V. DUMITRU
Department of Economic Cybernetics
Academy of Economic Studies

Gradient Methods for Unconstrained Optimization

Under this title we include the methods in which the optimum is successively approximated by moves on the direction of the gradient or on directions linked in a certain way to gradient. After some of the most important gradient methods in unconstrained optimization are presented, numerical results are given on the efficiency of algorithms in solving four test problems, known in literature through their degree of difficulty. Some new variants of mixed type algorithms and a generalized Fletcher-Reeves formula are also presented.

1. *Let us consider the following problem:*

$$min\ f(x),\ x = (x_1,\ x_2,...,x_n) \in E^n \tag{1.1}$$

where E^n represents the n dimensional Euclidian space. On function $f(x)$ various hypotheses are made which will be specified for each method taken separately. As is known, a necessary condition for x to be a local optimum, for example, a minimum, is that $\nabla f(x) = 0$. Nevertheless, this condition is not sufficient except the case when $H = \| h_{ij} \|$, $h_{ij} = \dfrac{\partial^2 f(x^*)}{\partial_{x_i}\, \partial_{x_j}}$ is positive definite or what is equivalent to say that the function is locally convex in the neighbourhood of x^*. If the function is convex in the whole space E^n, x^* is a global optimum. By $\nabla f(x)$ we note the gradient of $f(x)$, that is the vector of components:

$$\nabla^T f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \right).$$

In the following we suppose that a local minimum is seeked, at a finite distance, through iterative methods by setting out from a point x°. It is noticed that the problem of unconstrained minimization is a particular case of the problem of constrained minimization

$$min\ f(x),\ x \in S \subset E^n \tag{1.2}$$

for $S = E^n$.

A general method for solving this problem and consequently the problem (1.1) is the method of admissible directions [1], [2] which, generally can be described [2], as follows:

a) Start with $x^\circ \in S$, $f(x^\circ) < \mu^\circ$.

b) Let us suppose that points $x^h \in S$, $h \leqslant k$, have been determined. Then, at iteration $k+1$ point x^{k+1} is determined as follows:

1. A direction d^k is stablished so that the function should decrease on the direction d^k:

$$\nabla^T f(x^k).\quad d^k < 0.$$

2. A length of the step λ_k on the direction d^k is determined so that:

$$0 < \lambda_k \leqslant \lambda_a^k$$

$$0 < \lambda_k \leqslant \lambda_m^k$$

where: λ_a^k is the highest value of λ for which point $x^k + \lambda^k \in S$, and λ_m^k is the value of λ for which

$$\min_{\lambda < 0} \{f(x^k + \lambda d^k)\} = f(x^k + \lambda_m^k d^k).$$

3. Point $x^{k+1} = x^k + \lambda_k d^k$ is established.

c) The stop criterion is checked. If it is not verified $k := k+1$ and the process is repeated from b). Obviously, for the unconstrained problem, S representing the whole space $\lambda_a = \infty$ and $0 < \lambda_k \leqslant \lambda_m^k$. Moreover, besides specifying otherwise, we suppose that the directional optimization was done completely, namely $\lambda_k = \lambda_m^k$. Usually λ_m^k is determining the first local minimum on d_k.

The following convergence theorem is demonstrated in [2]

Theorem 1: If

a) $f(x)$ is twice differentiable in $S(x^\circ) = \{x \mid f(x) \leqslant f(x^\circ)\}$

b) $S(x^\circ)$ is bounded.

c) $\Sigma\, \theta_{k_i}^2$ is divergent for any convergent subsequence x^{k_i} of the sequence x^k where $\theta_k = \dfrac{-\nabla f(x^k)^T d^k}{|\nabla f(x^k)\,\|\, d^k|}$

d) $H(x)$ is bounded in $S(x^\circ)$, where $H(x)$ is the matrix of the partial derivates of the second order of function $f(x)$.

Then $\nabla f(x^*) = 0$ for any accumulation point x^* of sequence x^k.

2. *Cauchy's method*

If in the method of admisible directions $d^k = -\nabla f(x^k)$ we have the Cauchy's optimization method of steepest descent [4]. Though Cauchy presented his method already in 1847, its convergence, for continuous and differential functions, was demonstrated only in 1944 by Curry [4]. We notice that the problem of convergence of the steepest descent is solved immediately as a direct consequence of theorem 1. Indeed, in case that $d^k = -\nabla f(x^k)$, $\theta_k = 1$, for any k from where $\Sigma\theta_k^2 = \infty$. Nevertheless, the method is convergent without supposing the existence of partial derivatives of the second order; in other words, the method is a method of the first order. (An algorithm applying only to partial derivates up to the order n is called a method of the order n.)

Curry's theorem: If $\nabla f(x)$ is continuous, in this case any limit point of the sequence of points generated by the method of steepest descent is a stationary point.

In case that $f(x)$ is a quadratic form positive definite, Kantorovich pointed out that the convergence rate of Cauchy's gradient method is at least geometrical. More precisely, if λ_1 and λ_n are the lowest and respectively highest characteristic value of the matrix A of quadratic $f(x) = \frac{1}{2} x^T A x + bx + c$

we have

$$f(x^{k+1}) \leqslant \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 f(x^k) \tag{2.1.}$$

Forsythe in [6] has demonstrated that relation (2.1) takes place with an equal sign. Even in this case, formula (2.1) shows that the convergence process is very slow if λ_n is very big in comparison with λ_1.

Apart from the "zig-zag" phenomenon generated by the fact that the determination of λ_k as min $f(x^k + \lambda_k d^k)$ leads to the condition

$$\nabla f(x^{k+1}) \cdot \nabla f(x^k) = 0 \tag{2.2}$$

equivalent to the fact that the successive directions d^k and d^{k+1} are orthogonal, a "zig-zag" phenomenon justifying a weak convergence speed, the steepest descent method also presents the inconvenience of greater instability to perturbations which may appear as a result of the rounding off errors or of the difficulty in finding out with exactitude the unidimensional minimum [7].

Obviously, various algorithms can be inferred from the presented algorithmic scheme according to the way in which either the values of parameter λ or the directions d^k are chosen.

3. *In the subrelaxation method* [8] *we have*:

$$x^{k+1} = x^k + \alpha\lambda_m^k d^k \tag{3.1}$$

$$d^k = -\nabla f(x^k),\ \lambda_m^k:\ \min_{\lambda} f(x^k + \lambda \cdot d^k) = f(x^k + \lambda_m^k d^k),\ 0 < \alpha < 1.$$

It can be seen that the method differs from the optimization gradient method only by the length of the step. It seems from experiments that a good value for α is $\alpha = 0.9$ and that the method would be superior, to some extent,

to the optimization gradient, maybe to the extent to which the algorithm remaining convergent, the diminishing of λ reduces the "zig-zag" phenomenon.

A problem which remains open is the establishment of an inferior limit under which λ should not be reduced.

4. Another method very close to the optimization gradient method is the diagonal method [9], of Forsythe and Motzkin:

$$d^k = -\nabla f(x^k),\ k \neq m \cdot p. \tag{4.1}$$

$$d^k = x^k - x^{k-2},\ k \equiv 0\ (mod\ p) \tag{4.2}$$

the other elements of the algorithm remaining identical as in the algorithm of the steepest descent. It is specified in [10] that according to the experiments reported in [9] the best strategy for carrying out the movement on the diagonal would be for $p = 7$.

5. *The method of conjugate directions*

When the minimizing function is a positive definite quadratic form

$$f(x) = \frac{1}{2} x^T\ Ax + bx + c^{1)} \tag{5.1}$$

a relatively large class of algorithms is that in which the displacement directions are mutually conjugate to matrix A:

$$d^k\ A\ d^{k-i} = 0,\quad i = 1\ ,.,., k-1. \tag{5.2}$$

The method of conjugate directions was initially elaborated as a method of conjugate gradients by Hestenes and Stiefel [11] for solving the systems of linear equations.

As in the case of positive definite quadratic forms this class of algorithms has the property Q_n [12], that is minimizes the quadratic form in at most n iterations, and it is supposed that these algorithms will have a better convergence speed even if applied to non-quadratic functions.

The demonstration results immediately as the consequence of the following theorem:

Theorem 2: Consider x the point determined by k successive minimizations of the quadratic forms (5.1) on the directions $d^0, d^1,...,d^{k-1}$, multually conjugate with matrix A.

$$\text{In this case } d^s\ \nabla f(x^k) = 0,\ s = 0,1...,(k-1). \tag{5.3}$$

Corrollary: As directions d^j being mutually conjugate form a system of independent linear vectors it results that after n minimizations $\nabla f(x^n)$ will be orthogonal at n independent linear vectors, thus $\nabla f(x^n) \equiv 0$ and thus an algorithm of the class of conjugate directions will have the property Q_n.

In the following, we shall make the notations: $g^k = g(x^k) = \nabla f(x^k)$.

---

[1] To avoid a rather complicated writing when the danger of a confusion is small, we shall no longer write the transposition sign so that $x^T A x$ will appear as $x A x$.

An algorithm of the class of conjugate directions will consequently have the following form [2]:

1. Start with an x° and d° satisfying $(g^\circ)^T \cdot d^\circ < 0$.

2. If $x^h, g^h\ d^h, h \leqslant h$ have already been determined, then:

a. we calculate λ_k as $min\ f(x^k + \lambda d^k) = f(x^k + \lambda_k d^k)$
b. we determine $x^{k+1} = x^k + \lambda_k d^k$ and g^{k+1};
c. we determine a direction d^{k+1} so that:

i) In case the minimizing function is a quadratic form (5.1), direction $d^{k\ 1}$ be mutually conjugated to all preceding directions:

$$(d^k)^T \cdot A \cdot d^{k+1} = 0, \quad \text{for } h = 0,1,..., k;$$

ii) d^K is a descent direction: $(g^{k+1})^T\ d^{k+1} < 0$.

3. Stop at the moment when conditions c_i or c_{ii} can no longer be fulfiled.

6. *The Fletcher and Reeves algorithm*

It is natural to try to establish directions d^{k+1} according to a relation of the form:

$$d^{k+1} = -g^{k+1} + \gamma_{k+1} d^k \tag{6.1}$$

that is as a linear combination of the gradient's direction in point x^{k+1} and of the previous direction according to which the unidimensional minimization which had generated point x^{k+1} was made.

Coefficient γ_{k+1} will be established so that d^{k+1} and d^k in the event of minimizing the quadratic (5.1) be mutually conjugate, supposing that the previous directions d^j, $j = 0,1...,k$ have already been established as mutually conjugate directions. We have:

$$d^{k+1} \cdot A \cdot d^k = (-g^{k+1} + \gamma_{k+1} d^k)\ A\ d^k = 0$$

wherefrom

$$\gamma^{k+1} = \frac{g^{k+1} A\ d^k}{d^k\ A\ d^k}.$$

from step 2b we have:

$$d_k = \frac{x^{k+1} - x^k}{\lambda_k}$$

wherefrom

$$g^{k+1} A\ d^k = \frac{1}{\lambda_k} g^{k+1} A\ (x^{k+1} - x^k) = \frac{1}{\lambda_k} g^{k+1}\ (g^{k+1} - g^k) = \frac{1}{\lambda_k} g^{k+1} \cdot g^{k+1}$$

And this because from theorem 2,

$$g^{k+1} \cdot d^s = 0,\ s = 0, 1, ..., k$$

thus, it results:

$$g^{k+1} \cdot d^s = g^{k+1}\ (-g^s + \gamma_s d^{s-1}) = g^{k+1}\ g^s = 0,\ s = 1, ..., k.$$

On the other hand

$$d^k A d^k = (-g^k + \gamma_k d^{k-1}) A d^k = -g^k A d^k = -g^k \frac{g^{k+1} - g^k}{\lambda_k} = \frac{{}^k g^k}{\lambda_k}$$

Thus:

$$\gamma_{k+1} = \frac{g^{k+1} g^{k+1}}{g^k g^k} \tag{6.2}$$

supposing that we have shown that d° and d_1 are conjugated, which results immediately if we consider for $d^\circ = -g^\circ$.

Indeed, we have

$$d^\circ = -g^\circ = -Ax^\circ - b$$

$$x^1 = x^\circ + \lambda_0 d^\circ, \text{ where } \lambda_0 \text{ satisfies } d^\circ (A(x^\circ + \lambda_0 d^\circ) + b) = 0,$$

in case of the quadratic form (5.1.), or is found as min $f(x^\circ + \lambda d^\circ)$ in case of a general function.
We show now that a γ_i can be established so that $d^1 = -g^1 + \gamma_1 d^\circ$, and d^1 and d° be conjugate, that is $d^\circ A d^1 = 0$.

We have:

$$d^\circ A d^1 = -d^\circ A (g^1 - \gamma_1 d^\circ) = -\frac{1}{\lambda_0} (x^1 - x^\circ) A (g^1 - \gamma_1 d^\circ) =$$

$$= -\frac{1}{\lambda_0} (g^1 - g^\circ) \cdot (g^1 - \gamma_1 d^\circ) = -\frac{1}{\lambda_0} (g^1 - g^\circ) (g^1 + \gamma_1 g^\circ) =$$

$$= -\frac{1}{\lambda_0} (g^1 g^1 - g^\circ g^\circ \gamma_1) = 0$$

It results thereof

$$\gamma_1 = \frac{g^1 g^1}{g^\circ g^\circ}$$

and in this way the induction is complete.

The application of this method for nonquadratic functions was suggested by Fletcher and Reeves in [15] underlining that better results are obtained if after $n + 1$ iteration we restart by

$$d^j = -g^j, \; j \equiv 0 \; (mod \; n).$$

The idea at the basis of restartings is that of eliminating errors, in establishing the directions generated by the fact that only in the proximity of the extremum point the function is well approximated by a quadratic form.

P. Wolfe in his lecture on nonlinear programming held at Columbia University in 1969—1970 proposed for experimenting a modified Fletcher —Reeves formula in which the directions, for maximum, are given by

$$d^k = g^k + \frac{g^k g^k - g^k g^{k-1}}{d^{k-1} g^{k-1}} \cdot d^{k-1} \tag{6.3}$$

and for which our tests have supplied results comparable with the Fletch-Reeves method or better.

We mention on this occasion that the numerical results previously published by us in [30] because of a programming error cannot be attributed to a formula (6.3) but are given by optimizations on directions calculated according to formula

$$d^k = g^k + \frac{\| g^k \| - g^k g^{k-1}}{d^{k-1} g^{k-1}} d^{k-1}$$

Finnally, we also mention the variant of conjugated gradients initially given by Hestenes and Stiefel [11] for minimizing the functions and applied by Fletcher and Reeves [15] to the general case:

$$d^\circ = \frac{g^\circ}{g^\circ g^\circ} \tag{6.4}$$

$$d^k = -\frac{g^k}{g^k g^k} + d^{k-1}$$

In [2] it is demonstrated on this variant that under the conditions of Theorem 1, the Fletcher-Reeves method is convergent or more precisely "weakly convergent".

The notion of "weakly convergent", in the meaning of Zoutendiyk, is linked with the replacing of condition (3) in theorem 1 by the weaker demand that $\Sigma_k \theta_k^2$ be divergent, in which case it can be demonstrated: If $\Sigma_k \theta_k^2$ is divergent, in this case there is at least an accumulation point x^* for which $\bigtriangledown f(x^*) = 0$.

7. For the "*Method of parallel tangents*" [16], we have a brief description in [10], after which we make the presentation.

Iteration O:

x° is chosen and a $d^\circ = \bigtriangledown f(x^\circ)$ and a x^2 is determined so that

$$f(x^2) = \min_\lambda \{f(x^\circ + \lambda d^\circ);\ \lambda \geqslant 0\} \tag{7.1}$$

Iteration k. Given the points x^{2k-2} and x^{2k}, points x^{2k+1} and x^{2k+2} are determined in the following way:

a) x^{2k+1} is established so that $f(x^{2k+1}) = \min_\lambda \{f(x^{2k} + \lambda d^{2k});\ \lambda \geqslant 0\}$

where $d^{2k} = \bigtriangledown f(x^{2k})$ (7.2)

b) x^{2k+2} is established so that: $f(x^{2k+2}) = \min_\lambda \{f(x^{2k+1} + d^{2k+1});\ \lambda \geqslant 0\}$

where $d^{2k+1} = x^{2k+1} - x^{2k-2}$ (7.3)

It is demonstrated in [16] that vectors $y^k = x^{2k} - x^{2k-2}$ for $k = 1,...,n$ form a system of n vectors mutually conjugate to matrix A of quadratic form (5.1) and in this case the method ends after n iterations that is after $2n - 1$ unidimensional optimizations.

All methods presented so far use only partial derivates of order I. The methods of order II use partial derivates of order II.

8. *The Newton-Raphson method* which determines

$$x^{k+1} = x^k + \lambda_k d^k$$

$$d^k = - H^{-1}(x^k) g^k \tag{8.1}$$

where $H(x^k)$ is the Hessian matrix in point x^k, has remarkable properties as regards the convergence in the proximity of the optimum point and moreover, being applied to a positive definite quadratic form is leading to finding out the optimum in a single iteration. The disadvantages of the method obviously consist in the necessity of calculating at each iteration H^{-1} and moreover, if x° is far from the optimum point, the method may be even not convergent.

9. *The Davidon-Fletcher-Powell Method*

In 1959, Davidon [17] had the idea of approximating $H^{-1}(x)$ with the aid of the difference of gradients between the successive points previously obtained.

The algorithm, initially called by Davidon a method with variable metrics, remains unknown because of its inaccessible form of presentation. In 1963 Fletcher and Powell [18] remake the presentation of the algorithm by making their own contribution and by exemplifying its efficaciousness. Today, the Davidon-Fletcher-Powell method is considered to be one of the best. Being described in nearly all articles presenting gradient methods, the reader may find, besides the original works, a good presentation in [7].

The algorithm has the following steps in iteration k:

(i) $d^k = - H_k^{-1} g^k$ is calculated

(ii) λ_k is calculated so that $min\ \{f(x^k + \lambda d^k);\ \lambda \geqslant 0\} = f(x^k + \lambda_k d^k)$

(iii) compute:

a) $x^{k+1} = x^k + \lambda_k d^k$

b) $p_k = x^{k+1} - x^k$

c) $\varphi_k = g^{k+1} - g^k$

(iv) determine:

$$H_{k+1}^{-1} = H_k^{-1} + \frac{p_k p_k^T}{p_k^T \varphi_k} - \frac{H_k^{-1} \varphi_k \varphi_k^T H_k^{-1}}{\varphi_k^T H_k^{-1} \varphi_k}$$

As a test for completing the method we have $\|g^{k+1}\| \leqslant \varepsilon$.

Usually, at the first iteration $H^\circ = I$. If an estimation of H^{-1} is known, it can be used on condition that it be positive definite. If H° is positive definite, it can be shown that all H^k remain positive definite. As was shown in [31], for maximum, the algorithm has the same steps with the difference that at step (ii) it is maximized on direction d^k and as a first approximation of the Hessian one starts from a symmetrical matrix but negative definite, which can be $-I$.

It is important to mention that in case of quadratic function $f(x) = \frac{1}{2} xAx + bx$, $H_m^{-1} = A^{-1}$, so that the algorithm is quadratic convergent. By using only derivates of order I, the D.F.P. method is a method of order I.

Though it can be shown that directions d^k are descending directions, in case H^k are positive definite, this does not guarantee the convergence and there exists no demonstration of convergency for the D.F.P. method used to optimize a general function. The numerical tests are extremely favourable except the cases when H^k becomes nearly singular. The results concerning the convergency speed can be found in [19]. In [31] we are studying the D.F.P. procedure for maximization.

10. *Other developments of the Newton type methods*

The idea of the modified Newton type methods consists in the fact that if we consider that the function is sufficiently well approximated by terms up to the order 2,

$$f_k = f_{k-1} + g^{k-1} p_k + \frac{1}{2} p_k H(x^{k-1} + \theta p_k) p_k, \quad 0 \leqslant \theta \leqslant 1 \qquad (10.1)$$

where $f_k = f(x^k)$, $g^k = g(x^k) = \nabla f(x^k)$, $p_k = x^k - x^{k-1}$

and H is the Hessian matrix, the iterative formula

$$x^k = x^{k-1} + \lambda_k d^k \qquad (10.2)$$

$$d^k = -H^{-1}(x^{k-1}) g^{k-1} \qquad (10.3)$$

is justified by the fact that supposing that in x^k the function would attain the optimum x^*, in this case the gradient in x^k would be zero and as from (10.1) we have, by partial derivation,

$$v_k = g^k - g^{k-1} = H p_k \qquad (10.4)$$

would give:

$$x^* = x^{k-1} - H^{-1} g^{k-1}$$

As $H_{-1}^{k-1} = H^{-1}(x^{k-1})$ is but an approximation of $H^{-1}(x^{k-1} + \theta p_k)$ and as x^k is but an approximation, initially very far off, of x^* it is natural that we establish d^k according to formula (10.3) and try to optimize (10.2) on the direction d^k, by establishing a λ_k so that $f(x^k) = \underset{\lambda}{opt}\, f(x^{k-1} + \lambda d^k)$.

However, we shall see that we can admit in certain conditions the variant of (10.2) as well with $\lambda = 1$. In the following we propose to examine other methods for the approximation of H^{-1} by taking into consideration the fact that for general functions and in the event of a big number of variables H^{-1} can be, if not impossible to calculate, at least a big time consumer.

11. *Rank one Algorithms*

If $f(x) = \frac{1}{2} xAx + bx$, for any displacement $p_k = x^k - x^{k-1}$ we have

$$Ap_k = g^k - g^{k-1} = \gamma_k \tag{11.1}$$

With the aid of formula (11.1), supposing that we do not know A, we might establish it following displacements in n independent linear directions p_k.

Indeed, ofter carrying out the n mouves and we have built matrices $P_n = (p_1, p_2,...,p_n)$ and $V = (\varphi_1, \varphi_2,...,\varphi_n)$, relation (11.1) is equivalent to:

$$AP_n = V_n \tag{11.2}$$

where of $A = V_n \; P_n^{-1}$ or, what in reality interests us for optimization, $A^{-1} = P_n \; V_n^{-1}$

In the case of nonquadratic functions, supposing that we have reached a region where the function is sufficiently well approximated by a quadratic, we try the construction of H or H^{-1} so that in the case of quadratic $H = H_n = A$, respectively $H^{-1} = H_n^{-1} = A^{-1}$; H should be defined by a recurrence relation, such as relation (iv) in the D.F.P. algorithm.

If two succesive approximations of the Hessian or of the inverse of Hessian, differ between them by a matrix of the rank one, it is said that we have a rank one method. For references, we quote [20], [21], [22] and as an example we give here the algorithm presented by Murtagh and Sargent in [21], under the name of Method 1 of approximating the Hessian.

An approximation $H^{-1} = P_k V_k^{-1}$ of the Hessian is tried at iteration k by setting out from two nonsingular matrices P_o, V_o, usually the unit matrix.

Obviously, matrices $P_n = (p_1, p_2,...,p_n)$ and $V_n = (\varphi_1, \varphi_2,...\varphi_n)$. can be constructed in n steps, by setting out from any P_o and V_o through the formulas:

$$P_k = P_{k-1} + (p_k - P_{k-1} \, e_k) \, e_k^T \tag{11.3}$$

$$V_k = V_{k-1} + (\varphi_k - V_{k-1} \cdot e_k) \, e_k^T \tag{11.4}$$

and for the inverse, by taking into account the formula which gives the inverse of a matrix knowing the inverse of the matrix which differes from it by a single column.

$$V_k^{-1} = V_{k-1}^{-1} + \frac{(e_k - V_{k-1}^{-1} \varphi_k^1) \, e_k^T \; V_{k-1}^{-1}}{e_k^T \, V_{k-1}^{-1} \, \varphi_k} \tag{11.5}$$

By replacing (11.3) and (11.4) in

$$H_k^{-1} = P_k \, V_k^{-1} \tag{11.6}$$

and noting with $y_k^T = \frac{e_k^T \; V_{k-1}^{-1}}{e_k^T \; V_{k-1} \; \varphi_k}$ the recurrence formula is obtained

$$H_k^{-1} = H_{k-1}^{-1} + (p_k - H_{k-1}^{-1} \varphi_k) \; y_k^T \tag{11.7}$$

Obviously, V_k^{-1} exists if, in (11.5), $e_k^T V^{k-1} v_k \neq 0$ and this relation is true if $v_k \neq 0$ and if component k of vector $V_{k-1}^{-1} v_k$ is different from zero.
If component k of vector $V_{k-1}^{-1} v_k$ is zero, in this case e_k can be replaced by e_l, where e_l is the component of $V_{k-1} v_k$ with the highest absolute value. This method has the disadvantage that it can lead to the replacement of a column which was replaced recently or to repeated replacements of the same column. A better alternative would be that of replacing column j which was replaced at the aerliest iteration and for which $e_j^T V_{k-1}^{-1} v_k$ is not smaller than a given value.

12. *The algorithm of Zoutendiyk's projected gradient* [1] is, at the same time, an algorithm with conjugate directions whose presentation we make according to [19].

At iteration k we have the following steps:

a. Start: $x_0^k = x_{n_{k-1}}^{k-1}$, $P_0^k = I$, $j = 0$

b. j: = j + 1

Calculate at step j

1. $d_j^k = -P_{j-1}^k g_{j-1}^k$ where $g_{j-1}^k = \nabla f(x_{j-1}^k)$

If $g_{j-1}^k = 0$, x_{j-1}^k is a stationary point, stop.

If $P_{j-1}^k g_{j-1}^k \neq 0$, $g_{j-1}^k \neq 0$ then $j - 1 = n_k$, $x_{n_k}^k = x_{j-1}^k$, and go to a.

2. Establish.

$$x_j^k = \{x \mid \min_{\lambda} f(x_{j-1}^k + \lambda d_j^k)$$

If $g_j^k = 0$ stop.

3. Establish the projection matrix:

$$P_j^k = P_{j-1}^k - \frac{P_{j-1}^k v_j^k (v_j^k)^T P_{j-1}^k}{(v_j^k)^T P_{j-1}^k v_j^k} \tag{12.1}$$

where $V_j^k = g_j^k - g_{j-1}^k$ and restart from *b*.

The projection P matrix has an equivalent definition

$$P_j^k = \{I - V_j^k [(V_j^k)^T V_j^k)]^{-1} (V_j^k)^T\} \tag{12.2}$$

where $v_j^k = (v_1^k, v_2^k, ..., v_j^k)$ so that, at step j of iteration k, the gradient vector g_{j-1}^k is projected into the subspace orthogonal to that generated by vectors $v_1^k, v_2^k, ..., v_{j-1}^k$
We have the following Theorem: the projection algorithm minimizes a positive definite quadratic form in n steps or fewer.

13. *A Newton type method without using the derivative* is described in [20]. At each iteration, in $n(n+1)$ steps, a gradient and an approximation of the Hessian are established after which a unidimensional optimum point is determined according to a Newton type formula.

The approximation formulas are inferred, as in most cases, from the study of the case in which $f(a)$ would be exactly a quadratic form. In this case, the matrix of the quadratic form can be constructed exactly. Indeed if

$$f(x) = \frac{1}{2} xHx + bx$$

then

$$f(x^{k+1}) = f(x^k) + (g^k)^T p_k + \frac{(p_k)^T Hp_k}{2} \tag{13.1}$$

If x^{k+1} was established, so that

$$f(x^{k+1}) = \min_{\lambda} f(x^{k+1} + \lambda d^k)$$

in this case

$$g^{k+1} d^k = 0 = g^{k+1} p_k$$

$$p_k = x^{k+1} - x^k$$

As we have also

$$f(x^k) = f(x^{k+1}) - g^{k+1} p_k + \frac{(p_k^T) Hp_k}{2}$$

it results

$$f(x^k) = f(x^{k+1}) + \frac{(p_k)^T Hp_k}{2} \tag{13.2}$$

From (13.1) and (13.2) result

$$(g^k)^T p_k = -(p_k)^T Hp_k \tag{13.3}$$

a value which introduced in (13.1) gives:

$$f(x^{k+1}) = f(x^k) - \frac{(p_k)^T Hp_k}{2} \tag{13.4}$$

We should note now that if the displacement directions would be just the directions of the coordinate axes,

$$d^k = e^k$$

e^k being unitary vectors with 1 on position k, in this case

$$x^{k+1} - x^k = p_k = \lambda_k e^k$$

and formula (13.4) is written

$$f(x^{k+1}) = f(x^k) - \frac{\lambda_k^2}{2} h_{kk} \tag{13.5}$$

where we note with h_{ij} a certain element of H, whereof

$$h_{kk} = \frac{-2(f^{k+1} - f^k)}{\lambda^2_k} \tag{13.6}$$

After making n iterations which established the elements on the diagonal of the matrix of the Hessian we should renumber for facillitating the writing $x^\circ := x^n$. However, we store the initial value x° as z° and x^n as z^1.

We want to establish now the elements h_{ij}. If we note with

$$d^k_{ikjk} = e^{ik} + e^{jk},\ i_k \neq j_k \tag{13.7}$$

we shall have to carry out $\frac{n(n-1)}{2}$ minimizations on all d_{ij} distinct directions

If

$$x^{k+1}_{ikjk} = x^k_{ikjm} + \lambda^k_{ikjk}\, d^k_{ikjk}$$

where x^{k+1}_{ikjk} was established so that

$$f(x^{k+1}_{ikjk}) = f^{k+1}_{ikjk} = min\, f(x^k_{icjm} + \lambda^k_{ikjk} \cdot d^k_{ikjk})$$

in this case

$$h_{ikjk} = \frac{2(f^{k+1}_{ikjk} + f^k_{ikjm}) - (\lambda^k_{ikjk})^2 (h_{ikik} + h_{jkjk})}{2(\lambda^k_{ikjk})^2} \tag{13.8}$$

which is checked immediately by analysing the expression of the product $-\frac{(p_k)^T H\, p_k}{2}$ from (13.4) in case that $p_k = \lambda^k_{ikjk}\, d^k_{ikjk}$) and d^k_{ikjk} is given by (13.7)

It can be shown [20] that H once obtained, in case of a quadratic function, the gradient in point z^1 has the components given by:

$$\frac{\partial f(z^1)}{\partial x_i} = \sum_{j=i+1}^{n} h_{ij}(z^1_j - z^0_j) \tag{13.9}$$

where we noted with z^k_j z^0_j the component of j order of vector z^1 respectively z^0.

The gradient in a point z^1 once known, the gradient in any other point u will be given by

$$\nabla f(u) = \nabla f(z^1) + H(u - z^1) \tag{13.10}$$

And thus in point $x^{n(n+1)/2}$ we have come to in the last iteration a direction

$$d^{n(n+1)/2} = -H^{-1} \nabla f(x^{u(n+1)/2})$$

can be established, after which, by a single unidimensional optimization, in the case of the quadratic function, the minimum is obtained.

NOTES, COMENTS, NUMERICAL RESULTS

14. *A generalized Fletcher-Reeves formula.*

The calculating effort for establishing the unidimensional optimum increases with the growing exactness with which it is determined. The study of an algorithm of conjugated directions in which the unidimensional optimum is established with approximation is thus justified.

Let us consider a problem of maximum and take

$$d^0 = g^0 = \nabla f(x^0) \tag{14.1}$$

$$x^{k+1} = x^k + \lambda_k d^k \tag{14.2}$$

where directions d^k are considered to be of the form

$$d^k = g^k + \sum_{i=0}^{k-1} \gamma_i^k d^i \tag{14.3}$$

coefficients γ_i^k being established from the conjugacy relation

$$d^k \; A \; d^i = 0, \quad i = 0,1,..., \; k-1 \tag{14.4}$$

and λ_k so that $f(x^{k+1}) > f(x^k)$.

Under these conditions for d^k the following expression has resulted by a simple calculus:

$$d^k = g^k - \sum_{i=0}^{k-1} \gamma_i^k d_i \tag{14.5}$$

where

$$\gamma_i^k = \frac{(g^k)^T \cdot (g^{k+1} - g^i)}{(d^i)^T \cdot (g^{i+1} - g^i)} \tag{14.6}$$

For minimum:

$$d^k = -g^k + \sum_{i=0}^{k-1} \gamma_i^k d^i \tag{14.7}$$

coefficients γ_i^k being calculated according to the same formula (14.6).

Obviously, both the Fletcher-Reeves formula (6.2) and the modified Fletcher-Reeves forms (6.3) are particular cases of the formula (14.7).

15. *The method of conjugate directions, matrix inversion and a mixed type optimization algorithm.* It is known that if A is a symmetric and definite matrix, and D a matrix whose column vectors are A-orthonormal, then $A^{-1} = \pm DD^T$, the sign being plus if A is positive definite and minus if A is negative definite. We are only mentioning here a method by which A^{-1} can be obtained in n iterations, in an optimization process with an algorithm of conjugate directions, of the quadratic form $f(x) = \frac{1}{2} x A^T A x$. If A is a certain, but nonsingular, in this case $A^{-1} = DD^T A^T$ where the column vectors of D form an $A^T A$ orthonormal set and they are obtained by minimizing the quadratic $f(x) = \frac{1}{2} xA^T Ax$. The procedure is directly linked with the Hestenes and Steifel method [11] of solving the linear equation systems by the method of conjugated gradients. Since the method does not require the explicite knowledge of A, but only of the values and gradient of the function it can be used, as we hawe shown in [32], in establishing

for a nonquadratic function, an approximation of the inverse of the Hessian. This enables the creation of new optimization algorithms of a mixed type by combining an algorithm of conjugate directions with a Newton type algorithm.

16. *The Gradient-Hessian algorithm* is a variant of Murtagh and Sargent's algorithm presented in paragraph 11.

The algorithm proposed by us differes from the Murtagh and Sargent algorithm by the fact that it uses for establishing the optimization direction the approximation of the Hessian inverse only from $n+1$ to $n+1$ iterations and also by formula used by us for approximating the inverse of the Hessian, that is a direct application of the Sherman and Morrison [28] formula.

$$(A + u\varrho^T)^{-1} = A^{-1} - \frac{(A^{-1}u)(\varrho^T A^{-1})}{1 + \varrho^T A^{-1} u} \tag{15.1}$$

where A is a square matrix with the known, inverse u and ϱ being column vectors.

The algorithmic variant tested under the name of Gradient-Hessian algorithm is then the following:

$$\text{Given : } x^0,\ P_0 = I,\ V_0 = I,\ k = 1$$

a) If $$k \not\equiv 0 \mod (n+1)$$

$$x^k = x^{k-1} + \lambda_k d^k$$

$$d^k = \nabla f(x^{k-1})$$

$$\lambda_k : \min_{\lambda} f(x^{k-1} + \lambda d^k) = f(x^{k-1} + \lambda_k d^k)$$

We calculate $\nabla f(x^k)$

If $$\| \triangle f(x^k) \| < \varepsilon \text{ stop.}$$

If not: $$P_k = P_{k-1} + (p_k - P_{k-1} e_k) e_k^T$$

$$V_k^{-1} = V_{k-1}^{-1} - \frac{[V_{k-1}^{-1}(\varrho_k - e_k)][e_k^T V_{k-1}^{-1}]}{1 + e_k^T V_{k-1}^{-1}(\varrho_k - e_k)}$$

where

$$p_k = x_k - x^{k-1}$$

$$\varrho_k = \nabla f(x^k) - \nabla f(x^{k-1})$$

b) If $k \equiv 0 \ (mod\ n+1)$

$$x_k = x^{k-1} + \lambda_k d^k$$

where:

$$d^k = -H_k^{-1} \nabla f(x^{k-1}) = -P_k V_k^{-1} \nabla f(x^{k-1})$$

$$\lambda_k : \min_{\lambda} f(x^{k-1} + \lambda d^k) = f(x^{k-1} + \lambda_k d^k)$$

If $\| \nabla f(x^k) < \| \varepsilon$, stop

$$k = 0,\ P_0 = V_0 = I$$

c) $k = k + 1$ repeat from a).

The scheme of the Gradient-Hessian algorithm was thought out for testing a whole class of algorithms characterized by optimizations on n independent linear directions followed by an optimization on a Newton type direction. As the Murtagh and Sargent's algorithm, the Gradient-Hessian is a rank one type algorithm.

16. The Newton-Fletcher-Reeves algorithm is one of the same class with the gradient-Hessian algorithm, with which it has exactly the same steps with the difference that at point a)

$$d^k = -g^k + \frac{\| g^k \|^2}{\| g^{k-1} \|^2} d^{k-1}$$

where $g^k = \nabla f(x^k)$.

17. *Numerical results*

For the Cauchy method (*C*), Subrelaxed gradient (*GS*), Diagonal-Gradient (*GD*), with the carrying out of the movement on the diagonal at every fifth iteration, Fletcher-Reeves (*FR*), Fletcher-Reeves modified (*FRM*), Parallel tangents (*PT*), Davidon-Fletcher-Powell (*DFP*) and the mixed Gradient-Hessian algorithm (*GH*) we have made studies of efficiency by programming and testing them on the same test problems on which the algorithms not using the gradient were also tested.

Part of the results obtained are given in the tables at the end of the article in two variants: one using the STEP subroutine which seeks the optimum on line by successive interpolations with a quadratic form, without using the derivates and with the STEPG subroutine which interpolates with a cubic and uses the gradient.

A description of linear search technique used in subroutine STEP is given in [29] and for subroutine STEPG in [35].

To have the possibility of a comparison with the algorithms not using the derivate, the number $ne = nf + n \cdot ng$ was calculated, where nf is the number of evaluations of the function, ng the number of evaluations of the gradient and n the number of variables.

Our results place the Davidon-Fletcher-Powell and Fletcher-Reeves algorithms, just as they are estimated, among the most efficient algorithms, but there are exceptions too. The modified Fletcher-Reeves seems to be, on an average, more efficient. The algorithm of Parallel tangents and the gradient Hessian algorithm give almost the same results in case the STEPG subroutine is used.

Comparing the results of the efficiency tests for the algorithms using the derivate to the results obtained with the algorithms not using the derivate [30], it does not result as we would have expected that the algorithms using the information given by the derivate are superior, and this seems to be true even for one dimensional optimization subroutine. It is possible that this be true if the dimension of the problem is growing. However, the STEPG

subroutine proves more accuracy. In the literature mentioned below, reference is also made to the behaviour of one or an other of the algorithms. One of the relatively recent references we want to mention is [26] in which reference can be found in connection with the behaviour of the first degree algorithms. Through the number of algorithms tested and of the problems, our results give a wider image of the behaviour of the nonlinear optimization algorithms.

The author avails of this opportunity to acknowledge his gratitude to his colleagues Sigrid Moga and Florica Luban of Economic Data Processing-Laboratory of the Academy of Economic Studies for their cooperation in programming and testing on the computer the algorithms.

REFERENCES

1. ZOUTENDIJK, G.: *Methods of Feasible Directions*, Elsevier, Amsterdam, 1960.
2. ZOUTENDIJK, G.: *Nonlinear Programming*, Computational methods in [3].
3. *Integer and Nonlinear Programming*, J. Abadie Editor, Amsterdam 1970
4. CAUCHY AUGUSTIN: *Méthods gènérales pour la résolution des systèmes d'équations simultanées*. C. R. de l'Academie des Sciences 25. Juillet-Décembre 1947
5. CURRY J. B.: *The Method of Steepest Descent for Nonlinear Minimization Problems*, Quarterly J-of Applied Mathematics, vol 2, no 3, October 1944.
6. FORSYTHE, G. E. — *On the Asymptotic Directions of the Dimensional Optimum Gradient Method*, Computer Science Department, Standford University. Technical Report, no. CS, 61, 1967.
7. KOVALIK, J., OSBORNE, M. R.: *Methods for Unconstrained Optimization Problems*, Elsevier, New York 1968.
8. STEIN, M. L.: *Gradient Methods in the Solution of Systems of Linear Equations*, Nat. Bureau Standards Report 52-7, 1951.
9. FORSYTHE A. I. and MOTZKIN, G. E.: *IBM Experiments with Accelerated Gradient Methods for Linear Equations*, Nat. Bureau Standards Report 1643 (1952).
10. WOLFE, P.: *Methods of Nonlinear Programming in* [14].
11. HESTENES, M. R. and STIEFEL, E.: *Methods of Conjugate Gradients for Solving Linear Systems*, J. Research National Bureau of Standards, 49, (1952).
12. FLETCHER, R.: *A Review of Methods for Unconstrained Optimization* in [13.]
13. FLETCHER, R.: *Optimization*, Academic Press, London 1969.
14. *Nonlinear Programming*, J. Abadie Editor, North-Holland Publishing Company, Amsterdam, 1967.
15. FLETCHER, R. and REEVES C. M.: *Function Minimization by Conjugate Gradients*, Computer J., 7(1964), 149.
16. SHAH, B. V., R. J. BUEHLER and O. KEMPTHORNE: *Some Algorithms for Minimizing a Function of Several Variables*, J. Soc. Indust. Appl. Math. 12, 74—92 (1964)
17. DAVIDON, W. C.: *Variable Metric Method for Minimization*, A.E.C. Research and Development Report ANL5990.
18. FLETCHER, R., POWELL M.J.D.: *A Rapidly Convergent Descent Method for Minimization*, Computer J. 6, 1963.
19. McCORMICK G. P., PEARSON J. D.: *Variable Metric Methods and Unconstrained Optimization*, in [13].
20. FIACCO A. V., MC CORMICK G. P.: *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Willy, N. Y. 1968.
21. MURTAGH B. A., SARGENT W. H.: *A Constrained Minimization Method with Quadratic Convergence*, in [13].
22. POWELL M. J. D.: *Rank One Methods for Unconstrained Optimization* in [13].
23. BROYDEN C. G.: *Quasi-Newton Methods and their Application to Function Minimization*, Math. Comp. 21 (1967).
24. WOLFE P.: *Convergence Theory in Nonlinear Programming*, in [3].

25. RIBIERE G.: *Sur la methode de Davidon-Fletcher-Powell pour la minimisation des-functions*, Management Science, vol. 16 (1970), no.9.
26. MURTAGH B. A., SARGENT R.V.H.: *Computational Experience with Quadratically Convergent Minimization Methods*, The Computer Journal, No 13 (1970) nr.2.
27. DUMITRU V., SERBAN R.: *The Convergence Speed and the Choose of the First or of the Best Unidimensional Optimum Point: An Experiment Using Rosenbrock Function.* Economic Computation and Economic Cybernetics Studies and Research (to be published).
28. SCHERMAN J., MORRISON J.: *Adjustment of an Inverse Matrix Corresponding to Changes in a Given Column or a Given Row of the Original Matrix*, Ann. Math. stat. vol. 21. pp. 124/1949.
29. AMUNATEGUI J., DUMITRU V.; *Computer Experiments with an Unconstrained Nonlinear Program.* Economic Computation and Economic Cybernetics Studies and Research, no. 3/1970.
30. DUMITRU V.: *Function Optimization without Evaluating Derivatives*, Economic Computation and Economic Cybernetics Studies and Research, Part I no. 2/1973, Part II no.1/1974
31. DUMITRU V.: *On the Davidon-Fletcher-Powell Algorithm for Maximization.* Economic Computation and Economic Cybernetics Studies and Research. (to be published)
32. DUMITRU V.: *Conjugate Directions Method, Matrix Inversion and a Mixed Type Algorithm for Optimization.* (to be published)
33. Mc CORMICK, G. P., RITTER K.: *Projection Method for Unconstrained Optimization*, vol 10, no.2, 1972.
34. ZANGWILL N. I.: *Nonlinear Programming: A Unified Approach*, Prentice-Hall, N.Y., 1969.
35. *IBM System/360 Scientific Subroutine Package*, IBM 1968.

METHODES DE GRADIENT DANS L'OPTIMISATION SANS RESTRICTIONS

Après avoir passé en revue les principales méthodes de gradient dans l'optimisation sans restrictions, l'auteur présente les résultats des tests d'efficience effectués sur un nombre de huit algorithmes qui utilisent le gradient, dans le processus d'optimisation. On étudie, en outre, de nouvelles variantes d'algorithmes mixtes d'optimisation.

МЕТОДЫ ГРАДИЕНТА ПРИ ОПТИМИЗАЦИИ БЕЗ ОГРАНИЧЕНИЙ

После обзора важнейших методов градиента при оптимизации без ограничений в настоящей статье приводятся результаты тестов эффективности, проведенных автором по восьми алгоритмам, применяющим градиент в процессе оптимизации. Рассматриваются также новые варианты смешанных алгоритмов оптимизации.

*) $min\ f(x_1, x_2) = 100\ (x_2 - x_1^2)^2 + (1 - x_1)^2$

$x^* = (1, 1);\ f(x^*) = 0$

$x^\circ = (-1.2; 1);\ f(x^\circ) = 24.2$

	CAUCHY	SUB — RELAXED GRADIENT	GRADIENT DIAGONAL	FLETCHER-REEVES	MODIFIED FLETCHER — REEVES	PARALLEL TANGENTS	DAVIDON — PLETCHER POWELL	GRADIENT— HESSIAN
STEP	$x^k = \begin{bmatrix}0.98949\\0.97905\end{bmatrix}$	$x^k = \begin{bmatrix}0.9925\\0.9851\end{bmatrix}$	$x^k = \begin{bmatrix}1.00028\\1.00056\end{bmatrix}$	$x^k = \begin{bmatrix}1.000000\\1.000000\end{bmatrix}$	$x^k = \begin{bmatrix}0.99993\\0.99986\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}0.99994\\0.99987\end{bmatrix}$
	$f_k = 0.00011$	$f_k = 0.000006$	$f_k = 0.00000$	$f_k = 0.9\times10^{-10}$	$f_k = 0.45\times10^{-8}$	$f_k = 0.3\times10^{-11}$	$f_k = 0.32\times10^{-10}$	$f_k = 0.4\times10^{-8}$
	$a_k = 0.009$	$a_k = 0.006$	$a_k = 0.00056$	$a_k = 0.00002$	$a_k = 0.00006$	$a_k = 0.3\times10^{-5}$	$a_k = 0.00001$	$a_k = 0.5\cdot10^{-4}$
	$k = 904$	$k = 528$	$k = 47$	$k = 60$	$k = 32$	$k = 50$	$k = 19$	$k = 31$
	$n_f = 6982$	$n_f = 4289$	$n_f = 340$	$n_f = 327$	$n_f = 197$	$n_f = 600$	$n_f = 107$	$n_f = 352$
	$n_g = 905$	$n_g = 529$	$n_g = 48$	$n_g = 61$	$n_g = 33$	$n_g = 150$	$n_g = 20$	$n_g = 32$
	$n_e = 7792$	$n_e = 5347$	$n_e = 436$	$n_e = 549$	$n_e = 263$	$n_e = 900$	$n_e = 147$	$n_e = 416$
STEPG	$x^k = \begin{bmatrix}0.99835\\0.99669\end{bmatrix}$	$x^k = \begin{bmatrix}0.99993\\0.99986\end{bmatrix}$	$x^k = \begin{bmatrix}0.99696\\0.99394\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\end{bmatrix}$		$x^k = \begin{bmatrix}1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00001\end{bmatrix}$
	$f_k = 0.27\times10^{-5}$	$f_k = 0.5\times10^{-8}$	$f_k = 0.9\times10^{-5}$	$f_k = 0.8\times10^{-11}$	$f_k = 0.22\times10^{-12}$		$f_k = 0.0$	$f_k = 0.15\times10^{-8}$
	$a_k = 0.001$	$a_k = 0.00006$	$a_k = 0.009$	$a_k = 0.5\times10^{-5}$	$a_k = 0.9\times10^{-6}$	—	$a_k = 0.0$	$a_k = 0.07\times10^{-3}$
	$k = 500$	$k = 421$	$k = 500$	$k = 34$	$k = 34$		$k = 20$	$k = 36$
	$n_f = 1044$	$n_f = 893$	$n_f = 961$	$n_f = 73$	$n_f = 83$		$n_f = 82$	$n_f = n_g = 90$
	$n_g = 1044$	$n_g = 893$	$n_g = 961$	$n_g = 74$	$n_g = 83$		$n_g = 82$	$n_e = 270$
	$n_e = 3132$	$n_e = 2679$	$n_e = 2883$	$n_e = 222$	$n_e = 249$		$n_e = 246$	

*) ROSENBROCK'S TEST PROBLEM

*) $\min f(x_1, x_2, x_3, x_4) = 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90\,(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1\,[(x_2 - x_1)^2 + (x_4 - x_1)^2] + 19.8\,(x_2 - x_1)\,(x_4 - x_1)$

$x^* = (1, 1, 1, 1)$; $f(x^*) = 0$

$x^\circ = (-3, -1, -3, -1)$; $f(x^\circ) = 19192$

	CAUCHY	SUB — RELAXED GRADIENT	GRADIENT DIAGONAL	FLETCHER — REEVES	MODIFIED FLETCHER — REEVES	PARALLEL TANGENTS	DAVIDON — FLETCHER — POWELL	GRADIENT — HESSIAN
STEP	$x^k = \begin{bmatrix}1.00063\\1.00126\\0.99943\\0.99886\end{bmatrix}$	$x^k = \begin{bmatrix}1.0112\\1.0233\\0.9884\\0.9765\end{bmatrix}$	$x^k = \begin{bmatrix}1.00322\\1.00648\\0.99671\\0.99343\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\\0.99997\\0.99994\end{bmatrix}$	$x^k = \begin{bmatrix}1.0001\\1.0002\\0.9999\\0.9998\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\\0.99999\\0.99999\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\\1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\\1.00000\\1.00000\end{bmatrix}$
	$f_k = 0.00000$	$f_k = 0.00049$	$f_k = 0.00004$	$f_k = 0.3\times10^{-8}$	$f_k = 0.3\times10^{-7}$	$f_k = 0.14\times10^{-9}$	$f_k = 0.2\times10^{-10}$	$f_k = 0.3\times10^{-11}$
	$a_k = 0.00405$	$a_k = 0.1$	$a_k = 0.0095$	$a_k = 0.00009$	$a_k = 0.0008$	$a_k = 0.00007$	$a_k = 0.00009$	$a_k = 0.4\times10^{-4}$
	$k = 899$	$k = 1000$	$k = 198$	$k = 141$	$k = 53$	$k = 149$	$k = 47$	$k = 62$
	$n_f = 7332$	$n_f = 7700$	$n_f = 1591$	$n_f = 687$	$n_f = 490$	$n_f = 1460$	$n_f = 315$	$n_f = 522$
	$n_g = 900$	$n_g = 1001$	$n_g = 199$	$n_g = 142$	$n_g = 54$	$n_g = 448$	$n_g = 48$	$n_g = 63$
	$n_e = 10932$	$n_e = 11688$	$n_e = 2387$	$n_e = 1255$	$n_e = 598$	$n_e = 3252$	$n_e = 507$	$n_e = 744$
STEPG	$x^k = \begin{bmatrix}1.0143\\1.0288\\0.9855\\0.9711\end{bmatrix}$	$x^k = \begin{bmatrix}1.0122\\1.0246\\0.9876\\0.9754\end{bmatrix}$	$x^k = \begin{bmatrix}1.0219\\1.0446\\0.9775\\0.9554\end{bmatrix}$	$x^k = \begin{bmatrix}0.99977\\0.99994\\1.00000\\1.00010\end{bmatrix}$	$x^k = \begin{bmatrix}0.99999\\0.99998\\1.00000\\1.00000\end{bmatrix}$		$x^k = \begin{bmatrix}1.00000\\1.00000\\1.00000\\1.00000\end{bmatrix}$	$x^k = \begin{bmatrix}1.00000\\1.00000\\1.00000\\1.00000\end{bmatrix}$
	$f^k = 0.0007$	$f^k = 0.0005$	$f^k = 0.0017$	$f_k = 0.3\times10^{-8}$	$f_k = 0.5\times10^{-9}$	—	$f_k = 0.0$	$f_k = 0.7\times10^{-10}$
	$a^k = 0.03$	$a^k = 0.02$	$a_k = 0.14$	$a_k = 0.00008$	$a_k = 0.00005$		$a_k = 0.0$	$a_k = 0.7\times10^{-4}$
	$k = 500$	$k = 500$	$k = 500$	$k = 154$	$k = 231$		$k = 38$	$k = 68$
	$n_f = 1007$	$n_f = 1007$	$n_f = 924$	$n_f = 402$	$n_f = 562$		$n_f = 199$	$n_f = ng = 148$
	$n_g = 1007$	$n_g = 1007$	$n_g = 924$	$n_g = 402$	$n_g = 562$		$n_g = 199$	$n_e = 740$
	$n_e = 5035$	$n_e = 5035$	$n_e = 4620$	$n_e = 2010$	$n_e = 2810$		$n_e = 995$	

*) WOOD'S TEST PROBLEM

*) $min\ f(x_1\, x_2) = \sum_{i-1}^{3} [c_i - x_1\,(1 - x_2^i)]^2$; $c_1 = 1.5,\ c_2 = 2.25,\ c_3 = 2.625$

$x^* = (3,\ 0.5)$; $f(x^*) = 0$

$x^\circ = (0,\ 0)$; $f(x^\circ) = 14.2$

	CAUCHY	SUB — RELAXED GRADIENT	GRADIENT — DIAGONAL	FLETCHER — REEVES	MODIFIED FLETCHER — REEVES	PARALLEL TANGENTS	DAVIDON FLETCHER — POWELL	GRADIENT HESSIAN
STEP	$x^k = \begin{bmatrix}2.99970\\0.49993\end{bmatrix}$	$x^k = \begin{bmatrix}2.99980\\0.49994\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.49999\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$	$x^k = \begin{bmatrix}3.00010\\0.50002\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.49999\end{bmatrix}$
	$f^k = 0.1\times10^{-7}$	$f_k = 0.9\times10^{-8}$	$f_k = 0.2\times10^{-9}$	$f_k = 0.5\times10^{-10}$	$f_k = 0.6\times10^{-9}$	$f_k = 0.0$	$f_k = 0.0$	$f_k = 0.14\times10^{-9}$
	$a^k = 0.00008$	$a_k = 0.00009$	$a_k = 0.00001$	$a_k = 0.00003$	$a_k = 0.00002$	$a_k = 0.0$	$a_k = 0.0$	$a = 0.7\times10^{-4}$
	$k = 155$	$k = 242$	$k = 17$	$k = 16$	$k = 10$	$k = 34$	$k = 8$	$k = 12$
	$n_f = 904$	$n_f = 1269$	$n_f = 115$	$n_f = 99$	$n_f = 63$	$n_f = 527$	$n_f = 43$	$n_f = 129$
	$n_g = 156$	$n_g = 243$	$n_g = 18$	$n_g = 17$	$n_g = 11$	$n_g = 102$	$n_f = 9$	$n_g = 13$
	$n_e = 1216$	$n_e = 1755$	$n_e = 151$	$n_e = 133$	$n_c = 85$	$n_e = 731$	$n_e = 61$	$n_e = 155$
STPEG	$x^k = \begin{bmatrix}2.99980\\0.49995\end{bmatrix}$	$x^k = \begin{bmatrix}2.99980\\0.49995\end{bmatrix}$	$x^k = \begin{bmatrix}2.99980\\0.49995\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$	$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$		$x^k = \begin{bmatrix}3.00000\\0.50000\end{bmatrix}$	$x^k = \begin{bmatrix}3.00010\\0.50001\end{bmatrix}$
	$f_k = 0.5\times10^{-8}$	$f_k = 0.5\times10^{-8}$	$f_k = 0.5\times10^{-8}$	$f_k = 0.1\times10^{-10}$	$f_k = 0.1\times10^{-11}$		$f_k = 0.9\times10^{-11}$	$f_k = 0.4\times10^{-9}$
	$a_k = 0.00005$	$a_k = 0.00005$	$a_k = 0.00005$	$a_k = 0.00003$	$a_k = 0.00001$	—	$a_k = 0.00001$	$a_k = 0.4\times10^{-4}$
	$k = 99$	$k = 99$	$k = 118$	$k = 11$	$k = 9$		$k = 8$	$k = 12$
	$n_f = 201$	$n_f = 201$	$n_f = 220$	$n_f = 27$	$n_f = 25$		$n_f = 25$	$n_f = \mathrm{ng} = 30$
	$n_g = 201$	$n_g = 201$	$n_g = 220$	$n_g = 27$	$n_g = 25$		$n_g = 25$	$n_e = 90$
	$n_e = 603$	$n_e = 603$	$n_e = 660$	$n_e = 81$	$n_e = 75$		$n_e = 75$	

*) $min\ f(x_1, x_2, x_3) = \sum_{\mu} \{[e^{-\mu x_1} - e^{-\mu x_2}] - x_3 [e^{-\mu} - e^{-10\mu}]\}^2$; $\mu = \{0.1, 0.2, ..., 1\}$

$x^* = (1.\ 10.\ 1)$; $f(x^*) = 0$ sau $\forall x(x_1 = x_2\ ,\ x_3 = 0)$

$x^\circ = (0, 20, 20)$; $f(x^\circ) = 1021.7$

	CAUCHY	SUB — RELAXED GRADIENT	GRADIENT— DIAGONAL	FLETCHER — REEVES	MODIFIED FLETCHER — REEVES	PARALLEL TANGENTS	DAVIDON FLETCHER — POWELL	GRADIENT— HESSIAN
STEP	$x^k = \begin{bmatrix} 0.87918 \\ 12.1830 \\ 1.09200 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.80041 \\ 13.2430 \\ 1.1269 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.9975 \\ 10.0310 \\ 1.0017 \end{bmatrix}$	$x^k = \begin{bmatrix} 1.0000 \\ 9.9999 \\ 0.9999 \end{bmatrix}$	$x^k = \begin{bmatrix} 6.1391 \\ 6.1330 \\ 0.0003 \end{bmatrix}$	$x^k = \begin{bmatrix} 10.759 \\ 10.783 \\ 0.0003 \end{bmatrix}$	$x^k = \begin{bmatrix} 2.5928 \\ 2.5928 \\ 0.000004 \end{bmatrix}$	$x^k = \begin{bmatrix} 1.0019 \\ 9.9730 \\ 0.9985 \end{bmatrix}$
	$f_k = 0.002$	$f_k = 0.0044$	$f_k = 0.4 \times 10^{-6}$	$f_k = 0.1 \times 10^{-10}$	$f_k = 0.1 \times 10^{-6}$	$f_k = 0.6 \times 10^{-6}$	$f_k = 0.1 \times 10^{-11}$	$f_k = 0.3 \times 10^{-6}$
	$a_k = 0.003$	$a_k = 0.019$	$a_k = 0.00009$	$a_k = 0.8 \times 10^{-5}$	$a_k = 0.00005$	$a_k = 0.00007$	$a_k = 0.4 \times 10^{-6}$	$a_k = 0.2 \times 10^{-4}$
	$k = 500$	$k = 500$	$k = 165$	$k = 34$	$k = 9$	$k = 41$	$k = 13$	$k = 53$
	$n_f = 2746$	$n_f = 3366$	$n_f = 965$	$n_f = 261$	$n_f = 80$	$n_f = 491$	$n_f = 120$	$n_f = 601$
	$n_g = 501$	$n_g = 501$	$n_g = 166$	$n_g = 35$	$n_g = 10$	$n_g = 123$	$n_f = 14$	$n_g = 54$
	$n_e = 4249$	$n_e = 4869$	$n = 1463$	$n_e = 366$	$n_e = 110$	$n_e = 860$	$n_e = 162$	$n_e = 763$
STEPG	$x^k = \begin{bmatrix} 0.70801 \\ 19.65210 \\ 1.23431 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.7080 \\ 19.6521 \\ 1.2343 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.7076 \\ 19.7250 \\ 1.2358 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.9990 \\ 10.0090 \\ 1.0006 \end{bmatrix}$	$x^k = \begin{bmatrix} 0.9988 \\ 10.0160 \\ 1.0009 \end{bmatrix}$		$x^k = \begin{bmatrix} 1.0000 \\ 9.9997 \\ 1.0000 \end{bmatrix}$	$x^k = \begin{bmatrix} 19.39600 \\ 19.39600 \\ 0.00000 \end{bmatrix}$
	$f_k = 0.0258$	$f_k = 0.0258$	$f_k = 0.026$	$f_k = 0.54 \times 10^{-7}$	$f_k = 0.12 \times 10^{-6}$		$f_k = 0.12 \times 10^{-9}$	$f_k = 0.7 \times 10^{-10}$
	$a_k = 0.005$	$a^k = 0.005$	$a_k = 0.004$	$a_k = 0.00009$	$a_k = 0.00003$	—	$a_k = 0.7 \times 10^{-5}$	$a_k = 0.2 \times 10^{-4}$
	$k = 500$	$k = 500$	$k = 500$	$k = 81$	$k = 19$		$k = 19$	$k = 8$
	$n_f = 1008$	$n_f = 1008$	$n_f = 925$	$n_f = 177$	$n_f = 54$		$n_f = 11$	$n_f = \text{ng} = 41$
	$n_g = 1008$	$n_g = 1008$	$n_g = 925$	$n_g = 177$	$n_g = 54$		$n_g = 111$	$n_e = 164$
	$n_e = 4032$	$n_e = 4032$	$n_f = 3700$	$n_e = 708$	$n_e = 216$		$n_e = 444$	

* BOX'S TEST PROBLEM