An efficient conjugate directions method without linear minimization

A. I. Manevich

*The Department of Theoretical Mechanics and Strength of Materials,
Ukrainian University of Chemical Technology,
8, Gagarina Avenue, 320640 Dniepropetrovsk, Ukraine*

E. Boudinov

NIKHEF, Postbus 41882, NL-1009 DB Amsterdam, The Netherlands

Abstract

An algorithm of conjugate directions for unconstrained minimization is presented, called single-step conjugate directions method SSCDM, that differs from known algorithms in the following points: 1) only one step is made on each iteration, without the linear searches of a minimum; 2) each step includes simultaneous displacements along all conjugate directions found on previous steps (instead of consequent steps along each conjugate vector); 3) each new conjugate direction is determined by the general procedure of construction of conjugate vectors, without assumption that minima along the conjugate directions were reached on previous steps. These features of the algorithm eliminate high sensitivity to computations errors pertinent to usual conjugate directions methods and simultaneously essentially reduce the running time. Performance of the algorithm is demonstrated by applying it to a set of standard test-functions and solving a problem of high energy physics with a Monte-Carlo type objective function.

1 Introduction

The most powerful methods of unconstrained minimization, the conjugate directions algorithms and variable metric methods, were proposed in the fifties and sixties. During the last decades these methods have been improved and various program packages have been elaborated. The most popular in high energy physics is probably the MINUIT Package[1],[2],[3]. But in spite of large efforts and achievements, the efficiency of these algorithms sometimes turns out to be insufficient. When calculation of the objective function and/or its gradient is very time consuming (in particular, in many problems of high energy physics) it is very important to minimize the number of steps, and an improvement of the unconstrained minimization algorithms is of evident practical interest.

It is well known that the main cause of possible failure or slowing-down of the conjugate directions and variable metric methods is the degeneration of the conjugate vectors matrix and the covariance matrix, respectively. This degeneration is not an accidental drawback, but is a difficulty yielding from the essence of the problem. For functions having narrow valleys (only for such functions the minimization is a serious problem) the conjugate vectors matrix and covariance matrix are actually very close to degenerative matrices (the more close the more narrow are valleys). Even for quadratic objective functions the best algorithms turn out to be very sensitive to the rounding errors, and their theoretical high rate of convergence is not realized.

The main source of error accumulation is the one-dimensional minimization along a given line. Most of the unconstrained minimization algorithms are constructed as a sequence of linear minimizations in some directions, and are based on the assumption that minima along the lines were found on previous steps (in particular, such a procedure ensures monotonous decrease of the objective function). However, the linear minimization on each iteration requires a relatively large number of function calculations (of order 10-25). Moreover even very small errors in finding the linear minima are inevitably accumulated and lead to degeneration of governing matrices.

Therefore in order to decrease the danger of the matrices degeneration it is necessary to construct algorithms without the requirement about reaching the linear minimum on previous iterations, but ensuring a decrease of the objective function. In some algorithms of variable metric method, in particular, presented in MINUIT, linear searches are eliminated and replaced simply by Newton step (with the same convergence for a quadratic function), and this procedure is one of the reasons of their efficiency.

For the second class of potentially powerful methods – algorithms of conjugate directions – such an improvement was not until now introduced in computational practice although in the eighties a corresponding modification of the method was proposed and its efficiency has been demonstrated[4]. Note that these methods could have an advantage over the variable metric methods for certain class of objective functions (in particular, in the cases when reliable estimation of second derivatives is impossible, e.g., for Monte-Carlo type functions).

In this paper a modified algorithm of conjugate directions, called the single-step conjugate direction method (SSCDM), is presented. On each iteration a new conjugate direction is determined by the general procedure of construction of conjugate directions, i.e. without the requirement about reaching minima along lines on previous steps. This

enables us to facilitate and improve the strategy of the determination of the direction and length of each step. On each iteration only one step is made, which includes displacements in all conjugate directions found. Such a procedure ensures considerable reduction of sensitivity to calculations errors and a decrease of the running time.

In comparison with the variable metric methods this algorithm is simpler and at the same time reliable and efficient. The performance of the algorithm is demonstrated by applying it to a set of standard test-functions and solving a problem of high energy physics with the Monte-Carlo type objection function. Comparison of the algorithm efficiency with that of MINUIT is presented.

2 Description of the method

Correlations between variables for quadratic function can be excluded by the use of conjugate directions. For non-quadratic functions these correlations can be reduced by this method. The minimum can be found by successive movements along each conjugate direction. In the conjugate directions methods the construction of conjugate vectors is based on the difference of the gradients in two sequential points:

$$\Delta g_k = g_{k+1} - g_k, \quad (g_k \equiv g(x_k)) \quad (1)$$

which is equal to $\mathbf{A}(x_{k+1} - x_k)$ for the quadratic function with hessian matrix $\mathbf{A}$, x is the vector of variables, k is the number of the step. For quadratic functions any vector orthogonal to Δg_k , and the vector $\Delta x_k = x_{k+1} - x_k$ are conjugate vectors (c.v.) with respect to the second derivatives (hessian) matrix. According to the Fletcher-Reeves conjugate gradient method[5], each step Δx_k is performed only along the c.v. p_k , and so after the k -th step the new conjugate vector p_{k+1} is sought as a linear combination of the gradient g_{k+1} and the vector p_k from the orthogonality condition with respect to vector $e_k = \Delta g_k$. As it is assumed that the minimum along p_k is reached on each step, the conjugacy conditions with respect to other conjugate vectors p_i ($i = 1, \dots, k-1$) are not imposed explicitly (they must be satisfied for quadratic functions automatically).

In practice this scheme exhibits inherent shortcomings. Inevitable errors in the determination of the linear minima and rounding errors yield violation of conjugacy condition with respect to p_i ($i = 1, \dots, k-1$), and the disturbance of conjugacy increases from step to step.

In the proposed algorithm on each step a new conjugate vector is determined by the general procedure using conjugacy conditions with respect to all conjugate vectors found previously, without requiring that minima along these vectors were reached on previous steps, i.e. the new conjugate vector is a linear combination of the gradient g_k and p_i ($i = 1, \dots, k-1$) (lower index – the number of c.v., upper index – the step number; it is assumed that $k \leq N$, N is the number of variables):

$$p_k = -g_k + \sum_{i=1}^{k-1} \beta_i^{(k)} p_i \quad (2)$$

Each step includes simultaneous displacements along all conjugate directions that were found on the previous steps:

$$x_{k+1} = x_k + \sum_{i=1}^k \alpha_i^{(k)} \frac{p_i}{\|p_i\|} \quad (3)$$

These displacements, i.e. coefficients $\alpha_i^{(k)}$, are taken as Newton steps. They can be easily determined using the gradients in the last two points and the derivatives along the conjugate vectors ($i = 1, \dots, k-1$):

$$\alpha_i^{(k)} = -C_i^{(k)} g_{p_i}(x_k), \quad C_i^{(k)} = C_i^{(k-1)} \frac{g_{p_i}(x_{k-1})}{g_{p_i}(x_{k-1}) - g_{p_i}(x_k)} \quad (4)$$

where $g_{p_i}(x_k)$ denotes the derivative along p_i in the point x_k . A displacement along a new c.v. p_k can not be calculated according to (4) since the derivative along this c.v. is known only in one point, therefore this displacement is "trial", it can not be determined in a unique fashion and should be chosen using other considerations either as a constant step or varying by a certain rule.

It is obvious that for quadratic functions the minima along p_k would be reached on the $k+1$ -th step; for non-quadratic functions these minima are gradually corrected in passing onto following steps.

As each step includes simultaneous displacements along all conjugate directions, the new c.v. p_k should not be conjugate to the previous step Δx_{k-1} , and so the condition of conjugacy of p_k with respect to p_i ($i = 1, \dots, k-1$) does not coincide now with condition of orthogonality of p_k and Δg_{k-1} . Vectors e_i ($i = 1, \dots, k-1$), to which p_k must be orthogonal, are determined by the recurrent formula:

$$e_{k-1} = \frac{\|p_{k-1}\|}{\alpha_{k-1}^{(k-1)}} \left(\Delta g_{k-1} - \sum_{i=1}^{k-2} \alpha_i^{(k-1)} \frac{e_i}{\|p_i\|} \right) \quad (5)$$

(this expression follows for quadratic functions from (3) after multiplication by $\mathbf{A}$ from the left and substitution k with $k-1$). So the coefficients $\beta_i^{(k)}$ in (2) are determined by the orthogonality conditions ((x, y) denotes the scalar multiplication of vectors x and y):

$$\beta_i^{(k)} = \frac{(g_k, e_i)}{(p_i, e_i)} \quad (6)$$

3 Ensuring convergence for non-quadratic functions

The basic procedure described above should certainly be modified to ensure convergence for non-quadratic functions. If we maintain the set of c.v. at $k > N$ and continue the descent along these vectors, the search can be expected to be inefficient for non-quadratic functions since the hessian matrix does not remain constant, and conjugacy of these vectors is not preserved. In order to account for a changing hessian the set of conjugate vectors should be renewed. Let us consider two possible alternative strategies of c.v. renewal.

3.1 Sequential substitution of each conjugate vector with the new one

This strategy was proposed and discussed in [4]. When $k > N$ one may on each step substitute some conjugate vector with a new c.v., preserving the other c.v.'s. But one has to take into account possible degeneration (linear dependency) of the c.v. set due to the influence of non-quadratic terms and rounding errors. The protection from degeneration can be based on the result proved in [6] (Powell's criterion): the determinant of the matrix whose columns are vectors, scaled so that $(p_i, \mathbf{A} p_i) = 1$, takes its maximum value if and only if the vectors p_i are mutually conjugate. Therefore at $k > N$ the new c.v. $\tilde{p}_{k^*}$ should substitute the old vector p_{k^*} , $k^* = k - tN$, ($t = 1, 2, \dots$), $k^* \leq N$, only under the condition that $\det(p_i)$ increases. Computation of the determinant can be avoided by decomposing the vector $\tilde{p}_{k^*}$ in terms of the vectors p_i , ($i = 1, \dots, N$):

$$\tilde{p}_{k^*} = \sum_{i=1}^N \nu_i p_i \quad (7)$$

Assuming that the p_i are normalized conjugate vectors in the sense that $(p_i, e_j) = \delta_{ij}$, where δ_{ij} is the Kronecker symbol, we have $\nu_i = (\tilde{p}_{k^*}, e_i)$. When introducing $\tilde{p}_{k^*}$ instead of the vector p_{k^*} (which is eliminated from the set of c.v. at determination of p_k by (2)) the change of the determinant depends only upon ν_{k^*} . If the vector $\tilde{p}_{k^*}$ is also scaled, i.e. $(\tilde{p}_{k^*}, e_k) = 1$, the choice between $\tilde{p}_{k^*}$ and p_{k^*} is carried out as follows: if $\nu_{k^*} < 1$, then the vector p_{k^*} remains; if $\nu_{k^*} > 1$, then p_{k^*} is replaced with the vector $\tilde{p}_{k^*}$.

But there is some difficulty because of the necessity of additional calculations: the vector e_k can be determined only after a step, containing displacement along $\tilde{p}_{k^*}$, since one should make the next step, to calculate the vector e_k and then to use the criterion presented.

3.2 Total renewal of the conjugate vectors set

The more simple procedure is to renovate the set of conjugate vectors after several (2-5) cycles (cycle consists of N steps). But the efficiency of such an algorithm depends upon the choice of the direction of the first new conjugate vector and the length of the step along this vector. A too small step does not guarantee sufficient accuracy in the determination of c.v., a too large step can remove the point from an optimum zone (near the minimum) and therefore slow down the convergence.

In (2) the new first conjugate vector on the cycle was chosen along the anti-gradient in the last point. Our experience shows that it is preferable to take this vector along the line connecting the two best points $x^{(s-1)}$, $x^{(s)}$, obtained on the current and previous cycles (one may expect this direction to be close to the direction of a possible valley, similar to the "precipitous step" in the Gelfand algorithm[7]). The initial step length along this direction in the presented algorithm was taken equal to the Newton step (determined using the derivatives along the chosen direction in the two points).

3.3 Other resources to improve efficiency of the algorithm

The effective strategy of the search may be different far from the optimum and in the vicinity of the optimum. Near the optimum the function gradient should be calculated with high accuracy. So for numerical differentiation one should employ central differences. Far from the optimum one-sided differences can be used which require a smaller number of function calls. Besides, near the optimum renewal of the conjugate vectors set should be ceased. In the vicinity of the optimum the quadratic approximation is sufficient, and frequent changes in the set of c.v. would hinder to attain the optimum with high accuracy in calculations given by (4).

The idea of different search strategies far and near the optimum is realized in the proposed algorithm. A criterion of reaching a vicinity of the optimum was chosen as follows:

$$g_k \leq \epsilon_2, \quad \epsilon_2 \approx (10 - 100)\epsilon_1 \quad (8)$$

where ϵ_1 determines the required accuracy of the optimum in the gradient (the minimum is assumed to be found when $g_k \leq \epsilon_1$).

The following safeguards against "unreasonable" steps, as they can occur in highly non-quadratic functions, were introduced in the basic procedure:

a) a restriction on the distances of displacements along each c.v.:

$$\alpha_i^{(k)} \leq \Delta_{max}^{(k)} \quad (9)$$

where $\Delta_{max}^{(k)}$ decreases depending on k according to a certain rule, for example:

$$\Delta_{max}^{(k)} = \frac{\Delta_{max}^{(0)}}{1 + \theta k}, \quad \theta = 0.05 - 0.1 \quad (10)$$

b) a restriction on the change of the coefficients $C_i^{(k)}$ in formula (4):

$$\eta_1 C_i^{(k-1)} \leq C_i^{(k)} \leq \eta_2 C_i^{(k-1)}, \quad \eta_1 = 0.1 - 0.5; \quad \eta_2 = 2 - 5 \quad (11)$$

Apart from these modifications, the algorithm includes some other safeguards and adjustment parameters (changing trial step length along subsequent conjugate vectors, e.g. multiplying an initial length by some factor, restriction on the minimal length of the steps when conjugate vectors are calculated to prevent loss of accuracy, this length varying with k in some way, e.g. similar to (10)).

The program, using this algorithm, is not complicated. It consists of less than three hundred short Fortran lines.

4 Performance of the method (numerical experiments)

The algorithm was applied to some standard test-functions (both quadratic and non-quadratic) and to a problem of high energy physics with a Monte-Carlo type objective function.

All computations were performed with double precision. The gradient was calculated numerically in all problems. Wherever it was possible, results were compared with those of MINUIT (the variable metric method).

4.1 Quadratic functions

We considered the following function of N variables (here the lower index is the number of the variable):

$$f(x_1, \dots, x_N) = \sum_{i=1}^N \frac{x_i^2}{2^{i-1}} + \sum_{i=1}^{N-1} \frac{x_i x_{i+1}}{2^i},$$

with start point $x^{(0)} = (1, \dots, 1)$ and minimum $f(0, \dots, 0) = 0$

Results for various values of N obtained in SSCDM and a comparison with those of MINUIT are given in Table 1.

Table 1: Results for a quadratic function.

	This method			MINUIT	
Number of variables	Number of steps	Number of funct.calls	Accuracy in F	Number of funct. calls	Accuracy in F
N=10	11	242	$0.2 * 10^{-25}$	448	10^{-25}
N=20	21	882	$0.2 * 10^{-27}$	975	10^{-22}
N=30	31	1922	$0.1 * 10^{-27}$	1658	$0.4 * 10^{-25}$
N=40	43	3526	$0.2 * 10^{-21}$	2383	$0.1 * 10^{-21}$

The convergence of SSCDM for quadratic functions turns out to be close to the theoretical limit - for functions of N variables minima were reached after $N + 1$ steps with high accuracy (may be, plus 1 – 4 steps at large N exceeding approximately 40–60). To compare this algorithm with that of MINUIT, the last one was required to reach approximately the same accuracy of the function at the minimum.

Both algorithms are very effective for quadratic functions. It is clear that the most efficient method for quadratic functions is Newton's method. As the MINUIT – algorithm on the first iteration calculates the hessian matrix and then does Newton steps, its efficiency is close to that of the Newton's method. But nevertheless the SSCDM – algorithm does not compare unfavorably with these methods. If we compare the total numbers of function calls in SSCDM and MINUIT (for calculations of derivatives for quadratic functions there were used central differences, so one call of the gradient is equivalent to $2N$ calls of the function) we can see for not very large N ($N \leq 25$) these numbers are less in SSCDM, for large N MINUIT has some advantage. But accuracy, as a rule, is higher in SSCDM.

4.2 Non-quadratic functions

We considered the following test-functions:

NQ1 (Rosenbrock's curved valley, N=2):

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

start point $x^{(0)} = (-1.2, 1)$, minimum $f(1, 1) = 0$

NQ2 (Fletcher and Powell's helical valley, N=3):

$$f(x) = 100\{(x_3 - 10\Psi(x_1, x_2)^2) + (\sqrt{x_1^2 + x_2^2} - 1)^2\} + x_3^2$$

where $2\pi\Psi(x_1, x_2) = \begin{cases} \arctan\frac{x_2}{x_1} & (x_1 > 0) \\ \pi + \arctan\frac{x_2}{x_1} & (x_1 < 0) \end{cases}$

start point $x^{(0)} = (-1, 0, 0)$, minimum $f(1, 0, 0) = 0$

NQ3 (Powell's quartic function, N=4):

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

start point $x^{(0)} = (3, -1, 0, 1)$, minimum $f(0, 0, 0, 0) = 0$

NQ4 (Wood's function, N=4):

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

start point $x^{(0)} = (-3, -1, -3, -1)$, minimum $f(1, 1, 1, 1) = 0$

NQ5 (N=4):

$$f(x) = \sum_{i=1}^{10} (e^{-0.2i} + 2e^{-0.4i} - x_1 e^{-0.2x_2i} - x_3 e^{-0.2x_4i})^2$$

start point $x^{(0)} = (0.5, 0, 2.5, 3)$, minimum $f(1, 1, 2, 2) = 0$

Computations were carried out with full renewal of the conjugate vectors sets after 1–2 cycles and with the first step in "the direction of valley". For the non-quadratic function one cannot speak about any theoretical limit on the required number of the steps in order to reach the minimum of the function. We required both algorithms to search for the minimum with a reasonable accuracy and with a minimum number of the function calls. The results obtained by SSCDM and MINUIT are presented in Table 2 [†]

[†]The following parameters of the SSCDM – algorithm were assumed: $\epsilon_1 = 10^{-4} - 10^{-6}$; $\epsilon_2 = 10^{-1} - 10^{-4}$; $STEP = 0.1 - 0.2$ (initial step length); $\Delta_{max}^{(0)} = 0.4 - 2.0$; $\eta_1 = 0.2 - 0.5$; $\eta_2 = 2.0 - 5.0$, $H1 = 0.00002 - 0.0001$ (step in the calculation of derivatives).

Table 2: Results for non-quadratic test-functions.

	This method			MINUIT	
Function	Number of steps	Number of funct. calls	Accuracy in F	Number of funct. calls	Accuracy in F
NQ1	57	218	$0.6 * 10^{-14}$	214	$0.4 * 10^{-7}$
NQ2	55	272	$0.1 * 10^{-13}$	188	$0.9 * 10^{-6}$
NQ3	37	247	$0.5 * 10^{-9}$	285	$0.1 * 10^{-6}$
NQ4	57	374	$0.1 * 10^{-12}$	785	$0.8 * 10^{-8}$
NQ5	79	595	$0.2 * 10^{-12}$	485	$0.2 * 10^{-9}$

Algorithm SSCDM solves all test-functions successfully. As a rule, it finds minima with better accuracy than MINUIT. The accuracy depended on the value ϵ_1 and might be easily improved by reducing ϵ_1 (with a small increase of the step numbers).

In all test – problems the errors in the determination of the optimal values for the function variables by SSCDM were smaller than 10^{-5} (with exception of function NQ3).

As for the number of function calls by SSCDM, it is usually less than that for MINUIT. Besides, MINUIT does many other operations with the covariance matrix (e.g. calculation of its determinant and eigenvalues), sometimes its accuracy is insufficient. SSCDM is a more simple algorithm and nevertheless it is rather reliable and stable.

It is worthwhile to note that the efficiency of the SSCDM – algorithm for non-quadratic functions (as well as that of MINUIT) strongly depends upon the choice of several control (adjustment) parameters ($\epsilon_1, \epsilon_2, \Delta_{max}^{(0)}, \eta_1, \eta_2$, the differentiation step $H1$, the reduction coefficient θ for maximal and minimal step length). A good choice of these parameters for non-quadratic functions remains a matter of art and intuition of a researcher and can be based on preliminary information about the behaviour of the function. A bad choice of these parameters can result in a pronounced increase of the number of steps, but nevertheless optima are found usually by the SSCDM – algorithm with the required accuracy.

4.3 A Monte-Carlo type objective function

In high energy physics different types of detectors are used to study physical processes. Simulated events are needed to understand the behaviour of the detector and to be able to determine the errors. The simulation of events with a detailed treatment of the detector is usually carried out with Monte-Carlo programs, such as DELSIM[8], GEANT[9].

It is very time-consuming to generate large Monte-Carlo samples. Therefore, to explore the physical processes at a level where the detector details are not important, fast techniques are used[10]. In some cases, the parameterized response of a detector is used in Monte-Carlo programs. The parameterization uses some quantities which reflect the

detector characteristics such as resolutions, detector symmetries, background condition, etc. One can vary these parameters and fit them to the full detector simulation. This parameterization can then be used to study other physical parameters, e.g. efficiencies, purities, acceptance.

The algorithm SSCDM was applied to the Monte-Carlo function used in reference[11] to investigate the systematics on the particle identification by the DELPHI ring-imaging Cherenkov detectors[12]. The function was a χ^2 -function as a result of applying the least squares method to the Monte-Carlo response and the real data. The function $F = \chi^2/(N-3)$ has three variables and equals approximately to unity at the minimum (N=4 is the number of experimental quantities to be compared with the Monte Carlo ones.) The step in the variables in the numerical calculation of the derivatives, H1, was chosen to be 0.2 in order to have a reasonable behaviour of the derivatives. The number of simulated events in the sample was chosen rather large in order to obtain the required accuracy in the variables. The minimum of $F(7.8 \pm 0.1, 7.6 \pm 0.1, 1.4 \pm 0.1) \simeq 1$ was successfully reached after 6 – 12 steps from different starting points : $F(7, 9, 2) = 45$, $F(9, 7, 2) = 54$, $F(7, 7, 1) = 16$, $F(9, 9, 2) = 37$. The parameters found at minimum have variations due to limited statistics of the Monte Carlo function.

5 Conclusions

The modified conjugate direction method (SSCDM) is presented which has essential distinctions from other known methods and gives an efficient, reliable and rather simple technique for unconstrained minimization of functions. The high sensitivity to computations errors pertinent to the usual conjugate directions methods is greatly reduced due to the use of the general procedure for the construction of conjugate vectors. Simultaneously the number of function calls and the running time are strongly decreased due to the absence of the linear minimization on each iteration.

Convergence of SSCDM for quadratic functions turns out to be close to the theoretical limit – for functions of N variables the minimum is reached after $N+1$ steps with very high accuracy. For non-quadratic functions a stable updating scheme for conjugate vectors, different strategies of search far from the optimum and near optimum, separate control for step components and some other features of the algorithm also ensure rapid convergence.

The current version is written in FORTRAN. Due to its simplicity this algorithm can be easily rewritten in C/C++ and included in various program packages.

6 Acknowledgments

It is a pleasure to acknowledge the interest and encouragement of P. van Dam, P. Kluit and J. Timmermans. Their questions and criticisms saved the authors from being on different conjugate directions with the readers.

References

- [1] F.James, M.Roos, *MINUIT – a system for function minimization and analysis of the parameter errors and correlations*. Comput. Phys. Com. **10**,343-367 (1975).
- [2] MINUIT. Function Minimization and Error Analysis. Reference Manual. Version 94.1. CERN Program Library Long Writeup D506. CERN Geneva, Switzerland. 1994.
- [3] Fletcher R. *A new approach to variable metric algorithms* Computer J., 1970, v. 13, No.3, p.317-322.
- [4] Manevich A.I., Polyanchikov P.I. *Single-step method of conjugate directions* Izvestija of USSR Academy of Sciences. Technical Cybernetics (in Russian). 1984, N 6, p. 41-47.
- [5] Fletcher R., Reeves C.M. *Function minimization by conjugate gradients* Computer J., 1964, v. 7, No.2, p.149-154.
- [6] Powell M.J.D. *An efficient method for finding the minimum of a function of several variables without calculating derivatives* Computer J., 1964, v. 7, No 2, p. 155-162.
- [7] Gelfand I.M., Tsetlin M.L. *Principle of nonlocal search in automatic optimization systems* Reports of USSR Acad. of Sci., 1961, v. 137, N 2, p. 295 – 298 (in Russian).
- [8] Yu. Belokopytov et al., DELPHI Detector Description–Application Package User Manual, DELPHI 90-37 PROG-154, 1990.
- [9] Detector Description and Simulation Tool, CERN Program Library Long Writeup W5013.
- [10] J. Cuevas et al., Nucl. Instr. and Meth. **A274** (1989) 459.
- [11] E. Boudinov, P. Kluit, F. Cossutti, K. Huet, M. Gunther and O. Botner, *Measurement of the strange quark forward- backward asymmetry around the Z peak*, DELPHI note 98-115 CONF 177, contributed paper # 127 to the ICHEP’98 Conference, Vancouver 1998.
- [12] W. Adam et al., Nucl. Instr. and Meth. **A343** (1994) 68.