International Journal of Contemporary Mathematical Sciences
Vol. 15, 2020, no. 4, 181 - 189
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ijcms.2020.91445

Numerical Performance of a Descent Conjugate Gradient Algorithm

Idowu A. Osinuga and Yvonne N. Nwodo

Department of Mathematics, College of Physical Sciences
Federal University of Agriculture, PMB 2240, Abeokuta, Nigeria

Abstract

We present a modification of the conjugate gradient (CG) algorithm. The proposed method is based on adapting the approach of a three-term CG hybridized linearly with a spectral-scaling memoryless BFGS update. Our numerical experiments indicate that our proposed method is competitive and robust.

Mathematics Subject Classifications 90C26

Keywords: Unconstrained optimization, conjugate gradient algorithm, memoryless BFGS, Wolfe line search

1. Introduction

The conjugate gradient methods remain one of the many tools for the solution of unconstrained optimization due to their simplicity and low memory storage. They remain well known for engineers and mathematicians (see Bamigbola et al. [9], Moyi et al. [24] and Navon and Ledger [25]) encountered with large-scale problems of the form:

$$\min\{ f(x) : x \in \Re^n \}, \tag{1}$$

where $f : \Re^n \to \Re$ is continuously differentiable and its gradient is available. The iterates of the classical CG algorithm can be formulated as

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

and

$$d_{k+1} = -g_{k+1} + \beta_k d_k, d_0 = -g_0 \tag{3}$$

where g_k is the gradient of $f(x)$ at the point x_k, d_k is the search direction, β_k is the so-called conjugate parameter, and α_k is the positive scalar step-size which is determined by some line search. The most well known conjugate gradient algorithms are the Hestenes and Stiefel (HS) [17], Fletcher and Reeves (FR) [14], Polak-Ribiere- Polyak (PRP) [26, 27], Liu and Storey (LS) [23], Dai and Yuan (DY) [13], and Fletcher (CD) [15], respectively while the corresponding parameters β_k are

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2},$$
$$\beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T y_k}, \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}, \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T y_k} \quad (4)$$

where $y_{k+1} = g_k - g_{k+1}$ and $\| \ \|$ denotes Euclidean norm of vectors. The rest of the paper is organized as follows. Section 2 is devoted to a brief literature review that describes the evolution of the nonlinear CG methods and the development of hybrids. In Section 3, after a brief introduction of scaled three-term CG proposed by Arzuka et al. (STCG) [7], a modified scaled three-term CG is proposed. A corresponding algorithm together with descent properties is given without any line search. Section 4 reported preliminary numerical results under standard Wolfe line search. Finally, Section 5 presents our concluding remarks.

2. Review of related works

It is well known that the choice of β_k affect the numerical performance of the method, hence many researchers studied choices of β_k. According to Andrei [2], the CG algorithms, based on β_k computation, can be classified as classical, hybrid, scaled and parametric. The classical algorithms are defined by (2) and (3), where the CG parameter is computed as in (4). Modified classical algorithms are abound in literature, to mention a few, consult (Adeleke and Osinuga [1], Andrei [2], Hager and Zhang [16], Gilbert and Nocedal [18], Taqi [28] and Touati-Ahmed and Storey [29]) and references therein. Hybrids have been derived to exploit the exciting features of the classical algorithms using projections (Adeleke and Osinuga [1], Andrei [3, 4], Taqi [28] and Zhang et al. [32]), consideration of linear and convex combination of classical schemes (Andrei [6], Babaie- Kafaki and Ghanbari [8], Liu et al. [21], Li and Li [22], Xu and Kong [31], Zhang et al. [32-34]) as well the use of notion involving the classical CG and quasi-Newton methods which started with Buckley [12]. Several others in this category can be found in (Ibrahim et al. [19-20] and Wan Osman et al. [30]). The other CG methods classified as scaled and parametric can be found in Liu et al. [21], Zhang et al. [32] and Arzuka et al. [7].

The development of three term classical CG started with Beale [10] and followed by Babaie-Kafaki [8], Taqi [28] and references therein, to mention a

few. Nevertheless, applications of the three-term classical CG and hybrids are many in the field of sciences and engineering. For more details, consult Bamigbola et al. [9], Taqi [28] and Moyi et al. [24], etc. Another innovation to the so-called three-term CG method is the case in which the search direction is determined as a linear combination of g_k, s_k and y_k as reported in Arzuka et al. [7], Zhang et al. [33-34] among others. Herein, a descent spectral-scaling three-term (DCG) method is proposed by incorporating the BFGS updating scheme of the inverse Hessian approximation within the frame of a memoryless quasi-Newton approach. In this case, the inverse Hessian approximation is restarted as a multiple of the identity matrix with a spectral scaling parameter in every iteration.

3. Motivation and the proposed CG algorithm

This section present the idea proposed in the DCG method. Recall the STCG due to Arzuka et al. [7], where the BFGS update is restarted with a multiple of the identity matrix with a positive scaling parameter as

$$Q_{k+1} = \mu_k I + \frac{s_k s_k^T}{s_k^T y_k} - \mu_k \frac{y_k y_k^T}{y_k y_k^T} \tag{5}$$

and thus, the search direction is given by

$$d_0 = -g_{0,}$$
$$d_{k+1} = -Q_{k+1} g_{k+1}, \text{ k = 0, 1, 2,…} \tag{6}$$

Motivated by the approaches of authors (Andrei [6], Azurka et al. [7] and Zhang et al. [34]) and based on the concept of memoryless BFGS update where at each iteration the inverse Hessian approximation is restarted as a multiple of the identity matrix with a spectral scaling parameter adapted from [21], we propose our method as

$$Q_{k+1} = \theta_k I - \theta_k \frac{s_k s_k^T}{s_k^T s_k} + \frac{y_k y_k^T}{y_k s_k^T} \tag{7}$$

and thus, the search direction is defined by

$$d_{k+1} = -Q_{k+1} g_{k+1} = -\theta_k g_{k+1} - \theta_k \frac{g_{k+1} s_k s_k^T}{s_k^T s_k} + \frac{y_k g_{k+1} y_k^T}{y_k s_k^T} \tag{8}$$

The proposed method new search direction is defined by

$$d_{k+1} = -\theta_k g_{k+1} - t_1 s_k + t_2 y_k, k = 0, 1, 2,\ldots, \quad d_0 = g_0 \tag{9}$$

where $\theta_k = \frac{d_{k-1}^T y_{k-1}}{\|g_{k-1}\|^2}, t_1 = \theta_k \frac{g_{k+1} s_k^T}{s_k^T s_k}, t_2 = \frac{g_{k+1} y_k}{s_k^T y_k}$ (10)

It is obvious to see from (9) that $d_{k+1}^T g_k < 0$. Similar to Theorem 2.1 in (Zhang et al. [33], pp 632), we state the following theorem without prove.

Theorem 3.1: Let d_{k+1} be defined by (9). The d_{k+1} is a descent function of f at x_k.

Based on Theorem 3.1, we present specific algorithm for DCG method, as follows.

Algorithm 3.2 (DCG algorithm).

Step 0: Give the initial point $x_0 \in \Re^n$ and set $\sigma, \varepsilon, \delta > 0$. Set $k = 0$.

Step 1: If $\|g_k\| \leq \varepsilon$, stop.

Step 2: Determine $\alpha > 0$ using the standard Wolfe line search:

$$f(x_k + \alpha_k d_d) \leq f(x_x) + \delta \alpha_k g_k^T d_k, \tag{11}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{12}$$

where $0 < \delta < \sigma < 1$.

Step 3: Let the next iterate be $x_{k+1} = x_k + \alpha_k \nu_k d_k$, where ν_k are computed according to ([26], page 4) else $x_{k+1} = x_k + \alpha_k d_k$

Step 4: Generate the next direction d_{k+1} by (9) where θ_k, t_1 and t_2 are computed by (10).

Step 5: Let $k := k+1$, go to step1.

Obviously, in the DCG method, the relation $d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2$ still holds. The following corollary is an immediate consequence of Theorem 3.1 and shows that the search direction satisfies the sufficient descent condition.

Corollary 3.3. Let d_{k+1} be given by (9) and (10). Then, for any $k \geq 0$, the following equality holds.

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2$$

4. Numerical Experiments

In this section, we report numerical results to study the performance of Algorithm 3.1compared to others in the solution of unconstrained optimization problems. The test functions used in our computational study were from the CUTE library Bongart et al. [11] and Andrei [5]. We tested 5 large-scale problems 4 times in extended or generalized form for a gradually increasing number of variables: $n = 800$, 1000, 5000, … , 10000. The parameters such as number of iterations (NI) and CPU time in seconds (CPU) were considered to evaluate the computational capability of DCG as compared with the STCG [7], TTCG [6] and TTPRP [34] methods. For each test problem, the stopping rule applied was $\| g_k \| \leq 10^{-6}$ and the number of iterations exceeds a limit of 10,000. All problems

implement the standard Wolfe line search with $\sigma = 0.9$ and $\delta = 0.0001$ using MATLAB R2013 with CPU 1.30 GHz and 3.00GB RAM, on SAMSUNG PC notebook. A failure is reported (denoted by 'F') in the tabulated results.

4.1 Discussion of results

Numerical performance of all the four algorithms is reported in Table 1. From the results in Table 1, it follows that DCG, STCG and TTPRP methods are competitive and solved all tested problems. The competitiveness is noticeable between DCG and STCG and includes TTPRP only in the problem 1. However, STCG outperforms in almost all cases especially at higher dimensions based on the number of iterations and CPU time in the first four problems while DCG seems to more efficiently solve medium sized problems with dimensions $500 \leq \dim \leq 1000$ and few large-scale problems. Note that TTCG recorded failures in almost all the problems but perform favourably and competitively in the solutions of problems 1, 3 and 5.

Conclusions

The contributions of this work are in two folds. The first is incorporating the BFGS updating scheme of the inverse Hessian approximation within the frame of a memoryless quasi-Newton approach. The other is to propose a spectral scaling parameter in the hybridized three-term CG formula such that the search directions always possess descent property independent of the line search technique. Numerical tests have indicated that our algorithm is competitive and robust compared to others by less number of iterations or CPU time based on some tested problems and otherwise in others. In future research, due to its competitiveness and robustness the proposed method is deemed promising and can still do better in solving medium and large-scale optimization problems from many fields of sciences and engineering.

References

[1] O. J. Adeleke and I. A. Osinuga, A five-term hybrid conjugate gradient method with global convergence and descent properties for unconstrained optimization problems, *Asian Journal of Scientific Research*, **11** (2018), 185-194, https://doi.org/10.3923/ajsr.2018.

[2] N. Andrei, Numerical comparison of conjugate algorithms in unconstrained optimization, *Studies in Informatics and Control*, **16** (2007), 333-352. https://doi.org/10.24846/v22i4y201302

[3] N. Andrei, Hybrid conjugate gradient algorithm for unconstrained optimization, *Journal of Optimization Theory Applications*, **141** (2) (2009), 249 – 264. https://doi.org/10.1007/s10957-008-9505-0

[4] N. Andrei, Accelerated hybrid conjugate gradient algorithm with modified secant condition for unconstrained optimization, *Numerical Algorithms*, **54** (1) (2010), 23 – 46. https://doi.org/10.1007/s11075-009-9321-0

[5] N. Andrei, An unconstrained optimization test functions collection, *Advanced Modelling and Optimization*, **10** (1) (2008), 147 - 161.

[6] N. Andrei, On three-term conjugate gradient algorithm for unconstrained Optimization, *Applied Mathematics and Computation*, **219** (11) (2013), 6316 – 6327. https://doi.org/10.1016/j.amc.2012.11.097

[7] I. Arzuka, M. R. Abubakar and W. J. Leong, A scaled three-term conjugate gradient method for unconstrained optimization, *Journal of inequalities and applications*, **325** (2016). https://doi.org/10.1186/s13660-016-1239-1

[8] S. Babaie – Kafaki and R. Ghanbari, An extended three - term conjugate gradient method with sufficient descent property, *Miskole Mathematics Notes*, **16** (1) 2015, 45 – 55. https://doi.org/10.18514/mmn.2015.1266

[9] O. M. Bamigbola, M. M. Ali and M. O. Oke, Mathematical modeling of electric power flow and the minimization of power losses on transmission lines, *Applied Mathematics and Computation*, **241** (2014), 214 – 221. https://doi.org/10.1016/j.amc.2014.05.039

[10] E. M. I. Beale, A derivative of conjugate gradients in numerical methods for nonlinear optimization, F.A. Lootsma, Ed., (1972), 39 – 43, Academic Press, London, UK.

[11] I. Bongart, A. R. Conn, N. I. M. Gould and P. L. Toint, CUTE: constrained and unconstrained testing environments, *ACM TOMS*, **21** (1995), 123-160. https://doi.org/10.1145/200979.201043

[12] A. G. Buckley, A combined conjugate-gradient quasi-Newton minimization algorithm, *Mathematical Programming*, **15** (1978), 200 – 210. https://doi.org/10.1007/bf01609018

[13] Y. H. Dai and Y. Yuan, An efficient hybrid conjugate gradient for unconstrained optimization, *Annals of Operations Research*, **103** (2001), 33 – 47. https://doi.org/10.1023/a:1012930416777

[14] R. Fletcher and C.Reeves, Function minimization by conjugate gradients, *Computer Journal*, **7** (1964), 149-154. https://doi.org/10.1093/comjnl/7.2.149

[15] R. Fletcher, *Practical Methods of Optimization vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, (1987).

[16] W. W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific Journal of Optimization*, **2** (2006), 35 – 58.

[17] M. R. Hestenes and E. L. Stiefel, Method of conjugate gradients for solving linear Systems, *Journal of Research National Bureau Standards*, **49** (1952), 409-436. https://doi.org/10.6028/jres.049.044

[18] E. G. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for Optimization, *SIAM Journal of Optimization*, **2** (1992), 21 - 42. https://doi.org/10.1137/0802003

[19] M. A. H. Ibrahim, M. Mustafa and W. J. Leong, The hybrid BFGS - CG Method in solving unconstrained optimization problems, *Abstract and Applied Analysis*, **2014** (2014), article ID 507102, 6 pp.
https://doi.org/10.1155/2014/507102

[20] M. A. H. Ibrahim, M. Mustafa, W. J. Leong and Z. S. Azfi, The Algorithms of Broyden – CG for unconstrained optimization problems, *International Journal of Mathematical Analysis*, **8** (52) 2014, 2591 – 2600.
https://doi.org/10.12988/ijma.2014.49272

[21] J. K. Liu, Y. M. Feng and L. M. Zou, A spectral conjugate gradient method for solving large-scale unconstrained optimization, *Computers and Mathematics with Applications*, **77** (2019), 731-739.
https://doi.org/10.1016/j.camwa.2018.10.002

[22] J. K. Li and S. J. Li, New hybrid conjugate Gradient Method for unconstrained optimization, *Applied Mathematics Computation*, **245** (2014), 36 – 43. https://doi.org/10.1016/j.amc.2014.07.096

[23] Y. Liu and C.Storey, Efficientgeneralizedconjugategradientalgorithms, Part1: Theory, *Journal of Optimization Theory Application*, **69** (1991), 129-137.
https://doi.org/10.1007/bf00940464

[24] A. U. Moyi, W. J. Leong and S. Ibrahim, On the application of three – term conjugate gradient method in regression analysis, *International Journal of Computer Applications*, **102** (8) (2014), 1 - 4. https://doi.org/10.5120/17832-8517

[25] M. Navon and D. N. Legler, Conjugate gradient methods for large-scale minimization in meteorology, *Monthly Weather Review*, **115** (1987), 1479 – 1502.
https://doi.org/10.1175/1520-0493(1987)115<1479:cgmfls>2.0.co;2

[26] E. Polak and G. Ribieré, Note sur la convergence de directions conjugeés, *Rev. Francaise Informat Recherche Operationelle*, 3e Année, **16** (1969), 35 - 43.
https://doi.org/10.1051/m2an/196903r100351

[27] B. T. Polyak, The conjugate gradient method in extreme problems,*USSR Comp. Mathematics and Mathematical Physics,* **9** (1969), 94-112. https://doi.org/10.1016/0041-5553(69)90035-4

[28] A. H. Taqi, Improved three - term conjugate algorithm for training neural network, *Journal of Kufa for Mathematics and Computer,* **2** (3) (2015), 93 – 100.

[29] D. Touati - Ahmed and C. Storey, Efficient hybrid conjugate gradient techniques, *Journal Optimization Theory and Applications*, **64** (2) (1990), 379 – 397. https://doi.org/10.1007/bf00939455

[30] W. F. H. Wan Osman, M. A. H. Ibrahim and M. Mamat, Hybrid DFP-CG method for solving Unconstrained optimization problems, *Journal of Physics: Conf. Series,* (2017), 890. https://doi.org/10.1088/1742-6596/890/1/012033.

[31] X. Xu and F. Kong, New hybrid conjugate gradient methods with the generalized Wolfe line search, *SpringerPlus*, **5** (2016), 881. https://doi.org/10.1186/s40064-016-2522-9

[32] L. Zhang, W. Zhou and D. Li, Global convergence of a modified FR conjugate gradient method with Armidjo-type line search, *Numeriche Mathematik,* **104** (2006), 561 – 572. https://doi.org/10.1007/s00211-006-0028-z

[33] L. Zhang, W. Zhou and D. Li, A descent modified Polak – Ribiere – Polyak conjugate Method and its global convergence, *IMA Journal Numerical Analysis*, **26** (2006), 629 – 640. https://doi.org/10.1093/imanum/drl016

[34] L. Zhang, W. Zhou and D. Li, A descent modified Polak – Ribiere – Polyak conjugate method and its global convergence, *Optimization Methods & Software*, **22** (2007), 697 – 711. https://doi.org/10.1080/10556780701223293

Appendix

The following functions were used in our computational study.

1. Extended Rosenbrock function, [5]:

$$f(x)=\sum_{i=1}^{n/2} c(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2, x_0=[-1,...,1], c=100$$

2. Extended QP1 function, [5]:

$$f(x)=\sum_{i=1}^{n-1}(x_i^2-2)^2+(\sum_{i=1}^{n}x_i^2-0.5)^2, x_0=[1,...,1]$$

3. Extended Himmelblau function, Bongart et al. [11]:

$$f(x)=\sum_{i=1}^{n/2}(x_{2i-1}^2+x_{2i}-11)^2+(x_{2i-1}+x_{2i}^2-7)^2, x_0=[1,...,1]$$

4. Diagonal 5 function [5]: $f(x)=\sum_{i=1}^{n}\log(\exp(x_i)+\exp(-x_i), x_0=[1,...,1]$

5. Raydan 1function [5]: $f(x)=\sum_{i=1}^{n}\frac{i}{10}(\exp(x_i)-x_i), x_0=[1,...,1]$

Table 1. Performance of the methods

Test functions	Dim	**DCG** *NI / CPU*	**STCG** *NI / CPU*	**TTCG** *NI / CPU*	**TTPRP** *NI / CPU*
1	863	**8**/0.424588	**5**/0.075388	F	**5**/0.093623
	1000	**7**/0.037437	**15**/0.084508	**7**/0.035623	**7**/0.076455
	6500	**6**/0.116492	**4**/0.102176	**8**/0.049671	**7**/0.104759
	11400	**18**/0.198895	**6**/0.125843	F	**6**/0.120379
2	863	**8**/0.267298	**11**/0.419386	F	**25**/0.247796
	1000	**8**/0.211096	**12**/0.372051	F	**22**/0.219962
	6500	**10**/0.263113	**6**/0.288581	F	**12**/0.281836
	11400	**10**/0.301597	**5**/0.284948	F	**12**/0.326839
3	863	**8**/0.207479	**12**/0.234796	**6**/0.057461	**46**/0.116223
	1000	**8**/0.216312	**12**/0.226462	**6**/0.052752	**37**/0.097757
	6500	**10**/0.30055	**5**/0.322132	F	**21**/0.143006
	11400	**10**/0.368419	**5**/0.304715	F	**54**/0.381402
4	863	**12**/0.090473	**5**/0.08349	F	**57**/0.145593
	1000	**10**/0.090974	**16**/0.102955	F	**46**/0.151859
	6500	**11**/0.117782	**4**/0.11234	F	**28**/0.192908
	11400	**18**/0.184339	**5**/0.167173	F	**21**/0.218948
5	863	**4**/0.081658	**4**/0.07619	**4**/0.093857	**1999**/1.13287
	1000	**4**/0.081907	**4**/0.093341	**4**/0.09179	**1999**/1.37951
	6500	**4**/0.104291	**4**/0.096191	**4**/0.100891	**1999**/6.83202
	11400	**4**/0.117792	**4**/0.117524	**4**/0.116761	**1999**/11.9269

Received: June 3, 2020; Published: July 14, 2020