V. DUMITRU
Department of Economic Cybernetics
Academy of Economic Studies

Conjugate Direction Method, Hessian Approximation and a New Class of Mixed Type Algorithms for Optimization*

Hestenes and Stiefel gave in [1] an algorithm for solving a linear system of equations by using the conjugate gradient method. It is natural to think that such a method might also be used in inverting matrices.

In the following, we shall point out that the inverse of a definite matrix can be obtained in certain conditions in an optimization process of the associate quadratic form, by using only the values of the function and the gradient of quadratic form, a fact which allows to determine an approximation of the Hessian inverse for nonquadratic functions. Hence, a new class of mixed type algorithms for optimization results by combining a conjugate direction algorithm with a Newton-Raphon type algorithm.

Let us recall some definitions and theorems. We shall note a real matrix by A. The Roman letters with a single index note the vectors.

If

$$d_i^T A d_j = 0, \quad i \neq j \tag{1}$$

we say that vectors d_i and d_j are mutually conjugate to matrix A or that they are A-orthogonal.

If

$$d^T A d = 1 \tag{2}$$

we say that d is A-normalized.

* A paper presented to the editorial board in July 1964 and reported in brief to "The Fifth International Conference of the Theory of Probabilities and Applications", Braşov, Sept., 1964.

If the vectors d_i, $i = 1, \ldots, s$ are mutually conjugate and A-normalized we say that they form a set of vectors A-orthonormal.

We say that a real matrix A is positive definite if for any $x \neq 0$:

$$x^T A x > 0. \tag{3}$$

It is semi-positive definite or non-negative definite if (3) is also satisfied with equal sign. If inequality (3) is satisfied with an opposite sign A is negative definite.

We note that the notion of a definite matrix has a sense also for non-symmetrical matrices. Moreover, the same test criterion can be applied for establishing the character. Indeed, let's consider A a non-symmetric matrix and $A_1 = \left(a^1_{ij} = \frac{a_{ij} + a_{ji}}{2}\right)$ the associate symmetric matrix. Obviously $A_1 = \frac{1}{2}(A + A^T)$ and hence

$$xAx = xA_1x.$$

The following proposition follows:
The non-symmetric matrix A is positive definite if the associate symmetric matrix A_1 is positive definite. To establish the character of matrix A we can thus use Sylvester's criterion, for example, applied to matrix A_1.
The following propositions are immediate:
Theorem 1. If matrix A is nonsingular and D is a matrix whose column vectors form a set of n A-orthonormal lineary independent vectors, then

$$A^{-1} = DSD^T \tag{4}$$

where S is a diagonal matrix, with $s_{ii} = \pm 1$, sign being the sign of d_iAd_i, where, d_i note the column vectors of D. Proof As the vectors of A are mutually conjugate, $D^TAD = S$, where $s_{ij} = 0$ for $i \neq j$. Thus, S is a matrix with null elements outside of the diagonal. Let us show that $s_{ii} \neq 0$, $\forall i$. Since we supposed matrix A nonsingular and d_i lineary independent vectors, it results that rang $S = n$ and thus s_{ii} cannot be null. From $D^TAD = S$, in the conditions of the theorem, formula 4 is immediate.

Observation 1. The theorem does not imply that there are no vectors x for which $xAx = 0$.

Observation 2. The theorem is a generalization of the well known theorem: If A has n lineary independent eigenvectors and matrix P, having this eigenvectors as column vectors, is orthogonal, then $A = P\Lambda P^T$, where Λ is the diagonal matrix formed with the eigenvalues of matrix A.

Theorem 2. A set of vectors mutually conjugate with a positive (negative) definite matrix forms a set of vectors lineary independent.
Proof. Let's consider d_i, $i = 1, \ldots, k, k \leqslant r$ a set of k vectors mutually conjugate with positive definite matrix A. Let us suppose that the vectors are not lineary independent. Then,

$$\sum_{i=1}^{k} \alpha_i d_i = 0, \; d_i \text{ not all of them zero.}$$

If $\alpha_k \neq 0$ then $d_k = \sum_{i=1}^{i \neq k} \beta_i d_i$ and thereof, since the vectors are mutually conjugate, $d_k^T A d_j = 0 = \beta_j d_j^T A d_j$, $\forall j < k$ which contradicts the hypothesis that A is positive definite.

Corollary 1. It results from theorems 1 and 2 that if D is a matrix the column vectors of which form a set of n vectors A-orthonormal, A symmetric and positive (negative) definite, then

$$A^{-1} = \pm DD^T \tag{5}$$

the sign being plus if A is positive definite and minus if A is negative definite.

Has a nonsymmetric matrix A n conjugate directions? It is known that if A is a normal matrix, that is, if it satisfies $A^TA = AA^T$, then it has its n eigenvectors lineary independent. A matrix having less than n lineary independent eigenvectors is called defective. However if a matrix is not normal or if its eigenvalues are repeated, it does not result that the matrix is defective, in all cases, if the matrix is not symetric we cannot proof that the eigenvectors are orthogonal. [5]

Corollary 2. Formula (5) shows that if A is a definite regular matrix, but is nonsymmetric, then we cannot find n directions mutually conjugate with A.

Theorem 3. If A is a nonsingular matrix, A^TA is symmetric and positive definite.

Proof. Obviously A^TA is symmetric. It is also positive definite. Indeed

$$xA^TAx = z^Tz > 0 \text{ where } z = Ax.$$

Thus A^TA is at least semipositive definite.

If $z^Tz = 0$ it results $z = 0$ that is $Ax = 0$. As A is nonsingular, it results that $x = 0$, and, thus $xA^TAx > 0$ for any $x \neq 0$.

Theorem 4

If A is a nonsingular matrix and D a matrix whose column vectors form a set of n vectors — T orthonormal to matrix $T = A^TA$, then

$$A^{-1} = DD^TA^T. \tag{6}$$

Proof. From theorem 3, A^TA is positive definite. On the other hand, in the stated hypotheses we have:

$$D^TA^TAD = I.$$

As, from theorem 2, D and thus also D^T admit an inverse then $A^TA = (D^T)^{-1}D^{-1} = (DD^T)^{-1}$ As if A is nonsingular A^TA is nonsingular, we have $(A^TA)^{-1} = DD^T$. Hence $A^{-1} = DD^TA^T$.

INVERTING MATRICES WITH THE AID OF CONJUGATE DIRECTIONS

It is known that in unconstrained optimization, some of the most efficient algorithms are the conjugate direction algorithms, for example, the Fletcher-Reeves algorithm [3]. They have the property of being quadraticaly

convergent, that is of optimizing a quadratic form $f(x) = \frac{1}{2} x^T Ax + x^T b$ in a finite number of steps. It results that if, in such a process, had we found a convenient formula for A-normalization of mutually conjugate directions generated in the optimizing process, we could obtain as a secondary product A^{-1} and this, the more so, as it is known that the method can be used in solving the linear systems [1], [2 .

To explain the ideas, let us suppose that A is symmetric and positive definite and that we solve the following optimization, problem,

$$min\, f(x) = \frac{1}{2} xAx + x^T b \tag{7}$$

with an iterative algorithm

$$x_{k+1} = x_k + \alpha_k^* d_k \tag{8}$$

where

$$d_i^T A d_j = 0,\ i \neq j \tag{9}$$

$$f(x_k + \alpha_k^* d_k) = \min_{\alpha_k > 0} f(x_k + \alpha_k d_k). \tag{10}$$

In order to obtain matrix D, the only thing to do is that at each iteration we should A-normalize vectors d_k. It results from $d_k A d_k = \gamma_k$ that the normalizing factor is $\beta_k = \frac{1}{\sqrt{|\gamma_k|}}$. We have considered $|\gamma_k|$, for writing formulas valid also in case that A was not positive definite. The direct determination from the above formula of the normalizing factor requires the explicite knoledge of matrix A and thus it cannot be extended to finding a formula for approximating the inverse of the Hessian in case that $f(x)$ is no more a quadratic form.

If we note with $g_k = \nabla f(x_k)$, resulting from (7) is

$$g_{k+1} - g_k = A(x_{k+1} - x_k) \tag{11}$$

and from (8) and (11), we have:

$$g_{k+1} - g_k = \alpha_k^* A d_k. \tag{12}$$

Multiplying (12) to the left by d_k^T and considering that from (10) we have

$$d_k^T g_{k+1} = 0 \tag{13}$$

results

$$\gamma_k = -\frac{d_k^T g_k}{\alpha_k^*} \tag{14}$$

We should note that if A is positive definite γ_k given by (14) is positive and if is negative definite, it is negative, so that the normalizing factor is

$$\beta_k = 1/\sqrt{|\gamma_k|}. \tag{15}$$

It is obvious that if A is symmetric and negative definite we seek

$$max\ f(x) = \frac{1}{2}\, xAx + xb \tag{16}$$

or $min\ \{-f(x)\}$ and obtain $A^{-1} = -DD^T$ so that

$$A^{-1} = \pm DD^T. \tag{17}$$

If A is symmetric but not definite and in the optimizing process we come to create n mutually conjugate indepenedent directions, in this case

$$A^{-1} = DSD^T \tag{18}$$

where S is diagonal with $s_{ii} = \dfrac{\gamma_i}{|\gamma_i|}$.

If, we are interested only to obtain A^{-1}, we achieve the generation of conjugate directions by optimizing $f(x) = \dfrac{1}{2}\, xAx$.

In all cases, the method is dependent on the initial point, that is it must be so that the iterative process should converage to a stationary point in n iterations,. Consequently, in this, just as in the general case, when we do not know, as it happens usually, the character of A, for inverting A, we seek

$$min\ f(x) = \frac{1}{2}\, xA^TAx \tag{19}$$

We note that A^TA is symmetric and semipositive definite.

The column vectors of D are this time mutually conjugate with A^TA

$$d_iA^TAd_j = 0 \tag{20}$$

the normalizing factor, however, is calculated with the same formulas (14) and (15) and we have $D^TA^TAD = I$, from where

$$A^{-1} = DD^TA^T, \tag{21}$$

The method remains valid even if A is singular or of a $(m \times n)$ type, in which case by minimizing (19), we obtain

$$A^+ = DD^TA^T \tag{22}$$

the generalized inverse of A. The calculus of the generalized inverse is a current problem in practical applications, being equivalent to the problem of minimizing deviations through the method of least squares [5].

We note that, in case that A is symmetric, formula (17) remains valid even if $b = 0$ so that, if we are interested only in determining A^{-1}, in this case we generate the directions in a process of optimizing $f(x) = \frac{1}{2} xAx$.

A CLASS OF MIXED TYPE ALGORITHMS

The main purpose of our studies are the optimization algorithms. An important part among them was lately held by the mixed type algorithms [7], [9].

As in the optimizing methods on mutually conjugate directions, the elements of formula (14) are usually determined — in the methods using the gradient the product $d_k^T g_k$ is even calculated — it results that in a process of optimizing a quadratic form with a mutually conjugate directions method, the inverse matrix of the quadratic form can be obtained as a secondary product. Moreover, since (14) does not explicitely contain A it results that for a nonquadratic function $DD^T = H^{-1}$ is an approximation of the Hessian $H = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)$. Of course, the approximation of H^{-1} is better as $f(x)$ can be better approximated by a quadratic form.

This approximation can be used in various moments of the optimizing process in order to determine a Newton type optimization direction:

$$d_k = - H^{-1} \nabla f(x_k). \tag{23}$$

And, as for quadratic forms the optimization on this direction leads us to an optimum in a single iteration, in comparison with a process of generation of conjugate directions which if optimizing we come to the optimum, in general in n iterations, we hope this direction to be better also for nonquadratic forms.

By combining in various ways the algorithms of conjugate directions with movements on directions given by (23) we can create a whole class of mixed type algorithms, among which we describe below what we shall call *the Fletcher-Reeves-Newton Algorithm.*

Variant 1.

Considering x_k:

We calculate

$$x_{k+1} = x_k + \alpha_k^* d_k$$

where

a)

$$d_k = \begin{cases} - g_k & \text{for } k = 0 \text{ and } k \equiv 0(n+1) \\ - H_k^{-1} g_k & \text{for } k \equiv 0(n) \\ - g_n + \dfrac{g_k g_k}{g_{k-1} g_{k-1}} d_{k-1} & \text{for the other values of } k. \end{cases}$$

By H_k^{-1} we noted the approximation of the Hessian inverse obtained after n optimizations on mutually conjugate directions.
b) The value α_k^* of the step on direction can be obtained with the aid of a unidimensional optimization subroutine, so that

$$\min_{\alpha > 0} f(x_k + \alpha_k d_k) = f(x_k + \alpha_k^* d_k)$$

Obviously, directions d_k can be determined in the optimizations on Fletcher-Reeves directions and with the modified formula:

$$d_k = - g_k + \frac{g_k(g_k - g_{k-1})}{d_{k-1} g_{k-1}} d_{k-1}$$

Variant 2. The same method as for variant 1 with the difference that the movements on a Newton type directions are carried out only if the current point is sufficiently close to the optimum point and this in order to have a sufficiently good approximation of the inverse of the Hessian matrix. This can be achieved on condition that the movement on the Newton type direction be made only if $\| g_k \| < \mu$.

Variant 3. is the same as variant 1, with the difference that the approximation of the inverse of the Hessian matrix once obtained, a given number of optimizations are made on directions of a Newton type, determined with the aid of this inverse.

Numerical results. The method was tested on test problems which have already become classical as well as on several quadratic problems. The tests were carried out in a double precision on an IBM 370/135 computer.

For optimization on the directions we used either the STEP subroutine [8], which determine the optimum on direction by successive interpolations with a second degree polynom and does not use the gradient, or the STEPG subroutine which determines the direction by successive interpolations with a second degree polynom and uses the gradient. The STEPG subroutine is equivalent to the program section, corresponding to the direction optimization, of the Fletcher-Reeves subroutine in [10].

The results of the tests on various optimization variants described above are given in the tables and graphs at the end of the study, where Nf shows the number of function estimations, $Ne = Nf + n$. Ng, Ng being the number of valuations of the gradient, until iteration k ,and $d = \| x_k - x^* \|$. Analysing these results, it seems that we might draw the conclusion that Variant 1 is less efficient than the Fletcher-Reeves method and Variant 2 is more efficient and this because, at the beginning of the optimization process, when we are far from the optimum, the Newton type direction $d_{n+1} = - H_{n+1}^{-1} g_{n+1}$ is less good than a $n + 1$ Fletcher-Reeves direction. This may happen either because the Newton direction $d_k = - H^{-1} g_k$, determined with the inverse matrix $H^{-1}(x^k)$, calculated with exactitude, is less good than the Fletcher-Reeves direction, or because the approximation H_k^{-1} of $H^{-1}(x^k)$, determined with the method proposed by us,

is grosser if the current point is far from the optimum. It is known that the Newton-Raphson method may have difficulties if the initial point is far from the optimum. The results of some tests are given in [6], tests in which theNewton Raphson method, called here generalized Newton, $x^{k+1}=x^k-\alpha^* H^{-1}(x^k)$, is superior to the Davidon-Fletcher-Powell method. We do not have yet numerical results with which we decide which was the situation in case of the tests carried out by us. But, we can say that in the neighbourhood of the opimum point, as shown by the results obtained in the tables, the approximation of the Hessian inverse according to the method given by us, is good and comparable to that given by the Davidon-Fletcher-Powell method [4]. By $H^{-1}(x^*)$ we noted the inverse matrix calculated with exactitude in the theoretical optimum point x^*.

Variant [3] is less efficient than both variants and all the three, for the tests carried out by us, less efficient than the Davidon-Fletcher-Powell method. We mention that if results of a very great precision are desired, the calculus in double precision is essential. In a simple precision, below a certain value of the gradient norm, the process may no longer be convergent. This explains also the great differences between the results presented here and the results previously presented by us in [8] or [9]. Comparing these results to those in [9], we realize that such an oscillating process took place, in simple precision, for the results presented as having been obtained with the STEPG subroutine in [9]. It is now clear to us that in order not to have such a surprize, the valuation of efficiency should be made at various precision levels along the optimization process.

Though the results of the tests carried out show that a mixed type Fletcher-Reeves-Newton algorithm is not more efficient than the Davidon-Fletcher-Powell algorithm, we think that the idea of approximating the inverse of the Hessian matrix through the above method, besides, the interest in itself, has a practical importance as well. Indeed, the knowledge at the end of the optimization process of $H^{-1}(x^*)$ is extremely useful. For example, with its aid, we can, obviously, postoptimize immediately a quadratic unconstrained problem when the vector of coefficients of the first degree terms varies. Of course, we have this possibility also with the Davidon-Fletcher-Powell algorithm, but, if the problem is of big dimensions and the matrix does not find place in the internal storage, it is more advantageous to work with a vectorial algorithm of conjugate directions and to form the approximation of the Hessian inverse only in the last part of the optimizing process on an external storage socket, from where it can be used for determining the optimization direction for the last iteration.

The author wishes to express his warmest thanks to his colleague Florica Luban of the Economic Informatics Laboratory of the Academy of Economic Studies who, together with the author, had programmed and tested the method on the computer.

T.1.Rosenbrock's problem

$$\min f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$
$$x^\circ = (-1.2,1);\ f(x^\circ) = 24.2$$
$$x^* = (1,1);\ f(x^*) = 0.\ \|x^\circ - x^*\| = 2.2$$

$\|x^k - x^*\| \leq$	CAUCHY (STEPG, DP)	SUB-RELAXED GRADIENT (STEPG, DP)	DIAGONAL GRADIENT (p = 3, STEPG, DP)	DIAGONAL GRADIENT (p = 5, STEPG D.P)	DIAGONAL GRADIENT (p=7, STEPG DP)	DIAGONAL GRADIENT (p=3, STEP, D.P)
1	$d = 0.9954$ $f = 0.32$ $k = 46$, nf=109 $ne = 326$	$d = 0.8165$ $f = 0.26$ $k = 19$, $nf = 62$ $ne = 186$	$d = 0.9238$ $f = 0.29$ $k = 12$, $nf = 28$ $ne = 84$	$d = 0.9939$ $f = 0.33$ $k = 20$, $nf = 45$ $ne = 135$	$d = 0.8263$ $f = 0.36$ $k = 22$, $nf = 52$ $ne = 156$	$d = 0.8260$ $f = 0.23$ $k = 19$, $nf = 138$ $ne = 178$
10^{-1}	$d = 0.0882$ $f = 0.0018$ $k = 121$, $nf = 284$ $ne = 852$	$d = 0.0505$ $f = 0.0023$ $k = 35$, $nf = 111$ $n = 333$	$d = 0.07600$ $f = 0.16 \times 10^{-2}$ $k = 18$, $nf = 41$ $n = 123$	$d = 0.0032$ $f = 0.17 \times 10^{-3}$ $k = 31$, $nf = 68$ $ne = 204$	$d = 0.0845$ $f = 0.002$ $k = 33$, $nf = 74$ $ne = 222$	$d = 0.0083$ $f = 0.44 \times 10^{-4}$ $k = 31$, $nf = 228$ $ne = 296$
10^{-2}	$d = 0.0099$ $f = 0.22 \times 10^{-4}$ $k = 157$, $nf = 356$ $ne = 1068$	$d = 0.0099$ $f = 0.6 \times 10^{-4}$ $k = 83$, $nf = 255$ $ne = 765$	$d = 0.0013$ $f = 0.28 \times 10^{-5}$ $k = 19$, $nf = 44$ $ne = 132$	$d = 0.448 \times 10^{-4}$ $f = 0.25 \times 10^{-7}$ $k = 33$, $nf = 72$ $ne = 216$	$d = 0.0014$ $f = 0.13 \times 10^{-4}$ $k = 36$, $nf = 81$ $ne = 243$	$d = 0.96 \times 10^{-5}$ $f = 0.48 \times 10^{-8}$ $k = 38$, $nf = 265$ $ne = 343$
10^{-3}	$d = 0.00095$ $f = 0.2 \times 10^{-6}$ $k = 203$, $nf = 448$ $ne = 1344$	$d = 0.00019$ $f = 0.48 \times 10^{-7}$ $k = 171$, $nf = 519$ $ne = 1557$	$d = 0.00018$ $f = 0.47 \times 10^{-7}$ $k = 21$, $nf = 48$ $n = 144$	$d = 0.443 \times 10^{-4}$ $f = 0.39 \times 10^{-9}$ $k = 34$, $nf = 74$ $ne = 222$	$d = 0.46 \times 10^{-4}$ $f = 0.12 \times 10^{-7}$ $k = 38$, $nf = 85$ $ne = 255$	$d = 0.91 \times 10^{-5}$ $f = 0.16 \times 10^{-10}$ $k = 40$, $nf = 274$ $ne = 356$
10^{-4}	$d = 0.96 \cdot 10^{-4}$ $f = 0.2 \times 10^{-8}$ $k = 249$, $nf = 540$ $ne = 1620$	$d = 0.43 \times 10^{-4}$ $f = 0.23 \times 10^{-8}$ $k = 202$, $nf = 612$ $n = 1836$	$d = 0.46 \times 10^{-5}$ $f = 0.48 \times 10^{-11}$ $k = 22$, $nf = 51$ $n = 153$	$d = 0{,}7 \times 10^{-6}$ $f = 0.61 \times 10^{-11}$ $k = 35$, $nf = 76$ $ne = 228$	$d = 0{,}45 \times 10^{-4}$ $f = 0.42 \times 10^{-9}$ $k = 39$, $nf = 87$ $ne = 261$	$d = 0.14 \times 10^{-7}$ $f = 0.19 \times 10^{-13}$ $k = 45$, $nf = 298$ $ne = 390$
10^{-5}	$d = 0.9 \times 10^{-5}$ $f = 0.21 \times 10^{-10}$ $k = 295$, $nf = 632$ $ne = 1896$	$d = 0.92 \times 10^{-7}$ $f = 0.7 \times 10^{-12}$ $k = 247$, $nf = 747$ $ne = 2241$	$d = 0.96 \times 10^{-11}$ $f = 0.14 \times 10^{-21}$ $k = 25$, $nf = 57$ $ne = 171$	$d = 0.14 \times 10^{-8}$ $f = 0.43 \times 10^{-18}$ $k = 36$, $nf = 83$ $ne = 249$	$d = 0.15 \times 10^{-5}$ $f = 0.14 \times 10^{-10}$ $k = 40$, $nf = 89$ $ne = 267$	$d = 0.12 \times 10^{-7}$ $f = 0.32 \times 10^{-16}$ $k = 46$, $nf = 302$ $ne = 396$
	$f = 0.11 \times 10^{-11}$ $k = 324$, $nf = 690$ $\|g\| = 0.97 \times 10^{-6}$ $ne = 2070$	$f = 0.61 \times 10^{-12}$ $k = 291$, $nf = 879$ $\|g\| = 0.7 \times 10^{-6}$ $ne = 2637$	$f = 0.14 \times 10^{-21}$ $k = 25$, $nf = 57$ $\|g\| = 0.5 \times 10^{-9}$ $ne = 171$	$f = 043 \times 10^{-18}$ $k = 36$, $nf = 83$ $\|g\| = 0.37 \times 10^{-8}$ $ne = 249$	$f = 0.48 \times 10^{-12}$ $k = 41$, $nf = 91$ $\|g\| = 0.62 \times 10^{-6}$ $ne = 273$	$f = 0.32 \times 10^{-16}$ $k = 46$, $nf = 302$ $\|g\| = 0.14 \times 10^{-7}$ $ne = 396$

T.1

*) $min\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
$x^* = (1.2;\ 1),\ f(x^\circ) = 24.2,\ x^* = (1;1)\ f(x^*) = 0.\ \|x^\circ - x^*\| = 2.2.$

$\|x^k - x^*\| \leqslant$	FLETCEER-REEVES (—„—„—)	MODIFIED FLETCHER-REEVES (—·—·—)	DAVIDON-FLETCHER-POWELL (.....)	CAUCHY-NEWTON (——··——)	FLETCHR-REEVES NEWTON $\mu < 1$ (═══)	FLETCHER-REEVES NEWTON (▬ ▪ ▬)
1	$d = 0.89665$ $f = 0.36$ $k = 15.\ nf = 35$ $ne = 105$	$d = 0.91832$ $f = 0.45$ $k = 15,\ nf = 35$ $ne = 105$	$d = 0.94203$ $f = 0.29$ $k = 10,\ nf = 24$ $ne = 72$	$d = 0.99402$ $f = 0.3741$ $k = 18,\ nf = 45$ $ne = 135$	$d = 0.89665$ $f = 0.36054$ $k = 15,\ nf = 35$ $ne = 105$	$d = 0.8089$ $f = 0.3309$ $k = 11,\ nf = 24$ $ne = 72$
10^{-1}	$d = 0.04592$ $f = 0.\ 81 \times 10^{-3}$ $k = 26,\ nf = 56$ $ne = 168$	$d = 0.04396$ $f = 0.82 \times 10^{-3}$ $k = 26, nf = 56$ $ne = 168$	$d = 0.0579$ $f = 0.18 \times 10^{-2}$ $k = 24,\ nf = 54$ $ne = 162$	$d = 0.05266$ $f = 0.0018$ $k = 30,\ nf = 75$ $ne = 225$	$d = 0.06267$ $f = 0.00101$ $k = 23,\ nf = 51$ $ne = 153$	$d = 0.022$ $f = 0.26 \times 10^{-3}$ $k = 29,\ nf = 60$ $ne = 180$
10^{-2}	$d = 0.003703$ $f = 0{,}18 \times 10^{-4}$ $k = 28,\ nf = 60$ $ne = 180$	$d = 0.003324$ $f = 0.15 \times 10^{-4}$ $k = 28,\ nf = 60$ $ne = 180$	$d = 0.0054$ $f = 0.16 \times 10^{-4}$ $k = 26,\ nf = 58$ $ne = 174$	$d = 0.0057$ $f = 0.92 \times 10^{-5}$ $k = 33,\ nf = 83$ $ne = 249$	$d = 0.0079$ $f = 0.54 \times 10^{-4}$ $k = 25,\ nf = 55$ $ne = 165$	$d = 0.34 \times 10^{-3}$ $f = 0.25 \times 10^{-6}$ $k = 32,\ nf = 66$ $ne = 198$
10^{-3}	$d = 0.36 \times 10^{-6}$ $f = 0.43 \times 10^{-10}$ $k = 31,\ nf = 67$ $ne = 201$	$d = 0.2 \times 10^{-6}$ $f = .27 \times 10^{-10}$ $k = 31,\ nf = 67$ $ne = 201$	$d = 0.42 \times 10^{-3}$ $f = 0.52 \times 10^{-7}$ $k = 27,\ nf = 61$ $ne = 183$	$d = 0.18 \times 10^{-4}$ $f = 0.21 \times 10^{-9}$ $k = 36,\ nf = 89$ $ne = 267$	$d = 0.13 \times 10^{-4}$ $f = 0.13 \times 10^{-7}$ $k = 28,\ nf = 61$ $ne = 183$	$d = 0.35 \times 10^{-7}$ $f = 0.62 \times 10^{-13}$ $k = 35,\ nf = 72$ $ne = 216$
10^{-4}	$d = 0.89 \times 10^{-7}$ $f = 0.16 \times 10^{-14}$ $k = 32,\ nf = 69$ $ne = 207$	$d = 0.8 \times 10^{-7}$ $f = 0.14 \times 10^{-14}$ $k = 32,\ nf = 69$ $ne = 207$	$d = 0.73 \times 10^{-5}$ $f = 0.33 \times 10^{-10}$ $k = 28,\ nf = 63$ $ne = 189$	$d = 0.54 \times 10^{-5}$ $f = 0.19 \times 10^{-10}$ $k = 38,\ nf = 94$ $ne = 289$	$d = 0.40 \times 10^{-10}$ $f = 0.36 \times 10^{-18}$ $k = 31,\ nf = 68$ $ne = 204$	—„—
10^{-5}	—„—	—„—	$d = 0.3 \times 10^{-8}$ $f = 0.26 \times 10^{-17}$ $k = 29,\ nf = 66$ $ne = 198$	$d = 0{,}75 \times 10^{-9}$ $f = 0.16 \times 10^{-18}$ $k = 39,\ nf = 97$ $ne = 291$	—„—	—„—
	$f = 0.16 \times 10^{-14}$ $\|g\| = 0.38 \times 10^{-7}$ $k = 32,\ nf = 69$ $ne = 207$	$f = 0.14 \times 10^{-14}$ $\|g\| = 0.35 \times 10^{-7}$ $k = 32,\ nf = 69$ $ne = 207$	$f = 0.26 \times 10^{-17}$ $\|g\| = 0.41 \times 10^{-7}$ $k = 29,\ nf = 66$ $ne = 192$	$f = 0.16 \times 10^{-18}$ $\|g\|^2 = 0.99 \times 10^{-16}$ $k = 39,\ nf = 97$ $ne = 291$	$f = 0.36 \times 10^{-18}$ $k = 31,\ nf = 68$ $\|g\|^2 = 0.72 \times 10^{-15}$ $ne = 204$	$f = 0.62 \times 10^{-13}$ $k = 35,\ nf = 72$ $\|g\|^2 = 0.12 \times 10^{-9}$ $ne = 216$ T.S: $\|g_k\|^2 < 10^{-8}$

*) Rosenbrock's problem STEPG DOUBLE PRECISION

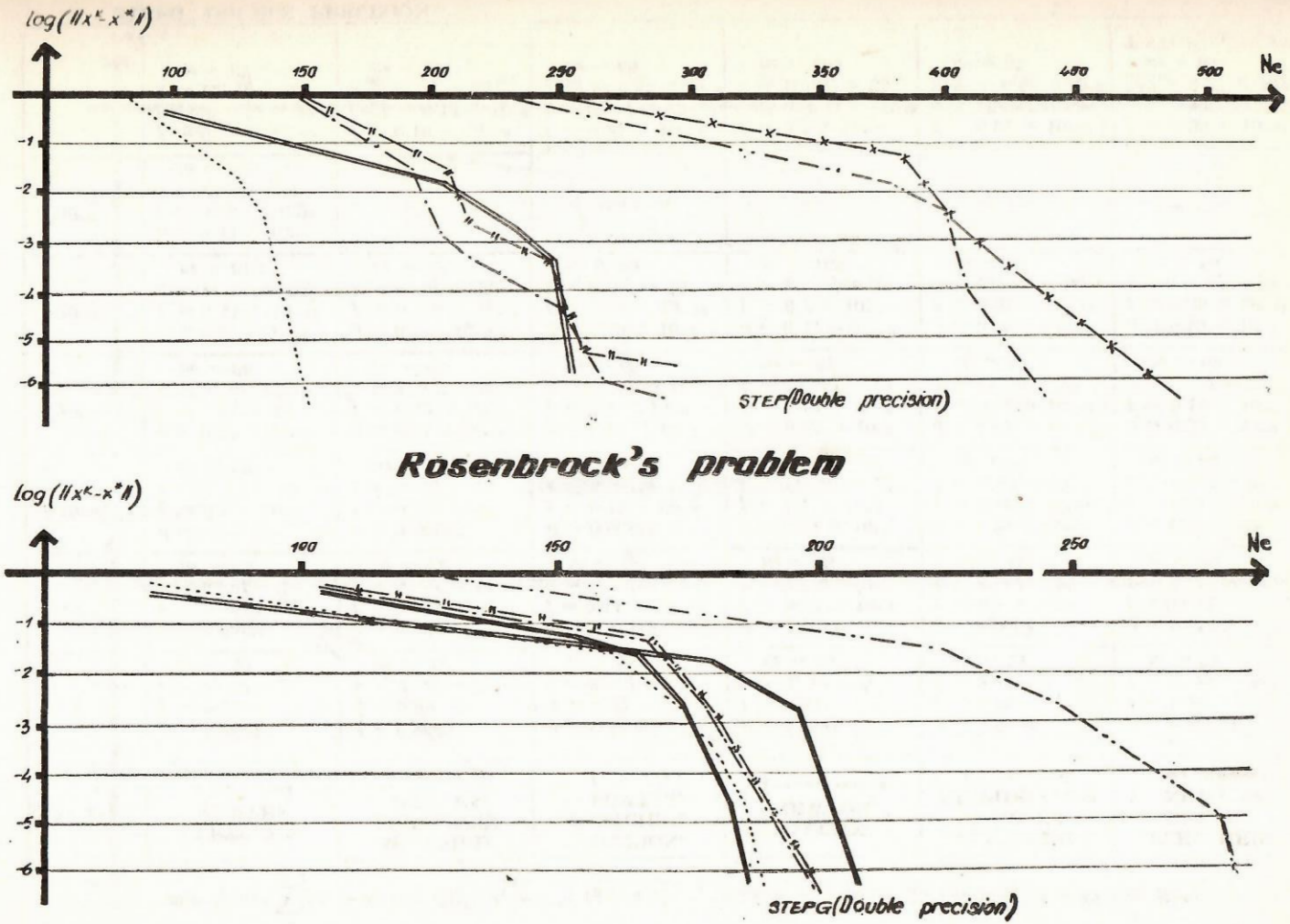

log(||x^k - x*||)
100
150
200
250
300
350
400
450
500
Ne
-1
-2
-3
-4
-5
-6
STEP(Double precision)
Rosenbrock's problem
log(||x^k - x*||)
100
150
200
250
Ne
-1
-2
-3
-4
-5
-6
STEPG(Double precision)

Fig. 1

T.2. **Beale's problem**

$$min\ f(x) = \sum_{i=1}^{3}[c_i - x_1(1 - x_2^i)]^2;\ c_1 = 1.5;\ c_2 = 2.25;\ c_3 = 2.625\ x^\circ = (0.0),\ f(x^\circ) = 14{,}2,\ x^* = (3;0.5),\ f(x^*) = 0$$

$\|x^k - x^*\| \leq$	FLETCHER-REEVES (—„—„—)	MODIFIED FLETCHER-REEVES (—·—·—)	DAVIDON-FLETCHER-POWELL (·····)	CAUCHY-NEWTON, (—·····——)	FLETCHER-REEVES NEWTON ($\mu \leq 1$) (═══)	FLETCHER-REEVES NEWTON (▬▪▬)
1	$d = 0.8602$ $f = 0.29$ $k = 2,\ nf = 7$ $ne = 21$	$d = 0.8602$ $f = 0.29$ $k = 2,\ nf = 7$ $ne = 21$	$d = 0.8602$ $f = 0.29$ $k = 2,\ nf = 7$ $ne = 21$	$d = 0.9165$ $f = 0.34$ $k = 2,\ nf = 5$ $ne = 15$	$d = 0.8602$ $f = 0.29$ $k = 2, nf = 7$ $ne = 21$	$d = 0.86024$ $f = 0.29$ $k = 2,\ nf = 7$ $ne = 21$
10^{-1}	$d = 0.08693$ $f = 0.0013$ $k = 6,\ nf = 15$ $ne = 45$	$d = 0.00945$ $f = 0.15 \times 10^{-3}$ $k = 5,\ nf = 14$ $ne = 42$	$d = 0.01415$ $f = 0.14 \times 10^{-3}$ $k = 5,\ nf = 16$ $ne = 48$	$d = 0.0086$ $f = 0.97 \times 10^{-4}$ $k = 9,\ nf = 21$ $ne = 63$	$d = 0.0024$ $f = 0.1 \times 10^{-4}$ $k = 8,\ nf = 20$ $ne = 60$	$d = 0.03746$ $f = 0.002$ $k = 5,\ nf = 14$ $ne = 42$
10^{-2}	$d = 0.00154$ $f = 0.1 \times 10^{-4}$ $k = 8,\ nf = 20$ $ne = 60$	$d = 0.00677$ $f = 0.7 \times 10^{-5}$ $k = 6,\ nf = 16$ $ne = 48$	$d = 0.00988$ $f = 0.15 \times 10^{-4}$ $k = 6,\ nf = 18$ $ne = 54$	$d = 0.82 \times 10^{-3}$ $f = 0.1 \times 10^{-5}$ $k = 11,\ nf = 27$ $ne = 81$	$d = 0.31 \times 10^{-3}$ $f = 0.16 \times 10^{-7}$ $k = 9,\ nf = 23$ $ne = 69$	$d = 0.59 \times 10^{-3}$ $f = 0.11 \times 10^{-5}$ $k = 8,\ nf = 21$ $ne = 63$
10^{-3}	$d = 0.73 \times 10^{-3}$ $f = 0.81 \times 10^{-7}$ $k = 9,\ nf = 22$ $ne = 66$	$d = 0.54 \times 10^{-4}$ $f = 0.42 \times 10^{-9}$ $k = 8,\ nf = 21$ $ne = 63$	$d = 0.14 \times 10^{-4}$ $f = 0.1 \times 10^{-8}$ $k = 7,\ nf = 21$ $ne = 63$	$d = 0.55 \times 10^{-4}$ $f = 0.52 \times 10^{-9}$ $k = 12,\ nf = 29$ $ne = 87$	$d = 0.84 \times 10^{-8}$ $f = 0.16 \times 10^{-14}$ $k = 11,\ nf = 28$ $ne = 84$	$d = 0.35 \times 10^{-3}$ $f = 0{,}19 \times 10^{-7}$ $k = 9,\ nf = 23$ $ne = 69$
10^{-4}	$d = 0.64 \times 10^{-7}$ $f = 0.54 \times 10^{-13}$ $k = 11,\ nf = 27$ $ne = 81$	$d = 0{,}3 \times 10^{-6}$ $f = 0.19 \times 10^{-13}$ $k = 9,\ nf = 23$ $ne = 69$	$d = 0.12 \times 10^{-5}$ $f = 0.25 \times 10^{-12}$ $k = 8,\ nf = 23$ $ne = 69$	$d = 0.23 \times 10^{-10}$ $f = 0.7 \times 10^{-21}$ $k = 15,\ nf = 35$ $ne = 165$	$d = 0.78 \times 10^{-10}$ $f = 0.91 \times 10^{-21}$ $k = 12,\ nf = 30$ $ne = 90$	$d = 0.15 \times 10^{-7}$ $f = 0.28 \times 10^{-14}$ $k = 11,\ nf = 28$ $ne = 84$
10^{-5}	$d = 0.43 \times 10^{-7}$ $f = 0.28 \times 10^{-15}$ $k = 12, nf = 29$ $ne = 87$	—„—	—„—	—„—	—„—	—„—
	$f = 0.28 \times 10^{-15}$ $\|g_k\| = 0.13 \times 10^{-7}$ $k = 12,\ nf = 29$ $ne = 87$	$f = 0.19 \times 10^{-13}$ $\|g_k\| = 0.14 \times 10^{-6}$ $k = 9,\ nf = 23$ $ne = 69$	$f = 0.25 \times 10^{-12}$ $k = 8,\ nf = 23$ $\|g_k\| = 0.81 \times 10^{-6}$ $ne = 69$	$f = 0.7 \times 10^{-21}$ $\|g\|^2 = 0.61 \times 10^{-19}$ $k = 15,\ nf = 35$ $ne = 165$	$f = 0.91 \times 10^{-21}$ $k = 12,\ nf = 30$ $\|g\|^2 = 0{,}58 \times 10^{-21}$ $ne = 90$	$f = 0.28 \times 10^{-14}$ $k = 11,\ nf = 29$ $\|g\| = 0.27 \times 10^{-12}$ $ne = 84$ T.STOP $\|g_k\|^2 \leq 10^{-8}$

STEPG DOUBLE PRECISION

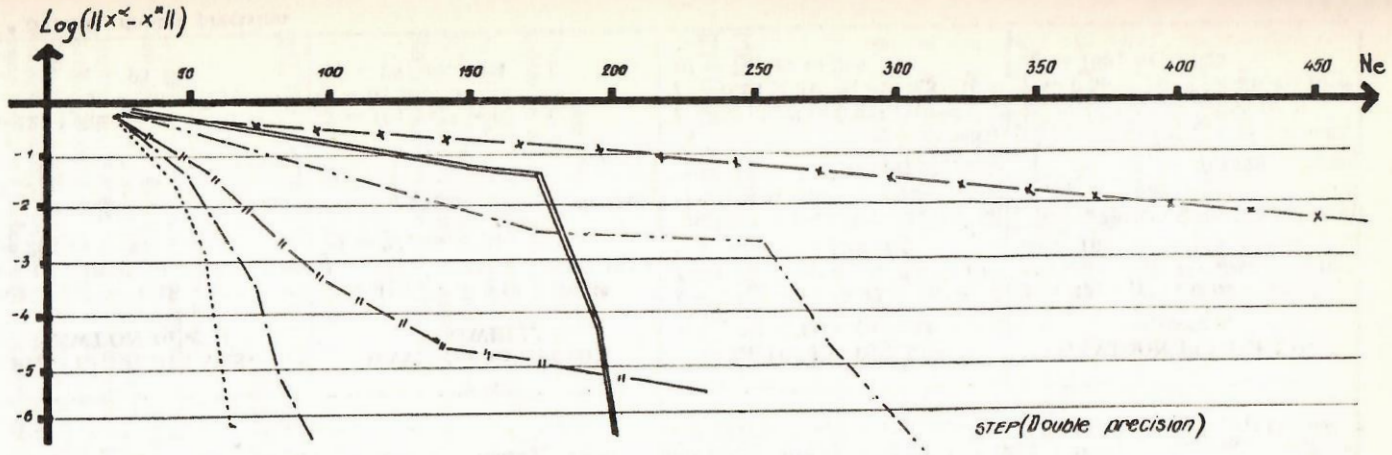
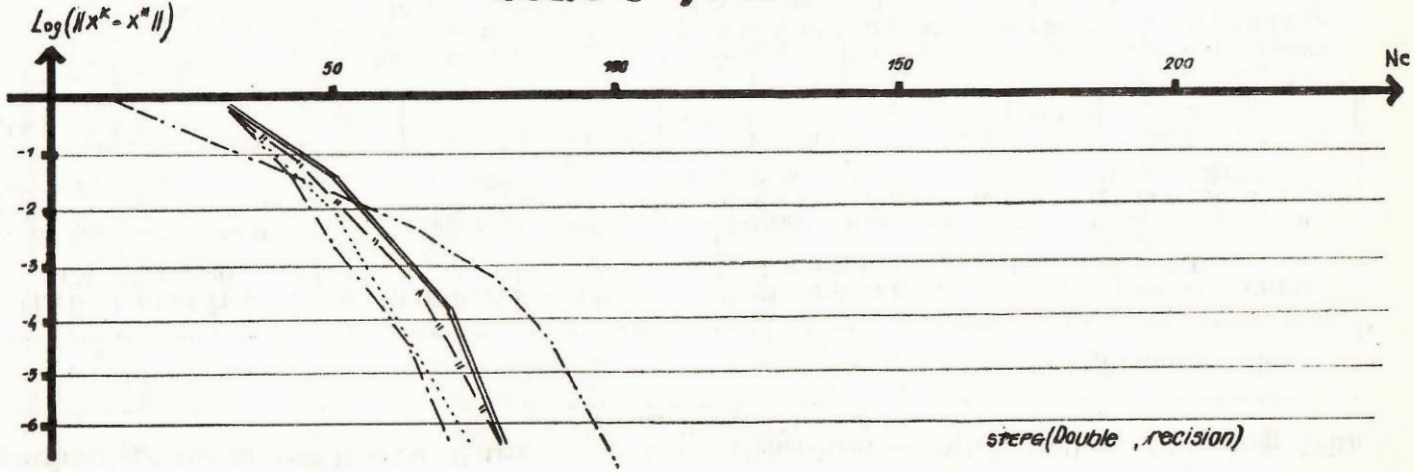


Fig. 2

Beale's problem



Approximating* the Inverse Hessian Matrix Using the Fletcher-Reeves-Newton Method, Comparing With Davidon-Fletcher-Powell Method

<table>
<tr><td colspan="2">$min\ f(x) = 100\ (x_2 - x_1^2)^2 + (1\text{-}x_1)^2$ (Rosenbrock)</td><td colspan="2">$min\ f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$ (Witte & Holst)</td></tr>
<tr><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)</td><td>DAVIDON-FLETCHER-POWELL</td><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)</td><td>DAVIDON-FLETCHER-POWELL</td></tr>
<tr><td>$k = 31,\ \|g\|^2 = 0.72 \times 10^{-15}$
$f = 0.36 \times 10^{-18};\ d = 0.4 \times 10^{-10}$
$nf = 68;\ ne = 204$
$H_k^1 = \begin{pmatrix} 0.50000 & 0.99999 \\ 0.99999 & 2.0050 \end{pmatrix}$</td><td>$k = 29;\ \|g\|^2 = 0.1 \times 10^{-17}$
$f = 0.41 \times 10^{-20};\ d = 0.1 \times 10^{-9}$
$nf = 69;\ ne = 207$
$H_k^{-1} = \begin{pmatrix} 0.50002 & 1.0004 \\ 1.0004 & 2.0058 \end{pmatrix}$
$k = 30;\ \|g\|^2 = 0.$
$f = 0\ ;\ d = 0.$
$nf = 72;\ ne = 216$</td><td>$k = 16;\ \|g\|^2 = 0.28 \times 10^{-9}$
$f = 0.71 \times 10^{-13};\ d = 0{,}94 \times 10^{-7}$
$nf = 38;\ ne = 114$
$H_k^{-1} = \begin{pmatrix} 0.50019 & 1.5003 \\ 1.5003 & 4.5050 \end{pmatrix}$
$k = 17;\ \|g\|^2 = 0.11 \times 10^{-15}$
$f = 0.21 \times 10^{-17};\ d = 0.45 \times 10^{-8}$
$nf = 41;\ ne = 123$</td><td>$k = 34;\ \|g\|^2 = 0.4 \times 10^{-16}$
$f = 0.12 \times 10^{-19};\ d = 0.22 \times 10^{-10}$
$nf = 108;\ ne = 324$
$H_k^1 = \begin{pmatrix} 0.50000 & 1.5000 \\ 1.5000 & 4.5050 \end{pmatrix}$
$k = 35;\ \|g\|^2 = 0.4 \times 10^{-25}$
$f = 0.24 \times 10^{-28};\ d = 0.12 \times 10^{-13}$
$nf = 114;\ ne = 342$</td></tr>
<tr><td colspan="2">$min\ f(x) = \sum_{i=1}^{3} [ci - x_1(1 - x_2^i)]^2,\ c_1 = 1.5;\ c_2 = 2.25;\ c_3 = 2.625$
(Beale)</td><td colspan="2">$min\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1\ [(x_2 - x_1)^2 + (x_4 - x_1)^2] + 19.8\ (x_2 - x_1)(x_4 - x_1)$ (Wood)</td></tr>
<tr><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)</td><td>DAVIDON-FLETCHR-POWELL</td><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)</td><td>DAVIDON-FLETCHER-POWELL</td></tr>
<tr><td>$k = 11;\ \|g\|^2 = 0.16 \times 10^{-12}$
$f = 0.16 \times 10^{-14};\ d = 0.84 \times 10^{-8}$
$nf = 29;\ ne = 87$
$H_k^1 = \begin{pmatrix} 3.1222 & 0.77439 \\ 0.77439 & 0.21376 \end{pmatrix}$
$k = 12;\ \|g\|^2 = 0.58 \times 10^{-21}$
$f = 0.91 \times 10^{-21},\ d = 0.7 \times 10^{-10}$
$nf = 31;\ ne = 93$</td><td>$k = 9;\ \|g\|^2 = 0.16 \times 10^{-20}$
$f = 0.18 \times 10^{-22};\ d = 0.2 \times 10^{-11}$
$nf = 27;\ ne = 81$
$H_k^{-1} = \begin{pmatrix} 3.1238 & 0.7746 \\ 0.7746 & 0.2137 \end{pmatrix}$
$k = 10;\ \|g\|^2 = 0.$
$f = 0;\ d = 0$
$nf = 28;\ ne = 84$</td><td>$k = 29;\ \|g\|^2 = 0.27 \times 10^{-13}$
$f = 0.23 \times 10^{-16};\ d = 0.47 \times 10^{-8}$
$nf = 65;\ ne = 325;\ H_k^{-1} =$
$= \begin{pmatrix} 0.1427; 0.2836; -0.1352; -0.2713 \\ 0.5686;\ -0.2712\ -0.5440 \\ 0.1427;\ 0.2835 \\ 0.5686 \end{pmatrix}$
$k = 30;\ \|g\|^2 = 0.37 \times 10^{-19}$
$f = 0.23 \times 10^{-22};\ d = 0.3 \times 10^{-11}$
$nf = 76;\ ne = 380$</td><td>$k = 37;\ \|g\|^2 = 0.16 \times 10^{-12}$
$f = 0.37 \times 10^{-15};\ d = 0.13 \times 10^{-7}$
$ne = 163;\ ne = 815\ H_k^1 =$
$= \begin{pmatrix} 0.1427; 0.2837; -0.1353; -0.2714 \\ 0.5688; -0.2713; -0.5442 \\ 0.1428;\ 0.2836 \\ 0.5689 \end{pmatrix}$
$k = 38;\ \|g\|^2 = 0.4 \times 10^{-17}$
$f = 0.59 \times 10^{-20};\ d = 0.4 \times 10^{-10}$
$nf = 165;\ ne = 825$</td></tr>
</table>

* STEPG, double precision.

REFERENCES

1. HESTENES, M. R., STIEFEL, E.; *Methods of Conjugate Gradients for Solving Linear Systems.* Report 1659, National Bureau of Standards, (1952).
2. BECKMAN, F. S.: *The Solution of Linear Equations by the Conjugate Gradient Method, in Mathematical Methods for Digital Computers.* Vol., A. Ralston and H. Wilf Editors, John Wiley, New York (1967).
3. FLETCHER, R., REEVES, C. M.: *Function Minimization by Conjugate Gradients.* Computer J., 7 (1964).
4. FLETCHER, R., POWELL, M. J. D.: *A Rapidly Convergent Descent Method for Minimization.* Computer J., vol. 6 (1963).
5. NOBLE, B.: *Applied Linear Algebra.* Prentice-Hall, N. J. (1969).
6. FIACCO, A. V., MC CORMICK, G. P.: *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley, New York (1968).
7. ZANGWILL W. I.: *Nonlinear Programming: A Unified Approach.* Prentice Hall, N. J. (1969).
8. AMUNATEGUI, J., DUMITRU, V.: *Computer Experiments with an Unconstrained Nonlinear Program.* Ec. Comp. and. Ec. Cyb. J. No.3 (1970).
9. DUMITRU, V.: *Gradient Methods for Unconstrained Optimization.* Ec. Comp. and. Ec. Cyb. No. 4 (1974).
10. * * * : *IBM System/360 Scientific Subroutines Package.* Version III (H20—0205—3).

METHODE DES DIRECTIONS CONJUGUEES, APPROXIMATION DU HESSIEN ET UNE NOUVELLE CLASSE D'ALGORITHMES D'OPTIMISATION DE TYPE MIXTE

Hestenes et Stiefel ont présenté un algorithme pour résoudre les systèmes d'équations linéaires, utilisant la méthode des gradients conjugés. C'est normal de penser qu'une telle méthode pourrait être utilisée aussi à l'inversion des matrices.

L'auteur observe que l'inversion d'une matrice définie peut être obtenue dans certaines conditions dans un processus d'optimisation de la forme quadratique associée, fait qui permet, pour les fonctions non-quadratiques, de déterminer une approximation de l'inverse du hessien. De là, en combinant un algoritme de directions conjuguées et un algoritme de type Newton-Raphson, il résulte une nouvelle classe d'algorithmes d'optimisation de type mixte.

МЕТОД СОПРЯЖЕННЫХ НАПРАВЛЕНИЙ, ПРИБЛИЖЕНИЕ МЕТОДА ГЕССА И НОВЫЙ КЛАСС АЛГОРИТМОВ ОПТИМИЗАЦИИ СМЕШАННОГО ТИПА

Гестенес и Штифель дали в труде (1) алгоритм решения систем линейных уравнений с применением метода сопряженных градиентов. Естественно думать, что подобный метод можно было бы применять и для обратного построения матриц.

Автор указывает в дальнейшем, что обратное построение определенной матрицы можно получить в некоторых условиях в процессе оптимизации ассоциированной квадратической формы, используя только величины функций и градиента квадратической формы, что дает возможность для неквадратических функций определять приближение обратного метода Гесса. Отсюда путем сочетания алгоритма сопряженных направлений с алгоритмом типа Ньютона-Рафсона получается новый класс алгоритмов оптимизации смешанного типа.

4.03.1975