Filomat 37:5 (2023), 1461–1478
https://doi.org/10.2298/FIL2305461W

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A family of hybrid derivative-free methods via acceleration parameter for solving system of nonlinear equations

Mohammed Yusuf Waziri[a,c], Hadiza Usman Muhammad[a,c], Abubakar Sani Halilu[b,c,*], Kabiru Ahmed[a,c]

[a]*Department of Mathematical Sciences, Bayero University, Kano, Nigeria*
[b]*Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria*
[c]*Numerical Optimization Research Group, Department of Mathematical Sciences, Bayero University, Kano, Nigeria*

Abstract. In this paper, we present some derivative-free methods for solving system of nonlinear equations based on approximating the Jacobian matrix via acceleration and correction parameters. Furthermore, we compute the step length using inexact line search procedure. Under appropriate conditions, we proved that the proposed methods are globally. We also present some numerical results to show the efficiency of the proposed methods by comparing them with some existing derivative-free methods in the recent literature.

1. Introduction

Problems involving system of nonlinear equations usually arise in areas of human endeavor such as sciences and engineering, and as such, 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 $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is a continuously differentiable mapping. As stated above, systems of nonlinear equations have wide applications, and a clear case is presented in [49, 50], where an economic equilibrium problem is reformulated as (1). Hayat et al. [38] discussed the impact of Cattaneo-Christov heat flow in the stagnation point flow of rate type fluids, which is a phenomena that is modeled in the form of (1). Also, Hayat et al. [37] considered the characteristics of variable thermal conductivity and thermal relaxation in stagnation flow over a variable thickness stretched surface with chemical reaction. The study described involves a mathematical model in the form of (1). In [45], the authors discussed variable separation solutions from positive-power ansatz, by constructing nonlinear models, which involves equations in the form of (1). In the same vain, non-linear difference equations, which appear in modern textile engineering, and are used to describe phenomena in engineering are usually solved by discretizing into the form of (1) as presented in [44]. Studies similar to [44, 45] can be found in [46–48]. Several iterative methods for solving (1) include, derivative-free methods [6, 14, 23, 34, 51], double step length methods [12, 21, 22, 33, 43], double direction methods [13, 20, 29], Newton's methods and its improved version, i.e., the quasi-Newton methods

2020 *Mathematics Subject Classification*. Primary 65K05; Secondary 90C30, 90C53
Keywords. system of nonlinear equations, Newton method, Derivative-free method, hybrid approach, Global convergence
Received: 29 September 2020; Revised: 28 February 2021; Accepted: 29 April 2021
Communicated by Predrag Stanimirović
* Corresponding author: Abubakar Sani Halilu
Email address: abubakars.halilu@slu.edu.ng (Abubakar Sani Halilu)

[3, 4, 8, 15]. But the prominent among them is the Newton's method due to its attractive features such as easy implementation and rapid convergence. However, it requires the computation of the Jacobian matrix at each iteration and generates a sequence of points using the recursive formula:

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

where k=0,1,2,.., α_k is a step length to be computed by a suitable line search technique, x_{k+1} represents a current iterate and x_k is the previous iterate, while d_k is the search direction that can be calculated by solving the following system of linear equation,

$$F'(x_k)d_k = -F(x_k), \tag{3}$$

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

Furthermore, (1) can come from an unconstrained optimization problem, a saddle point and equality constrained problem [3]. Let f be a merit function defined by

$$f(x) = \frac{1}{2}\|F(x)\|^2. \tag{4}$$

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

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

Generally, the search direction d_k is required to satisfy the descent condition

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

The step length α_k can be determined in different ways either by exact or inexact line search technique. The most commonly used line search in practice is the inexact line search as proposed in [6, 9, 15, 17], which sufficiently decreases the function values i.e to establish

$$\|F(x_k + \alpha_k d_k)\| \leq \|F(x_k)\|. \tag{5}$$

However, it is known that in Newton's method, the computation of partial derivatives of some functions are very expensive in practice and sometimes are not even available. In such cases, Newton's method cannot be used directly [8]. To overcome these shortcomings, some methods have been proposed over the years. This includes the spectral gradient method proposed in [27], which is easy to implement and also efficient for large-scale problems. The method in [27] was extended by Cruz and Raydan [38] to large-scale systems of nonlinear equations by introducing a spectral algorithm known in short as(SANE). The scheme converges globally by means of a variation of the nonmonotone line search strategy of Grippo et al. [39]. Similarly, Zhang and Zhou [41] developed a method for solving nonlinear monotone equations by combining the spectral gradient method [27] with the projection method by Solodov and Svaiter [40]. The method converges globally, when the nonlinear equations are Lipschitz continuous. Only recently, Waziri et al. [28, 42] proposed two conjugate gradient (CG) methods for systems of nonlinear equations. They generate descent search directions and the authors proved their global convergence under mild conditions.

To improve the performance of some CG methods, hybrid aproaches have been studied by many researchers in the past decade (see Refs. [52–54] for instances). One type of hybrid CG methods are obtained by constructing a new update parameter as a linear combination of two or more classical or modified CG update parameters. By employing the classical Fletcher-Reeves (FR) [55] method and the method by Wei et al (WYL) [57] as a linear combination, Gonglin [58] proposed a hybrid method for unconstrained optimization, which is defined as

$$\beta_k^H = \lambda_1 \beta_k^{WYL} + \lambda_2 \beta_k^{FR}, \quad \lambda_1 \geq 0, \lambda_2 \geq 0. \tag{6}$$

Here, $\beta_k^{WYL} = \frac{g_{k+1}^T\left(g_{k+1}-\left(\frac{\|g_{k+1}\|}{\|g_k\|}\right)g_k\right)}{\|g_k\|^2}$, $\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$ with $g_k = \nabla f(x_k)$.
In [59], Xu and Kong presented a hybrid method by implementing a linear combination of the update parameters by Dai and Yuan (DY) [65] and Hestenes and Stiefel (HS) [63], i.e.,

$$\beta_k = \alpha_1\beta_k^{DY} + \alpha_2\beta_k^{HS}, \tag{7}$$

where

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \quad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \tag{8}$$

and α_1 and α_2 are nonnegative numbers with both or at least one not equal to zero. In addition, they satisfy the following

$$0 < \alpha_1 < 2\alpha_2 < \frac{1}{1+\sigma_2} < 1, \quad 0 \le \sigma_2 < 1. \tag{9}$$

Another type of hybrid CG methods are developed as convex combinations of other CG methods. In [56], Liu and Li proposed a hybrid CG method for unconstrained optimization, where the parameter is a convex combination of the classical Liu and Storey (LS) [61] and Dai and Yuan (DY) [65] update parameters, namely

$$\beta_k = (1-\theta_k)\beta_k^{LS} + \theta_k\beta_k^{DY}, \tag{10}$$

where β_k^{DY} is as defined in (8) and

$$\beta_k^{LS} = -\frac{g_{k+1}^T y_k}{d_k^T g_k}, \quad y_k = g_{k+1} - g_k, \quad \theta_k \in [0,1]. \tag{11}$$

Appropriate value of the parameter θ_k in the convex combination is chosen and search direction d_k generated by the method turns out to be the Newton direction, which also satisfies the popular Dai-Liao (DL) [64] conjugacy condition and the sufficient descent condition, namely

$$d_k^T y_k = -ts_k^T g_{k+1}, \quad t \ge 0, \tag{12}$$

and

$$d_k^T g_k = -c\|g_k\|^2, \quad c \ge 0, \tag{13}$$

where (13) is independent of the line search procedure employed.

The importance and contribution of this article is to develop hybrid derivative-free methods for solving (1) since such methods for systems of nonlinear equations are rare in the literature. In [18], an accelerated gradient descent method (SM) method, is presented with the iterative scheme given by

$$x_{k+1} = x_k - \gamma_k^{-1} t_k g_k \tag{14}$$

where g_k is the gradient of the function F at x_k and γ_k represent the acceleration parameter which is a scaler approximation of the Hessian and given by

$$\gamma_{k+1} = 2\gamma_k \frac{\gamma_k[f(x_{k+1}) - f(x_k)] + t_k\|g_k\|^2}{t_k^2\|g_k\|^2}, \tag{15}$$

where the step length t_k is computed by the Armijo's backtracking inexact line search technique. Accelerated double step size scheme is primarily proposed in [21], where, the Hessian matrix is approximated with diagonal matrix via acceleration parameter. The authors proved global convergence of the scheme under suitable conditions. In, the addition preliminary numerical result has shown that the method in [21] is very effective. Consequently, the authors in [12], incorporated the idea used in [21], and presented a transformed double step length method for solving large-scale systems of nonlinear equations. This is made possible by approximating the Jacobian with diagonal matrix via acceleration parameter. This method is a new approach that reduces the two step lengths into a single one. Furthermore, the scheme employs a derivative-free line search technique proposed in [3], to compute the step length. The proposed acceleration parameter presented in [12] is defined as

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \frac{1}{2}\alpha_k \gamma_k) y_k^T d_k}. \tag{16}$$

However, an improved derivative-free double direction method for systems of nonlinear equations has been presented in [13], where the Jacobian matrix is approximated via acceleration parameter given as

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2 \gamma_k) y_k^T d_k}. \tag{17}$$

This paper is organized as follows: In the next section, the hybrid optimization models are presented. Section 3 deals with derivation of the schemes and their algorithms. In Section 4, we analyze global convergence of the methods, while numerical results of some experiments conducted are presented in section 5. Concluding remarks are made in section 6.

2. Hybrid optimization models

In this section, we consider Picard-Mann hybrid iterative process presented in [7], where the Picard-Mann hybrid iterative process is defined as

$$\begin{cases} x_1 = x \in \mathbb{R}^n \\ w_k = (1-\beta_k)x_k + \beta_k T x_k, \\ x_{k+1} = T w_k, \end{cases} \tag{18}$$

where $T : C \longrightarrow C$ is a mapping defined on nonempty convex subset C of a normed space $\mathbb{E}$, x_k and w_k are sequences determined by the iteration (18) and β_k is the sequence of positive numbers in (0,1). In this paper, β_k is denoted as correction parameter.

The authors in [7] choose a constant value $\beta = \beta_k \in (0,1)$ as the correction parameter, $\forall k$, and also showed that the process converges faster than the Picard, Mann and Ishikawa iterative process [10, 11, 26]. These three mentioned schemes are defined with the next sets of relations, respectively:
The Picard iterative process [26] is defined by the sequence $\{u_k\}$ as

$$\begin{cases} u_1 = u \in \mathbb{C} \\ u_{k+1} = T u_k, k \in \mathbb{N}. \end{cases}$$

The Mann iterative process [11] is defined by the sequence $\{v_k\}$ as

$$\begin{cases} v_1 = v \in \mathbb{C} \\ v_{k+1} = (1-\alpha_k)v_k + \alpha_k T v_k, k \in \mathbb{N}. \end{cases}$$

where $\{\alpha_k\} \in (0,1)$ and

$$\begin{cases} z_1 = 1 \in \mathbb{C} \\ z_{k+1} = (1-\alpha_k)z_k + \alpha_k T y_k, \\ y_k = (1-\beta_k)z_k + \beta_k T z_k, k \in \mathbb{N}. \end{cases}$$

where, y_k and z_k are the sequences defined by the proposed expressions and $\{\beta_k\}$ and $\{\alpha_k\}$ are the sequences of positive numbers [10] which satisfy the conditions

- $0 \leq \alpha_k \leq \beta_k \leq 1, k \geq 0$
- $\lim_{k\to\infty} \beta_k = 0$
- $\sum_{k=0}^{\infty} \alpha_k\beta_k = \infty$

In [19], the hybridization of the (SM) method [18] with the Picard-Mann hybrid iterative process is presented. In addition, the authors determined the accelerated parameter γ_k using the Taylor's series expansion of the second order as

$$\gamma_{k+1} = 2\gamma_k \frac{\gamma_k[f(x_{k+1}) - f(x_k)] + (\beta_k + 1)\alpha_k\|g_k\|^2}{(\beta_k + 1)^2\alpha_k^2\|g_k\|^2}, \quad \{\alpha_k\}, \{\beta_k\} \in (0, 1). \tag{19}$$

Moreover, the step length t_k is computed using the inexact backtracking line search technique. The numerical results presented in [19], have shown that the proposed method is more efficient than the SM method [18], because it has the least number of iterations, CPU time, and number of function evaluation. Moreover, in order to improve the numerical performance of the scheme in [21], it was hybridized [1] with the Picard-Mann hybrid approach [7]. The proposed hybrid method [1] was shown to be numerically effective by comparing it with the double direction method [21] existing in the literature. In [2], the Picard-Mann hybrid approach was also applied to a transformation of accelerated double step size method for unconstrained optimization presented in [5]. In addition, the proposed method [2] was shown to be globally convergent under the assumption that the gradient of the objective function is Lipschitz continuous in an open convex set. Furthermore, the numerical experiments reported in [2] have shown that the proposed method produced much better results than the method in [5].

Motivated by [19], we incorporate the idea to system of nonlinear equations in order to develop a derivative-free method with the Picard-Mann hybrid iterative process via

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

where I is an identity matrix and $F'(x_k)$ is the Jacobian matrix of $F(x_k)$ at x_k. The presented method has a norm descent property without computing the Jacobian matrix with less number of iterations and CPU time that is globally convergent.

3. Derivation of the Methods and their Algorithms

In this section, we present algorithms of our proposed methods. By using (18) and the mapping T is defined as $Tx_k = x_k - \alpha_k\gamma_k^{-1}F(x_k)$, we have

$$x_1 = x \in \mathbb{R}^n. \tag{20}$$

$$w_k = (1 - \beta_k)x_k + \beta_k Tx_k. \tag{21}$$

$$x_{k+1} = Tw_k. \tag{22}$$

Therefore,

$$w_k = x_k - \beta_k\alpha_k\gamma_k^{-1}F(x_k), \tag{23}$$

by substituting (23) in (22), we obtain

$$x_{k+1} = x_k - \alpha_k(\beta_k + 1)\gamma_k^{-1}F(x_k), \tag{24}$$

where γ_k and β_k are acceleration and correction parameters respectively. From (24) we can define our first proposed direction as:

$$d_k^{(1)} = -\beta\gamma_k^{-1}F(x_k), \tag{25}$$

where $\beta = (\beta_k + 1) \in (1, 2)$.
From (24) and (25) we present the general scheme as:

$$x_{k+1} = x_k + \alpha_k d_k^{(1)}. \tag{26}$$

Now, to find γ_{k+1}, we consider the Taylor's series expansion of order 1 at x_{k+1} as

$$F(x_{k+1}) = F(x_k) + F'(\xi)(x_{k+1} - x_k), \tag{27}$$

where $\xi_k \in (x_k, x_{k+1})$. The distance between x_k and x_{k+1} is small enough and $\xi_k = x_k + \rho(x_{k+1} - x_k)$, $\rho \in [0, 1]$, we take $\rho = 1$ such that $\xi_k = x_{k+1}$. Therefore, we assume that

$$F'(\xi) \approx \gamma_{k+1}I. \tag{28}$$

By substituting (28) in (27), we obtain

$$F(x_{k+1}) - F(x_k) = \gamma_{k+1}(x_{k+1} - x_k), \tag{29}$$

where $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = (x_{k+1} - x_k) = -\alpha_k(\beta_k + 1)\gamma_k^{-1}F(x_k)$ such that

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

by multiplying both side of (30) by y_k^T, we obtain the proposed acceleration parameter as

$$\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}. \tag{31}$$

To compute the step-length α_k, we use the derivative-free line search proposed in [3]. Let $\omega_1 > 0$, $\omega_2 > 0$ and $r \in (0, 1)$ be constants and let η_k be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty \tag{32}$$

and

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1\|\alpha_k F(x_k)\|^2 - \omega_2\|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{33}$$

Let i_k be the smallest non negative integer i such that (33) holds for $\alpha = r^i$. Let $\alpha_k = r_k^i$.

Algorithm 1: (HDAP1)

Input: Given $x_0, \gamma_0 = 1, \epsilon > 0, \beta \in (1, 2)$, set $k = 0$.
Step 1: Compute $F(x_k)$.
Step 2: If $\|F(x_k)\| \leq \epsilon$, stop, else goto Step 3.
Step 3: Compute $d_k^{(1)}$ (using (25)).
Step 4: Compute step length α_k (using (33)).
Step 5: Set $x_{k+1} = x_k + \alpha_k d_k^{(1)}$.
Step 6: Compute $F(x_{k+1})$.
Step 7: Determine $\gamma_{k+1} = \dfrac{y_k^T y_k}{y_k^T s_k}$.
Step 8: Set $k = k + 1$, and go to Step 2.

The Proposed second choice of the correction parameter
Going by Barzilai and Borwein [27], and considering

$$\gamma_k^{BB} = \frac{s_k^T y_k}{s_k^T s_k}, \tag{34}$$

we adopt (34) to be our correction parameter β_k i.e

$$\beta_k = \frac{s_k^T y_k}{s_k^T s_k}, \tag{35}$$

where $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$. Substituting (35) in (24) gives

$$x_{k+1} = x_k - \alpha_k \left(\frac{s_k^T y_k}{s_k^T s_k} + 1 \right) \gamma_k^{-1} F(x_k), \tag{36}$$

and we propose a second search direction as

$$d_k^{(2)} = -\left(\frac{s_k^T y_k}{s_k^T s_k} + 1 \right) \gamma_k^{-1} F(x_k), \tag{37}$$

where, $\gamma_k = \frac{y_k^T y_k}{y_k^T s_k}$.

Algorithm 2: (HDAP2)

Input: Given $x_0, \gamma_0 = 1, \epsilon > 0, \beta_0 = 0.5$, set $k = 0$.
Step 1: Compute $F(x_k)$.
Step 2: If $\|F(x_k)\| \le \epsilon$, then stop, else goto Step 3.
Step 3: Compute $d_k^{(2)}$ (using (37)).
Step 4: Compute step length α_k (using (33)).
Step 5: Compute $x_{k+1} = x_k + \alpha_k d_k^{(2)}$.
Step 6: Compute $F(x_{k+1})$.
STEP 7: Determine $\beta_{k+1} = \frac{s_k^T y_k}{s_k^T s_k}$.
STEP 8: Determine $\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T s_k}$.
STEP 9: Set $k = k + 1$, and go to Step 2.

Remark 3.1. *For the correction parameter , if in some iterations the value for β_k is not in (0,1), then we take β_k to be equal to 0.5.*

4. Convergence Analysis

In this section, we present the global convergence of our methods. First, we define the level set

$$\Omega = \{x : \|F(x)\| \le \|F(x_0)\|\}. \tag{38}$$

where x_0 is some available point.
To analyze the convergence of Algorithms 1 and 2, we need the following assumptions:

Assumption 4.1.
(1) There exists $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$.
(2) F is continuously differentiable in some neighborhood say N of x^* containing Ω.
(3) The Jacobian of F is bounded and positive definite on N. i.e there exists a positive constants $M > m > 0$ such that

$$\|F'(x)\| \leq M \quad \forall x \in N, \tag{39}$$

and

$$m\|d\|^2 \leq d^T F'(x) d \quad \forall x \in N, d \in \mathbb{R}^n. \tag{40}$$

Remark 4.1. *We give the following remarks.*
Assumption 3.1 implies that there exists a constants $M > m > 0$ such that

$$m\|d\| \leq \|F'(x)d\| \leq M\|d\| \quad \forall x \in N, d \in \mathbb{R}^n. \tag{41}$$

$$m\|x - y\| \leq \|F(x) - F(y)\| \leq M\|x - y\| \quad \forall x, y \in N. \tag{42}$$

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

$$m\|x - x^*\| \leq \|F(x)\| = \|F(x) - F(x^*)\| \leq M\|x - x^*\|, \tag{43}$$

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

Lemma 4.1. Suppose that Assumption 4.1 holds and $\{x_k\}$ is generated by Algorithm 2. Then there exists a constant $m > 0$ such that for all k.

$$s_k^T[F(x_k + \alpha_k d_k^{(2)}) - F(x_k)] \geq m\|s_k\|^2. \tag{44}$$

Proof. By mean-value theorem and (40) we have,

$$s_k^T[F(x_k + \alpha_k d_k^{(2)}) - F(x_k)] = s_k^T F'(\xi) s_k \geq m\|s_k\|^2, \tag{45}$$

where, $\xi_k = x_k + \zeta(x_{k+1} - x_k)$, $\zeta \in (0, 1)$. The proof is complete.

Using $y_k^T s_k \geq m\|s_k\|^2 > 0$, γ_{k+1} is always generated by the update formula (31), and we can deduce that $\gamma_{k+1}I$ inherits the positive definiteness of $\gamma_k I$. By the above lemma and (42), we obtained

$$\frac{y_k^T s_k}{\|s_k\|^2} \geq m, \qquad \frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{M^2}{m}. \tag{46}$$

Lemma 4.2. Suppose that Assumption 4.1 holds and $\{x_k\}$ is generated by algorithm 2. Then we have

$$\lim_{k\to\infty} \|\alpha_k d_k^{(2)}\| = 0, \tag{47}$$

and

$$\lim_{k\to\infty} \|\alpha_k F(x_k)\| = 0. \tag{48}$$

Proof. By (33) we have for all $k > 0$

$$\begin{aligned} \omega_2\|\alpha_k d_k^{(2)}\|^2 &\leq \omega_1\|\alpha_k F(x_k)\|^2 + \omega_2\|\alpha_k d_k^{(2)}\|^2 \\ &\leq \|F(x_k)\|^2 - \|F(x_{k+1})\|^2 + \eta_k\|F(x_k)\|^2. \end{aligned} \tag{49}$$

By summing the above inequality, we have

$$\begin{aligned}\omega_2\sum_{i=0}^{k}\|\alpha_i d_i^{(2)}\|^2 &\le \sum_{i=0}^{k}\left(\|F(x_i)\|^2-\|F(x_{i+1})\|^2\right)+\sum_{i=0}^{k}\eta_i\|F(x_i)\|^2\\ &= \|F(x_0)\|^2-\|F(x_{k+1})\|^2+\sum_{i=0}^{k}\eta_i\|F(x_i)\|^2\\ &\le \|F(x_0)\|^2+\|F(x_0)\|^2\sum_{i=0}^{k}\eta_i\\ &\le \|F(x_0)\|^2+\|F(x_0)\|^2\sum_{i=0}^{\infty}\eta_i.\end{aligned} \tag{50}$$

So, from the level set and the fact that $\{\eta_k\}$ satisfies (32) then the series $\sum_{i=0}^{\infty}\|\alpha_i d_i^{(2)}\|^2$ is convergent. This implies (47). By similar argument we can prove that (48) holds.

Lemma 4.3. Suppose that Assumption 4.1 holds and $\{x_k\}$ is generated by Algorithm 2. Then there exists some positive constants m_2 such that for all $k > 0$,

$$\|d_k^{(2)}\| \le m_2. \tag{51}$$

Proof. From (37) and(42), we have

$$\begin{aligned}\|d_k^{(2)}\| &= \left\|-\frac{(\beta_k+1)F(x_k)y_k^T s_k}{\|y_k\|^2}\right\|\\ &\le \frac{|\beta_k+1|\|F(x_k)\|\|s_k\|\|y_k\|}{m^2\|s_k\|^2}\end{aligned} \tag{52}$$

But, from (34) we have

$$|\beta_k| = \left|\frac{s_k^T y_k}{s_k^T s_k}\right| \le \frac{\|s_k\|\|y_k\|}{\|s_k\|\|s_k\|} \le \frac{M\|s_k\|}{\|s_k\|} = M, \tag{53}$$

which leads to

$$|\beta_k+1| \le (M+1). \tag{54}$$

Therefore,

$$\begin{aligned}\|d_k^{(2)}\| &\le \frac{(M+1)\|F(x_k)\|M\|s_k\|}{m^2\|s_k\|}\\ &= \frac{(M+1)\|F(x_k)\|M}{m^2}\\ &\le \frac{(M+1)\|F(x_0)\|M}{m^2}.\end{aligned} \tag{55}$$

Setting $m_2 = \frac{(M+1)\|F(x_0)\|M}{m^2}$, we have (51), which completes the proof.

Since $\gamma_k I$ approximates $F'(x_k)$ along direction s_k, we can give the following assumption.

Assumption 4.2.
$\gamma_k I$ is a good approximation to $F'(x_k)$, i.e

$$\|(F'(x_k) - \gamma_k I)d_k^{(2)}\| \leq \epsilon \|F(x_k)\|, \tag{56}$$

where $\epsilon \in (0, 1)$ is a small quantity [15].

Lemma 4.4. Let Assumption 4.2 hold and $\{x_k\}$ be generated by Algorithm 2. Then d_k is a descent direction for $f(x_k)$ at x_k i.e

$$\nabla f(x_k)^T d_k^{(2)} < 0. \tag{57}$$

Proof. From (37) and (46) we have

$$\begin{aligned} \nabla f(x_k)^T d_k^{(2)} &= F(x_k)^T F'(x_k) d_k^{(2)} \\ &= F(x_k)^T [(F'(x_k) - \gamma_k I)d_k^{(2)} - (\beta_k + 1)F(x_k)] \\ &= F(x_k)^T (F'(x_k) - \gamma_k I)d_k^{(2)} - (\beta_k + 1)\|F(x_k)\|^2, \end{aligned} \tag{58}$$

Using Cauchy-Schwartz inequality, we have,

$$\begin{aligned} \nabla f(x_k)^T d_k^{(2)} &\leq \|F(x_k)\|\|(F'(x_k) - \gamma_k I)d_k^{(2)}\| - (m + 1)\|F(x_k)\|^2 \\ &\leq \|F(x_k)\|\epsilon\|F(x_k)\| - (m + 1)\|F(x_k)\|^2 \\ &\leq -((m + 1) - \epsilon)\|F(x_k)\|^2. \end{aligned} \tag{59}$$

Hence for $\epsilon \in (0, 1)$ this Lemma is true.
Lemma 4.5. Let Assumption 4.2 hold and $\{x_k\}$ be generated by Algorithm 2. Then $\{x_k\} \subset \Omega$.
Proof. By Lemma 4.4 we have $\|F(x_{k+1})\| \leq \|F(x_k)\|$. Moreover, we have for all k.

$$\|F(x_{k+1})\| \leq \|F(x_k)\| \leq \|F(x_{k-1})\| \leq \ldots \leq \|F(x_0)\|.$$

This implies that $\{x_k\} \subset \Omega$.

Now we are going to establish the following global convergence theorem to show that under some suitable conditions,there exist an accumulation point of $\{x_k\}$ which is a solution of problem (1).

Theorem 4.1.
Suppose that Assumption 4.1 holds,$\{x_k\}$ is generated by Algorithm 2. Assume further for all $k > 0$,

$$\alpha_k \geq c \frac{|F(x_k)^T d_k^{(2)}|}{\|d_k^{(2)}\|^2}, \tag{60}$$

where c is some positive constant. Then

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{61}$$

Proof. From lemma 4.2 we have (51). Therefore by (47) and the boundedness of $\{\|d_k\|^{(2)}\}$, we have

$$\lim_{k\to\infty} \alpha_k \|d_k^{(2)}\|^2 = 0. \tag{62}$$

From (60) and (62) we have

$$\lim_{k\to\infty} |F(x_k)^T d_k^{(2)}| = 0. \tag{63}$$

On the other hand from (37) we have,

$$F(x_k)^T d_k^{(2)} = -\lambda_k \gamma_k^{-1} \|F(x_k)\|^2 \tag{64}$$

$$\begin{aligned} \|F(x_k)\|^2 &= \| - F(x_k)^T d_k^{(2)} \gamma_k \lambda_k^{-1} \| \\ &\leq |F(x_k)^T d_k^{(2)}| |\gamma_k| |\lambda_k^{-1}|. \end{aligned} \tag{65}$$

But from (46) we have,

$$\gamma_k \leq \frac{M^2}{m}.$$

Also, from (46) we have,

$$\lambda_k = (\beta_k + 1) \geq (m + 1),$$

$$\lambda_k^{-1} < \frac{1}{m+1}.$$

So from (65) we have,

$$\|F(x_k)\|^2 \leq |F(x_k)^T d_k^{(2)}| \left(\frac{M^2}{m}\right)\left(\frac{1}{m+1}\right). \tag{66}$$

Thus,

$$0 \leq \|F(x_k)\|^2 \leq |F(x_k)^T d_k^{(2)}| \left(\frac{M^2}{m}\right)\left(\frac{1}{m+1}\right) \longrightarrow 0. \tag{67}$$

Therefore,

$$\lim_{k \to \infty} \|F(x_k)\| = 0. \tag{68}$$

This completes the proof.

Remark 4.2. *If the correction parameter* $(\beta_k + 1) = \beta \in (1, 2)$, $\forall k$, *then the convergence result of Algorithm 1 (HDAP1) follows.*

5. Numerical Results

In this section, we carry out some numerical experiments to highlight the effectiveness of the (HDAP1) and (HDAP2) methods by comparing them with an improved derivative-free method via double direction approach for solving systems of nonlinear equations (IDFDD) [13] and a transformed double step-length method for solving large-scale systems of nonlinear equations [12]. For all the algorithms, the following parameters are set $\omega_1 = \omega_2 = 10^{-4}$, $r = 0.2$ and $\eta_k = \dfrac{1}{(k+1)^2}$. We however set $\beta = 1.9$, in Algorithm 1 (HDAP1). The computational codes were written in Matlab (8.3.0 532) R2014a and run on a personal computer 1.60 GHz CPU processor and 4 GB RAM memory. The iteration is set to terminate if the total number of iterations exceed 1000 or when $\|F(x_k)\| \leq 10^{-4}$. We claim that the method fails, and use the symbol "-" to indicate failure due to; (1) Memory requirement (2) Number of iterations exceed 1000. (3) If $\|F(x_k)\|$ is not a number.

Problem 1 [28] **(The discretized Chandrasekhar's H-equation)**

$F_i(x) = x_i - (1 - \frac{c}{2n} \sum_{j=1}^{n} \frac{\mu_i x_j}{\mu_i + \mu_j})^{-1}, \quad i = 1, 2, \ldots, n,$
with $c \in [0, 1)$ and $\mu_i = \frac{i - 0.5}{n}$, for $1 \leq i \leq n$. (In our experiment we take $c = 0.1$).

$x_0 = (0.3, 0.3, , ..., 0.3)^T$

Problem 2 [32]

$$F_i(x) = x_i^2 - 4, \quad i = 1, 2, ..., n,$$

$$x_0 = (0.2, 0.2, , ..., 0.2)^T.$$

Problem 3 [31]

$$F_i(x) = x_i^2 + x_i - 2, \quad i = 1, 2, 3, ..., n,$$

$$x_0 = (0.2, 0.2, , ..., 0.2)^T.$$

Problem 4[32]

$$F_i(x) = x_i^2 - cos(x_i - 1), \quad i = 1, 2, ..., n,$$

$$x_0 = (0.5, 0.5, ..., 0.5)^T.$$

Problem 5 [13]

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

$$x_0 = (0.1, 0.1, ..., 0.1)^T.$$

Problem 6 [31]

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

$$x_0 = (0.4, 0.4, ..., 0.4)^T.$$

Problem 7 [13]

$$F_1(x) = x_1(x_1^2 + x_2^2) - 1,$$

$$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1,$$

$$F_n(x) = x_n(x_{n-1}^2 + x_n^2), \quad i = 2, 3, ...n - 1.$$

$$x_0 = (0.8, 0.8, ..., 0.8)^T.$$

Problem 8 [13]

$$F_{3i-2}(x) = x_{3i-2} - x_{3i-1} - x_{3i}^2 - 1,$$

$$F_{3i-1}(x) = x_{3i-2} x_{3i-1} x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,$$

$$F_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}, i = 1, ..., \frac{n}{3}.$$

$$x_0 = (1.5, 1.5, ..., 1.5)^T$$

Problem 9 [12]

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (e^{x_1} - 1, ..., e^{x_n} - 1)^T.$$

$x_0 = (0.01, 0.01, ..., 0.01)^T$.

Problem 10 [12]

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (\sin x_1 - 1, ..., \sin x_n - 1)^T.$$

$x_0 = (0.9, 0.9, ..., 0.9)^T$.

The numerical results of the two methods are reported in the table below, where 'NI' and 'Time' stand for the total number of all iterations and the CPU time in seconds respectively, while $\|F(x_k)\|$ is the norm of the residual at the stopping point.

Table 1: The numerical results of HDAP1, HDPA2, IDFDD and TDS for problems 1 to 10

		HDAP1			HDAP2			IDFDD			TDS		
Problems	Dim	NI	CPU	$\|\|F(x_k)\|\|$	NI	CPU	$\|\|F(x_k)\|\|$	NI	CPU	$\|\|F(x_k)\|\|$	NI	CPU	$\|\|F(x_k)\|\|$
1	100	7	0.00593	3.67E-05	5	0.009751	6.47E-05	_	_	_	16	0.026014	7.90E-05
	1000	9	0.01448	5.36E-05	8	0.015088	3.38E-05	_	_	_	13	0.047095	5.11E-05
	10000	11	0.120445	3.43E-05	9	0.152764	8.79E-05	_	_	_	17	0.621718	1.81E-07
2	100	13	0.017995	6.25E-05	7	0.004315	4.62E-05	29	0.03988	7.69E-05	15	0.03735	4.32E-05
	1000	15	0.011562	3.90E-05	8	0.008266	2.89E-05	31	0.033866	9.96E-05	16	0.043527	5.46E-05
	10000	16	0.081123	9.76E-05	9	0.052476	1.81E-05	34	0.195773	8.26E-05	17	0.342734	6.91E-05
3	100	13	0.006577	2.59E-05	9	0.005491	6.91E-05	32	0.02234	7.89E-05	19	0.027275	5.05E-05
	1000	14	0.013497	6.49E-05	10	0.010782	7.69E-05	35	0.037172	7.85E-05	20	0.042639	7.98E-05
	10000	16	0.088598	4.05E-05	11	0.070012	8.54E-05	38	0.253204	7.81E-05	22	0.331333	6.31E-05
4	100	11	0.005111	6.78E-05	8	0.005696	9.16E-05	19	0.013456	9.52E-05	14	0.013993	9.60E-05
	1000	13	0.013599	4.25E-05	10	0.012767	5.73E-05	23	0.033748	8.09E-05	17	0.041219	6.56E-05
	10000	15	0.095024	2.65E-05	12	0.080011	3.59E-05	26	0.208599	9.55E-05	19	0.561042	7.46E-05
5	100	13	0.007775	7.00E-05	5	0.004141	2.16E-05	16	0.013096	7.78E-05	12	0.024931	8.78E-05
	1000	15	0.017883	4.37E-05	6	0.008767	1.34E-05	19	0.026529	6.45E-05	14	0.04753	4.44E-05
	10000	17	0.111422	2.73E-05	7	0.062109	8.37E-06	21	0.184433	8.35E-05	15	0.318188	5.62E-05
6	100	12	0.006542	8.98E-05	8	0.006284	1.17E-05	28	0.022538	9.84E-05	15	0.035392	4.65E-05
	1000	14	0.017078	5.61E-05	9	0.014304	7.32E-06	31	0.040581	8.16E-05	16	0.038844	5.88E-05
	10000	16	0.12225	3.51E-05	9	0.072386	7.32E-05	34	0.320728	6.77E-05	17	0.390777	7.45E-05
7	100	18	0.010192	7.81E-05	13	0.010736	5.59E-05	35	0.021717	8.69E-05	25	0.031415	9.82E-05
	1000	18	0.023018	9.50E-05	14	0.03174	4.12E-05	38	0.054704	9.46E-05	26	0.045267	6.35E-05
	10000	19	0.132834	6.90E-05	14	0.128105	7.90E-05	38	0.349107	7.72E-05	25	0.40698	6.32E-05
8	100	14	0.013666	8.02E-05	9	0.009669	2.33E-05	32	0.02833	9.75E-05	19	0.026912	7.15E-05
	1000	16	0.027242	2.24E-05	10	0.019057	8.02E-06	35	0.06221	9.67E-05	21	0.063626	5.59E-05
	10000	18	0.139498	2.54E-05	10	0.091141	8.02E-05	38	0.33641	9.54E-05	22	0.393686	8.77E-05
9	100	4	0.064565	6.38E-05	3	0.056732	3.28E-05	24	0.259925	8.99E-05	12	0.812017	6.69E-05
	1000	6	0.255465	6.10E-05	4	0.198266	2.45E-05	27	1.112368	7.63E-05	12	0.506232	7.35E-05
	10000	6	20.9957	5.09E-05	5	21.00411	5.58E-05	32	132.6868	8.92E-05	15	60.66498	7.83E-05
10	100	14	0.132477	3.07E-05	10	0.109129	1.85E-05	34	0.370885	9.51E-05	24	0.272104	6.32E-05
	1000	14	0.533679	5.06E-05	11	0.41393	3.32E-05	37	1.699962	9.11E-05	27	1.15811	9.31E-05
	10000	16	56.86448	8.47E-05	14	47.5293	1.62E-05	41	171.0149	7.69E-05	28	119.5646	9.24E-05

Table 2: Summary of results from Table 1 for HDAP1, HDAP2, IDFDD and TDS methods

	Method	NI	Percentage	CPU time	Percentage
Number of Problems and	HDAP1	0	0%	6	20%
percentage for each method with	HDAP2	30	100%	24	80%
respect to iterations and CPU time.	IDFDD	0	0%	0	0%
	TDS	0	0%	0	0%

In addition, a summary of the test results reported in Table 1 are presented in Table 2. Also, using the performance profile of Dolan and Moré as an evaluation tool, we present two figures to approximately assess the performance and efficiency of each of the methods.

It can be observed from Table 1 that, for the exception of the IDFDD method, which fails to solve problem 1, all the methods attempted to solve all the problems. In Table 2, the summarized results exhibits the performance of each of the four methods with respect to number of iterations and CPU time respectively. It can be observed from the summary table that the HDAP2 method represents the most efficient scheme among the four methods as it solves 100% of the problems with the least number of iterations compared to the remaining three methods. The summary table also shows that the HDAP2 scheme outperforms the other three methods with respect to CPU time as it solves 80% of the problems with least CPU time as against the HDAP1 method, which solves 20% and the IDFDD and TDS methods, which both record 0% respectively. It is worth nothing at this juncture that of the two proposed methods, the HDAP2 method exhibits better performance against the HDAP1 method as a result of updating the correction parameter β_k

in each iteration which leads to faster convergence when compared with HDAP1 method that has a fixed value for the correction parameter through out the work.

Furthermore, we present the performance profiles of all the four methods in Figures $1-2$ with respect to number of iterations and CPU time by using results reported in Tables 1 and the idea introduced by Dolan and Moré [30]. We achieved this by plotting the fraction $\mathrm{p}(\mu)$ of the problems for which each method is within μ of the smallest number of iterations and CPU time respectively. We observed from Figure 1 that the HDAP2 method exhibits the best performance and has an edge over the other methods with least number of iteration. This can be seen from the curve representing the HDAP2 scheme, which stays above the other curves representing other methods. Fig 1 also shows that the HDAP1 method is more efficient than the IDFDD and TDS methods as indicated by the curve representing the HDAP1 scheme, which stays top of the curve representing the IDFDD and TDS methods. From Figure 2, it is observed that HDAP2 method performs better than HDAP1 method in terms of least CPU time, and in turn, the HDAP1 method exhibits better performance compared to the IDFDD and TDS methods. All these can be seen from the curves representing the four methods. Hence, the HDAP1 and HDAP2 methods are more efficient as they all outperforms the IDFDD and TDS methods with respect to least number of iterations and CPU time.

In addition, to explain the accuracy of our results, we take the average of the norm of the residuals recorded at the stopping point for each of the four methods, and the HDAP2 scheme tops the list with 4.33×10^{-5}, HDAP1 5.42×10^{-5}, TSD 6.67×10^{-5}, and IDFDD 8.54×10^{-5}. This clearly shows that our proposed methods converge faster to the solution than the other methods. As a further insight into the importance of this research, it can be seen that our proposed methods can be applied to solve the decritized form of the popular Chandrasekhar Integral equation presented in problem 1. This is important because of the role played by the Chandrasekhar Integral equation in radiactive transfer and transport theory [67]. The Chandrasekhar Integral equation is given by

$$H(\mu) = 1 + H(\mu) \int_0^1 \frac{\mu}{\mu + t} \psi(t) H(t) dt. \tag{69}$$

The most common approach of finding approximate solution of (69) is discretizing it by a vector $\bar{x} \in R^n$, and then replacing the integrals by quadrature sums and the derivatives by difference quotients involving only the component of $\bar{x} \in R^n$ (see[66]). And so, (69) becomes a problem of finding the solution of system of n nonlinear equations with n unknowns as presented in problem 1.

Therefore, considering results reported in Tables 1, its summary in Table 2 and Figures $1-2$, we conclude that the proposed HDAP1 and HDAP2 methods are more effective for solving large-scale nonlinear equations than the IDFDD and TDS methods.

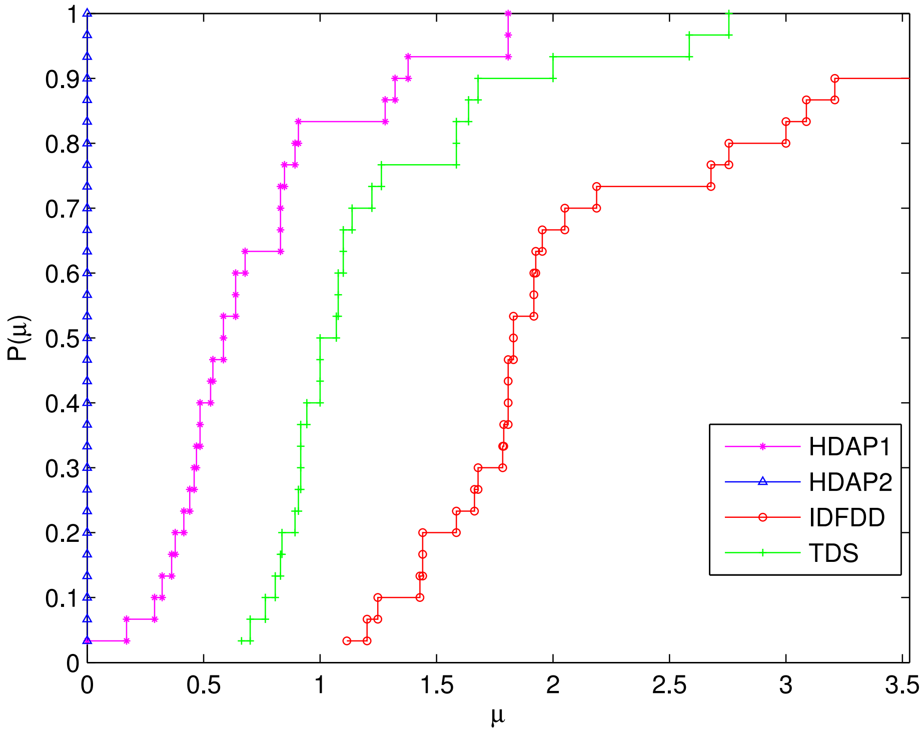

Figure 1: Performance profile of HDAP1, HDAP2, TDS and IDFDD methods with respect to the number of iteration for the problems 1-10.

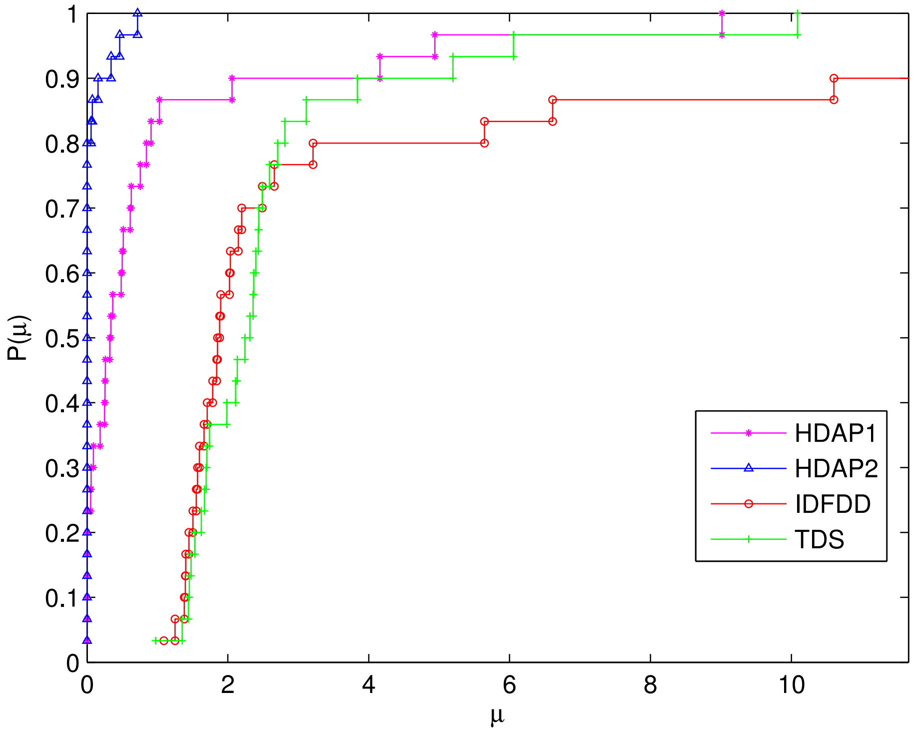

Figure 2: Performance profile of HDAP1, HDAP2, TDS and IDFDD methods with respect to the CPU time for the problems 1-10.

6. Conclusion

In this paper, two derivative-free decent methods via acceleration parameter for solving systems of nonlinear equations are presented. The methods are obtained by approximating the Jacobian matrix via acceleration and correction parameters. Attractive features of the methods includes derivative-free, generating descent search direction and easy implementation. By using basic assumptions, we prove global convergence of the schemes proposed. Numerical comparisons using a set of large-scale test problems show that the proposed methods are promising. Future research include, modification of the proposed method to solve convex constrained monotone nonlinear equation with applications ℓ_1 norm problems arising in signal and image processing.

References

[1] M.J. Petrović, *Hybridization Rule Applied on Accelerated Double Step Size Optimization Scheme*, Filomat, 33(3) (2019) 655–665. doi.org/10.2298/FIL1903655P

[2] M.J. Petrović, V. Rakocević, D. Valjarević et al.,*A note on hybridization process applied on transformed double step size model*, Numer Algor. 85 (2020) 449–465. doi.org/10.1007/s11075-019-00821-8

[3] D. Li, M. Fukushima, *A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equations*, SIAM J. Num. Anal. 37 (1999) 152–172.

[4] J.E. Dennis, and J.J. More, *A characterization of superlinear convergence and its application to quasi-newton methods*, Math Comput. 1974; 28: 549–560.

[5] P.S. Stanimirović, G.V. Milovanović, M.J. Petrović, N. Kontrec, *A transformation of accelerated double step size method for unconstrained optimization*, Math. Probl. Eng. Article ID 283679, (2015) 8 pages. doi.org/10.1155/2015/283679.

[6] M.Y. Waziri, K. Ahmad, A.S. Halilu, *Enhanced Dai–Liao conjugate gradient methods for systems of monotone nonlinear equations*, SeMA Journal, 2020; https://doi.org/10.1007/s40324-020-00228-9.

[7] S.H. Khan, *A Picard-Mann hybrid iterative process*, Fixed Point Theory and Applications, 69 (2013).

[8] M.Y. Waziri, W.J. Leong, M.A. Hassan, M. Monsi, *Jacobian computation-free Newton method for systems of Non-Linear equations*, J. Numer. Math. Stoch. 2 (2010) 54–63.

[9] M. Raydan, *On Barzilai and Borwein choice of step length for the gradient method*, IMA J. Numer. Anal. 13 (1993) 321-326.

[10] S. Ishikawa, *Fixed points by a new iteration method*, Proc. Am. Math. Soc. 44 (1974) 147-150.

[11] W.R. Mann, *Mean value methods in iterations*, Proc. Am. Math. Soc. 4 (1953) 506-510.

[12] A.S. Halilu, M.Y. Waziri, *A transformed double steplength method for for solving large-scale systems of nonlinear equations*, J. Numer. Math. Stoch. 9(1) (2017) 20-23.

[13] A.S. Halilu, M.Y. Waziri, *An improved derivative-free method via double direction approach for solving systems of nonlinear equations*, J. Ramanujan Math. Soc. 33(1) (2018), 75-89.

[14] A.S. Halilu, M.Y. Waziri, I. Yusuf, *Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations*, Yugos. J. Oper. Res. 30(4) (2020) 399–412. DOI: https://doi.org/10.2298/YJOR160515005H

[15] G. Yuan, X. Lu, *A new backtracking inexact BFGS method for symmetric nonlinear equations*, J. Comp. Math. Appl. 55 (2008) 116–129.

[16] A.A. Goldstein, *On steepest descent*, SIAM J. Control 3, (1965) 147–151.

[17] F.A. Potra, Y. Shi, *Efficient line search algorithm for unconstrained optimization*, J. Optim. Theory Appl. 85 (1995) 677–704.

[18] M.J. Petrović, P.S. Stanimirović, Accelerated gradient descent methods with line search. Numer. Algor. 51(2010) 503–520.

[19] M. Stanimirović, V. Rakocević, N. Kontrec, S. Panic, and D. Ilic, *Hybridization of accelerated gradient descent method*, Numer Algor. 79 (2018) 769–786.

[20] M.J. Petrović, P.S. Stanimirović, *Accelerated double direction method for solving unconstrained optimization problems*, Mathl. Probl. Eng. (2014).

[21] M.J.Petrović, *An accelerated double step size method in unconstrained optimization*, Applied Math. Comput. 250, 309–319 (2015) https://doi.org/10.1016/j.amc.2014.10.104.

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

[23] H. Abdullahi, A.S. Halilu, M.Y. Waziri, *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.

[24] Y.H. Dai, L.Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, IMA J. Numer. Anal. 22 (2002)1–10.

[25] M.J.D. Powell, *Some Global Convergence Properties of a Variable-Metric Algorithm For Minimization Without Exact Line Search*, AIAM–AMS Proc. Philadelphia, 9 (1976) 53–72.

[26] E. Picard, *Memoire sur la theorie des equations aux derivees partielles et la meth-ode des approximations successives*, J. Math. Pures Appl. 6 (1890) 145–210.

[27] J. Barzilai, J.M. Borwein, *Two point step size gradient method*, IMA J. Numer. Anal. 8 (1988) 141–148.

[28] M.Y. Waziri, K. Ahmed, J. Sabi'u, *A family of Hager–Zhang conjugate gradient methods for system of monotone nonlinear equations*, Appl. Math. Comp. 361 (2019) 645-660.

[29] A.S. Halilu, M.Y. Waziri, *Solving systems of nonlinear equations using improved double direction method*, J. Nig. Mathl. Soc. 32(2) (2020) 287–301.

[30] E. Dolan, J. Moré, *Benchmarking optimization software with performance profiles*, J. Mathl. Progr. 91(2)(2002) 201-2013.

[31] W. La Cruz, J.M. Martinez, M. Raydan, *Spetral residual method without gradient information for solving large-scale nonlinear systems of equations*, Theory and experiments, P. optimization (2004) 76-79.

[32] A.B. Abubakar, *On Improved Broyden-Type Method For Systems of Nonlinear Equations*, M.Sc. Thesis Bayero University, Kano. (2014) 51-54.

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

[34] M.Y. Waziri, H.U. Muhammad, A.S. Halilu, K. Ahmed, *A Modified matrix-free methods for solving system of nonlinear equations*, Opt. (2020) 1–20. DOI: 10.1080/02331934.2020.1778689

[35] T. Hayat, M. Ijaz Khan, M. Farooq, T. Yasmeen, A. Alsaedi, *Stagnation point flow with Cattaneo-Christov heat flux and homogeneous-heterogeneous reactions*, J. of Mol. Liq. 220 (2016) 49–55.

[36] M.I. Khan, M. Waqas, T. Hayat, A. Alsaedi, *Colloidal study of Casson fluid with homogeneous-heterogeneous reactions*, J. Colloid and Interface Science, (2017) doi: http://dx.doi.org/10.1016/j.jcis.2017.03.024.

[37] T. Hayat, M. Ijaz Khan, M. Farooq, A. Alsaedi, M. Waqas, T. Yasmeen, *Impact of Cattaneo-Christov heat flux model in flow of variable thermal conductivity fluid over a variable thicked surface*, Int. J. Heat and Mass Transfer, 99(2016) 702–710.

[38] W. La Cruz, M. Raydan, *Nonmonotone spectral methods for large-scale nonlinear systems, Optim. Methods Softw*, 18 (2003) 583–599.

[39] L. Grippo, F. Lampariello, S. Lucidi, *A nonmonotone linesearch technique for Newton's method*, SIAM J. Numer. Anal. 23 (1986) 707–716.

[40] M.V. Solodov, B.F. Svaiter, *A globally convergent inexact Newton method for systems of monotone equations*, In: Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Springer, (1999) 355–369.

[41] L. Zhang, W. Zhou, *Spectral gradient projection method for solving nonlinear monotone equations*, J. Comput Appl. Math. 196 (2006)478–484.

[42] M.Y. Waziri, K. Ahmed, J. Sabi'u, *A Dai-Liao conjugate gradient method via modified secant equation for system of nonlinear equations*, Arabian Journal of Mathematics. http://doi.org/10.1007/s40065-019-0264-6, (2019) 1–15.

[43] A.S. Halilu, A. Majumder, M.Y. Waziri, H. Abdullahi, *Double direction and step length method for solving system of nonlinear equations*, Euro. J. Mol. Clinic. Med. 7(7) (2020) 3899–3913.

[44] D. Da-Jiang, J. Di-Qing, D. Chao-Qing, *Analytical Solutions of Differential-Difference Sine-Gordon Equation*, Thermal Science, 21, No. 4 (2017) 1701–1705.

[45] B. Zhang, X. Zhang, C. Dai, *Discussions on localized structures based on equivalent solution with different forms of breaking soliton model*, Nonlinear Dyn. DOI 10.1007/s11071-016-3197-z (2016).

[46] C. Dai, Y. Wang, Y. Fan, J. Zhang, *Interactions between exotic multi-valued solitons of the (2+1)-dimensional Korteweg-de Vries equation describing shallow water wave*, Appl.Mathel. Model. https://doi.org/10.1016/j.apm.2019.11.056 (2019).

[47] C. Dai, Y. Fan, Y. Wang, *Three-dimensional optical solitons formed by the balance between different-order nonlinearities and high-order dispersion/diffraction in parity-time symmetric potentials*, Nonlinear Dyn. https://doi.org/10.1007/s11071-019-05206-z (2019).

[48] L. Kong, J. Liu, D. Jin, D. Ding, C. Dai, *Soliton dynamics in the three-spine á-helical protein with inhomogeneous effect*, Nonlinear Dyn. DOI 10.1007/s11071-016-3027-3 (2016).

[49] S.P. Dirkse, M.C. Ferris, MCPLIB, *A collection of nonlinear mixed complementarity problems*, Optim. Methods Softw. 5 (1995) 319–345.

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

[51] A.S.Halilu, M.K. Dauda, M.Y. Waziri, M. Mamat, *A derivative-free decent method via acceleration parameter for solving systems of nonlinear equations*, Open J. Sci. Tech. 2(3) (2019) 1–4.

[52] N. Andrei, *Another hybrid conjugate gradient algorithm for unconstrained optimization*, Numer Algor. 47 (2008) 143–156.

[53] N. Andrei, *Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization*, J. Optim. Theory Appl. 141(2009) 249–264.

[54] J. Jian, L. Han, X. Jiang, A hybrid conjugate gradient method with descent property for unconstrained optimization, Appl. Mathl. Model. 39 (2015) 1281–1290.

[55] R. Fletcher, C. Reeves, *Function minimization by conjugate gradients*, The Computer Journal, 7 (1964) 149–154.

[56] J.K. Li, S.J. Liu, *New hybrid conjugate gradient method for unconstrained optimization*, Appl. Math. Comp. 245(2014) 36–43

[57] Z. Wei, S. Yao, L. Liu, *The convergence properties of some new conjugate gradient methods*, Appl. Math. Comp. 183 (2006) 1341–1350.

[58] Y. Gonglin, *A conjugate gradient method for unconstrained optimization problems*, Int. J. Math. Mathl. Sci. Article ID 329623, 14 pages (2009).

[59] X. Xu, F. Kong, *New hybrid conjugate gradient methods with the generalized Wolfe line seac*,. Springer plus, 5 10.1186/s40064-016-2522-9, (2016).

[60] G. Wu, Y. Li, G. Yuan, *A Three-Term Conjugate Gradient Algorithm with Quadratic Convergence for Unconstrained Optimization Problems*, Mathl. Prob. Eng. (2018) Article ID 4813030, 15 pages. https://doi.org/10.1155/2018/4813030.

[61] Y. Liu, C. Storey, *Efficient Generalized Conjugate Gradient Algorithms*, Part 1: Theory, J. Optim. Theory Appl. 69(1991) 129–137.

[62] J. Liu, X. Wu, *New three-term conjugate gradient method for solving unconstrained optimization problems*, ScienceAsia 40(2014) 295-300.

[63] M.R. Hestenes, E.L. Stiefel, *Methods of conjugate gradients for solving linear systems* J. Research Nat. Bur. Standards 49, (1952) 409–436.

[64] Y.H. Dai, L.Z. Liao, *New conjugacy conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim. 43 (2001)87–101.

[65] Y.H. Dai, Y. Yuan, *A Nonlinear Conjugate Gradient Method with a strong Global Convergence Property*, SIAM J. Optim., 10 (1999) 177–182.

[66] M.O. James, C.R. Werner, *On Discretization and Differentiation of Operators with Application to Newton's Method* SIAM J. Numer. Anal. 3(1)(1966) 143–156.

[67] G.A. Hively, *On a Class of Nonlinear Integral Equations arising in Transport Theory*, SIAM J. on Mathl. Anal. 9(5) (1978) 787–792.