Accelerating Steepest Descent Algorithm by Aitken iterations in Nonlinear Optimization

Muna M. M. Ali
Department of Mathematics, College of Computer sciences and mathematics, University of Mosul , Mosul , Iraq
(Received 16 / 9 / 2008 , Accepted 1 / 3 / 2009)

Abstract
In this study, we develope an improvement is made on the Steepest Descent algorithm, which is based on Aitken iterations.
The proposed algorithm uses Aitken's steps for Accelerating the Steepest Descent algorithm and finds the min x* without using line search. we more precisely compare the new proposed algorithm with the standard **S.D.** algorithm using few nonlinear test function.
The numerical results are better than previous algorithm.

1. Introduction:-
The method of steepest descent is an algorithm of the type just described. It is defined by stipulating that d_k should be the negative gradient of f at x_k. It turns out that this negative gradient points in the direction of the residual, $g(x)=b-Ax_k$. In the actual programming of this algorithm, the successive vectors $x_0,x_1,\ldots,x_n$ need not be saved; the current x-vector can be overwritten, The same remark holds for the direction vectors $d_0,d_1,\ldots$, Then the method of steepest descent is one of the most fundamental procedures for minimizing a differentiable function of several variable in optimization. Recall that a vector d_k is called a direction of descent of a function f at x if there exists $\delta > 0$ such that $f(x+\lambda d) < f(x)$ for all $\lambda \in (0,1)$, [1]. Inparticular, if $\lim_{\lambda\to 0}[f(x+\lambda d) - f(x)]/\lambda < 0$ then d is a direction of descent. The method of steepest descent moves along the direction d with ||d||=1, which minimizes the above limit,[1].
The following lemma shows that if f is differentiable at x with a none zero gradient, then $-g(x)/\|g(x)\|$ is indeed the direction of steepest descent. For this reason, in the presence of differentiability, the method of steepest descent is sometimes called the gradient method,[2].

2. Preliminaries
Lemma (2.1):
Suppose that $f: R^n \to R^{'}$ is differentiable at x. Suppose that $g(x) \neq 0$. Then, the optimal solution at the problem to minimize $f^{'}(x;d)$ subject to ||d|| < 1 is given by $\bar{d} = -g(x)/\|g(x)\|$; that is $-g(x)/\|g(x)\|$ is the direction of steepest descent of f at x. ,[3].

2.2 Summary of Steepest Descent Algorithm
Given point x, the steepest descent algorithm proceeds by performing a line search along the direction $-g(x)/\|g(x)\|$; or, equivalently, along the direction $-g(x)$. The summary of the method is given by:

2.2.1 Initialization step:-

Let $\varepsilon > 0$ be the terminal scalar. Choose a starting point x_1, let k=1, and go to the main step.

2.1.2 Main step:-

If $\|g(x_k)\| < \varepsilon$, otherwise, let $d_k = -g(x_k)$, and let λ_k be an optimal solution of the problem to minimize $f(x_k + \lambda d_k)$ subject to $\lambda \geq 0$.
Let $x_{k+1} = x_k + \lambda_k d_k$, replace k by k+1, and replace the main step. [4].

3. Rate of Convergence of the Steepest Descent Algorithm
The foregoing analysis can be extended to a general quadratic function,[5] $f(x) = c + b^T x + \frac{1}{2} x^T G x$, (1)

The unique minimizer x^* for this function is given by the solution to the system $Gx^* = -b$ obtained by setting $g(x^*) = 0$,[5]. Also, give an iterate x_k, the optimal step length λ and the revised iterate x_{k+1} are given by

$$\lambda = \frac{g_k^T g_k}{g_k^T G g_k} \text{ and } x_{k+1} = x_k - \lambda g_k \text{ ,[1].} \quad (2)$$

Now, to evaluate the rate of convergence, let us employ a convenient measure for convergence given by the following error function:

$$e(x) = \frac{1}{2}(x - x^*)^T G (x - x^*) = f(x) + \frac{1}{2} x^{*T} G x^* \quad (3)$$

where we have used fact that $Gx^* = -b$. Note that e(x) is different from f(x) by only a constant and equals zero if and if $x = x^*$. In fact, it can be show that

$$e(x_{k+1}) = \left[1 - \frac{(g_k^T g_k)^2}{(g_k^T G g_k)(g_k^T G^{-1} g_k)}\right] e(x_k) \leq \frac{(\alpha-1)^2}{(\alpha+1)^2} e(x_k), \quad (4)$$

where α is the condition number of G. Hence $\{e(x_k)\} \to 0$ at a linear or geometric convergence rate bounded above by $(\alpha-1)^2/(\alpha+1)^2$; and so as before, we can expect the convergence to become increasingly slower as α increase, depending on the initial solution x_0 ,[3].
The method of steepest descent is rarely used on this problem because it is too slow,[6].
In spite of its theoretical appeal, this algorithm often performs poorly, spending much time zig-zagging in directions that not to point toward the global minimum,

and so the zig-zagging is an inveitable feature of this procedure.

4. The Aitken's Δ^2 Acceleration Process:-

Suppose that g is differentiable in some open interval I containing a fixed-point x^* of g and $|g^*(x)| \le L < I. \forall x \in I$.

Furthermore, Let $x_0 \in I$ and $x_{n+1} = g(x_n)$ remains in I. Then the sequence $(x_n)_{n\in N}$ converges to x^* and

$|x_n - x^*| \le L^n |x_0 - x^*|$, n=1,2,... (5)

$|x_n - x^*| \le \frac{L^n}{1-L} |x_1 - x_0|$, n=1,2,... (6)

are satisfied. more over if

$g'(x) \ne 0$ on I and $x_0 \ne x^*$, then $e_n \ne 0, \forall n \ge 0$ and the fixed-point iteration

$x_{n+1} = g(x_n)$ is linearly converges,[7]. That is

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|} = |g'(x^*)|, \qquad 0 < |g'(x^*)| < 1 \cdot \quad (7)$$

If $|g'(x^*)|$ is close to 1 the convergence may be very slow,[8]. In what follows we construct a sequence $(\overline{x}_n)\ n \in N$ that converges faster to x^*,[6]. By (mid point theorem) there exists c_{n+1} between x_{n+1} and x^* such that

$x_{n+2} - x^* = g'(c_{n+1})(x_{n+1} - x^*)$, Solving for x^0 we get

There exists c_{n+1} between x_{n+1} and x^0 such that

$$x_{n+1} - x^* = g'(c_{n+1})(x_{n+1} - x^*) \qquad (8)$$

Solving for x^* we get

$$x^* = \frac{x_{n+2} - x_{n+1}\, g'(c_{n+1})}{1 - g'(c_{n+1})} \qquad (9)$$

If we replace

$$g'(c_{n+1}) \approx \frac{g(x_{n+1}) - g(x_n)}{x_{n+1} - x_n} = \frac{x_{n+2} - x_{n+1}}{x_{n+1} - x_n} \qquad (10)$$

In (10) we get the following approximation $\overline{x}_n$ to x^*

$$\overline{x}_n = x_n - \frac{[x_{n+1} - x_n]^2}{x_{n+2} - 2x_{n+1} + x_n}, n \ge 0 \qquad (11)$$

Using the abbreviations

$$\Delta x_k = x_{k+1} - x_k\ , k \ge 0$$

$$\Delta^2 x_k = \Delta x_{k+1} - \Delta x_k = x_{k+2} - 2x_{k+1} + x_k\ , k \ge 0 \qquad (12)$$

The process (11) can be written as

$$\overline{x}_n = x_n - \frac{[\Delta x_n]^2}{\Delta^2 x_n}, n \ge 0 \qquad (13)$$

which is known as the Aitken's Δ^2 method. It can be proved, with the above assumptions, that the sequence $(\overline{x}_n)_{n\in N}$ converges to x^* faster than $(x_n)_{n\in N}$. That is

$$\lim_{n\to\infty} \frac{\overline{x}_n - x^*}{x_n - x^*} = 0 \qquad (14)$$

Theorem on Aitken Acceleration:-

Let $(x_n)_{n\in N}$ be a sequence of number that converges to a limit x^*. Then the new sequence

$$\overline{x}_n = \frac{x_n x_{n+2} - x_{n+2}^2}{x_{n+2} - 2x_{n+1} + x_n}, n \ge 0 \qquad (15)$$

Convergence to x^* faster than $(x_n)_{n\in N}$ if

$x_{n+1} - x^* = (c + \delta_n)(\overline{x}_n - x^*)$ with $|c| < 1$ and $\lim_{n\to\infty} \delta_n = 0 \cdot$ (16)

Indeed $(\overline{x}_n - x^*)/(x_n - x^*) \Rightarrow 0$ an $n \to \infty$. (17)

Proof:-

Define the error sequence $h_n = x_n - x^*$.

The start calculation reveals that

$$\overline{x}_n = \frac{(x^* + h_n)(x^* + h_{n+2}) - (x^* + h_{n+1})^2}{(x^* + h_{n+2}) - 2(x^* + h_{n+1}) + (x^* + h_n)} \qquad (18)$$

$$= x^* + \frac{h_n\, h_{n+2} - h_{n+1}^2}{h_{n+2} - 2h_{n+1} + h_n}$$

se the hypothesis $h_{n+1} = (c + \delta_n) h_n$ to obtain

$h_{n+2} = (c + \delta_{n+1})(c + \delta_n) h_n$ and

$$\overline{x}_n - x^* = \frac{h_n (c + \delta_{n+1})(c + \delta_n) h_n - (c + \delta_n)^2 h_n^2}{(c + \delta_{n+1})(c + \delta_n) h_n - 2(c + \delta_n) h_n + h_n}$$

$$= h_n \frac{(c + \delta_{n+1})(c + \delta_n) - (c + \delta_n)^2}{(c + \delta_{n+1})(c + \delta_n) - 2(c + \delta_n) + 1} \qquad (19)$$

It is now clear that $\lim_{n\to\infty} (\overline{x}_n - x^0)/h_n = 0$,[7]. since in Equation the numerator converges to 0 and the denominator converges to $(c-1)^2$, which is not 0,[2].

5. A New approach to Accelerate the Steepest descent direction:-

In this section we present a modification of Steepest descent using Aitken method. We compute $x_{k+1} = x_k + \lambda_k d_k$ where $k < 3$, and applying Aitken method to compute x_{k+1} (where $k \rangle 3$) without line search.

It should be noticed that the Aitken method can be used to accelerate any linearly convergent sequence regardless of its origin. And we have in this new algorithm accelerate the steepest descent from linear rate convergence to superior linear convergence.

6. Out line of the algorithm:

Step (1):- set x_1 is initial point, $\in$,k=1

Step (2):- if ($k \le 3$) then compute $d_k = -g_k$

Step (3):- find λ_k to minimize $f(x_k + \lambda_k d_k)$

Step (4):- put $x_{k+1} = x_k + \lambda_k d_k$

Step (5):- if $\| g_{k+1} \| < \varepsilon$ go to step (10)

Step (6):- set k=k+1 and go to (2), else go to (7)

Step (7):- compute $x_{k+3} = x_k - \frac{(x_{k+1} - x_k)^2}{(x_{k+2} - 2x_{k+1} + x_k)}$

Step (8):- if $\| g_{k+3} \| < \varepsilon$ stop

Step (9):- k=k+1

Step (10):- stop.

7. Numerical results:-

The comparative tests involve well known standard test functions with different dimensions where $2 \le n < 1000$. All the results are obtained using double precision on the (Pentium-computer),[9],[10].

All programs are written in FORTRAN language and for all cases the stopping criterion is taken to be $\| g_{k+1} \| < 1 \times 10^{-5}$.

Table (1) gives the comparison between the standard steepest descent algorithm and the new algorithm. This table indicates that the new algorithm is better than the standard steepest descent algorithm in about 34% NOI and 31% NOF.

8. Conclusions:-

In this study we have modified SD direction by accelerating this direction using Aitken steps. The new search direction converges faster than the standard SD direction.

Table (1) Comparative performance of the two algorithms for the group of test functions.

Test function	N	S.D. method	New method
		NOI (NOF)	NOI (NOF)
Rosen	2	20(35)	4(14)
	4	24 (35)	4 (15)
	100	30 (50)	14 (31)
	1000	40(55)	20(39)
Powell	2	80 (170)	60 (127)
	4	80 (170)	69(121)
	100	101 (241)	71 (209)
	1000	104 (241)	97 (211)
Wood	2	33(70)	26(56)
	4	33 (75)	26 (56)
	100	44 (93)	33 (70)
	1000	151 (202)	106 (171)
Cubic	2	15(40)	5 (20)
	4	15 (40)	10(35)
	100	13 (40)	13 (39)
	1000	20(50)	15(41)
Dixon	2	20(35)	20(31)
	4	25(50)	24(50)
	100	25 (55)	25 (55)
	1000	41 (111)	38 (27)
		914 (1858)	680 (1418)

9. Appendix:

1- Generalized Powell Function:

$$f = \sum_{i=1}^{n} \left[\begin{array}{l} (x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 \\ + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \end{array} \right],$$

$$x_0 = (3,-1,0,1;\cdots)^T.$$

2- Generalized Wood Function:

$$f = \sum_{i=1}^{n} 100\left[(x_{4i-2} - x_{4i-3}^2)^2\right] + (1 - x_{4i-3})^2$$

$$+ 9(x_{4i} - x_{4i-1}^2)^2 + (1 - x_{4i-1}^2)^2 + 10.1\left[\begin{array}{l} (x_{4i-2} - 1)^2 \\ + (x_{4i} - 1)^2 \end{array} \right]$$

$$+ 19.8(x_{4i-2} - 1)^2 (x_{4i} - 1),\ x_0 = (-3,-1,-3,-1;\cdots)^T.$$

3- Generalized Sum of Quadratics Function:

$$f = \sum_{i=1}^{n} (x_i - 1)^4,$$

$$x_0 = (2;\cdots)^T.$$

4- Generalized Dixon Function:

$$f = \sum_{i=1}^{n} \left[(1 - x_i)^2 + (1 - x_n)^2 + \sum_{i=1}^{n-1} (x_i^2 - x_{i-1})^2 \right],$$

$$x_0 = (-1;\cdots)^T.$$

5- Generalized Rosenbrock Function:

$$f = \sum_{i=1}^{n} \left[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \right],$$

$$x_0 = (-1,2,1;\cdots)^T.$$

6- Generalized Cubic Function:

$$f = \sum_{i=1}^{n} \left[100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2 \right],$$

$$x_0 = (-1,2,1;\cdots)^T.$$

7- Generalized Tri Function:

$$f = \sum_{i=1}^{n} (i x_i^2)^2,$$

$$x_0 = (-1;\cdots)^T.$$

References

[1] Biggs, M.C., (1973) " A Note of Minimization Algorithm which make use of Non-Quadratic properties of the objective function", Journal of institute of Mathematics and its Application (12),PP. 337-338.

[2] Scales, L. E. (1985) "Introduction to Nonlinear Optimization" Macmillan, London.

[3] Bazaraa, M., Sherali, H. and Shetty,C.M.,(1993) "Non linear Programming Theory and application", New York, John Wiley and Sons, Inc.

[4] Ruessell, D., (1970) " Optimization Theory ", New York, W.A. Benjamin Inc.

[5] Rao, S., S., (1994) " Optimization Theory and Applications", Wiley Eastern limited .

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

[7] Kincaid David, D., cheney Wand . (2002) " Numerical Analysis Mathematics of scientific Computing 3rd ed".

[8] Bhatti, M.A.,(2000) " Practical optimization methods ", New York : Springer-Verlag.

[9] Bazaraa, M.S. (2000)" Nonlinear Programming", England, Universities Press, London.

[10] Dennis, J.E., Jr., and R.B. Schnabel, (1983)," Numerical Methods for Unconstrained Optimization and Nonlinear Equations ", Englewood cliffs, NJ: preutice-hall

.

تعجيل خوارزمية SD بخطوات Aitken في الامثلية اللاخطية

منى محسن محمد علي

قسم الرياضيات ، كلية علوم الحاسبات والرياضيات ، جامعة الموصل ، الموصل ، العراق

(تاريخ الاستلام: ١٦ / ٢٠٠٨/١٠ ، تاريخ القبول: ١ / ٣ / ٢٠٠٩)

الملخص

في هذه الدراسة تم تحسين خوارزمية Steepest Descent وذلك باستخدام خطوات Aitken. حيث تم التعجيل بإدخال خطوات Aitken على الخوارزمية المقترحة وكيفية حساب المتجه والتخلص من حساب حجم الخطوة (حيث اننا لا نحتاج الى خط البحث). لقد تم مقارنة الخوارزمية مع خوارزمية Steepest Descent القياسية لعدد من الدوال اللاخطية. وحصلنا على نتائج افضل من الخوارزمية السابقةلها.