J. KSIAM **Vol.8**, No.2, 15-21, 2004

A LOGARITHMIC CONJUGATE GRADIENT METHOD INVARIANT TO NONLINEAR SCALING

I.A. MOGHRABI

ABSTRACT. A Conjugate Gradiant (CG) method is proposed for unconstained optimization which is invariant to a nonlinear scaling of a strictly convex quadratic function. The technique has the same properties as the classical CG-method when applied to a quadratic function. The algorithm derived here is based on a logarithmic model and is compared to the standard CG method of Fletcher and Reeves [3]. Numerical results are encouraging and indicate that nonlinear scaling is promising and deserves further investigation.

1. INTRODUCTION

In this paper, we propose a non-linear logarithmic model and use it as the basis for deriving a new CG algorithm. If $q(x)$ is a quadratic function and f is defined as a nonlinear scaling of $q(x)$ if the following conditions hold: $f = F(q(x))$, $df > 0$ for $x = x^*$, where x^* is the minimizer of $q(x)$ with respect to x.

The following properties immediately follow from the above conditions:
i)every contour line of $q(x)$ is a contour line of f,
ii) if x^* is a minimizer of $q(x)$, then it is also a minimizer of f.

Various related work have been published in this area: and we mention some as follows:
a) a conjugate gradient method which minimizes the function $f(x) = q(x)^p, p > 0$ and $x \in R^n$ in at most n iterations has been described by Fried in [4].
b) The following minimization problem has been considered by Goldfarb in [5]:
Minimize $f(x) = F(q(x))$ where $dF/dq = F' > 0, q > 0$.
This last relation corresponds to nonlinear scaling and used by Spedicato [6] to define invariancy to nonlinear scaling.
c) The special case $F(q(x)) = \varepsilon_1 q + (1/2)\varepsilon_2 q^2$, where ε_1 and ε_2 are real scalars, have been investigated by Boland et al [1].
d) Another model has been considered by Tassopoulos and Storey [7], is given by

2000 *Mathematics Subject Classification.* 65K10.
Key words and phrases. Unconstrained optimization, Conjugate Gradiant methods, nonlinear scaling.

$$F(q(x)) = (\varepsilon_1 q + 1)/\varepsilon_2 q(x), \text{ for } \varepsilon_2 < 0.$$

In the next section, we investigate, in a similar context, a logarithmic model on which we base our new algorithm.

2. The Logarithmic Model

We consider here the model

$$F(q(x)) = \ln(q(x)). \tag{1}$$

We first observe that $F(q(x))$ and $q(x)$ have identical contours and the same unique minimizer x^* but with different function values. For any F satisfying $dF/dq = F' > 0$, $q > 0$, it has been shown in [1] that the updating process specified below genetrates identical congugate directions and the same sequence of iterates x_i, to the minimum x^*, as does the original method of Fletcher and Reeves [3] when applied to $f(x) = q(x)$ and for $H_i = I$.

We outline now our algorithm. Given $x_0 \in R^n$ as an initial estimate to the minimum x^*, then

$$p_0 = -g(x_0),$$

then iteratively we use

$$x_{i+1} = x_i + \alpha_i p_i, \text{ for } i \geq 0, \tag{2}$$

and then

$$p_{i+1} = -H_i g(x_{i+1}) + \beta_i p_i, \text{ for } i \geq 1, \tag{3}$$

where H_i is some positive definite matrix that is usually updated at each iteration. H_i is set to I in the standard Fletcher-Reeves method. Also,

$$\beta_i = (\delta_i g_i^T g_i)/g_{i-1}^T g_{i-1}, \tag{4}$$

where δ_i is to obtain invariancy to nonlinear scaling and is derived next.

It is well known that CG methods need to be restarted in practice to reset any accumulation of errors that may influence the numerical stability of the methods, caused by generating search directions which are not necessarily downhill. A well known criteria for restarting is given by

$$p_i^T g_i \geq \theta g_i^T g_i, \text{ for } -0.8 \leq \theta \leq 0.96.$$

This condition indicates that the search direction is not sufficiently downhill.

The key feature of the proposed algorithm is in the derivation of the parameter θ, where $\theta = F'_{i-1}/F'_i$, which must be easily computable using available data such as function and gradient evaluations. It follows from (1) that

$$g_i = F'_i A(x_i - x^*) \tag{5}$$

and

$$g_{i-1} = F'_{i-1} A(x_{i-1} - x^*), \tag{6}$$

where A is the Hessian of q. Thus,

$$\theta_i = F'_{i-1}/F'_i = (g_{i-1}^T(x_i - x^*))/g_i^T(x_{i-1} - x^*).$$

Also,

$$g_{i-1}^T(x_i - x^*) = g_{i-1}^T(x_{i-1} + \alpha_{i-1}p_{i-1} - x^*),$$

using (2).
Similarly,

$$g_i^T(x_{i-1} - x^*) = g_i^T(x_i + \alpha_{i-1}p_{i-1} - x^*).$$

Therefore,

$$\theta_i = (g_{i-1}^T[x_{i-1} - x^*] + \alpha_{i-1}p_{i-1}^T g_{i-1})/g_i^T[x_i - x^*] + \alpha_{i-1}p_{i-1}^T g_i. \tag{7}$$

Using (5), (6) and (7), we obtain

$$\begin{aligned}\theta_i \;=\; & [F'_{i-1}(x_{i-1} - x^*)^T A(x_{i-1} - x^*) + \alpha_{i-1}p_{i-1}^T g_{i-1}]/ \\ & [F'_i(x_i - x^*)^T A(x_i - x^*) - \alpha_{i-1}p_{i-1}^T g_i]\end{aligned}$$

Therefore,

$$\theta_i = [2F'_{i-1}q_{i-1} + \alpha_{i-1}p_{i-1}^T g_{i-1}]/[2F'_i q_i - \alpha_{i-1}p_{i-1}^T g_i] \tag{8}$$

From (1), we can express F'_{i-1} and F'_i as

$$F'_{i-1} = 1/q_{i-1} \text{ and } F'_i = 1/q_i,$$

which, upon substitution in (8), yield

$$\theta_i = [2 + \alpha_{i-1} p_{i-1}^T g_{i-1}]/[2 + \alpha_{i-1} p_{i-1}^T g_i]. \quad (9)$$

In case of exact line searches, relation (9) becomes

$$\theta_i = 1 + \alpha_{i-1} g_{i-1}^T g_{i-1}/2.$$

3. Numerical Results and Conclusions

Seventeen standard test functions (see appendix) are employed, in dimensions up to 1000, in order to examine the overall effectiveness of the two new algorithms. The algorithms were tested using C++ on a PIV 200 processor, using double precision. The line search accuracy parameter α is chosen to satisfy

$$p_{i+1}^T g_{i+1} \geqslant 0.75 g_{i+1}^T g_{i+1}.$$

Whenever relation (7) is not satisfied the iteration is restarted, as follows: the estimated error-vector term is used as in [6], i.e, $e_{n+1} = \sum_{i=0}^{n} \varepsilon_i p_i$, where $\varepsilon_i = (\alpha_i g_{i+1}^T p_i)/(y_i^T p_i)$. This estimated error-vector term is added to x_{n+1} to find x_{n+2}, i.e $x_{n+2} = x_{n+1} + e_{n+1}$ and the iteration is then restarted with $-g_{n+2}$.

The line-search algorithm used is a standard cubic interpolation. Table 1 contains the respective numerical results for the standard Fletcher-Reeves algorithm [3] and the new algorithm. The table reports the number of function calls (NOF), the number of iterations (NOI) and the corresponding function value F are given for each test function. Overall totals are also given for NOF and NOI.

Comparisons are affected by the choice of test function, accuracy required, line search and restarting criterion. Nevertheless, the computational results indicate clearly that the new algorithm gives an overall improvement of at least 10% on NOF or NOI, although on individual functions there can be a loss of efficiency.

It is generally evident that the new algorithm has a clear advantage especially on higher dimensions.

An iteration terminates when $| f - f_{min} | < 1 \times 10^{-10}$

TABLE : (1)

TEST		**F/R**	**NEW**
FUNCTIONS	N	NOI(NOF)	NOI(NOF)
ROSEN	2	22(54)	23(59)
CUBE	2	22(57)	23(62)
BEALE	2	8(20)	9(28)
BOX	2	9(41)	9(40)
FREUD	2	6(18)	6(21)
BIGGS	3	11(31)	11(31)
HELICAL	3	18(39)	17(34)
RECIPE	3	6(19)	7(21)
MIELE	4	30(83)	28(77)
POWELL	4	31(67)	23(59)
WOOD	4	19(42)	20(41)
DIXON	10	17(37)	23(49)
OREN	10	12(64)	13(56)
NON-DIGN	20	20(46)	21(47)
TRI-DIGN	30	28(57)	29(58)
OREN	30	21(95)	20(91)
SHALLOW	40	6(18)	6(19)
FULL	40	39(79)	38(83)
EX-ROSEN	60	23(57)	24(60)
EX-POWELL	60	40(83)	42(70)
EX-WOOD	60	17(42)	18(46)
EX-POWELL	80	43(88)	41(84)
WOLFE	80	48(75)	41(79)
NON-DIGN	90	23(53)	21(51)
EX-WOOD	100	19(42)	19(39)
EX-ROSEN	100	23(57)	23(48)
Wood	200	29(65)	30(58)
POWELL	200	52(133)	41(93)
POWELL	1000	89(240)	85(186)
TOTAL	NOI	731	712
	NOF	1802	1690

References

[1] oland, W.R., Kamgnie ,E.R. and Kowalik, J.S, A Conjugate Gradiant Optimization method Invariant to Nonlinear Scaling, JOTA 27, 1979

[2] undy, B, Basic Optimization Method, Edward Amold, London, 1984.

[3] letcher, R. and Reeves, C.M., Function Minimization by Conjugate Gradients, Comput. J.7 1964.

[4] ried, I. N-step Conjugate Gradient Minimization Scheme for nonquadratic functions, AIAA J. 9, 1971.

[5] oldfarb, D. Variable Metric and Conjugate Direction Methods, Unconstrained Optimization- Recent Developments, Assoc. Comput. Mach.Proc.1972.

[6] pedicato, E. A, Variable Metric Method for Function Minimization Derived from Invariancy to Nonlinear Scaling, JOTA 20, 1976.
[7] assopoulos, A. and Story, C.A, Conjugate Direction Method Based on a Nonquadratic Model, JOTA 43, 1984.

Appendix

1. Rosenbrock banana function, n=2,
$f=100\ (x_2-x_1^2)^2+(1-x_1)^2$, $x_0=(-1.2,1.0)^T$.
2. Cube function, n=2,
$f=100\ (x_2-x_1^3)^2+(1-x_1)^2$, $x_0=(-1.2,1.0)^T$.
3. Scale function, n=2,
$f=(1.5-x_1\ (1-x_2))^2+\ (2.25 - x_1\ (1 - x_2^2))^2+\ (2.625 - x_1(1 - x_2^3))^2$, $x_0 = (0,0)^T$.
4. Box function, n = 2,
$f = \sum_{i=1}^{n}(\ e^{-x1}z_i - e\ -x^2z_i - e^{-zi} + e^{-10zi})\ ^2$,where $z_i = (0.1)^i$ and $x_0 = (5,0)^T$, i=1,...
5. Frudenstein and Roth function, n=2,
$f = [-13 + x_1 + ((5-x_2)x_2-2)x_2]^2 + [-29 + x_1 + ((1+x_2)x_2-14)x_2]^2$, $x_0 = (30,3)^T$.
6. Recipe function, n = 3,
$f = (x_1-5)^2 + x_2^2 + x_3^2/(x_1-x_2)^2$, $x_0 = (2,5,1)^T$.
7. Biggs function, n=3,
$f=\sum_{i=1}^{n}(e^{-x1zi} - x_3\ e^{-x2zi}- e^{-zi} + 5e^{-10zi})\ ^2$, where $z_i = (0.1)$ i and $x_0= (1,2,1)^T$,i=1...
8. Helical Valley function, n=3,
$f = 100\ \{\ [x_3-1.0\ ]^2 + [r-1]^2\ \}+ x_3^2$,where r=1/2 arctan $(x_2/x_l\ )$, for $x_l > 0$
and r = 1/2 + 1/2 arctan $(x_2/x_l\ )$ for $x_l < 0$, $x_0 = (-1,0,0)^T$.
9. Miele and Cornwell function, n = 4,
$f = (e^{x_1} - 1)^2 + \tan 4\ (x_3 - x_4) + 100\ (x_2- x_3)^2 + 8x_1+ (x_4- l)^2$, $x_0= (1, 2, 2, 2)^T$.
10. Dixon function, n = 10,
$f = (1- x_l\ )^2 + (1- x_{10})^2 + \sum_{i=1}^{n}(x_i^2 - x_i+1)^2$, $x_0 = (-1;..\ )^T$, i=2,...
11. Oren and Spedicato power function, n = 10,30,
$f = \sum_{i=1}^{n}(i - x_i^2\ )^2$, $x_0 = (1,.....)^T$.
12. Non diagonal variant of Rosenbrock function, n = 20, 90,
$f = \sum_{i=1}^{n}[100\ (x_l - x_i^2)^2 + (\ 1 - x_i\ )^2\ ]$, $x_0 = (-1,\ ..)\ ^T$, i=1,..
13. Tri-diagonal function, n = 30,
$f = [\ \sum_{i=2}^{n}(2x_i - x_{i-1}\ )^2\ ]$, $x_0 = (1;....)^T$.
14. Full set of distinct eigenvalues problem, n = 40,
$f = (x_l-1)^2 + \sum_{i=2}^{n}(2x_i - xi_{-1})^2$, $x_0 = (1;..)^T$.
15. Shallow function (Generalized form), n = 40,
$f = \sum_{i=1}^{n/2}(x_{2i-1}^2 - x_{2i}\ )^2 + (1 - x_{2i-1}\ )^2$, $x_0 = (-2;.\ )^T$.
16. Powell function (Generalized form), n = 60, 80,
$f = \sum_{i=1}^{n/4}[(\ x_{4i-3} + 10\ x\ _{4i-2})^2 + 5\ (x\ _{4i-1} - x_{4i}\ )^2 + (x_{4i-2}-2\ x_{4i-1})^4 + 10\ (x_{4i-3} - x^{4i})^4]$,
$x_0 = (3,-1,0,1;...)^T$.
17. Wood function(Generalized form), n = 60,100,
$\sum_{i=1}^{n/4}f = [100\ (x_{4i-2} - x_{4i-3}^2)^2 + (1- x_{4i-3})^2+ 90\ (x_{4i} - x_{4i-1}^2)^2 + (1 - x_{4i-1})^2$

$+ 10.1\ (x_{4i-2} - 1)^2 + (x_{4i} - 1)^2 + 19.8\ (x_{4i-2} - 1)(x_{4i-l}),\ x_0 = (-3,-1;-3,-1,\ldots)^T.$

Department of Computer Science
Faculty of Science, Beirut Arab University,
P.O. Box 11-5020, Beirut, Lebanon
email: i_moghrabi@yahoo.com