Iranian Journal of Numerical Analysis and Optimization
Vol. 12, No. 2, pp 426–448
DOI: 10.22067/IJNAO.2022.74201.1083
https://ijnao.um.ac.ir/

How to cite this article
Research Article

Modification of the double direction approach for solving systems of nonlinear equations with application to Chandrasekhar's Integral equation

A.I. Kiri, M.Y. Waziri and A.S. Halilu*

Abstract

This study aims to present an accelerated derivative-free method for solving systems of nonlinear equations using a double direction approach. The approach approximates the Jacobian using a suitably formed diagonal matrix by applying the acceleration parameter. Moreover, a norm descent line search is employed in the scheme to compute the optimal step length. Under the primary conditions, the proposed method's global convergence is proved. Numerical results are recorded in this paper using a set of large-scale test problems. Moreover, the new method is successfully used to address the problem of Chandrasekhar's integral equation problem appearing in radiative heat transfer. This method outperforms the existing Newton and inexact double step length methods.

AMS subject classifications (2020): 65K05, 90C53, 65D32, 34G20.

Keywords: Credit default swap (CDS); Acceleration parameter; Matrix-free, Inexact line search; Jacobian matrix.

* Corresponding author

Received 15 December 2021; revised 8 February 2022; accepted 28 February 2022

Aliyu Ibrahim Kiri
Department of Mathematics, Department of Mathematical Sciences, Bayero University, Kano, Nigeria. e-mails: aikiri.mth@buk.edu.ng

Mohammed Yusuf Waziri
Numerical Optimization Group, Department of Mathematical Sciences, Bayero University, Kano, Nigeria. e-mails: mywaziri.mth@gmail.com

Abubakar Sani Halilu
Numerical Optimization Group, Bayero University, Kano, Nigeria,
Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria. e-mail: abubakars.halilu@slu.edu.ng

1 Introduction

Scientists are interested in nonlinear problems because most engineering, biology, mathematics, physics, and other science problems are naturally nonlinear. The standard nonlinear equation system is represented by

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear map. The space $\mathbb{R}^n$ denotes the n-dimensional real space, $\| \cdot \|$ is the Euclidean norm, and $F_k = F(x_k)$ is used throughout this paper. Further applications of problem (1) can be found in chemical equilibrium systems [16] and signal and image processing [25]. The Chandrasekhar H-equation that arises in the theory of radioactive heat transfer is a nonlinear integral equation that can be discretized into nonlinear equations [20]. Iterative methods for solving these problems include the Newton and quasi-Newton methods [4, 21, 26, 14], Levenberg–Marquardt methods [15, 13, 12], matrix-free methods [17, 1, 8], and tensor methods [2]. Typically, the iterative formula for solving these methods is given by

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

where x_{k+1} represents a current iterate, x_k is the previous iterate, α_k is a step length, and d_k is the search direction can be calculated by solving system of linear equations as follows:

$$F_k + F_k' d_k = 0, \tag{3}$$

where F_k' is the Jacobian matrix of F_k at x_k. One of the most important requirements of the line search is to reduce the function values sufficiently [9, 11], as shown below:

$$\|F_{k+1}\| \leq \|F_k\|. \tag{4}$$

Irrespective of how appealing the Newton and quasi-Newton approaches are, the Jacobian matrix or its approximation can be calculated at each iteration, making them unsuitable for solving large-scale problems. Due to the drawbacks of these methods, the double direction technique has been proposed [6], with the following iterate:

$$x_{k+1} = x_k + \alpha_k d_k + \alpha_k^2 b_k, \tag{5}$$

where d_k and b_k are search directions, respectively.

Suppose that f is a merit function defined by

$$f(x) = \frac{1}{2}\|F(x)\|^2. \tag{6}$$

Then problem (1) is analogous to the unconstrained optimization problem described below:

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{7}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, and condition (4) is equivalent to

$$f(x_k + \alpha_k d_k) \leq f(x_k). \tag{8}$$

The iterative method generating the sequence $\{x_k\}$ that satisfies (4) is called the norm descent method [9]. If d_k is a descent direction of f at x_k, then condition (8) holds for all $\alpha_k > 0$ small enough. The Newton method (NM) with line search is norm descent. Nonetheless, d_k might not be a descent direction of f at x_k for quasi-Newton methods, even if the approximation of the Jacobian matrix B_k is positive definite and symmetric. Li and Fukushima [14] proposed an approximately norm-descent line search approach. However, the proposed method is not norm descent, but they established a global convergence theorem under the assumption that Jacobian is uniformly nonsingular.

The concept of the double direction approach was suggested by Duranović-Miličić [6] by using a multi-step iterative scheme and curve search to generate new iterates. However, in [7], another double direction algorithm was also presented to minimize nondifferentiable functions. Motivated by the work presented in [7], Petrović and Stanimirović [19] suggested a double direction model for solving unconstrained optimization problems. They used the acceleration parameter γ_k to approximate the Hessian matrix, that is,

$$\bigtriangledown^2 f(x_k) \approx \gamma_k I, \tag{9}$$

where I is the identity matrix, and the sequence of iterates $\{x_k\}$ is generated using (5). The attractive feature of the scheme in [19] is that the two directions presented in their work are derivative-free. Therefore, it enables their method to solve large-scale problems. However, the literature is infrequent to study derivative-free double direction methods for solving nonlinear equations. Based upon the idea presented in [19], Halilu and Waziri used the scheme in (5) to propose a method for solving a system of nonlinear equations using a double direction approach. They used the acceleration parameter $\gamma_k > 0$ in their work to approximate the Jacobian matrix, that is,

$$F'_k \approx \gamma_k I, \tag{10}$$

where I is an identity matrix and the acceleration parameter is derived as

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2 \gamma_k) y_k^T d_k}. \tag{11}$$

The method's global convergent is proved by assuming that the Jacobian of F is positive definite and bounded. The double direction scheme is justified by the fact that scheme (5) contains two corrections. If one of the iterative corrections fails, then the system will be corrected by the second.

The implementation of double direction is additionally enhanced by Petrović [18], where the double step length scheme for the unconstrained optimization problem is presented as

$$x_{k+1} = x_k + \alpha_k d_k + \beta_k b_k, \tag{12}$$

where α_k and β_k are two different step lengths. The numerical results indicated the approach is quite effective compared to the double direction method in [19]. The authors in [8] incorporated the concept in (12) and transformed the double step length method for solving (1) to improve the numerical results and global convergence properties of the double direction scheme. The numerical results exhibited that the method in [8] is more reasonable than the method in [10] because it converges faster. Furthermore, the method [8] is globally converged using the line search proposed in [14]. Motivated by the work in [8], Halilu and Waziri [11] presented an inexact double step length method for solving (1). The attractive feature of this method is that it has a double step length and a single direction that satisfies the decent properties independent of line search. Despite the good convergence properties of the method in [10], its numerical performance is defined as weaker. Therefore, motivated by this reason, we aim to develop a globally converged derivative-free method with a line search to solve a system of nonlinear equations without calculating the Jacobian matrix.

Table 1: Authors' contribution table

Author's Name	Derivative-free	Matrix-free	Double Direction	Global Convergence	Application
Duranović-Miličić [6]	✓	✓	✓	✓	
Halilu and Waziri [10]	✓	✓	✓	✓	
Duranovic-Milicic [7]	✓	✓	✓	✓	
Musa, Waziri, and Halilu [17]	✓			✓	
Abdullahi, Waziri, and Halilu [1]	✓	✓		✓	✓
Petrović and Stanimirović [19]	✓	✓	✓	✓	
Kanzow et al. [12]	✓			✓	
Halilu and Waziri [11] ✓	✓	✓	✓	✓	
Yuan and Lu [26]	✓			✓	
Waziri et al. [23]	✓	✓			✓
Halilu and Waziri [8]	✓	✓	✓	✓	
Li and Fukushima [14]	✓			✓	
Halilu and Waziri [9]	✓	✓	✓	✓	
petrović [18]	✓	✓	✓	✓	
This article	✓	✓	✓	✓	✓

The research gap between the existing method and this article is described in Table 1 above. The table clearly shows that only the proposed method is derivative-free, matrix-free, double direction method, globally convergent, and can be applied to solve discretized Chandrasekhar's integral equation among the listed articles.

We now describe how the paper is structured. The proposed method's algorithm will be presented in section 2. Section 3 illustrates the convergence results. Section 4 contains a list of numerical experiments and applications of the proposed method to Chandrasekhar's integral equation, which arises in radiative heat transfer. Section 5 concludes the paper.

2 Main result

In this section, we present the algorithm of our method. We suggest that the d_k and b_k in (5) are defined as follows:

$$d_k = -\gamma_k^{-1} F_k \tag{13}$$

and

$$b_k = -F_k, \tag{14}$$

where $\gamma_k > 0$ is an acceleration parameter. By substituting (13) and (14) into (5), we obtain

$$x_{k+1} = x_k - (\alpha_k + \alpha_k^2 \gamma_k)\gamma_k^{-1} F_k. \tag{15}$$

The acceleration parameter can be obtained using Taylor's series expansion below:

$$F_{k+1} \approx F_k + F'(\psi)(x_{k+1} - x_k). \tag{16}$$

By multiplying (16) through by θ_k, we have

$$\theta_k F_{k+1} \approx \theta_k F_k + \theta_k F'(\psi)(\alpha_k + \alpha_k^2 \gamma_k) d_k, \tag{17}$$

where $\theta_k > 0$ and ψ satisfies the conditions $\psi \in [x_k, x_{k+1}]$ and

$$\psi = x_k + \zeta(x_{k+1} - x_k), \quad 0 \leq \zeta \leq 1. \tag{18}$$

Taking $\zeta = 1$ in (18), obtain $\psi = x_{k+1}$.
We like to make Jacobian approximations via

$$\theta_k F'(\psi) \approx \gamma_{k+1} I. \tag{19}$$

Using (17) and (19), it is easy to confirm that

$$\gamma_{k+1} s_k = \theta_k y_k, \tag{20}$$

where $s_k = (\alpha_k + \alpha_k^2 \gamma_k) d_k$, $y_k = F_{k+1} - F_k$, and $\theta_k = \dfrac{s_k^T s_k}{y_k^T s_k}$ (see [24]).

The proposed acceleration parameter is defined by multiplying y_k^T on both sides of (20)

$$\gamma_{k+1} = \frac{\|s_k\|^2 \|y_k\|^2}{(\alpha_k + \alpha_k^2 \gamma_k)^2 (y_k^T d_k)^2}. \tag{21}$$

Our proposed scheme is given by equation (22) below based on (13) and (15):

$$x_{k+1} = x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k. \tag{22}$$

Algorithm 1 Modification of the double direction approach (MDFDD)

```
Input: Given $x_0$, $\gamma_0 = 1$, $\epsilon = 10^{-5}$, $\phi_1 > 0$, $\phi_2 > 0$, and $r \in (0,1)$, set $k = 0$.
Step 1: Compute $F_k$.
Step 2: If $\|F_k\| \leq \epsilon$, then stop; otherwise, proceed to Step 3.
Step 3: Calculate search direction $d_k = -\gamma_k^{-1} F_k$.
Step 4: Set $x_{k+1} = x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k$, where $\alpha_k = r^{a_k}$ with $a_k$ being the smallest nonnegative integer $a$ such that
    $$f(x_k + (\alpha_k + \alpha_k^2 \gamma_k) d_k) - f(x_k) \leq -\phi_1 \|\alpha_k F_k\|^2 - \phi_2 \|\alpha_k d_k\|^2 + \tau_k f(x_k). \tag{23}$$
    Let $\{\tau_k\}$ be a given positive sequence such that
    $$\sum_{k=0}^{\infty} \tau_k < \tau < \infty. \tag{24}$$
Step 5: Compute $F_{k+1}$.
Step 6: Determine $\gamma_{k+1} = \frac{\|s_k\|^2 \|y_k\|^2}{(\alpha_k + \alpha_k^2 \gamma_k)^2 (y_k^T d_k)^2}$.
Step 7: Consider $k = k + 1$ and go to Step 2.
```

3 Convergence Analysis

We present how the proposed Algorithm 2 (MDFDD) converges globally in this section. Let us start by defining the level set

$$\Omega = \{x | \|F(x)\| \leq \|F(x_0)\|\}. \tag{25}$$

However, we require the following assumptions:

Assumption 1. However, we state the following assumptions:

1. There exists $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.
2. F is continuously differentiable in some neighborhood say Q of x^* containing Ω.
3. The Jacobian of F is bounded and positive definite on Q. That is, there exist positive constants $H > h > 0$ such that

$$\|F'(x)\| \leq H \quad \text{for all } x \in Q, \tag{26}$$

and

$$h\|d\|^2 \le d^T F'(x)d \quad \text{for all } x \in Q, d \in \mathbb{R}^n. \tag{27}$$

Remark 1. We make the following remark:
Assumption 1 implies that there exist constants $H > h > 0$ such that

$$h\|d\| \le \|F'(x)d\| \le H\|d\| \quad \text{for all } x \in Q, d \in \mathbb{R}^n, \tag{28}$$

$$h\|x-y\| \le \|F(x) - F(y)\| \le H\|x-y\| \quad \text{for all } x, y \in Q. \tag{29}$$

Since $\gamma_k I$ approximates F_k' along s_k, the following assumption can be made.

Assumption 2. $\gamma_k I$ is a good approximation to F_k', that is,

$$\|(F_k' - \gamma_k I)d_k\| \le \varepsilon \|F_k\|, \tag{30}$$

where $\varepsilon \in (0, 1)$ is a small quantity [26].

Lemma 1. Suppose that Assumption 2 holds, and let $\{x_k\}$ be generated by the MDFDD algorithm. Then d_k is a sufficient descent direction for $f(x_k)$ at x_k, that is,

$$\triangledown f(x_k)^T d_k < c\|F_k\|^2, \quad c > 0. \tag{31}$$

Proof. From (13), we have

$$\begin{aligned} \triangledown f(x_k)^T d_k &= F_k^T F_k' d_k \\ &= F_k^T [(F_k' - \gamma_k I)d_k - F_k] \\ &= F_k^T (F_k' - \gamma_k I)d_k - \|F_k\|^2, \end{aligned} \tag{32}$$

by the Cauchy–Schwarz inequality, we have

$$\begin{aligned} \triangledown f(x_k)^T d_k &\le \|F_k\| \|(F_k' - \gamma_k I)d_k\| - \|F(x_k)\|^2 \\ &\le -(1-\epsilon)\|F(x_k)\|^2. \end{aligned} \tag{33}$$

This lemma is true for $\varepsilon \in (0, 1)$.

We can conclude from Lemma 1 that the norm function $f(x_k)$ is a descent along d_k, which means that $\|F_{k+1}\| \le \|F_k\|$ is true. □

Lemma 2. Suppose that Assumption 1 holds, and let $\{x_k\}$ be generated by the MDFDD algorithm. Then $\{x_k\} \subset \Omega$.

Proof. From Lemma 1, we have $\|F_{k+1}\| \le \|F_k\|$. Furthermore, for all k,

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

This means that $\{x_k\} \subset \Omega$. □

Lemma 3 (see [26])**.** Suppose that Assumption 1 holds, and let $\{x_k\}$ be generated by the MDFDD algorithm. Then there exists a constant $m > 0$ such that for all k,

$$y_k^T s_k \geq h\|s_k\|^2. \tag{34}$$

Lemma 4. Suppose that Assumption 1 holds and that $\{x_k\}$ is generated by the MDFDD algorithm. Then

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0 \tag{35}$$

and

$$\lim_{k\to\infty} \|\alpha_k F_k\| = 0. \tag{36}$$

Proof. By (23) for all $k > 0$

$$\begin{aligned} \phi_2\|\alpha_k d_k\|^2 &\leq \phi_1\|\alpha_k F_k\|^2 + \phi_2\|\alpha_k d_k\|^2 \\ &\leq \|F_k\|^2 - \|F_{k+1}\|^2 + \tau_k\|F_k\|^2. \end{aligned} \tag{37}$$

By summing the above inequality, we have

$$\begin{aligned} \phi_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 \leq \sum_{i=0}^{k} \left(\|F_i\|^2 - \|F_{i+1}\|^2\right) + \sum_{i=0}^{k} \eta_i\|F_i\|^2, \\ = \|F_0\|^2 - \|F_{k+1}\|^2 + \sum_{i=0}^{k} \tau_i\|F_i\|^2, \\ \leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{k} \tau_i, \\ \leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{\infty} \tau_i. \end{aligned} \tag{38}$$

From the level set and the fact that $\{\tau_k\}$ satisfies (24), then the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ converges. This implies (35). Using the same logic as above, but this time with $\phi_1\|\alpha_k F_k\|^2$ on the left, we obtain (36). □

Lemma 5. Suppose that Assumption 1 holds, and let $\{x_k\}$ be generated by the MDFDD algorithm. Then there exists a constant $m_1 > 0$ such that for all $k > 0$,

$$\|d_k\| \leq B. \tag{39}$$

Proof. From (13) and (21), we have

$$\begin{aligned}\|d_k\| &= \left\| -\frac{(y_{k-1}^T s_{k-1})^2 F_k}{\|y_{k-1}\|^2\|s_{k-1}\|^2} \right\| \\ &\leq \frac{\|F_k\|\|s_{k-1}\|^2\|y_{k-1}\|^2}{\|s_{k-1}\|^2\|y_{k-1}\|^2} \\ &\leq \|F_0\|.\end{aligned} \tag{40}$$

Choosing $B = \|F_0\|$, we have (39). □

Theorem 1. Suppose that Assumption 1 holds, and let $\{x_k\}$ be generated by the MDFDD algorithm. Assume further, for all $k > 0$,

$$\alpha_k \geq \lambda \frac{|F_k^T d_k|}{\|d_k\|^2}, \tag{41}$$

where λ is some positive constant. Then

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{42}$$

Proof. From Lemma 5, we have (39). Also, from (35) and the boundedness of $\{\|d_k\|\}$, we have

$$\lim_{k\to\infty} \alpha_k\|d_k\|^2 = 0. \tag{43}$$

From (41) and (43), we have

$$\lim_{k\to\infty} |F_k^T d_k| = 0. \tag{44}$$

Also, from (13), we have

$$F_k^T d_k = -\gamma_k^{-1}\|F_k\|^2, \tag{45}$$

$$\begin{aligned}\|F_k\|^2 &= \| - F_k^T d_k \gamma_k\| \\ &\leq |F_k^T d_k||\gamma_k|.\end{aligned} \tag{46}$$

Since

$$\gamma_k^{-1} = \frac{(y_{k-1}^T s_{k-1})^2}{\|y_{k-1}\|^2\|s_{k-1}\|^2} \geq \frac{h^2\|s_{k-1}\|^4}{\|y_{k-1}\|^2\|s_{k-1}\|^2} \geq \frac{h^2\|s_{k-1}\|^2}{H^2\|s_{k-1}\|^2} = \frac{h^2}{H^2},$$

then

$$|\gamma_k^{-1}| \geq \frac{h^2}{H^2}.$$

Therefore from (46), we have

$$\|F_k\|^2 \leq |F_k^T d_k| \left(\frac{H^2}{h^2}\right). \tag{47}$$

As a result,

$$0 \leq \|F_k\|^2 \leq |F_k^T d_k| \left(\frac{H^2}{h^2}\right) \longrightarrow 0. \tag{48}$$

Hence,

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{49}$$

□

4 Numerical experiments

The first part of this section provides numerical results to demonstrate the efficacy of the proposed method by comparing it with existing methods.

- IDFDD: Algorithm 1 proposed in [10].
- IDSL: Algorithm 1 proposed in [11].

The MDFDD method is used in the second part to solve the problem of Chandrasekhar's integral equation in radiative heat transfer. The computer codes used were written in MATLAB 9.4.0 (R2018a) and ran on a computer with a 1.80 GHz CPU processor and 8 GB RAM.

4.1 Experiment of some nonlinear systems of equations

In the experiments, we implemented the three algorithms using the same line search (23), with $\phi_1 = \phi_2 = 10^{-4}$, $r = 0.2$, and $\tau_k = \dfrac{1}{(k+1)^2}$. The iteration is set to stop for the three methods if $\|F_k\| \leq 10^{-5}$ or when the number of iterations overreaches 1000, but there is no x_k meeting the stopping criterion. The numerical effects of the three methods are shown in Tables 3–9, where "ITRN," "CTM(S)," and "IP" represent the total number of iterations, CPU time (in seconds), and initial points, respectively. In addition, $\|Fk\|$ represents the residual value at the stopping point. The symbol "-" indicates failure due to a memory requirement or when some iterations exceed 1000. We tested the three methods on the current seven test problems, each with a different set of initial points and dimensions (n values). The experiment was carried out with the dimensions 100, 1,000, 2,000, 10,000, 50,000, and 100,000 to demonstrate the comprehensive numerical experiments of the MDFDD, IDFDD, and IDSL methods. Table 2 contains the starting points for the test problems.

The experiments made use of the following test problems:

Problem 1 [8]

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

Table 2: Initial points used in test problems

INITIAL POINTS (IP)	VALUES
IP1	$\left(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}\right)^T$
IP2	$\left(\frac{1}{5}, \frac{1}{5}, \dots, \frac{1}{5}\right)^T$
IP3	$\left(\frac{3}{2}, \frac{3}{2}, \dots, \frac{3}{2}\right)^T$
IP4	$\left(\frac{2}{5}, \frac{2}{5}, \dots, \frac{2}{5}\right)^T$
IP5	$\left(0, \frac{1}{2}, \frac{2}{3}, \dots, 1 - \frac{1}{n}\right)^T$
IP6	$\left(\frac{1}{4}, \frac{-1}{4}, \dots, \frac{(-1)^n}{4}\right)^T$
IP7	$\left(1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n}\right)^T$.

$$F_i = x_i - e^{cos\left(\frac{x_{i-1}+x_i+x_{i+1}}{n+1}\right)},$$
$$F_n = x_n - e^{cos\left(\frac{x_{n-1}+x_n}{n+1}\right)}, \quad i = 2, 3, \dots, n-1.$$

Problem 2 [11]
$F_i(x) = x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2 + (1 - x_i^2), \quad i = 1, 2, \dots, n.$

Problem 3 [24]
$F_i(x) = x_i - x_i\left(\sin x_i - \frac{11}{50}\right) + 2, \quad i = 1, 2, \dots, n.$

Problem 4 [10]
$F_1(x) = (x_1^2 + x_2^2)x_1 - 1,$
$F_i(x) = (x_{i-1}^2 + 2x_i^2 + x_{i+1}^2)x_i - 1,$
$F_n(x) = (x_{n-1}^2 + x_n^2)x_n, \quad i = 2, 3, \dots, n-1.$

Problem 5 [10]
$F_i(x) = 2x_i - \sin|x_i|, \quad i = 1, 2, \dots, n.$

Problem 6 [11]
$F(x) = Ax + b_1,$

where $A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$, and $b_1 = (e_1^x - 1, \dots, e_n^x - 1)^T$.

Problem 7 [8]

$$F(x) = Bx + b_2,$$

$$\text{where } B = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}, \text{ and } b_2 = (\sin x_1 - 1, \ldots, \sin x_n - 1)^T.$$

Table 3: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 1

		MDFDD			IDFDD			IDSL		
Dimension	IP	ITRN	CTM(S)	$\lVert F_k \rVert$	ITRN	CTM(S)	$\lVert F_k \rVert$	ITRN	CTM(S)	$\lVert F_k \rVert$
100	IP1	6	0.009	7.44E-06	60	0.031	8.35E-06	41	0.015	9.86E-06
	IP2	11	0.020	2.67E-06	60	0.023	9.48E-06	42	0.013	7.83E-06
	IP3	6	0.013	9.57E-06	58	0.022	7.93E-06	40	0.022	7.72E-06
	IP4	7	0.009	7.55E-06	60	0.020	8.73E-06	42	0.015	7.21E-06
	IP5	7	0.023	7.17E-06	59	0.023	8.78E-06	41	0.014	7.88E-06
	IP6	8	0.024	7.36E-06	61	0.018	8.49E-06	42	0.026	9.23E-06
	IP7	10	0.027	6.68E-06	61	0.019	7.64E-06	42	0.023	8.3E-06
1,000	IP1	3	0.005	4.35E-07	96	0.057	8.59E-06	45	0.031	7.51E-06
	IP2	3	0.005	5.13E-07	96	0.077	9.76E-06	45	0.051	8.52E-06
	IP3	3	0.007	2.09E-07	94	0.056	8.17E-06	43	0.029	8.41E-06
	IP4	3	0.005	4.6E-07	96	0.058	8.98E-06	45	0.057	7.84E-06
	IP5	3	0.009	3.17E-07	95	0.057	8.8E-06	44	0.035	8.34E-06
	IP6	3	0.007	6.38E-07	97	0.058	8.74E-06	46	0.055	7.03E-06
	IP7	3	0.011	5.66E-07	97	0.092	7.98E-06	45	0.051	9.17E-06
10,000	IP1	3	0.017	1.38E-10	111	0.486	9.05E-06	48	0.239	8.14E-06
	IP2	3	0.035	1.63E-10	112	0.492	7.81E-06	48	0.206	9.24E-06
	IP3	3	0.022	6.64E-11	109	0.477	8.6E-06	46	0.266	9.13E-06
	IP4	3	0.034	1.46E-10	111	0.485	9.45E-06	48	0.182	8.51E-06
	IP5	3	0.037	1E-10	110	0.483	9.23E-06	47	0.232	9.01E-06
	IP6	3	0.021	2.03E-10	112	0.500	9.2E-06	49	0.182	7.63E-06
	IP7	3	0.034	1.8E-10	112	0.490	8.42E-06	48	0.206	9.97E-06
50,000	IP1	3	0.103	4.96E-13	114	1.948	8.88E-06	50	0.752	8.92E-06
	IP2	3	0.087	5.96E-13	115	1.985	7.66E-06	51	0.778	7.09E-06
	IP3	2	0.072	8.87E-06	112	1.910	8.44E-06	48	0.718	1E-05
	IP4	3	0.113	4.96E-13	114	1.949	9.28E-06	50	0.748	9.32E-06
	IP5	3	0.101	2.99E-13	113	1.934	9.05E-06	49	0.903	9.87E-06
	IP6	3	0.093	6.95E-13	115	1.978	9.03E-06	51	0.752	8.36E-06
	IP7	3	0.079	5.96E-13	115	1.933	8.27E-06	51	0.760	7.65E-06
100,000	IP1	2	0.082	6.57E-06	115	4.097	9.54E-06	51	1.465	8.83E-06
	IP2	2	0.116	7.75E-06	116	4.347	8.23E-06	52	1.534	7.02E-06
	IP3	2	0.104	3.14E-06	113	3.826	9.07E-06	49	1.392	9.9E-06
	IP4	2	0.108	6.95E-06	115	3.872	9.97E-06	51	1.497	9.23E-06
	IP5	2	0.104	4.76E-06	114	3.852	9.73E-06	50	1.473	9.77E-06
	IP6	2	0.095	9.65E-06	116	3.923	9.71E-06	52	1.536	8.27E-06
	IP7	2	0.099	8.57E-06	116	3.827	8.89E-06	52	1.537	7.57E-06

From Tables 3–9, we can observe that the three methods are trying to solve (1). However, the improvement and effectiveness of the proposed method are pretty straightforward. The tables indicated that modifying the IDFDD method in the proposed scheme is a good improvement. The MDFDD method remarkably outperforms the IDFDD and IDSL methods for nearly all the problems assessed since it has the least number of iterations, which are far below the number of iterations for the IDFDD and IDSL methods. Moreover, the proposed method has less CPU time than the IDFDD method. However, the MDFDD method has a higher CPU time than

Table 4: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 2

		MDFDD			**IDFDD**			**IDSL**		
Dimension	IP	ITRN	CTM(S)	$\lVert F_k \rVert$	ITRN	CTM(S)	$\lVert F_k \rVert$	ITRN	CTM(S)	$\lVert F_k \rVert$
100	IP1	13	0.026	6.15E-06	26	0.022	7.35E-06	37	0.010	8.96E-06
	IP2	20	0.036	9.95E-06	19	0.014	9.06E-06	28	0.012	9.49E-06
	IP3	15	0.043	5.55E-06	30	0.022	7.09E-06	43	0.021	7.75E-06
	IP4	18	0.018	4.46E-06	27	0.018	9.54E-06	33	0.017	8.71E-06
	IP5	20	0.024	6.27E-06	64	0.041	9.19E-06	34	0.009	8.09E-06
	IP6	11	0.017	5.10E-06	30	0.023	7.61E-06	38	0.012	9.29E-06
	IP7	23	0.043	6.62E-06	39	0.024	6.34E-06	43	0.021	9.82E-06
1,000	IP1	19	0.062	2.86E-07	28	0.020	9.52E-06	40	0.013	9.72E-06
	IP2	17	0.040	7.93E-06	22	0.025	7.51E-06	32	0.025	7.21E-06
	IP3	20	0.072	5.27E-06	32	0.032	9.18E-06	46	0.028	8.4E-06
	IP4	16	0.050	6.52E-06	30	0.024	7.91E-06	36	0.027	9.45E-06
	IP5	18	0.062	8.70E-06	71	0.054	7.79E-06	34	0.027	7.27E-06
	IP6	12	0.045	5.73E-06	32	0.020	9.85E-06	42	0.039	7.06E-06
	IP7	24	0.065	6.80E-06	43	0.018	5.72E-06	43	0.032	9.51E-06
10,000	IP1	33	0.300	5.24E-06	31	0.174	7.89E-06	44	0.153	7.38E-06
	IP2	41	0.345	3.48E-06	24	0.143	9.72E-06	35	0.131	7.82E-06
	IP3	14	0.208	7.43E-06	35	0.129	7.61E-06	49	0.185	9.11E-06
	IP4	35	0.269	3.35E-06	33	0.174	6.55E-06	40	0.101	7.17E-06
	IP5	17	0.220	6.73E-06	70	0.209	8.45E-06	33	0.119	9.98E-06
	IP6	14	0.207	4.16E-07	35	0.119	8.17E-06	45	0.141	7.65E-06
	IP7	29	0.350	5.4E-06	44	0.157	8E-06	46	0.137	7.38E-06
50,000	IP1	37	0.719	2.83E-06	33	0.443	7.23E-06	46	0.389	8.08E-06
	IP2	45	0.792	2.38E-07	26	0.339	8.91E-06	37	0.334	8.57E-06
	IP3	16	0.594	2.96E-06	37	0.461	6.97E-06	51	0.472	9.99E-06
	IP4	43	1.100	9.23E-07	34	0.433	9.38E-06	42	0.408	7.86E-06
	IP5	14	0.629	4.38E-06	73	0.825	8.01E-06	33	0.281	9.93E-06
	IP6	14	0.515	8.34E-06	37	0.556	7.48E-06	47	0.435	8.39E-06
	IP7	32	1.103	8.78E-06	45	0.527	8.09E-06	48	0.447	7.49E-06
100,000	IP1	36	1.114	8.69E-06	34	0.759	6.54E-06	47	0.786	8E-06
	IP2	50	1.775	8.2E-07	27	0.645	8.06E-06	38	0.650	8.48E-06
	IP3	11	0.595	7.05E-07	37	0.865	9.86E-06	52	0.873	9.89E-06
	IP4	42	1.368	5.53E-06	35	0.760	8.49E-06	43	0.697	7.78E-06
	IP5	17	1.597	6.23E-06	72	1.480	8.15E-06	33	0.578	9.92E-06
	IP6	21	1.756	3.55E-06	38	0.833	6.77E-06	48	0.775	8.3E-06
	IP7	34	2.416	6.9E-06	46	1.029	9.52E-06	49	0.823	7.3E-06

Table 5: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 3

		MDFDD			**IDFDD**			**IDSL**		
Dimension	IP	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$
100	IP1	16	0.034	8.52E-06	32	0.012	9.96E-06	37	0.010	8.32E-06
	IP2	12	0.033	9.99E-06	31	0.023	8.84E-06	26	0.015	7.7E-06
	IP3	20	0.044	2.25E-06	35	0.021	6.42E-06	41	0.008	8.47E-06
	IP4	17	0.021	1.76E-06	32	0.025	8.45E-06	36	0.017	7.56E-06
	IP5	20	0.045	6.07E-06	34	0.012	6.95E-06	40	0.020	7.81E-06
	IP6	10	0.023	1.06E-06	28	0.016	7.37E-06	35	0.017	8.38E-06
	IP7	17	0.030	7.24E-06	31	0.023	7E-06	35	0.007	8.61E-06
1,000	IP1	22	0.071	6.32E-06	35	0.047	8.26E-06	40	0.015	9.02E-06
	IP2	15	0.039	4.47E-06	34	0.048	7.33E-06	29	0.012	8.35E-06
	IP3	26	0.074	2.66E-07	37	0.044	8.32E-06	44	0.047	9.19E-06
	IP4	24	0.038	3.26E-06	35	0.032	7.01E-06	39	0.030	8.2E-06
	IP5	28	0.051	7.39E-07	36	0.049	9.28E-06	43	0.040	8.83E-06
	IP6	12	0.037	3.85E-06	30	0.026	9.54E-06	38	0.033	9.09E-06
	IP7	13	0.082	4.29E-06	33	0.036	7.05E-06	38	0.028	7.25E-06
10,000	IP1	26	0.295	6.95E-06	38	0.200	6.85E-06	43	0.114	9.78E-06
	IP2	24	0.345	1.95E-06	36	0.194	9.5E-06	32	0.168	9.05E-06
	IP3	37	0.506	3.4E-06	40	0.181	6.9E-06	47	0.211	9.97E-06
	IP4	27	0.319	9.21E-06	37	0.178	9.09E-06	42	0.118	8.89E-06
	IP5	30	0.263	9.76E-06	39	0.164	7.73E-06	46	0.166	9.64E-06
	IP6	17	0.331	9.04E-06	33	0.136	7.92E-06	41	0.139	9.86E-06
	IP7	21	0.256	7.66E-06	35	0.166	8.76E-06	41	0.154	7.64E-06
50,000	IP1	29	0.760	6.52E-06	39	0.506	9.81E-06	46	0.477	7.5E-06
	IP2	29	0.941	7.04E-06	38	0.520	8.7E-06	34	0.334	9.92E-06
	IP3	40	1.244	5.85E-06	41	0.529	9.87E-06	50	0.618	7.65E-06
	IP4	28	0.823	3.9E-06	39	0.540	8.32E-06	44	0.491	9.74E-06
	IP5	35	0.871	3.34E-06	41	0.540	7.08E-06	49	0.575	7.4E-06
	IP6	22	0.825	6.22E-06	35	0.488	7.25E-06	44	0.433	7.56E-06
	IP7	24	1.087	8.32E-06	37	0.486	7.99E-06	43	0.446	8.36E-06
100,000	IP1	36	2.601	8.24E-06	40	0.985	8.88E-06	47	0.888	7.43E-06
	IP2	33	2.573	6.18E-06	39	1.132	7.88E-06	35	0.727	9.82E-06
	IP3	43	2.423	9.86E-07	42	1.332	8.94E-06	51	1.080	7.57E-06
	IP4	34	2.529	5.42E-06	40	0.998	7.54E-06	45	0.887	9.65E-06
	IP5	42	2.712	9.63E-06	42	1.056	6.41E-06	50	0.924	7.32E-06
	IP6	20	1.531	2.37E-06	36	0.904	6.56E-06	45	0.862	7.49E-06
	IP7	24	1.634	5.4E-06	38	0.980	7.23E-06	44	0.825	8.27E-06

Table 6: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 4

		MDFDD			**IDFDD**			**IDSL**		
Dimension	IP	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$
100	IP1	37	0.061	9.93E-06	45	0.018	8.01E-06	45	0.011	8.46E-06
	IP2	32	0.066	9.00E-06	47	0.029	8.47E-06	46	0.023	8.83E-06
	IP3	36	0.053	8.51E-06	53	0.032	8.04E-06	54	0.025	9.45E-06
	IP4	37	0.067	7.28E-06	47	0.030	8.34E-06	41	0.019	8.61E-06
	IP5	33	0.056	8.14E-06	49	0.030	9.87E-06	51	0.026	8.19E-06
	IP6	41	0.044	6.93E-06	49	0.022	9.01E-06	52	0.010	7.39E-06
	IP7	26	0.029	7.11E-06	41	0.022	8.36E-06	44	0.027	9.03E-06
1,000	IP1	33	0.084	9.83E-06	46	0.036	1E-05	50	0.040	9.31E-06
	IP2	32	0.090	9.25E-06	48	0.030	9.59E-06	49	0.039	7.86E-06
	IP3	46	0.102	5.74E-06	53	0.043	7.68E-06	59	0.043	7.05E-06
	IP4	28	0.063	7.2E-06	47	0.052	8.51E-06	47	0.017	9.56E-06
	IP5	40	0.080	9.51E-06	52	0.059	9.66E-06	50	0.018	8.99E-06
	IP6	37	0.101	8.49E-06	49	0.032	9.95E-06	55	0.052	7.38E-06
	IP7	24	0.052	7.93E-06	43	0.032	9.41E-06	48	0.038	7.61E-06
10,000	IP1	36	0.657	9.61E-06	48	0.212	8.89E-06	53	0.189	9.81E-06
	IP2	40	0.567	3.06E-06	50	0.215	9.42E-06	54	0.227	9.17E-06
	IP3	58	0.594	9.09E-06	53	0.234	8.38E-06	64	0.216	9.44E-06
	IP4	39	0.761	9.19E-06	50	0.235	9.73E-06	50	0.190	9.54E-06
	IP5	51	0.718	7.54E-06	54	0.283	1E-05	51	0.244	8.49E-06
	IP6	49	0.751	5.9E-06	54	0.254	7.38E-06	58	0.190	7.3E-06
	IP7	34	0.341	9.06E-06	43	0.221	9.2E-06	48	0.165	8.37E-06
50,000	IP1	36	2.321	6.59E-06	48	0.698	9.53E-06	56	0.595	7.59E-06
	IP2	39	1.581	7.05E-06	53	0.770	9.97E-06	54	0.666	8.81E-06
	IP3	68	2.331	9.27E-06	55	0.810	8.88E-06	63	0.670	8.29E-06
	IP4	34	1.625	5.01E-06	51	0.752	9.55E-06	51	0.537	8.47E-06
	IP5	53	2.435	9.69E-06	55	0.795	8.83E-06	53	0.638	7.7E-06
	IP6	48	2.340	8.73E-06	54	0.834	9.04E-06	61	0.676	9.28E-06
	IP7	38	1.301	2.77E-06	44	0.655	8.59E-06	50	0.560	8.04E-06
10,0000	IP1	41	4.922	9.92E-06	49	1.467	9.82E-06	54	1.132	9.69E-06
	IP2	46	4.246	9.16E-06	53	1.650	9.8E-06	59	1.311	9.21E-06
	IP3	73	4.694	5.2E-06	55	1.908	9.38E-06	63	1.342	9.92E-06
	IP4	46	4.739	6.53E-06	52	1.455	8.6E-06	55	1.113	9.46E-06
	IP5	59	5.184	3.46E-06	56	1.598	8.14E-06	58	1.240	9.03E-06
	IP6	53	3.656	9.1E-06	55	1.586	8.23E-06	61	1.309	8.11E-06
	IP7	39	2.125	5.45E-06	44	1.418	9.99E-06	51	1.079	7.92E-06

Table 7: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 5

		MDFDD			**IDFDD**			**IDSL**		
Dimension	IP	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$
100	IP1	19	0.026	2.41E-06	48	0.031	9.76E-06	37	0.023	9.63E-06
	IP2	16	0.015	5.8E-06	45	0.015	8.7E-06	35	0.007	7.63E-06
	IP3	16	0.037	6.63E-06	53	0.033	8.1E-06	41	0.008	7.85E-06
	IP4	13	0.026	1.27E-06	48	0.036	7.74E-06	37	0.007	7.61E-06
	IP5	14	0.032	2.97E-06	51	0.032	8.6E-06	39	0.012	9.66E-06
	IP6	6	0.007	7.47E-07	48	0.027	7.65E-06	33	0.014	8.53E-06
	IP7	12	0.029	8.6E-06	44	0.019	7.78E-06	34	0.012	7.58E-06
1,000	IP1	14	0.042	3.31E-06	53	0.035	7.83E-06	41	0.025	7.31E-06
	IP2	10	0.036	5.19E-06	49	0.054	9.18E-06	38	0.019	8.27E-06
	IP3	17	0.041	6.63E-06	57	0.048	8.55E-06	44	0.039	8.51E-06
	IP4	19	0.040	3.13E-06	52	0.060	8.17E-06	40	0.013	8.26E-06
	IP5	9	0.028	4.13E-06	55	0.057	9.45E-06	43	0.036	7.66E-06
	IP6	9	0.039	2.42E-07	52	0.030	8.07E-06	36	0.018	9.25E-06
	x7	11	0.052	9.45E-06	44	0.040	7.8E-06	34	0.019	7.6E-06
10,000	IP1	13	0.237	5.88E-07	57	0.215	8.26E-06	44	0.108	7.93E-06
	IP2	15	0.333	6.48E-06	53	0.171	9.69E-06	41	0.112	8.97E-06
	IP3	16	0.202	1.66E-06	61	0.192	9.02E-06	47	0.159	9.23E-06
	IP4	13	0.150	7.8E-06	56	0.254	8.62E-06	43	0.132	8.95E-06
	IP5	19	0.402	3.93E-06	60	0.225	7.63E-06	46	0.213	8.36E-06
	IP6	13	0.213	3.62E-06	56	0.252	8.52E-06	40	0.093	7.02E-06
	IP7	13	0.211	2.74E-07	44	0.153	7.8E-06	34	0.079	7.6E-06
50,000	IP1	10	0.618	1.34E-07	60	0.664	8.1E-06	46	0.398	8.69E-06
	IP2	12	0.552	4.71E-06	56	0.604	9.51E-06	43	0.349	9.83E-06
	IP3	12	0.465	5.39E-06	64	0.722	8.85E-06	50	0.422	7.08E-06
	IP4	17	1.063	2.64E-06	59	0.689	8.46E-06	45	0.441	9.81E-06
	IP5	17	0.961	5.8E-06	62	0.693	9.86E-06	48	0.379	9.17E-06
	IP6	11	0.534	5.8E-06	59	0.655	8.36E-06	42	0.332	7.69E-06
	IP7	12	0.682	1.68E-06	44	0.500	7.8E-06	34	0.287	7.6E-06
100,000	IP1	11	1.156	7.49E-06	61	1.299	8.71E-06	47	1.125	8.61E-06
	IP2	9	0.684	4.11E-06	58	1.145	7.77E-06	44	0.871	9.73E-06
	IP3	14	1.359	7.52E-06	65	1.336	9.51E-06	51	0.805	7.01E-06
	IP4	10	1.360	7E-06	60	1.480	9.09E-06	46	0.873	9.71E-06
	IP5	15	1.204	4.48E-06	64	1.405	8.05E-06	49	0.877	9.08E-06
	IP6	17	1.563	1.77E-06	60	1.206	8.99E-06	43	0.699	7.62E-06
	IP7	11	1.106	1.63E-06	44	1.113	7.8E-06	34	0.520	7.6E-06

Table 8: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 6

		MDFDD			IDFDD			IDSL		
Dimension	IP	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$
100	IP1	27	0.205	8.46E-06	52	0.128	9.42E-06	38	0.073	8.21E-06
	IP2	20	0.195	6.73E-06	47	0.104	7.62E-06	35	0.072	8.98E-06
	IP3	44	0.264	8.86E-06	60	0.132	9.66E-06	40	0.068	8.38E-06
	IP4	31	0.319	7.88E-06	51	0.115	9.37E-06	37	0.068	9.22E-06
	IP5	34	0.264	4.6E-06	55	0.123	9.69E-06	40	0.071	7.5E-06
	IP6	24	0.239	5.94E-06	47	0.116	8.84E-06	35	0.061	9.78E-06
	IP7	22	0.213	7.94E-06	45	0.106	7.77E-06	37	0.067	9.35E-06
1,000	IP1	31	2.441	9.82E-06	54	1.154	9.19E-06	41	0.664	8.53E-06
	IP2	25	2.532	8.98E-06	50	1.092	9.64E-06	38	0.608	9.09E-06
	IP3	59	2.687	8.13E-06	57	1.178	8.11E-06	43	0.697	8.2E-06
	IP4	31	2.273	8.28E-06	53	1.135	8.6E-06	40	0.638	9.51E-06
	IP5	45	2.836	8.66E-06	55	1.133	9.24E-06	43	0.680	8.2E-06
	IP6	25	2.066	8.18E-06	49	1.047	7.81E-06	38	0.614	9.21E-06
	IP7	22	2.083	6.55E-06	45	0.988	7.83E-06	37	0.587	9.37E-06
2,000	IP1	26	5.618	4.72E-06	56	3.854	7.96E-06	42	2.135	8.42E-06
	IP2	25	6.920	9.43E-06	52	3.626	8.66E-06	39	2.011	8.95E-06
	IP3	59	8.466	7.28E-06	59	4.029	9.91E-06	44	2.260	8.07E-06
	IP4	27	6.605	8.86E-06	55	3.972	8.4E-06	41	2.090	9.39E-06
	IP5	43	7.773	7.29E-06	60	4.223	9.13E-06	44	2.283	8.13E-06
	IP6	27	7.955	6.38E-06	51	3.586	7.86E-06	39	1.978	8.99E-06
	IP7	25	8.424	6.73E-06	45	3.235	7.83E-06	37	1.891	9.37E-06

Table 9: Numerical outcomes of MDFDD, IDFDD, and IDSL methods for problem 7

		MDFDD			IDFDD			IDSL		
Dimension	IP	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$	ITRN	CTM(S)	$\|F_k\|$
100	IP1	17	0.226	9.74E-06	34	0.080	7.68E-06	31	0.053	9.24E-06
	IP2	23	0.253	5.6E-06	41	0.099	8.19E-06	36	0.063	9.62E-06
	IP3	26	0.221	8.98E-06	45	0.102	7.32E-06	40	0.067	7.41E-06
	IP4	16	0.175	8.8E-06	38	0.082	7.76E-06	34	0.063	7.14E-06
	IP5	22	0.174	6.76E-06	42	0.110	9.28E-06	38	0.076	7.42E-06
	IP6	24	0.168	8.55E-06	44	0.100	7.63E-06	39	0.088	8.13E-06
	IP7	21	0.164	5.67E-06	42	0.094	9.17E-06	38	0.073	7.26E-06
1,000	IP1	17	1.582	8.61E-06	35	0.743	8.31E-06	31	0.497	9.65E-06
	IP2	24	1.553	9.77E-06	44	0.940	9.98E-06	40	0.646	7.28E-06
	IP3	28	1.807	4.01E-06	48	1.012	8.61E-06	43	0.687	7.64E-06
	IP4	22	1.580	7.03E-06	41	0.870	9.4E-06	37	0.592	7.62E-06
	IP5	25	1.827	7.8E-06	46	1.024	8.31E-06	41	0.657	8.08E-06
	IP6	25	1.422	7.42E-06	47	1.046	9.26E-06	42	0.670	8.73E-06
	IP7	28	1.849	8.2E-06	46	1.069	8.51E-06	41	0.655	8.27E-06
2,000	IP1	18	5.262	7.49E-06	36	2.459	7.66E-06	32	1.656	8.08E-06
	IP2	28	5.929	8.16E-06	46	3.167	7.43E-06	41	2.121	7.21E-06
	IP3	31	6.115	6.11E-06	49	3.397	8.82E-06	44	2.235	7.54E-06
	IP4	27	6.253	9.63E-06	42	2.907	9.64E-06	38	1.995	7.54E-06
	IP5	30	7.231	5.2E-06	47	3.298	8.55E-06	42	2.194	8.01E-06
	IP6	31	6.675	4.99E-06	48	3.387	9.5E-06	43	2.283	8.64E-06
	IP7	25	6.070	5.32E-06	47	3.219	8.77E-06	42	2.188	8.22E-06

the IDSL method due to the computation of double direction in the MDFDD methods.

Figures 1–2 display the interpretation of the numerical results of each of the three methods using Dolan and Moré [5] performance profiles. We achieve this by plotting fraction $p(\tau)$ of problems for each method within τ of the smallest number of iterations and CPU time. As shown in Figures 1 and 2, the curves representing the MDFDD method remain above the IDFDD and IDSL methods in number iterations. Furthermore, it is above the curve representing the IDFDD method for the CPU time. Therefore, the proposed method outperforms the IDFDD and IDSL methods in fewer iterations and is thus the most efficient method. Finally, from the results in Tables 3–9, it is evident that the MDFDD method successfully solves problem (1).

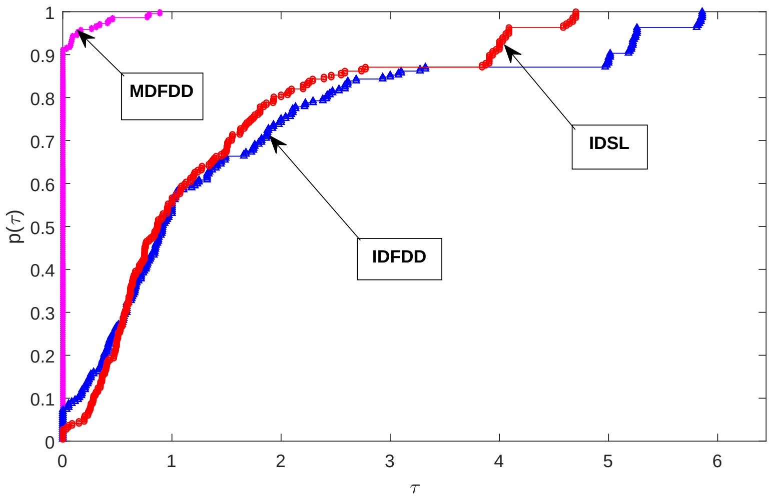

Figure 1: Performance profile with respect to the number of iterations

4.2 Application in integral equations

Chandrasekhar and Breen [3] computed H-equation as the solution of the nonlinear integral equation that gives the complete nonlinear equations technique. The nonlinear integral equation arising in radiative heat transfer problem is given by

$$H(x) = 1 + c\frac{x}{2}H(x)\int_0^1 \frac{H(y)}{x+y}dy, \tag{50}$$

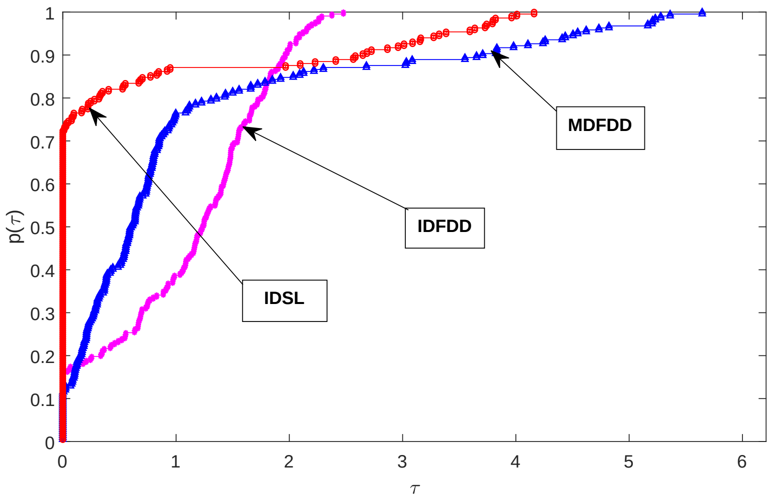

Figure 2: Performance profile with respect to the CPU time (in second)

with parameter $c \in [0,1]$ and $H : [0,1] \to \mathbb{R}$ is an unknown function.

Equation (50) can be written as

$$H(x) = \left[1 - \frac{c}{2}\int_0^1 \frac{xH(y)}{x+y}dy\right] = 1. \tag{51}$$

By multiplying both sides of (51) with $\left(1 - \frac{c}{2}\int_0^1 \frac{xH(y)}{x+y}dy\right)^{-1}$, we have

$$F(H)(x) = H(x) - \left(1 - \frac{c}{2}\int_0^1 \frac{xH(y)}{x+y}dy\right)^{-1} = 0, \tag{52}$$

which is called the Chandrasekhar H-equation [23]. However, (52) can be discretized by using the midpoint quadrature formula

$$\int_0^1 f(\mu)d\mu = h\sum_{j=1}^{n} f(\mu_j), \tag{53}$$

for $\mu_j = (j - 0.5)h$, $0 \leq j \leq 1$, and $h = \frac{1}{n}$.

As a result, we have the following system of nonlinear equations:

$$F_i(x) = x_i - \left(1 - \frac{c}{2n}\sum_{j=1}^{n}\frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1} \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n, \tag{54}$$

which is known as the discretized Chandrasekhar H-equation that can be solved by using some iterative methods. If the initial point $x_0 = (1, 1, 1, \dots, 1)^T$, then the system in (54) has a solution for all $c \in (0, 1)$. However, the hardest part of the problem (54) is that the Jacobian is singular at $c = 1$. Therefore as c approaches 1, the Jacobian approaches the singularity point. Since our method is derivative-free, then it has the advantage to solve problem (54) even when c approaches 1.

To highlight the performance of the MDFDD approach furthermore, we conduct some numerical experiments by comparing it with the classical NM, IDFDD method [10], and IDSL method [11]. The iteration is also set to terminate when $\|x_{k+1} - x_k\| + \|F_k\| \leq 10^{-5}$ or when the iterations exceed 1000, but no point of x_k satisfying the stopping criterion. We have tried the three methods with the starting point of $x_0 = (1, 1, 1, \dots, 1)^T$. Furthermore, we use the dimensions (n values) 100 to 20,000 to show the performance of each of the three methods.

Table 10: Numerical results of discretized Chandrasekhar H-equation

		NM		**IDFDD**		**IDSL**		**MDFDD**	
	Dimension	ITER	TIME	ITER	TIME	ITER	TIME	ITER	TIME
c=0.1	100	12	0.078	-	-	38	0.029	13	0.015
	500	15	0.785	-	-	41	0.017	14	0.015
	1000	15	3.056	-	-	42	0.018	11	0.018
	10000	20	1747	-	-	45	0.150	12	0.141
	20000	-	-	-	-	46	0.219	20	0.349
c=0.9	100	13	0.111	-	-	38	0.009	9	0.009
	500	15	0.789	-	-	41	0.022	17	0.029
	1000	18	3.729	-	-	42	0.026	15	0.024
	10000	-	-	-	-	45	0.231	15	0.215
	20000	-	-	-	-	46	0.216	14	0.377
c=0.99	100	13	0.111	-	-	38	0.012	12	0.011
	500	17	0.874	-	-	41	0.032	17	0.020
	1000	18	3.733	-	-	42	0.019	12	0.018
	10000	-	-	-	-	45	0.123	11	0.125
	20000	-	-	-	-	46	0.218	13	0.285
c=0.999	100	13	0.112	-	-	38	0.019	13	0.014
	500	17	0.883	-	-	41	0.013	16	0.018
	1000	18	3.766	-	-	42	0.025	16	0.029
	10000	-	-	-	-	45	0.130	13	0.176
	20000	-	-	-	-	46	0.318	12	0.282

The numerical results of the methods used to solve Chandrasekhar H-equation with different values of parameter c, are shown in Table 10. The table clearly indicates that the proposed method outperformed the NM because the NM failed when the number of dimension increased. This is due to the fact that as c approaches 1, the Jacobian approaches the singularity point. Moreover, the CPU time (in second) of the NM is higher than other methods because it solved the Jacobian matrix at each iteration. From Table 10, we can also observe that the IDFDD method has totally failed because it has poor numerical performance, as we have made mentioned earlier in the introduction section of this article. Although, the IDSL method solved problem (54) completely, but it has more number of iterations than the MDFDD method. This shows that our method has effectively solved the discretized Chandrasekhar H-equation with the least number of iterations and CPU time.

5 Conclusion

In this article, numerical comparisons were made using a set of large-scale test problems. Furthermore, Tables 3–9 and Figures 1–2 showed that the presented method is practically quite efficient because it has fewer iterations than the IDFDD and IDSL methods. Furthermore, we have successfully used the proposed method to deal with experiments on the Chandrasekhar H-equation in radiative heat transfer. The experiments were carried out and reported in Table 10 with different c values, demonstrating a better efficiency for the MDFDD method. The numerical results showed that the employed method solved the discretized integral equation with fewer iterations and CPU time than the NM, IDFDD, and IDSL methods. Future research includes applying the MDFDD scheme to solve the discretized three-dimensional nonlinear Poisson problem.

References

1. Abdullahi, H., Halilu, A.S. and Waziri M.Y. *A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear systems*, J. numer. math. stoch. 10(1) (2018), 32–44.

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

3. Chandrasekhar, S. and Breen, F. H. *On the radiative equilibrium of a stellar atmosphere*, XIX, Astrophys. J. 106 (1947) 143–144.

4. Dennis, J.E., and Schnabel, R.B. Numerical methods for unconstrained optimization and non-linear equations. Prentice Hall, Englewood Cliffs, NJ, 1983.

5. Dolan, E. and Moré, J. *Benchmarking optimization software with performance profiles*, Math. Program. 91(2) (2002), Ser. A, 201–213.

6. Duranović-Miličić, N.I. *A multi-step curve search algorithm in nonlinear optimization,* Yugosl. J. Oper. Res. 18 (2008), 47–52.

7. Duranović-Miličić, N. I. and Gardasevic-Filipovic, M. *A multi-step curve search algorithm in nonlinear convex case: nondifferentiable convex case,* Facta Univ. Ser. Math. Inform. 25 (2010), 11–24.

8. Halilu, A.S. and Waziri, M.Y. *A transformed double step length method for solving large-scale systems of nonlinear equations,* J Num. Math. Stoch. 9 (2017) 20–32.

9. Halilu, A.S. and Waziri M.Y. *Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations,* Int. J. appl. Math Res. 6 (2017) 147–156.

10. Halilu, A.S. and Waziri, M.Y. *An improved derivative-free method via double direction approach for solving systems of nonlinear equations,* J. Ramanujan Math. Soc. 33 (2018), no. 1, 75–89.

11. Halilu, A.S., and Waziri, M.Y. *Inexact double step length method for solving systems of nonlinear equations,* Stat., Optim. Inf. Comput., 8 (2020) 165–174.

12. Kanzow, C., Yamashita, N. and Fukushima, M. *Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties,* J. Comput. Appl. Math. 172 (2004) 375–397.

13. Levenberg, K. *A method for the solution of certain non-linear problems in least squares,* Quart. Appl. Math. 2 (1944), 164–168.

14. Li, D. and Fukushima M. *A globally and superlinearly convergent Gauss-Newton-based BFGS method for symmetric nonlinear equations,* SIAM J. Numer. Anal. 37 (1999), 152–172.

15. Marquardt, D.W. *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math. 11 (1963) 431–441.

16. Meintjes, K. and Morgan, A.P. *A methodology for solving chemical equilibrium systems*, Appl. Math. Comput. 22 (1987), 333–361.

17. Musa, Y. B., Waziri, M.Y., and Halilu, A.S. *On computing the regularization Parameter for the Levenberg-Marquardt method via the spectral radius approach to solving systems of nonlinear equations,*. J. Numer. Math. Stoch. 9 (2017), 80–94.

18. Petrović, M.J. *An accelerated double step size model in unconstrained optimization,* Appl. Math. Comput. 250 (2015), 309–319.

19. Petrović, M.J. and Stanimirović, P.S. *Accelerated double direction method for solving unconstrained optimization problems*, Math. Probl. Eng. 2014, Art. ID 965104, 8 pp.

20. Sun, M., Tian, M.Y. and Wang, Y.J. *Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization,* Discrete Dyn. Nat. Soc. 2019, Art. ID 4745759, 1–14.

21. Waziri, M.Y., Leong, W.J. and Hassan M.A. *Jacobian-free diagonal Newton's method for solving nonlinear systems with singular Jacobian,* Malays. J. Math. Sci. 5(2) (2011), 241–255.

22. Waziri, M.Y., Leong W.J., Hassan M.A. and Monsi M. *A new Newton's method with diagonal Jacobian approximation for system of nonlinear equations,* J. Math. Stat. 6 (2010) 246–252.

23. Waziri, M.Y., Leong W.J., Hassan, M.A, and Monsi, M. *A low memory solver for integral equations of Chandrasekhar type in the radiative transfer problems*, Math. Probl. Eng. 2011, Art. ID 467017, 12 pp.

24. Waziri, M.Y., and Sabiu, J. *A derivative-free conjugate gradient method and its global convergence for symmetric nonlinear equations,* Int. J. Math. Math. Sci. 2015, Art. ID 961487, 8 pp.

25. Xiao, Y.H. and Zhu, H. A *conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing,* J. Math. Anal. Appl. 405 (2013) 310–319.

26. Yuan, G. and Lu, X. *A new backtracking inexact BFGS method for symmetric nonlinear equations,* Comp. Math. App. 55 (2008) 116–129.

How to cite this article

A.I. Kiri, M.Y. Waziri and A.S. Halilu Modification of the double direction approach for solving systems of nonlinear equations with application to Chandrasekhar's Integral equation. *Iranian Journal of Numerical Analysis and Optimization*, 2022; 12(2): 426-448. doi: 10.22067/IJNAO.2022.74201.1083.