4^{th} GRACM Congress on Computational Mechanics
GRACM 2002
Patra, 27–29 June, 2002

A NON-MONOTONE CONIC METHOD FOR UNCONSTRAINED OPTIMIZATION

G. E. Manoussakis[1], D. G. Sotiropoulos[2], C. A. Botsaris[1], and T. N. Grapsa[2]

(1) Division of Statistics, Probability and Operational Research,
Department of Mathematics,
University of Patras,
GR–261.10 Patras, Greece,
e-mail :botsaris@otenet.gr

(2) Division of Computational Mathematics and Informatics,
Department of Mathematics,
University of Patras,
GR–261.10 Patras, Greece,
e-mail :grapsa@math.upatras.gr

Keywords : Unconstrained Optimization, Conic, Nonmonotone

Abstract

In this paper we present a new algorithm for finding the unconstrained minimum of a continuously differentiable function $f(x)$ in n variables. This algorithm is based on a conic model function, which does not involve the conjugacy matrix or the Hessian of the model function. The conic method in this paper is combined with a non-monotone line search using the Barzilai and Borwein step. The method does not guarantee descent in the objective function at each iteration. Also, the choice of step length is related to the eigenvalues of the Hessian at the minimizer and not to the function value. The use of the stopping criterion introduced by Grippo, Lampariello and Lucidi allows the objective function to increase at some iterations and still guarantees global convergence. The new algorithm converges in $n+1$ iterations on conic functions and, as numerical results indicate, rapidly minimizes general functions.

1. INTRODUCTION

The general problem we deal with is

$$\min f(x), \quad x \in \mathbb{R}^n$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$, is a continuously differentiable function in n variables $x = (x_1, x_2, \ldots, x_n)^\top$

Standard methods for unconstrained minimization are based on a quadratic model of the form:

$$q(x) = \frac{1}{2}(x-\beta)^\top Q(x-\beta) + q(\beta), \tag{1.1}$$

where β is the location of the minimum and Q is a $n \times n$ constant positive definite matrix of the second partial derivatives of $q(x)$, the Hessian.

Various authors in the past have introduced non-quadratic algorithms for function minimization. Davidon[5] introduced conic models for unconstrained optimixation. Botsaris and Bacopoulos [4] presented an algorithm based on a conic model which is conceptually different than those used by most of the conic methods so far, in that it does not involve the conjugacy matrix or the Hessian of the model function.

A conic function has the form :

$$c(x) = \frac{1}{2}\frac{x^\top Q x}{(1+p^\top x)^2} + \frac{b^\top x}{1+p^\top x} + a, \tag{1.2}$$

where Q is an $n \times n$ symmetric matrix and $p \in \mathrm{IR}^n$ is the vector defining the horizon of the conic function, i.e. the hyperplane where $c(x)$ takes an infinite value, which is defined by the equation :

$$1 \ + \ p^\top x \ = \ 0. \tag{1.3}$$

The function $c(x)$ has a unique minimum whenever Q is positive definite and then the location β of the minimizer is determined by solving the equation :

$$\frac{\beta}{1 + p^\top \beta} + Q^{-1}\beta = 0, \tag{1.4}$$

or equivalently :

$$\beta = -\frac{Q^{-1}\beta}{1 + p^\top Q^{-1}\beta}, \tag{1.5}$$

provided that such a solution exists, i.e. $1 + p^\top Q^{-1}\beta \neq 0$.

The conic can be brought in the form :

$$c(x) = \frac{1}{2}\ \frac{1 + p^\top x}{1 + p^\top \beta}\ \nabla c(x)\ (x - \beta) \ + \ c(\beta), \tag{1.6}$$

As can be seen the conic model in this form does not involve and, therefore, it does not require, an estimate of the conjugacy matrix Q or the Hessian matrix of the objective function.

Assuming that the objective function is conic we obtain :

$$\left[2f(x)p^\top + (1 + p^\top x)g^\top(x)\right]\beta \ - \ 2(1 + p^\top \beta)f(\beta) = (1 + p^\top x)g^\top(x)x - 2f(x). \tag{1.7}$$

If the horizon p is known, then by calculating (1.7) at $n+1$ distinct points $x_1, x_2, \ldots, x_{n+1}$ we have an $(n+1) \ \times \ (n+1)$ linear system :

$$A\alpha = s - \phi \quad , \quad \alpha = \begin{bmatrix} \beta \\ \hline \omega \end{bmatrix} \tag{1.8}$$

where

$$A \ = \ \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n+1} \end{pmatrix} \ \begin{bmatrix} p^\top & 0 \end{bmatrix} \ + \ G \tag{1.9}$$

$$G \ = \ \begin{pmatrix} (1 + p^\top x_1) {g_1}^\top & -1 \\ (1 + p^\top x_2) {g_2}^\top & -1 \\ \vdots & \\ (1 + p^\top x_{n+1}) {g_{n+1}}^\top & -1 \end{pmatrix} \tag{1.10}$$

$$s \ = \ \begin{pmatrix} (1 + p^\top x_1) {g_1}^\top x_1 \\ (1 + p^\top x_2) {g_2}^\top x_2 \\ \vdots \\ (1 + p^\top x_{n+1}) {g_{n+1}}^\top x_{n+1} \end{pmatrix} \tag{1.11}$$

$$\phi \ = \ 2 \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n+1} \end{pmatrix} \tag{1.12}$$

$$\omega \ = \ 2(1 + p^\top \beta) f(\beta) \tag{1.13}$$

This system is linear in the $n+1$–dimentional unknown vector α consisting of the location of the minimum β and the scaled value of the minimum ω, assuming that the gauge vector p is available.

A recursive method to solve the system above is presented in [4]

In [3] a gradient method has been proposed where the search direction is always the negative gradient direction, but the choice of the step length is not the classical choice of the steepest descent method. The motivation for this choice is that it provides two-point approximation to the secant equation underlying quasi-Newton methods [14] . This yields the iteration :

$$x_{i+1} \;=\; x_i \;-\; t_i g(x^i), \tag{1.14}$$

where the step t_i is given by :

$$t_i \;=\; \frac{\delta_{i-1}^\top \delta_{i-1}}{\delta_{i-1}^\top \psi_{i-1}} \tag{1.15}$$

where $\delta_{i-1} = \Delta x = x_i - x_{i-1}$, and $\psi_{i-1} = \Delta g = g_i - g_{i-1}$.

In [9] a nonmonotone line search for Newton–type methods has been proposed and in [13],[17] some computational advantages of this technique have been pointed out. The method imposes that the objective function value f of each iteration must satisfy the Armijo's condition with respect to the maximum objective function value of a prefixed number M of previous iterations. Formally, the condition is given by :

$$f(x_k - t_k \nabla f(x_k)) - \max_{0\le j\le M} \{f(x_{k-j})\} \le -\sigma t_k \|\nabla f(x_k)\|^2 \tag{1.16}$$

where M is a nonnegative integer, and $0 < \sigma < 1$. The above condition allows an increase in the function values without affecting the global convergence properties [13],[17]. This method has low storage requirements and inexpensive computations.

2. THE NEW METHOD

In order to find the point β which minimizes a given function $f\colon \mathcal{D} \subset \mathrm{I\!R}^n \to \mathrm{I\!R}$ in a specific domain $\mathcal{D}$, we shall try to obtain a sequence of points $x^k,\ k = 0, 1, \dots$ which converges to β.

Following the same approach as in [4], and assuming that f is a conic function, then we have to solve the system defined by 1.8

Let

$$\rho_i = \frac{1 + p^\top x_{i+1}}{1 + p^\top x_i} \tag{2.1}$$

Then

$$\rho_i = \frac{\Delta f + k_i}{g_{i+1}^\top \Delta x} \tag{2.2}$$

where $\Delta f = f_{i+1} - f_i$, $\Delta x = x_{i+1} - x_i$, and

$$k_i = \left[\Delta f^2 - g_{i+1}^\top \Delta x g_i^\top \Delta x\right]^{\frac{1}{2}} \tag{2.3}$$

provided that the quantity under the square root is non-negative, which is true for a suitable choice of the step length in the line search algorithm.

The gauge vector p can be determined by solving the linear system :

$$Zp = r \tag{2.4}$$

where $z_i = x_{i+1} - \rho_i x_i$, $Z \;=\; \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}$, $\eta_i = \rho_i - 1$, $r \;=\; \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{pmatrix}$.

From 1.8, 2.4, we have that the location of the minimum β can be determined through the equation :

$$p = Z^{-1} r, \quad \alpha = A^{-1}(s - \phi) \tag{2.5}$$

We carry out the inversions in Equation recursively as new points are constructed by the algorithm. Using Householder's formula for matrix inversion it can be verified that :

$$Z_i^{-1} \;=\; Z_{i-1}^{-1} \;-\; \frac{Z_{i-1}^{-1} e_l (z_i^\top Z_{i-1}^{-1} - e_l^\top)}{z_i^\top Z_{i-1}^{-1} e_l} \tag{2.6}$$

and

$$p_i \;=\; p_{i-1} \;+\; \frac{Z_{i-1}^{-1} e_l (\eta_i - z_i^\top p_{i-1})}{z_i^\top Z_{i-1}^{-1} e_l} \tag{2.7}$$

provided that $|z_i^\top Z_{i-1}^{-1} e_l|$is bounded away from zero. We note that e_l is a vector with zero elements except the position $l = i$, where it has unity.

The solution to the linear system is proved to be :

$$\alpha \;=\; u \;-\; \left(\frac{1 + q^\top u}{1 + q^\top v}\right) v \tag{2.8}$$

where

$$q^\top \;=\; \begin{bmatrix} p^\top & 0 \end{bmatrix}, \tag{2.9}$$

$$u \;=\; G^{-1} s \tag{2.10}$$

and

$$v \;=\; G^{-1} \phi. \tag{2.11}$$

Let us further define :

$$\Lambda_i \;=\; \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & \lambda_i \end{pmatrix}, \quad \lambda_i = \frac{1 + p_i^\top x_i}{1 + p_{i-1}^\top x_i} \tag{2.12}$$

and

$$y_{i+1}^\top \;=\; \left[\frac{1 + p_i^\top x_{i+1}}{\lambda_i g_{i+1}^\top} \quad -1 \right]. \tag{2.13}$$

Then G_{i+1}^{-1} can be computed according to the recursive formula :

$$G_{i+1}^{-1} \;= \frac{\Lambda_i}{\lambda_i} \left[G_i^{-1} \;-\; \frac{G_i^{-1} e_c (y_{i+1}^\top G_i^{-1} - e_c^\top)}{y_{i+1}^\top G_i^{-1} e_c} \right] \tag{2.14}$$

provided that $|y_{i+1}^\top G_i^{-1} e_c|$ is bounded away from zero.

The recursive equations for the vectors u and v, required to compute α_{i+1} from 2.8 , are found to be :

$$u_{i+1} \;= \Lambda_i \left[u_i \;-\; \frac{G_i^{-1} e_c (y_{i+1}^\top u_i - \theta_{i+1})}{y_{i+1}^\top G_i^{-1} e_c} \right] \tag{2.15}$$

and

$$v_{i+1} \;= \frac{\Lambda_i}{\lambda_i} \left[v_i \;-\; \frac{G_i^{-1} e_c (y_{i+1}^\top v_i^{-1} - \xi_{i+1})}{y_{i+1}^\top G_i^{-1} e_c} \right] \tag{2.16}$$

where

$$\theta_{i+1} = \frac{1 + p_i^\top x_{i+1}}{\lambda_i g_{i+1}^\top x_{i+1}} \;, \quad \xi_{i+1} = 2 f_i + 1 \tag{2.17}$$

So the following algorithm may now be stated :

ALGORITHM *Non-Monotone Conic Method.*

Step 1. Assume x_0 is given. Set i=0.

Step 2. Set $\alpha = 0.1$; $k = 0$; $j = 0$; $W(j) = f_0$.

Step 3. Set $d_0 = -g_0$, and $x_1 = x_0 - t_0 d_0$, where t_0 is determined by the nonmonotone line search.

Step 4. Set $\alpha_0^\top = \begin{bmatrix} x_1^\top & 0 \end{bmatrix}$, $G_0 = I$, $Z_0 = I$, $u_0 = \alpha_0$, $v_0 = 0$, $p_0 = 0$, $\lambda_0 = 1$, c=1, l=1.

Step 5. If $\|g_0\| \le \epsilon_1 (1 + |f_0|)$ then stop; else go to Step 6.

Step 6. Use 2.12 , 2.13 to calculate L_{i+1}, y_{i+1}.

Step 7. If $|y_{i+1}^\top G_I^{-1} e_c| < \epsilon_3$, then set $x_0 = x_{i+1}$ and go to Step 3; else go to Step 8.

Step 8. Use 2.7 , 2.8 , 2.9 , 2.14 , 2.15, 2.16 to calculate G_{i+1}^{-1}, u_{i+1}, v_{i+1}, a_{i+1}.

Step 9. If $|(x_i - \beta_i)^\top g_i| < \epsilon_4$ then then set $x_0 = x_{i+1}$ and go to Step 3; else go to Step 10.

Step 10. If $f(\beta_{i+1}) \leq f_{max}$, then store the function value in the record, set $x_{i+1} = \beta_{i+1}$, and go to step 11; else go to step 3.

Step 11. Set $i = i + 1$; If $c = n + 1$, then reset $c = 1$; else $c = c + 1$.

Step 12. Set $d_i = \mu_i(x_i - \beta_i)$,where $\mu_i = -sign\ g_i^\top(x_i - \beta_i)$.

Step 13. If $\|d_i\| + |\gamma_i| \leq M$, then determine the Barzilai step t_i, set $x_{i+1} = x_i + t_i d_i$ and go to step 14; else set $x_0 = x_i$ and go to Step 3.

Step 14. If $\delta_i = (f_{i+1} - f_i)^2 - g_{i+1}^\top(x_{i+1} - x_i) g_i^\top(x_{i+1} - x_i) < 0$, then using the nonmonotone line search take a step from x_{i+1} along d_i and repeat this procedure until the new x_{i+1} so obtained satisfies $\delta_I > 0$; go to Step 15.

Step 15. If $|z_i^\top Z_{i-1} e_l| < \epsilon_3$, then set $x_0 = x_{i+1}$ and go to Step 3; else go to Step 16.

Step 16. Use 2.6 to calculate Z_{i+1}^{-1}, p_{i+1}

Step 17. If $l = n$, then reset l=1; else set $l = l + 1$.

Step 18. Go to Step 5.

3. THE CONVERGENCE OF THE NEW METHOD

Convergence of our algorithm can be established using the following algorithm model, which searches for a point with a desirable property, and two theorems by Polak and Botsaris.

Algorithm Model Let A be a set-valued search function mapping a closed subset T of a Banach space B into the set of all nonempty subsets of T (which is written as $A : T \to 2^T$) and let $h = T \to \Re^1$.

Step 1. Compute $x_0 \in T$.

Step 2. Set $I = 0$.

Step 3. Compute a point $y_i \in A(x_i)$.

Step 4. Set $x_{i+1} = y_i$.

Step 5. If $h(x_{i+1} \geq h(x_i)$, then stop; else set $i = i + 1$ and go to Step 3.

THEOREM *Suppose that:*

(i) $h(x)$ is either continuous at all nondesirable points $x \in T$ or else $h(x)$ is bounded from below for $x \in T$; and

(ii) for every $x \in T$ which is nondesirable there exists an $\epsilon(x) > 0$ and a $\delta(x) < 0$ such that $h(x'') - h(x') \leq \delta(x) < 0$ for all $x' \in T$ such that $\|(x' - x)\| \leq \epsilon(x)$ and for all $x'' \in A(x')$.

Then either the sequence $\{x_i\}$ constructed by the above algorithm model is finite and its last element is desirable, or else the sequence is infinite and every accumulation point of $\{x_i\}$ is desirable.

Proof. See Polak in [14].

Consider the sequence $\{x_i\}$ generated by our algorithm. We make the following assumptions:

(a) x_i is desirable iff $\|g(x_i)\| = 0$.

(b) $f(x)$ is a continuously differentiable function in $\Re^n$.

(c) There exists an $x_0 \in \Re^n$ such that the level set $L_0 = \{x \mid f(x) \leq f(x_0)\}$ is compact. Consider the sequence $\{\bar{x}_i\}$ such that $f(\bar{x}_i) = f_{max}$ at the i iteration. The compactness assumption, together with the fact that because of the rule $f(\bar{x}_i + 1) \leq f(\bar{x}_i)$, implies that the subsequence $\{\bar{x}_i\}$ generated by our algorithm is bounded and therefore has accumulation points. Therefore also the sequence $\{x_i\}$ has accumulation points.

(d) Let $M > 0$ such that $M - 2 \geq sup\|g(x)\|, \quad x \in L_0$.

(e) The algorithm terminates if $\|g(x_i)\| = 0$.

Then either the sequence is finite and terminates at a desirable point or else it is infinite and every accumulation point x^* of $\{x_i\}$ is desirable.

Main Theorem *The sequence $\{x_i\}$ generated by our algorithm is finite and terminates at a desirable point or else it is infinite and every accumulation point x^* of $\{x_i\}$ is desirable.*

Proof. See Bacopoulos and Botsaris in [4].

4. NUMERICAL APPLICATIONS

The algorithm described in Section 3 has been implemented using the new FORTRAN program named MONCON. MONCON was tested a Pentium III PC compatible with random problems of varying dimensions. Our experience is that the algorithm behaved predictably and reliably and the results were quite satisfactory. Some typical computational results are given below. For the following problems, the reported parameters are :

— n dimension,
— $x^0 = (x_1, x_2, \ldots, x_n)$ starting point,
— $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ approximate local minimum computed within an accuracy of $\varepsilon = 10^{-4}$,
— IT the total number of iterations required in obtaining x^*,
— TE the total number of function and gradient evaluations.

In Tables 1–4 we compare the numerical results obtained, for various starting points, by applying other methods (Armijo's quadratic method[1], ARMBIS[19], Fletcher-Reeves[8], Polak-Ribiere[15], DFP[15]) including the classic conic method, with the corresponding numerical results of the method presented in this paper. In these tables x^0 indicates the starting point while the index α indicates the classical starting point. We used for all methods an accuracy of $\varepsilon_1 = 10^{-8}$ and an initial step 10. We also used $\varepsilon_3 = \varepsilon_4 = 10^{-15}$. For our method we set the size of the line search record to be $M = n$. Furthermore, D indicates divergence or non convergence after 10000 iterations.

Example 4.1. *Rosenbrock function*[11]. This example gives the local minimum for the function :

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \tag{4.1}$$

with $f(x^*) = 0$ at $x^* = (1,\ 1)$. (See table 1)

Example 4.2. *Extended Powell singular function*[11]. In this case we obtain the local minimum for the function :

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \tag{4.2}$$

Here we have $f(x^*) = 0$ at the origin. (See table 2)

Example 4.3. *Watson function* [11]. In this case f is given by :

$$f(x) = \sum_{i=1}^{m} {f_i}^2(x), \quad m = 31, \tag{4.3}$$

where :

$$f_i(x) = \sum_{j=2}^{n} (j-1) x_j t_i^{j-2} - \left(\sum_{j=1}^{n} x_j t_i^{j-1} \right)^2 - 1, \quad 1 \le i \le 29, \quad t_i = i/29, \tag{4.4}$$

$$f_{30}(x) = x_1, \quad f_{31}(x) = x_2 - x_1^2 - 1, \tag{4.5}$$

(See table 3)

Example 4.4. *Brown and Dennis function*[11]. This example gives the local minimum for the function :

$$f(x) = \sum_{k=1}^{20} \left[(x_1 + t_k x_2 - exp\,[t_k])^2 + (x_3 + x_4 sin(t_k) - cos(t_k))^2 \right]^2, \tag{4.6}$$

where $t_k = \frac{k}{5}$ with $f(x^*) = 85822{,}2$. (See table 4)

Table 1: Rosenbrock function

x^0	Armijo		ARMBIS			FR		PR		DFP		Conic		MONCON	
	IT	TE	IT	TE	AS	IT	TE	IT	TE	IT	TE	IT	TE	IT	TE
$(-1.2, 1)^\alpha$	794	8901	957	5303	19700	193	3211	19	364	23	355	20	97	33	118
$(-1.2, -1)$	745	8283	485	2800	13584	39	615	15	278	15	235	80	341	35	117
$(0, -1.2)$	778	8697	484	2807	15258	31	469	17	315	16	233	41	181	40	156
$(3, 3)$	1325	15824	519	2989	14195	134	2992	26	509	26	395	D		133	675
$(10, -10)$	782	8750	711	3983	16166	31	511	16	308	18	271	64	278	27	114

Table 2: Extended Powell singular function

x^0	Armijo		ARMBIS			FR		PR		DFP		Conic		MONCON	
	IT	TE	IT	TE	AS	IT	TE	IT	TE	IT	TE	IT	TE	IT	TE
$(3, -1, 0, 1)^\alpha$	1197	16515	710	6799	23237	8660	122575	5361	76052	32	548	648	2799	405	1536
$(1, -1, 1, -1)$	1184	16316	598	5712	20062	6105	86593	5234	74220	28	475	529	2227	471	1821
$(0, 10, 20, 30)$	1319	18379	794	7246	27109	9884	139879	542	8668	34	601	751	3221	300	1120
$(0, 1, 2, 3)$	1262	17505	635	6194	24992	9943	140655	262	4456	27	478	939	4086	111	389
$(100, 100, 100, 100)$	1230	17016	D			5913	84033	5090	72243	36	638	1090	4686	388	1415

Table 3: Watson function, $n = 2$

x^0	Armijo		ARMBIS			FR		PR		DFP		Conic		MONCON	
	IT	TE	IT	TE	AS	IT	TE	IT	TE	IT	TE	IT	TE	IT	TE
$(0, 0)^\alpha$	48	352	20	119	510	25	476	10	157	11	219	12	38	12	38
$(1, -1)$	56	421	3	15	159	27	542	12	211	12	230	11	36	11	36
$(10^6, 0)$	1361	16300	D			D		D		D		96	341	80	255
$(1.5, 0.5)$	209	1946	4	20	210	11	211	14	246	14	229	12	39	15	47
$(1, -2)$	110	927	3	15	162	21	354	15	266	14	219	11	38	18	51

Table 4: Brown and Dennis function

x^0	Armijo		ARMBIS			FR		PR		DFP		Conic		MONCON	
	IT	TE	IT	TE	AS	IT	TE	IT	TE	IT	TE	IT	TE	IT	TE
$(25, 5, -5, -1)^\alpha$	57	544	10	98	1020	35	605	35	602	54	893	90	454	51	177
$(0, 0, 0, 0)$	54	511	10	98	960	20	363	19	338	30	480	82	325	43	153
$(1, 2, 4, 8)$	72	709	15	146	1496	30	523	18	316	24	418	93	403	45	160
$(-5, -1, 1, 5)$	72	708	13	126	1407	33	565	30	516	20	349	D		45	163
$(-100, 0, 100, 1)$	113	1182	15	147	2050	36	652	36	665	33	578	D		71	239

5. Concluding Remarks and Further Research

In this paper we presented a new method of optimization which combines a nonmonotone line search with the conic approximation of the objective function. The new method has the advantages of both. That is a fast convergence rate with a relatively small cost in function and gradient evaluations. Furthermore the method does not need second order information nor matrix inversions, although in most cases it performs better than other methods which do.

References

[1] Armijo, L. (1966), "Minimization of functions having Lipschitz continuous first partial derivatives", Pacific J. Math., 16, 1–3.

[2] Avriel, M. (1976), *Nonlinear Programming, Analysis and Methods*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey.

[3] Barzilai, J. and Borwein, J.M. (1988), "Two-point step size gradient methods", IMA J. Num. Anal., 8, 141–148.

[4] Bacopoulos A. and Botsaris, C.A. (1992), "A new conic method for unconstrained minimization", J. Math. Anal. Appl., 167, 12–31.

[5] Davidon, W.C. (1959), *Variable metric methods for minimization*, A. E. C., Research and Development Report, No. ANL–5990, Argonne Nat'l Lab., Argonne, Illinois.

[6] Dennis Jr., J.E. and Schnabel, R.B. (1983), *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey.

[7] Fletcher, R. and Powell, M. (1963), "A rapidly convergent descent method for minimization", Comput. J., 6, 163–168.

[8] Fletcher, R. and Reeves, C. (1964), "Function minimization by conjugate gradients", Comput. J., 7, 149–154.

[9] Grippo, L., Lampariello, F. and Lycidi, S. (1986), "A nonmonotone line search technique for Newton's method", SIAM J. Numer. Anal., 23, 707–716.

[10] Luenberger, D.G. (1973), *Introduction to linear and nonlinear programming*, Addison–Wesley Publishing Company, Inc., Reading, Massachusetts.

[11] Moré, B.J., Garbow, B.S. and Hillstrom, K.E. (1981), "Testing unconstrained optimization", ACM Trans. Math. Software, 7, 17–41.

[12] Ortega, J.M. and Rheinbolt, W.C. (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

[13] Plagianakos, V.P., Sotiropoulos, D.G. and Vrahatis, M.N. (1998), "A Nonmonotone Backpropagation Training Method for Neural Networks", Dept. of Mathematics, Univ. of Patras, Technical Report No.98-04.

[14] Polak, E. (1971), *Computational Methods in Optimization*, Academic Press, New York.

[15] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1992), *Numerical Recipes, The Art of Scientific Computing*, 2nd ed., Cambridge University Press, New York.

[16] Rao, S.S. (1984), *Optimization, Theory and Applications*, Wiley Eastern Limited, New Delhi.

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

[18] Varga, R. (1962), *Matrix Iterative Analysis*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey.

[19] Vrahatis, M.N., Androulakis, G.S. and Manoussakis, G.E. (1996), "A New Unconstrained Optimization Method for Imprecise Function and Gradient Values", J. Math. Anal. Appl., 197, 586-607.