Journal of Computational Mathematics, Vol.24, No.2, 2006, 149–156.

A NEW STEPSIZE FOR THE STEEPEST DESCENT METHOD *1)

Ya-xiang Yuan
(*LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*)

Abstract

The steepest descent method is the simplest gradient method for optimization. It is well known that exact line searches along each steepest descent direction may converge very slowly. An important result was given by Barzilar and Borwein, which is proved to be superlinearly convergent for convex quadratic in two dimensional space, and performs quite well for high dimensional problems. The BB method is not monotone, thus it is not easy to be generalized for general nonlinear functions unless certain non-monotone techniques being applied. Therefore, it is very desirable to find stepsize formulae which enable fast convergence and possess the monotone property. Such a stepsize α_k for the steepest descent method is suggested in this paper. An algorithm with this new stepsize in even iterations and exact line search in odd iterations is proposed. Numerical results are presented, which confirm that the new method can find the exact solution within 3 iteration for two dimensional problems. The new method is very efficient for small scale problems. A modified version of the new method is also presented, where the new technique for selecting the stepsize is used after every two exact line searches. The modified algorithm is comparable to the Barzilar-Borwein method for large scale problems and better for small scale problems.

Mathematics subject classification: 65L05, 65F10.
Key words: Steepest descent, Line search, Unconstrained optimization, Convergence.

1. Introduction

The steepest descent method, which can be traced back to Cauchy (1847), is the simplest gradient method for unconstrained optimization:

$$\min_{x\in R^n} f(x), \tag{1.1}$$

where $f(x)$ is a continuous differential function in R^n. The method has the following form:

$$x_{k+1} = x_k + \alpha_k(-g_k), \tag{1.2}$$

where $g_k = g(x_k) = \nabla f(x_k)$ is the gradient vector of $f(x)$ at the current iterate point x_k and $\alpha_k > 0$ is the stepsize. Because the search direction in the method is the opposite of the gradient direction, it is the steepest descent direction locally, which gives the name of the method. Locally the steepest descent direction is the best direction in the sense that it reduces the objective function as much as possible.

The stepsize α_k can be obtained by exact line search:

$$\alpha_k^* = argmin\{f(x_k + \alpha(-g_k))\}, \tag{1.3}$$

or by some line search conditions, such as Goldstein conditions or Wolfe conditions (see Fletcher, 1987). It is easy to show that the steepest descent method is always convergent. That is, theoretically the method will not terminate unless a stationary point is found.

* Received September 30, 2004.
1) This work is partially supported by Chinese NSF grant 10231060.

However, even for the simplest case when the objective function $f(x)$ is a strictly convex quadratic, namely

$$f(x) = g^T x + \frac{1}{2} x^T H x, \tag{1.4}$$

where $g \in \Re^n$, $H \in \Re^{n \times n}$ symmetric and positive definite, the steepest descent method may not be very efficient. Assume that we are using exact line searches. Though we can show that the method converges linearly (see Akaike, 1959), the convergence rate can be very slow, especially when the condition number of the Hessian matrix H is very large, as the Q-linear fact of the convergence is

$$\frac{\lambda_1(H) - \lambda_n(H)}{\lambda_1(H) + \lambda_n(H)} \tag{1.5}$$

where $\lambda_1(H)$ and $\lambda_n(H)$ are the largest and smallest eigenvalues of H respectively. Forsythe(1968) gives an interesting analysis to show that the gradients $g(x_k)$ will approach zero eventually along two direction alternatively.

An surprising result was given by Barzilai and Borwein (1988), where gives formulae for the stepsize α_k which lead to superlinear convergence. The main idea of Barzilai and Borwein's approach is to use the information in the previous interation to decide the stepsize in the current iteration. The iteration (1.2) is viewed as

$$x_{k+1} = x_k - D_k g_k, \tag{1.6}$$

where $D_k = \alpha_k I$. In order to force the matrix D_k having certain quasi-Newton property, it is reasonable to require either

$$\min \|s_{k-1} - D_k y_{k-1}\|_2 \tag{1.7}$$

or

$$\min \|D_k^{-1} s_{k-1} - y_{k-1}\|_2, \tag{1.8}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$, because in a quasi-Newton method we have that $x_{k+1} = x_k - B_k^{-1} g_k$ and the quasi-Newton matrix B_k satisfies the condition

$$B_k s_{k-1} = y_{k-1}. \tag{1.9}$$

Now, from $D_k = \alpha_k I$ and relations (1.7)-(1.8) we can obtain two stepsizes:

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|_2^2}, \tag{1.10}$$

and

$$\alpha_k = \frac{\|s_{k-1}\|_2^2}{s_{k-1}^T y_{k-1}} \tag{1.11}$$

respectively. For convex quadratic function in two variables, Barzilar and Borwein (1988) shows that the gradient method (1.2) with α_k given by (1.10) converges R-superlinearly and R-order is $\sqrt{2}$.

The result of Barzila and Borwein(1988) has triggered off many researches on the steepest descent method. For example, see Dai(2001), Dai et al(2002), Dai and Yuan(2003), Dai and Zhang(2001), Fletcher (2001), Friedlander et al. (1999), Nocedal et al(2000) and Raydon(1993, 1997).

The BB method performs quite well for high dimensional problems. The BB method is not monotone, and it is not easy to generalized to general nonlinear functions unless certain non-monotone techniques being applied. Therefore, it is very desirable to find stepsize formula which enables fast convergence and possesses the monotone property.

This paper tries to propose such a stepsize α_k for the steepest descent method. Due to the results of Forsythe(1968), the behave of steepest descent method for higher dimensional problems are essential the same as it for two dimensional problems. Therefore we obtain our formula for α_k based our analysis on two dimension problems. The α_k we obtained has the property that it terminate after three iterations. In the next section we derive the new formula for α_k and give some equivalent conditions. Some numerical results are presented in Section 3 and a brief discussion is given in Section 4.

2. A New Stepsize

For the analysis of this section, we assume that the objective function is as follows

$$f(x) = g^T x + x^T H x \tag{2.1}$$

where $g \in \Re^2$ and $H \in \Re^{2\times 2}$ symmetric and positive definite. We want the method to find the unique minimizer of $f(x)$ in finitely many iterations. It is easy to see that exact line search must be taken in the last iteration before the algorithm stops at the solution. Without any pre-provided information, we assume that we also use the exact line search in the first iteration, as we do not want to through away the fortunate case when the algorithm can find the solution in the first iteration. Therefore, in general case, the best we can hope is to have an algorithm that has the following form

$$x_2 = x_1 - \alpha_1^* g_1 \tag{2.2}$$

$$x_3 = x_2 - \alpha_2 g_2 \tag{2.3}$$

$$x_4 = x_3 - \alpha_3^* g_3, \tag{2.4}$$

where α_1^* and α_3^* are obtained by exact line searches and x_4 is the solution. We need to find a formula for α_2 so that x_4 will be the minimizer of the objective function.

The steepest descent method is invariant with respect to orthogonal transformations. To make our analysis simple, we study the case when g_1 and g_2 are the two axes. Due to the exact line search in the first iteration, the gradients g_1 and g_2 are orthogonal. Therefore we can express all the vectors x by linear combinations of g_1 and g_2. Consider the function

$$f(x_2 + t\frac{g_1}{\|g_1\|_2} + u\frac{g_2}{\|g_2\|_2}) = \begin{pmatrix} 0 \\ \|g_2\|_2 \end{pmatrix}^T \begin{pmatrix} t \\ u \end{pmatrix} + \frac{1}{2}\begin{pmatrix} t \\ u \end{pmatrix}^T \begin{pmatrix} g_1^T H g_1/\|g_1\|_2^2 & g_1^T H g_2/\|g_1\|_2\|g_2\|_2 \\ g_1^T H g_2/\|g_1\|_2\|g_2\|_2 & g_2^T H g_2/\|g_2\|_2^2 \end{pmatrix} \begin{pmatrix} t \\ u \end{pmatrix}. \tag{2.5}$$

Due to the exact line search in the first iteration, we have that $\alpha_1^* = \|g_1\|_2^2/g_1^T H g_1$ and $g_1^T H g_2 = -\|g_2\|_2^2/\alpha_1^*$. Using the notation $\alpha_2^* = \|g_2\|_2^2/g_2^T H g_2$, we have that

$$f(x_2 + t\frac{g_1}{\|g_1\|_2} + u\frac{g_2}{\|g_2\|_2}) = \begin{pmatrix} 0 \\ \|g_2\|_2 \end{pmatrix}^T \begin{pmatrix} t \\ u \end{pmatrix} + \frac{1}{2}\begin{pmatrix} t \\ u \end{pmatrix}^T \begin{pmatrix} 1/\alpha_1^* & -\|g_2\|_2/\alpha_1^*\|g_1\|_2 \\ -\|g_2\|_2/\alpha_1^*\|g_1\|_2 & 1/\alpha_2^* \end{pmatrix} \begin{pmatrix} t \\ u \end{pmatrix}. \tag{2.6}$$

Hence we see that the minimizer of the objective function should be

$$\begin{pmatrix} t^* \\ u^* \end{pmatrix} = -\frac{\|g_1\|_2\|g_2\|_2}{\|g_1\|_2^2/\alpha_2^* - \|g_2\|_2^2/\alpha_1^*} \begin{pmatrix} \|g_2\|_2 \\ \|g_1\|_2 \end{pmatrix}. \tag{2.7}$$

In order to have

$$x_4 = x_2 + t^* g_1/\|g_1\|_2 + u^* g_2/\|g_2\|_2, \tag{2.8}$$

we need the gradient direction g_3 parallel to the residual vector $x_4 - x_3$, which requires the two directions

$$\begin{pmatrix} t^* \\ u^* \end{pmatrix} - \begin{pmatrix} 0 \\ -\alpha_2\|g_2\|_2 \end{pmatrix} \tag{2.9}$$

and

$$\begin{pmatrix} 0 \\ \|g_2\|_2 \end{pmatrix} + \begin{pmatrix} 1/\alpha_1^* & -\|g_2\|_2/\alpha_1^*\|g_1\|_2 \\ -\|g_2\|_2/\alpha_1^*\|g_1\|_2 & 1/\alpha_2^* \end{pmatrix} \begin{pmatrix} 0 \\ -\alpha_2\|g_2\|_2 \end{pmatrix} \tag{2.10}$$

are parallel. These two directions are parallel to the following directions

$$\begin{pmatrix} \|g_2\|_2 \\ \|g_1\|_2 - \alpha_2(\|g_1\|_2^2/\alpha_2^* - \|g_2\|_2^2/\alpha_1^*)/\|g_1\|_2 \end{pmatrix} \tag{2.11}$$

and

$$\begin{pmatrix} \alpha_2\|g_2\|_2/\alpha_1^*\|g_1\|_2 \\ 1 - \alpha_2/\alpha_2^* \end{pmatrix} \tag{2.12}$$

respectively. Assume that

$$\begin{pmatrix} \|g_2\|_2 \\ \|g_1\|_2 - \alpha_2(\|g_1\|_2^2/\alpha_2^* - \|g_2\|_2^2/\alpha_1^*)/\|g_1\|_2 \end{pmatrix} = \lambda \begin{pmatrix} \alpha_2\|g_2\|_2/\alpha_1^*\|g_1\|_2 \\ 1 - \alpha_2/\alpha_2^* \end{pmatrix} \tag{2.13}$$

for some $\lambda \in \Re$. From the first line in the above equation we can see that $\lambda = \alpha_1^*\|g_1\|_2/\alpha_2$. Substituting this relation into the second line of (2.13), we obtain that

$$1 - \alpha_2(1/\alpha_2^* - \|g_2\|_2^2/\alpha_1^*\|g_1\|_2^2) = \alpha_1^*/\alpha_2 - \alpha_1^*/\alpha_2^*. \tag{2.14}$$

The above equation is equivalent to

$$(1/\alpha_1^*\alpha_2^* - \|g_2\|_2^2/(\alpha_1^*\|g_1\|_2)^2)\alpha_2^2 - (1/\alpha_1^* + 1/\alpha_2^*)\alpha_2 + 1 = 0. \tag{2.15}$$

Since H is positive definite, we have that

$$\Gamma = 1/\alpha_1^*\alpha_2^* - \|g_2\|_2^2/(\alpha_1^*\|g_1\|_2)^2 > 0. \tag{2.16}$$

Equation (2.15) has two positive solutions for α_2:

$$\frac{(1/\alpha_1^* + 1/\alpha_2^*) \pm \sqrt{(1/\alpha_1^* + 1/\alpha_2^*)^2 - 4\Gamma}}{2\Gamma}. \tag{2.17}$$

The smaller one is chosen and it can written as follows:

$$\alpha_2 = \frac{2}{\sqrt{(1/\alpha_1^* - 1/\alpha_2^*)^2 + 4\|g_2\|_2^2/\|s_1\|_2^2} + 1/\alpha_1^* + 1/\alpha_2^*}, \tag{2.18}$$

where $s_1 = x_2 - x_1 = -\alpha_1^* g_1$. Thus, we have found the formula for α_2 which ensures the method finds the solution after three iterations.

For convex quadratic functions in $n(> 2)$ dimensional spaces, based on (2.18) we suggest the following method.

Algorithm 2.1. *(A new stepsize for steepest descent method)*

Step 0 Given an initial point x_1, Compute g_1, set $k = 1$.

Step 2 Compute the exact line search step α_{2k-1}^*; Set

$$x_{2k} = x_{2k-1} - \alpha_{2k-1}^* g_{2k-1}. \tag{2.19}$$

Step 3 If $g(x_{2k}) = 0$ then stop;

Step 4 Compute the exact line search step α_{2k}^*, Set

$$\alpha_{2k} = \frac{2}{\sqrt{(1/\alpha_{2k-1}^* - 1/\alpha_{2k}^*)^2 + 4\|g_{2k}\|_2^2/\|s_{2k-1}\|_2^2} + 1/\alpha_{2k-1}^* + 1/\alpha_{2k}^*} \tag{2.20}$$

and

$$x_{2k+1} = x_{2k} - \alpha_{2k} g_{2k}. \tag{2.21}$$

If $g_{2k+1} = 0$ then stop;

Step 5 *k:= k+1, go to Step 2.*

Our analysis given above indicates that the following finite termination result holds.

Theorem 2.1. *Assume that Algorithm 2.1 is applied to a convex quadratic function in 2 dimensional space. Then there exists $k \le 4$ such that $\nabla f(x_k) = 0$.*

From the definition of our stepsize α_{2k} in (2.20), it is easy to see that

$$\frac{1}{1/\alpha^*_{2k-1} + 1/\alpha^*_{2k}} < \alpha_{2k} < \min(\alpha^*_{2k-1}, \alpha^*_{2k}), \tag{2.22}$$

which coincides the common sense that a slightly shortened exact line search step would improve the efficiency of the steepest descent method. Due to relation (2.22) the algorithm is monotone, namely we have that

$$f(x_{k+1}) < f(x_k) \tag{2.23}$$

for all k. Furthermore, from the monotone property and the fact that exact line searches are used in all the odd iterations, it is trivial that the following convergence result holds:

Theorem 2.2. *Assume that Algorithm 2.1 is applied to a convex quadratic function in n dimensional space. Then the iterate points x_k generated by the algorithm either terminates at the solution or converges to the solution linearly. Futhermore, $f(x_{k+1}) < f(x_k)$ for all k.*

The formula (2.20) require the exact line search stepsize for all iterations even though it is only taken as the step in odd iterations. For even iterations, it is used in the calculation of the stepsize. In the following, we derive an equivalent condition for the stepsize α_{2k} without computing the exact line search step α^*_{2k}. For convex quadratic functions, it is true that

$$[g(x_{2k} - \alpha g_{2k}) - g_{2k}]^T g_{2k} = \frac{\alpha}{\alpha^*_{2k}}[g(x_{2k} - \alpha^*_{2k} g_{2k}) - g(x_{2k})]^T g_{2k} = -\frac{\alpha}{\alpha^*_{2k}}\|g_{2k}\|_2^2 \tag{2.24}$$

for all $\alpha \in \Re$. Particularly, we have that

$$[g(x_{2k} - \alpha^*_{2k-1} g_{2k}) - g_{2k}]^T g_{2k} = -\frac{\alpha^*_{2k-1}}{\alpha^*_{2k}}\|g_{2k}\|_2^2. \tag{2.25}$$

Thus, it follows that

$$\frac{1}{\alpha^*_{2k}} = \frac{1}{\alpha^*_{2k-1}}\left(1 - \frac{g(x_{2k} - \alpha^*_{2k-1} g_{2k})^T g_{2k}}{\|g_{2k}\|_2^2}\right). \tag{2.26}$$

Substituting this relation into (2.20), we obtain that

$$\alpha_{2k} = \frac{2}{\frac{1}{\alpha^*_{2k-1}}\left(2 - \frac{\bar{g}_{2k}^T g_{2k}}{\|g_{2k}\|_2^2}\right) + \sqrt{\frac{(\bar{g}_{2k}^T g_{2k})^2}{(\alpha^*_{2k-1}\|g_{2k}\|_2^2)^2} + 4\frac{\|g_{2k}\|_2^2}{\|s_{2k-1}\|_2^2}}}\ , \tag{2.27}$$

where

$$\bar{g}_{2k} = g(x_{2k} - \alpha^*_{2k-1} g_{2k}). \tag{2.28}$$

Using this formula, we can modify our algorithm as follows.

Algorithm 2.2. *(Modified version of our new steepest descent method)*

Step 0 Given an initial point x_1, Compute g_1, set $k = 1$.

Step 2 Compute the exact line search step α^*_{2k-1}; Set

$$x_{2k} = x_{2k-1} - \alpha^*_{2k-1} g_{2k-1}. \tag{2.29}$$

Step 3 If $g(x_{2k}) = 0$ then stop;

Step 4 Define $\bar{s}_{2k} = -\alpha^*_{2k-1} g_{2k}$.
Compute $g(x_{2k} + \bar{s}_{2k})$ and let

$$\beta = \frac{g(x_{2k} + \bar{s}_{2k})^T g_{2k}}{\|g_{2k}\|_2^2}. \tag{2.30}$$

Compute

$$\alpha_{2k} = \frac{2}{\frac{1}{\alpha^*_{2k-1}}(2-\beta) + \sqrt{\frac{(\beta)^2}{(\alpha^*_{2k-1})^2} + 4\frac{\|g_{2k}\|_2^2}{\|s_{2k-1}\|_2^2}}} \tag{2.31}$$

and

$$x_{2k+1} = x_{2k} - \alpha_{2k} g_{2k}. \tag{2.32}$$

If $g_{2k+1} = 0$ then stop;

Step 5 k:= k+1, go to Step 2.

The stepsize (2.31) is exactly the same as the one given in (2.20) if the objective function $f(x)$ is convex quadratic. To generalizing our approach to general nonlinear functions, from (2.15), for the even iterations it is reasonable to impose the following line search conditions

$$\left(1 - \frac{\alpha_{2k}}{\alpha^*_{2k-1}}\right) \frac{g(x_{2k} - \alpha_{2k} g_{2k})^T g(x_{2k})}{\|g_{2k}\|_2^2} = \frac{\|g_{2k}\|_2^2}{\|s_{2k-1}\|_2^2} \alpha_{2k}^2, \tag{2.33}$$

and $g(x_{2k} - \alpha_{2k} g_{2k})^T g(x_{2k}) > 0$. It is easy to see that such an α_{2k} exists in the interval between zero and the smallest positive stationary point of $f(x_{2k} - \alpha g_{2k})$.

3. Numerical Results

We compare the numerical performance of our algorithm with the Barzilar-Borwein (BB) method, the Alternate stepsize gradient method (AS) by Dai(2001) and the Alternative minimization method (AM) by Dai and Yuan. Two versions of the BB method are compared, which are called BB1 and BB2 where (1.10) and (1.11) are used respectively. The Alternate stepsize gradient method uses exact line search on odd iterations and use BB stepsize (1.10) on even iterations. The Alternative minimization method carries out exact line searches in odd iterations and minimizes the norm of the gradient in even iterations. Namely,

$$\alpha_k^{AM} = \begin{cases} \frac{\|g_k\|_2^2}{g_k^T H g_k} & \text{if } k \text{ is odd,} \\ \frac{g_k^T H g_k}{g_k^T H^2 g_k} & \text{if } k \text{ is even .} \end{cases} \tag{3.1}$$

For our new method, we also implement two versions. Version A is the un-modified version of Algorithm 2.1. While version B uses one stepsize in the form of (2.20) after every two exact line searches. Namely, the modified algorithm choosed the stepsizes by

$$\alpha_{3k-2} = \alpha^*_{3k-2}, \tag{3.2}$$

$$\alpha_{3k-1} = \alpha^*_{3k-1}, \tag{3.3}$$

$$\alpha_{3k} = \frac{2}{\sqrt{(1/\alpha^*_{3k-1} - 1/\alpha^*_{3k})^2 + 4\|g_{3k}\|_2^2/\|s_{3k-1}\|_2^2} + 1/\alpha^*_{3k-1} + 1/\alpha^*_{3k}}. \tag{3.4}$$

The problem we used to compare the algorithms are the following

$$f(x) = (x - x^*)^T Diag(\sigma_1, \cdots, \sigma_n)(x - x^*). \quad x \in \Re^n. \tag{3.5}$$

We consider the cases when $n = 2, 3, 10, 100, 1000, 10000$. The solution vector $x_i^*(i = 1, ..., n) \in (-5, 5)$ are randomly generated. We let $\sigma_1 = 1$ and $\sigma_n = Cond(= 10, 100, 1000, 10000)$ which is the condition number of the Hessian of function $f(x)$. $\sigma_i(i = 2, \cdots, n-1)$ are randomly chosen in the interval $(1, \sigma_n)$. For all problems the initial point is the zero vector $(0, \cdots, 0)^T$ and the stop condition is $\|g_k\| \le 10^{-8}$.

The numerical results are reported in Table 1. For each case, 10 runs are made and average numbers of iterations required by each algorithm are listed in the Table.

Table 1. Iteration numbers for different methods

n	σ_n	BB1	BB2	AS	AM	2.1 (A)	2.1 (B)
2	10	8.4	7.2	6.3	9.7	3	4
2	100	6	6	5	8.8	3	4
2	1000	6	5.8	5	8	3	4
2	10000	4.7	4	3.8	7.4	3	3.9
3	10	20.2	19.4	18.2	34.3	13.2	16.1
3	100	24.6	23.2	16.7	88.1	8.2	21
3	1000	27.3	23.2	19.8	191.3	11.1	23.4
3	10000	28.3	19.9	18.8	1237	7.6	19.5
10	10	35	35	33.2	48.8	41.4	34.2
10	100	113.1	114.6	100.6	132.8	144.8	109.2
10	1000	323.7	327.5	260.5	559.5	871.2	350
10	10000	659.9	649.8	537.5	2235.7	3085.2	1089.5
100	10	41.3	41.4	42.9	57.4	40.8	39
100	100	141.6	141.2	138.2	157.2	219.6	134.1
100	1000	464.2	448.6	457.8	695.7	1714.4	517.6
100	10000	1253.5	1063.9	1304.4	4187	16570.6	2012.3
1000	10	43.4	42.1	42	59.4	42.8	41.8
1000	100	147	148.2	150.9	167.9	234.2	140.3
1000	1000	401.9	493.1	501.8	739.4	1875.2	561.5
1000	10000	1434.8	1324.6	1303.6	4380.6	14630.5	2177.2
10000	10	46.1	42.8	44.9	62	45	43.4
10000	100	159.8	158.9	158.1	172	263.1	151.4
10000	1000	529.6	497.1	563.1	775.6	1930	553.5
10000	10000	1570.7	1335.1	1413.7	4332.8	17293.6	2136.4

The numerical results confirmed that the new stepsize ensures the solution can be found within 3 iterations. The results also showed that the new method is very efficient for small scale problems. For large scale problems, the new method is also as efficient as the BB methods if the condition number of the Hessian of the objective function is not large. If the condition number of the Hessian of the objective function is very large, especially for large scale problems, the new method is much worse than the BB method. The modified version of our algorithm performed quite well. It is comparable to the BB method for large scale problems and better for small scale problems. Please notice that both versions of our new method have the monotone property which the BB method does not have.

4. Discussion

In this paper we have suggested a new stepsize for the steepest descent method. An algorithm with this new stepsize in even iterations and exact line search in odd iterations is proposed. The improvement of the modified version of our new method over the unmodified version is unexpected. It might be interested to investigate other possibilities, such as taking a type (2.20) step after every m exact line search iterations. Another possibility is to suggest a stepsize formula for α_k which depends on the pervious two exact line search steps α^*_{k-1} and α^*_{k-2}.

References

[1] H. Akaike, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method, *Ann. Inst. Statist. Math. Tokyo*, **11** (1959), 1-16.

[2] J. Barzilai and J.M. Borwein, Two point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141-148.

[3] A. Cauchy, Méthode générale pour la résolution des systèms d'equations simultanées, *Comp. Rend. Sci. Paris*, **25** (1847), 46-89.

[4] Y.H. Dai, Alternate step gradient method, Report AMSS-2001-041, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, 2001.

[5] Y.H. Dai, J.Y. Yuan, and Y. Yuan, Modified two-point stepsize gradient methods for unconstrained optimization, *Computational Optimization and Applications*, **22** (2002), 103-109.

[6] Y.H. Dai and Y. Yuan, Alternate minimization gradient method, *IMA Journal of Numerical Analysis*, **23** (2003), 377-393.

[7] Y.H. Dai and H. Zhang, An Adaptive Two-Point Stepsize Gradient Method, Research report, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 2001.

[8] R. Fletcher, Practical Methods of Optimization(second Edition), John Wiley and Sons, Chichester, 1987.

[9] R. Fletcher, On the Barzilar-Borwein method, Research Report, University of Dundee, UK, 2001.

[10] G. E. Forsythe, On the asymptotic directions of the s-dimensional optimum gradient method, *Numerische Mathematik*, **11** (1968), 57-76.

[11] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, Gradient method with retards and generalizations, *SIAM J. Numer. Anal.*, **36** (1999), 275-289.

[12] J. Nocedal, A. Sartenaer and C. Zhu, On the behavior of the gradient norm in the steepest descent method, Research report, Northwestern University, USA, 2000.

[13] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, *IMA J. Numer. Anal.*, **13** (1993), 321-326.

[14] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM J. Optim.*, **7** (1997), 26-33.

[15] M.N. Vrahatis, G.S. Androulakis, J.N. Lambrinos and G.D. Magoulas, A class of gradient unconstrained minimization algorithms with adaptive stepsize, *J. Comput. and Appl. Math.*, **114** (2000), 367-386.