E-05	–	2.751142	0.019988
	a4	4	0.017327	7.38E-06	18	0.075335	9.03E-05	–	2.571422	0.021318
	a5	7	0.026686	7.16E-05	20	0.063516	4.97E-05	–	2.506555	0.023931
	a6	6	0.024686	6.89E-05	18	0.052401	4.98E-05	–	2.582045	0.02353
	a7	6	0.025518	6.94E-05	14	0.042693	7.83E-05	93	0.278512	8.63E-05
100,000	a1	5	0.135281	3.81E-05	21	0.569527	7.18E-05	–	21.0978	0.001773
	a2	5	0.135464	6.75E-06	20	0.458992	5.6E-05	–	21.29192	0.002229
	a3	6	0.173664	4.13E-05	23	0.550873	5.2E-05	–	21.38213	0.001775
	a4	5	0.148619	2.55E-05	21	0.499927	5.6E-05	–	21.21942	0.001894
	a5	5	0.146304	9.31E-05	22	0.657306	7.18E-05	–	21.22142	0.002131
	a6	5	0.148733	2.93E-06	20	0.439507	7.18E-05	–	21.11174	0.002093
	a7	4	0.140636	1.5E-05	14	0.320194	7.83E-05	87	2.142283	8.64E-05

Table 9: Numerical results of Problem 7.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
1,000	a1	4	0.003593	2.81E-05	18	0.008638	6.23E-05	44	0.013049	9.25E-05
	a2	4	0.003503	3.63E-05	16	0.006238	9.71E-05	41	0.011743	8.25E-05
	a3	7	0.004795	2.9E-05	19	0.007773	8.59E-05	49	0.020651	7.68E-05
	a4	4	0.002018	4.72E-05	17	0.006284	9.87E-05	43	0.012813	9.65E-05
	a5	7	0.002994	6.76E-06	19	0.007582	6.33E-05	47	0.017203	8.49E-05
	a6	6	0.003444	2.15E-05	17	0.008682	9.44E-05	43	0.021896	9.54E-05
	a7	5	0.003508	8.96E-05	14	0.004145	8.36E-05	35	0.012827	9.22E-05
10,000	a1	4	0.011784	8.89E-05	19	0.037319	9.85E-05	48	0.12004	9.76E-05
	a2	5	0.014811	5.75E-06	18	0.035711	7.68E-05	45	0.167162	8.7E-05
	a3	7	0.020568	9.17E-05	21	0.040635	6.79E-05	53	0.13007	8.1E-05
	a4	5	0.013503	7.46E-06	19	0.035666	7.81E-05	48	0.131741	7.74E-05
	a5	7	0.01663	2.18E-05	21	0.042724	5.03E-05	51	0.117967	9.02E-05
	a6	6	0.016562	6.8E-05	19	0.049162	7.47E-05	48	0.112577	7.65E-05
	a7	5	0.011228	8.96E-05	14	0.028872	8.37E-05	35	0.075278	9.23E-05
100,000	a1	5	0.111909	1.41E-05	21	0.278435	7.79E-05	53	0.876666	7.83E-05
	a2	5	0.120333	1.82E-05	20	0.261158	6.07E-05	49	0.849232	9.18E-05
	a3	8	0.133753	1.45E-05	23	0.278786	5.37E-05	57	1.10498	8.55E-05
	a4	5	0.087237	2.36E-05	21	0.270465	6.17E-05	52	0.84769	8.17E-05
	a5	7	0.166646	6.9E-05	22	0.268697	7.97E-05	55	0.946469	9.52E-05
	a6	7	0.117077	1.07E-05	21	0.262944	5.9E-05	52	0.876528	8.07E-05
	a7	5	0.088781	8.96E-05	14	0.172282	8.37E-05	35	0.562445	9.23E-05

Table 10: Numerical results of Problem 8.

		HDDSL			IDSL			IDFDD		
		NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$	NI	CT	$\|F_k\|$
1,000	a1	25	0.011827	9.9E-05	34	0.013644	7.48E-05	37	0.014598	9E-05
	a2	27	0.01083	6.68E-05	25	0.009974	7.11E-05	40	0.026014	9.76E-05
	a3	27	0.011541	8.05E-05	39	0.016629	7.12E-05	43	0.026974	9.2E-05
	a4	26	0.01425	9.68E-05	38	0.011414	7.34E-05	39	0.020335	9.14E-05
	a5	29	0.011346	7.81E-05	25	0.012756	9.85E-05	43	0.018004	8.57E-05
	a6	24	0.012609	9.85E-05	33	0.012674	9.77E-05	42	0.021794	8.07E-05
	a7	16	0.008257	6.42E-05	21	0.008738	9.99E-05	35	0.01418	9.44E-05
10,000	a1	26	0.063604	9.43E-05	36	0.067588	8.52E-05	39	0.107004	8.51E-05
	a2	27	0.071332	7.89E-05	28	0.057171	7.14E-05	42	0.109131	9.12E-05
	a3	27	0.06353	6.19E-05	40	0.075413	7.91E-05	44	0.103635	8.98E-05
	a4	27	0.089224	8.57E-05	32	0.059848	9.44E-05	41	0.101038	8.75E-05
	a5	29	0.06437	8.23E-05	24	0.047676	6.32E-05	45	0.127968	9.05E-05
	a6	25	0.084594	5.84E-05	33	0.0729	8.58E-05	44	0.113948	9.12E-05
	a7	16	0.048434	5.62E-05	22	0.049437	8.96E-05	35	0.083353	9.85E-05
100,000	a1	26	0.716189	7.9E-05	38	0.840042	9.79E-05	41	1.076758	9.29E-05
	a2	27	0.705306	7.4E-05	29	0.657375	9.07E-05	44	1.078258	8.59E-05
	a3	26	0.625941	7.7E-05	41	0.919964	7.96E-05	46	1.117782	8.29E-05
	a4	27	0.623825	6.65E-05	37	0.837999	5.5E-05	45	1.095316	6.94E-05
	a5	29	0.680253	7.89E-05	25	0.574306	9.75E-05	49	1.168136	6.71E-05
	a6	23	0.613712	8.12E-05	37	0.810282	9.19E-05	46	1.103121	7.91E-05
	a7	16	0.375065	6.98E-05	24	0.545617	6.98E-05	37	0.905213	8.79E-05

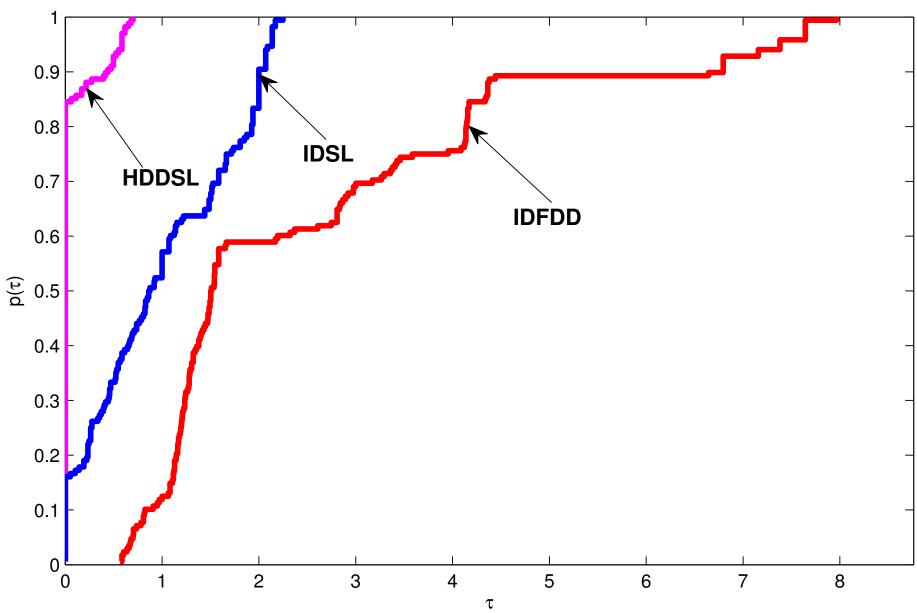

Figure 1: Performance profile for the number of iterations.

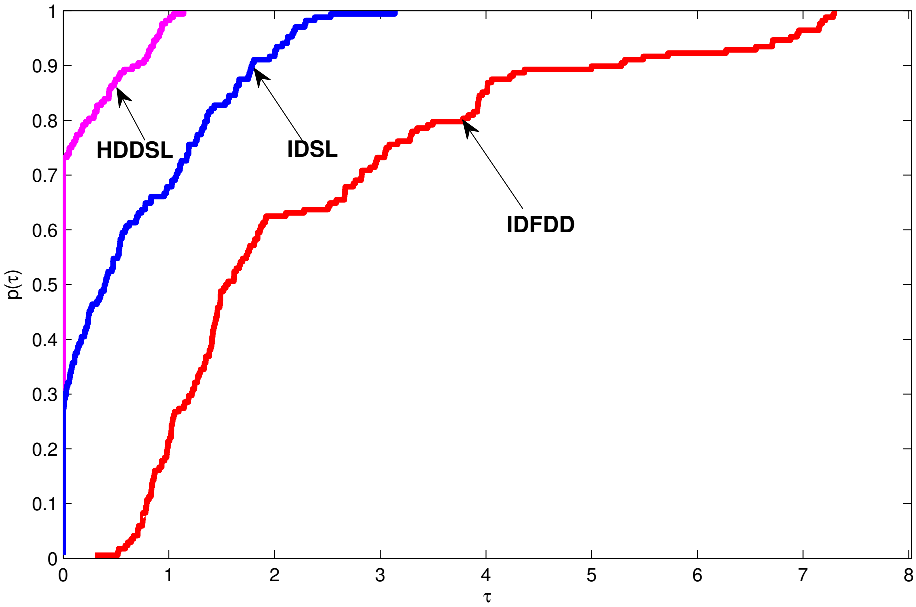

Figure 2: Performance profile for the CPU time (in seconds).

7 Results and Discussion

Tables (3-10) above reported the numerical results of the three methods, where "NI" and "CT" mean the number of iterations and the CPU time (in seconds), respectively. At the same time, $\|F_k\|$ is the norm of the residual at the stopping point. From the Tables (3-10), one can quickly note that all the three methods aim to solve a nonlinear system of equations (1). However, the effectiveness of the HDDSL algorithm was evident because it solves problems that IDFDD fails to solve (see Problems 6). Furthermore, the HDDSL method significantly outperforms the IDSL and IDFDD methods for nearly all the test problems examined. Because it has fewer iterations and CPU time, which are less than those of IDSL and IDFDD methods except for Problem 2, where the number of iterations and CPU time of the proposed method are greater than those of the IDSL method, this is due to the contribution of the correction parameter in each iteration of the HDDSL algorithm.

Figures (1-2) were created using the Dolan and Moré [5] performance profiles to demonstrate the performance of each of the three methods. For each problem $\rho \in \mathcal{P}$ and solver $s \in \mathcal{S}$, the performance profile is obtained in term of the performance measure $t_{\rho,s} > 0$. For any pair (ρ, s) of problem ρ and solver s, the performance ratio is given as

$$r_{\rho,s} = \frac{t_{\rho,s}}{\min\{t_{\rho,s}|s \in \mathcal{S}\}}. \tag{59}$$

The best solver for a particular problem reaches the lower bound $r_{\rho,s} = 1$. If a solver s fails to meet the convergence test for problem ρ, then $r_{\rho,s}$ is set to infinity. The performance profile of a solver s is defined as

$$p(\tau) = \frac{1}{n_\rho}\text{size}\{\rho \in \mathcal{P}|r_{\rho,s} \leq \tau\}, \tag{60}$$

where n_p is the number of problems. Therefore, $p(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance ratio $r_{\rho,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. However, the function P is the cumulative distribution function for the performance ratio. The fraction $p(\tau)$ of problems for which each method is within τ of the smallest number of iterations, CPU time, and function evaluations is plotted for this purpose.

Figures 1 and 2 show that the curves corresponding to the HDDSL method remain above the other curves representing the IDSL and IDFDD methods. Therefore, it indicates that the proposed method performs better than IDFDD and IDSL methods in fewer iterations and CPU time. Consequently, it is the most efficient method. Finally, the proposed method successfully solves large-scale nonlinear problems from the results in Tables (3-10).

It is essential to state the advantages of the HDDSL method had over the IDFDD, and IDSL methods, including:

(i) The proposed method converges much faster to solutions of the problems, which is shown by the final norm value attained for each problem,

(ii) Better performance for solving the test problems with uniform and mixed valued initial starting points.

8 Conclusion

A hybrid double direction and step length method for solving a system of nonlinear equations is presented in this work. We achieved this by modifying the method in [8] as well as approximating the Jacobian matrix via acceleration parameter. Furthermore, the Picard-Maan hybrid iterative scheme is employed to obtain the correction parameter. We used a set of large-scale test problems to make numerical comparisons. The proposed method is an entirely derivative-free iterative method, which is why it successfully solved large-scale problems. Moreover, Table (3-10) and Figure (1-2) demonstrated that the proposed method is practically quite efficient because it has the fewest number of iterations and CPU time compared to the IDSL and IDFDD methods. In addition, the proposed method successfully solved Problem 6, a discretized Chandrasekhar H-equation problem arising in heat transfer. The limitations of the HDDSL method lie in the fact that if the correction parameter θ is assigned outside the interval $(1, 2)$, then it takes much time to converge and, in some instances, diverges. The idea proposed in this paper can be used to solve monotone nonlinear equations in future research, with application in compressive sensing.

Acknowledgement We thank the reviewers for the comments and suggestions.

Conflicts of Interest The authors declare no conflict of interests.

References

[1] H. Abdullahi, A. S. Halilu & M. Y. Waziri (2018). A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear equations. *Journal of Numerical Mathematics and Stochastics*, *10*(1), 32–44. https://doi.org/10.13140/RG.2.2.23757.67045.

[2] P. N. Brown & Y. Saad (1994). Convergence theory of nonlinear Newton-Krylov algorithms search. *SIAM Journal on Optimization*, *4*(2), 297–330. https://doi.org/10.1137/0804017.

[3] J. E. Dennis & J. J. More (1974). A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation*, *28*, 549–560. https://doi.org/10.2307/2005926.

[4] J. E. Dennis & R. B. Schnabel (1996). *Numerical method for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics, United States, US.

[5] E. Dolan & J. Moré (2002). Benchmarking optimization software with performance profiles. *Journal of Mathematical Programming*, *91*(2), 201–213. https://doi.org/10.1007/s101070100263.

[6] A. S. Halilu & M. Y. Waziri (2017). A transformed double steplength method for for solving large-scale systems of nonlinear equations. *Journal of Numerical Mathematics and Stochastics*, *9*(1), 20–23.

[7] A. S. Halilu & M. Y. Waziri (2018). An improved derivative-free method via double direction approach for solving systems of nonlinear equations. *Journal of the Ramanujan Mathematical Society*, *33*, 75–89.

[8] A. S. Halilu & M. Y. Waziri (2020). Inexact double step length method for solving systems of nonlinear equations. *Statatistics, Optimization and Information Computing*, *8*, 165–174. https://doi.org/10.19139/soic-2310-5070-532.

[9] D. Li & M. Fukushima (1999). A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equations. *SIAM Journal on Numerical Analysis*, *37*, 152–172. https://www.jstor.org/stable/2587031.

[10] K. Meintjes & A. P. Morgan (1987). A methodology for solving chemical equilibrium systems. *Applied Mathematics and Computation*, *22*, 333–361. https://doi.org/10.1016/0096-3003(87)90076-2.

[11] N. I. D. Milicic (2008). A multi step curve search algorithm in nonlinear optimization. *Yugoslav Journal of Operation Research*, *18*, 47–52. https://doi.org/10.2298/YJOR0801047D.

[12] N. I. D. Milicic & M. G. Filipovic (2010). A multi step curve search algorithm in nonlinear convex case. *Facta Universitatis Series: Mathematics and Informatics*, *25*, 11–24.

[13] M. J. Petrovic (2015). An accelerated double step size model in unconstrained optimization. *Applied Mathematics and Computation*, *250*, 309–319. https://doi.org/10.1016/j.amc.2014.10.104.

[14] M. J. Petrović (2019). Hybridization rule applied on accelerated double step size optimization scheme. *Filomat*, *33*(3), 655–665.

[15] M. J. Petrovic & P. S. Stanimirovic (2014). Accelerated double direction method for solving unconstrained optimization problems. *Mathematical Problem in Engineering*, *2014*, Article ID: 965104. https://doi.org/10.1155/2014/965104.

[16] M. J. Petrović, P. S. Stanimirović, N. Kontrec & J. Mladenovíc (2018). Hybrid modification of accelerated double direction method. *Mathematical Problems in Engineering*, *2018*, Article ID 1523267. https://doi.org/10.1155/2018/1523267.

[17] H. K. Safeer (2013). A Picard-Mann hybrid iterative process. *Fixed Point Theory and Applications*, *69*, 1–10. https://doi.org/10.1186/1687-1812-2013-69.

[18] M. Sun, M. Y. Tian & Y. J. Wang (2019). Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization. *Discrete Dynamics in Nature and Society*, *2019*, Article ID: 4745759. https://doi.org/10.1155/2019/4745759.

[19] M. Y. Waziri, W. J. Leong, M. A. Hassan & M. Monsi (2011). A low memory solver for integral equations of Chandrasekhar type in the radiative transfer problems. *Mathematical Problems in Engineering*, *2011*, Article ID: 467017. https://doi.org/10.1155/2011/467017.

[20] G. Yuan & X. Lu (2008). A new backtracking inexact BFGS method for symmetric nonlinear equations. *Journal of Computers and Mathematics with Applications*, *55*, 116–129. https://doi.org/10.1016/j.camwa.2006.12.081.

[21] S. Zhen-Jun & S. Jie (2006). Convergence of nonmonotone line search method. *Journal of Computational and Applied Mathematics*, *193*, 397–412. https://doi.org/10.1016/j.cam.2005.06.033.