International Journal of
Mathematical Modelling & Computations
Vol. 07, No. 02, Summer 2017, 145- 157

An Alternative Three-Term Conjugate Gradient Algorithm for Systems of Nonlinear Equations

L. Muhammad[1,*] and M. Y. Waziri[2]

[1]*Department of Mathematics, Faculty of Science, Northwest University, Kano,*
[2]*Department of Mathematical Sciences Faculty of Science, Bayero University Kano, Kano, Nigeria.*

Abstract. This paper presents an alternative three-term conjugate gradient algorithm for solving large-scale systems of nonlinear equations. The proposed method is the modification of memoryless BFGS quasi-Newton update for which the direction is descent using projection based technique together with Powel restarting criteria. Moreover, we proved the global convergence of the proposed method with a derivative free line search under suitable assumptions. The numerical results show that the proposed method is promising.

Received: 27 October 2016, Revised: 14 August 2017, Accepted: 04 November 2017.

Keywords: BFGS update, Systems of onlinear equations, Conjugate gradient, Derivative free line search.

Index to information contained in this paper

1. Introduction

Consider the system of nonlinear equations

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

where $F : R^n \to R^n$ is a nonlinear mapping. Often, the mapping, F is assumed to satisfying the following assumptions:
A1. There exists an $x^* \in R^n$ s.t $F(x^*) = 0$;

*Corresponding author. Email: lawalmuhd.nuw.maths@gmail.com

A2. F is montone and continuously differentiable mapping.

The general conjugate gradient conjugate gradient method for solving (1) generate iterative points $\{x_k\}$ from initial given point $\{x_0\}$ as follows:

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

where $\alpha_k > 0$ is attained using line search produre, and the direction d_k is defined by

$$d_{k+1} = -F(x_{k+1}) + \beta_k d_k, \quad k \geqslant 1 \qquad\qquad d_0 = -F(x_0), \quad k = 0. \tag{3}$$

The β_k is a scalar termed as conjugate gradient parameter.

Several methods have been developed for solving (1), most of these methods fall within the framework of Newton and quasi-Newton strategy [5, 13, 17], and are particularly accepted because of their rapid convergence property from a sufficiently good initial guess. Notwithstanding, they are usually not powerful for large-scale nonlinear systems of equations because it requires compution and storage of the Jacobian matrix and its inverse as well as solving n linear system of equations in each iteration and the convergence may even be lost when the Jacobian is singular.

The conjugate gradient methods are also developed in order to eliminate some of the shortcomings of Newton's and Quasi-Newton methods for unconstrained optimization. An extention of its application to handle nonlinear system of equations was considered possible by some reserchers see [6, 13, 14, 18, 21].

In this paper we are particularly interested in three term conjugate gradient algorithm for solving large scale systems of nonlinear equations which is obtained by modifying the update of the memoryless BFGS inverse approximation of the inverse Jacobian restarted as a θ multiple of an identity matrix at every iteration. The method posseed low memory requirement and, 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 recent three-term conjugate gradient method [2] for unconstrained optimization. In other words our algorithm can be thought as an application to three-term conjugate gradient method to a general systems of nonlinear equations. We present extensive numerical results and performance comparism with a DF-SDCG algorithm for large-scale nonlinear equations [9] which illustrated that the proposed algorithm is efficient and promising.

The rest of the paper is organized as follows: In section 2, we present the details of the proposed method. Subsequently Convergence results are presented in section 3. Some numerical results are reported in section 4. Finally, conclusions are made in section 5.

Throughout this paper, $\|.\|$ denote the euclidean norm of a vector. For simplicity, we abbreviate $F(x_k)$ as F_k in the context.

2. Algorithm

This section, presents a three term conjugate gradient algorithm for solving large-scale systems of nonlinear equations.

It is well known that if the Jacobian matrix J_k of F is positive definite, then the most efficient search direction at x_k is the Newton direction

$$d_{k+1} = -(J_{k+1})^{-1}F_{k+1}, \tag{4}$$

Thus, for the Newton direction d_{k+1} holds true that

$$s_k^T J_{k+1} d_{k+1} = -s_k^T F_{k+1}, \tag{5}$$

where $s_k = x_{k+1} - x_k$, and from secant condition that,

$$J_{k+1}s_k = y_k \tag{6}$$

and $y_k = F_{k+1} - F_k$.
(5) can be approximated by

$$y_k^T d_{k+1} = -s_k^T F_{k+1}. \tag{7}$$

Therefore any a search direction d_{k+1} that is required to satisfy (7), then it can be regarded as an approximate Newton direction.
Recall that, the BFGS update of the Jacobian inverse approximation is represented as

$$B_{k+1} = B_k - \frac{s_k y_k^T B_k + B_k y_k s_k^T}{y_k^T s_k} + \left(1 + \frac{y_k^T B_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k} \tag{8}$$

By setting $B_k = \theta_k I$ (8) can be transformed into

$$Q_{k+1} = \theta_k I - \frac{s_k y_k^T \theta_k - \theta_k y_k s_k^T}{y_k^T s_k} + \left(1 + \frac{y_k^T \theta_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k} \tag{9}$$

Where,

$$\theta_k = \frac{s_k^T s_k}{s_k^T y_k} \tag{10}$$

For the θ_k see Raydan [16] for details. It can be noted that, the matrix Q_k in (9) is a modification of (8) in the sense that it is restarted with the multiple of an identity matrix at every step ($B_k = \theta_k I$) and modifying the sign infront of $y_k s_k^T$ in the second term of (9) to get the decsent property.
Multiplying both sides of (9) by $F(x_{k+1})$ we obtained

$$Q_{k+1}F_{k+1} = \theta_k F_{k+1} - \left(\frac{s_k y_k^T \theta_k - \theta_k y_k s_k^T}{y_k^T s_k}\right)F_{k+1} + \left(1 + \frac{y_k^T \theta_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k}F_{k+1} \tag{11}$$

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

$$d_{k+1} = -Q_k F_{k+1} \tag{12}$$

hence our new direction is;

$$d_{k+1} = -\theta_k F_{k+1} - \delta_k s_k - \eta_k y_k \tag{13}$$

where

$$\delta_k = \left(1 + \frac{y_k^T \theta_k y_k}{y_k^T s_k}\right) \frac{s_k^T F_{k+1}}{y_k^T s_k} - \frac{y_k^T \theta_k F_{k+1}}{y_k^T s_k}, \tag{14}$$

$$\eta_k = \frac{\theta_k s_k^T F_{k+1}}{y_k^T s_k}, \tag{15}$$

Solodov and Svaiter [17] introduced projection based algorithm for optimization problems which requires a hyperplane that separate the current iterate x_k from zero of the system of the equation, and this can easely be done by setting

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{||F(z_k)||^2} F(z_k). \tag{16}$$

Therefore incoporating this projection based algorithm together with the proposed search direction and the derivative freee line search of Li and Li [14] we find $\alpha_k = \max\{s, \rho s, \rho^2 s, ...\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geqslant \sigma \alpha_k ||F(x_k + \alpha_k d_k)|| ||d_k||^2. \tag{17}$$

Where $\sigma, s > 0$ and $\rho \in (0, 1)$. we describe our proposed algorithm as follows;

Algorithm 2.1 (ATTCG)

Step 1 : Given x_0 $\sigma \in (0, 1)$, and compute $d_0 = -F_0$, set $k = 0$.
Step 2 : If $||F_k|| < \epsilon$. then stop; otherwise continue with Step 3.
Step 3 : Determine the stepsize α_k by using conditions in (17),
Step 4 : Let the next iterate be $z = x_k + \alpha_k d_k$.
Step 5 : If $||F(z_k)|| < \epsilon$ then stop; otherwise compute x_{k+1} by (16)
Step 6 : Find the search direction by (13).
Step 7 : Restart criterion. If $|F_{k+1}^T F_k|^2 > 0.2||F_{k+1}||^2$, then set $d_{k+1} = -F_{k+1}$.
Step 8 : Consider $k = k + 1$ and go to step 2.

3. Convergence Result

In this Section, we will establish the global convergence for the ATTCG algorithm .

Assumption A

(i) The level set $S = \{x \in R^n : ||F(x)|| \leqslant ||F(x_0)||\}$ is bounded, i.e there exists positive constant $B > 0$ such that, for all $x \in S, ||x|| \leqslant B$.

(ii) The function F is lipschitz continuous on R^n i.e there exist a positive constant $L > 0$ s.t $\forall\ x, y \in R^n$

$$\|F(x) - F(y)\| \leqslant L\|x - y\| \tag{18}$$

Assumption A(ii) implies that there exists a constant $\lambda > 0$ such that

$$||F_k|| \leqslant \lambda. \tag{19}$$

The following lemma is in Solodov and Svaiter [17], from Theorem 2.1 which only state the proof can be in similar way as in [17].

Lemma 1

Suppose that Assumption A holds and $\{x_k\}$ is generated by TTCG Algorithm let x_* be a solution of problem (1) with $F(x_*) = 0$ Then

$$||x_{k+1} - x_*||^2 \leqslant ||x_k - x_*||^2 - ||x_{k+1} - x_k||^2 \tag{20}$$

holds and the sequence $\{x_k\}$ is bounded. Furthermore, either the sequence $\{x_k\}$ is finite and the last iteration is a solution or the sequence is infinite and

$$\sum_{K\geqslant 0} ||x_{k+1} - x_k||^2 < \infty \tag{21}$$

Lemma 2 Suppose that the line search satifies the condition (17) and F is uniformly convex then d_{k+1} given by (13) and (14)-(15) is a descent direction, and

$$||F_{k+1}|| \leqslant ||d_{k+1}||$$

Proof. Since F is uniformly convex it follows that $y_k^T s_k > 0$. Now, by direct computation, we have

$$F_{k+1}^T d_{k+1} = -||F_{k+1}||^2 - \left(1 + \frac{||y_k||^2|\theta_k|}{y_k^T s_k}\right) \frac{(s_k^T F_{k+1})^2}{y_k^T s_k} \leqslant 0. \tag{22}$$

$$||F_{k+1}|| \leqslant -F_{k+1}^T d_{k+1}$$

$$\leqslant ||F_{k+1}||^T ||d_{k+1}||$$

$$||F_{k+1}|| \leqslant ||d_{k+1}|| \tag{23}$$

Lemma 3 Suppose that assumptions A hold, and consider the ATTCG algorithm and (13) where d_k is a descent direction and α_k is computed by the condition (17) . Suppose that F is uniformly convex function on S, i.e there exists a constant $\mu > 0$ such that

$$(F(x) - F(y))^T (x - y) \geqslant \mu||x - y||^2$$

for all $x, y \in S$; then

$$||d_{k+1}|| \leqslant \kappa + \frac{\kappa}{\mu}(2 + L + \frac{L^2}{\mu}). \tag{24}$$

Proof.

From Lipchitz continuity, we know that $||y_k|| \leqslant L||s_k||$. On the other hand by uniform convexity, it yields

$$y_k^T s_k \geqslant \mu||s_k||^2. \tag{25}$$

Thus, using the Cauchy Shwartz inequality, assumptions A and the above inequalities, we have

$$|\delta_k| \leqslant \frac{|s_k^T F_{k+1}|}{|y_k^T s_k|} + \frac{||y_k||^2|\theta_k s_k^T F_{k+1}|}{|y_k^T s_k|^2} + \frac{|y_k^T \theta_k F_{k+1}|}{|y_k^T s_k|} \tag{26}$$

$$\leqslant \frac{\kappa}{\mu||s_k||} + \frac{L^2\kappa}{\mu^2||s_k||} + \frac{L\kappa}{\mu||s_k||} \tag{27}$$

$$= \frac{\kappa}{\mu}(1 + L + \frac{L^2}{\mu})\frac{1}{||s_k||}. \tag{28}$$

Since

$$|\eta_k| = \frac{|\theta_k s_k^T F_{k+1}|}{|y_k^T s_k|} \leqslant \frac{||s_k||||F_{k+1}||}{\mu||s_k||^2} \leqslant \frac{\kappa}{\mu||s_k||}, \tag{29}$$

Finally

$$||d_{k+1}|| \leqslant |\theta_k|||F_{k+1}|| + |\delta_k|||s_k|| + |\eta_k|||y_k|| \leqslant \kappa + \frac{\kappa}{\mu}(2 + L + \frac{L^2}{\mu}). \tag{30}$$

Hence, d_{k+1} is bounded.

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

Lemma 4

Let the Assumption A hold. Then ATTCG algorithm produces an iterate of $x_{k+1} = x_k + \alpha_k d_k$, in a finite number of backtraking steps.

Proof:

We suppose that $||F_k|| \to 0$ does not hold, or the algorithm is stoped. Then there exists a constant $\epsilon_0 > 0$ such that

$$||F_k|| \geqslant \epsilon_0, \forall \quad k \geqslant 0 \tag{31}$$

up to subsequnce
The aim is to guarantee that the linesearch strategy (17) is always terminated in a finite number of steps with a positive steplength α_k. we will get this by contradiction. Suppose that for some iterate indexes such as k_* the condition (17) is not true.
Then by letting $\alpha_{k_*}^m = \rho^m s$, it can be concluded that

$$-F(x_{k_*} + \alpha_{k_*}^m d_{k_*})^T d_{k_*} < \sigma \alpha_{k_*}^m ||F(x_{k_*} + \alpha_{k_*}^m d_{k_*})|| ||d_{k_*}||^2, \qquad \forall \quad m \geqslant 0.$$

combining the above inequality with (22), we have

$$||F(x_{k_*})||^2 = -d_{k_*}^T F_{k_*} - \left(1 + \frac{||y_{k_*}||^2 |\theta_{k_*}|}{y_{k_*}^T s_{k_*}}\right) \frac{(s_{k_*}^T F_{k_*})^2}{y_{k_*}^T s_{k_*}}$$

$$= [F(x_{k_*} + \alpha_{k_*}^m d_{k_*}) - F(x_{K_*})]^T d_k - F(x_{k_*} + \alpha_{k_*}^m d_{k_*})^T d_{k_*}$$

$$< [L + \sigma ||F(x_{k_*} + \alpha_{k_*}^m d_{k_*})] \alpha_{k_*}^m ||d_{k_*}||^2, \quad \forall \quad m \geqslant 0.$$

By (23) and (31)

$$||F(x_{k_*} + \alpha_{k_*}^m d_{k_*})|| \leqslant ||F(x_{k_*} + \alpha_{k_*}^m d_{k_*}) - F_k|| + ||F_k||$$

$$\leqslant L \alpha_{k_*}^m ||d_{k_*}||$$

$$\leqslant Ls(\kappa + \frac{\kappa}{\nu}(2 + L + \frac{L^2}{\mu})).$$

Thus, we obtain

$$\alpha_{k_*}^m > \frac{||F_k||^2}{[L + \sigma ||F(x_{k_*} + \alpha_{k_*} d_{k_*})||] ||d_{k_*}||^2}$$

$$> \frac{\epsilon_0^2}{L + Ls(\kappa + \frac{\kappa}{\mu}(2 + L + \frac{L^2}{\mu})) > 0 \forall m \geqslant 0.}$$

Thus, it contradicts with the defination of $\alpha_{k_*}^m$. consequently, the line search procedure (17) can attain a positive steplength α_k in a finite number of backtracking steps. Hence the proof is complete.

Theorem 3.1 Let assumption A hold and $\{\alpha_k, d_k, x_{k+1}, F_{k+1}\}$ be generated by ATTCG algorithm. Then

$$\liminf_{k\to\infty} ||F_k|| = 0 \tag{32}$$

Proof: We prove this theorem by contradiction, namely the relation (31) holds. By (23) we have

$$||d_k|| \geqslant ||F_k|| \geqslant \epsilon_0 \forall k \geqslant 0 \tag{33}$$

By lemma 2

$\alpha_k||d_k|| \to 0, k \to \infty$

It follows that from (33)

$$\lim_{k\to\infty} \alpha_k = 0 \tag{34}$$

By linesearch technique (17) we have

$$-F(x_k + \alpha_k{}' d_k)^T d_k < \sigma\alpha_k{}' ||F(x_k + \alpha_k{}' d_k)||||d_k||^2 \tag{35}$$

where $\alpha_k{}' = \frac{\alpha_k}{\rho.}$. By the boundedness of $\{x_k\}$ in lemma 3 we can deduce that there exist an accumution point $\bar{x}$ and a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ such that

$$\lim_{j\to\infty} x_{k_j} = \bar{x}.$$

The sequence $\{d_k\}$ is bounded. So $\bar{d} \in R^n$ and a subsequence $\{d_{k_j}\}$ (we assume that without loss of generality the subsequences $\{x_{k_j}\}$and $\{d_{k_j}\}$ have the same indices) such that

$$\lim_{k\to\infty} d_{k_j} = \bar{d},$$

thus, taking the limit as $k \to \infty$ up to subsequence in both sides of (35) generates

$$F(\bar{x})^T \bar{d} > 0.$$

According to the other hand, by taking the limit as $k \to \infty$ up to subsequence in both sides of (17) we get

$$F(\bar{x})^T \bar{d} \leqslant 0.$$

Then a contradiction is generated. So the proof is complete.

4. Numerical Results

In this section, we tested ATTCG algorithm and compare it's performance with a DF-SDCG Algorithm [9] :
The test functions are listed as follows. Examples 1 and 5 are from [19], problem 6 and 8 are from [22] and problem 9 and 10 are from [21] where as the rest are arbitrarily constructed by us.

Example 1: *The strictly convex function:*
$F_i(x) = e^{x_i} - 1$
$i = 1, 2, ..., n.$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 2: System of n nonlinear equations
$F_i(x) = x_i - 3x_i(\frac{sinx_i}{3} - 0.66) + 2$
$i = 2, 3, ..., n.$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 3:System of n nonlinear equations
$F_i(x) = \cos x_1 - 9 + 3x_1 + 8e^{x_2},$
$F_i(x) = \cos x_i - 9 + 3x_i + 8e^{x_{i-1}},$
$i = 1, 2, ..., n$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 4:System of n nonlinear equations
$F_i(x) = (0.5 - x_i)^2 + (n + 1 - i)^2 - 0.25x_i - 1,$
$F_n(x) = \frac{n}{10}1 - e^{-x_n^2},$
$i = 1, 2, ..., n - 1.$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 5: System of n nonlinear equations
$F_i(x) = 4x_i + x_{i+1} - 2x_i - x_{\frac{i+1}{3}},$
$F_n(x) = 4x_n + x_{n-1} - 2x_n - x_{\frac{n+1}{3}},$
$i = 1, 2, ..., n - 1.$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 6: System of n nonlinear equations
$F_i(x) = x_i^2 - 4,$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 7:System of n nonlinear equations
$F_1(x) = \sin(x_1 - x_2) - 4e^{2-x_2} + 2x_1,$
$F_i(x) = \sin(2 - x_i) - 4e^{x_i - 2} + 2x_i + \cos(2 - x_i) - e^{2-x_i},$
$i = 2, 3, ..., n.$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 8:System of n nonlinear equations
$F_i(x) = \sum_{i=1}^{n} x_i x_{i-1} + e^{x_{i-1}} - 1,$
$i = 2, 3, ..., n$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
Example 9:System of n nonlinear equations
$F_i(x) = x_i - \sum_{i=1}^{n} \frac{x_i^2}{n^2} + \sum_{i=1}^{n} x_i - n,$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$
$i = 1, 2, ..., n$
Example 10:System of n nonlinear equations
$F_i(x) = 5x_i^2 - 2x_i - 3,$
$i = 1, 2, ..., n$
$x_0 = (0.5, 0.5, 0.5, ..., 0.5)^T$

Table 1. **Numerical results based on dimension of problem (n), Number of iterations and CPU Time (in seconds) of problems 1-6**

	ATTCG Algorithm				DF-SDCG Algorithm		
P	Dim	Iter	$\|\|F_k\|\|$	CPU time	Iter	$\|\|F_k\|\|$	CPU time
1	100	8	9.4560e-06	0.028474	25	6.1207e-05	0.597602
	1000	9	2.1144e-06	0.147652	26	8.2241e-05	2.233604
	5000	9	2.9903e-06	0.251236	27	6.9111e-05	4.507234
	10000	9	9.4560e-06	2.792765	29	7.4551e-05	39.114200
2	100	86	8.9037e-06	0.106252	22	3.9120e-05	0.412667
	1000	90	9.9122e-06	0.434176	22	8.7476e-05	1.935173
	5000	92	9.8911e-06	0.783972	23	5.5264e-05	3.744463
	10000	126	8.4830e-06	12.153144	37	4.5238e-05	46.917528
3	100	-	-	-	-	-	-
	1000	80	9.7709e-06	0.823754	-	-	-
	5000	80	9.7709e-06	1.549614	73	9.6995e-05	10.591806
	10000	82	9.6675e-06	15.591125	75	8.9336e-05	103.256340
4	100	110	9.4602e-06	0.788803	29	5.88E-05	0.608293
	1000	-	-	-	214	9.4753e-05	17.300552
	5000	-	-	-	40	7.2186e-06	7.480797
	10000	-	-	-	40	0	65.391351
5	100	6	6.8151e-07	0.032962	105	9.3785e-05	2.126298
	1000	72	7.7178e-06	1.13311	27	9.1619e-05	2.246521
	5000	11	4.4383e-08	0.440859	117	9.0373e-05	16.357734
	10000	44	3.6077e-06	15.770507	215	2.2548e-07	296.202722
6	100	18	6.6893e-06	0.042864	12	9.7213e-05	0.242993
	1000	19	5.3848e-06	0.159687	14	2.4332e-05	1.246371
	5000	19	7.6152e-06	0.290403	14	3.4411e-05	2.454668
	10000	20	8.6693e-06	3.311656	15	4.8282e-05	20.169900

Table 2. **Numerical results based on dimension of problem (n), Number of iterations and CPU Time (in seconds) of problems 7-10**

	ATTCG Algorithm				TPRP Algorithm		
P	Dim	Iter	$\|\|F_k\|\|$	CPU time	Iter	$\|\|F_k\|\|$	CPU time
7	100	47	7.7441e-06	0.206083	88	9.4836e-05	1.790001
	1000	47	8.8743e-06	2.26032	169	9.8728e-05	12.475926
	5000	54	8.9868e-06	2.26032	103	8.4127e-05	14.483080
	10000	46	9.9537e-06	2.26032	124	9.9146e-05	174.486999
8	100	33	5.4102e-06	0.050556	56	7.9809e-05	1.179710
	1000	35	7.5751e-06	0.22769	59	8.7283e-05	4.438580
	50000	36	6.4277e-06	0.421889	60	9.7255e-05	8.267500
	10000	38	7.3174e-06	5.162862	65	9.3375e-05	84.734466
9	100	10	1.6671e-06	0.024914	14	4.7435e-05	0.110289
	1000	11	2.3311e-06	0.0345	14	8.2750e-06	0.130116
	5000	12	2.4561e-06	-	15	1.8503e-05	0.126861
	10000	12	3.1133e-06	-	17	2.6168e-05	0.339827
10	100	3	7.8660e-07	0.019034	84	9.4809e-05	1.641791
	1000	3	1.7589e-06	0.087521	92	9.0329e-05	6.522887
	5000	3	2.4875e-06	0.164283	95	9.2769e-05	12.342695
	10000	8	1.3604e-12	2.205701	92	9.1449e-05	116.483472

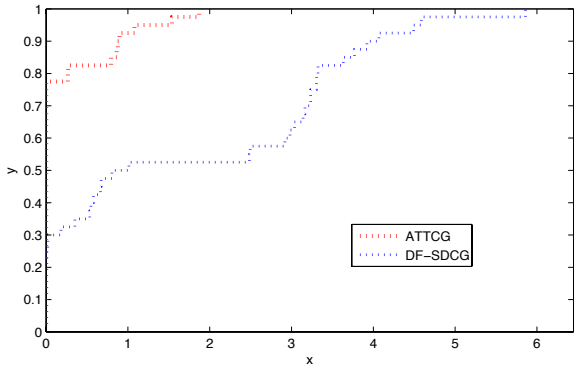

Figure 1. Performance profile of ATTCG and DF-SDCG methods with respect to number of iterations for Examples 1-10

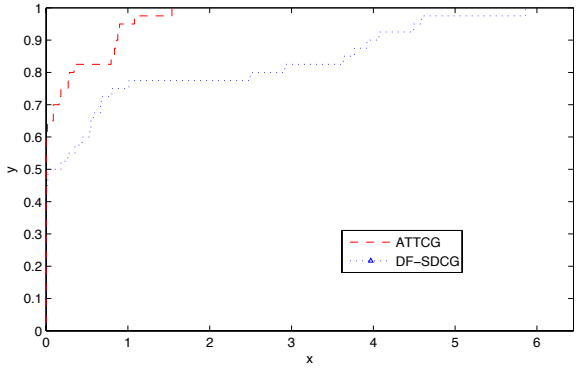

Figure 2. Performance profile of ATTCG and DF-SDCG methods with respect to CPU time in seconds for Examples 1-10

In the experiments, we compare the performance of the method introduced in this work with that of DF-SDCG conjugate gradient methods for large scale nonlinear systems of equations in order to check it's 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 100, 1000, 5000 and 10000 to test the performance of the proposed method in terms of the number of iterations (NI) and the CPU time (in seconds). We defined a termination of the method whenever

$$\|F(\mathrm{x}_k)\| < 10^{-4}. \tag{36}$$

The table list the numerical results, where Iter and Time stand for the total number of all iterations and the CPU time in seconds, respectively; $||F_k||$ is the norm of the residual at the stopping point. We also used "-" to represent failure during iteration process due one of the following:

1. The number of iteration and/or the CPU time in second reaches 1000;
2. Failure on code execution due to insufficient memory;
3. If $||F_k||$ is not a number (NaN).

In Table 1 and 2 we listed numerical results.The numerical results indicated that the proposed method, ATTCG algorithm compared to a DF-SDCG Algorithm [9] for large-scale nonlinear equations has minimum number of iterations and CPU time. Figures 1 and 2 are performance profiles derived by Dolan and More [11] which show that our claim is justified, that is, less CPU time and number of iterations for each test problem. Furthermore, on average, our$||F_k||$ is small which signifies that the solution obtained is true approximation of the exact solution compared to the DF-SDCG Algorithm [9].

5. Conclusion

In this paper a new three-term conjugate gradient algorithm as modification of memoryless BFGS quasi-Newton update with descent direction, 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 a DF-SDCG Algorithm for large-scale nonlinear equations [9].

References

[1] N. Andrei, *A modified Polak-Ribi'ere-Polyak conjugate gradient algorithm for unconstrained optimization*, Journal of Computational and Applied Mathematics, **60** (2011) 1457-1471.

[2] N. Andrei, *A simple three-term conjugate gradient algorithm for unconstrained optimization*, Journal of Computational and Applied Mathematics, **241** (2013) 19-29.

[3] N. Andrei, *On three-term conjugate gradient algorithms for unconstrained optimization*, Applied Mathematics and Computation, **219** (2013) 6316-6327.

[4] A. Bouaricha, R. B. Schnabel, *Tensor methods for large sparse systems of nonlinear equations*, Math. Program, **82** (1998) 377-400.

[5] S. Buhmiler, N. Kreji and Z. Luanin, *Practical quasi-Newton algorithms for singular nonlinear systems*, Numer. Algorithms, **55** (2010) 481-502.

[6] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations.* Math. Comput, (1965) 19 577-593.

[7] C. S. Byeong, et al *A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems* . Appl. Math. Comput, **217** (2010) 3190-3198.

[8] W. Cheng, *A PRP type method for systems of monotone equations,*Journal of Computational and Applied Mathematics, **50** (2009) 15-20.

[9] 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.

[10] Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim, **10** (1999) 177-182, Comput., **19** (1965) 577-593.

[11] E. D. Dolan and J. J. More, *Benchmarking optimization software with performance profiles*, Mathematical Programming, **91 (2)** (2002) 201213.

[12] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, PA: SIAM, Philadelphia, (1995).

[13] W. Leong, M. A. Hassan and M. Y. Waziri, *A matrix-free quasi-Newton method for solving nonlinear systems.* Computers and Mathematics with Applications, **62** (2011) 2354-2363.

[14] Q. Li and D. Hui L, *A class fo derivative free- methods for large-scale nonlinear monotone equations* journal of Numerical Analysis, **31** (2011) 1625-1635.

[15] 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.

[16] M. Raydan, *The Barzilaiai and Borwein gradient method forthe large scale unconstrained minimization problem*, SIAM Journal on Optimization, **7** (1997) 26-33.

[17] M. V. Solodov and B. F. Svaiter, *A globally convergent inexact Newton method for systems of monotone equations*, in: M. Fukushima, L. Qi (Eds.)

[18] G. Yuan and M. Zhang, *A three-terms Polak-Ribiere-Polyak conjugate gradient algorithm for large-scale nonlinear equations* Journal of Computational and Applied Mathematics, **286** (2015) 186-195.

[19] La Cruz. W, Martinez. J. M and Raydan. M, *spetral residual method without gradient information for solving large-scale nonlinear systems of equations: Theory and experiments*, optimization, **76 (79)** 1008-00.

[20] L. Zhang, W. Zhou and D. H. Li, *A descent modified Polak-Ribi'ere-Polyak conjugate gradient method and its global convergnece*, IMA J. Numer.Anal, **26** (2006) 629-640.

[21] M. Y. Waziri and J. Sabi'u, *A deivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations*, International Journal of Mathematics and mathematical Science, 961487 **8** (2015) .

[22] M. Y. Waziri, H. A. Aisha and M. Mamat, *A structured Broyden's-Like method for solving systems of nonlinear equations*, Applied mathematical Science, **8 (141)** (2014) 7039-7046.