Journal of Contemporary Applied Mathematics
V. 7, No 1, 2017, June
ISSN 2222-5498

Effective Algorithm for Solving Symmetric Nonlinear Equations

J. Sabi'u

Abstract. In this article, effective and efficient solver for symmetric nonlinear equations without computing exact gradient and Jacobian matrix with a very low memory requirement is proposed. The global convergence of the proposed method was also established under some rigorous conditions with nonmonotone line search. Numerical experiment shows the proposed method is efficient.

Key Words and Phrases: Nonmonotone line search; Jacobian matrix; symmetric nonlinear equations; Conjugate gradient method.

2010 Mathematics Subject Classifications: 90C30, 65K05, 90C53, 49M37, 15A18

1. Introduction

Considering the systems of symmetric 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 $F(x^*) = 0$
A2. F is a continuously differentiable mapping in a neighborhood of x^*
A3. $F'(x^*)$ is invertible
A4. The Jacobian $F^{'}(x)$ is symmetric.
where the symmetry means that the Jacobian $J(x) := F^T(x)$ is symmetric; that is, $J(x) = J(x)^T$. This class of special equations come from many practical problems such as an unconstrained optimization problem, Karush-Kuhn-Tucker (KKT) of equality constrained optimization problem, saddle point problem, the discritized two-point boundary value problem, the discritized elliptic boundary value problem, and etc. Equation (1) is the first-order necessary condition for the unconstrained optimization problem where F is the gradient mapping of some function $f : R^n \longrightarrow R$,

$$Min f(x), \quad x \epsilon R^n. \tag{2}$$

A large number of efficient solvers for large-scale symmetric nonlinear equations have been proposed, analyzed, and tested by different researchers. Among them are [4, 2, 8]. Still, the matrix storage and solving of an n-linear system of equations are required in the BFGS type methods presented in the literature. The new designed non-monotone spectral gradient algorithm [1] falls within the framework of matrix-free.
Researches on conjugate gradient methods for symmetric nonlinear equations has received a proper attention and take an appropriate progress. However, Li and Wang [**?**] proposed a modified Fletcher-Reeves conjugate gradient method which is based on the work of Zhang et al. [3], and the results illustrated that their proposed conjugate gradient method is promising. In line with this development, further studies on conjugate gradient are [6, 9, 7, 11, 14]. Extensive numerical experiments showed that each over mentioned method performs quite well. Therefore, motivated by [6] this article is aimed at developing a derivative-free conjugate gradient method for solving symmetric nonlinear equations without computing the Jacobian matrix with less number of iterations and CPU timethat is globally convergent. This paper is organized as follows: Next section presents the details of the proposed method. Convergence results are shown in Section 3. Some numerical results are reported in Section 4.

2. Details of the Method

Recall that, in [11] we used the term

$$g_k = \frac{F(x_k + \alpha_k F_k) - F_k}{\alpha_k} \tag{3}$$

To approximate the gradient $\nabla f(x_k)$, which avoids computing exact gradient. Sumit and Nath in [6] proposed a conjugate gradient parameter given by

$$\beta_k = \frac{-t\nabla f(x_k)^T s_{k-1} - (\gamma - 1)\nabla f(x_k)^T \nabla f(x_{k-1})}{d_k^T \nabla f(x_{k-1})}, \tag{4}$$

where $\gamma \in (0, 1)$ and t is any positive constant. From now on, problem (1) is assume to be symmetric and $f(x)$ is defined by

$$f(x) = \frac{1}{2}||F(x)||^2. \tag{5}$$

Then the problem (1) is equivalent to the global optimization problem (2). However, when $f(x)$ is given by (5):

$$\nabla f(x_k) = J(x_k)^T F(x_k) = J(x_k) F(x_k) \tag{6}$$

which requires the computions of both the Jacobian and the gradient of f. However, in this research our scheme is defined by

$$d_k = \begin{cases} -g_k & if \quad k = 0 \\ -g_k + \beta_k^* d_{k-1} & if \quad k \geq 1 \end{cases} \tag{7}$$

where β_k^* is given by

$$\beta_k^* = \frac{-t_k g_k^T s_{k-1} - (\gamma - 1) g_k^T g_{k-1}}{d_k^T g_{k-1}}, \tag{8}$$

$||.||$ is the Euclidean norm, $\gamma \in (0,1)$ and t_k is the inverse of the Rayleigh quotient [13] defined by

$$t_k = \frac{s^T s_k}{s_k^T y_k}. \tag{9}$$

Lemma 2.1. *Let the above assumption be satisfied and* $\{\alpha_k, d_k, x_{k+1}\}$ *be generated by MHCG algorithm. Then there exist a positive constant* m *such that for all* k

$$y_k^T s_k \geq m \|s_k\|^2. \tag{10}$$

Proof. By the mean-value theorem, we have

$$y_k^T s_k = s_k^T (g_{k+1} - g_k) = s_k^T \nabla g(\xi) s_k \geq m \|s_k\|^2, \tag{11}$$

where $\xi = x_k + \varsigma(x_{k+1} - x_k), \quad \varsigma \in (0,1)$. Moreover, the direction d_k given by (7) may not be a descent direction of (5), then the standard wolfe and Armijo line searches can not be used to compute the stepsize directly. Therefore, the nonmonotone line search used in [9, 10, 11] is the best choice to compute the stepsize α_k. Let $\omega_1 > 0$, $\omega_2 > 0$, $r \in (0,1)$ be constants and $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \infty. \tag{12}$$

Let $\alpha_k = max \left\{1, r^k\right\}$ that satisfy

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\omega_1 ||\alpha_k F(x_k)||^2 - \omega_2 ||\alpha_k d_k||^2 + \eta_k f(x_k). \tag{13}$$

Algorithm 1
Step 1 : Given x_0, $\alpha_k > 0$, $\omega \in (0,1)$, $r \in (0,1)$ and a positive sequence η_k satisfying (12), then compute $d_0 = -g_0$ and set $k = 0$.
Step 2 **:** Test a stopping criterion. If yes, then stop; otherwise continue with Step 3.
Step 3 : Compute α_k by the line search (13).
Step 4 : Compute $x_{k+1} = x_k + \alpha_k d_k$.
Step 5 : Compute the search direction by (7).

Step 6 : Consider $k = k + 1$ and go to step 2.

3. Convergence Result

This section presents global convergence results of an efficient hybrid CG method. To begin with, defined the level set

$$\Omega = \{x | f(x) \leq e^{\eta} f(x_0)\}, \tag{14}$$

where η satisfies

$$\sum_{k=0}^{\infty} \eta_k \leq \eta < \infty. \tag{15}$$

Lemma 3.1. *[4] Let the sequence $\{x_k\}$ be generated by algorithm 1. Then the sequence $\{||F_k||\}$ converges and $x_k \epsilon \Omega$ for all $k \geq 0$.*

Proof. For all k, from (13) we have $\|F_{k+1}\| \leq (1+\eta_k)^{\frac{1}{2}} \|F_k\| \leq (1+\eta_k)\|F_k\|$. Since η_k satisfies (12), we conclude that $\{\|F_k\|\|$ converges. Moreover, we have for all k

$$\begin{aligned}
\|F_{k+1}\| &\leq (1+\eta_k)^{\frac{1}{2}} \|F_k\| \\
&\vdots \\
&\leq \prod_{i=0}^{k} (1+\eta_k)^{\frac{1}{2}} \|F_0\| \\
&\leq \|F_0\| \left[\frac{1}{k+1} \sum_{i=0}^{k} (1+\eta_i) \right]^{\frac{k+1}{2}} \leq \|F_0\| \left[1 + \frac{1}{k+1} \sum_{i=0}^{k} \eta_i \right]^{\frac{k+1}{2}} \\
&\leq \|F_0\| \left(1 + \frac{\eta}{k+1} \right)^{\frac{k+1}{2}} \leq \|F_0\| \left(1 + \frac{\eta}{k+1} \right)^{k+1} \leq e^{\eta} \|F_0\|,
\end{aligned}$$

where η is a constant satisfying (12). This implies that $x_k \in \Omega$.
In order to get the global convergence of DFCG algorithm, we need the following assumptions.
(i) The level set Ω defined by (14) is bounded
(ii) In some neighbourhood N of Ω, the Jacobian of F is symmetric, bounded and positive definite. Namely, there exists $L > 0$ such that

$$||J(x) - J(y)|| \leq L||x - y||, \quad \forall x, y \in N. \tag{16}$$

Li and Fukushima in [4] showed that, there exists positive constants M_1, M_2 and L_1 such that

$$||F(x)|| \leq M_1, \quad ||J(x)|| \leq M_2, \quad \forall x \epsilon N, \tag{17}$$

$$||\nabla f(x) - \nabla f(y)|| \leq L_1||x-y||, \quad ||J(x)|| \leq M_2, \quad \forall x, y \epsilon N. \tag{18}$$

Lemma 3.2. *Let the properties of* (1) *above hold. Then we have*

$$\lim_{k\to\infty} ||\alpha_k d_k|| = \lim_{k\to\infty} ||s_k|| = 0, \tag{19}$$

and

$$\lim_{k\to\infty} ||\alpha_k F_k|| = 0. \tag{20}$$

Proof. by (12) and (13) we have for all $k > 0$,

$$\omega_1||\alpha_k F(x_k)||^2 + \omega_2||\alpha_k d_k||^2 \leq f(x_k) - f(x_{k+1}) + \eta_k f(x_k), \tag{21}$$

by summing the above k inequality, then we obtain:

$$\sum_{i=0}^{m} \omega_1||\alpha_k F(x_k)||^2 + \omega_2||\alpha_k d_k||^2 \leq f(x_1) - f(x_m) + \sum_{i=0}^{m} \eta_i f(x_k). \tag{22}$$

So, from (18) and the fact that $\{\eta_k\}$ satisfies (12) the result follows.
The following result shows that our algorithm 1 is globally convergent.

Theorem 3.3. *Let the properties of* (1) *hold. Then the sequence* $\{x_k\}$ *be generated by algorithm 1 converges globally, that is,*

$$\liminf_{k\to\infty} ||\nabla f(x_k)|| = 0. \tag{23}$$

Proof. We prove this theorem by contradiction. Suppose that (23) is not true, then there exists a positive constant τ such that

$$||\nabla f(x_k)|| \geq \tau, \quad \forall k \geq 0. \tag{24}$$

Since $\nabla f(x_k) = J_k F_k$, (24) implies that there exists a positive constant τ_1 satisfying

$$||F_k|| \geq \tau_1, \quad \forall k \geq 0. \tag{25}$$

Case (i): $\limsup_{k\to\infty} \alpha_k > 0$. then by (19), we have $\liminf_{k\to\infty} ||F_k|| = 0$. This and Lemma (3.1) show that $\lim_{k\to\infty} ||F_k|| = 0$, which contradicts with (24). Case (ii): $\limsup_{k\to\infty} \alpha_k = 0$. Since $\alpha_k \geq 0$,this case implies that

$$\lim_{k\to\infty} \alpha_k = 0. \tag{26}$$

by definition of g_k in (3) and the symmetry of the Jacobian, we have

$$||g_k - \nabla f(x_k)|| = ||\frac{F(x_k + \alpha_{k-1}F_k) - F_k}{\alpha_{k-1}} - J_k^T F_k||$$

$$= || \int_0^1 J(x_k + t\alpha_{k-1}F_k) - J_k)dtF_k|| \leq LM_1^2\alpha_{k-1},$$

where we use (17) and (18) in the last inequality. (12), (13) and (24) show that there exists a constant $\tau_2 > 0$ such that

$$||g_k|| \geq \tau_2, \quad \forall k \geq 0. \tag{27}$$

By (3) and (17), we get

$$||g_k|| = \| \int_0^1 J(x_k + t\alpha_{k-1}F_k)F_k dt\| \leq M_1M_2, \quad \forall k \geq 0. \tag{28}$$

From (28) and (18), we obtain

$$\begin{aligned} ||y_k|| &= ||g_k - g_{k-1}|| \\ \leq ||g_k - \nabla f(x_k)|| &+ ||g_{k-1} - \nabla f(x_{k-1})|| + ||\nabla f(x_k) - \nabla f(x_{k-1})|| \\ &\leq LM_1^2(\alpha_{k-1} + \alpha_{k-2}) + L_1||s_{k-1}||. \end{aligned} \tag{29}$$

This together with (26) and (19) shows that $\lim_{k\to\infty} ||y_k|| = 0$. Again from the definition of our β_k^* we obtain

$$|\beta_k^*| \leq \frac{-t_k g_k^T s_{k-1} - (\gamma - 1)g_k^T g_{k-1}}{d_k^T g_{k-1}} ||y_{k-1}|| \longrightarrow 0 \tag{30}$$

which implies there exists a constant $\rho \in (0, 1)$ such that for sufficiently large k

$$|\beta_k^*| \leq \rho. \tag{31}$$

Without lost of generality, assume that the above inequalities holds for all $k \geq 0$. Clearly from (31) and (29) we can conclude that the sequence $\{d_k\}$ is bounded. Since $\lim_{k\to\infty} \alpha_k = 0$, then $\alpha_k' = \frac{\alpha_k}{r}$ does not satisfy (13), namely

$$f(x_k + \alpha_k' d_k) > f(x_k) - \omega_1||\alpha_k' F(x_k)||^2 - \omega_2||\alpha_k' d_k||^2 + \eta_k f(x_k), \tag{32}$$

which implies that

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

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

$$\frac{f(x_k + \alpha_k' d_k) - f(x_k)}{\alpha_k'} = \nabla f(x_k + \delta_k \alpha_k' d_k)^T d_k. \tag{34}$$

Since $\{x_k\} \subset \Omega$ is bounded, without loss of generality, we assume $x_k \longrightarrow x^*$. By (3) and (7), we have

$$\lim_{k\to\infty} d_k = -\lim_{k\to\infty} g_k + \lim_{k\to\infty} \beta_k^* d_{k-1} = -\nabla f(x^*), \tag{35}$$

where we use (30), (13) and the fact that the sequence $\{d_k\}$ is bounded. On the other hand, we have

$$\lim_{k\to\infty} \nabla f(x_k + \delta_k \alpha_k^{'} d_k) = \nabla f(x^*). \tag{36}$$

Hence, from (33)-(36), we obtain

$$-\nabla f(x^*)^T \nabla f(x^*) \geq 0, \tag{37}$$

which means $||\nabla f(x^*)|| = 0$. This contradicts with (24). The proof is completed.

4. Numerical results

In this section, we compared the performance of the proposed method with the norm descent conjugate gradient method for symmetric nonlinear equations (NDCG) [7]. For the proposed algorithm the following parameters are set to $\omega_1 = \omega_2 = 10^{-4}$, $\alpha_0 = 0.01$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$. while for NDCG; $\xi = 10$, $\rho = 0.3$, $\delta = 0.001$, $\eta_k = \frac{1}{(k+1)^2}$ and $\theta = 0.2$. The codes for both methods were written in Matlab 7.4 R2010a and run on a personal computer 1.8 GHz CPU processor and 4 GB RAM memory. We stopped the iteration if the toatal number of iterations exceeds 2000 or $||F_k|| \leq 10^{-4}$. "-" to represents failure due to; (i) Memory requirement (ii) Number of iteration exceed 1000 (iii) If $||F_k||$ is not a number. The methods were tested on some Benchmark test problems with different initial points. Problem 1 and2 are from [11] whilethe remaining one is an artifitial problem.

Problem 1

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

$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1 \quad ; 1, 2, \ldots, n-1$

$F_n(x) = x_n(x_{n-1} + x_n^2)$.

Problem 2. *(n is multiple of 3)* for $i = 1, 2, , n/3$,

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

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

$F_3i(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}$.

Problem 3. *The variable band function:*

$F_1(x) = -2x_1^2 + 3x_1 - 2x_2 + 0.5x_3 + 1$

$F_i(x) = -2x_i^2 + 3x_i - x_{i-1} - 1.5x_{i+1} + 1$ for $i = 2, 3, \ldots, n-1$

$F_n(x) = -2x_n^2 + 3x_n - 0.5x_{n-1} + 1$

Table 1: Numerical comparison of the proposed method versus NDCG

			Problem 1		
		Algorithm		**NDCG**	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
1000	x_1	175	0.420227	157	0.773793
	x_2	31	0.131822	183	0.733555
	x_3	58	0.198267	86	0.369955
	x_4	199	0.515923	214	0.836359
20000	x_1	164	6.449805	157	7.684803
	x_2	33	1.352192	264	13.27992
	x_3	42	2.217185	193	9.416138
	x_4	476	17.99319	124	6.435678
			Problem 2		
		Algorithm		**NDCG**	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
1000	x_1	5	0.033245	133	0.671819
	x_2	24	0.117488	-	-
	x_3	32	0.141892	-	-
	x_4	21	0.108126	188	0.965073
20000	x_1	8	0.50871	71	4.008832
	x_2	29	1.29933	-	-
	x_3	35	1.641559	-	-
	x_4	15	0.806084	136	7.643776
			Problem 3		
		Algorithm		**NDCG**	
Dimension	*Guess*	*iter*	*Time*	*iter*	*Time*
1000	x_1	4	0.040597	-	-
	x_2	0	0.006984	0	0.006419
	x_3	4	0.026863	12	0.073842
	x_4	21	0.101723	188	0.836854
20000	x_1	9	0.521719	-	-
	x_2	0	0.144088	0	0.123382
	x_3	10	0.628647	-	-
	x_4	8	0.472832	-	-

The tables listed 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.The numerical results indicate that the proposed Algorithm has rigorous advantages when compared to NDCG, i.e. minimum number of iteration and CPU time in almost all the tested problems. Also $x_1 = (1, 1, \ldots, n)$, $x_2 = (0, 0, \ldots, 0)$, $x_3 = (0.1, 0.1, \ldots, 0.1)$ and $x_4 = (1 - 1, 1 - \frac{1}{2}, 1 - \frac{1}{3}, \ldots, 1 - \frac{1}{n})$.

Competing Interests

The author declare that no competing interests exist.

References

[1] W. Cheng, Z. Chen, " Nonmonotone Spectral method for large-Scale symmetric nonlinear equations", Numer. Algorithms, vol. 62, pp.62149-162, 2013

[2] G.Z. Gu, D.H. Li,L. Qi, S.Z. Zhou, "Descent direction of quasi-Newton methods for symmetric nonlinear equations", SIAM J. Numer. Anal., vol. 40, pp. 1763-1774, 2002.

[3] L. Zhang, W. Zhou, D.H. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search", Numer. Math., vol. 104, pp.561-572, 2006.

[4] D.H. Li, Fukushima, "A globally and superlinearly convergent Gauss-Newton-based BFGS methods for symmetric nonlinear equations", SIAM J. Numer. Anal., vol.37, pp.152-172, 1999. bibitemjam61D.H. Li, X. Wang, " A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations", Numer. Algebra Control Optim., vol1, pp.71-82, 2011.

[5] W. Zhou, "A globally and R-linearly hybrid HS and PRP method and its inexact version with applications", Numer. Math., vol. 104, pp. 561-572,2006.

[6] S. Sumit, N. Biprajit, "A modified form of conjugate direction for general nonlinear function and its convergence", Int. J. of comp. sci. and mathematics, vol.7, pp. 25-31, 2005.

[7] X. Yunhai, W. Chunjie, Y.W. Soon, " Norm descent conjugate gradient method for solving symmetric nonlinear equations", J. Glo. Optim, DOI 10.1007/s10898-014-0218-7, 2014.

[8] G. Yuan, X. Lu, Z. Wei " BFGS trust-region method for symmetric nonlinear equations", J. Comp. Appl., vol.230, pp.44-58, 2009.

[9] Sabi'u J, "Enhanced derivative-free conjugate gradient method for solving symmetric nonlinear equations", International Journal of Advances in Applied Sciences, vol. 5, p.1, 2016.

[10] J. sabi'u, U. Sanusi, " An efficient new conjugate gradient approach for solving symmetric nonlinear equations" , Asian Journal of Mathematics and Computer Research, vol.12, pp.34-43, 2016.

[11] M.Y. Waziri, J. Sabi'u, " A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations", International J. of mathematics and mathematical science, vol.2015, doi:10.1155/2015/961487.

[12] W.W. Hager, H. Zhang, " A New conjugate gradient Method with Guaranteed Descent and an efficient line search", SIAM J. Optim., vol.16, pp.170-192, 2005.

[13] M. Raydan, " The Barzilai and Borwein gradient method for the large uncostrained minimization problem", SIAM Journal on Optimization, vol.7, pp. 26-33, 1997.

[14] M.Y. Waziri, J. Sabiu, " An alternative conjugate gradient approach for large-scale symmetric nonlinear equations", vol.6(5): 855, 2016.

Jamilu Sabi'u
Northwest University Kano, Nigeria
*E-mail:*sabiujamilu@gmail.com

Received 5 April 2017
Accepted 20 April 2017