On the Picard-Mann approach for hybridizing the double direction method for solving a system of nonlinear equations

Aliyu Ibrahim Kiri[1], Mohammed Yusuf Waziri[2], and Abubakar Sani Halilu[3]
[1,2]Department of Mathematical Sciences, Bayero University, Kano, Nigeria.
[3]Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria.
[2,3]Numerical Optimization research Group, Bayero University, Kano, Nigeria.

Corresponding Author E-mail address: abubakars.halilu@slu.edu.ng

Abstract

In this article, the improvement of the numerical performance of the iterative scheme presented by Halilu and Waziri in [5] is considered. This is made possible by hybridizing it with Picard-Mann hybrid iterative process. In addition, the step length is calculated using the inexact line search technique. Under the preliminary conditions, the proposed method's global convergence is established. The numerical experiment shown in this paper depicts the efficiency of the proposed method, which improved the results than the double direction method [5], existing in the literature.
2010 Mathematics Subject Classification: Primary 65K05; Secondary 90C30, 90C53
Keywords: Acceleration parameter, , Jacobian matrix, Double direction method, Picard-Mann process.

1. Introduction

Systems of nonlinear equations usually arise in the areas of human endeavor such as sciences and engineering. Researchers are tasked with developing efficient and robust iterative methods to solve them. Typically, a system of nonlinear equations is represented as

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

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map. Throughout this paper, the space $\mathbb{R}^n$ denote the $n-$dimensional real space, $\|\cdot\|$ is the Euclidean norm and $F_k = F(x_k)$.

Some iterative approaches for solving these problems include derivative-free methods [5–9], Newton and quasi-Newton methods [2–4]. However, Newton's method is prominent due to its attractive features, such as easy implementation and rapid convergence. However, the method requires the computation as well as storage of Jacobian matrix at each iteration and generates a sequence of points using the recursive formula:

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

where, $s_k = x_{k+1} - x_k$ and α_k is a step length. The Newton's search direction d_k is determined by solving the following linear system of equations,

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

where, F_k' is the Jacobian matrix of F at x_k. However, in Newton's method, the derivative F' is computed at each iteration, which may be unavailable or could not be obtained precisely. 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 [4, 11], and its search direction is given by

$$d_k = -B_k^{-1} F_k, \tag{4}$$

where B_k is $n \times n$ matrix that approximate the Jacobian of F at x_k.

Moreover, (1) can be obtained from an unconstrained optimization problem [11]. Suppose f be a merit function defined by

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

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

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

Newton and quasi-Newton's methods require the computation of the Jacobian matrix or its approximation at each iteration, despite the attractive characteristics of these methods. Therefore, they are not ideal for solving large-scale problems because they require massive matrix storage at each iteration which is costly in numerical experiments. Matrix-free methods are proposed to overcome these problems. The double direction method is among the successful matrix-free methods [13], that generates a sequence of iterates via

$$x_{k+1} = x_k + \alpha_k d_k + \alpha_k^2 b_k, \tag{6}$$

where x_{k+1} is the current iterate, x_k is the previous iterate, while b_k and d_k are search directions respectively. The rationale behind double direction method is that, there are two corrections in the scheme (6), if one correction fails during iterative process then the second one will correct the system.

In [13], Petrovic and Stanimirovic proposed a double direction method for solving unconstrained optimization problems. In their work, an approximation to Hessian matrix is obtained via acceleration parameter $\gamma_k > 0$, i.e.,

$$\bigtriangledown^2 f(x_k) \approx \gamma_k I,$$

where I is an identity matrix. The sequence of iterates $\{x_k\}$ is generated using (6). Petrovic further improves the performance of double direction in [14], where the double step length scheme for the unconstrained optimization problem is presented as:

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

where, α_k and β_k are two different step lengths. The Numerical results reported in [14] have shown that the proposed method is quite effective compared to the double direction method in [13]. Because it has the number of iterations and CPU time than the compared method [13]. Moreover, to improve the convergence properties and numerical results of the double direction methods, Petrović et al. [15] hybridized the double direction method for unconstrained optimization problem in [13], with Picard-Mann hybrid iterative process proposed by Khan in [1]. The Picard-Mann hybrid iterative process is defined as three relations:

Definition 1.1 *The Picard-Mann hybrid iterative process is defined as three relations:*

$$x_1 = x \in \mathbb{R}^n, \tag{8}$$
$$y_k = (1 - \eta_k)x_k + \eta_k T x_k, \tag{9}$$
$$x_{k+1} = T y_k, \quad k \in \mathbb{N}, \tag{10}$$

where $T : \Omega \longrightarrow \Omega$ is a mapping defined on nonempty convex subset Ω of a normed space $\mathbf{E}$*, y_k and x_k are sequences determined by the iterations in* (9) *and* (10)*, and $\{\eta_k\}$ is the sequence of positive numbers in* $(0, 1)$.

In this paper, η_k denotes the correction parameter.

Since the research of derivative-free double direction methods for solving systems of nonlinear equations is scarce in the literature, this motivated Halilu and Waziri [5] to use the scheme in (6) and proposed a derivative-free method via double direction approach for solving system of nonlinear equations. The method is proved to be globally convergent by assuming that the Jacobian of F is bounded and positive definite. Abdullahi et al. [9] further improved the performance of the double direction scheme where they modified the idea in [5] based on conjugate gradient approach to solve symmetric nonlinear equations. The method converged globally using the derivative-free line search proposed by Li and Fukushima in [11]. Recently, Halilu and Waziri [10] solved the system of nonlinear equations by improving the double direction iteration approach in (6). The global convergence of the method was established under some mild conditions, and the numerical experiments demonstrated in the paper showed that the proposed method is very efficient.

Motivated by the hybridization method presented in [15], This article is aimed at hybridizing the double direction method in [5] with the Picard-Mann hybrid iterative process proposed by Khan [1]. The paper is organized as follows. In the next section, we will present the algorithm of the proposed method. Section 3 presents the proposed algorithm's convergence analysis. Section 4 lists some numerical experiments. The article concluded in section 4.

2. Main result

Let us consider the derivative-free double direction method in [5]. The method developed a derivative-free method for solving systems of nonlinear equations via

$$F_k' \approx \gamma_k I, \tag{11}$$

where I is an identity matrix and $\gamma_k > 0$ is an acceleration parameter. The method in [5] produces a sequence of iterates $\{x_k\}$ such that $x_{k+1} = x_k + s_k$, where $s_k + (\alpha_k + \alpha_k^2\gamma_k)d_k$ and the direction d_k is given as

$$d_k = -\gamma_k^{-1}F_k. \tag{12}$$

The acceleration parameter is obtained by using first-order Taylor's expansion as

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2\gamma_k)y_k^T d_k}, \tag{13}$$

where $y_{k-1} = F_k - F_{k-1}$.

Although the method [5] has strong convergence properties, its numerical performance is weak when γ_k approaches or is equal to 1. For this reason, we are motivated to propose a hybrid method with good numerical results. To define a hybrid form of the method in [5], the mapping T in definition (1.1) is assumed to be defined by an improved double direction method as $Ty_k = y_k - (\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k$. By this assumption and the definition (1.1) we have

$$x_1 = x \in \mathbb{R}^n,$$

$$y_k = (1-\eta_k)x_k + \eta_k T x_k = x_k - (\eta_k + 1)(\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \tag{14}$$

$$x_{k+1} = Ty_k = y_k - (\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \quad k \in \mathbb{N}. \tag{15}$$

From (14) and (15) we obtain the iterative scheme,

$$x_{k+1} = x_k - t_k(\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \tag{16}$$

where, $t_k = (\eta_k + 1) \in (1,2)$ is a correction parameter. we can easily show that, the search direction in (16) is defined as:

$$d_k = -t_k\gamma_k^{-1}F_k. \tag{17}$$

Next, the proposed method algorithm is specified as follows:

Algorithm 1: hybrid double direction method (HDDPM)

Input: Given x_0, $\gamma_0 = 1$, $\epsilon = 10^{-5}$, $\omega_1 > 0$, $\omega_2 > 0$ and $r \in (0,1)$, $t \in (1,2)$, set $k = 0$.
Step 1: Compute F_k.
Step 2: If $\|F_k\| \le \epsilon$ then stop; otherwise, proceed to Step 3.
Step 3: Compute search direction $d_k = -t\gamma_k^{-1}F_k$.
Step 4: Set $x_{k+1} = x_k + (\alpha_k + \alpha_k^2\gamma_k)d_k$, where, $\alpha_k = r^{m_k}$ with m_k being the smallest nonnegative integer m such that

$$f(x_k + (\alpha_k + \alpha_k^2\gamma_k)d_k) - f(x_k) \le -\omega_1\|\alpha_k F_k\|^2 - \omega_2\|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{18}$$

Let $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty}\eta_k < \eta < \infty. \tag{19}$$

Step 5: Compute F_{k+1}.
Step 6: Determine $\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2\gamma_k)y_k^T d_k}$.
Step 7: Consider $k = k+1$ and go to Step 2.

3. Convergence Analysis

We present how the proposed Algorithm 1 (HDDPM) converges globally in this section. To begin, let's define the level set.

$$\Omega = \{x | \|F(x)\| \leq \|F(x_0)\|\}. \quad (20)$$

Assumption 3.1 *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. i.e., there exist positive constants $H > h > 0$ such that*

$$\|F'(x)\| \leq H \quad \forall x \in Q, \quad (21)$$

and

$$h\|d\|^2 \leq d^T F'(x) d \quad \forall x \in Q, d \in \mathbb{R}^n. \quad (22)$$

Remark 3.2 *We make the following remark:*
Assumption 3.1 implies that there exist constants $H > h > 0$ such that

$$h\|d\| \leq \|F'(x)d\| \leq H\|d\| \quad \forall x \in Q, d \in \mathbb{R}^n. \quad (23)$$

$$h\|x - y\| \leq \|F(x) - F(y)\| \leq H\|x - y\| \quad \forall x, y \in Q. \quad (24)$$

Since $t^{-1}\gamma_k I$ approximates F_k' along s_k, the following assumption can be made.

Assumption 3.3 *$t^{-1}\gamma_k I$ is a good approximation to F_k', i.e.,*

$$\|(F_k' - t^{-1}\gamma_k I)d_k\| \leq \epsilon \|F_k\|, \quad (25)$$

where, $\varepsilon \in (0,1)$ is a small quantity [4].

Lemma 3.4 *Suppose Assumption 3.3 holds, and let $\{x_k\}$ be generated by Algorithm 1. Then d_k is a sufficient descent direction for $f(x_k)$ at x_k i.e.,*

$$\nabla f(x_k)^T d_k < c\|F_k\|^2, \;\; c > 0. \quad (26)$$

Proof From (5), (12), and (25), we have

$$\begin{aligned} \nabla f(x_k)^T d_k &= F_k^T F_k' d_k \\ &= F_k^T [(F_k' - t^{-1}\gamma_k I)d_k - F_k] \\ &= F_k^T (F_k' - t^{-1}\gamma_k I)d_k - \|F_k\|^2, \end{aligned} \quad (27)$$

by Chauchy-Schwarz we have,

$$\begin{aligned} \nabla f(x_k)^T d_k &\leq \|F_k\| \|(F_k' - t^{-1}\gamma_k I)d_k\| - \|F_k\|^2 \\ &\leq -(1-\epsilon)\|F_k\|^2. \end{aligned} \quad (28)$$

Since $\epsilon \in (0,1)$, taking $c = 1 - \epsilon$, this lemma is true.
We can conclude from Lemma 3.4 that the norm function $f(x_k)$ is a descent along d_k, which means that $\|F_{k+1}\| \leq \|F_k\|$ is true. ■

Lemma 3.5 *Suppose that Assumption 3.1 holds and let $\{x_k\}$ be generated by Algorithm 1. Then $\{x_k\} \subset \Omega$.*

Proof From Lemma 3.4, we have $\|F_{k+1}\| \leq \|F_k\|$. Furthermore, for all k,

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

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

Lemma 3.6 *Suppose Assumption 3.1 holds and $\{x_k\}$ be generated by Algorithm 1. Then there exists a constant $m > 0$ such that for all k,*

$$y_k^T s_k \geq h\|s_k\|^2. \tag{29}$$

Proof By mean-value theorem and (22),
$y_k^T s_k = s_k^T (F(x_{k+1}) - F(x_k)) = s_k^T F'(\xi) s_k \geq h\|s_k\|^2$.
Where $\xi = x_k + \zeta(x_{k+1} - x_k)$, $\zeta \in (0,1)$.
Using $y_k^T s_k \geq h\|s_k\|^2 > 0$, γ_{k+1} is always generated by the update formula (**??**). Therefore, $\gamma_{k+1} I$ inherits the positive definiteness of $\gamma_k I$. From Lemma 3 and (**??**), the following inequality holds.

$$\frac{y_k^T s_k}{\|s_k\|^2} \geq h, \qquad \frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{H^2}{h}. \tag{30}$$

■

Lemma 3.7 *Suppose that Assumption 3.1 holds and $\{x_k\}$ is generated by Algorithm 1. Then we have*

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

and

$$\lim_{k \to \infty} \|\alpha_k F_k\| = 0. \tag{32}$$

Proof From (18) for all $k > 0$

$$\begin{aligned} \omega_2 \|\alpha_k d_k\|^2 &\leq \omega_1 \|\alpha_k F_k\|^2 + \omega_2 \|\alpha_k d_k\|^2 \\ &\leq \|F_k\|^2 - \|F_{k+1}\|^2 + \eta_k \|F_k\|^2. \end{aligned} \tag{33}$$

By summing the above inequality, we have

$$
\begin{aligned}
\omega_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 &\leq \sum_{i=0}^{k} \left(\|F_i\|^2 - \|F_{i+1}\|^2 \right) + \sum_{i=0}^{k} \eta_i \|F_i\|^2, \\
&= \|F_0\|^2 - \|F_{k+1}\|^2 + \sum_{i=0}^{k} \eta_i \|F_i\|^2, \\
&\leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{k} \eta_i, \\
&\leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{\infty} \eta_i.
\end{aligned} \tag{34}
$$

From the level set and the fact that $\{\eta_k\}$ satisfies (19), then the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ converges. This implies (31). Using the same logic as above, but this time with $\omega_1 \|\alpha_k F_k\|^2$ on the left, we obtain (32). ■

Lemma 3.8 *Suppose Assumption 3.1 holds and let $\{x_k\}$ be generated by Algorithm 1. Then there exists a constant $M > 0$ such that for all $k > 0$,*

$$\|d_k\| \leq M. \tag{35}$$

Proof From (12) and (13) we have

$$
\begin{aligned}
\|d_k\| &= \left\| -t \frac{(\alpha_{k-1} + \alpha_{k-1}^2 \gamma_{k-1}) y_k^T d_k}{y_{k-1}^T y_{k-1}} \right\| \\
&= \left\| -\theta \frac{y_{k-1}^T s_{k-1} F_k}{\|y_{k-1}\|^2} \right\| \\
&\leq \frac{t \|F_k\| \|s_{k-1}\| \|y_{k-1}\|}{h^2 \|s_{k-1}\|^2} \\
&\leq \frac{t \|F_k\| H \|s_{k-1}\|}{h^2 \|s_{k-1}\|} \\
&\leq \frac{t \|F_k\| H}{h^2} \\
&\leq \frac{t \|F_0\| H}{h^2}.
\end{aligned} \tag{36}
$$

Taking $M = \frac{t\|F_0\|H}{h^2}$, we have (35). ■

Theorem 3.9 *Suppose that Assumption 3.1 holds and $\{x_k\}$ be generated by Algorithm 1. Assume further for all $k > 0$,*

$$\alpha_k \geq \lambda \frac{|F_k^T d_k|}{\|d_k\|^2}, \tag{37}$$

where λ is some positive constant. Then

$$\lim_{k \to \infty} \|F_k\| = 0. \tag{38}$$

Proof From Lemma 3.8 we have (35). Also, from (31) and the boundedness of $\{\|d_k\|\}$, we have

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

from (37) and (39) we have

$$\lim_{k\to\infty} |F_k^T d_k| = 0. \tag{40}$$

Also, from (12) we have,

$$F_k^T d_k = -t\gamma_k^{-1}\|F_k\|^2, \tag{41}$$

$$\begin{aligned} \|F_k\|^2 &= |-F_k^T d_k t^{-1}\gamma_k| \\ &\le t^{-1}|\gamma_k||F_k^T d_k|. \end{aligned} \tag{42}$$

Since

$$\gamma_k^{-1} = \frac{(\alpha_{k-1} + \alpha_{k-1}^2\gamma_{k-1})y_k^T d_k}{y_{k-1}^T y_{k-1}} = \frac{y_{k-1}^T s_{k-1}}{\|y_{k-1}\|^2} \ge \frac{h\|s_{k-1}\|^2}{\|y_{k-1}\|^2} \ge \frac{h\|s_{k-1}\|^2}{H^2\|s_{k-1}\|^2} = \frac{h}{H^2}.$$

Then,

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

Therefore from (42) we have,

$$\|F_k\|^2 \le |F_k^T d_k| \left(\frac{H^2}{th}\right). \tag{43}$$

As a result,

$$0 \le \|F_k\|^2 \le |F_k^T d_k| \left(\frac{H^2}{th}\right) \longrightarrow 0. \tag{44}$$

Hence,

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{45}$$

The proof is completed. ■

4. Numerical Experiments

In this section, we test the efficiency and robustness of our proposed method (HDDPM) using the following existing methods in the literature:

- An improved derivative-free method via double direction approach for solving systems of nonlinear equations (IDFDD) [5].

The computer codes utilized 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 two algorithms were implemented with the same line search (18) in the experiments, and the following parameters are set: $\omega_1 = \omega_2 = 10^{-4}$, $r = 0.2$, and $\eta_k = \frac{1}{(k+1)^2}$, as they are taken in [5]. We, however, set $t = 1.2$ in our algorithm. The program execution is stopped if the total number of iterations exceeds 1000 or $\|F_k\| \leq 10^{-5}$. To show the extensive numerical experiments of HDDPM and IDFDD methods, we have tried these methods on the previous three Benchmark test problems with different initial points and dimensions (n values) between 1000 and 100,000.

Problem 1 [7]
$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2, \quad i = 1,2,...,n.$

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

Problem 3 [8]
$F_1 = x_1 - e^{\cos\left(\frac{x_1 + x_2}{n+1}\right)},$
$F_i = x_i - e^{\cos\left(\frac{x_{i-1} + x_i + x_{i+1}}{n+1}\right)},$
$F_n = x_n - e^{\cos\left(\frac{x_{n-1} + x_n}{n+1}\right)}, \quad i = 2,3,...,n-1.$

Table 1: Initial points

INITIAL POINTS (IP)	VALUES
$x1$	$\left(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\right)^T$
$x2$	$\left(\frac{1}{5}, \frac{1}{5}, ..., \frac{1}{5}\right)^T$
$x3$	$\left(\frac{3}{2}, \frac{3}{2}, ..., \frac{3}{2}\right)^T$
$x4$	$\left(\frac{2}{5}, \frac{2}{5}, ..., \frac{2}{5}\right)^T$
$x5$	$\left(0, \frac{1}{2}, \frac{2}{3}, ..., 1 - \frac{1}{n}\right)^T$
$x6$	$\left(\frac{1}{4}, \frac{-1}{4}, ..., \frac{(-1)^n}{4}\right)^T$
$x7$	$\left(1, \frac{1}{2}, \frac{1}{3}, ..., \frac{1}{n}\right)^T.$

Table 2: Numerical results of Problem 1

		HDDPM			IDFDD		
Dimension	IP	ITER	TIME	$\|\|F_k\|\|$	ITER	TIME	$\|\|F_k\|\|$
1000	X1	25	0.030161	6.33E-06	28	0.025821	9.52E-06
	X2	18	0.013831	9.53E-06	22	0.016813	7.51E-06
	X3	26	0.008954	7.04E-06	32	0.014239	9.18E-06
	X4	26	0.009624	9.48E-06	30	0.009129	7.91E-06
	X5	35	0.014716	9.12E-06	71	0.025831	7.79E-06
	X6	28	0.011728	7.73E-06	32	0.016323	9.85E-06
	X7	43	0.021941	8.47E-06	43	0.012281	5.72E-06
10,000	X1	27	0.074013	6.45E-06	31	0.074321	7.89E-06
	X2	20	0.054277	9.72E-06	24	0.055588	9.72E-06
	X3	28	0.082364	7.18E-06	35	0.091056	7.61E-06
	X4	28	0.065391	9.67E-06	33	0.075691	6.55E-06
	X5	35	0.098831	8.85E-06	70	0.122021	8.45E-06
	X6	30	0.069162	7.89E-06	35	0.079785	8.17E-06
	X7	45	0.107822	7.78E-06	44	0.091102	8E-06
100,000	X1	29	0.610661	6.59E-06	34	0.733469	6.54E-06
	X2	22	0.466959	9.92E-06	27	0.606211	8.06E-06
	X3	30	0.634731	7.33E-06	37	0.834964	9.86E-06
	X4	30	0.642824	9.87E-06	35	0.743687	8.49E-06
	X5	35	0.773071	8.81E-06	72	1.379716	8.15E-06
	X6	32	0.699817	8.05E-06	38	0.796549	6.77E-06
	X7	46	0.992078	9.97E-06	46	0.962775	9.52E-06

Table 3: Numerical results of Problem 2

		HDDPM			IDFDD		
Dimension	IP	ITER	TIME	$\|\|F_k\|\|$	ITER	TIME	$\|\|F_k\|\|$
1000	X1	27	0.013652	5.9E-06	35	0.012921	8.26E-06
	X2	24	0.008641	8.37E-06	34	0.017416	7.33E-06
	X3	29	0.012786	6.99E-06	37	0.022244	8.32E-06
	X4	26	0.013085	7.56E-06	35	0.018312	7.01E-06
	X5	28	0.010574	8.16E-06	36	0.017413	9.28E-06
	X6	24	0.012878	6.97E-06	30	0.016861	9.54E-06
	X7	23	0.010061	7.07E-06	33	0.012262	7.05E-06
10,000	X1	29	0.091644	6.02E-06	38	0.111675	6.85E-06
	X2	26	0.082632	8.54E-06	36	0.108409	9.5E-06
	X3	31	0.100012	7.14E-06	40	0.120401	6.9E-06
	X4	28	0.100563	7.71E-06	37	0.116236	9.09E-06
	X5	30	0.106658	8.38E-06	39	0.120113	7.73E-06
	X6	26	0.082597	7.11E-06	33	0.103787	7.92E-06
	X7	24	0.080271	7.28E-06	35	0.096995	8.76E-06
100,000	X1	31	0.716661	6.15E-06	40	0.856886	8.88E-06
	X2	28	0.673629	8.72E-06	39	0.831083	7.88E-06
	X3	33	0.713915	7.29E-06	42	0.915424	8.94E-06
	X4	30	0.629491	7.87E-06	40	0.842118	7.54E-06
	X5	32	0.689811	8.56E-06	42	0.883616	6.41E-06
	X6	28	0.600796	7.26E-06	36	0.773243	6.56E-06
	X7	26	0.576332	6.63E-06	38	0.814174	7.23E-06

Table 4: Numerical results of Problem 3

		HDDPM			IDFDD		
Dimension	IP	ITER	TIME	$\|F_k\|$	ITER	TIME	$\|F_k\|$
1000	X1	47	0.022701	8.17E-06	96	0.043566	8.59E-06
	X2	47	0.024558	9.28E-06	96	0.042647	9.76E-06
	X3	45	0.025712	8.86E-06	94	0.049323	8.17E-06
	X4	47	0.024715	8.54E-06	96	0.044737	8.98E-06
	X5	46	0.023917	8.93E-06	95	0.044291	8.8E-06
	X6	48	0.030872	7.79E-06	97	0.043297	8.74E-06
	X7	47	0.022964	9.99E-06	97	0.045644	7.98E-06
10,000	X1	50	0.158942	9.33E-06	111	0.283166	9.05E-06
	X2	51	0.157945	7.54E-06	112	0.302464	7.81E-06
	X3	49	0.157441	7.2E-06	109	0.270956	8.6E-06
	X4	50	0.155546	9.75E-06	111	0.286364	9.45E-06
	X5	50	0.154582	7.23E-06	110	0.281598	9.23E-06
	X6	51	0.157617	8.89E-06	112	0.290812	9.2E-06
	X7	51	0.156528	8.14E-06	112	0.283862	8.42E-06
100,000	X1	54	1.545911	7.58E-06	115	2.875245	9.54E-06
	X2	54	1.495309	8.61E-06	116	2.845327	8.23E-06
	X3	52	1.448103	8.22E-06	113	2.723431	9.07E-06
	X4	54	1.506604	7.93E-06	115	2.806242	9.97E-06
	X5	53	1.492609	8.25E-06	114	2.792279	9.73E-06
	X6	55	1.539513	7.23E-06	116	2.868197	9.71E-06
	X7	54	1.493281	9.29E-06	116	2.871967	8.89E-06

Tables (2-4) above reported the numerical results of the two methods, where 'ITER' and 'TIME' stand for the number of iterations and the CPU time (in seconds), respectively, while $\|F_k\|$ is the norm of the residual at the stopping point. From the Tables, HDDPM and IDFDD methods attempt to solve the problem (1), but it is clear that the HDDPM method outperforms the IDFDD method. In particular, the HDDPM method considerably outperforms the IDFDD for almost all the tested problems, as it has the least iteration and CPU time than the IDFDD method. Due to the contribution of the computation of correction parameter at each iteration. Thus, the proposed method successfully solves the large-scale system of nonlinear equations.

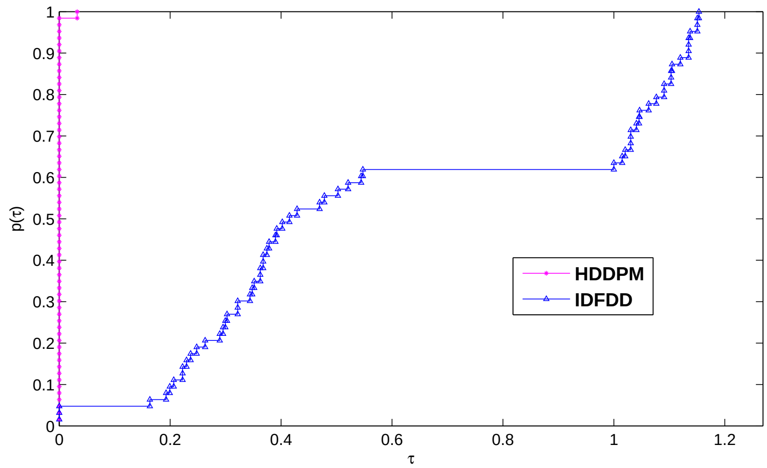

Figure 1: Performance profile with respect to the number of iterations

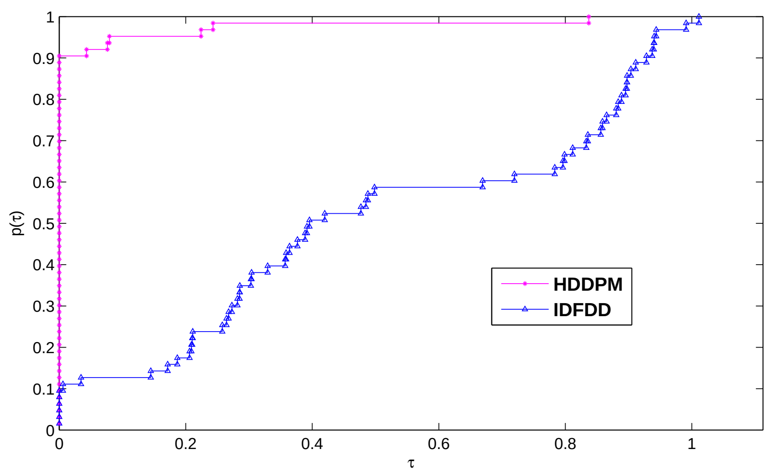

Figure 2: Performance profile with respect to the CPU time (in second)

Using the performance profile of Dolan and More [12], we generate Figures 1 and 2 to show the performance and efficiency of each of the three methods. That is, for each method, we plot the fraction $P(\tau)$ of the problems for which the method is within a factor τ of the best time. Figures 1 and 2 show that the curves corresponding to the HDDPM method stay above the other curve representing the IDFDD method. This indicates that the proposed method outperforms the compared method in terms of fewer iterations and CPU time (in second), and hence, it is the most efficient. Finally, it is clear from both Figures that our method effectively solves the large-scale nonlinear system of equations.

5. Conclusion

Hybridization of double direction method for solving system of nonlinear equations via Picard-Mann hybrid iterative process in [1] is presented in this work. This was achieved by modifying the method in [5] using the correction parameter. The proposed method is an entirely derivative-free iterative method, which is why it is more efficient in solving large-scale problems. Numerical comparisons have been made using a set of large-scale test problems. In addition, Table (2-4) and Figure (1-2) have shown that the proposed method is very efficient because it has the least iteration and CPU time compared to the IDFDD method. In future research, the idea proposed in this scheme will be applied to solve the monotone nonlinear equations with application in compressive sensing.

Conflict of Interests

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

References

[1] Safeer H.K. :A Picard-Mann hybrid iterative process, Fixed Point Theory and Applications. 2013; 69: 10 pages.

[2] Dennis J.E and Schnabel R.B.: Numerical Methods for Unconstrained Optimization and NonLinear Equations. Prentice Hall, Englewood Cliffs, NJ, 1983.

[3] Waziri M.Y., Leong W.J., Hassan M.A. Jacobian-Free Diagonal Newton's Method for Solving Nonlinear Systems with Singular Jacobian. Malay. J. Mathl. Sci. 2011; 5: 241–255.

[4] Yuan G, Lu X.: A new backtracking inexact BFGS method for symmetric nonlinear equations. Comp. Math. App. 2008; 55: 116–129.

[5] Halilu A.S. and Waziri M.Y.: An improved derivative-free method via double direction approach for solving systems of nonlinear equations, J of the Ramanujan Mathl. Society, 2018; 33: 75–89.

[6] Halilu A.S. and Waziri M.Y.: Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations. Int J app Math Res. 2017; 6: 147–156.

[7] Halilu A.S., and Waziri M.Y.: Inexact Double Step Length Method for Solving Systems of Nonlinear Equations. Stat., Optim. Inf. Comput., 2020; 8: 165–174.

[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. 2017; 9: 20–32.

[9] 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. Num Math Stoch. 2018; 10(1): 32–44.

[10] Halilu, A.S. and Waziri, M.Y.: Solving systems of nonlinear equations using improved double direction method, J of the Nigerian Mathl. Soc. 2020; 32(2): 287–301.

[11] Li D., Fukushima M.: A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equation. SIAM J Numer Anal 1999; 37: 152-172.

[12] Dolan, E. and Moré, J.:Benchmarking opyimization software with performance profiles, Journal of Mathematical program, 2002; 91: 201–213.

[13] Petrovic, M.J. and Stanimirovic, P.S.: Accelerated double direction method for solving unconstrained optimization problems. Mathematical Problems in Engineering 2014; 2014: 1-8.

[14] Petrovic, M.J.: An accelerated double step size model in unconstrained optimization, Applied mathematics and computation, 2015; 250: 309–319.

[15] Petrović, M.J., Stanimirović, P.S., Kontrec, N., Mladenović, J. Hybrid Modification of accelerated double direction method. Math. Probl. Eng. 2018; Article D 1523267: 8 pages. doi.org/10.1155/2018/1523267.