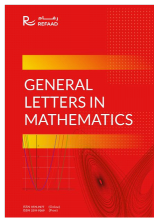
Available online at www.refaad.com
Gen.Lett. Math., 12 (2) (2022), 57-63
Research Article

General Letters in Mathematics (GLM)

Journal Homepage: https://www.refaad.com/Journal/Index/1
ISSN: 2519-9277 (Online) 2519-9269 (Print)



New search direction of steepest descent method for solving large linear systems

Ayad Ramadhan Ali[a,*], Bayda Ghanim Fathi[a]

[a] *Department of Mathematics, Faculty of Science, University of Zakho, Zakho, Kurdistan Region, Iraq.*

Abstract

The steepest descent (SD) method is well-known as the simplest method in optimization. In this paper, we propose a new SD search direction for solving system of linear equations $Ax = b$. We also prove that the proposed SD method with exact line search satisfies descent condition and possesses global convergence properties. This proposed method is motivated by previous work on the SD method by Zubai'ah-Mustafa-Rivaie-Ismail (ZMRI)[2]. Numerical comparisons with a classical SD algorithm and ZMRI algorithm show that this algorithm is very effective depending on the number of iterations (NOI) and CPU time.

Keywords: Steepest Descent Method, Positive Definite Matrices, Matrix Condition Number, and Systems of Linear Equations.

1. Introduction

The steepest-descent method (SDM), which can be traced back to Cauchy (1847)[8], is the simplest gradient method for solving positive definite linear equations systems. The iterative method for solving the system of algebraic equations can be derived from the discretization of certain ordinary differential equations (ODEs) systems [5]. In particular, some descent methods can be interpreted as the discretization of gradient flows [10]. Indeed, the continuous algorithms have been investigated in many literature works for a long time, for example, Hirsch and Smale [11] and Chu[9]. The Lyapunov methods used in the analysis of iterative methods have been made by Ortega and Rheinboldt [14] and Bhaya and Kaszkurewicz [5], [6] and [7].

Consider the linear system

$$Ax = b, \tag{1.1}$$

where $A \in R^{n\times n}$ is symmetric positive definite (SPD) matrix and $b \in R^{n\times 1}$. Since A is symmetric and positive definite, all of the eigenvalues are real and positive. Let the eigenvalues of the matrix A be given by $0 < \lambda_1 \leqslant \lambda_2 \ldots \leqslant \lambda_n$. The solution x_* is the unique global minimizer of a strictly convex quadratic function[13]

$$f(x) = \frac{1}{2}x^T Ax - b^T x. \tag{1.2}$$

*Corresponding author
Email addresses: ayad.ali@staff.uoz.edu.krd. (Ayad Ramadhan Ali), Bayda.fathi@uoz.edu.krd. (Bayda Ghanim Fathi)

doi:10.31559/glm2022.12.2.2

Received 16 Apr 2022 : Revised : 20 May 2022 Accepted: 28 May 2022

The gradient method is of the form[12]

$$x_{k+1} = x_k + \alpha_k p_k, \text{ for } k = 1, 2, 3, ... \quad (1.3)$$

where $p_k = -\bigtriangledown f(x_k)$ and $\bigtriangledown f(x_k) = Ax_k - b$. Cauchy defined the step-size by the reciprocal of a Rayleigh quotient of Hessian matrix A

$$\alpha_k^{SD} = -\frac{p_k^T g_k}{p_k^T A p_k}, \quad (1.4)$$

which is also called Cauchy step-size. It minimizes the function f or the $A - norm$ error and gives theoretically an optimal result in each step

$$\alpha_k^{SD} = \underset{\alpha}{argmin}\, f(x_k + \alpha p_k). \quad (1.5)$$

Andrei[4] stated that for functions very well-conditioned, the method proved to be effective but for functions poorly conditioned, it is excessively slow, thus being of no practical value. The gradient descent method with exact line search is known to behave increasingly badly when the condition number of the matrix deteriorates even for quadratic functions. Early attempts to increase the performance of the method have been considered by Schinzinger[16]. As we know, the direction of the negative gradient is the best direction of search for a minimizer of function $f(x)$ at the current point x_k , however as soon as we move in this direction, it ceases to be the best and continues to deteriorate until it becomes orthogonal to $-\bigtriangledown f(x_k)$. Without making significant progress to minimum, the method begins to take small steps. This is the major drawback of the gradient descent method. The steps it takes are too long and there are some other points on the line segment connecting x_k andx_{k+1} , which provides a better new search direction.

This classical method is known to behave badly in practice. The directions generated tend to asymptotically alternate between two directions leading to a slow convergence[3]. In particular, when the method of steepest descent with exact line searches is used on a strongly convex quadratic function, then one can show that:

$$f(x_{k+1}) - f(x_*) \leqslant [\frac{\eta(A) - 1}{\eta(A) + 1}]^2 f(x_k) - f(x_*), \quad (1.6)$$

Where $\eta(A)$ is the condition number of the matrix A and it is defined by $\eta(A) = \frac{\lambda_n}{\lambda_1}$. A similar bound can be derived for the case of a general nonlinear objective function, if we assume that α_k is the global minimizer along the search direction.

This paper is organized as follows: In section two, the new search direction of SD method is presented with its algorithm. In section three, we will prove the sufficient descent condition and analyze global convergence of the proposed method with exact line searches. In section four the numerical results and discussion are presented. In section five, we will give the conclusion.

2. New search direction and algorithm

Zubai'ah et al.[2] introduced a new search direction for SD method and proved that the new method is globally convergent under exact-line's search[1]. Their search direction is given as follows:

$$p_k^{ZMRI} = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k - \|g_k\| g_{k-1} & \text{if } k \geqslant 1. \end{cases} \quad (2.1)$$

Our objective in this section is to introduce a new search direction for SD method that has higher efficiency in solving systems of linear equations compared with current SD methods. This new search direction is motivated by Zubai'ah et al[2].

Hence, we suggest a new search direction for SD method stated as follows

$$p_k^{New} = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k - \dfrac{u}{\|g_{k-1}\|} g_{k-1} & \text{if } k \geqslant 1 \text{ and odd} \\ -g_k - \|g_k\| g_{k-1} & \text{if } k \geqslant 1 \text{ and even} \end{cases} \tag{2.2}$$

where $u = 0.001$.

The new algorithm based on (2.2) is derived as follows:

Algorithm 2.1:

Step (1): Given a symmetric positive definite matrix $A \in R^{n\times n}$, a vector $b \in R^{n\times 1}$, and an initial approximation of the solution vector $x_0 \in R^{n\times 1}$.

Step (2): Set $p_0 = -g_0$ where $g_0 = Ax_0 - b$.

Step (3): for $k = 1, 2, 3, ...$

Step (4): Compute $\alpha_{k-1} = -\dfrac{p_{k-1}^T g_{k-1}}{p_{k-1}^T A p_{k-1}}$.

Step (5): Update a new point $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$.

Step (6): Set $g_k = Ax_k - b$.

Step (7): Set if k is odd : $p_k = -g_k - \dfrac{u}{\|g_{k-1}\|} g_{k-1}$ else $p_k = -g_k - \|g_k\| g_{k-1}$.

Step (8): Until $\|g_k\|^2$ is zero or small enough.

The efficiency of the above algorithm is tested using some test problems.

3. Convergence analysis

The convergence analysis based on (2.2) has been discussed carefully in this section. In order to prove that an algorithm will converge, it must possess the sufficient descent and global convergence properties.

3.1. *Sufficient descent condition*

For the sufficient descent condition to hold,

$$g_k^T p_k \leqslant -c\|g_k\|^2, \text{ for } k \geqslant 0 \text{ and real number } c > 0. \tag{3.1}$$

Theorem 3.1: Consider an SD method with the search direction (2.2) and the step-size determined by exact procedure (1.5). Then condition (3.1) holds for all $k \geqslant 0$.

Proof:

case1: If $k = 0$, then $p_0 = -g_0$, therefore $g_0^T p_0 = -g_0^T g_0 = -\|g_0\|^2$. Hence, condition (3.1) holds true. Now we need to show that for $k \geqslant 1$, condition (3.1) will also hold true.

case2: if $k \geqslant 1$ and odd, then

$$p_k = -g_k - \frac{u}{\|g_{k-1}\|} g_{k-1}. \tag{3.2}$$

Multiply both sides by g_k and note that $g_k^T\ p_{k-1} = 0$ for exact-line's search and we get

$$g_k^T p_k = - \parallel g_k \parallel^2 - \frac{u}{\parallel g_{k-1} \parallel} g_k^T g_{k-1}.$$

From [15], $g_k^T g_{k-1} \geqslant \varepsilon \parallel g_k \parallel^2$ where ε is any real number in interval $(0,1]$, which implies that

$$g_k^T p_k \leqslant - \parallel g_k \parallel^2 - \frac{\varepsilon u \parallel g_k \parallel^2}{\parallel g_{k-1} \parallel}$$

$$g_k^T p_k \leqslant - \parallel g_k \parallel^2 (1 + \frac{\varepsilon u}{\parallel g_{k-1} \parallel}) \text{ with } \varepsilon \in (0,1] \text{ and } u = 0.001.$$

we can see that $c = (1 + \frac{\varepsilon u}{\parallel g_{k-1} \parallel})$ and $c > 0$. Hence condition (3.1) holds and the proof is complete, which implies that p_k is a sufficient descent direction for case 2.

case3: if $k \geqslant 1$ and even, then

$$p_k = -g_k - \|g_k\| g_{k-1}. \tag{3.3}$$

It was proved by Zubai'ah et al[1], which implies that p_k is a sufficient descent direction for case 3. Now, from all cases, we can say that our new direction p_k^{New} is a sufficient descent direction.

3.2. *Global convergence*

The following assumptions and lemma are needed in the analysis of global convergence of SD methods.

Assumption 3.1.

1. The level set $\Omega = \{x \in R^n | f(x) \leqslant f(x_0)\}$ is bounded where x_0 is the initial point.
2. In some neighborhood N of Ω , the objective function is continuously differentiable, and its gradient is Lipchitz continuous, namely, there exists a constant $l \geqslant 0$ such that $\parallel g(x) - g(y) \parallel \leqslant l \parallel x - y \parallel$ for any $x, y \in N$.

These assumptions yield the following Lemma 3.1, which was proven by Zoutendijk[17].

Lemma 3.1: Suppose that Assumption 3.1 holds true. Let x_k be generated by Algorithm 2.1 and p_k satisfies (3.1), then the following condition, known as Zoutendijk condition, holds.

$$\sum_{k=0}^{\infty} \frac{\parallel g_k \parallel^4}{\parallel p_k \parallel^2} < \infty \tag{3.4}$$

Theorem 3.2: Suppose that Assumption 3.1 holds true. Consider x_k generated by Algorithm 2.1, α_k is obtained by using exact line search and the sufficient descent condition is satisfied. Then, either

$$\lim_{k\to\infty} \parallel g_k \parallel = 0 \qquad \text{or} \qquad \sum_{k=0}^{\infty} \frac{(g_k^T p_k)^2}{\parallel p_k \parallel^2} < \infty. \tag{3.5}$$

Proof: The proof is done by contradiction. Assume that Theorem 3.2 is not true, that is, $\lim_{k\to\infty} \parallel g_k \parallel \neq 0$. Then there exists a positive constant $\delta > 0$, such that

$$\parallel g_k \parallel \geqslant \delta. \tag{3.6}$$

case1: if

$$p_k = -g_k - \frac{u}{\|g_{k-1}\|} g_{k-1}.$$

using Cauchy-Schwarz and taking into account that $g_k^T p_{k-1} = 0$, we get

$$\| p_k \| \leqslant \| g_k \| + \frac{u}{\|g_{k-1}\|} \| g_{k-1} \|$$

$$\| p_k \| \leqslant \| g_k \| + u$$

Dividing both sides by $\| g_k \|^2$ yields,

$$\frac{\| p_k \|}{\| g_k \|^2} \leqslant \frac{1}{\| g_k \|} + \frac{u}{\| g_k \|^2}$$

By using the assumption that $\| g_k \| \geqslant \delta.$, where $\delta > 0$, we now have

$$\frac{\| p_k \|}{\| g_k \|^2} \leqslant \frac{1}{\delta} + \frac{u}{\delta^2}$$

let $M = \frac{1}{\delta} + \frac{u}{\delta^2}$ and $u = 0.001$, then we have

$$\frac{\| p_k \|}{\| g_k \|^2} \leqslant M$$

and squaring both sides of the equation, we get

$$\frac{\| p_k \|^2}{\| g_k \|^4} \leqslant M^2$$

which implies,

$$\sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\| p_k \|^2} \geqslant \infty \tag{3.7}$$

This contradicts Zoutendijk condition in Lemma 3.1.
Therefore from (3.7), it follows that,

$$\sum_{k=0}^{\infty} \frac{(g_k^T p_k)^2}{\| p_k \|^2} < \infty.$$

Hence, the proof is completed.

case2: if

$$p_k = -g_k - \|g_k\| g_{k-1}.$$

It was proved by Zubai'ah et al[1].
Now, from both cases we can say that our new direction p_k^{New} is globally convergence.

4. Numerical Results

Algorithm 2.1 has been tested and compared its performance with the classical SD method algorithm and modified SD method algorithm developed by [2]. The comparison is on CPU time and the number of iterations (NOI) on each test system $Ax = b$ with symmetric positive definitive matrix A in different condition numbers $\eta(A)$. The stopping criteria used for all methods are $\rho = 10^{-6}$ and $\| g_k \| \leqslant \rho$. All programs are written in MatlabR2014a using with laptop Intel @Core^{TM} i5-4200H (2.8GHz with 12 GB (RAM)).

The performances are establishing on CPU time running in seconds and number of iterations (NOI) respectively. We generate randomly seven systems with dimensions (10, 100, 500, 1000, 2000, 3000, and 5000) and with matrix condition numbers $\eta(A)$ (10, 100, 500, 1000, and 5000), see results in table1.

Table 1: Comparison among SD algorithm, ZMRI algorithm, and new algorithm for solving linear systems.

Case	Matrix Size	Matrix condition number $\eta(A)$	SD		ZMRI		New	
			NOI	CPU Time	NOI	CPU Time	NOI	CPU Time
1	$A_{10\times10}$	10	35	0.0010436	32	0.0007958	23	0.00060152
		100	313	0.0075218	210	0.0036978	80	0.0016978
		500	943	0.018751	571	0.011751	169	0.0031751
		1000	3437	0.06325	656	0.01227	469	0.0062051
		5000	14485	0.13103	1192	0.014671	725	0.0073583
2	$A_{100\times100}$	10	30	0.0026909	29	0.0018163	29	0.0011334
		100	291	0.014815	240	0.0067418	186	0.0037418
		500	1573	0.068533	724	0.034908	183	0.0074067
		1000	3405	0.14171	1421	0.061691	479	0.019026
		5000	14259	0.43796	2426	0.074033	779	0.026509
3	$A_{500\times500}$	10	42	0.050296	45	0.043368	37	0.034829
		100	333	0.25678	116	0.13391	105	0.03391
		500	1533	1.4697	343	0.32752	243	0.22918
		1000	3165	2.997	849	0.8092	225	0.21462
		5000	14533	13.6554	3372	3.1536	1398	1.3053
4	$A_{1000\times1000}$	10	42	0.19093	44	0.20351	37	0.1659
		100	319	1.576	142	0.58148	103	0.18148
		500	1839	8.3568	517	2.2301	231	0.97955
		1000	3137	13.4073	1025	4.3871	421	1.7289
		5000	15099	63.3001	2536	10.6061	1371	5.696
5	$A_{2000\times2000}$	10	49	0.79766	53	0.86558	41	0.67094
		100	247	4.0881	105	1.7521	77	1.2752
		500	955	10.4921	445	7.3734	265	4.4801
		1000	2575	11.9833	715	11.8271	471	7.8004
		5000	13049	213.1214	3514	55.9763	1127	18.1484
6	$A_{3000\times3000}$	10	48	1.8114	55	1.9976	45	1.6774
		100	323	11.9365	126	4.7187	109	4.1223
		500	1695	16.8759	403	14.342	271	9.5594
		1000	2933	29.4613	1079	25.6866	523	18.6497
		5000	13881	504.4423	3042	107.7721	1536	54.2899
7	$A_{5000\times5000}$	10	51	5.2189	53	5.4195	47	4.8252
		100	317	32.6631	130	13.3464	121	12.8052
		500	1769	181.1542	789	80.7324	279	28.4313
		1000	2493	255.6515	433	44.5051	351	36.1508
		5000	15061	1472.68	1590	154.4684	1274	123.6482

Experimental results in table 1 confirm that the new SD algorithm (New) is superior to SD and ZMRI with respect to the number of iterations NOI and the elapsed CPU time, and it has shown that a new SD method has more efficiency and faster than the other SD methods for all tested systems.

5. Conclusion

We have presented a new search direction of steepest descent method for solving large systems of linear equations and we proved that this new search direction satisfied sufficient descent condition and globally convergence properties. The numerical results have shown that our proposed search direction substantially outperforms the other tested methods and is very efficient in solving all linear systems with symmetric positive definite matrix.

References

[1] Z. Z. Abidin, M. Mamat, and M. Rivaie, *A new steepest descent method with global convergence properties*, AIP Conference Proceedings, **1739**(1) (2016),020070, https://aip.scitation.org/doi/abs/10.1063/1.4952550. https://doi.org/10.1063/1.4952550 2, 3.1, 3.2

[2] Z. Z. Abidin, M. Mamat, M. Rivaie, and I. Mohd, *A new steepest descent method*, AIP Conference Proceedings, **1602**(1) (2014),020070, https://aip.scitation.org/doi/abs/10.1063/1.4882499. (document), 2, 2, 4

[3] 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, **11**(1) (1959), 1–16. https://doi.org/10.1007/bf01831719 1

[4] N. Andrei, *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*, Numerical Algorithms, **42**(1) (2006), 63–73,. https://doi.org/10.1007/s11075-006-9023-9 1

[5] A. Bhaya and E. Kaszkurewicz, *Control perspectives on numerical algorithms and matrix problems*, SIAM, (2006). 1

[6] A. Bhaya and E. Kaszkurewicz, *A control-theoretic approach to the design of zero finding numerical methods*, IEEE Transactions on Automatic Control, **52**(6) (2007), 1014–1026. https://doi.org/10.1109/tac.2007.899109 1

[7] A. Bhaya and E. Kaszkurewicz, *Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method*, Neural Networks, **17**(1) (2004), 65–71. https://doi.org/10.1016/s0893-6080(03)00170-9 1

[8] A. CAUCHY, *Méthode générale pour la résolution des systèms d'équations simultanées*, Comput. Rend. Sci. Paris, **25** (1847), 536–538. 1

[9] M. T. Chu, *On the continuous realization of iterative processes*, SIAM review, **30**(3) (1988), 375–387. https://doi.org/10.1137/1030090 1

[10] U. Helmke and J. B. Moore, *Optimization and dynamical systems*, Springer Science and Business Media, (2012). 1

[11] M. W. Hirsch and S. Smale, *On algorithms for solving f (x)= 0*, in The Collected Papers of Stephen Smale, **25**, World Scientific, (2000), 1076–1107. 1

[12] C. Lemaréchal, *Cauchy and the gradient method*, Doc Math Extra, **251**(254), (2012), 10. 1

[13] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*,2. Springer, (1984). 1

[14] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, SIAM, (2000). 1

[15] M. J. D. Powell, *Restart procedures for the conjugate gradient method*, Mathematical programming, **12**(1), (1977), 241–254. https://doi.org/10.1007/bf01593790 3.1

[16] R. Schinzinger, *Optimization in electromagnetic system design*, in Recent Advances in Optimization Techniques, Wiley, (1966), 163–213. 1

[17] G. Zoutendijk and J. Abadie, *Integer and Nonlinear programming computational methods*, North Holland Amsterdam, (1970), 37–86. 3.2