DISCUSSION PAPER NO. 229

A MODIFIED CONJUGATE GRADIENT ALGORITHM

by

Avinoam Perry
Assistant Professor
Department of Decision Sciences

March 1976

Revised June 1976

Abstract

In this technical report we present a modification of the Fletcher-Reeves conjugate gradient algorithm. This modification results in an improved algorithm as reflected by the computational experience presented in this report.

Introduction

Conjugate direction algorithms for minimizing unconstrained nonlinear programs can be divided into two major classes. The first class consists of algorithms with no memory such as Fletcher-Reeves conjugate gradient algorithm (FRCG) [8], Polak-Rebière method (PRCG) [12,20], and the modified version of the conjugate gradient (PMCG) which is presented here. The second class consists of the quasi-Newton methods which apply a matrix update approximating the hessian inverse of f(x). Among the most popular quasi-Newton precedures we have the DFP update (Davidon [4], Fletcher and Powell [7]), BR1 update (Broyden's Rank one [1]), Pearson's algorithms [18], the BFGS update (Broyden [2], Fletcher [6], Goldfarb [9], Shanno [23]), and Huang general family [11].

Recent developments in the field of unconstrained optimization concentrate their efforts on algorithms with inaccurate or no line search (this is due to the fact that the line search part of an algorithm is the most time-consuming part). One of the most recent examples of an algorithm belonging to this class is the one developed by Davidon [5].

Optimal-conditioning and self-scaling procedures such as the ones developed by Oren and Luenberger [15,16], Oren and Spedicato [17], Shanno [23] and others [3], contribute greatly to the overall efficiency and local convergence properties of algorithms with inaccurate line search. However, some of these algorithms result in a departure from the pure* quadratic termination phenomenon [14,15].

---

*By pure quadratic termination we refer to algorithms which minimize a quadratic function in, at most, n + 1 steps.

Computational experience with quasi-Newton type algorithms leads to the conclusion that in order to eliminate severe accumulated rounding errors and ineffective updated matrices, the procedure should be restarted after a prespecified number of iterations. The most popular heuristics are the ones which restart the quasi-Newton procedure after every n or n+1 steps. Algorithms without memory utilize information obtained in steps k and k-1 only. As a result, the danger of accumulating rounding errors and constructing erroneous updates due to inaccurate line search and rounding error accumulation is reduced considerably. Naturally, under inaccurate line search conditions, one might expect conjugate direction algorithms with no memory to outperform the quasi-Newton methods. This expectation proves to be incorrect. In fact, our computational experience with conjugate direction algorithms suggests that the performance of the Fletcher-Reeves algorithm deteriorates quite rapidly as the accuracy of the line search is reduced. A similar conclusion, although somewhat less acute, can be reached upon examining the performance of the Polak-Rebière method. Nevertheless, the new version presented in this paper and the DFP and BFGS algorithms tend to show relative stability when line search accuracy is reduced.

The conclusions reached from the above observations are that the relatively poor performance of the Fletcher-Reeves conjugate gradient algorithm is not a result of its inability to accumulate information, but rather its strong dependence on problem structure. In the construction of the direction equation of the Fletcher-Reeves conjugate gradient the assumption: $\nabla f(x_k)' \nabla f(x_{k+1}) = 0$ is made explicitly.

This assumption always holds for quadratic programs with perfect line search, but does not hold under more general conditions.

The deterioration of the Polak-Rebière method under inaccurate line search conditions can be attributed to the fact that in the construction of its direction vector ≡ d_{k+1}, the assumption $d_k' \cdot \nabla f(x_{k+1})=0$ is made explicitly. This assumption always holds for quadratic programs with perfect line search while for nonquadratic programs with an "almost perfect" line search this assumption does not seem to be unreasonable. However, when line search accuracy is reduced, the above assumption does not hold, and performance of the method becomes unsatisfactory in most cases.

Relaxation of the above orthogonality assumptions and an additional correction leads to a modification of the Fletcher-Reeves and the Polak-Rebière equations. This modification results in a better algorithm, the superiority of which becomes more significant under inaccurate line search conditions.

Although the performance of our modified version of the conjugate gradient algorithm is somewhat less than competitive with the DFP and BFGS methods, it is, nevertheless, useful and attractive. This argumemt takes its justification from the field of constrained optimization. Recent versions of the GRG method [13] employ a modification of the BFGS algorithm which accomodates upper and lower bounds on the variables (see Goldfarb [10]). This may not be suitable for large-scale nonlinear programs because of the need for storing and updating the large matrix approximating the hessian inverse of f(x). Therefore, whenever storage is scarce, the BFGS algorithm is replaced by a modified version of the conjugate gradient method [22].

II. Derivation of the Modified Conjugate Gradient Algorithm

Let $p_k \equiv x_{k+1} - x_k$ and $q_k \equiv g_{k+1} - g_k$ where $g_k \equiv \nabla f(x_k)$ and the unconstrained nonlinear program is: minimize $f(x)$, $x \in E^n$.

Let $d_k \equiv (1/\alpha_k)p_k$ where α_k is a scalar minimizing the one dimensional program: $\min\limits_{\alpha \geq 0} \ f(x_k + \alpha d_k)$.

Conjugate directions in E^n have the property

(1) $p_k' F_k p_{k+1} = \alpha_k^2 d_k' F_k d_{k+1} = 0$

where F_k is the hessian of $f(x_k)$. If $f(x)$ is quadratic and F_k is constant (i.e., $F_k = F$) then (1) is equivalent to the requirement

(2) $q_k' p_{k+1} = \alpha_k q_k' d_{k+1} = 0.$

A conjugate gradient direction at stage k+1 is constructed by taking a linear combination of the negative gradient at stage k+1 and the direction vector at stage k.

(3) $d_{k+1} = -g_{k+1} + \beta_k d_k$

equation (2) implies

(4) $\beta_k = \dfrac{q_k' \, g_{k+1}}{q_k' \, d_k}$

and (3) becomes

(5) $d_{k+1} = -g_{k+1} + \dfrac{q_k' \, g_{k+1}}{q_k' d_k} \cdot d_k$

If f(x) is quadratic and α is computed with perfect accuracy we have

(6) $q_k' \, g_{k+1} = (g_{k+1} - g_k)' \, g_{k+1} = g_{k+1}' \, g_{k+1}$

and

(7) $q_k' \, d_k = (g_{k+1} - g_k)' \, d_k = (g_{k+1} - g_k)' \, (-g_k + \beta_k d_{k-1}) = g_k' \, g_k$

and (5) becomes

(8) $d_{k+1} = -g_{k+1} + \dfrac{g_{k+1}' \, g_{k+1}}{g_k' \, g_k} \, d_k$

(8) is the well known Fletcher-Reeves conjugate gradient direction [8].

Relaxing the assumption regarding the orthogonality of consecutive gradient vectors we can rewrite (5) as follows

(9) $d_{k+1} = -g_{k+1} + \dfrac{q_k' g_{k+1}}{(g_{k+1} - g_k)' d_k} \cdot d_k$

Assuming $d_k' \, g_{k+1} = 0$, then (9) becomes

(10) $d_{k+1} = -g_{k+1} - \dfrac{q_k' \, g_{k+1}}{g_k' \, d_k} \cdot d_k = -g_{k+1} - \dfrac{q_k' g_{k+1}}{g_k' (-g_k + \beta_{k-1} d_{k-1})}$

$$= -g_{k+1} + \frac{q_k' g_{k+1}}{g_k' \, g_k} \cdot d_k$$

(10) is the well known Polak-Rebière conjugate gradient direction [12,20].

Relaxing both orthogonality assumptions implied by (8) and (10) respectively, we obtain

$$(11)\quad d_{k+1} = -g_{k+1} + \frac{q_k' g_{k+1}}{q_k' d_k} \cdot d_k = -g_{k+1} + \frac{q_k' g_{k+1}}{q_k' p_k} \cdot p_k = -[I - \frac{p_k q_k'}{p_k' q_k}] g_{k+1}$$

Upon denoting the matrix $[I - \frac{p_k q_k'}{p_k' q_k}]$ as D_{k+1} (11) becomes

$$(12)\quad d_{k+1} = - D_{k+1} g_{k+1}$$

The matrix D_{k+1} is not of full rank and is, therefore, positive semi-definite rather than positive definite. Another important property which is a major characteristic of quasi-Newton methods is not present in D_{k+1}.

$$(13)\quad p_k' \neq q_k' D_{k+1} = 0$$

It is possible to correct these two defficiencies by adding the rank one matrix $\frac{p_k p_k'}{p_k' q_k}$ to D_{k+1}.

The new matrix

$$(14)\quad S_{k+1} = D_{k+1} + \frac{p_k p_k'}{p_k' q_k} = I - \frac{p_k q_k'}{p_k' q_k} + \frac{p_k p_k'}{p_k' q_k}$$

is of full rank and positive definite (given $p_k' q_k > 0$). It also possesses the desired property:

(15) $p_k' = q_k' S_{k+1}$

which typifies all algorithms of the Huang family [11].

Upon replacing D_{k+1} with S_{k+1} in the direction equation of (12) we obtain

(16) $$d_{k+1} = -S_{k+1} g_{k+1} = -\left[I - \frac{p_k q_k'}{p_k' q_k} + \frac{p_k p_k'}{p_k' q_k} \right] g_{k+1} = -g_{k+1} + \frac{(q_k - p_k)' g_{k+1}}{p_k' q_k} \cdot p_k$$

Denoting

(17) $$\gamma_k = \frac{(q_k - \alpha_k d_k)' g_{k+1}}{d_k' q_k}$$

we obtain a modified conjugate gradient equation

(18) $$d_{k+1} = -g_{k+1} + \gamma_k d_k$$

The modified conjugate gradient algorithm based on (18) posseses the property of quadratic termination. This is proved by the fact that for a given quadratic function f(x) and a perfect line search, the direction generated by the new method is identical to the one obtained by Fletcher-Reeves conjugate gradient and the DFP methods.

III. Computational Experience

Experiments with the new modified conjugate gradient algorithm (PMCG) as well as, Fletcher-Reeves (FRCG), Polak-Rebière (PRCG), BR1, DFP, and BFGS methods involved seven well known test problems which are denoted here as functions I through VII respectively (see appendix for a detailed description of each function and its source).

All problems were solved by each one of the algorithms above under two different line search accuracy measures. The line search technique applied is the well known quadratic interpolation method. In order to insure successful implementation of the line search procedure an effort was made to secure a unimodel region before the first interpolation was performed. We also define "number of iterations per line search" as the number of times the quadratic interpolation was performed along a given direction.

Another measure of accuracy used in our study is δ where:

$$(19)\quad \delta > \frac{|p(\alpha^*) - f(\alpha^*)|}{p(\alpha^*)}$$

and where:

$p(\alpha^*)$ is the value of the interpolating polynomial at the point $x_k + \alpha^* d_k$, and $f(\alpha^*)$ is the value of the function at the same point.

The term "stage" is defined as the step carrying a point x_k along a direction d_k to a new point $x_{k+1} = x_k + \alpha_k d_k$. It follows that the total number of stages per algorithm is equal to the total number of gradient evaluations.

The statistics "total number of function evaluations" includes the number of gradient evaluations multiplied by n (If one is interested in number of function evaluations not including gradient evaluations, the total number of stages multiplied by n should be subtracted from the total number of function evaluations). Each time a direction vector pointed upwards rather than downwards it was replaced by the direction of steepest descent. Although we do not provide statistics regarding the number of times per experiment this phenomenon took place, we wish to note that this procedure was a significant factor in determining the algorithmic mapping of BR1, as well as, FRCG and PRCG whenever line search accuracy measures were light.

The stopping rule applied throughout was

(20) $|\nabla f(x^*)| \leq 0.0001$

If an experimental run exceeded a given number of stages before reaching the point x^* in (20) the run was terminated by the operator.

All computer programs were coded in APL using interactive mode [19] and were run on the CDC6400 computer at Northwestern University.

In the following tables we present our computational results under two measures of line search accuracy. These measures are denoted as mode 1 ($N=5$, $\delta=.01$) and mode 2 ($N=1$) where the variable name N stands for the maximum number of iterations per one dimensional search, and δ is defined as in (19). The one dimensional search was terminated whenever one of these constraints became active.

Function 1
$x_0 = -1.2,\ 1$

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	65	206	581	1764	$2.67 \cdot 10^{-10}$	$7.46 \cdot 10^{-10}$
PRCG	23	31	253	325	$8.61 \cdot 10^{-14}$	$1.84 \cdot 10^{-13}$
PMCG	23	25	234	228	$1.87 \cdot 10^{-10}$	$1.93 \cdot 10^{-15}$
BR1	26	24	284	223	$5.34 \cdot 10^{-16}$	$1.31 \cdot 10^{-13}$
DFP	23	25	224	207	$1.91 \cdot 10^{-11}$	$6.05 \cdot 10^{-12}$
BFGS	20	25	209	216	$2.85 \cdot 10^{-13}$	$8.23 \cdot 10^{-11}$

Function 2
$x_0 = -1.2,\ 1$

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	8	7	85	69	$8.16 \cdot 10^{-9}$	$8.27 \cdot 10^{-9}$
PRCG	6	9	64	79	$3.53 \cdot 10^{-10}$	$7.48 \cdot 10^{-14}$
PMCG	5	6	60	62	$3.03 \cdot 10^{-12}$	$1.51 \cdot 10^{-12}$
BR1	5	6	58	55	$1.51 \cdot 10^{-10}$	$9.69 \cdot 10^{-10}$
DFP	5	7	56	68	$7.89 \cdot 10^{-11}$	$2.75 \cdot 10^{-15}$
BFGS	5	7	57	67	$1.76 \cdot 10^{-11}$	$9.84 \cdot 10^{-14}$

Function 3
$x_0 = -1.2,\ 1$

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	5	5	69	65	$4.70 \cdot 10^{-12}$	$3.69 \cdot 10^{-15}$
PRCG	5	5	60	62	$3.33 \cdot 10^{-17}$	$3.77 \cdot 10^{-14}$
PMCG	5	5	65	56	$5.86 \cdot 10^{-18}$	$2.02 \cdot 10^{-13}$
BR1	3	5	50	56	$4.93 \cdot 10^{-17}$	$3.17 \cdot 10^{-14}$
DFP	4	4	51	45	$8.59 \cdot 10^{-17}$	$5.35 \cdot 10^{-12}$
BFGS	3	4	50	45	$6.35 \cdot 10^{-18}$	$3.78 \cdot 10^{-11}$

Function 4
$x_0 = -1.2,\ 1$

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	13	27	161	264	$6.91 \cdot 10^{-11}$	$3.82 \cdot 10^{-10}$
PRCG	13	27	152	284	$6.30 \cdot 10^{-12}$	$3.05 \cdot 10^{-10}$
PMCG	13	23	138	206	$1.99 \cdot 10^{-13}$	$3.15 \cdot 10^{-15}$
BR1	16	24	220	254	$1.88 \cdot 10^{-9}$	$5.55 \cdot 10^{-12}$
DFP	15	24	168	226	$1.07 \cdot 10^{-12}$	$7.11 \cdot 10^{-16}$
BFGS	14	22	161	215	$1.66 \cdot 10^{-12}$	$1.66 \cdot 10^{-16}$

Function 5
x_0= -3,-1,-1,-1

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	1500*	1500*	14031*	13542*	$2.84 \cdot 10^{-3}$	$3.27 \cdot 10^{-1}$
PRCG	85	115*	870	1193*	$4.37 \cdot 10^{-10}$	$1.56 \cdot 10^{-4}$
PMCG	75	115	839	1263	$1.66 \cdot 10^{-10}$	$2.26 \cdot 10^{-9}$
BR1	74	207	765	2362	$8.12 \cdot 10^{-11}$	$1.08 \cdot 10^{-9}$
DFP	71	128	780	1489	$7.93 \cdot 10^{-12}$	$2.66 \cdot 10^{-16}$
BFGS	38	44	436	478	$4.54 \cdot 10^{-14}$	$1.11 \cdot 10^{-14}$

Function 6
x_0= 1,1,1,1

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	100*	100*	975*	962*	$4.23 \cdot 10^{-6}$	$3.29 \cdot 10^{-4}$
PRCG	97	100*	1060	1030*	$1.44 \cdot 10^{-9}$	$4.37 \cdot 10^{-6}$
PMCG	99	89	1085	959	$3.30 \cdot 10^{-9}$	$4.20 \cdot 10^{-9}$
BR1	100*	100*	1002*	1031*	$3.65 \cdot 10^{-4}$	$2.01 \cdot 10^{-4}$
DFP	16	24	201	231	$3.33 \cdot 10^{-9}$	$1.61 \cdot 10^{-10}$
BFGS	15	19	196	197	$1.63 \cdot 10^{-9}$	$5.79 \cdot 10^{-10}$

Function 6
x_0= 3,-1,0,1

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	100*	100*	1026*	1062*	$3.18 \cdot 10^{-5}$	$2.47 \cdot 10^{-6}$
PRCG	74	64	822	656	$7.10 \cdot 10^{-9}$	$7.22 \cdot 10^{-9}$
PMCG	78	61	891	664	$6.33 \cdot 10^{-9}$	$6.65 \cdot 10^{-9}$
BR1	100*	100*	1140*	1059*	$9.44 \cdot 10^{-5}$	$1.05 \cdot 10^{-5}$
DFP	20	19	253	214	$9.93 \cdot 10^{-10}$	$5.54 \cdot 10^{-9}$
BFGS	18	19	233	212	$3.02 \cdot 10^{-11}$	$3.07 \cdot 10^{-11}$

Function 7
x_0= 1, 1

Algorithm	# stages		# function evaluations		Reported best value	
	Mode 1	Mode 2	Mode 1	Mode 2	Mode 1	Mode 2
FRCG	10	15	112	140	$2.44 \cdot 10^{-11}$	$1.00 \cdot 10^{-11}$
PRCG	6	6	73	62	$1.39 \cdot 10^{-12}$	$1.31 \cdot 10^{-12}$
PMCG	6	6	72	59	$1.22 \cdot 10^{-12}$	$3.26 \cdot 10^{-12}$
BR1	6	7	72	72	$1.19 \cdot 10^{-12}$	$1.93 \cdot 10^{-13}$
DFP	6	6	73	66	$2.76 \cdot 10^{-15}$	$6.68 \cdot 10^{-18}$
BFGS	6	6	73	65	$2.74 \cdot 10^{-15}$	$1.24 \cdot 10^{-17}$

*Terminated by operator

Concluding remarks

As reflected by our computational results, the new modified conjugate gradient algorithm (PMCG) performed better than Fletcher-Reeves conjugate gradient method (FRCG) whose performance is the worse across the board. It also performed better than the Polak-Rebiere method in most cases but fell behind it only in one case associated with Powell's function (function 6) under the condition N=5, δ=.01 x_o=3,-1,0,1. It is also shown that PMCG exhibits more stability under varying degrees of line search measures of accuracy than either FRCG or PRCG and its performance seems to be better on the average. It is also interesting to note that the only significant case in which both the DFP and BFGS methods clearly outperformed PMCG is the case associated with Powell's function. The only case in which BFGS clearly outperformed every other method is the case associated with Wood's function. However, when comupter space is scarce due to large scale programs, then PMCG seems to be the best choice available.

Acknowledgement

I wish to express my sincere appreciation to professors L.S. Lasdon and M.A. Saundres whose comments on an earlier draft resulted in a significant improvement in the presentation of the paper.

Appendix

1: $100(x_2 - x_1^2)^2 + (1-x_1)^2$ [1]

2: $(x_2 - x_1^2)^2 + (1-x_1)^2$ [2]

3: $(x_2 - x_1^2)^2 + 100(1-x_1)^2$ [2]

4: $100(x_2 - x_1^3)^2 + (1 - x_1)^2$ [2]

5: $100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2$ [3]
$+ (1-x_3)^2 + 10.1 [(x_2-1)^2 + (x_4 - 1)^2]$
$+ 19.8 (x_2-1)(x_4-1)$

6: $(x_1 + 10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$ [4]

7: $(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ [5]

---

[1] H.H. Rosenbrock, Computer J., 3: 175 (1960)

[2] B.F. Whitte and W.R. Holst, paper submitted at the 1964 Spring Joint Computer Conference, Washington D.C., (1964).

[3] C.F. Wood, Westinghouse Research Laboratories, (1968).

[4] M.J.D. Powell, Computer J., 5: 147 (1962).

[5] D.M. Himmelblau, Applied Nonlinear Programming, pp. 428. McGraw-Hill (1972).

References

[1] Broyden, C.G., "Quasi-Newton Methods and Their Application to Function Minimization", Math. Comp., 21, 1967 pp. 368-381.

[2] Broyden, C.G., "The Convergence of a Class of Double Rank Algorithms, Parts I and II," J. Inst. Math. Appl., 7, 1971, pp. 76-90, 222-236.

[3] Broyden, C.G., J.E. Dennis, and J.J. More, "On the Local and Superlinear Convergence of Quasi-Newton Methods", J. Inst. Math Appl. 12, 1973 pp. 223-246.

[4] Davidon, W.C., "Variable Metric Method for Minimization", Research and Development Report ANL-5990, U.S. Atomic Energy Commission, Argonne National Laboratories, 1959.

[5] Davidon, W.C., "Optimally Conditioned Algorithms without Line Searches", Math. Prog. Vol. 9, No. 1, 1975, pp. 1-30.

[6] Fletcher, R. "A New Approach to Variable Metric Algorithms", Comp. J., 13, 1970, pp. 317-322.

[7] Fletcher, R. and M.J.D. Powell, "A Rapidly Convergent Descent Method for Minimization", Comp.J., 6, 1963, pp. 163-168.

[8] Fletcher, R. and C.M. Reeves, "Function Minimization by Conjugate Gradients", Comp. J., 7, 1964, pp. 149-154.

[9] Goldfarb, D., "A Family of Variable Metric Methods Derived by Variational Means", Math. Comp. 24, 1970, pp. 23-26.

[10] Goldfarb, D. "Extension of Davidon's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints", Siam J. Appl. Math. 17, No. 4, July 1969.

[11] Huang, H.Y., "Unified Approach to Quadradically Convergent Algorithms for Function Minimization J.O.T.A. 5, 1970, pp. 405-423.

[12] Klessig, R. and E. Polak, "Efficient Implementation of the Polak-Rebiere Conjugate Gradient Algorithm," Siam J. Control 10, 1972, pp. 524-549.

[13] Lasdon, L.S., A.D. Waren, A. Jain and M. Ratner, "Design and Testing of a Generalized Reduced Gradient Code For Nonlinear Programming", Technical Report SoL 76-3, Department of Operations Research, Stanford University, Feb. 1976.

[14] Luenberger, D.G., "Introduction to Linear and Nonlinear Programming", Addison-Wesley Publishing Co., 1973.

[15] Oren, S.S., "Self Scaling Variable Metric Algorithms, Part II", Management Science, Vol. 20, No. 5, 1974, pp. 863-874.

[16] Oren, S.S. and D.G. Luenberger, "Self Scaling Variable Metric Algorithms, Part I", Management Science, Vol. 20, No. 5, 1974, pp. 845-862.

[17] Oren, S.S. and E. Spedicato, "Optimal Conditioning of Self Scaling Variable Metric Algorithm", Math Prog. Vol 10. No. L Feb. 1976, pp. 70-90.

[18] Pearson, J.D., "Variable Metric Methods of Minimization", Comp. J., 13, 1969, pp. 171-178.

[19] Perry, Avinoam, "UCNLP-An Interactive Package of Programs for Unconstrained Nonlinear Optimization Purposes", A Working Paper, Nov. 1975, Revised May 1976.

[20] Polak, E. and G. Rebière, "Note Sur La Convergence de Methodes de Directions Conjugees", Revue Francaise Inf. Rech, Oper., 16 RI 1969, pp. 35-43.

[21] Powell, M.J.D. "Recent Advances in Unconstrained Optimization", Math. Prog. 1, 1971, pp. 26-57.

[22] Saundres M.A. and B.A. Murtagh, "Nonlinear Programming for Large, Sparse Systems" Technical Report, Department of Operations Research, Stanford University, June 1976.

[23] Shanno, D.F., "Conditioning of Quasi-Newton Methods for Function Minimization", Math. Comp. 24, 1970, pp. 647-656.