



A New Hybrid Approach for Solving Large-scale Monotone Nonlinear Equations

Jamilu Sabi'u^{1,2*}, Abdullah Shah¹,
Mohammed Yusuf Waziri³ & Muhammad Kabir Dauda⁴

¹Department of Mathematics, COMSATS University Islamabad, Pakistan

²Department of Mathematics, Northwest University Kano, Nigeria

³Department of Mathematical Sciences, Bayero University Kano, Nigeria

⁴Department of Mathematical Sciences, Kaduna State University, Nigeria

*E-mail: sabiujamilu@gmail.com

Abstract. In this paper, a new hybrid conjugate gradient method for solving monotone nonlinear equations is introduced. The scheme is a combination of the Fletcher-Reeves (FR) and Polak-Ribière-Polyak (PRP) conjugate gradient methods with the Solodov and Svaiter projection strategy. Using suitable assumptions, the global convergence of the scheme with monotone line search is provided. Lastly, a numerical experiment was used to enumerate the suitability of the proposed scheme for large-scale problems.

Keywords: *global convergence; line search; monotone equations; projection strategy.*

Mathematics Subject Classification 2010: 90C30; 90C26.

1 Introduction

Nonlinear problems are of great interest to engineers, physicists, mathematicians and many other scientists. The natural behavior of nonlinear systems is described in mathematics by a system nonlinear of equations. Consider the nonlinear system

$$g(x) = 0, \quad (1)$$

where the function $g : R^r \rightarrow R^n$ is a continuously differentiable. The system is described as an overdetermined system when $n > r$, and classified as underdetermined when $r > n$. Moreover, when $n = r$, then the system is referred to as a square nonlinear system. Specifically, the system is considered to be monotone whenever

$$(g(x) - g(s))^T (x - s) \geq 0, \quad \forall x, s \in R^r. \quad (2)$$

The purpose of this hybrid method is to provide an alternative approach for globally converging large-scale monotone equations with a smaller computational burden and less CPU time consumption without computing the

Jacobian matrix. Popular schemes for solving Eq. (1) are the Newton and quasi-Newton methods [2,4,13,14]. They use the iterate

$$x_{k+1} = x_k - (g'(x_k))^{-1} g(x_k), \quad k = 0, 1, 2, \dots \quad (3)$$

Interesting qualities of these schemes include rapid convergence and easy implementation. However, they are generally not much used for solving large-scale monotone equations because they have to compute and store the Jacobian matrix or its approximation at every iteration. Conjugate gradient (CG) methods are specifically attractive due to their smaller memory requirement for large-scale problems. Waziri and Sabi'u [9] developed a CG method for solving symmetric nonlinear equations. They demonstrated the global convergence for their method with the aid of a non-monotone line search. The numerical comparison indicated that their scheme is reliable and competent. In line with the literature, Cheng and Li [5] extended the prominent Hager and Zhang non-monotone line search method [10] to solve large-scale nonlinear systems. Furthermore, a significant number of conjugate gradient methods have been proposed in the literature, among others [1-3,6,8,16-19]. Comprehensive numerical results evidenced that each method is appropriate for large-scale problems. This paper presents three-term hybrid Polak-Ribière-Polyak (PRP) and Fletcher-Reeves (FR) CG methods for large-scale monotone equations. It is organized as follows: in Section 2 we give the details of our approach followed by the convergence result in Section 3. Numerical comparisons can be found in Section 4 and the conclusion is given in Section 5.

2 Details of the Proposed Scheme

Given a point x_0 , the CG method generates the following iterates:

$$x_k = x_{k-1} + \alpha_k d_{k-1}, \quad k = 1, 2, \dots \quad (4)$$

where the term d_k is the conjugate gradient direction and α_k is the step size to be computed using typical line search (Backtracking, Armijo or Wolfe line search, and others). Let $v_k = x_{k-1} + \alpha_k d_{k-1}$. The hyperplane

$$H_k = \{x \in R^r \mid (x - v_k)^T g(v_k) = 0\} \quad (5)$$

strictly separates x_k from the solution set of Eq. (1). From this fact Solodov and Svaiter in [7] denoted the next iterate x_k to be

$$x_k = x_{k-1} - \frac{g(v_k)^T (x_{k-1} - v_k)}{\|g(v_k)\|^2} g(z_k). \quad (6)$$

In the work of Zhang, *et al.* [14] the CG direction is given as

$$d_k = \begin{cases} -g_k, & \text{for } k = 0, \\ -g_k + \beta_k^{PRP} d_{k-1} - \theta_k^{PRP} y_{k-1}, & \text{for } k \geq 1, \end{cases} \quad (7)$$

where $\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}$ and $\theta_k^{PRP} = \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2}$. Motivated by the above ideas of Zhang *et al.* [14], we present a hybrid three-term CG direction as:

Recall that from [1], $\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$ and $\theta_k^{FR} = \frac{d_{k-1}^T y_{k-1}}{\|g_{k-1}\|^2}$. Now we define our rigorous direction to be

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{H*} d_{k-1} - \theta_k^{H*} y_{k-1}, & \text{if } k \geq 1, \end{cases} \quad (8)$$

where

$$\beta_k^{H*} = \frac{\min\{g_k^T y_{k-1}, \|g_k\|^2\}}{\|g_{k-1}\|^2}, \text{ and } \theta_k^{H*} = \frac{\min\{d_{k-1}^T y_{k-1}, g_k^T d_{k-1}\}}{\|g_{k-1}\|^2}. \quad (9)$$

For the computation of stepsize α_k , the derivative free line search used in [11] is an interesting idea. Let, $\alpha_k = \max\{s, rs, r^2s, \dots\}$, satisfying

$$-g(x_k + \alpha_k d_k)^T d_k \geq \omega \alpha_k \|g(x_k + \alpha_k d_k)\| \|d_k\|^2. \quad (9)$$

Algorithm

Step 1 : Given $x_0, r, r\omega \in (0,1)$, set $k = 0$ and $d_0 = -g_0$.

Step 2 : Verify the stopping conditions. If yes, then stop; else go to Step 3.

Step 3 : Use Eq. (8) to compute d_k .

Step 4 : Determine α_k by Eq. (10).

Step 5 : Compute $x_k = x_{k-1} - \frac{g(z_k)^T (x_{k-1} - z_k)}{\|g(z_k)\|^2} g(z_k)$.

Step 6 : Allow $k = k + 1$ and repeat step 2.

3 Convergence Result

For our algorithm to converge globally, we assume that g is Lipschitz continuous and for $L_1, L_2 > 0$ we have

$$\|g(x)\| \leq L_1 \text{ and } \|g(v)\| \leq L_2. \quad (11)$$

Lemma 3.1 Assume $g(x^*) = 0$ and $\{x_k\}$ as obtained by our algorithm. We obtain $\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2$. Hence, $\{x_k\}$ is bounded and $\sum_{k=0}^{k=\infty} \|x_{k+1} - x_k\|^2 < \infty$.

Lemma 3.2 [8] Let $\{x_k\}$ and $\{v_k\}$ be computed by the proposed algorithm. We obtain

$$\alpha_k \geq \min \left\{ \beta_k^{H^*}, \frac{\delta\theta}{L^+, \omega} \frac{\|g_k\|^2}{\|g(v'_k)\| \|d_k\|^2} \right\} \quad (12)$$

Where $z' = x_k + \alpha'_k d_{k-1}$.

Theorem 3.3 Suppose $\{x_k\}$ is obtained by our algorithm. Let $s_k = x_k - x_{k-1}$. Then

$$\lim_{k \rightarrow \infty} \|\alpha_k d_k\| = \lim_{k \rightarrow \infty} \|s_k\| = 0, \quad (13)$$

Proof.

$$\|x_k - x_{k-1}\| = \frac{|g(v_k)^T (x_{k-1} - v_k)|}{\|g(v_k)\|} \leq \frac{\|g(v_k)\| \|x_{k-1} - v_k\|}{\|g(v_k)\|} \quad (14)$$

$$= \|x_k - v_k\| = \alpha_k \|d_k\|. \quad (15)$$

Theorem 3.4 Let $\{x_k\}$ be generated by our algorithm. Then

$$\liminf_{k \rightarrow \infty} \|g(x_k)\| = 0. \quad (16)$$

Proof. Assume Eq. (16) does not hold, i.e.

$$\|g(x_k)\| \geq \tau, \quad \forall k \geq 0. \quad (17)$$

Clearly, $\|g_k\| \leq \|d_k\|$, which implies

$$\|d_k\| \geq \tau, \quad \forall k \geq 0. \quad (18)$$

Observe that,

$$\|y_k\| = \|g(x_k) - g(x_{k-1})\| \leq L \|s_k\|, \quad (19)$$

from the definition of $\beta_k^{H^*}$ we have

$$|\beta_k^{H^*}| \leq \frac{\min\{L \|s_k\|, L_1\}}{L_1}, \quad (20)$$

Also, for $\theta_k^{H^*}$ we get

$$|\theta_k^{H^*}| \leq \frac{\min\{\tau L \|s_k\|, \tau L_1\}}{L_1}. \quad (21)$$

Therefore, we let the following inequalities be true for all $k \geq 0$. We obtain

$$\begin{aligned} \|d_k\| &\leq \|g_k\| + |\beta_k^{H^*}| \|d_{k-1}\| - |\theta_k^{H^*}| y_k \\ &\leq L_1 + \tau \frac{\min\{L \|s_k\|, L_1\}}{L_1} - \frac{\min\{\tau L \|s_k\|, \tau L_1\}}{L_1} \\ &\leq L_1 + \tau \frac{\min\{L \|s_k\|, L_1\}}{L_1}. \end{aligned} \quad (22)$$

Let $M^* = L_1 + \tau \frac{\min\{L \|s_k\|, L_1\}}{L_1}$. We get

$$\|d_k\| \leq M^*. \quad (23)$$

This combined with Lemma 3.2 and Eq. (11) and inequalities $\|d_k\| \geq \tau$, $\|g_k\| \geq \tau$ implies for all k sufficiently large,

$$\|d_k\| \alpha_k \geq \min\left\{\beta_k^{H^*}, \frac{\delta\theta}{L+\omega \|g(v'_k)\| \|d_k\|^2}\right\} \|d_k\| > 0. \quad (24)$$

The last inequality gives a contradiction with Eq. (17). Therefore, Eq. (16) is true and hence the proof.

4 Numerical Experiment

In this section, our algorithm is compared with the three-term Polak-Ribière-Polyak (TPRP) CG method [6]. For both algorithms we set $\omega = 10^{-4}$, $r = 0.2$ and $s = 1$. The code was implemented in Matlab 7.4 R2010a and run on a computer with 4 GB RAM memory capacity and a 1.8 GHz CPU processor. Iterations were stopped when the number of iterations was greater than 2000 or $\|g_k\| \leq 10^{-4}$. Eleven problems were tested to demonstrate the robustness of our algorithm compared to the TPRP method.

Problem 1

$$g_j(x) = e^{x_j} - 1, \quad j = 1, 2, \dots, n$$

Problem 2

$$g_1(x) = x_1(x_1^2 + x_2^2) - 1$$

$$g_j(x) = x_j(x_{j-1}^2 + 2x_j^2 + x_{j+1}^2) - 1, \quad j = 1, 2, \dots, n-1$$

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

Problem 3 For $j = 1, 2, \dots, n/3$,

$$g_{3j-2}(x) = x_{3j-2}x_{3j-1} - x_{3j}^2 - 1,$$

$$g_{3j-1}(x) = x_{3j-2}x_{3j-1}x_{3j} - x_{3j-2}^2 + x_{3j-1}^2 - 2,$$

$$g_{3j}(x) = e^{-x_{3j-2}} - e^{-x_{3j-1}}.$$

Problem 4 For $j = 2, 3, \dots, n-1$

$$g_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2)$$

$$g_j(x) = -x_{j-1}e^{x_{j-1}-x_j} + x_j(4 + 3x_j^2) + 2x_{j+1} + \sin(x_j - x_{j+1})\sin(x_j + x_{j+1}) - 8,$$

$$g_n(x) = -x_{n-1}e^{x_{n-1}-x_n} + 4x_n - 3.$$

Problem 5

$$g_1(x) = x_1 - e^{\frac{\cos(x_1+x_2)}{n+1}}$$

$$g_j(x) = x_j - e^{\frac{x_{j-1}+x_j+x_{j+1}}{n+1}}, \quad j = 2, 3, \dots, n-1$$

$$g_n(x) = x_n - e^{\frac{\cos(x_{n-1}+x_n)}{n+1}}.$$

Problem 6

$$g_1(x) = \cos(x_1) - 9 + 3x_1 + 8e^{x_2}$$

$$g_j(x) = \cos(x_j) - 9 + 3x_j + 8e^{x_{j-1}}, \quad j = 2, 3, \dots, n$$

Problem 7

$$g_j(x) = x_j^2 + (x_j - 3)\log(x_j + 3) - 9, \quad j = 1, 2, 3, \dots, n$$

Problem 8

$$g_j(x) = (x_j^2 - 1)^2 - 2, \quad j = 1, 2, 3, \dots, n$$

Problem 9

$$g_j(x) = 4x_j + (x_{j+1} - 2x_j) - \frac{x_{j+1}^2}{3}, \quad j = 1, 2, \dots, n-1$$

$$g_n(x) = 4x_n + (x_{n-1} - 2x_n) - \frac{x_{n-1}^2}{3}.$$

Problem 10

$$g_1(x) = \sin(x_1 - x_2) - 4e^{2-x_2+2x_1}$$

$$g_j(x) = \sin(2 - x_1) - 4ex_j - 2 + 2x_j + \cos(2 - x_1) - e^{(2-x_j)}, \quad j = 2, 3, \dots, n$$

Problem 11

$$g_j(x) = \sum_{j=1}^n (x_j - 2) + \cos(x_j - 2) - 1, \quad j = 1, 2, 3, \dots, n$$

From Figures 1 and 2 it can be observe that our proposed algorithm has the advantages of using less CPU time and less iterations over TPRP in almost all the eleven problems tested in our experiment using Dolan and Moré's performance profile [20].

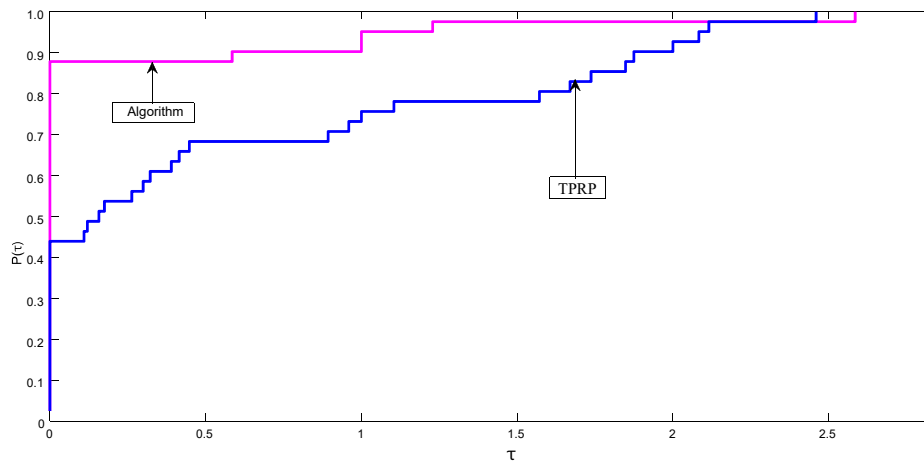


Figure 1 Performance profile of our algorithm versus the TPRP method [6] for number of iterations.

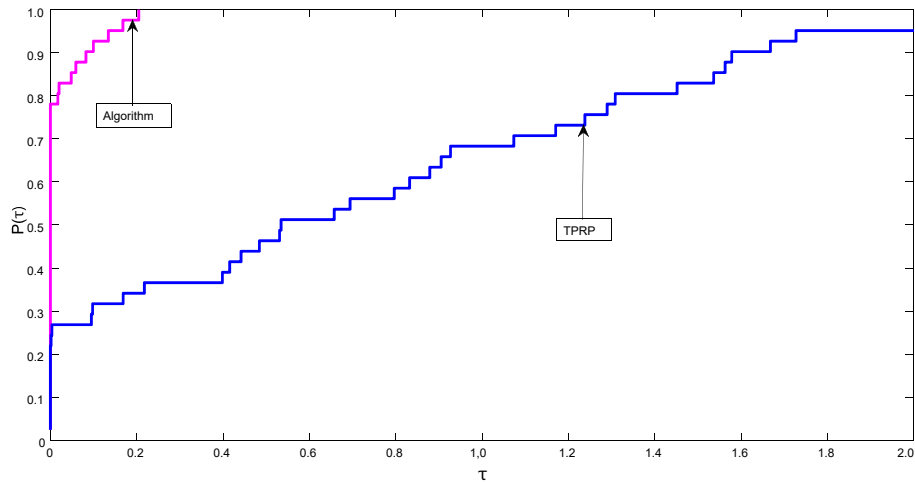


Figure 2 Performance profile of our algorithm versus the TPRP method [6] for CPU time in seconds.

5 Conclusion

This article proposes an effective new hybrid approach for solving monotone nonlinear equations with less iterations as well as less CPU time consumption without computing the Jacobian matrix. The numerical results show that this method is more efficient compared to TPRP [6]. The extension of this method to a more general system of smooth and nonsmooth equations will be investigated in a future research.

Acknowledgement

The first author is grateful to TWAS/CUI for the 2017 CUI-TWAS Full-time Postgraduate Fellowship with FR number 3240299486.

References

- [1] Zhang, L., Zhou, W. & Li, D., *Global Convergence of a Modified Fletcher-Reeves Conjugate Gradient Method with Armijo-Type Line Search*, *Numerische Mathematik*, **104**(4), pp. 561-572, 2006.
- [2] Birgin, E.G., Krejić, N. & Martínez, J.M., *Globally Convergent Inexact Quasi-Newton Methods for Solving Nonlinear Systems*, *Numerical algorithms*, **32**(2-4), pp. 249-260, 2003.
- [3] Barzilai, J. & Borwein, J.M., *Two-point Step Size Gradient Methods*, *IMA Journal of Numerical Analysis*, **8**(1), pp. 141-148, 1988.

- [4] Brown, P.N. & Saad, Y., *Convergence Theory of Nonlinear Newton–Krylov Algorithms*, SIAM Journal on Optimization, **4**(2), pp. 297-330, 1994.
- [5] Cheng, W. & Li, D.H., *A Derivative-free Nonmonotone Line Search and Its Application to the Spectral Residual Method*, IMA Journal of Numerical Analysis, **29**(3), pp. 814-825, 2008.
- [6] Yuan, G. & Zhang, M., *A Three-Terms Polak–Ribière–Polyak Conjugate Gradient Algorithm for Large-Scale Nonlinear Equations*, Journal of Computational and Applied Mathematics, **286**, pp. 186-195, 2015.
- [7] Solodov, M.V. & Svaiter, B.F., *A Globally Convergent Inexact Newton Method for Systems of Monotone Equations. In Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, Springer, Boston, MA, pp. 355-369, 1998.
- [8] Yan, Q.R., Peng, X.Z. & Li, D.H., *A Globally Convergent Derivative-Free Method for Solving Large-Scale Nonlinear Monotone Equations*, Journal of Computational and Applied Mathematics, **234**(3), pp. 649-657, 2010.
- [9] Waziri, M.Y. & Sabi’u, J., *A Derivative-free Conjugate Gradient Method and Its Global Convergence for Solving Symmetric Nonlinear Equations*, International Journal of Mathematics and Mathematical Sciences, Article ID 961487, pp. 1-8, 2015. DOI: 10.1155/2015/961487
- [10] Zhang, H. & Hager, W.W., *A Nonmonotone Line Search Technique and Its Application to Unconstrained Optimization*, SIAM Journal on Optimization, **14**(4), pp. 1043-1056, 2004.
- [11] Li, Q. & Li, D.H., *A Class of Derivative-free Methods for Large-Scale Nonlinear Monotone Equations*, IMA Journal of Numerical Analysis, **31**(4), pp. 1625-1635, 2011.
- [12] Sabi’u, J., *Enhanced Derivative-free Conjugate Gradient Method for Solving Symmetric Nonlinear Equations*, International Journal of Advances in Applied Sciences, **5**(1), pp. 50-57, 2016.
- [13] Zhou, W. & Li, D., *Limited Memory BFGS Method for Nonlinear Monotone Equations*, Journal of Computational Mathematics, **25**(1), pp. 89-96, 2007.
- [14] Zhang, L., Zhou, W. & Li, D.H., *Some Descent Three-term Conjugate Gradient Methods and Their Global Convergence*, Optimization Methods and Software, **22**(4), pp. 697-711, 2007.
- [15] Zhang, L., Zhou, W. & Li, D.H., *A Descent Modified Polak–Ribière–Polyak Conjugate Gradient Method and Its Global Convergence*, IMA Journal of Numerical Analysis, **26**(4), pp. 629-640, 2006.
- [16] Waziri, M.Y., Ahmed, K. & Sabi’u, J., *A Family of Hager–Zhang Conjugate Gradient Methods for System of Monotone Nonlinear Equations*, Applied Mathematics and Computation, **361**, pp. 645-660, 2019.

- [17] Waziri, M.Y., Ahmed, K. & Sabi'u, J., *A Dai–Liao Conjugate Gradient Method via Modified Secant Equation for System of Nonlinear Equations*, *Arabian Journal of Mathematics*, **2019**, pp. 1-15, 2019.
- [18] Sabi'u, J. & Gadu, A.M., *A Projected Hybrid Conjugate Gradient Method for Solving Large-Scale System of Nonlinear Equations*, *Malaysian Journal of Computing and Applied Mathematics*, **1(2)**, pp. 10-20, 2018.
- [19] Abubakar, A.B. & Kumam, P., *A Descent Dai-Liao Conjugate Gradient Method for Nonlinear Equations*, *Numerical Algorithms*, **81(1)**, pp. 197-210, 2019.
- [20] Dolan, E.D. & Moré, J.J., *Benchmarking Optimization Software with Performance Profiles*, *Mathematical programming*, **91(2)**, pp. 201-213, 2002.