Results in Control and Optimization 6 (2022) 100072

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Double direction three-term spectral conjugate gradient method for solving symmetric nonlinear equations

Mohammed Yusuf Waziri [a,d], Yusuf Muhammad Kufena [b,d], Abubakar Sani Halilu [c,d,*]

[a] *Department of Mathematical Sciences, Bayero University, Kano, Nigeria*
[b] *Department of Mathematics and Statistics, School of Applied Sciences, Nuhu Bamalli Polytechnic, Zaria, Nigeria*
[c] *Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria*
[d] *Numerical Optimization Research Group, Bayero University, Kano, Nigeria*

ARTICLE INFO

MSC:
65K05
90C30
90C53

Keywords:
Double direction
Spectral conjugate gradient method
Secant condition
Descent condition
Global convergence

ABSTRACT

We propose a new three-term spectral conjugate gradient method via double direction approach by considering the first direction to be the one proposed by Halilu and Waziri (2018) and the other obtained by extending the direction proposed by Birgin and Mertinez (2001) to three-term spectral conjugate gradient (CG) direction. The propose method generates a descent direction via an inexact line search. The global convergence of the propose algorithm was established under appropriate conditions and numerical experiments on some benchmark test problems demonstrate its efficiency over some existing ones.

1. Introduction

We consider the following system of nonlinear equations:

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

where $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, is a nonlinear map. Problem (1) can be obtained from an unconstrained optimization problem, a saddle point and equality constrained problem [1]. Let f be a function defined by:

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

Problem (1) is equivalent to the following unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$. Several methods have been developed for solving nonlinear system of equations. Most of these methods fall under the Newton and quasi-Newton approaches and are particularly welcomed because of their rapid convergence properties from a sufficiently good initial guess. Nevertheless, these methods are unattractive for large-scale nonlinear system of equations because; they require computation and storage of Jacobian matrix and its inverse or its approximation, solving n linear system of equations at each iteration, or convergence may even be lost when the Jacobian is singular. Hence, various methods have been developed to handle such problems [1–10]. Spectral gradient methods were introduced so as to solve potentially large-scale unconstrained

* Corresponding author at: Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria.
E-mail addresses: mywaziri.mth@buk.edu.ng (M.Y. Waziri), mathstardon@gmail.com (Y.M. Kufena), abubakars.halilu@slu.edu.ng (A.S. Halilu).

https://doi.org/10.1016/j.rico.2021.100072
Received 30 September 2020; Received in revised form 24 July 2021; Accepted 5 November 2021
Available online 18 November 2021

optimization problems whereby only gradient directions are used at each line search which makes the methods to outperforms conjugate gradient algorithms in many problems [11]. They generate an iterative sequence of points $\{x_k\}$ via:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{4}$$

where k = 0,1,2, …and $\alpha_k > 0$ is the step length which is obtained using line search. Most of the frequently used line search in practice is the inexact type [1,2,12], which is chosen such that the function values along $x_k + \alpha_k d_k$ decrease, that is:

$$f(x_k + \alpha_k d_k) \leq f(x_k), \tag{5}$$

and the direction d_k is obtained using

$$d_k = \begin{cases} -g(x_k), & \text{if} \quad k = 0, \\ -\theta_k g(x_k), & \text{if} \quad k \geq 1, \end{cases} \tag{6}$$

where θ_k is the spectral parameter and g_k is the gradient of $f(x_k)$ at x_k. Nonlinear conjugate gradient method was first introduced in 1964 by Fletcher and Reeves [13] and since then, research on CG became active and quite noticed in the literature. Now CG-method has been assimilated to solving large scale nonlinear system of equations [14]. Also, in recent years, much effort are made in proposing new and effective formula for the conjugate parameter β_k in order to make the method more easier in other various fields with good numerical performance and convergence properties. Such type of considerations lead to Hybrid methods, which are based on two or more different methods whereby one has a good performance at the initial stage, for example the gradient method, and the other one has good performance at the final stage, for example Newton's methods [15].

In order to improve the efficiency of the classical conjugate gradient method, some three-term conjugate gradient methods were developed [16–18]. Yuan and Zhang [16] proposed a three-term CG direction which possessed both sufficient descent and trust region property. Waziri et al. [19] also developed an efficient method with low storage requirements that does not require computing the Jacobian matrix. Furthermore, Waziri and Muhammad [17], proposed a decent three-term CG method with the direction proposed as:

$$d_{k+1} = -g_{k+1} - \delta_k s_k - \eta_k y_k,$$

where $\delta_k = \left(1 - \min\left\{1, \frac{\|y_k\|^2}{y_k^T s_k}\right\}\right) \frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k}$ and $\eta_k = \frac{s_k^T g_{k+1}}{y_k^T s_k}$.

The acceleration and restart strategies were incorporated into the algorithm's design to improve its numerical performance, which demonstrated the method's efficiency over the compared one. Studies on CG methods were inspired for solving large-scale nonlinear symmetric equations. Zhang et al. [20], proposed a descent PRP method which was further extended by Zhou and Shen [21] by combining it with the work of Li and Fukushima [1] and successfully used for solving symmetric system of Eqs. (1). Waziri, et al. [22] proposed a derivative-free three-term spectral CG method for symmetric nonlinear equations which is an improvement to the work of Waziri and Sabi'u [14] and was found to be more efficient and effective than some existing methods in the literature. Halilu and Waziri [23] presented an improved double direction iterative scheme as:

$$x_{k+1} = x_k + \alpha_k c_k + \alpha_k^2 d_k, \tag{7}$$

where x_{k+1} and x_k are the new iterate and previous iterate respectively, α_k the step length, while c_k and d_k are the search directions respectively. The double direction iterative method for unconstrained optimization has been proposed by many researchers in the literature [24–26] with iterative scheme as (7). Abdullahi et al. [27] used the scheme in (7) and presented two directions defined as $c_k = -F(x_k)$ and d_k as:

$$d_k = \begin{cases} -F(x_k), & \text{if} \quad k = 0, \\ -F(x_k) + \beta_k^* d_{k-1} - v_k y_k, & \text{if} \quad k \geq 1, \end{cases} \tag{8}$$

with $\beta_k^* = \frac{(y_k - s_k)^T F(x_k) + v_k \|y_k\|^2}{y_k^T d_{k-1}}$ and $v_k = \frac{F(x_k)^T d_{k-1}}{\|F(x_k)\|^2}$.

To improve the numerical performance of double direction method in [27], Abdullahi et al. [28] presented a modified double direction method for solving large-scale symmetric nonlinear equations. This is made possible by using the spectral secant condition in [29] as well as combining the CG direction proposed in [20] with the classical Newton's direction. The CG search direction d_k was obtained as in (8), with

$$\beta_k = \frac{(\theta_k y_k - s_k)^T F(x_k) + \theta_k v_k \|y_k\|^2}{\theta_k y_k^T d_k}, \tag{9}$$

where $\theta_k = \frac{s_k^T s_k}{s_k^T y_k}$ [14], and $v_k = \frac{F(x_k)^T d_{k-1}}{\|F(x_k)\|^2}$.

One of the advantages of double direction method over single direction is the presence of two direction vectors which work jointly as correction factors toward boosting the convergence of the system [30–35]. It also helps the system in such a way that if one of the directions fail, the other direction will automatically correct the system [24]. For nonlinear monotone equations with applications, see [5–9] among others.

Motivated by the work of Abdullahi et al. [28], we intend to obtain a new direction by hybridizing a three-term spectral CG direction obtained by extending the direction proposed by Birgin and Mertinez [36] with the direction proposed by Halilu and Waziri [23].

Throughout this paper, the space $\mathbb{R}^n$ denote the n-dimensional real space equipped with the Euclidean norm $\|\cdot\|$, and $F_k = F(x_k)$. The remaining part of this paper is organized as thus; the derivation of the proposed method is presented in Section 2 followed by the convergence analysis in Section 3 then numerical results and comparisons in Section 4. Finally, conclusions are drawn in Section 5.

2. Main result

Here, we derive our proposed acceleration parameter η_k by combining a three-term spectral direction obtained by extending the direction proposed by Birgin and Mertinez [36] with the direction proposed by Halilu and Waziri [23]. i.e.

$$d_k^{TTSCG} = -\theta_k F_k + \beta_k s_{k-1} - \varepsilon_k y_{k-1}, \tag{10}$$

where $\varepsilon_k = \frac{\theta_k s_{k-1}^T F_k}{y_{k-1}^T s_{k-1}}$ [18], $\theta_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}$ [37] and

$$d_k^{IDFDD} = -\gamma_k^{-1} F_k, \tag{11}$$

where

$$\gamma_k = \frac{y_{k-1}^T y_{k-1}}{y_{k-1}^T s_{k-1}}. \tag{12}$$

Our proposed direction is:

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ (1-\lambda_k) d_k^{IDFDD} + \lambda_k d_k^{TTSCG}, & \text{if} \quad k \geq 1. \end{cases} \tag{13}$$

Substituting (10) and (11) in (13), we have:

$$d_k = -\gamma_k^{-1} F_k + \gamma_k^{-1} \lambda_k F_k - \theta_k \lambda_k F_k + \beta_k \lambda_k s_{k-1} - \lambda_k \varepsilon_k y_{k-1}. \tag{14}$$

Recall the Quasi-Newton's direction:

$$d_k^{QN} = -B_k^{-1} F_k. \tag{15}$$

From (14) and (15), we have:

$$-B_k^{-1} F_k = -\gamma_k^{-1} F_k + \gamma_k^{-1} \lambda_k F_k - \theta_k \lambda_k F_k + \beta_k \lambda_k s_{k-1} - \lambda_k \varepsilon_k y_{k-1}, \tag{16}$$

which implies that:

$$F_k = \gamma_k^{-1} B_k F_k - \gamma_k^{-1} \lambda_k B_k F_k + \theta_k \lambda_k B_k F_k - \beta_k \lambda_k B_k s_{k-1} + \lambda_k \varepsilon_k B_k y_{k-1}. \tag{17}$$

Multiply through by s_{k-1}^T, i.e.:

$$s_{k-1}^T F_k = \gamma_k^{-1} s_{k-1}^T B_k F_k - \gamma_k^{-1} \lambda_k s_{k-1}^T B_k F_k + \theta_k \lambda_k s_{k-1}^T B_k F_k - \beta_k \lambda_k s_{k-1}^T B_k s_{k-1} + \lambda_k \varepsilon_k s_{k-1}^T B_k y_{k-1}. \tag{18}$$

By secant equation (i.e. $B_k s_{k-1} = y_{k-1}$) and the symmetric property of B_k, we have:

$$s_{k-1}^T B_k = y_{k-1}^T. \tag{19}$$

From (18) and (19), we get:

$$s_{k-1}^T F_k = \gamma_k^{-1} y_{k-1}^T F_k - \gamma_k^{-1} \lambda_k y_{k-1}^T F_k + \theta_k \lambda_k y_{k-1}^T F_k - \beta_k \lambda_k y_{k-1}^T s_{k-1} + \lambda_k \varepsilon_k y_{k-1}^T y_{k-1}. \tag{20}$$

Now, from (20):

$$s_{k-1}^T F_k = \gamma_k^{-1} y_{k-1}^T F_k + \lambda_k (\theta_k y_{k-1}^T F_k - \gamma_k^{-1} y_{k-1}^T F_k - \beta_k y_{k-1}^T s_{k-1} + \varepsilon_k y_{k-1}^T y_{k-1}), \tag{21}$$

Applying some algebraic calculation, we can write our proposed parameter as:

$$\lambda_k = \frac{(s_{k-1} - \gamma_k^{-1} y_{k-1})^T F_k}{(\theta_k y_{k-1} - \gamma_k^{-1} y_{k-1})^T F_k - \beta_k y_{k-1}^T s_{k-1} - \varepsilon_k \|y_{k-1}\|}, \tag{22}$$

where $\gamma_k = \frac{y_{k-1}^T y_{k-1}}{y_{k-1}^T s_{k-1}}$, and we choose $\beta_k^{FR} = \frac{\|F_k\|^2}{\|F_{k-1}\|^2}$.

One can observe that in (13), if $\lambda_k = 0$ then we have (11) (i.e. d_k^{IDFDD}), if $\lambda_k = 1$ then we have (10). However, if $\lambda_k \in (0,1)$ then we have (13).

Furthermore, we used the derivative-free line search proposed by Li and Fukushima [1] in order to compute our step-length α_k. Suppose that, $\omega_1 > 0$, $\omega_2 > 0$ and $r \in (0,1)$ be constants and let $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty. \tag{23}$$

Hence, the step-length α_k can be computed as follows:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1 \|\alpha_k F_k\|^2 - \omega_2 \|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{24}$$

Let i_k be the smallest non negative integer i such that (24) holds for $\alpha = r^i$. and let $\alpha_k = r^{i_k}$. We then describe our algorithm as follows:

Algorithm 2.1 (*DDTTS*).

STEP 1: Given $x_0 \in \mathbb{R}^n$, $\epsilon = 10^{-4}$, $d_0 = -F_0$, set $k = 0$.
STEP 2: Compute F_k.
STEP 3: If $\|F_k\| \leq \epsilon$ then stop, else go to STEP 4.
STEP 4: Compute the step length α_k using (24).
STEP 5: Determine: $\theta_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}$ and $\epsilon_k = \frac{\theta_k s_{k-1}^T F_k}{y_{k-1}^T s_{k-1}}$, then γ_k, β_k^{FR} and λ_k using (22).
STEP 6: Evaluate the search direction using (13).
STEP 7: Set $x_{k+1} = x_k + \alpha_k d_k$.
STEP 8: Set $k = k + 1$ and go to step 2.

Remark 2.1. We re-scaled the convex parameter λ_k, if in some iterations the value for λ_k is less than 0 we take λ_k to be equal to 0 and if λ_k is greater than 1 then we take λ_k to be equal to 1

3. Convergence result

In this section, convergence analysis of the DDTTS Algorithm is presented. To begin with, let us define the level set

$$\Omega = \{x \in \mathbb{R}^n |\ \|F(x)\| \leq \|F(x_0)\|\}. \tag{25}$$

To analyze the convergence of Algorithm 2.1, the following assumption is needed:

Assumption 3.1.

(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 Derivative of $F(x)$, that is the Jacobian is bounded, Symmetric and positive definite on N. i.e. there exists a positive constant $M > m > 0$, such that,

$$\|F'(x)\| \leq M, \quad \forall x \in N, \tag{26}$$

and

$$m\|d\|^2 \leq d^T F'(x) d, \quad \forall x \in N, \quad d \in \mathbb{R}^n. \tag{27}$$

From the level set we have:

$$\|F(x)\| \leq m_1, \quad \forall x \in \Omega. \tag{28}$$

Remark 3.1. Assumption 3.1 implies that there exists a constant $M > m > 0$ such that

$$m\|d\| \leq \|F'(x)d\| \leq M\|d\|, \quad \forall x \in N, \quad d \in \mathbb{R}^n, \tag{29}$$

$$m\|x - y\| \leq \|F(x) - F(y)\| \leq M\|x - y\|, \quad \forall x, y \in N. \tag{30}$$

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

$$m\|x - x^*\| \leq \|F(x)\| = \|F(x) - F(x^*)\| \leq M\|x - x^*\|, \tag{31}$$

where x^* is a unique solution of (1) in N.

Remark 3.2. To establish the global convergence property of DDTTS Algorithm, the proposed λ_k in (22) needs to be bounded. Therefore, we set

$$|\lambda_k| \leftarrow \min\{|\lambda_k|, H\}, \tag{32}$$

where H is a suitable positive number.

Lemma 3.1 (*See [15]*). *Suppose Assumption 3.1 holds and $\{x_k\}$ be generated by DDTTS Algorithm. Then there exists a constant $m > 0$ such that for all k*

$$y_k^T s_k \geq m\|s_k\|^2. \tag{33}$$

Proof. By mean-value theorem and (27) we have:

$$y_k^T s_k = s_k^T (F_{k+1} - F_k) = s_k^T F'(\zeta) s_k \geq m\|s_k\|^2, \tag{34}$$

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

Using $y_k^T s_k \geq m\|s_k\|^2 > 0$, γ_k is always generated by (12). By Lemma 3.1 and (30), we obtained:

$$\frac{y_k^T s_k}{\|s_k\|^2} \geq m, \quad \frac{\|y_k\|^2}{y_k^T s_k} \leq \frac{M^2}{m}. \tag{35}$$

Lemma 3.2. *Suppose Assumption 3.1 holds and $\{x_k\}$ is generated by the DDTTS Algorithm, then:*

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0 \tag{36}$$

and

$$\lim_{k\to\infty} \|\alpha_k F_k\| = 0. \tag{37}$$

Proof. From (23) and (24), we have for all $k > 0$.

$$\begin{aligned} \omega_2\|\alpha_k d_k\|^2 &\leq \omega_1\|\alpha_k F_k\|^2 + \omega_2\|\alpha_k d_k\|^2, \\ &\leq \frac{1}{2}\|F_k\|^2 - \frac{1}{2}\|F_{k+1}\|^2 + \eta_k \frac{1}{2}\|F_k\|^2. \end{aligned} \tag{38}$$

By summing up (38) up to kth term, we have:

$$\begin{aligned} 2\omega_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 &\leq \sum_{i=0}^{k} \left(\|F_i\|^2 - \|F_{i+1}\|^2\right) + \sum_{i=0}^{k} \eta_i \|F_i\|^2, \\ &= \|F_0\|^2 - \|F_{k+1}\|^2 + \sum_{i=0}^{k} \eta_i \|F_i\|^2, \\ &\leq \|F_0\|^2 + \|F_0\|^2 \sum_{i=0}^{k} \eta_i, \\ &\leq m^2 + m^2 \sum_{i=0}^{\infty} \eta_i. \end{aligned} \tag{39}$$

So from Assumption 1 and the fact that $\{\eta_k\}$ satisfies (23), then the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ is convergent, which implies (36). By similar arguments as the above but with $\omega_1\|\alpha_k F_k\|^2$ on the left hand side, we obtain (37).

Next, the following result shows that the DDTTS Algorithm is globally convergent.

Theorem 3.1. *Suppose that Assumption 3.1 holds, the sequence $\{x_k\}$ generated by the DDTTS Algorithm converges globally. That is,*

$$\liminf_{k\to\infty} \|F_k\| = 0. \tag{40}$$

Proof. We prove by contradiction, that is, suppose that (40) is not true and there exists a positive constant τ such that

$$\|F_k\| \geq \tau. \tag{41}$$

We divide the prove into two (2) parts:

Case (i): Consider $\limsup_{k\to\infty} \alpha_k > 0$. Then from (37) we have $\liminf_{k\to\infty} \|F_k\| = 0$. This with Lemma 3.2 shows that $\lim_{k\to\infty} \|F_k\| = 0$, which contradicts with (41).

Case (ii): Consider $\limsup_{k\to\infty} \alpha_k = 0$. Since $\alpha_k \geq 0$, this case implies that:

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

Next for $\theta_k = \frac{s_k^T s_k}{s_k^T y_k}$, from (33) we have

$$|\theta_k| = \left|\frac{s_k^T s_k}{s_k^T y_k}\right| \leq \frac{\|s_k\|^2}{s_k^T y_k} \leq \frac{\|s_k\|^2}{m\|s_k\|^2} = \frac{1}{m}. \tag{43}$$

Therefore

$$|\theta_k| \leq \frac{1}{m}. \tag{44}$$

Hence, θ_k is bounded.

Next, from the definition of ε_k,

$$\varepsilon_k = \frac{\theta_k s_k^T F_k}{y_k^T s_k}, \tag{45}$$

which gives us:

$$|\varepsilon_k| = \left|\frac{\theta_k s_k^T F_k}{y_k^T s_k}\right| \le \frac{|\theta_k| \|s_k\| \|F_k\|}{m\|s_k\|^2} \le \frac{m_1}{m^2\|s_k\|} \longrightarrow 0. \tag{46}$$

Therefore, there exists a positive number q such that:

$$|\varepsilon_k| < q. \tag{47}$$

Hence, ε_k is bounded.

Also, from the definition of β_k;

$$\begin{aligned} |\beta_k| &= \left|\frac{F_{k+1}^T F_{k+1}}{F_k^T F_k}\right|, \\ &\le \frac{\|F_{k+1}\|^2}{\|F_k\|^2}, \\ |\beta_k| \|F_k\|^2 &\le \|F_{k+1}\|^2 \le m_1^2, \\ |\beta_k| \|F_k\|^2 &\le m_1^2. \end{aligned} \tag{48}$$

But for $a.b \le c$, then either $a \le c$ or $b \le c$ $\forall a, b, c > 0$.

Therefore from (48) we obtain;

$$|\beta_k| \le m_1^2. \tag{49}$$

Hence, $|\beta_k|$ is also bounded.

Now from (13), Remark 3.2, and using Cauchy–Schwarz inequality, we have

$$\begin{aligned} \|d_k\| &= \|(1-\lambda_k) d_k^{IDFDD} + \lambda_k d_k^{TTSCG}\| = \|d_k^{IDFDD} - \lambda_k d_k^{IDFDD} + \lambda_k d_k^{TTSCG}\|, \\ &\le \|d_k^{IDFDD}\| - \|\lambda_k d_k^{IDFDD}\| + \|\lambda_k d_k^{TTSCG}\|, \\ &\le \|d_k^{IDFDD}\| + \|\lambda_k d_k^{TTSCG}\|, \\ &= \|\frac{y_{k-1}^T s_{k-1} F_k}{y_{k-1}^T y_{k-1}}\| + \|\lambda_k(-\theta_k F_k + \beta_k s_{k-1} - \varepsilon_k y_{k-1})\|, \\ &\le \frac{\|y_{k-1}^T s_{k-1} F_k\|}{m^2\|s_{k-1}\|^2} + |\lambda_k|(\| - \theta_k F_k + \beta_k s_{k-1} - \varepsilon_k y_{k-1}\|), \\ &\le \frac{\|y_{k-1}\| \|s_{k-1}\| \|F_k\|}{m^2\|s_{k-1}\|^2} + |\lambda_k|(|\theta_k| \|F_k\| + |\beta_k| \|s_{k-1}\| + |\varepsilon_k| \|y_{k-1}\|), \\ &\le \frac{M\|s_{k-1}\|^2 \|F_k\|}{m^2\|s_{k-1}\|^2} + |\lambda_k|(|\theta_k| \|F_k\| + |\beta_k| \|s_{k-1}\| + |\varepsilon_k| M\|s_{k-1}\|), \\ &\le \frac{Mm_1}{m^2} + H(\frac{1}{m}m_1 + m_1^2 p + \frac{m_1}{m^2}M). \end{aligned} \tag{50}$$

Since $\lim_{k\to\infty} \|s_k\| = 0$, then there exists a positive number p, such that $\|s_k\| < p$. Taking $M_2 = \frac{Mm_1}{m^2} + H(\frac{1}{m}m_1 + m_1^2 p + \frac{m_1}{m^2}M)$ we have $\|d_k\| \le M_2$ and hence $\{d_k\}$ is bounded.

Since $\lim_{k\to\infty} \alpha_k = 0$, then $\alpha_k' = \frac{\alpha_k}{r}$ does not satisfy (24), that is:

$$f(x_k + \alpha_k d_k) - f(x_k) > -\omega_1 \|\alpha_k' F(x_k)\|^2 - \omega_2 \|\alpha_k' d_k\|^2 + \eta_k f(x_k), \tag{51}$$

which implies:

$$\frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k'} > -\omega_1 \|\alpha_k' F(x_k)\|^2 - \omega_2 \|\alpha_k' d_k\|^2. \tag{52}$$

By the mean-value theorem, there exists $\delta_k \in (0,1)$ such that:

$$\frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k'} = g(x_k + \delta_k \alpha_k' d_k)^T d_k. \tag{53}$$

Ortega and Rheinboldt [38] presented an approximation to the gradient $g_k = F_k$ in order to avoid computing exact gradient as:

$$F_k = \frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k}. \tag{54}$$

Since $\{x_k\} \subset \Omega$ is bounded, without loss of generality, we assume that $x_k \to x^*$. From (13) and (54), we obtain:

$$\begin{aligned}\lim_{k\to\infty} d_k &= \lim_{k\to\infty}((1-\lambda_k)d_k^{IDFDD} + \lambda_k d_k^{TTSCG}),\\ &\leq -\lim_{k\to\infty} F_k + Hm_1 \lim_{k\to\infty}\|s_{k-1}\|,\\ &\leq -F(x^*).\end{aligned} \tag{55}$$

That is using (13), (43)–(49) and the fact that the sequence $\{d_k\}$ is bounded, i.e (50).

On the other hand, we have:

$$\lim_{k\to\infty} g(x_k + \delta_k \alpha_k' d_k) = F(x^*). \tag{56}$$

Therefore, from (52)–(56), it follows that $-F(x^*)^T F(x^*) \geq 0$. That is $\|F(x^*)\| = 0$. Hence contradiction with (41). Which completes the proof.

4. Numerical results

In this section, we present the numerical performance of our proposed DDTTS-method for solving (1) which is compared with IDFDD-method proposed in [23] and STTCG-method proposed in [19]. For unbiasedness, we set $\omega_1 = 10^{-4}$, $\omega_2 = 10^{-4}$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$ for the three methods.

The codes were written in MATLAB R2014a 7.71 GB and run on a personal computer with Windows 10pro, intel(R) core(TM)i3-3217U 1.8 GHz CPU processor and 4 GB RAM memory. We use the symbol (—) if the algorithm fails to find a solution and is terminated if the total number of iterations exceeds 1000 but no x_k satisfying $\|F(x_k)\| \leq 10^{-4}$ or failure on code execution due to insufficient memory. The methods were tested using ten test problems with different initial points and dimensions (n values).

Problem 1 (*[23]*).
$F_1(x) = x_1(x_1^2 + x_2^2) - 1,$
$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2),$
$F_n(x) = x_n(x_{n-1}^2 + x_n^2).$
$i = 2, 3, \dots, n-1.$
$x_0 = (0.09, 0.09, \dots, 0.09)^T.$

Problem 2 (*[39]*).
$F_1(x) = 3x_1^3 + 2x_2 - 5 + (sin(x_1 - x_2))(sin(x_1 + x_2)),$
$F_i(x) = -x_{i-1}e^{(x_{i-1}-x_i)} + x_i(4 + 3x_i^2) + 2x_{i+1} + (sin(x_i - x_{i+1}))(sin(x_i + x_{i+1})) - 8,$
$F_n(x) = -x_{n-1}e^{(x_{n-1}-x_n)} + 4x_n - 3,$
$i = 2, 3, \dots, (n-1).$
$x_0 = (0.5, 0.5, \dots, 0.5)^T.$

Problem 3 (*[40]*).
$$F_i(x) = x_i - \left(1 - \frac{c}{2n}\sum_{j=1}^{n}\frac{(\mu_i x_j)}{(\mu_i + \mu_j)}\right)^{-1}, \ for\ i = 1, 2, \dots, n.$$
$wth\ c = 2,\ and\ \mu_i = \frac{i-0.5}{n},\ for\ 1 \leq i \leq n.$
$x_0 = (0.25, 0.25, \dots, 0.25)^T.$

Problem 4 (*[40]*).
$F_i(x) = x_i - 3x_i(\frac{sin(x_i)}{3} - 0.66) + 2.$
$i = 1, 2, \dots, n.$
$x_0 = (0.05, 0.05, \dots, 0.05)^T.$

Problem 5 (*[23]*).
$F_1(x) = x_1 - e^{cos\left(\frac{x_1+x_2}{n+1}\right)},$
$F_i(x) = x_i - e^{cos\left(\frac{x_{i-1}+x_i+x_{i+1}}{n+1}\right)},$
$F_n(x) = x_n - e^{cos\left(\frac{x_{n-1}+x_n}{n+1}\right)},$
$i = 2, 3, \dots, n-1.$
$x_0 = (0.7, 0.7, \dots, 0.7)^T.$

Problem 6 (*[23]*).
$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2.$
$i = 1, 2, \dots, n.$
$x_0 = (0.03, 0.03, \dots, 0.03)^T.$

Table 1
Numerical results of DDTTS, IDFDD and STTCG methods for Problems 1–5.

		DDTTS			IDFDD			STTCG		
Prob.	Dim	NI	T (s)	$\|F(x)\|$	NI	T (s)	$\|F(x)\|$	NI	T (s)	$\|F(x)\|$
1	100	44	0.025212	9.32E−05	36	0.016401	8.19E−05	58	0.049915	9.82E−05
	1000	38	0.035606	9.29E−05	37	0.032017	9.89E−05	58	0.0524	2.49E−05
	10000	27	0.177236	9.81E−05	39	0.265441	8.84E−05	57	0.412654	9.35E−05
	100000	20	1.201372	8.55E−05	41	2.795064	7.90E−05	60	3.668323	6.80E−05
	1000000	30	23.31443	8.38E−05	43	34.34329	8.03E−05	58	48.55991	9.09E−05
2	100	20	0.015887	7.12E−05	20	0.016951	9.68E−05	48	0.039109	9.76E−05
	1000	23	0.033186	8.73E−05	21	0.041152	6.99E−05	48	0.083647	9.20E−05
	10000	25	0.257481	9.70E−05	21	0.297642	8.56E−05	48	0.675494	8.48E−05
	100000	24	3.308572	5.44E−05	22	3.32404	6.20E−05	47	6.963354	9.19E−05
	1000000	23	32.5595	6.56E−05	22	38.40454	7.59E−05	46	79.10073	9.96E−05
3	100	62	0.024261	4.76E−05	–	–	–	10	0.007312	9.42E−05
	1000	78	0.059805	7.95E−05	–	–	–	12	0.017436	5.53E−05
	10000	79	0.42772	6.69E−05	–	–	–	14	0.108438	7.88E−05
	100000	79	4.265275	8.72E−05	–	–	–	16	1.057687	4.83E−05
	1000000	77	58.19483	7.46E−05	–	–	–	17	17.33523	8.59E−05
4	100	6	0.010934	3.50E−08	27	0.010731	9.41E−05	13	0.008021	8.73E−05
	1000	6	0.007775	1.43E−05	30	0.019695	7.81E−05	15	0.013233	4.42E−05
	10000	7	0.03862	1.82E−06	33	0.157053	6.48E−05	16	0.093596	5.59E−05
	100000	8	0.408793	2.31E−07	35	1.582895	8.39E−05	17	0.780552	7.07E−05
	1000000	8	5.004634	7.31E−07	38	22.10226	6.96E−05	18	12.16126	8.94E−05
5	100	3	0.011114	6.61E−08	45	0.01772	9.13E−05	12	0.016101	7.75E−05
	1000	2	0.002286	1.77E−07	81	0.060436	9.59E−05	5	0.007937	7.02E−05
	10000	1	0.007254	8.44E−05	97	0.554222	7.67E−05	1	0.008024	8.44E−05
	100000	1	0.065998	2.67E−06	101	6.298201	8.09E−05	1	0.060372	2.67E−06
	1000000	1	0.747329	8.44E−08	105	68.54312	8.54E−05	1	0.794783	8.44E−08

Problem 7 (*[23]*).
$F_i(x) = x_i - 0.1x_{i+1}^2$,
$F_n(x) = x_n - 0.1x_1^2$.
$i = 1, 2, \dots, n-1$.
$x_0 = (1.0, 1.0, \dots, 1.0)^T$.

Problem 8 (*[23]*).
$F_{3i-2}(x) = x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1$,
$F_{3i-1}(x) = x_{3i-2}x_{3i-2}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2$,
$F_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}$.
$i = 1, \dots, \frac{n}{3}$.
$x_0 = (0.4, 0.4, \dots, 0.4)^T$.

Problem 9 (*[23]*).

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (sinx_1 - 1, \dots, sinx_n - 1)^T.$$

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

Problem 10 (*[23]*).

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (e_1^x - 1, \dots, e_n^x - 1)^T.$$

$x_0 = (0.08, 0.08, \dots, 0.08)^T$.

The numerical results of the three methods were presented in Tables 1 and 2, where "NI" and "T" stand for total number of iterations and the CPU time in seconds respectively, while $\|F(x_k)\|$ is the norm of the residual at the stopping point. From Tables 1 and 2, we can easily observe that both the methods attempt to solve system of nonlinear equations in (1). In Table 3, the summary of the reported numerical results in Tables 1 and 2 were presented in order to highlight which among the three methods is a

Table 2
Numerical results of DDTTS, IDFDD and STTCG methods for Problems 6–10.

		DDTTS			IDFDD			STTCG		
Prob.	Dim	NI	T (s)	$\|F(x)\|$	NI	T (s)	$\|F(x)\|$	NI	T (s)	$\|F(x)\|$
6	100	3	0.003849	1.44E−05	20	0.011387	7.09E−05	8	0.006176	7.72E−05
	1000	3	0.004323	4.57E−05	22	0.015638	9.18E−05	9	0.010308	9.77E−05
	10000	4	0.020527	5.80E−08	25	0.135121	7.61E−05	11	0.054425	4.94E−05
	100000	4	0.199151	1.81E−05	27	1.258718	9.86E−05	12	0.541246	6.25E−05
	1000000	5	3.032572	2.29E−06	30	18.28974	8.17E−05	13	9.009599	7.91E−05
7	100	3	0.00863	1.00E−06	71	0.02453	7.84E−05	11	0.006375	8.52E−05
	1000	3	0.004224	3.16E−06	75	0.035569	8.27E−05	13	0.012553	4.31E−05
	10000	3	0.013053	1.00E−05	79	0.242121	8.73E−05	14	0.063823	5.45E−05
	100000	3	0.099408	3.16E−05	83	2.725867	9.21E−05	15	0.551168	6.89E−05
	1000000	3	1.268054	1.00E−04	87	35.3069	9.71E−05	16	8.475785	8.72E−05
8	100	29	0.021669	8.82E−05	34	0.01471	7.15E−05	132	0.070924	7.43E−05
	1000	30	0.030912	8.98E−05	37	0.042322	7.84E−05	148	0.172315	8.00E−05
	10000	27	0.153632	7.58E−05	40	0.243499	8.57E−05	164	1.295703	8.59E−05
	100000	26	1.231458	4.80E−05	43	2.085523	9.37E−05	180	12.21121	9.22E−05
	1000000	44	28.73111	7.55E−05	47	30.97109	7.20E−05	196	191.3045	9.91E−05
9	100	34	0.218527	7.52E−05	35	0.27144	7.87E−05	48	0.375593	9.17E−05
	1000	37	0.973078	8.92E−05	38	1.219254	9.21E−05	47	1.607772	8.68E−05
	10000	40	71.33891	7.64E−05	42	92.88534	8.03E−05	45	98.26631	9.38E−05
	100000	–	–	–	–	–	–	–	–	–
	1000000	–	–	–	–	–	–	–	–	–
10	100	13	0.144554	9.98E−05	31	0.241334	9.98E−05	34	0.230571	9.91E−05
	1000	13	0.279925	7.32E−05	34	0.918283	8.60E−05	38	1.090022	8.58E−05
	10000	16	17.59558	8.78E−05	40	60.41658	9.23E−05	30	45.88237	8.24E−05
	100000	–	–	–	–	–	–	–	–	–
	1000000	–	–	–	–	–	–	–	–	–

Table 3
Summary of results from table 1 and 2 for DDTTS, IDFDD and STTCG methods.

	Method	NI	Percentage	T	Percentage
Number of problems and percentage for each method with respect to iterations and CPU time.	DDTTS	31	62%	35	70%
	IDFDD	6	12%	3	6%
	STTCG	5	10%	8	16%
	Undecided	8	16%	4	8%
	Total	50	100%	50	100%

winner with respect to number of iterations and CPU time. The summary indicated that DDTTS method is more effective in terms of number of iterations as it solved 62% (31 out of 50) of all the problems with least number of iterations compared to IDFDD which solved 12%(6 out of 50) and STTCG which solved 10% (5 out of 50). For the undecided, it indicated that 16% (8 out of 50) of the problems were either solved by two or all the three methods with the same number of iterations or failed by two or the three methods concurrently. Similarly, it indicated that DDTTS method outperforms the other methods with respect to CPU time as it solved 70% (35 out of 50) of all the problems with least CPU time against IDFDD algorithm which solved 6% (3 out of 50) and STTCG algorithm which solved 16% (8 out of 50). Here for the undecided, it indicated that 8% (4 out of 50)of the problems failed for all the three algorithms concurrently.

Figs. 1 and 2 are generated from the results obtained in Tables 1–2 have shown the performance of our method relative to the number of iterations and CPU time which were evaluated using the profiles of Dolan and Moré [41]. That is, for each method we plot the fraction $F(\mu)$ of the problems for which the method is within a factor μ of the best time where the top curve stand for our proposed DDTTS method and the bottom curves stand for the IDFDD and STTCG methods. From the two figures, it was observed that DDTTS method solved the most problems as indicated on the vertical axis of each figure which represent percentage of the test problems for which a method is the best with respect to least number of iterations or CPU time. Hence, DDTTS method is more effective and efficient than the other methods with least number of iterations and CPU time. Hence, from the summary Table and Figs. 1 and 2, we conclude that our proposed DDTTS-method is more effective and efficient for solving large-scale symmetric system of nonlinear equations than IDFDD and STTCG methods.

5. Conclusion

In this paper, a Three-Term Spectral Conjugate Gradient method via double direction approach for solving large-scale symmetric system of nonlinear equations was presented and its performance was compared with that of IDFDD and STTCG methods proposed in [19,23] respectively, by conducting some numerical experiments. We however proved the convergence of our proposed method using a backtracking type line search [1], and the numerical results showed that our proposed method is more efficient in terms of accuracy and robustness and hence promising.

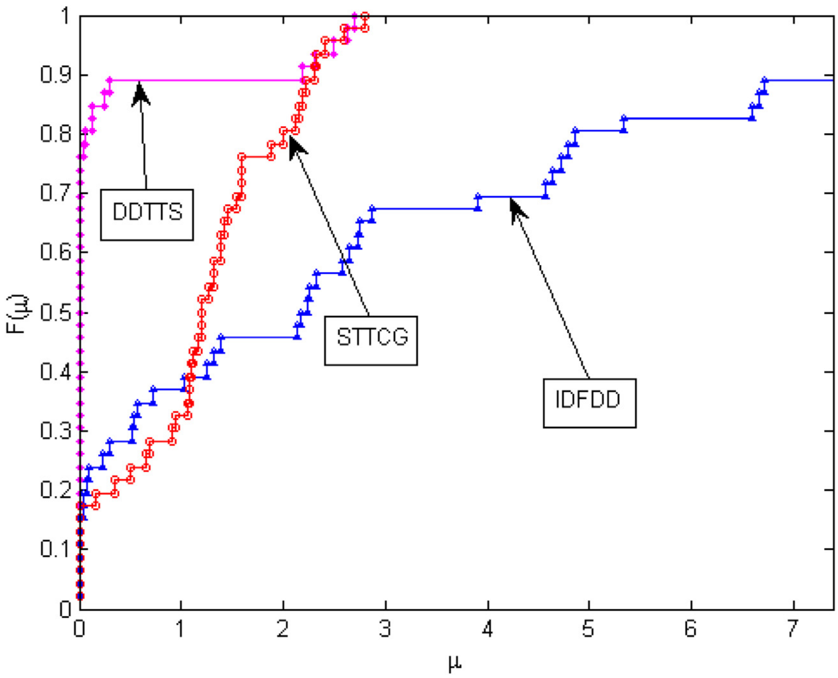

Fig. 1. Performance profile of DDTTS, IDFDD and STTCG methods with respect to the number of iterations for Problems 1–10.

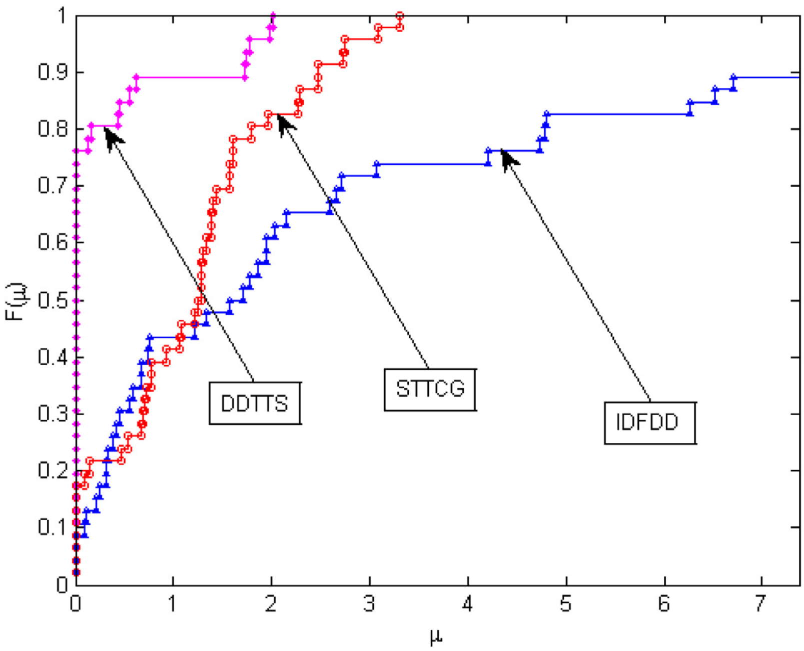

Fig. 2. Performance profile of DDTTS, IDFDD and STTCG methods with respect to the CPU time (in seconds) for Problems 1–10.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Li D, Fukushima M. A global and superlinear convergent Gauss–Newton base BFGS method for symmetric nonlinear equation. SIAM J Numer Anal 2000;37(1):152–72.
[2] Halilu AS, Waziri MY. A transformed double step-length method for solving large-scale system of nonlinear equations. J Numer Math Stoch 2017;9(1):20–32.
[3] Waziri MY, Leong WJ, Hassan MA, Monsi M. Jacobian computation-free Newton's method for system of nonlinear equations. J Numer Math Stoch 2010;2:54–63.
[4] Waziri MY, Leong WJ, Hassan MA, Monsi M. A new Newton's method with diagonal Jacobian approximation for system of nonlinear equations. J Math Stat 2010;6(3):246–52.
[5] Abubakar AB, Kumam P, Mohammed H. A note on the spectral gradient projection method for nonlinear monotone equations with applications. Comput Appl Math 2020;39(129).
[6] Abubakar AB, Kumam P, Mohammed H, Awwal AM. An efficient conjugate gradient method for convex constrained monotone nonlinear equations with applications. Mathematics 2019;7(9):767.
[7] Abubakar AB, Kumam P, Mohammed H, Awwal AM, Sitthithakerngkiet K. A modified Fletcher-Reeves conjugate gradient method for monotone nonlinear equations with some applications. Mathematics 2019;7(8):745.
[8] Ibrahim AH, et al. Least-square-based three-term conjugate gradient projection method for l_1-norm problems with application to compressed sensing. Mathematics 2020;8:602.
[9] Ibrahim AH, et al. A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. Heliyon 2020;6:e03466.
[10] Halilu AS, Majumder A, Waziri MY, Ahmed K. Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach. Math Comput Simulation 2021;187:520–39.
[11] Barzilai J, Borwein JM. Two point step-size gradient method. IMA J Numer Anal 1988;8:141–8.
[12] Hager WW, Zhang H. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J Optim 2005;16(1):170–92.
[13] Al-baali M. Descent property and global convergence of Fletcher-Reeves methods with inexact line search. IMA J Numer Anal 1985;5:121–4.
[14] Waziri MY, Sabi'u J. A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. Int J Math Math Sci 2015;961487:8.
[15] Yuan G, Lu X. A new backtracking inexact BFGS method for symmetric nonlinear equations. Comput Math Appl 2008;55(1):116–29.
[16] Yuan G, Zhang M. A three-terms Polak-Ribiere-Polyak conjugate gradient algorithm for large-scale nonlinear equations. J Comput Appl Math 2015;286:186–95.
[17] Waziri MY, Muhammad L. An accelerated three-term conjugate gradient algorithm for solving large-scale ststems of nonlinear equations. Sohag J Math 2017;4(2):1–8.
[18] Muhammad L, Waziri MY. An alternative three-term conjugate gradient algorithm for system of nonlinear equations. Int J Math Model Comput 2017;7(2):145–57.
[19] Waziri MY, Muhammad L, Sabiu J. A simple three-term conjugate gradient algorithm for solving symmetric system of nonlinear equations. Int J Adv Appl Sci 2017;306914657.
[20] Zhang L, Zhou W, Li DH. A descent modified Polak-Ribiere-Polyak conjugate gradient method and its global convergence. IMA J Numer Anal 2006;26(4):629–40.
[21] Zhou W, Shen D. Convergence properties of an iterative method for solving symmetric non-linear equations. J Optim Theory Appl 2015;164(1):277–89.
[22] Waziri MY, Kufena YM, Halilu AS. Derivative-free three-term spectral conjugate gradient method for symmetric nonlinear equations. Thai J Math 2020;18(3):1417–31.
[23] Halilu AS, Waziri MY. An improved derivative-free method via double direction approach for solving system of nonlinear equations. J. Ramanujan Math Soc 2018;33(1):75–89.
[24] Dbaruranovic-milicic NI. A multi-step curve search algorithm in nonlinear optimization. Yugosl J Oper Res 2008;18:47–52.
[25] Petrovic MJ, Stanimirovic PS. Accelerated double direction method for solving unconstrained optimization problems. Math Probl Eng 2014;965104:8.
[26] Dbaruranovic-milicic NI, Gardasevic-Filipovic M. A multi-step curve search algorithm in nonlinear convex case. Facta Univ Ser Math Inf 2010;25:11–24.
[27] Abdullahi H, Waziri MY, Dauda MK. A double direction conjugate gradient method for solving large-scale system of nonlinear equations. J. Math Comput Sci 2017;7(3):606–24.
[28] Abdullahi H, Halilu AS, Waziri MY. A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear equations. J. Numer Math Stoch 2018;10(1):32–44.
[29] Liu H, Yao Y, Qian X, Wang H. Some nonlinear conjugate gradient methods based on spectral scaling secant equations. Comput Appl Math 2016;35(2):639–51.
[30] Halilu AS, Waziri MY. Solving systems of nonlinear equations using improved double direction method. J Nigerian Math Soc 2020;32(2):287–301.
[31] Halilu AS, Majumder A, Waziri MY, Awwal AM, Ahmed K. On solving double direction methods for convex constrained monotone nonlinear equations with image restoration. J Comput Appl Math 2021. http://dx.doi.org/10.1007/s40314-021-01624-1.
[32] Halilu AS, Waziri MY, Yusuf I. Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations. Yugosl J Oper Res 2020;30(4):399–412.
[33] Halilu AS, Waziri MY. Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations. Int J App Math Res 2017;6:147–56.
[34] Halilu AS, Waziri MY. Inexact double step length method for solving systems of nonlinear equations. Stat Optim Inf Comput 2020;8:165–74.
[35] Halilu AS, Majumder A, Waziri MY, Abdullahi H. Double direction and step length method for solving system of nonlinear equations. Euro J Mol Clin Med 2020;7(7):3899–913.
[36] Birgin EG, Martınez JM. A spectral conjugate gradient method for unconstrained optimization. Appl Math Optim 2001;43(2):117–28.
[37] Raydan M. The barzilai and borwein gradient method for large scale unconstrained minimization problem. SIAM J Optim 1997;7(1):26–33.
[38] Ortega JM, Rheinboldt WC. Iterative solution of nonlinear equations in several variables. New York, USA: Academic Press; 1970.
[39] La Cruz W, Martinez JM, Raydan M. Spectral residual method without gradient information for solving large-scale nonlinear system of equations. Theory Exp P Optim 2017;6:76–9.
[40] Jamilu S, Waziri MY, Abba I. A new hybrid dai-yuan and hestenes-stiefel conjugate gradient parameter for solving system of nonlinear equations. MAYFEB J Math 2017;1:44–55.
[41] Dolan E, Moré JJ. Benchmarking optimization software with performance profiles. Math Program 2002;91(2):201–13.