2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science

A New Cyclic Gradient Method Adapted to Large-Scale Linear Systems

Qinmeng Zou and Frédéric Magoulès
Mathematics in Interaction with Computer Science
CentraleSupélec, Université Paris-Saclay
Paris, France
frederic.magoules@hotmail.com

***Abstract*—This paper proposes a new gradient method to solve the large-scale problems. Theoretical analysis shows that the new method has finite termination property for two dimensions and converges R-linearly for any dimensions. Experimental results illustrate first the issue of parallel implementation. Then, the solution of a large-scale problem shows that the new method is better than the others, even competitive with the conjugate gradient method.**

***Keywords*-gradient methods; linear systems; large-scale problems; Barzilai-Borwein methods;**

I. Introduction

We are interested in investigating new gradient methods for the solution of linear system

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite (SPD) and $b \in \mathbb{R}^n$. This problem is equivalent to the minimization of a convex quadratic function

$$f(x) = \frac{1}{2}x^\intercal Ax - b^\intercal x. \tag{2}$$

Gradient methods generate a sequence of the form

$$x_{k+1} = x_k - \alpha_k g_k, \quad k = 0, 1, \ldots, \tag{3}$$

where $g_k = Ax_k - b$. It is well known that the steepest descent (SD) method [1] performs poor in most cases, where the steplength can be written as follows

$$\alpha_k^{\text{SD}} = \frac{g_k^\intercal g_k}{g_k^\intercal A g_k}. \tag{4}$$

The iterates generated tend to asymptotically alternate between two directions [2]. In contrast, the conjugate gradient (CG) method [3] is often the method of choice that will terminate in at most n iterations. It is very attractive because of its high efficiency and low storage requirement. Nonetheless, CG iteration depends strongly on the search of direction calculation, i.e., any derivation such as round-off errors can seriously degrade performance [4].

In the past several decades, a renewed interest for gradient methods has appeared since Barzilai and Borwein [5] proposed two efficient nonmonotone steplengths

$$\alpha_k^{\text{BB1}} = \frac{g_{k-1}^\intercal g_{k-1}}{g_{k-1}^\intercal A g_{k-1}}. \tag{5}$$

$$\alpha_k^{\text{BB2}} = \frac{g_{k-1}^\intercal A g_{k-1}}{g_{k-1}^\intercal A^2 g_{k-1}}. \tag{6}$$

The motivation consists in approximating the Hessian and imposing some quasi-Newton properties. Some theories and experiments have shown that BB methods have good performance and are competitive with CG methods when low accuracy is required or small perturbation exists [4]. The convergence has been proven by Raydan [6]. Furthermore, Friedlander et al. [7] provided a general framework under the name of "gradient method with retards" that SD and BB both belong to it, as well as several alternate methods proposed later [8], [9], [10].

Motivated by the two-dimensional finite termination property, Yuan [11] provided a somewhat complicated steplength

$$\alpha_k^{\text{Y}} = \frac{2}{\sqrt{\left(\frac{1}{\alpha_{k-1}^{\text{SD}}} - \frac{1}{\alpha_k^{\text{SD}}}\right)^2 + \frac{4g_k^\intercal g_k}{s_{k-1}^\intercal s_{k-1}}} + \frac{1}{\alpha_{k-1}^{\text{SD}}} + \frac{1}{\alpha_k^{\text{SD}}}}, \tag{7}$$

where $s_{k-1} = x_k - x_{k-1}$. He gave two algorithms and some variants were investigated further by Dai and Yuan [12]. Among these methods, the second variant (DY) is the most efficient one according to the experiments in [12], where iterates are generated of the form

$$\alpha_k^{\text{DY}} = \begin{cases} \alpha_k^{\text{SD}}, & k \bmod 4 < 2, \\ \alpha_k^{\text{Y}}, & \text{otherwise.} \end{cases} \tag{8}$$

In this paper, we address the properties of cyclic gradient methods, especially their parallel behavior. We propose a new algorithm based on the Yuan steplength, which has also the two-dimensional finite termination property. In the next section, we introduce the cyclic gradient methods and propose our new steplength. In Section III, we give the convergence results of the new method. Some numerical results are presented in Section IV. Finally, a concluding remark is shown in Section V.

II. Cyclic Gradient Methods

Friedlander et al. [7] proposed an ingenious framework that gives rise to a great number of potentially efficient algorithms. Firstly, assume that $m \in \mathbb{N}$ represents retard that allows to employ the information from previous iterations. Let

$$\bar{k} = \max\{0,\ k - m\}, \tag{9}$$

then a collection of possible choices of steplength can be set as follows

$$\alpha_k^{\text{GMR}} = \frac{g_{\tau(k)}^\intercal A^{\rho(k)} g_{\tau(k)}}{g_{\tau(k)}^\intercal A^{\rho(k)+1} g_{\tau(k)}}, \tag{10}$$

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00058

where

$$\tau(k) \in \left\{\bar{k},\ \bar{k}+1,\ \ldots,\ k-1,\ k\right\}, \tag{11}$$

and

$$\rho(k) \in \{q_1,\ \ldots,\ q_m\}, \quad q_j \geq 0, \tag{12}$$

where $k \in \mathbb{N}$. The next theorem summarizes the convergence result in [7].

Theorem 1 (Friedlander et al., 1999)**.** *Consider the linear system* (1) *with $A \in \mathbb{R}^{n\times n}$ is SPD and $b \in \mathbb{R}^n$, where $x_* = A^{-1}b$ is the exact solution. Consider the gradient method* (3) *being used to solve* (1) *and the steplength α_k given by* (10)*. Then the sequence $\{x_k\}$ converges to x_* starting from any point x_0.*

For a proof of the above theorem, see [7]. Incidentally, several potential algorithms were provided therein, including the first cyclic gradient method under the name of cyclic steepest descent (CSD) as suggested in [8], which can be summarized as follows

$$\alpha_k^{\text{CSD}} = \begin{cases} \alpha_k^{\text{SD}}, & k \bmod m = 0, \\ \alpha_{k-1}, & \text{otherwise.} \end{cases} \tag{13}$$

Notice that if we choose $\rho(k) = 0$ and $\tau(k) = \bar{k} + 1,\ \ldots,\ k-1,\ k$, then (10) becomes CSD method, which satisfies the Theorem 1. On the other hand, Dai [8] proposed a variant called cyclic Barzilai-Borwein (CBB) method. They suggested that

$$\alpha_k^{\text{CBB}} = \begin{cases} \alpha_k^{\text{BB1}}, & k \bmod m = 0, \\ \alpha_{k-1}, & \text{otherwise.} \end{cases} \tag{14}$$

Similarly, if we choose $\rho(k) = 0$ and $\tau(k) = \bar{k},\ \bar{k}+1, \ldots,\ k-1$, then (10) becomes CBB method.

Although these methods greatly speed up the convergence, their motivation is too straightforward to further accelerate the iterations, which relies on the nonmonotone property to search the whole space without sink into any lower subspace spanned by eigenvectors [4]. This allows to reduce the gradient components more or less in the same asymptotic rate [12].

The recent literature showed that Yuan steplength may lead to efficient algorithms [11], [12]. All methods therein have two-dimensional finite termination property, i.e., if (8) is applied to a linear system in two-dimensional space, then the algorithm will terminate in at most 3 iterations. In general, such property seems not attractive in practice. However, experiments showed that they perform well in higher dimensions and are competitive with BB methods for large-scale problems [12].

Inspired by the Yuan steplength, we suggest a simple way of modifying steepest descent model to a cyclic gradient method. Consider a steplength of the form

$$\alpha_k^{\text{YB}} = \begin{cases} \alpha_k^{\text{SD}}, & k \bmod 3 = 0 \text{ or } 2, \\ \alpha_k^{\text{Y}}, & k \bmod 3 = 1. \end{cases} \tag{15}$$

Here we modify the order of SD and Y compared to the original YB formula, which is useful for the development of the new algorithm. Apart from this change, (15) is indeed the second algorithm propose by the pioneering work of Yuan [11]. It keeps the two-dimensional finite termination property that performs as well as BB for large-scale problems and better for small-scale problems. We could introduce simply the cyclic behavior based on (15) of the form

$$\forall m \in \mathbb{N},\ \text{if } k \bmod (3+m) > 2,\ \text{then } \alpha_k = \alpha_{k-1}. \tag{16}$$

Besides, we find that De Asmundis et al. [13] gives an interesting view about the iterations of SD method, where the technique of alignment was proposed therein to force the gradients into one-dimensional subspace and avoid the zigzag pattern. Notice that the inverse of constant Rayleigh quotient such as SD and BB steplengths has the similar property. Thus, constant SD with retards can also give rise to the alignment behavior and keep the nonmonotone benefit. To achieve this goal, we need to impose a repeat time to the zigzag process. Meanwhile, we want to keep the process based on Yuan steplength in the first several iterations. These motivations lead to a new method of the form

$$\alpha_k^{\text{CY}} = \begin{cases} \alpha_k^{\text{Y}}, & k \bmod (l+m+2) = 1, \\ \alpha_k^{\text{SD}}, & k \bmod (l+m+2) < l+2, \\ \alpha_{k-1}, & \text{otherwise,} \end{cases} \tag{17}$$

where $l \geq 1$ and $m \geq 1$. Such formula seems complicated, but indeed easy to understand. There are three components consisting in (17): the first SD and Y are used to insure the finite termination property; the parameter l acting on the second part of SD is used to keep several zigzag iterations; finally, the retard term m induces alignment and provides nonmonotone behavior to leap from the lower subspace.

III. Convergence Analysis

By the invariance property under any orthogonal transformation, we can assume without loss of generality that

$$A = \text{diag}(\lambda_1,\ \ldots,\ \lambda_n), \tag{18}$$

where

$$1 = \lambda_1 \leq \cdots \leq \lambda_n. \tag{19}$$

We follow the convergence framework established by Dai [8] and adapt it to our method. Let

$$G(k,\mu) = \sum_{i=1}^{\mu} g_{i,k}^2, \tag{20}$$

where $g_{i,k}$ is the ith component of g_k. A preliminary property is defined as follows.

Definition 1 (Property A)**.** Suppose that matrix A has the form (18) with condition (19) holds. If $\exists \xi \in \mathbb{N}$, $\exists M_1, M_2 > 0$, such that $\forall \mu \in \{1,\ldots,n-1\}$, $\forall \epsilon > 0$, $\forall j \in \{0,\ldots,\min\{k,\xi\}-1\}$,

- $\lambda_1 \leq \alpha_k^{-1} \leq M_1$;
- if $G(k-j,\mu) \leq \epsilon$ and $g_{\mu+1,k-j}^2 \geq M_2\epsilon$, then $\alpha_k^{-1} \geq \frac{2}{3}\lambda_{\mu+1}$,

then the steplength α_k has Property A.

The convergence framework of Dai can be deduced from Property A, stated as follows.

Theorem 2 (Dai, 2003)**.** *Consider the linear system* (1) *with* $A \in \mathbb{R}^{n\times n}$ *of the form* (18) *and* $b \in \mathbb{R}^n$*. Consider the gradient method* (3) *being used to solve* (1)*. If the steplength* α_k *has Property A, then the sequence* $\{\|g_k\|\}$ *converges to 0 R-linearly for any starting point* x_0*.*

For a proof of the above theorem, see [8]. Many gradient methods have Property A as mentioned in [8], e.g., the gradient method with retards (10). Inspired by the demonstration therein, we now develop a convergence result for the CY method.

Theorem 3. *Consider the linear system* (1) *with* $A \in \mathbb{R}^{n\times n}$ *of the form* (18) *and* $b \in \mathbb{R}^n$*. Consider the gradient method* (3) *with steplength* (17) *being used to solve* (1)*. Then the steplength* α_k^{CY} *has Property A.*

Proof: Note that (17) has three alternate steplengths, whereas the SD updating process and the constant process using the last SD steplength both follow the framework (10), which has been proven to have the Property A [8]. Therefore, we only investigate the Yuan steplength.

Recall that Yuan steplength has the following property

$$\left(\frac{1}{\alpha_{k-1}^{\mathrm{SD}}}+\frac{1}{\alpha_k^{\mathrm{SD}}}\right)^{-1} < \alpha_k^{\mathrm{Y}} < \min\left\{\alpha_{k-1}^{\mathrm{SD}},\ \alpha_k^{\mathrm{SD}}\right\}, \quad (21)$$

which is given in [11]. Hence,

$$\lambda_1 \le \frac{1}{\alpha_k^{\mathrm{SD}}} < \frac{1}{\alpha_k^{\mathrm{Y}}} < \frac{1}{\alpha_{k-1}^{\mathrm{SD}}}+\frac{1}{\alpha_k^{\mathrm{SD}}} \le 2\lambda_n. \quad (22)$$

Then the first condition of Property A holds by setting $M_1 = 2\lambda_n$. For the second one, let $M_2 = 2$ and $\xi = 1$, which yields $j = 0$. Suppose that

$$G(k,\mu) \le \epsilon, \quad g_{\mu+1,k}^2 \ge M_2\epsilon, \quad (23)$$

for all $\mu \in \{1,\dots,n-1\}$, and $\epsilon > 0$. Hence, the inverse of Yuan steplength becomes

$$\begin{aligned}\frac{1}{\alpha_k^{\mathrm{Y}}} &> \frac{1}{\alpha_k^{\mathrm{SD}}} = \frac{g_k^{\mathsf{T}}Ag_k}{g_k^{\mathsf{T}}g_k} = \frac{\sum_{i=1}^n \lambda_i g_{i,k}^2}{\sum_{i=1}^n g_{i,k}^2}\\ &\ge \frac{\lambda_{\mu+1}\sum_{i=\mu+1}^n g_{i,k}^2}{\sum_{i=1}^{\mu} g_{i,k}^2 + \sum_{i=\mu+1}^n g_{i,k}^2}\\ &\ge \frac{\lambda_{\mu+1}}{\frac{\sum_{i=1}^{\mu} g_{i,k}^2}{g_{\mu+1,k}^2}+1}\\ &\ge \frac{\lambda_{\mu+1}}{\frac{\epsilon}{2\epsilon}+1} = \frac{2}{3}\lambda_{\mu+1}\end{aligned} \quad (24)$$

Hence, the second condition of Property A is satisfied, which completes the proof. ∎

IV. Numerical Results

We first address the issue of parallel implementation. The dot product is engaged in the computation of steplength, which is the major obstacle of parallelization. Here we have two strategies to realize this goal. Let A_i be the band matrix stored in the ith processor. The first one (Gather Algorithm, GA) is to gather the vector $q_i = A_i * g$ and execute dot product with global vectors, shown as follows

$\texttt{Allgatherv}(q,\ q_i)$
$\alpha = \texttt{Dot}(g,\ g)\ /\ \texttt{Dot}(g,\ q)$

Then, the second one (Reduce Algorithm, RA) consists in computing the dot product locally, shown as follows

$c_i = \texttt{Dot}(g_i,\ q_i)$
$\texttt{Allreduce}(c,\ c_i,\ \texttt{SUM})$
$\alpha = \texttt{Dot}(g,\ g)\ /\ c$

Besides, we can see that global gradient vector is used in each iteration that we must proceed another `Allgatherv` function to communicate with other processor. Let p be the number of processors. The two experiments are proceeded by Alinea [14] (see also, e.g., [15], [16], [17]) and JACK [18], [19] (see also, e.g., [20]) and results are illustrated in Figures 1 and 2. We can see that generally the results are not good because the first one imposes so much computation and communication load, while the second one causes indeed the problem of loss of precision. These problems exist in all projection methods and by now we have not yet managed to find a solution.

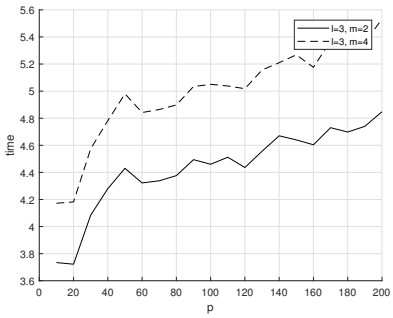

Figure 1. Parallel CY method with GA implementation

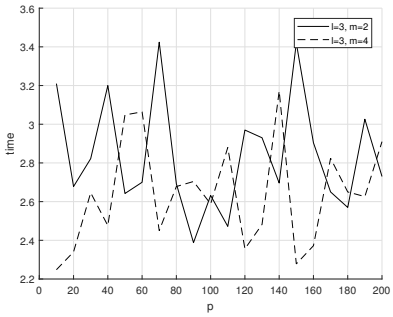

Figure 2. Parallel CY method with RA implementation

The second experiments are proceeded by Matlab R2017b with a large-scale problem provided by The SuiteSparse Matrix Collection [21], with $n = 50000$ and 349968 non-zero values. All parameters are chosen under

Table I
GRADIENT METHODS WITH DIFFERENT RESIDUAL THRESHOLD

	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
CG	58	735	2617	**4251**	**5786**	**7535**
CY	**13**	**208**	**1153**	**4275**	**6181**	\
CSD	27	212	**1470**	4357	6713	\
CBB	31	241	1965	7534	\	\
DY	**15**	**200**	1595	6415	\	\
BB1	356	984	2494	4612	8633	\
SD	61	5773	\	\	\	\

a training problem given in [5], such that $l = 4, m = 3$ for CY, $m = 3$ for CSD, and $m = 4$ for CBB. Average results are shown in Table I. We use bold numbers indicating the most efficient algorithms under each residual threshold. Backslash represents a number of iterations more than 10000. From Table I, we see that the CY method performs better than other methods except CG. Although we do not yet beat CG in high precision, our method is still competitive in most cases. Specifically, we can see that CY is stable throughout the iterations and much better than CG when low accuracy is required.

V. CONCLUSION

In this paper, we have proposed a new gradient method and shown that it is very competitive with CG method and better than the others for large-scale problems. However, it is still lack of theoretical evidence supporting such results. It is also important to find a better parallelization strategy. Therefore it still remains to study the properties and high performance implementation of gradient methods.

ACKNOWLEDGMENT

This work was supported by the French national programme LEFE/INSU and the project ADOM (Méthodes de décomposition de domaine asynchrones) of the French National Research Agency (ANR).

REFERENCES

[1] A. L. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1, pp. 536–538, 1847.

[2] H. Akaike, "On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method," *Annals of the Institute of Statistical Mathematics*, vol. 11, no. 1, pp. 1–16, 1959.

[3] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[4] R. Fletcher, "On the Barzilai-Borwein method," in *Optimization and Control with Applications*, L. Qi, K. Teo, and X. Yang, Eds. Boston, MA: Springer US, 2005, pp. 235–256.

[5] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[6] M. Raydan, "On the Barzilai and Borwein choice of steplength for the gradient method," *IMA Journal of Numerical Analysis*, vol. 13, no. 3, pp. 321–326, 1993.

[7] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, "Gradient method with retards and generalizations," *SIAM Journal on Numerical Analysis*, vol. 36, no. 1, pp. 275–289, 1999.

[8] Y.-H. Dai, "Alternate step gradient method," *Optimization*, vol. 52, no. 4-5, pp. 395–415, 2003.

[9] Y.-H. Dai and Y.-X. Yuan, "Alternate minimization gradient method," *IMA Journal of Numerical Analysis*, vol. 23, no. 3, pp. 377–393, 2003.

[10] B. Zhou, L. Gao, and Y.-H. Dai, "Gradient methods with adaptive step-sizes," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 69–86, 2006.

[11] Y.-X. Yuan, "A new stepsize for the steepest descent method," *Journal of Computational Mathematics*, vol. 24, no. 2, pp. 149–156, 2006.

[12] Y.-H. Dai and Y.-X. Yuan, "Analysis of monotone gradient methods," *Journal of Industrial and Management Optimization*, vol. 1, no. 2, pp. 181–192, 2005.

[13] R. De Asmundis, D. di Serafino, F. Riccio, and G. Toraldo, "On spectral properties of steepest descent methods," *IMA Journal of Numerical Analysis*, vol. 33, no. 4, pp. 1416–1435, 2013.

[14] F. Magoulès and A.-K. Cheik Ahamed, "Alinea: An advanced linear algebra library for massively parallel computations on graphics processing units," *The International Journal of High Performance Computing Applications*, vol. 29, no. 3, pp. 284–310, 2015.

[15] F. Magoulès, D. B. Szyld, and C. Venet, "Asynchronous optimized Schwarz methods with and without overlap," *Numerische Mathematik*, vol. 137, no. 1, pp. 199–227, 2017.

[16] F. Magoulès and C. Venet, "Asynchronous iterative substructuring methods," *Mathematics and Computers in Simulation*, vol. 145, pp. 34–49, 2018.

[17] F. Magoulès, G. Gbikpi-Benissan, and Q. Zou, "Asynchronous iterations of Parareal algorithm for option pricing models," *Mathematics*, vol. 6, no. 4, pp. 1–18, 2018.

[18] F. Magoulès and G. Gbikpi-Benissan, "JACK: an asynchronous communication kernel library for iterative algorithms," *The Journal of Supercomputing*, vol. 73, no. 8, pp. 3468–3487, 2017.

[19] F. Magoulès and G. Gbikpi-Benissan, "JACK2: An MPI-based communication library with non-blocking synchronization for asynchronous iterations," *Advances in Engineering Software*, vol. 119, pp. 116–133, 2018.

[20] F. Magoulès and G. Gbikpi-Benissan, "Distributed convergence detection based on global residual error under asynchronous iterations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 4, pp. 819–829, 2018.

[21] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Transactions on Mathematical Software*, vol. 38, no. 1, pp. 1:1–1:25, 2011.