A Simple Three-term Conjugate Gradient Algorithm for Solving Symmetric Systems of Nonlinear Equations

M. Y. Waziri\textsuperscript{1}, L. Muhammad\textsuperscript{2}, J. Sabi’u\textsuperscript{3}

\textsuperscript{1}Department of Mathematical Sciences, Faculty of Science, Bayero University Kano, Nigeria
\textsuperscript{2}\textsuperscript{3}Department of Mathematics, Faculty of Science, Northwest University, Nigeria

ABSTRACT

This paper presents a simple three-terms Conjugate Gradient algorithm for solving Large-Scale systems of nonlinear equations without computing Jacobian and gradient via the special structure of the underlying function. This three term CG of the proposed method has an advantage of solving relatively large-scale problems, with lower storage requirement compared to some existing methods. By incorporating the Powel restart approach in to the algorithm, we prove the global convergence of the proposed method with a derivative free line search under suitable assumptions. The numerical results are presented which show that the proposed method is promising. Mathematics Subject Classification: 65H11, 65K05, 65H12, 65H18.

1. INTRODUCTION

In real life problems, many problems are in large-scale systems of nonlinear equations such as concentration of chemical species, cross-sectional properties of structural elements and dimensional mechanical linkages e.t.c. Hence it is extremely important to develope an efficient algorithm to solve the following basic large-scale problem

\[ F(x) = 0, \]  

where \( F: \mathbb{R}^n \rightarrow \mathbb{R}^n \) is continuously differentiable, and the Jacobian \( J(x) = F'(x) \) is symmetric, that is \( J(x) = J(x)^T \).

Let define a norm function by \( f(x) = \frac{1}{2} ||F(x)||^2 \), where \( ||.|| \) is the Euclidean norm. Then (1) is equivalent to the following unconstrained optimization problem

\[ \min f(x), \quad x \in \mathbb{R}^n \] 

The general CG method for solving (2), is given as follows

\[ x_{k+1} = x_k + \alpha_k d_k, \] 

where \( \alpha_k > 0 \) is attained using line search, and direction \( d_k \) are obtained by

\[ d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k, \quad d_0 = -\nabla f(x_0), \] 

where \( \beta_k \) is called conjugate gradient parameter.
It is remarkable to mention that, many algorithms have been developed to solve nonlinear system of equations, the famous one is the Newton and quasi-Newton methods [1] which entails computation of Jacobian matrix or it’s approximation. Other methods include Gauss-Newton methods [2-4], the gradient-based and the conjugate gradient methods [5-9], the trust region method [10-12], the Levenberg-Marquardt methods [13-14], the tensor methods [15], the derivative free methods [16-18] and the subspace methods [19].

One of the most crucial features of each numerical algorithms for solving systems of nonlinear equations is how the procedure deals with large-scale problems. It is well known that choices of βk affect numerical performance of the method, and hence many researchers have studied effective choices of βk (see [5-6, 20], for example). Recently Hager and Zhang [6] presented conjugate gradient methods and their global convergence properties. The shortcoming of conjugate gradient methods is that most of conjugate gradient methods do not satisfy the descent condition \( F(x)^T d_k \leq 0 \). However, some researchers proposed three-term conjugate gradient methods which always generate descent search directions (see [6-9, 19] for example).

This is what motivated us, to proposed a simple three CG algorithm for solving large scale systems of nonlinear equations by a modifying the classical memoryless BFGS approximation of the Jacobian inverse restarted as a multiple of an identity matrix at every step. The method posses low memory requirement, global convergence properties and simple implementation procedure.

The main contribution of this paper is to construct a fast and efficient three-term conjugate gradient method for solving (1) the proposed method is based on the three-term conjugate gradient method proposed by [21] for unconstrained optimization. In other words our algorithm can be thought as an extension to three-term conjugate gradient method to a general systems of nonlinear equations. We present experimental numerical results and performance comparism with three-term DF-SDCG conjugate gradient method by [20] which illustarted that the proposed algorithm is efficient and promising. The rest of the paper is organized as follows: In section 2, we describe the proposed algorithm in details. Subsequently, Convergence results are presented in Section 3. Some numerical results are reported in Section 4 to show its practical performance. Finally, conclusions are made in Section 5.

2. ALGORITHM

This section, presents a simple three term CG method for solving large-scale systems of nonlinear equations via memoryless BFGS update. In general, quasi-Newton method is an iterative method that generates a sequence of points \( x_k \) from a given initial guess \( x_0 \) via the following form:

\[
x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k) \quad k=0, 1, 2, \ldots,
\]

where \( B_k \) is an approximation to the Jacobian which can be updated at each iteration for \( k=0, 1, 2, \ldots \), the updated matrix \( B_{k+1} \) is chosen in such a way that it satisfies the secant equation, i.e

\[
B_{k+1}s_k = y_k,
\]

where \( s_k = x_{k+1} - x_k \) and \( y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \)

Ortega and Rheinboldt in [22] presented approximation to the gradient \( \nabla f(x_k) \), in order to avoid computing exact gradient as

\[
g_k = \frac{F(x_k + \alpha_k F_k) - F_k}{\alpha_k},
\]

In our work we will use their idea and \( ak \) to be updated via line search technique. The update formula for the BFGS \( B_k \) is given as

\[
B_{k+1} = B_k - \frac{s_k y_k^T B_k + B_k y_k s_k^T}{y_k^T s_k} + (1 - \frac{y_k^T B_k y_k}{y_k^T s_k}) \frac{s_k s_k^T}{y_k^T s_k}.
\]

By letting \( B_k = \theta_k I \), (8) can be rewrite as:

\[
Q_{k+1} = \theta_k I - \frac{s_k y_k^T B_k + \theta_k y_k s_k^T}{y_k^T s_k} + (1 - \frac{y_k^T B_k y_k}{y_k^T s_k}) \frac{s_k s_k^T}{y_k^T s_k},
\]

where, \( \theta_k \) as in Raydan [23]
\[ \theta_k = \frac{s_k^T s_k}{s_k^T y_k} \]  
(10)

We further multiply both sides of (9) by \( g(x_{k+1}) \) to obtain

\[ Q_{k+1} g(x_{k+1}) = \theta_k g(x_{k+1}) - \left( \frac{s_k^T \theta_k - \theta_k s_k^T}{y_k^T s_k} \right) g(x_{k+1}) \left( 1 + \frac{y_k^T \theta_k s_k}{y_k^T s_k} \right) s_k^T y_k g(x_{k+1}) \]  
(11)

Observe that the direction \( d_{k+1} \) from (11) can be written as

\[ d_{k+1} = -Q_{k+1} g(x_{k+1}) \]  
(12)

Hence, our new direction is

\[ d_{k+1} = -\theta_k g_{k+1} - \delta_k s_k - \eta_k y_k \]  
(13)

where,

\[ \delta_k = \left( 1 + \frac{y_k^T \theta_k y_k}{y_k^T s_k} \frac{s_k^T g_{k+1}}{y_k^T s_k} \right) - \frac{y_k^T \theta_k g_{k+1}}{y_k^T s_k}, \]  
(14)

\[ \eta_k = \frac{\theta_k s_k^2}{y_k^T s_k}, \]  
(15)

Finally, we have

\[ x_{k+1} = x_k + \alpha_k d_k \]  
(16)

Therefore with the proposed search direction we are using the derivative free line search of Li and Li [16] to find \( \alpha_k = \max\{s, \rho s, \rho^2 s, \ldots\} \) such that

\[ -g(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k ||g(x_k + \alpha_k d_k)|| ||d_k||^2 \]  
(17)

where \( \sigma, s > 0 \) and \( \rho \in (0, 1) \).

We present the below algorithm

Algorithm 2.1 (STTCG)

Step 1: Given \( x_0, \alpha > 0, \sigma \in (0, 1), \epsilon = 10^{-4} \) and compute \( d_0 = -g_0 \), set \( k = 0 \).

Step 2: If \( ||g_k|| < \epsilon \), then stop; otherwise continue with Step 3.

Step 3: Determine the stepsize \( \alpha_k \) by using a line search conditions in (17).

Step 4: Determine \( \delta_k \) and \( \eta_k \) by (14) and (15) respectively.

Step 5: Find the search direction by (13).

Step 6: Powel restart criterion. If \( ||g_{k+1}^T g_{k+1}|| > 0.2 ||g_{k+1}||^2 \), then set \( d_{k+1} = -g_{k+1} \).

Step 7: Consider \( k = k + 1 \) and go to step 2.

3. CONVERGENCE RESULT

In this Section, we will present the global convergence of the simple three terms conjugate gradient method.

Definition 1

Let \( \Omega \) be the level set defined by

\[ \Omega = \{ x : f(x) \leq \tau f(x_0) \} \]

where \( \tau \) is a positive constant.

The following Assumptions are needed on the nonlinear systems \( F \) in order to establish the global convergence of our method Assumption A.
A Simple Three-term Conjugate Gradient Algorithm for Solving Symmetric Systems of … (M. Y. Waziri)

(i) The level set is bounded.

\[ \Omega = \{ x \mid f(x) \leq \tau f(x_0) \} \]

(ii) In some neighborhood \( N \) of \( \Omega \), the Jacobian is Lipschitz continuous, i.e., there exists a constant \( L > 0 \) s.t. for all \( x, y \in N \)

\[ \| F'(x) - F'(y) \| \leq L \| x - y \| \]  

(iii) There exists \( x^* \in \Omega \) such that \( F(x^*) = 0 \) and \( F'(x) \) is continuous for all \( x \).

Assumption \( A(ii) \) and \( A(iii) \) implies that there exist positive constants \( \kappa_1, \kappa_2 \) and \( L_1 \) such that

\[ \| J(x) \| \leq \kappa_1, \quad \forall x \in N \]
\[ \| \nabla f(x) - \nabla f(y) \| \leq L_1 \| x - y \|, \]
\[ \| J(x) \| \leq 2\kappa_2, \quad \forall x, y \in N. \]

The following lemma shows that the direction \( d_k \) determined by (13) is interesting

**Lemma 1** Suppose that \( F \) is uniformly convex then \( d_k \) is defined by (13), then we have

\[ g_k^T d_k = -\| g_k \|^2 \]  

and

\[ \| d_k \| \leq (\lambda + \frac{\lambda}{\nu} (2 + L + \frac{L^2}{\nu})) \]

**Proof.**

when \( k = 0 \) (21) and (22) hold since \( d_0 = -g_0 \). From the definition of \( d_k \) in (13) we have

\[ d_k^T g(x_{k+1}) = -\| g(x_{k+1}) \|^2 + \left[ 1 + \frac{y_k^T \theta_k g_k}{y_k^T s_k} \right] \left[ \frac{y_k^T g_{k+1}}{y_k^T s_k} - \frac{y_k^T \theta_k g_{k+1}}{y_k^T s_k} - \frac{\theta_k s_k^T g_{k+1}}{y_k^T s_k} \right] g(x_{k+1}) \]

Thus (21) hold for all \( k \geq 1 \) and By Lipchitz continuity, we know that \( \| y_k \| \leq L\| s_k \| \). On the other hand by uniform convexity, it yields

\[ g_k^T s_k \geq \frac{\nu}{2} \| s_k \|^2. \]

Thus,

\[ |\delta_k| \leq \frac{|y_k^T g_{k+1}|}{|y_k^T s_k|} + \frac{|\theta_k||y_k||^{2}| s_k^T g_{k+1}|}{|y_k^T s_k|^2} + \frac{|\theta_k||y_k^T g_{k+1}|}{|y_k^T s_k|} \]

\[ \leq \frac{\lambda}{\nu} \| s_k \| + \frac{\frac{L^2}{\nu}}{\nu^2} \| s_k \| + \frac{L\lambda}{\nu} \| s_k \| \]

\[ = \frac{\lambda}{\nu}(1 + L + \frac{L^2}{\nu}) \frac{1}{\| s_k \|}. \]

Since

\[ |\eta_k| \leq \frac{|\theta_k||y_k^T g_{k+1}|}{|y_k^T s_k|} \leq \frac{\| s_k \| \| g_{k+1} \|}{\nu \| s_k \| ^2} \leq \frac{\lambda}{\nu} \frac{1}{\nu \| s_k \|^2}. \]
Finally
\[ ||d_{k+1}|| \leq ||\delta_k|| ||g_{k+1}|| + ||\delta_k|| ||s_k|| + ||\eta_k|| ||y_k|| \leq \lambda + \frac{\lambda}{\nu} (2 + L + \frac{L^2}{\nu}) \] \hspace{1cm} (28)

Hence, \( d_{k+1} \) is bounded.

The coming lemma shows that the line search in step 3 of STTCG Algorithm is reasonable, then the presented algorithm is well defined.

**Lemma 2**

Let the Assumption A hold. Then STTCG Algorithm produces an iterate of \( z_k = x_k + \alpha d_k \) in a finite number of backtracking steps.

Proof:

We suppose that \( ||g_k|| \to 0 \) does not hold, or the algorithm is stoped. Then there exists a constant \( \epsilon_0 > 0 \) such that
\[ ||g_k|| \geq \epsilon_0, \forall k \in \mathbb{N} \bigcup \{0\} \] \hspace{1cm} (29)

We will get this by contradiction. Suppose that for some iterate indexes such as \( k \), the condition (17) is not true. Then by letting \( a_{k+m}^m = p^{m} s \), it can be concluded that
\[ -g(x_k, \alpha_k^m d_k)^T d_k < \sigma \alpha_k^m ||g(x_k + \alpha_k^m d_k)|| ||d_k||^2, \quad \forall m \in \mathbb{N} \bigcup \{0\}. \]

combining with assumption A (ii) and (21), we have
\[ ||g(x_k)||^2 = -d_k^T g_k = [g(x_k + \alpha_k^m d_k) - g(x_k)]^T d_k - g(x_k + \alpha_k^m d_k)^T d_k < [L + \sigma] ||g(x_k + \alpha_k^m d_k)|| ||d_k||^2, \quad \forall m \in \mathbb{N} \bigcup \{0\}. \] \hspace{1cm} (30)

By (19) and (28)
\[ ||g(x_k + \alpha_k^m d_k)|| \leq ||g(x_k + \alpha_k^m d_k) - g_k|| + ||g_k|| \leq L\alpha_k^m ||d_k|| \]
\[ \leq Ls(\lambda + \frac{\lambda}{\nu} (2 + L + \frac{L^2}{\nu})) \]

Thus, we obtain
\[ \alpha_k^m > \frac{||g_k||^2}{[L + \sigma] ||g(x_k + \alpha_k^m d_k)|| ||d_k||^2} \geq \frac{\epsilon_0^2 ||d_k||^2}{L + Ls(\lambda + \frac{\lambda}{\nu} (2 + L + \frac{L^2}{\nu}))} > 0, \forall m \in \mathbb{N} \bigcup \{0\} \]

Thus, it contradicts with the defination of \( \alpha_k^m \). Consequently, the line search procedure (17) can attain a positive steplength \( \alpha_k \) in a finite number of backtracking steps. Hence it turns out the result of this lemma. The proof is complete.

Now we establish the global convergence theorem

**Theorem** Let the properties of assumption A hold. Then the sequence \( \{x_k\} \) be generated by STTCG algorithm converges globally, that is,
\[ \liminf_{k \to \infty} ||\nabla f(x_k)|| = 0. \] \hspace{1cm} (31)

Proof. We prove this theorem by contradiction. Suppose that (31) is not true, then there exists a positive constant \( \tau \) such that
\[ ||\nabla f(x_k)|| \geq \tau, \quad \forall k \geq 0. \] \hspace{1cm} (32)

Since \( \nabla f(x_k) = J_k F_k \), (32) implies that there exists a positive constant \( \tau_1 \) satisfying
A Simple Three-term Conjugate Gradient Algorithm for Solving Symmetric Systems of ... (M. Y. Waziri)

\[ \|F_k\| \geq \tau_1, \quad \forall k \geq 0. \quad (33) \]

Case (i): \( \limsup_{k \to \infty} \alpha_k > 0 \). Then by (22), we have \( \liminf_{k \to \infty} \|F_k\| = 0 \). This and Lemma 1 show that \( \lim_{k \to \infty} \|F_k\| = 0 \), which contradicts (32).

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

\[ \lim_{k \to \infty} \alpha_k = 0. \quad (34) \]

by definition of \( g_k \) in (7), we have

\[ ||g_k - \nabla f(x_k)|| = \left\| \frac{F(x_k + \alpha_{k-1}F_k) - F_k}{\alpha_{k-1}} - J_k^T F_k \right\| \quad (35) \]

\[ = \left\| \int_0^1 J(x_k + t\alpha_{k-1}F_k) - J_k)dtF_k \right\| \quad (36) \]

\[ \leq LM_1^2 \alpha_{k-1}, \quad (37) \]

where we use (19) and (20) in the last inequality. (17) and (32) show that there exists a constant \( \tau_2 > 0 \) such that

\[ ||g_k|| \geq \tau_2, \quad \forall k \geq 0. \quad (38) \]

By (7) and (19), we get

\[ ||g_k|| = \int_0^1 J(x_k + t\alpha_{k-1}F_k)F_k dt \leq M_1M_2, \quad \forall k \geq 0. \quad (39) \]

From (20) and (39), we obtain

\[ ||y_k|| = ||g_k - g_{k+1}|| \leq ||g_k - \nabla f(x_k)|| + ||g_{k-1} - \nabla f(x_{k-1})|| + ||\nabla f(x_k) - \nabla f(x_{k-1})|| \]

\[ \leq L M_1^2 (\alpha_{k-1} + \alpha_{k-2}) + L_1 ||s_{k-1}|| \quad (40) \]

This together with (34) and Lemma 2 show that \( \lim_{k \to \infty} ||y_k|| = 0 \). From (38), (39), (40) and (41), we have

\[ ||y_{k+1}|| \leq \frac{||s_k||}{||s_k||} ||x_k|| \quad (43) \]

meaning there exists a constant \( \lambda \in (0, 1) \) such that for sufficiently large \( k \)

\[ |
\theta_{k+1} | \leq \lambda. \quad (44) \]

Since \( \lim_{k \to \infty} \alpha_k = 0 \), then \( \alpha'_k = \frac{\alpha_k}{r} \) does not satisfy (17) namely,

\[-g(x_k + \alpha'_kd_k)^T d_k > \alpha_k ||g(x_k + \alpha'_kd_k)|| ||d_k||^2 \]

Since \( \{x_k\} \subset \Omega \) is bounded and (28), without loss of generality, we assume \( x_k \to x^* \).

By (7), we have
\[
\lim_{k \to \infty} d_{k+1} = -\lim_{k \to \infty} \theta_k q_{k+1} + \lim_{k \to \infty} \delta_k s_k + \lim_{k \to \infty} \eta_k y_k
\]
\[
\leq - \lim_{k \to \infty} q_{k+1} + \lim_{k \to \infty} \delta_k s_k + \lim_{k \to \infty} \eta_k y_k
\]
\[
= -\nabla f(x^*)
\]  

(45)

the fact that the sequence \( \{d_k\} \) is bounded. on the other hand

\[
\lim_{k \to \infty} ||g(x_k + \alpha_k d_k)|| ||d_k||^2 = \nabla f(x^*)
\]  

(46)

Hence, from (45) and (46), we obtain \(-\nabla f(x^*)^T \nabla f(x^*) \geq 0\), which means \(||\nabla f(x^*)||=0\). This contradicts with (32). The proof is then completed.

4. NUMERICAL RESULTS

In this section, we tested a simple three term conjugate gradient algorithm and compare it’s performance with a family of derivative free conjugate gradient method for largescale nonlinear systems of equations [24]:

Problem 3, 5, 8, and 10 are constructed by us where as the remaining are the reference therein. The test functions are listed as follows

Problem 1: see [18]
\[ F_i(x) = x_i (\cos x_i - \frac{1}{x_i}) - x_n [\sin x_i - 1 - (x_i - 1)^2 - \frac{1}{n} \sum_{i=1}^{n} x_i] 
\]
i=1, 2, ..., n, \( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 2: [25]
\[ F_i(x) = e^i - 1 \]
i=1, 2, ..., n.
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 3: System of n nonlinear equations
\[ F_i(x) = 1 - x_i^3 + x_i + x_i x_{n-2} x_{n-1} x_n - 2; \]
i=2, 3, ..., n.
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 4: System of n nonlinear equations [18]
\[ F_i(x) = x_i - 3 \sin(t_i^2 - 0.66) + 2, \]
i=2, 3, ..., n - 1.
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 5: System of n nonlinear equations
\[ F_i(x) = \cos x_i - 9 + 3 x_i + 8 e^{x_i^2}, \]
\[ F_i(x) = \cos x_i - 9 + 3 x_i + 8 e^{x_i^2}, \]
i=1, 2, ..., n.
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 6: System of n nonlinear equations [25]
\[ F_i(x) = x_i^2 + (x_i - 3) \log(x_{i+1}) - 9 + (x - 3) \]
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 7: System of n nonlinear equations [18]
\[ F_i(x) = e^{x_i^2 - l} - \cos(1 - x_i), \]
i=1, 2, ..., n.
\( x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T \)

Problem 8: System of n nonlinear equations
\[ F_i(x) = (0.5 - x_i)^2 + (n + 1 - i)^2 - 0.25 x_i = 1, \]
\[ F_i(x) = \frac{n}{10} - e^{-x_i^2}, \quad i = 1, 2, \ldots, n. \]
\[ x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T \]

**Problem 9:** System of \( n \) nonlinear equations \([25]\)

\[ F_i(x) = 4x_i + x_{i+1} - 2x_i - x_{n+1}^2 \]
\[ F_n(x) = 4x_n + x_{n-1} - 2x_n - x_{n+1} \]
\[ i = 1, 2, \ldots, n - 1. \]
\[ x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T \]

**Problem 10:** System of \( n \) nonlinear equations

\[ F_i(x) = x_i^2 - 4, \]
\[ x_0 = (0.5, 0.5, 0.5, \ldots, 0.5)^T \]

### Table 1. Numerical Results

| Algorithm        | NF | CPU time | NF | CPU time |
|------------------|----|----------|----|----------|
| **STTCG Algorithm** |    |          |    |          |
| F Dim            |    |          |    |          |
| 1 1000           | 8  | 0.399162 | 27 | 0.485588 |
| 5000            | 9  | 0.225975 | 27 | 2.015259 |
| 10000          | 9  | 0.394157 | 27 | 4.539496 |
| 100000        | 9  | 3.706255 | 27 | 32.24164 |
| 1000000       | 86 | 0.146515 | 110| 1.87577  |
| 2 5000          | 93 | 0.608405 | 118| 7.177654 |
| 10000         | 97 | 1.056327 | 121| 14.69088 |
| 100000        | 108| 10.0249 | 132| 150.8311 |
| 1000000       | 50 | 0.122658 | 23 | 0.397317 |
| 3 5000          | 55 | 0.551853 | 23 | 0.728406 |
| 10000         | 57 | 1.065817 | 24 | 3.305191 |
| 100000        | 64 | 1.20678 | 25 | 30.46621 |
| 1000000       | 59 | 0.165684 | 17 | 0.270884 |
| 4 5000          | 64 | 0.735837 | 18 | 1.382743 |
| 10000         | 66 | 1.336673 | 18 | 2.614208 |
| 100000        | 89 | 1.470021 | 25 | 30.48919 |
| 1000000       | 90 | 0.312973 | 120| 2.058193 |
| 5 5000          | 97 | 1.402195 | 127| 8.248632 |
| 10000         | 99 | 2.499665 | 130| 16.05217 |
| 100000        | 108| 23.19104| 140| 170.3214 |
| 1000000       | 17 | 0.07497 | 20 | 0.356372 |
| 6 5000          | 18 | 0.29091 | 22 | 1.72317  |
| 10000         | 22 | 0.629363 | 30 | 0.916266 |
| 100000        | 45 | 11.62368 | 48 | 2.311587 |

**DF-SDCG Algorithm**

| NF | CPU time | NF | CPU time |
|----|----------|----|----------|
| 1  | 0.370238 | 4.52E | 9.01E |
| 5000 | 4.818576 | 7.20E | 9.81E |
| 10000 | 49.78434 | 6.90E | 3.622161 |
| 100000 | 6.90017 | 9.36E | 2.058193 |
| 1000000 | 8.81E | 6.90017 | 2.058193 |
| 10000000 | 18.05217 | 1.72317 | 2.058193 |
| 100000000 | 49.78434 | 6.90017 | 2.058193 |
| 1000000000 | 8.81E | 6.90017 | 2.058193 |

### Table 2. Numerical Results continue

| Algorithm        | NF | CPU time | NF | CPU time |
|------------------|----|----------|----|----------|
| **STTCG Algorithm** |    |          |    |          |
| F Dim            |    |          |    |          |
| 7 1000           | 21 | 0.06948 | 48 | 0.866479 |
| 5000            | 22 | 0.2508 | 51 | 3.622161 |
| 10000          | 38 | 0.457754 | 52 | 6.60117 |
| 100000        | 12 | 8.84E | 70 | 81.89273 |
| 8 5000          | 38 | 8.458E | 70 | 81.89273 |
| 10000         | 13 | 1.62693 | 38 | 2.795954 |
| 100000        | 14 | 3.553242 | 42 | 49.78434 |
| 1000000       | 6  | 9.46E | 25 | 0.486923 |
| 9 5000          | 7  | 0.127945 | 26 | 2.085478 |
| 10000         | 7  | 0.256841 | 27 | 4.818576 |
| 100000        | 7  | 2.880648 | 29 | 35.18388 |
| 1000000       | 59 | 0.103063 | 22 | 0.370238 |
| 10 5000         | 64 | 0.409039 | 22 | 1.638231 |
| 10000         | 66 | 0.766193 | 23 | 3.171502 |
| 1000000       | 99 | 1.20289 | 37 | 42.88408 |

**A Simple Three-term Conjugate Gradient Algorithm for Solving Symmetric Systems of ... (M. Y. Wazirji)**
In the computational experiments, we compare the performance of the method introduced in this work with that of a three-terms PolakRibirePolak conjugate gradient algorithm for large-scale nonlinear equations in order to check its effectiveness. Numerical computations have been performed in MATLAB R2013a on a PC with Intel CELERON(R) processor with 4.00GB of RAM and CPU 1.80GHz. We used 10 test problems with dimensions 1000, 5000, 10000 and 100000 to test the performance of the proposed method in terms of the number of iterations (NI) and the CPU time (in seconds). We declare a termination of the method whenever the number of iteration is greater than 300. The parameters were chosen as \( r = 0.1, \sigma = 0.01, s = 1, \rho = 0.1 \) and \( \epsilon = 10^{-4} \).

\[
\|F(x_k)\| < 10^{-4}. \tag{48}
\]

In the columns of table 1 we have the following:

- \( \text{Dim} \) : the dimension of the problem.
- \( \text{NI} \) : the number of iteration.
- \( \text{NF} \) : the function norm evaluation when the program is stopped.
- \( \text{CPUtime} \) : the cpu time in seconds.

The numerical result in Tables 1 and 2, when comparing STTCG with the DF-SDCG subject to CPU time in seconds, we see that STTCG is top performer. Comparing STTCG with DF-SDCG subject to number of iterations, we see that STTCG was better in 7 problems (i.e it achieved minimum number of iterations) while DF-SDCG was better in 3 problems. Therefore, in comparison is shown in Figures 1 and 2 are performance profile derived by Dolan and More [26], with DF-SDCG, STTCG appears to generate the best search direction and best stepleth. The direction \( d_{k+1} \) given by (13) used in STTCG satisfied the descent condition, and the restarted scheme proved to be more robust in numerical experiments and applications.

5. CONCLUSION

In this paper a new three-term conjugate gradient algorithm as a modification of BFGS quasi-Newton update for with descent direction is has been presented. The convergence of this algorithm was proved using a derivative free linesearch. Intensive numerical experiments on some benchmark nonlinear system of equations of different characteristics proved that the suggested algorithm is faster and more efficient compared to three term DF-SDCG algorithm [24].

REFERENCES

[1] S. Buhmiler, N. Kreji, and Z. Luanin, Practical quasi-Newton algorithms for singular nonlinear systems, Numer. Algorithms 55 (2010) 481502.
[2] G. Fasano, F. Lampariello, M. Sciacalone, A truncated nonmonotone GaussNewton method for large-scale nonlinear least-squares problems, Comput. Optim. Appl. 34 (2006) 343358.
[3] D. Li, M. Fukushima, A global and superlinear convergent GaussNewton-based BFGS method for symmetric nonlinear equations, SIAM J. Numer. Anal. 37 (1999) 152172.
[4] G. Gu, D. Li, L. Qi, S. Zhou, Descent directions of quasi-Newton methods for symmetric nonlinear equations, *SIAM J. Numer. Anal.* 40 (2002) 1763-1774. X. Tong, L. Qi, On the convergence of a trust-region method for solving constrained nonlinear equations with degenerate solutions, *J. Optim. Theory Appl.* 123 (2004) 187211.
[5] Y. H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, 10 (1999), 177-182.
[6] Y. Narushima and H. Yabe, Conjugate gradient methods based on secant conditions that generate descent search directions for unconstrained optimization, *Journal of Computational and Applied Mathematics* 236 (2012) 4303-4317.
[7] N. Andrei, A modified Polak-Ribière-Polyak conjugate gradient algorithm for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 60 (2011) 1457-1471.
[8] N. Andrei, A simple three-term conjugate gradient algorithm for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 241 (2013) 19-29.
[9] N. Andrei, On three-term conjugate gradient algorithms for unconstrained optimization, *Applied Mathematics and Computation*, 219 (2013) 6316-6327.
[10] G. Yuan, X. Lu, Z. Wei, BFGS trust-region method for symmetric nonlinear equations, *J. Comput. Appl. Math.* 230 (2009) 4458.
[11] G. Yuan, Z. Wei, X. Lu, A BFGS trust-region method for nonlinear equations, *Computing* 92 (2011) 317333.
[12] J. Zhang, Y. Wang, A new trust region method for nonlinear equations, *Math. Methods Oper. Res.* 58 (2003) 283298.
[13] C. Kanzow, N. Yamashita, M. Fukushima, Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties, *J. Comput. Appl. Math.* 172 (2004) 375397.
[14] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* 11 (1963) 431441.
[15] A. Bouaricha, R.B. Schnabel, Tensor methods for large sparse systems of nonlinear equations, *Math. Program.* 82 (1998) 377400.
[16] Q. Li, D. Hui L, A class of modified Polak-Ribière-Polyak conjugate gradient algorithms for unconstrained optimization problems, *Journal of Numerical Analysis* 31 (2011) 1625-1635.
[17] W. Leong, M.A. Hassan, M. Y. Waziri, A matrix-free quasi-Newton method for solving nonlinear systems. *Computers and Mathematics with Applications* (2011) 62 2354-2363.
[18] M. Y. Waziri, H.A. Aisha, M. Mamat, A structured Broyden’s-like method for solving systems of nonlinear equations, *Applied Mathematical Science* vol. 8 no.141 (2014) 7039-7046.
[19] L. Zhang, W. Zhou, D.H. Li, A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006) 629-640.
[20] G. Yuan, and M. Zhang, A three-term Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations Journal of Computational and Applied Mathematics , 286 (2015), 186-195 Gonglin Yuana, Maojun Zhang.
[21] S. Deng, Z. Wan, A three-term conjugate gradient algorithm for large-scale Unconstrained optimization problems, Applied Numerical Mathematics (2015), http://dx.doi.org/10.1016/j.apnum.2015.01.008
[22] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, USA 1970.
[23] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization.* 7 (1997) 2633.
[24] W. Cheng et. al, A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations, *Journal of Computational and Applied Mathematics* 222 (2009) 11-19.
[25] La Cruz, W., Martinez, J.M., Raydan, M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations: Theory and experiments, *P. optimization* 6 (2004) 76-79.
[26] E. D. Dolan and J. J. More, Benchmarking optimization software with performance profiles, *Mathematical Programming*, vol. 91, no. 2 (2002) 201213.

A Simple Three-term Conjugate Gradient Algorithm for Solving Symmetric Systems of ... (M. Y. Waziri)