Thai Journal of Mathematics
Volume 18 Number 3 (2020)
Pages 1417–1431

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Dedicated to Prof. Suthep Suantai on the occasion of his 60^{th} anniversary

Derivative-Free Three-Term Spectral Conjugate Gradient Method for Symmetric Nonlinear Equations

Mohammed Yusuf Waziri[1,4,*], Yusuf Muhammad Kufena[2,4] and Abubakar Sani Halilu[3,4]

[1] *Department of Mathematical Science, Bayero University, Kano, Nigeria*
e-mail : mywaziri.mth@buk.edu.ng (M. Y. Waziri)

[2] *Department of Mathematics and Statistics, Nuhu Bamalli Polytechnic, Zaria, Nigeria*
e-mail : mathstardon@gmail.com (Y. M. Kufena)

[3] *Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria*
e-mail : ashalilu.mth@gmail.com (A. S. Halilu)

[4] *Numerical Optimization Research Group, Bayero University, Kano, Nigeria*

Abstract In this paper, we proposed a new derivative-free three-term spectral conjugate gradient (DFTTS) method via extending the direction proposed by Birgin and Mertinez [E.G. Birgin, J.M. Martinez, A spectral Conjugate Gradient Method for Unconstrained Optimization, Appl. Math. Optim. 43 (2) (2001) 117–128] to three-term together with the classical Newton's direction. One of the important properties of the proposed method is that, it generated a descent direction using inexact line search. The global convergence of the proposed algorithm was established under appropriate conditions. Numerical results for the benchmark test problems demonstrated an improved efficiency of the method over some existing ones.

MSC: 49K35; 47H10; 20M12
Keywords: spectral conjugate gradient method; secant condition; descent condition; global convergence

Submission date: 15.03.2020 / Acceptance date: 08.07.2020

1. Introduction

We consider the following system of nonlinear equations:

$$F(x) = 0, \tag{1.1}$$

where $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, is a nonlinear map.
Throughout this paper, we take $y_k = F_{k+1} - F_k$, $s_k = x_{k+1} - x_k$ and $F(x_k) = F_k$ and we used $\|.\|$ to denote the Euclidean norm of vectors. Also, (1.1) can be obtained from

*Corresponding author.

Published by The Mathematical Association of Thailand.

an unconstrained optimization problem, a saddle point and equality constrained problem [2]. Let f be a function defined by:

$$f(x) = \frac{1}{2}\|F(x)\|^2. \tag{1.2}$$

Problem (1.1) is equivalent to the unconstrained optimization problem:

$$min f(x), \quad x \in \mathbb{R}^n. \tag{1.3}$$

Several methods have been developed for solving nonlinear systems of equations. Most of these methods fall into the Newton's and quasi-Newton's approaches and are particularly welcomed because of their rapid convergence properties from a sufficiently good initial guess.

Newton's and quasi-Newton's methods are unattractive for large-scale nonlinear systems of equations because they require computation and storage of the Jacobian matrix and its inverse or its approximation, they however, require solving n linear system of equations in each iteration, or convergence may even be lost when the Jacobian is singular. Hence, various methods have been developed to handle such problems. For some of the numerical methods for solving (1.1) see [2–5]. Newton's method generates sequence of points via $x_{k+1} = x_k - F'_k F_k$, $k = 0, 1, 2, 3, ...$, where F'_k is the Jacobian of F at x_k.

The conjugate gradient method was proposed in order to reduce or overcome the shortcomings of Newton's and Quasi-Newton's methods. It is a popular method used to solve (1.1) and efficient for handling large-scale problems because of its convergence properties, simple implementation and low storage requirement [6]. A sequence of iterates $\{x_k\}$ via

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.4}$$

where $k = 0, 1, 2, ...$ and $\alpha_k > 0$ is the step-size which is obtained using line search, and the conjugate gradient direction d_k using:

$$d_k = \begin{cases} -F_k, \text{ if } k = 0, \\ -F_k + \beta_k d_{k-1}, \text{ if } k \geqslant 1, \end{cases} \tag{1.5}$$

where β_k is the CG-parameter. Different CG-algorithms correspond to different choices of β_k in (1.5). Some of the well known $\beta_k's$ are:
$\beta_k^{HS} = \frac{F_{k+1}^T y_k}{d_k^T y_k}$ [7], $\beta_k^{FR} = \frac{\|F_{k+1}\|^2}{\|F_k\|^2}$ [8], $\beta_k^{PRP} = \frac{F_{k+1}^T y_k}{\|F_k\|^2}$ [9, 10], $\beta_k^{CD} = -\frac{\|F_{k+1}\|^2}{d_k^T F_k}$ [11], $\beta_k^{LS} = -\frac{F_{k+1}^T y_k}{d_k^T F_k}$, [12] and $\beta_k^{DY} = \frac{\|F_{k+1}\|^2}{d_k^T y_k}$ [13].

This choice gives rise to distinct CG methods with different computational efficiency and convergent properties.

Spectral gradient method was introduced so as to solve potentially large-scale unconstrained optimization problems whereby only gradient directions are used at each line search which makes the method to outperforms conjugate gradient algorithms in many problems [14]. It generates an iterative sequence of points $\{x_k\}$ via (1.4) and the direction d_k is obtained by:

$$d_k = \begin{cases} -g_k, \text{ if k} = 0, \\ -\theta_k g_k + \beta_k d_{k-1}, \text{ if } k \geqslant 1, \end{cases} \tag{1.6}$$

where θ_k is the Spectral Parameter and g_k is the gradient of f.

In Waziri et al. [15], states that the major shortcoming of CG-methods atimes is their inability to generate descent directions and the global convergence of conjugate gradient methods is relatively connected to the sufficient descent condition:

$$F_k^T d_k \leqslant -\lambda \|F_k\|^2. \tag{1.7}$$

In order to improve the efficiency of the classical conjugate gradient method, a type of three-term conjugate gradient methods was developed.

A three-term CG direction which possessed both sufficient descent and trust region property independent of line search were both the direction and the line search technique are the derivative-free approach was proposed by Yuan and Zhang [16]. Furthermore, a fast and efficient method without computing Jacobian and gradient and with lower storage requirement was derived in Waziri, et. al [15], which was compared to some existing methods using a derivative-free line search proposed by Li and Li (2011).
Also, Waziri and Muhammad [17], proposed a descent direction:

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

where $\delta_k = (1 - min\left\{1, \frac{||y_k||^2}{y_k^T s_k}\right\})\frac{s_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T g_{k+1}}{y_k^T s_k}$ and $\eta_k = \frac{s_k^T g_{k+1}}{y_k^T s_k}$, where the strategies of acceleration and restart were incorporated in designing the algorithm to improve its numerical performance which shows the efficiency of the method than other existing ones.

In this work, we are interested in deriving a derivative free 3-term CG method which will be used to handle large-scale problems with a low storage requirement that is globally convergent. This method is widely used for handling large-scale problems due to their efficiency, convergence properties, simple implementation as well as low storage [18]. Still, the study of three-terms CG methods for large-scale symmetric nonlinear systems of equations is very rear, which is why we are motivated to have this paper.

The recent proposed nonmonotone spectral CG slgorithm [19] falls under the matrix free methods. Li and Wang [20] proposed a modified Fletcher-Reeves CG based on the work of Zhang et al. [21] where the results shows that their proposed method is promising. Furthermore, studies on CG are inspired for solving large-scale nonlinear symmetric equations. The work of Zhang et al. [22] where a descent PRP method was proposed, which was further extended by Zhou and Shen [23] by combining it with the work of Li and Fukushima [2] and successfully used for solving symmetric system of equation (1.1).

The combination of CG algorithms and the Newton's direction was first proposed by Andrei [24, 25]. The direction proposed by Birgin and Martinez[1] was combined with classical Newton's direction where an enhanced CG parameter β_k was proposed by Waziri and Jamilu [26]. Motivated by their work, we present a new enhanced matrix and derivative free CG parameter β_k. This is made possible by combining our proposed three-term direction with classical Newton's direction. More recently, new CG algorithms were presented for solving monotone convex constraints nonlinear equations with applications, see [27–31].

The remaining part of this paper is organized as; 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. Derivation of the Proposed Method

We derive our proposed CG parameter β_k which will be obtain by combining the proposed three-term spectral direction obtained by extending the direction proposed by

Birgin and Mertinez [1] and that of classical Newton's direction to present our proposed search direction as: i.e.

$$d_k = \begin{cases} -F_k, \text{ if k} = 0, \\ -\theta_k F_k + \beta_k s_k - \varepsilon_k y_k, \text{ if } k \geqslant 1, \end{cases} \tag{2.1}$$

where $\theta_k = \frac{s_k^T s_k}{s_k^T y_k}$, (see [32]) and $\varepsilon_k = \frac{\theta_k s_k^T F_k}{y_k^T s_k}$, (see [6]).
Recall the classical Newton's direction:

$$d_k = -(F_k')^{-1} F_k. \tag{2.2}$$

Combining (2.1) and (2.2), we have:

$$-(F_k')^{-1} F_k = -\theta_k F_k + \beta_k s_k - \varepsilon_k y_k. \tag{2.3}$$

Multiplying (2.3) by J_k we get:

$$-F_k = -\theta_k (F_k') F_k + \beta_k (F_k') s_k - \varepsilon_k (F_k') y_k, \tag{2.4}$$

then, multiplying (2.4) by s_k^T we get:

$$-s_k^T F_k = -\theta_k s_k^T (F_k') F_k + \beta_k s_k^T (F_k') s_k - \varepsilon_k s_k^T (F_k') y_k. \tag{2.5}$$

Also from Secant condition:

$$(F_k') s_k = y_k. \tag{2.6}$$

After taking the transpose of (2.6) and the symmetric property of (F_k')we therefore get:

$$s_k^T (F_k') = y_k^T. \tag{2.7}$$

Now, from (2.5) and (2.7) we get:

$$-s_k^T F_k = -\theta_k y_k^T F_k + \beta_k y_k^T s_k - \varepsilon_k y_k^T y_k. \tag{2.8}$$

We, therefore, obtain our proposed β_k as:

$$\beta_k = \beta_k^{WJ} + \frac{\varepsilon_k y_k^T y_k}{y_k^T s_k}, \tag{2.9}$$

where $\beta_k^{WJ} = \frac{(\theta_k y_k - s_k)^T F_k}{y_k^T s_k}$ (see [26]).

Furthermore, we used the derivative-free line search proposed by Li and Fukushima [2] 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

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty. \tag{2.10}$$

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

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1 \|\alpha_k F(x_k)\|^2 - \omega_2 \|\alpha_k d_k\|^2 + \eta_k f(x_k). \tag{2.11}$$

Let i_k be the smallest non negative integer i such that (2.11) holds for $\alpha = r^i$. and let $\alpha_k = r^{i_k}$. We then describe our algorithm as follows:

Algorithm (1) (DFTTS)

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 (2.11).

STEP 5: Determine θ_k, ε_k and β_k using (2.1) and (2.9).
STEP 6: Evaluate the search direction using (2.1).
STEP 7: Set $x_{k+1} = x_k + \alpha_k d_k$.
STEP 8: Set $k = k + 1$ and go to step 2.

3. Convergence Analysis

In this section, convergence analysis of the algorithm (DFTTS) is presented under the following assumptions on the function F.
Let the level set be defined as:

$$\Omega = \{\ x \in \mathbb{R}^n |\ \|F(x)\| \leq \|F(x_0)\|\}, \tag{3.1}$$

which is bounded, that is, there exists a constant $K > 0$, such that,

$$\|x\| \leq K \quad \forall x \in \Omega. \tag{3.2}$$

Assumption 1:
(1) There exists $x^* \in \mathbb{R}^n$, such that $F(x^*) = 0$.
(2) F is continuously differentiable in a neighborhood of Ω containing x^*.
(3) In some neighborhood of Ω, F is Lipschitz continuous. That is, there exists a positive constant L, such that:

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Omega. \tag{3.3}$$

(4) The Derivative of $F(x)$, that is the Jacobian, is Symmetric.

However, it follows from the level set and assumption 1 that, there exists a positive constant M, such that,

$$\|F(x)\| \leq M, \quad \forall x \in \Omega. \tag{3.4}$$

Lemma 3.1. *Suppose assumption 1 holds and $\{x_k\}$ is generated by the algorithm (DFTTS), then:*

$$\lim_{k\to\infty} \|\alpha_k d_k\| = \lim_{k\to\infty} \|s_k\| = 0, \tag{3.5}$$

and

$$\lim_{k\to\infty} \|\alpha_k F_k\| = 0. \tag{3.6}$$

Proof. From (2.10) and (2.11), we have for all $k > 0$.

$$\begin{aligned} \omega_2\|\alpha_k d_k\|^2 &\leq \omega_1\|\alpha_k F(x_k)\|^2 + \omega_2\|\alpha_k d_k\|^2, \\ &\leq \|F(x_k)\|^2 - \|F(x_{k+1})\|^2 + \eta_k\|F(x_k)\|^2. \end{aligned} \tag{3.7}$$

By summing up (3.7) upto k^{th} term, we have:

$$\begin{aligned}\omega_2 \sum_{i=0}^{k} \|\alpha_i d_i\|^2 &\leq \sum_{i=0}^{k} \left(\|F(x_i)\|^2 - \|F(x_{i+1})\|^2\right) + \sum_{i=0}^{k} \eta_i \|F(x_i)\|^2, \\ &= \|F(x_0)\|^2 - \|F(x_{k+1})\|^2 + \sum_{i=0}^{k} \eta_i \|F(x_i)\|^2, \\ &\leq \|F(x_0)\|^2 + \|F(x_0)\|^2 \sum_{i=0}^{k} \eta_i, \\ &\leq M^2 + M^2 \sum_{i=0}^{\infty} \eta_i. \end{aligned} \tag{3.8}$$

So from assumption 1 and the fact that $\{\eta_k\}$ satisfies (2.10), then the series $\sum_{i=0}^{\infty} \|\alpha_i d_i\|^2$ is convergent, which implies (3.5). By similar arguments as the above but with $\omega_1 \|\alpha_k F(x_k)\|^2$ on the left hand side, we obtain (3.6). ∎

Next, the following result shows that the proposed DFTTS-method is globally convergent.

Theorem 3.2. *Suppose that assumption 1 holds, the sequence $\{x_k\}$ generated by the algorithm (DFTTS) converges globally. That is:*

$$\liminf_{k\to\infty} \|F_k\| = 0. \tag{3.9}$$

Proof. We prove by contradiction, that is, suppose that (3.9) is not true and there exists a positive constant τ and τ^0 such that

$$\|F_k\| \geq \tau, \tag{3.10}$$

and

$$\|F_k^0\| \geq \tau^0. \tag{3.11}$$

We divide the prove into two (2) parts:
Case (i): Consider $\limsup_{k\to\infty} \alpha_k > 0$. Then from (3.6) we have (3.9), with Lemma 4.1, shows that $\lim_{k\to\infty} \|F_k\| = 0$, which contradicts (3.10).
Case (ii): Consider $\limsup_{k\to\infty} \alpha_k = 0$. Since $\alpha_k \geq 0$, this case implies that:

$$\lim_{k\to\infty} \alpha_k = 0. \tag{3.12}$$

The inequalities (2.10), (2.11) and (3.11) shows that there exists a constant τ such that (3.10) hold for all $k \geq 0$ and:

$$\|F_k - F_k^0\| \leq L M_1^2 \alpha_{k-1}, \tag{3.13}$$

(i.e. from (3.10) and (3.11)). Then, it follows that:

$$\|y_k\| = \|F_{k+1} - F_k\| \leq \|F_{k+1} - F_{k+1}^0\| + \|F_k^0 - F_k\| + \|F_{k+1}^0 - F_k^0\|, \tag{3.14}$$

which gives:

$$\|F_{k+1} - F_k\| \leq LM_1^2\alpha_{k-1} + LM_1^2\alpha_{k-2} + L_1\|s_k\|. \tag{3.15}$$

Therefore (3.14) becomes:

$$\|y_k\| \leq h_1, \tag{3.16}$$

from (3.6), (3.12) and (3.16),

$$\lim_{k\to\infty} \|y_k\| = 0. \tag{3.17}$$

Next for $\theta_k = \frac{s_k^T s_k}{s_k^T y_k}$, we have:

$$|\theta_k| = |\frac{s_k^T s_k}{s_k^T y_k}| \leq \frac{\|s_k\|\|s_k\|}{\|s_k\|\|y_k\|} \longrightarrow 0, \tag{3.18}$$

as $k \longrightarrow \infty$, $\|s_k\| \longrightarrow 0$ (from (3.5)). It follows that:

$$|\theta_k| \longrightarrow 0. \tag{3.19}$$

That is there exists a constant $\lambda_1 \in (0, 1)$ such that for sufficiently large λ_1,

$$|\theta_k| \leq \lambda_1. \tag{3.20}$$

Next, from the definition of ε_k,

$$\varepsilon_k = \frac{\theta_k s_k^T F_k}{y_k^T s_k}, \tag{3.21}$$

which gives us:

$$|\varepsilon_k| = |\frac{\theta_k s_k^T F_k}{y_k^T s_k}| \leq \frac{|\theta_k|\|s_k\|\|F_k\|}{\|y_k\|\|s_k\|} \longrightarrow 0 \tag{3.22}$$

as $k \longrightarrow \infty$, $|\theta_k| \longrightarrow 0$, (from (3.19)). It follows that:

$$|\varepsilon_k| \longrightarrow 0. \tag{3.23}$$

That is there exists a constant $\lambda_2 \in (0, 1)$ such that for sufficiently large λ_2,

$$|\varepsilon_k| \leq \lambda_2. \tag{3.24}$$

Again, from the definition of β_k;

$$|\beta_k| = |\frac{(\theta_k y_k - s_k)^T F_k}{y_k^T s_k} + \frac{\varepsilon_k y_k^T y_k}{y_k^T s_k}|, \tag{3.25}$$

$$|\beta_k| \leq \frac{(|\theta_k|\|y_k\| - \|s_k\|)\|F_k\|}{\|y_k\|\|s_k\|} + \frac{|\varepsilon_k|\|y_k\|\|y_k\|}{\|y_k\|\|s_k\|} \longrightarrow 0, \tag{3.26}$$

which also follows that:

$$|\beta_k| \longrightarrow 0. \tag{3.27}$$

That is there exists a constant $\lambda_3 \in (0, 1)$ such that for sufficiently large λ_3,

$$|\beta_k| \leq \lambda_3. \tag{3.28}$$

Without loss of generality, we assume that the above inequalities holds ∀ k≥ 0. Now from the proposed direction (2.1), using Cauchy-Swartz inequality, we have:

$$\|d_k\| \leq \|\theta_k F_k\| + \|\beta_k s_k\| + \|\varepsilon_k y_k\|, \tag{3.29}$$

$$\|d_k\| \leq |\theta_k| \|F_k\| + |\beta_k| \|s_k\| + |\varepsilon_k| \|y_k\|, \tag{3.30}$$

$$\|d_k\| \leq \lambda_1 M + \lambda_3 \|s_k\| + \lambda_2 \|y_k\| \longrightarrow 0. \tag{3.31}$$

which shows that d_k is bounded. Since $\lim_{k\to\infty} \alpha_k = 0$, then $\alpha'_k = \frac{\alpha_k}{r}$ does'nt satisfy (2.11), that is:

$$f(x_k + \alpha_k d_k) - f(x_k) > -\omega_1 \|\alpha'_k F(x_k)\|^2 - \omega_2 \|\alpha'_k d_k\|^2 + \eta_k f(x_k), \tag{3.32}$$

which implies:

$$\frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha'_k} > -\omega_1 \|\alpha'_k F(x_k)\|^2 - \omega_2 \|\alpha'_k d_k\|^2. \tag{3.33}$$

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

$$\frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha'_k} = f'(x_k + \delta_k \alpha'_k d_k)^T d_k. \tag{3.34}$$

Ortega and Rheinboldt [33] presented an approximation to the gradient F'_k in order to avoid computing exact gradient as:

$$F_k = \frac{f(x_k + \alpha_k d_k) - f(x_k)}{\alpha_k}. \tag{3.35}$$

Since $\{x_k\} \subset \Omega$ is bounded, without loss of generality, we assume that $x_k \to x*$. From (3.35) and (2.1), we obtain:

$$\begin{aligned} \lim_{k\to\infty} d_k &= -\lim_{k\to\infty} \theta_k F_k + \lim_{k\to\infty} \beta_k s_k - \lim_{k\to\infty} \varepsilon_k y_k. \\ &\leq -\lim_{k\to\infty} F_k + \lim_{k\to\infty} \beta_k s_k - \lim_{k\to\infty} \varepsilon_k y_k = -F'(x*). \end{aligned} \tag{3.36}$$

That is using (2.1), (3.23) and (3.27) and the fact that the sequence $\{d_k\}$ is bounded. On the other hand, we have:

$$\lim_{k\to\infty} f'(x_k + \delta_k \alpha'_k d_k) = f'(x*). \tag{3.37}$$

Therefore, from (3.33)-(3.37), it follows that $-f'(x*)^T f'(x*) \geqslant 0$. That is $\|F(x*)\| = 0$. Hence contradiction with (3.11). Which completes the proof. ■

4. Numerical Results

In this section, we present the numerical performance of our method for solving (1.1) which is compared with simple three-term conjugate gradient (STTCG) method in [15] and derivative free conjugate gradient method via Broyden's update (DFCGB) in [34]. For unbiasness, we set $\omega_1 = 10^{-4}$, $\omega_2 = 10^{-4}$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$ for all the three methods.

Th codes were written in MATLAB R2014a 7.71GB and run on a personal computer with Windows 10pro, intel(R) core(TM)i3-3217U 1.8 GHz CPU processor and 4GB RAM memory. The iteration stopped if the total number of iterations exceeds 1000 or $\|F(x_k)\| \leq 10^{-4}$.The three methods were tested using ten test problems with different initial points and dimensions (n values).

Problem 1:[35]

$$F_i(x) = x_i^2 - 4$$
$$i = 1, 2, 3, ..., n.$$
$$x_0 = (0.01, 0.01, ..., 0.01)^T.$$

Problem 2:[36]

$$F_1(x) = x_1(x_1^2 + x_2^2) - 1,$$
$$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2),$$
$$F_n(x) = x_n(x_{n-1}^2 + x_n^2).$$
$$i = 2, 3, ..., n - 1.$$
$$x_0 = (0.8, 0.8, ..., 0.8)^T.$$

Problem 3:[36]

$$F_{3i-2}(x) = x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1,$$
$$F_{3i-1}(x) = x_{3i-2}x_{3i-2}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,$$
$$F_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}.$$
$$i = 1, ..., \frac{n}{3}.$$
$$x_0 = (0.07, 0.07, ..., 0.07)^T.$$

Problem 4:[36]

$$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2.$$
$$i = 1, 2, ..., n.$$
$$x_0 = (0.7, 0.7, ..., 0.7)^T.$$

Problem 5:[36]

$$F_i(x) = x_i - 0.1x_{i+1}^2,$$
$$F_n(x) = x_n - 0.1x_1^2.$$
$$i = 1, 2, ..., n - 1.$$
$$x_0 = (0.03, 0.03, ..., 0.03)^T.$$

Problem 6:[35]

$$F_i(x) = e^{x_i} - 1.$$
$$i = 1, 2, ..., n.$$
$$x_0 = (1.0, 1.0, ..., 1.0)^T.$$

Problem 7:[35]

$$F_i(x) = x_i^2 + x_i - 2$$
$$i = 1, 2, ..., n.$$
$$x_0 = (-0.05, -0.05, ..., -0.05)^T.$$

Problem 8:[37]

$$F_i(x) = x_i - 3x_i(\frac{sin(x_i)}{3} - 0.66) + 2.$$
$$i = 1, 2, ..., n.$$
$$x_0 = (0.2, 0.2, ..., 0.2)^T.$$

Problem 9:[36]

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

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

Problem 10:[36]

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

$$x_0 = (0.009, 0.009, ..., 0.009)^T.$$

TABLE 1. Results of DFTTS, STTCG and DFCGB methods for problems 1 - 5.

			DFTTS			STTCG			DFCGB	
Prob.	Dim	NI	T(s)	$\|F(x)\|$	NI	T(s)	$\|F(x)\|$	NI	T(s)	$\|F(x)\|$
1	100	8	0.002925	4.50E-08	14	0.005277	4.01E-05	9	0.00705	7.68E-05
	1000	8	0.004364	1.49E-05	14	0.007382	6.37E-05	10	0.085515	5.83E-05
	5000	8	0.016145	3.34E-05	15	0.031974	5.70E-05	11	1.592432	3.13E-05
	10000	9	0.031617	1.89E-06	15	0.047541	4.03E-05	11	6.766341	4.42E-05
	100000	9	0.237882	5.99E-06	16	0.430514	5.10E-05	—	—	—
	1000000	10	3.635959	7.57E-07	16	6.056891	8.07E-05	—	—	—
2	100	127	0.043803	9.61E-05	199	0.070788	9.70E-05	49	0.034648	8.91E-05
	1000	79	0.057418	9.58E-05	107	0.079023	9.33E-05	47	0.344276	7.22E-05
	5000	52	0.151704	8.94E-05	64	0.199595	9.93E-05	53	7.787202	9.17E-05
	10000	39	0.174559	9.45E-05	73	0.38685	9.49E-05	52	30.39018	8.91E-05
	100000	94	5.568504	9.44E-05	84	4.365787	9.09E-05	—	—	—
	1000000	47	32.42524	9.97E-05	72	51.44887	9.18E-05	—	—	—
3	100	50	0.021764	6.63E-05	147	0.088916	8.08E-05	16	0.016298	5.25E-05
	1000	77	0.077419	9.80E-05	143	0.178363	9.73E-05	17	0.133175	8.75E-05
	5000	58	0.175897	9.57E-05	153	0.831172	8.44E-05	19	2.806844	4.93E-05
	10000	70	0.390715	5.29E-05	148	1.147393	9.98E-05	19	11.455431	6.97E-05
	100000	69	3.890869	5.08E-05	145	10.87078	9.91E-05	—	—	—
	1000000	73	50.24391	7.70E-05	161	148.0232	9.58E-05	—	—	—
4	100	7	0.004652	2.68E-06	12	0.007203	8.93E-05	9	0.021984	2.81E-05
	1000	8	0.007606	7.31E-08	13	0.009938	5.86E-05	9	0.06871	8.89E-05
	5000	8	0.03458	1.63E-07	14	0.050523	5.24E-05	10	1.501034	4.77E-05
	10000	8	0.046677	1.46E-05	13	0.072794	9.34E-05	10	6.261546	6.75E-05
	100000	8	0.352478	4.62E-05	15	0.796191	4.72E-05	—	—	—
	1000000	9	6.020842	5.87E-06	15	10.42624	7.48E-05	—	—	—
5	100	2	0.002606	2.71E-06	4	0.002851	5.73E-05	2	0.009846	8.14E-06
	1000	2	0.002438	2.56E-08	2	0.002534	2.56E-08	2	0.018175	2.57E-05
	5000	2	0.013633	5.73E-08	2	0.006859	5.73E-08	2	0.289538	5.76E-05
	10000	2	0.009631	8.10E-08	2	0.009819	8.10E-08	2	1.153537	8.14E-05
	100000	2	0.064651	2.56E-07	2	0.073813	2.56E-07	—	—	—
	1000000	2	0.766187	8.10E-07	2	0.892765	8.10E-07	—	—	—

TABLE 2. Results of DFTTS, STTCG and DFCGB methods for problems 6 - 10.

		DFTTS			STTCG			DFCGB		
Prob.	Dim	NI	T(s)	$\|F(x)\|$	NI	T(s)	$\|F(x)\|$	NI	T(s)	$\|F(x)\|$
6	100	5	0.004417	1.94E-05	8	0.002743	4.91E-05	7	0.010756	1.36E-05
	1000	5	0.004085	6.01E-07	5	0.004036	6.01E-07	7	0.057597	4.29E-05
	5000	5	0.017718	1.34E-06	5	0.061304	1.34E-06	7	1.019867	9.59E-05
	10000	5	0.018356	1.90E-06	5	0.018386	1.90E-06	8	4.88343	6.04E-08
	100000	5	0.137455	6.01E-06	5	0.140794	6.01E-06	—	—	—
	1000000	5	1.798991	1.90E-05	5	1.788291	1.90E-05	—	—	—
7	100	9	0.003056	8.17E-07	15	0.0265	8.97E-05	12	0.01136	5.32E-05
	1000	10	0.007421	1.66E-07	17	0.012741	4.58E-05	13	0.096964	5.39E-05
	5000	10	0.358979	9.67E-05	18	0.078574	4.11E-05	14	2.009807	3.86E-05
	10000	11	0.048413	2.19E-05	18	0.084278	5.82E-05	14	8.447479	5.45E-05
	100000	12	0.457668	1.11E-05	19	0.735334	7.36E-05	—	—	—
	1000000	13	6.628422	5.61E-06	20	10.34492	9.31E-05	—	—	—
8	100	6	0.002947	9.50E-06	11	0.004356	8.60E-05	9	0.019582	3.27E-05
	1000	6	0.00471	3.00E-05	13	0.01004	4.35E-05	10	0.073247	2.46E-05
	5000	7	0.01916	2.70E-06	13	0.036364	4.88E-05	10	1.442949	5.50E-05
	10000	7	0.033885	3.82E-06	13	0.062909	6.90E-05	10	6.104019	7.78E-05
	100000	8	0.337238	4.86E-07	14	0.60931	4.38E-05	—	—	—
	1000000	8	4.653664	1.54E-06	15	9.00762	5.54E-05	—	—	—
9	100	19	0.076075	4.80E-05	38	0.172302	8.35E-05	33	0.662208	9.53E-05
	1000	21	0.450805	6.26E-05	41	1.05524	9.35E-05	31	1.25946	9.42E-05
	5000	23	7.12335	8.10E-05	44	15.62872	8.96E-05	31	18.533882	8.94E-05
	10000	17	19.0201	8.34E-05	35	46.69017	9.76E-05	81	202.69164	9.94E-05
	100000	—	—	—	—	—	—	—	—	—
	1000000	—	—	—	—	—	—	—	—	—
10	100	34	0.626957	9.28E-05	54	0.260427	9.70E-05	24	0.243119	8.61E-05
	1000	39	0.955638	7.54E-05	59	1.485643	9.33E-05	28	1.213629	9.87E-05
	5000	38	12.69361	8.22E-05	59	20.52936	8.58E-05	23	14.639339	9.82E-05
	10000	40	50.14935	8.53E-05	55	72.14628	9.84E-05	24	59.088595	9.07E-05
	100000	—	—	—	—	—	—	—	—	—
	1000000	—	—	—	—	—	—	—	—	—

TABLE 3. Summary of results from Table 1 and 2 for DFTTS, STTCG and DFCGB methods.

Method	NI	Percentage	CPUTime	Percentage
DFTTS	34	56.67%	47	78.33%
STTCG	1	1.67%	6	10.00%
DFCGB	10	16.67%	3	5.00%
Undecided	15	25.00%	4	6.67%
Total	60	100.00%	60	100.00%

The numerical results of the three methods are presented in Tables 1 and 2, where NI and T stands 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 systems of nonlinear equations in (1.1).

In Table 3, the summary of the reported numerical results from Table 1 and 2 are presented in order to show which, among the three methods, is a winner with respect to number of iterations and CPU time. The summary indicates that DFTTS method is more effective interms of number of iterations as it solves 56.67% (34 out of 60) of all the problems with least number of iterations compared to STTCG which solves just 1.67%(1 out of 60) and DFCGB which solves 16.67% (10 out of 60). For the undecided, it indicates that 25.00% (15 out of 60)of the problems were either solved by two or all the three methods with thesame number of iterations, or failed by all the three methods concurrently.

Similarly, from Table 3, it indicates that DFTTS method outperforms the other methods with respect to CPU time as it solves 78.33% (47 out of 60) of all the problems with least CPU time than STTCG and DFCGB algorithms which solved just 10.00% (6 out of 60) and 5.00% (3 out of 60) respectively. For the undecided under CPU time, it indicates that 6.67% (4 out of 60)of the problems failed for all the three algorithms concurrently.

Furthermore, on average, our $\|F(x_k)\|$ is too small compared to the other method, which signifies that the solution obtained is a good approximation to the exact solution compared with the remaining two methods.

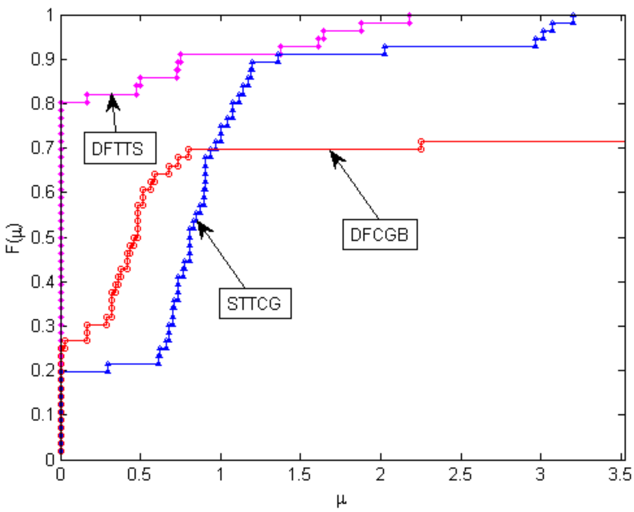

FIGURE 1. Performance profile of DFTTS, STTCG and DFCGB methods with respect to the number of iteration for problem 1-10.

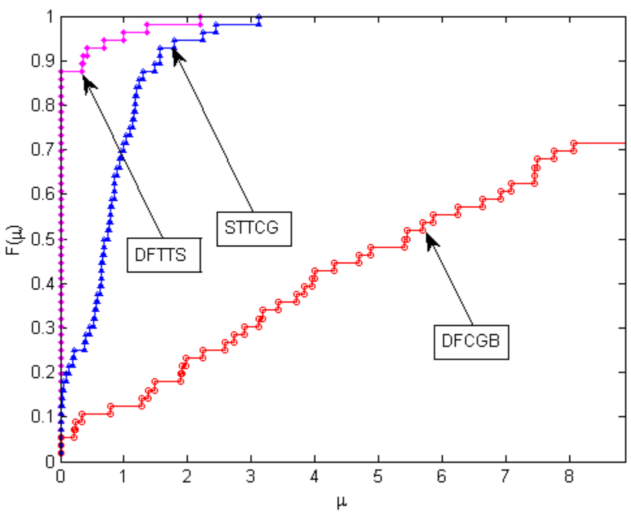

FIGURE 2. Performance profile of DFTTS, STTCG and DFCGB methods with respect to the CPU time (in seconds) for problems 1-10.

Figures (1-2), shows the performance of our method relative to the number of iterations and CPU time which were evaluated using the profiles of Dolan and Moré [38] which is a tool for evaluating and comparing the performance of iterative methods. The profile of each method is measured according to the ratio of its computational outcome, 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, which is the most effective, stand for our proposed DFTTS method and the bottom curves stand for STTCG and DFCGB methods.

5. Conclusion

In this paper, we present a derivative-free three-term spectral conjugate gradient (DFTTS) method for solving synnetric nonlinear equations and its performance was compared with that of simple three-term conjugate gradient(STTCG) method [36] and derivative free conjugate gradient method via Broydens update (DFCGB) [34] by conducting some numerical experiments. We, however, proved the convergence of our proposed method using a derivative-free type line search proposed in [2]. The numerical results show that our proposed method is more efficient interms of accuracy and robustness and hence promising.

Future Research

This research will be applied to the experiments on the l_1-norm regularization problems in compressive sensing.

Acknowledgements

We would like to thank the referees for their comments and suggestions on the manuscript.

References

[1] E.G. Birgin, J.M. Martinez, A spectral conjugate gradient method for unconstrained optimization, Appl. Math. Optim. 43 (2) (2001) 117–128.

[2] D. Li, M. Fukushima, A global and superlinear convergent Gauss-Newton base BFGS method for symmetric nonlinear equation, SIAM J. Numer. Anal. 37 (1) (2000) 152–172.

[3] A.S. Halilu, M.Y. Waziri, A transformed double steplength method for solving large-scale system of nonlinear equations, J. Numer. Math. Soch. 9 (1) (2017) 20–32.

[4] M.Y. Waziri, Jacobian computation-free Newton's method for systems of nonlinear equations, J. Numer. Math. Soch. 2 (2010) 54–63.

[5] M.Y. Waziri, W.J. Leong, M.A. Hassan, M. Monsi, A new Newton's method with diagonal Jacobian approximation for system of nonlinear equations, Journal of mathematics, Statistics 6 (3) (2010) 246–252.

[6] L. Muhammad, M.Y. Waziri, An alternative three-term conjugate gradient algorithm for systems of nonlinear equations, Int. J. Math. Model. Comput. 7 (2) (2017) 145–157.

[7] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–436.

[8] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, the computer Journal 7 (1964) 149–154.

[9] B.T. Polyak, The conjugate gradient method in extreme problems, USSR Computational Mathematics, Mathematical Physics 9 (1969) 94–112.

[10] E. Polak, G. Ribiere, Note sur la convergence de directions conjuge'es, ESAIM: Math. Model. Numer. Anal. 3E (1969) 35–43.

[11] R.R. Fletcher, Practical Method of Optimization, 2nd ed., New York, 2000.

[12] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms, Part I: Theory, J. Optim. Theory Appl. 69 (1991) 129–137.

[13] Y.H. Dai, Y.X. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optim. 10 (1999) 177–182.

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

[15] M.Y. Waziri, L. Muhammad, J. Sabiu, A simple three-term conjugate gradient algorithm for solving symmetric systems of Nonlinear Equations, Int. J. Advances Appl. Sci. (2017) 306914657.

[16] G. Yuan, M. Zhang, A three-terms Polak-Ribiere-Polyak conjugate gradient algorithm for large-scale nonlinear equations, J. Comput. Appl. Math. 286 (2015) 186–195.

[17] M.Y. Waziri, L. Muhammad, An accelerated three-term conjugate gradient algorithm for solving large-scale systems of nonlinear equations, Sohag J. Math. 4 (2) (2017) 1–8.

[18] E.L. Loannis, T. Vassilis, P. Panagiotis, A descent hybrid conjugate gradient method based on memory-less BFGS update, Numer. Algorithms 67 (2018) 31–48.

[19] W. Cheng, Z. Chen, Nonmonotone spectral method for large-scale symmetric nonlinear equations, Numer. Algorithms 62 (1) (2013) 149–162.

[20] D.H. Li, X.L. Wang, A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations, Numer. Algebra Control. Optim. 1 (1) (2011) 71–82.

[21] L. Zhang, W. Zhou, D.H. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, Numerische Mathematik 104 (4) (2006) 561–572.

[22] L. Zhang, W. Zhou, D.H. Li, A descent modified Polak-Ribiere-Polyak conjugate gradient method, its global convergence, IMA J. Numer. Anal. 26 (4) (2006) 629–640.

[23] W. Zhou, D. Shen, Convergence properties of an iterative method for solving symmetric non-linear equations, J. Optim. Theory Appl. 164 (1) (2015) 277–289.

[24] N. Andrei, A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, Appl. Math. Lett. 20 (6) (2007) 645–650.

[25] N. Andrei, Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, Optim. Methods Softw. 22 (4) (2007) 561–571.

[26] M.Y. Waziri, J. Sabi'u, A derivative-free conjugate gradient method, its global convergence for solving symmetric nonlinear equations, Int. J. Math. Sci. 8 (2015) 961487.

[27] A.M. Awwal, P. Kumam, A.B. Abubakar, A. Wakili, A projection Hestenes-Stiefel-like method for monotone nonlinear equations with convex constraints, Thai J. Math. 18 (1) (2019) 181–199.

[28] A.B. Abubakar, P. Kumam, A descent Dai-Liao conjugate gradient method for nonlinear equations, Numer. Algorithms 81 (1) (2018) 197–210.

[29] A.H. Ibrahim, G.A. Isa, H. Usman, J. Abubakar, A.B. Abubakar, Derivative-Free RMIL conjugate gradient algorithm for convex constrained equations, Thai J. Math. 18 (1) (2020) 211–231.

[30] A.M. Awwal, P. Kumam, A.B. Abubakar, Spectral modified Polak-Ribiere-Polyak projection conjugate gradient method for solving monotone systems of nonlinear equations, Appl. Math. Comput. 362 (2019) 124514.

[31] A.B. Abubakar, P. Kumam, A.M. Awwal, A descent Dai-Liao projection method for convex constrained nonlinear monotone equations with applications, Thai J. Math. 18 (1) (2019) 128–152.

[32] M. Raydan, The Barzilai, Borwein gradient method for large scale unconstrained minimization problem, SIAM J. Optim. 7 (1) (1997) 26–33.

[33] J.M. Ortega, W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, USA, 1970.

[34] M.K. Dauda, Derivative free conjugate gradient method via Broyden's update for solving symmetric systems of nonlinear equations, Journal of Physics: Conference Series 1366 (2019) 012099.

[35] W. La Cruz, J.M. Martinez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, Math. Comput. 6 (2017) 76–79.

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

[37] S. Jamilu, M.Y. Waziri, I. Abba, A new hybrid Dai-Yuan and Hestenes-Stiefel conjugate gradient parameter for solving system of nonlinear equations, MAYFEB Journal of Mathematics 1 (2017) 44–55.

[38] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2) (2002) 201–213.