REPORT NO.
UCB/SESM-84/7

STRUCTURAL ENGINEERING AND
STRUCTURAL MECHANICS

COMPUTER AIDED NUMERICAL AND STRUCTURAL OPTIMIZATION BY MEANS OF EVOLUTION STRATEGIES

by

DIETRICH HARTMANN

JUNE 1984

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

Computer Aided Numerical and Structural Optimization by means of Evolution Strategies

Dietrich Hartmann *

Department of Civil Engineering,
Structural Mechanics and Structural Optimization,
University of Dortmund,
Dortmund, West - Germany

ABSTRACT

The theoretical background and concept of the family of evolution strategies is described to provide the necessary information for a potential user interested in structural optimization problems. In particular, the implementation of corresponding optimization programs is demonstrated. Therefore, a variety of test samples is elaborated to give an impression of the program organisation required, and the applicability of the strategies. The main steps necessary to incorporate and enbed the evolution strategies are thoroughly discussed. In order to show the interdependencies of the individual types of strategies a generally valid matrix formulation is used.

July 5, 1984

* Professor Dr.-Ing.; at present, visiting scholar in Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, U.S.A.

ACKNOWLEDGEMENT

This report has been prepared during a stay as visiting scholar in Structural Engineering and Structural Mechanics at the University of California, Berkeley.

The author wishes to express his sincere appreciation for the generous support and fellowship awarded by the Alexander-von-Humboldt-Foundation, which enabled him to perform a research project entitled " Structural Optimization of Finite Element Models ". Portions of this research project form a constituent part of the present report.

Thanks are also due to the Department of Civil Engineering, University of California, Berkeley, for supporting the research project by providing computer facilities.

The author wishes to gratefully acknowledge Prof. G.H. Powell, Department of Civil Engineering, University of California, Berkeley, for his co-operation.

Dietrich Hartmann
Berkeley, U.S.A., June 1984

TABLE OF CONTENTS

1. Description of the applied optimization technique (evolution strategies)

1.1. General comments

The family of evolution strategies (see [1] until [9]) allows the solution of restrained optimization problems with highly nonlinear objective criteria and constraints. Both the objective criterion and the constraints may even have an arbitrarily algorithmical nature, and both may depend on a large number of optimization variables. Therefore, the family of evolution strategies can be regarded as a general purpose solution method within the field of engineering optimization. The individual alternative algorithms of this family of methods are all based on the same logical concept. According to the complicacy of the present optimization problem, the single strategies can be applied individually and/or in combination with each other.

Each of the individual strategies represent a direct solution technique through which potential optima can be iteratively found by means of a sophisticated random process. The constraints permitted must be formulated as inequality constraints where, without restricting generality, feasibility is assumed if the right hand side of the constraints is greater than or equal to 0. However, if equality constraints have to be regarded because they can not eliminated by virtue of separate calculations such equality constraints can be transferred into inequality constraints by using so called slip - variables or Langrangian or penalty formulations.

In the case of structural optimization, we assume that the only equation constraints incorporated are represented by finite element matrix equations , and that these equations can be solved separately from the optimization process. This assumption is advantageous because matured finite element codes are already available which can be efficiently applied to the solution of structural equation systems. Also, we can take advantage of the existing input and output data - bases of the finite element program .

1.2. Convergence behaviour

Convergence of an iterative process is a very significant feature which a user is interesting in because he wants to achieve sufficiently accurate results rapidly. At this point, it should be emphasized that in the field of structural optimization exceptionally complicated problems can occur. The feasible domain may be a highly dimensional space with an intricate topology and multiple local optima. Hence, nonlinear optimization problems are different from other nonlinear problems, for example, nonlinear problems in the structural analysis field where the nonlinear nature of the problem can be more accurately anticipated. Particularly, the quantity "convergence rate", which represents the speed of iteration and is usually conside as the most essential quantity, only characterizes one aspect among further ones if we have . decide on the quality of optimization strategies.

The philosphy of the evolution strategies is to provide numerical methods which are as generally applicable and flexible as possible , and which have reasonable convergence behaviour. However, the features "reliability", "robustness in operation" and "global convergence" are higher ranked than "convergence rate". With regard to convergence rate, the evolution strategies may possibly yield poor convergence rates in special problems,particularly, if linear or quadratic problems have to be solved. It must be outlined , however, that in the engineering field linear or quadratic optimization problems are the exception. Insofar, it is questionable whether these types of problems are representative at all.

The evolution strategies show a linear or superlinear convergence behaviuor depending on the nature of the given problem. That means that the sequence of iteratively calculated design vectors $\mathbf{x}^{(k)}$ converges linearly or superlinearly towards the corresponding optimum solution $\mathbf{x}^*$, where the superscript k denotes the k-th iteration step. We can mathematically express the convergence process by using the following relationship

$$\lim_{k \to \infty} \| \mathbf{x}^{(k)} - \mathbf{x}^* \| \to 0 \tag{1.1}$$

where the term $\| \mathbf{x}^{(k)} - \mathbf{x}^* \|$ represents an error norm (Euclidian norm). If the errors of subsequent iteration cycles are proportionally reduced such that

$$\frac{\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \|}{\| \mathbf{x}^{(k)} - \mathbf{x}^* \|} = c_{lin} \tag{1.2}$$

where c_{lin} stands for a constant proportional factor, the convergence is said to be linear. This means that the number of accurate digits is constantly improved within the iteration steps. A quadratic convergence would be achieved if the number of accurate digits were doubled per iteration step. Hence, superlinear convergence is represented by a reduction of the error which lies between the linear and the quadratic convergence.

Although, quadratic convergence seems to be very attractive this type of convergence can be exclusively obtained by means of first and second partial derivatives of the objective function and the constraints (Newton methods, conjugate gradient methods and special variable metric methods). Despite the fact that the evaluation of derivatives is complicated and time-consuming, divergence difficulties are encountered if the optimization problem is not quadratic or not quasi-quadratic.

Consequently, if we demand to permit all potential types of nonlinearity, we have to omit methods which are confined to special categories of problems. For this reason, only methods with linear or superlinear convergence are of interest. Within the category of such methods, the evolution strategies are preferred to other popular methods because they are distinguished by the following essential features which are already partially mentioned above:

-improved reliability in complicated cases
-appropriate robustness in operation in the multidimensional case
-sufficient probability to also find global optima
-reasonable convergence rates
-insignificant storage demands
-advanced flexibility with changing problems

1.3. Theoretical concept of the evolution strategies

The evolution strategies can be divided into two main categories:

-two-membered $(1+1)$-strategy
-multi-membered $(\mu+\lambda)$ or (μ,λ)-strategies

Herein, the expression "members" means that competitive design vectors $\mathbf{x}$ are simultaneously stored and compared with respect to the given constraints and optimization criterion. The essential computational steps of the these strategies are to be described in the next four chapters. The individual steps are similar in all popular optimization techniques and can be summarized as follows:

- generation of new competitive design vectors
- selection of improved feasible vectors
- adjustment of the optimization parameters
- check upon the convergence and termination criteria.

The discussion of the theoretical concept has the purpose of providing the necessary information for the user to control the optimization process. Along with the discussion, several further possible extensions of the theory will be outlined in order to demonstrate that the method can still be enhanced.

1.3.1. Generation process

1.3.1.1. (1+1)-strategy

In the case of the two-membered strategy two distinct vectors are considered, the currently best vector $\mathbf{x}^{(g)}$ and a new vector $\hat{\mathbf{x}}$ generated from the vector $\mathbf{x}^{(g)}$, both by means of a random process. Therefore, the self-explanatory notation (1+1) is used where each figure stands for a vector which competes with the other . The superscript (g) denotes the corresponding iteration cycle. The hat-symbol indicates that a check still has to be carried out to show whether an improved and feasible vector $\hat{\mathbf{x}}$ was obtained or not. The generation of a new vector $\hat{\mathbf{x}}$ is of crucial significance because the convergence behaviour is substantially affected by the logic of the generation. In general, we can state that the more powerful the variation mechanismi of the generation process are, the more efficient the convergence behaviour is. In the (1+1)-strategy the generation of new design vectors is governed by the simple vector equation

$$\hat{\mathbf{x}} = \mathbf{x}^{(g)} + \mathbf{z}^{(g)} = \mathbf{x}^{(g)} + \mathbf{R}\,\mathbf{s}^{(g)} \tag{1.3}$$

where the vector $\mathbf{z}^{(g)}$ is a random vector in which each of the n components $z_i^{(g)}$; $i=1,2,\ldots,n$ represents a Gauss-distributed step-length in the direction of the corresponding variable x_i. These step-lengths are generated by the random process

$$\mathbf{z}^{(g)} = \mathbf{R}\,\mathbf{s}^{(g)} \tag{1.4}$$

where $\mathbf{R}$ is a diagonal matrix called probability matrix,

$$\mathbf{R} = \begin{bmatrix} R_{11} & 0 & . & . & . & 0 \\ 0 & R_{22} & 0 & . & . & 0 \\ . & 0 & R_{33} & 0 & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{nn} \end{bmatrix} \tag{1.5}$$

and the vector $\mathbf{s}^{(g)}$ denotes the standard deviations of the step-length vector $\mathbf{z}^{(g)}$. The standard deviations represent the changes of the step-lenths and constitute important entities because they mainly influence the convergence rate of the iteration process.

To ensure a minimum of variation of the single variables x_i , $i=1,2,\ldots,n$, and to ensure that the last digit of an optimization variable has at least been varied, the following bounds are established:

$$s_i^{(g)} \geq \epsilon_{abs} \; ; \; i=1,2,3,\ldots,n \tag{1.6a}$$

and

$$s_i^{(g)} \geq \epsilon_{rel} \left| x_i^{(g)} \right| \tag{1.6b}$$

where ϵ_{abs} and ϵ_{rel} connotes the absolute and relative computer accuracy. ($\epsilon_{abs} \geq 0$ and $1+\epsilon_{rel} \geq 1$).

Each of the components R_{ii} , $i=1,2,\ldots,n$ is different and a standardized normally distributed parameter for which the Gauss distribution function $f(r)$ is valid. This function, written in terms of the general stochastic variable r, is defined by the following expression:

$$f(r) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2} r^2} \tag{1.7}$$

The corresponding distribution function $f(r)$ is a bell-shaped function and is shown in Fig. 1-1.

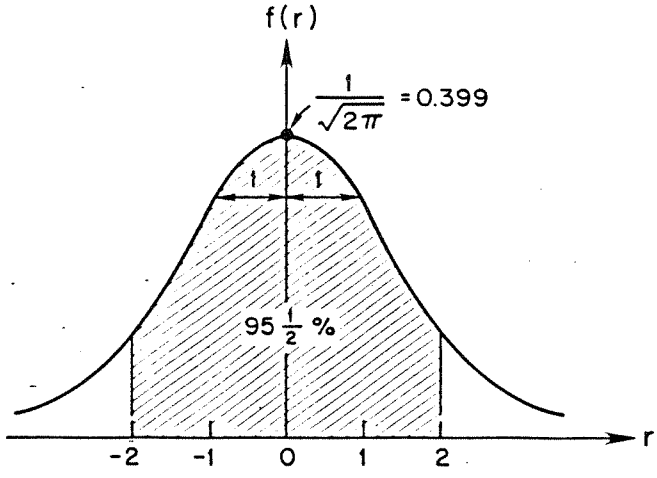

Fig.1-1: Standard normal distribution

It is important to note that the applied distribution function $f(r)$ generates great values of the parameter r less frequently, whereas small values of r are more probable. (This fact will have advantages if we want to approximate finite element equations during a potential re-analysis process to save computational effort. Due to the frequently small changes, we can approximate the corresponding finite element equations by Taylor-series expansions, without making major numerical errors.)

The Gauss normal distribution has been choosen because the generation of new points is considered as a mathematical simulation of the biological evolution process in which changes are also governed by normal distributions. Undoubtedly, the optimization mechanismi of the evolution process are very powerful and ,therefore , worth being simulated analogously within a mathematical optimization technique.

If we multiply the random parameter R_{ii}, for which the function $f(r)$ is valid, with the scalar $s_i^{(g)}$, according to the rules of statistic, we obtain a Gauss distributed parameter $z_i^{(g)}$ having the mean value 0 and the standard deviation $s_i^{(g)}$. The corresponding distribution function is given by

$$f(r\ s_i^{(g)}) = f(z_i^{(g)}) = f(R_{ii}\ s_i^{(g)}) = \frac{1}{\sqrt{2\pi}\, s_i^{(g)}}\ e^{-\frac{1}{2}\left[\frac{z_i^{(g)}}{s_i^{(g)}}\right]^2} \qquad (1.8)$$

The next figure (Fig. 1-2) provides a graphical presentation of the bell-shaped Gauss distributions created.

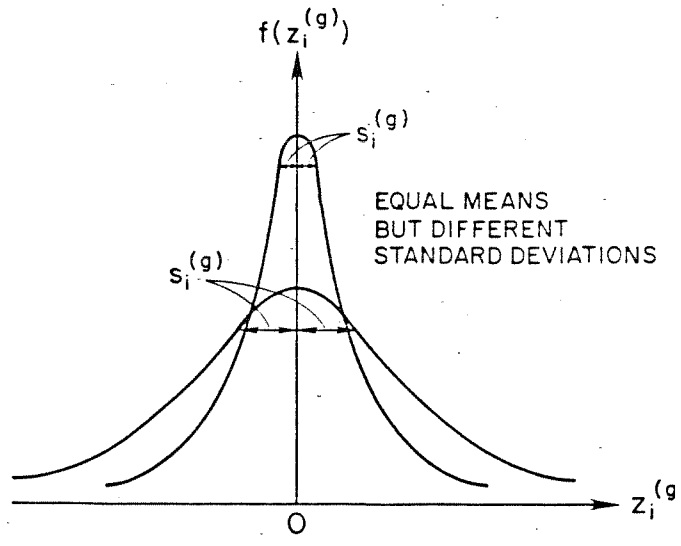

Fig.1-2: Gauss distribution

The product $R_{ii}\ s_i^{(g)}$ represents the step-length in the direction of the design variable x_i. It can be seen that the individual steps $z_i^{(g)}$ are all random and stochastically independent from each other because the matrix **R** is a diagonal matrix. Owing to this, the individual standard deviation $s_i^{(g)}$ only affects the step-lengths $z_i^{(g)}$, having the same index i, but not the other ones. Of course, we could also use a non-diagonal probability matrix which would correlate all or some of the step-lengths. However, the probabilistic model then involved would be inherently much more complex. Up till now, no appropriate theoretical solution could be found. Nevertheless, such a correlation would represent a further improvment of the optimization mechanism.

To illustrate the generation of $\hat{\mathbf{x}}$ vectors, for a two-dimensional case, a graphical impression of the vector equation (1.3) is given in Fig. 1-3 .

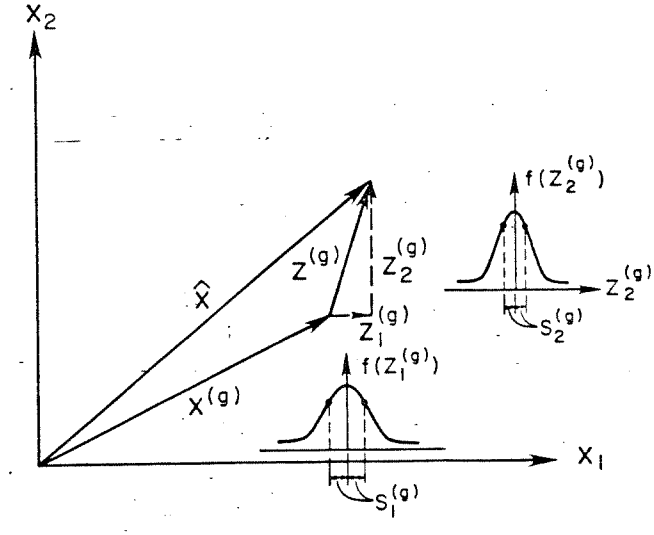

Fig.1-3: Random generation process

Due to the assumed stochastic independence of the individual step-lengths, we obtain a distribution function for the step-length vector which can be written as follows:

$$w(\mathbf{z}^{(g)}) = \prod_{i=1}^{i=n} w(z_i^{(g)}) = \frac{1}{(2\pi)^{\frac{n}{2}} \prod_{i=1}^{i=n} s_i^{(g)}} e^{-\frac{1}{2}\sum_{i=1}^{i=n}\left(\frac{z_i^{(g)}}{s_i^{(g)}}\right)^2} \tag{1.9}$$

By way of example, we can now calculate a first approximation for the mean value of the total length of the vector $\mathbf{z}^{(g)}$. For the sake of simplicity, let us assume that all standard deviations are equal to each other such that

$$s_i^{(g)} = s^{(g)}\ ;\ i=1,2,3,\ldots,n \tag{1.10}$$

This gives:

$$mean(|\mathbf{z}^{(g)}|) = s^{(g)}\sqrt{n} = s^{(g)}\ n^{\frac{1}{2}} \tag{10}$$

Accordingly, the most probable value for the total length of the vector $\mathbf{z}^{(g)}$ increases with respect to the square root of the dimension of the optimization problem. Vice versa, this result allows a very rough but useful estimation of the initial standard deviation s_{init}. If we replace the term $mean|\mathbf{z}^{(g)}|$ by the expected distance $\Delta\mathbf{x}$ between the starting vector and the (unknown) optimum, we obtain

$$s_{init} = \frac{\Delta\mathbf{x}}{\sqrt{n}} \tag{1.12}$$

This value can be used as a first approximation of the initial standard deviations. In addition, the variance can be calculated as well. We obtain:

$$var\ (|\mathbf{z}^{(g)}|) = 2s^{(g)}n^{\frac{1}{4}} \tag{1.12a}$$

Equation (1.9) also indicates that the geometrical location of points which have the same probability is represented by a n-dimensional hyper-ellipsoid. This can be seen by evaluating the expression

$$\sum\left(\frac{z_i^{(g)}}{s_i^{(g)}}\right)^2 = const. \tag{1.13}$$

which obviously constitutes the equation of a n-dimensional hyper-ellipsoid.

Compared with other popular deterministic optimization techniques, the variation mechanism of the (1+1)-strategy is distinguished by the following characteristics:

- -all directions in the n-dimensional space are equally probable, whereas the deterministic methods use fixed directions;
- -the search step-lengths are Gauss distributed values which are characterized by small changes in the average, whereas deterministic methods are using search steps of arbitrary magnitude.

1.3.1.2. Multimembered (μ+λ) and (μ,λ)-strategies

In the case of multimembered methods a vector pool or set of new design vectors $\hat{\mathbf{x}}$ is generated. The governing equation can be written analogously to that of the (1+1)-strategy:

$$\hat{\mathbf{x}}_{\alpha\beta} = \mathbf{x}_\alpha^{(g)} + \mathbf{z}_{\alpha\beta}^{(g)} = \mathbf{x}_\alpha^{(g)} + \mathbf{R}\ \mathbf{s}_{\alpha\beta}^{(g)} \tag{1.14}$$

where the subscript $\alpha=1,2,\ldots,\mu$ designates the vector $\mathbf{x}_\alpha^{(g)}$, which serves as one of the current

basic vectors of the generation process. The subscript $\beta=1,2,...,\nu$ indicates the current number of the new vector generated from the basic vector $\mathbf{x}_\alpha^{(g)}$. Again, the superscript (g) is the iteration cycle counter, and the hat-symbol has the same connotation as described above.

The vector equation for the vector pool can also be written by using only one subscript:

$$\hat{\mathbf{x}}_\gamma = \mathbf{x}_\alpha^{(g)} + \mathbf{z}_\gamma^{(g)} \tag{1.15}$$

where

$$\gamma = 1,2,...,\nu,\nu+1,\nu+2,...,2\nu,2\nu+1,2\nu+2, \cdots ,\mu\nu \tag{1.16}$$

and

$$\mu\nu=\lambda \tag{1.17}$$

In the following we will refer to the notation which only needs one subscript.

Two sub-categories of multimembered strategies can be classified: If we consider both the basic points $\mathbf{x}_\alpha^{(g)}$ and the new points $\hat{\mathbf{x}}_{\alpha\beta}$ or $\hat{\mathbf{x}}_\gamma$, resp. , as potential competitors for the next iteration step $(g+1)$, a $(\mu+\lambda)$ - strategy can be developed. The $(\mu+\lambda)$-notation designates the fact that $\mu+\lambda$ vectors are grouped into the vector pool . To express this feature more precisely, let us introduce a new matrix $\mathbf{X}$ such that

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1 \;\; \hat{\mathbf{x}}_2 \;\; , \ldots , \;\; \hat{\mathbf{x}}_\lambda \;\; | \;\; \mathbf{x}_1^{(g)} \;\; \mathbf{x}_2^{(g)} \;\; , \ldots , \;\; \mathbf{x}_\mu^{(g)}\right] \tag{1.18}$$

where the matrix $\hat{\mathbf{X}}$ represents the vector pool and is ,therefore, called a "pool matrix". (Note that the components of the matrix $\mathbf{X}$ are vectors). However, we will limit the selection of points only to new points generated, even if basic points may represent better constellations. This decision has to do with the fact that we are trying to simulate the biological evolution process. The restriction of using only new points is to simulate the lethality mechanism in the biological evolution process. Hence,we will be considering only vector pools of the following type:

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1 \;\; \hat{\mathbf{x}}_2 \;\; \hat{\mathbf{x}}_3 \;\; , \ldots , \;\; \hat{\mathbf{x}}_\lambda\right] \tag{1.19}$$

As can be seen, the vector pool exactly contains λ vectors which are generated from a vector set of μ vectors. To express this, the notation (μ,λ)-strategy is used,where the comma is to indicate that the μ basic vectors are not members of the vector pool $\mathbf{X}$. The decision to perpetually reject basic points is based on numerical research. It can namely be proved that the convergence rate is not affected by basic vectors being better than succeeding ones if the number λ is a value greater than $5\,\mu$.

The pool concept applied in the (μ,λ)- strategy couples the advantages of two different types of optimization strategies, sequential and simultaneous methods, by minimizing the disadvantages. Simultaneous methods (enumeration, Monto-Carlo-methods, etc.) improve the global convergence; however, they are time and memory consuming. Sequential methods (Newton, gradient and search methods) attempt to maximize the convergence rates; however, they are not reliable in multidimensional problems associated with multiple optima.

1.3.2. Selection process

1.3.2.1. (1+1)-strategy

In the (1+1)-strategy only one new design vector $\hat{\mathbf{x}}_\gamma$ exists. Also, only one basic vector $\mathbf{x}_\alpha^{(g)}$ is used. Thus, $\gamma=1$ and $\alpha=1$, and we can relinquish the subscripts. The new design vector is only accepted as successor of the vector $\mathbf{x}_\alpha^{(g)}$ if it represents an improved value or at least an identical value of the optimization criterion and, of course, if it is admissible. This principle of rejection can be easily encoded.

In order to have a consistent formal notation for both types of strategies, a matrix formulation is to be used. According to the $(\mu+\lambda)$ or (μ,λ)-notation, we obtain a pool matrix into which only two vectors are grouped.

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}} \mid \mathbf{x}^{(g)}\right] \tag{1.20}$$

The vector $\mathbf{x}^{(g+1)}$ of the next iteration cycle can be understood as the result of a matrix multiplication such that

$$\mathbf{x}^{(g+1)} = \hat{\mathbf{X}}\,\hat{\mathbf{B}}\,\mathbf{b}_1 \tag{1.21}$$

where $\hat{\mathbf{B}}$ is a Boolean matrix and $\mathbf{b}_1$ is a unit vector. With respect to the feasible region S^n the matrix $\mathbf{B}$ is defined as follows

$$\hat{\mathbf{B}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ if } f(\hat{\mathbf{x}}) \leqslant f(\mathbf{x}^{(g)}) \; ; \; \hat{\mathbf{x}}, \mathbf{x}^{(g)} \in S^n \tag{1.22}$$

or

$$\hat{\mathbf{B}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \text{ if } f(\hat{\mathbf{x}}) > f(\mathbf{x}^{(g)}) \tag{1.23}$$

The vector $\mathbf{b}_1$ is a vector in which the first component is equal to 1, while the second is 0. Thus

$$\mathbf{b}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{1.24}$$

If the first Boolean matrix holds the new vector $\hat{\mathbf{x}}$ is accepted as the successor. If not the old vector $\mathbf{x}^{(g)}$ is taken. Since the matrix multiplication only has a formal significance, the evaluation is not necessary. Rather, the rejection principle, being more rapid than a time consuming matrix multiplication, can be encoded by a fundamental condition statement.

1.3.2.2. (μ,λ)-strategy

In the (μ,λ)-strategy, we have a pool matrix which contains λ vectors. In order to obtain a code which can be easily programmed ,and to avoid a code which is too memory consuming, we decide on how many of the vectors of the pool are to be selected as basis vectors during the next iteration cycle $(g+1)$. We demand the selection of the μ best vectors of a total of $\mu\nu=\lambda$ potential vectors. This restriction allows us to make repetitve use of the vector generation-equation described above. Such a proceeding results in a constant number of basic vectors $\mathbf{x}_\alpha$, $\alpha=1,2,\ldots,\mu$. (constant population size). Of course, a variable number of basic vectors would be possible (e.g. during a phase of weak convergence, to improve the convergence rate); however, that would lead to a more complex approach. (Development in progress.)

The selection of the μ best vectors can be also described by the following matrix equation

$$\overline{\mathbf{X}} = \hat{\mathbf{X}}\,\hat{\mathbf{B}} \tag{1.25}$$

where $\hat{\mathbf{B}}$ is a Boolean matrix of type (λ,μ). In this matrix all elements are 0 except those elements which have the value of 1, and whose row and column numbers represent the position number of one of the μ best vectors $\hat{\mathbf{x}}_\gamma$ in the pool matrix $\mathbf{X}$. To elucidate the selection in more detail, let us consider the following example

$$\mu=2 \quad \nu=3 \quad \lambda=\mu\nu=6 \tag{1.26}$$

Thus, the pool matrix would be

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1 \; \hat{\mathbf{x}}_2 \; \hat{\mathbf{x}}_3 \; \hat{\mathbf{x}}_4 \; \hat{\mathbf{x}}_5 \; \hat{\mathbf{x}}_6\right] \tag{1.27}$$

Providing that the vectors $\hat{\mathbf{x}}_3$ and $\hat{\mathbf{x}}_4$ represent the best competitors within the pool, the position

numbers are 3 and 4 , respectively. As a consequence, the Boolean matrix $\hat{\mathbf{B}}$ must be written as

$$\hat{\mathbf{B}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.28}$$

Hence,

$$\bar{\mathbf{X}} = \hat{\mathbf{X}}\,\hat{\mathbf{B}} = \left[\hat{\mathbf{x}}_3 \;\; \hat{\mathbf{x}}_4\right] \tag{1.29}$$

The individual basic vectors can be derived from the matrix $\bar{\mathbf{X}}$ by post-multiplication with Boolean vectors $\mathbf{b}_\alpha$, $\alpha=1,2,...,\mu$. The subscript α indicates the row in which the element has the value of 1. All other elements are 0 by definition.

For our example mentioned above we would obtain

$$\mathbf{b}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{1.30a}$$

or, respectively

$$\mathbf{b}_2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \tag{1.30b}$$

Hence

$$\mathbf{x}_1^{(g+1)} = \bar{\mathbf{X}}\,\mathbf{b}_1 = \hat{\mathbf{x}}_3 \; ; \; \mathbf{x}_2^{(g+1)} = \bar{\mathbf{X}}\,\mathbf{b}_2 = \hat{\mathbf{x}}_4 \tag{1.31}$$

Consequently, we can describe the selection process of the (μ,λ) - strategy in accordance with the formulation applied in the (1+1) - strategy. (see equation (1.21)).

$$\mathbf{x}_\alpha^{(g+1)} = \hat{\mathbf{X}}\,\hat{\mathbf{B}}\,\mathbf{b}_\alpha \tag{1.32}$$

Again, it should be emphasized that the applied Boolean matrices have only formal significance and are not needed in the computer program creation because they can be replaced by more efficient and elementary statements having the same effect.

In comparision with the (1+1)-strategy the (μ,λ)-strategy gives improved and enhanced mechanismi which provide an improved reliability. The run-time, by contrast, must be expected to increase due to the multiple vector set. However, it is essential to note that the numerical effort does not proportionally increase with respect to the number of vectors per set, as one could perhaps expect. The reason for that lies in the more powerful optimization mechanismi incorporated which improve the convergence. In addition, it should be mentioned that computer time will be tremendously reduced if one can make use of advanced computer facilities, like modern parallel or pipeline processors. Due to the parallel logic of the (μ,λ)-stategy, the corrsponding algorithms are well suited for implementation with respect to such types of computers. Therefore, the multimembered strategy constitutes a numerical method which has further significant potential, not yet numerically materialized.

1.3.3. Adjustment of the optimization process

1.3.3.1. (1+1)-strategy

In order to achieve a rapid convergence it is obvious that the step-lengths, or their corresponding standard deviations, must have the appropriate size. The Fig. 1-4 illustrates

this demand for a two-dimensional situation with the design variables x_1 and x_2 and a smooth function representing a gradually ascending ridge.

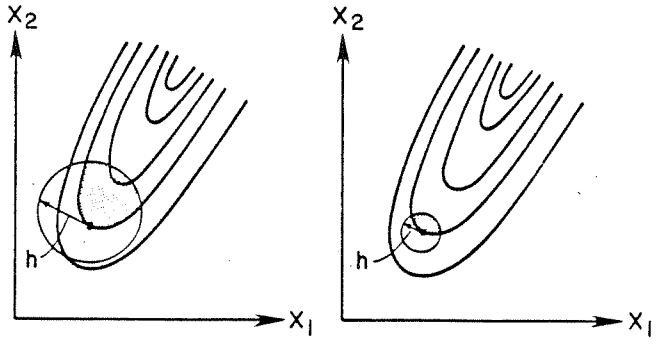

Fig.1-4: Effect of step-sizes

For the sake of simplicity, let us assume that all random step-lengths $\mathbf{z}^{(g)}$ may be generated on a cycle with the radius h. If h is too great the effectiveness of the search is poor, because although one success leads to a great advance , relatively few trials will be successful. (see dotted portion of the cycle). On the other hand, if h is too small the effectiveness is also low, because although a higher proportion of trials will be successful, the absolute movement will be small. Obviously , there must be an optimal values for h that guarantuees an optimal efficiency.

In order to find such an optimal step-length during the optimization, the ratio of number of successes to total number of trials may be used as a measure of convergence. This ratio represents the probability of effectiveness and can be called success probability.

$$w_e = \frac{number\ of\ successful\ trials}{total\ number\ of\ trials} \tag{1.33}$$

At first glance, it seems impossible to specify a value for w_e which may lead to convergence for every arbitary optimization criterion. However, it can be proved that this ratio is almost independent of the nature of the objective criterion. By regarding two special functions, both of which represent diametrically opposite optimization functions within a wide range of potential functions,it could be shown that the optimum value of w_e, guaranteeing the best adjusted step-lengths, varies only between $\frac{1}{3.8}$ and $\frac{1}{5.4}$. Therefore, an approximate value 1/5 was choosen. That means, if

$$w_e = \frac{1}{5} \tag{1.34}$$

the step-lengths are optimal and the assumed standard deviations are correctly dimensioned. If the ratio w_e is less than 1/5 the step-vector $\mathbf{z}^{(g)}$ is to be reduced , if w_e is greater than 1/5 the vector is to be increased.

The magnitude of the reduction or increment is based on a theoretical approach for the special functions that constitutes the inferior case. Theoretical calculations resulted in the fact that all standard deviations must be reduced with the factor $\frac{1}{1.2}$ if w_e is less than 1/5 , and must be increased with the factor 1.2 if w_e is greater than 1/5 . It should be mentioned that the 1/5 -rule used in the (1+1)-strategy makes no assertion about the relative proportions of the single standard deviations. Thus all elements of the vector $\mathbf{z}^{(g)}$ are reduced or incremented at the same time. In the multimembered strategies improved in-built mechanismi provide the

possibility to adjust the relations of the individual elements as well.

To measure the success probability the average value of w_e through several trails is to be calculated. For numerical purposes, the evaluation of the 1/5 - rule is performed such that, after a number of $10n$ trails, the number of successes is checked. Then, it is checked how many successes have been encountered during the last $10n$. Obviously,an amount of successful trials less than $2n$ corresponds to a value of w_e, which is less than $\frac{2n}{10n} = \frac{1}{5}$. In this case, the step-lengths must be reduced. If the amount is greater than $2n$ the step-lengths must be increased.

One particular difficulty is to be mentioned : The 1/5 -rule assumes that there is always a combination of standard deviations by which, on an average , at least one improvement of the optimization criterion within 5 trials is to be expected. However, constellations are conceivable at which this condition is not fulfilled at all times. Such a pathological case may occur on certain parts of the boundary where several constraints may be active with the consequence that the success probability is continously overestimated. Therefore, a premature halt not desired could take place.

This type of phenonema is demonstrated in the following example (see Fig. 1-5).

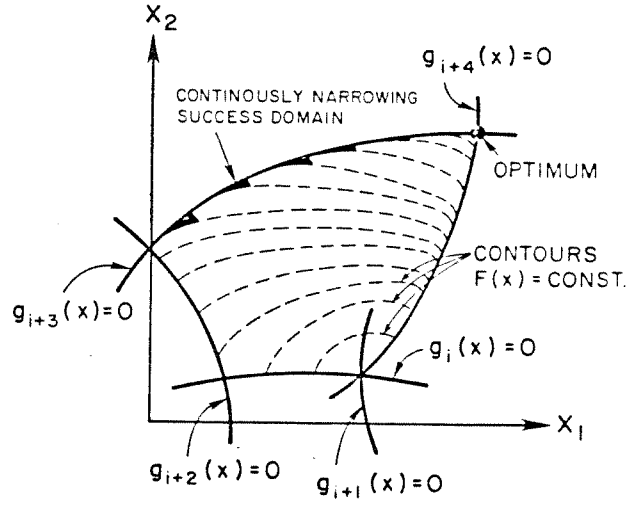

Fig.1-5: Pathological constellation

Using the 1/5 - rule, this example would result in a rapid reduction of the step-lenth because it is impossible to obtain 1 success within 5 trials, in the average. The reason for this is that there is only a very small domain of success (filled-in sector of the cycle). The present case is inherently much more sensitive than the general case represented by the 1/5 - rule. Therefore, it is a good idea to modify the general 1/5 -rule in such a way that a greater value than 5 is used. (In the subroutine EVOL this can be accomplished by increasing the controll parameter JSTEP , default-value = 1.)

Such types of numerical instability are problem dependent, and not a pecularity of the numerical optimzation routine applied. Actually, in other popular methods, numerical difficulties are also encountered. (They are, however, neither mentioned nor are there possibilities to handle such instabilities.) To overcome such instabilities, one could replace the current optimization criterion by an augmented objective criterion, using a Lagrangian or penalty formulation. (The active constraints are temporarily implemented into the augmented optimzation criterion and ,therefore, inactivated). A further way is to employ multimembered evolution strategies which are more powerful in pathological cases.

1.3.3.2. (μ,λ)-strategy

The (1,1)-strategy provides an adjustment of step-lengths which is similar to that of the popular deterministic optimization strategies. By contrast, the (μ,λ)-strategy provides an adjustment-mechanism essentially different. Adjustments are performed analogously to the biological evolution process.

To achieve maximal convergence rates, the standard deviations are additionally incorporated into the random variation and selection process, just as well as the design or optimization variables. That is to say that a design constellation is represented not only by its design variables but also by its standard deviations. Of course, the decision of whether a new design vector $\mathbf{x}$ will be accepted as a basis of the next iteration cycle $(g+1)$ solely depends on the design variables, and not on the values of the standard deviations. However, the magnitude and nature of the succeeding changes of the design variables is influenced by the random nature of the standard deviations associated with these design variables. Basically, the design or optimization variables constitute object parameters whereas variable standard deviations can be understood as strategy parameters. Both types of parameters are subjected to the random control mechanism. Therefore, the 1/5-success-rule used in the (1,1)-strategy is no longer necessary because the (μ,λ)-logic contains a natural success-rule enbodied in it.

However, the 1/5 -rule is replaced by a new rule which requires that the ratio of the number of the basic vectors μ and the number of new vectors λ generated from them must have the right proportion, namely such that

$$\frac{\lambda}{\mu} > 5 \tag{1.35}$$

While the $\frac{\lambda}{\mu}$-rule is fullfilled, eventually better basic design vectors $\mathbf{x}_\alpha^{(g)}$, α=1,2,3,...,μ have no effect on the convergence. Therefore, we preferred the (μ,λ)-strategy to the $(\mu+\lambda)$-strategy. (To a certain extend, that decision already constitutes a portion of the adjustment principle).

Along with the random generation of new basic vectors $\mathbf{x}_\alpha^{(g)}$ the standard deviations are also randomly generated. This random generation is governed by the following vector equation:

$$\mathbf{s}_\alpha^{(g+1)} = \mathbf{Z}\ \mathbf{s}_\alpha^{(g)} \tag{1.36}$$

where $\mathbf{s}_\alpha^{(g+1)}$ designates the vector of the standard deviations of the μ best design vectors $\mathbf{x}_\alpha^{(g+1)}$ within the next generation $(g+1)$. The matrix $\mathbf{Z}$ is a diagonal matrix containing random diagonal elements. Thus

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & 0 & . & . & . & 0 \\ 0 & Z_{22} & 0 & . & . & 0 \\ . & 0 & Z_{33} & 0 & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_{nn} \end{bmatrix} \tag{1.37}$$

The single elements Z_{ii} , ii=1,2,3,...,n are logarithmically normally distributed parameters. This type of distribution function is required because the random values of Z_{ii} must have the mean value 1 , due to the multiplication process, and increments and reductions. Also, increments and reductions generated need to be equally probable, and additionally, great changes of Z_{ii} values are to be less frequently than small ones. The distribution function satisfying these three requirements is a logarithmic normal distribution which can be written in terms of the general variable ζ as follows:

$$f(\zeta) = f(Z_{ii}) = \frac{1}{\sqrt{2\pi}\,\bar{\sigma}}\ \frac{1}{\zeta}\ e^{-\frac{1}{2}\left[\frac{\ln\zeta}{\bar{\sigma}}\right]^2} \tag{1.38}$$

where $\bar{\sigma}$ is an unknown parameter which represents the changes of the standard deviations (

standard deviation of the standard deviation). Although it would be possible to consider individual $\bar{\sigma}$ parameters for each of the subscripts i , $i=1,2,3,...,n$, up till now predictions of convergence can only be made for the particular case of one parameter $\bar{\sigma}$ which is common for all subscripts i. In this elementary case, based on the theory of probability, an appropriate estimation of $\bar{\sigma}$ could be found. (In general, the parameter $\bar{\sigma}$ depends on the number λ and the current topology of the optimization space). Let us consider a representative example: it is a good idea to set the unknown parameter $\bar{\sigma}$ equal to 1 for a (10,100)-strategy, and to increment this value sublinearly. The generation of ζ or Z_{ii}-values, respectively, can be readily encoded by using $(0,\bar{\sigma})$ normally distributed parameters, say ξ, and by evaluating the exponential expression e^{ξ}. Thus

$$Z_{ii} = \zeta = e^{\xi} \tag{1.39}$$

Using a vector pool provides another exceptionally powerful optimization mechanism called scaling or recombination of the individual optimization parameters. Recombination or scaling means the generation of new design vectors through combination of the individual design variables x_i , $i=1,2,3,...,n$ which are associated with the basic vectors $\mathbf{x}_{\alpha}^{(g)}$, $\alpha=1,2,3,...,\mu$. The recombination mechanism for a particular variable, say $\tilde{x}_i$, can be mathematically described by means of a scalar product of two vectors. The first vector (row vector) contains all those elements of the basic vector $\mathbf{x}_{\alpha}^{(g)}$ having the same subscript i. If this vector is denoted $\tilde{\mathbf{X}}_i^T$ we have:

$$\tilde{\mathbf{X}}_i^T = \left\{ x_{1,i}^{(g)} \; ; \; x_{2,i}^{(g)} \; ;...; \; x_{\mu,i}^{(g)} \right\} \tag{1.40}$$

The second vector (column vector), denoted $\mathbf{b}_{\kappa}$, is a Boolean vector with μ-elements all of which are "0", except for one which has the value "1". The position number κ of the value "1" is randomly determined, where all positions $\kappa=1,2,3,...,\mu$ are equally probable. Hence, a particular optimization variable $\tilde{x}_i$ recombined can be written as

$$\tilde{x}_i = \tilde{\mathbf{X}}_i^T \mathbf{b}_{\kappa} \tag{1.41a}$$

If the above scalar product is carried out for all subscripts i , $i=1,2,3,...,n$, a new recombined vector $\tilde{\mathbf{x}}_{\alpha}$ is created.

$$\tilde{\mathbf{x}}_{\alpha} = \left\{ \begin{matrix} \tilde{x}_1 \\ \tilde{x}_2 \\ . \\ . \\ \tilde{x}_n \end{matrix} \right\} ; \; \alpha=1,2,3,...,\mu \tag{1.41b}$$

Each recombined vector is then subjected to the regular random generation process already previously described. In the same fashion the standard deviations can be recombined. Accordingly, we obtain

$$\tilde{\mathbf{s}}_i = \tilde{\mathbf{S}}_i^T \mathbf{b}_{\kappa} \tag{1.41c}$$

and

$$\tilde{\mathbf{s}}_\alpha = \begin{Bmatrix} \tilde{s}_1 \\ \tilde{s}_2 \\ . \\ . \\ \tilde{s}_n \end{Bmatrix} ; \alpha = 1,2,3,\ldots,\mu \tag{1.41d}$$

Finally, it should be emphasized that the recombination process is of substantial significance with respect to global convergence and reliability. Solely by means of the recombination process, appropriately adjusted step-lengths can be achieved, especially in problems with an intricate topology. However, even though the (μ,λ)-strategy incorporates advanced mechanismi, nonlinear optimization problems with very intricated topologies must be carefully treated. The user always has to check the obtained results with regard to plausibility.

1.3.4. Convergence and termination criteria

The optimization is terminated if the CPU-time has elapsed, or if the the optimum is found with respect to the required convergence tolerances.

1.3.4.1. (1+1)-strategy

Convergence is assumed if one of the two required tolerances is achieved

$$f(\mathbf{x}^{(g-\Delta g)}) - f(\mathbf{x}^{(g)}) \leqslant \epsilon_{abs} \tag{1.42}$$

where ϵ_{abs} is the absolute tolerance and Δg indicates the intervall in which the convergence is to be checked. The relative convergence is checked by means of the criterion

$$f(\mathbf{x}^{(g-\Delta g)}) - f(\mathbf{x}^{(g)}) \leqslant \epsilon_{rel} \, | f(\mathbf{x}^{(g)}) | \tag{1.43}$$

where ϵ_{rel} denotes the relative convergence tolerance desired. The retarded check upon the convergence accuracy results in a damping effect with the purpose of avoiding a premature halt in cases where the step-lengths have to be rapidly changed. In the computer program creation it is assumed that

$$\Delta g = 20\,n \tag{1.44}$$

to secure sufficient increments or reductions of the step-lenths within the current Δg-intervall. (For this reason, in the subroutine EVOL the default value of the control parameter NCYCT is set equal to 1; this value corresponds to $\Delta g = 20\,n$ because the intervall is defined by the term NCYCT*JSTEP*10*NOPTV which yields 2*1*10*n = $20\,n$ when the default value of JSTEP=1 is used.)

1.3.4.2. (μ,λ)-strategy

The criteria of the (1+1)-strategy are slightly modified. Convergence is assumed if the values of the optimization criteria associated with the vectors $\mathbf{x}_\alpha^{(g)}$ are close to one another with regard to the absolute and the relative accuracy, ϵ_{abs} and ϵ_{rel} ,respectively. This demand can be mathematically described by means of the difference between the best and the worst value of the optimization criterion within the current set of basic vectors considered. Thus

$$\underset{\alpha}{Min}\left[f(\mathbf{x}_\alpha^{(g)}) \right] - \underset{\alpha}{Max}\left[f(\mathbf{x}_\alpha^{(g)}) \right] = Min\ f - Max\ f \leqslant \epsilon_{abs} \tag{1.45}$$

where

$$\alpha=1,2,3,\ldots,\mu \tag{1.46}$$

The relative convergence ϵ_{rel} is checked by way of the corresponding relative criterion as follows :

$$Min\ f-Max\ f \leqslant \frac{\epsilon_{rel}}{\mu} \sum_{\alpha=1}^{\alpha=\mu} f(\mathbf{x}_{\alpha}^{(g)}) \tag{1.47}$$

1.4. Causing the execution of the (1,1)-strategy EVOL

In order to cause the execution of the (1,1)-strategy EVOL the following FORTRAN - statement (CALL) must be provided in the master:

```
      CALL EVOL ( NOPTV,NICON,ITYPE,JSTEP,NCYCT,TLIMO,
     *            ABACC,REACC,ACOBJ,RCOBJ,RISTP,
     *            FOPTC,XOPTV,SDOPT,XAUXV,  F  ,  G  ,
     *            ELATIM,UNIRAN,GAUNOR )
```

The meaning of the individual actual arguments in the parameter list of the subroutine EVOL is described in table no. 1 in the next two pages.

no.	argument	type	description	comments
1	**NOPTV**	integer	total number of optimization (design) variables	input by user
2	**NICON**	integer	total number of inequality constraints (within the constraints the structural analysis procedure must be enbedded)	input by user
3	**ITYPE**	integer	identifier indicating the type of result of optimization process; The following incidences are possible: ITYPE=-2:::no feasible point available ITYPE=-1:::no feasible point within time limit ITYPE=+0:::infeasible initial point, feasible found ITYPE=+1:::optimization time limit exceeded ITYPE=+2:::optimization process converged	output re-define problem check input restart restart check solution
4	**JSTEP**	integer	parameter for the adjustment of the step-lengths according to the applied controll mechanism; within the (1,1)-evolution strategy the step-lengths for searching the optimum are continously "optimized". According to the theory; the step-lengths are optimal if the ratio "number of successful trials / total number of trials" equals approximately 1/5 in an average. This 1/5-success-rule is used if JSTEP=1. Modifications are possible in special cases (see theoretical concept)	default = 1
5	**NCYCT**	integer	parameter designating the number of iteration cycles after which the convergence tolerances are examined.In accordance to the 1/5-success-rule convergence is checked after 10*NOPTV*JSTEP*NCYCT iteration steps (see theoretical concept).	default = 2
6	**TLIMO**	integer	time limit parameter for optimization according to the time units used in the corresponding system CPU-time routine ELATIM (D).	input by user
7	**ABACC**	real	lower bound for the standard deviations in order to ensure at least a minimum variation of the individual design variables; corresponds with the absolute computer accuracy; via standard deviations Gauss-distributed step-lengths are generated.(e.g. 1.7E-38).	set default
8	**REACC**	real	lower bound of the standard deviations in relation to the design variables to ensure that the last digit of the design varibles is at least been varied; corresponds with the relative accuracy of the applied computer.(e.g. 2E-23)	set default
9	**ACOBJ**	real	absolute convergence tolerance for the objective criterion	input by user
10	**RCOBJ**	real	relative convergence tolerance for the objective criterion	input by user
11	**RISTP**	real	parameter for the reduction or increment of the step-lengths. If the examination of the 1/5 - rule yields a value greater or less, resp., than 1/5 the standard deviations are incremented or reduced, resp.,by the factor RISTP or 1/RISTP, resp.; if the value is equal to the value 1/5 the current standard deviations are kept.The default value is an approximator for general purposes, and was derived from the (1,1)-theory.	default = 0.85

Table No. 1 : Arguments of evolution-strategy EVOL -

no.	argument	type	description	comments
12	**FOPTC**	real	best value of the objective function or optimization criterion, respectively, according to the possible cases ; see ITYPE variable	output
13	**XOPTV**	array	one dimensional array of NOPTV components (optimization variables) for a designn vector $ bold x $. Initially, this array represents the starting values of $ bold x $. Finally, the best or optimum values of $ bold x $ are stored according to the variable ITYPE.	input, also output
14	**SDOPT**	array	one dimensional array of NOPTV components for the standard deviations. By means of the standard deviations, the step-lengths are generated through a random generator which generates GAUSS distributed step-lengths.	input by user
15	**XAUXV**	array	to store new generated optimization vectors $ bold x sup new $, temporarily, an auxiliary array with NOPTV optimization variables is used.	dummy array
16	**F**	function	objective function or criterion according to the given problem for which minimization is assumed; if maximization is required the negative function F must be minimized. The real function F must have the required formal structure described in the next chapter	defined by user
17	**G**	function	name for the set of constraints (side constraints,stress constraints, displacement constraints, etc.). Constraints are assumed to be feasible if the right hand side is ≥ 0. The individual constraint, for example, the J - th constraint, has to be activated by a switch . The structure of the the function G must be the same as described in the next chapter.	defined by user
18	**ELATIM**	function	system routine to determine the CPU-time. The structure of the invoked real function ELATIM must correspond with the requirements outlined in the next chapter.	given
19	**UNIRAN**	function	generator of uniformly distributed pseudo-random values in the intervall (0,1).The structure assumed is UNIRAN (D) where D is a dummy; see description next chapter.	given
20	**GAUNOR**	function	generator of Gauss-normally distributed random values using uniformly distributed random values generated by UNIRAN. The structure of GAUNOR must be GAUNOR (S , UNIRAN) where the formal argument S represents the corresponding component of the vector of the standard deviations , actual name SDOPT (I), I=1,2,3,...,NOPTV; the name UNIRAN designates the name of the generator of uniformly distributed pseudo-random values.	given

Continuation Table No. 1 : Arguments of evolution-strategy EVOL

1.5. Required functions and subroutines

The objective function or optimizationn criterion as well as the inequality constraints depend on the problem. Therefore, both must be formulated by the user. The user has to take notice of the fact that , within the subroutine EVOL, a special standard structure is assumed. In the case of the objective function the following formal definition is required:

```
      REAL FUNCTION F ( NOPTV,XOPTV )
      DIMENSION XOPTV ( NOPTV )
      .......
      .......
      F = ...
      RETURN
      END
```

In the case of the inequality constraints the standard form assumed is such as

```
      REAL FUNCTION G ( J , NOPTV , XOPTV )
      DIMENSION XOPTV ( NOPTV )
      GOTO ( 1,2,3, ..... "NICON" ) , J
    1 G = .....
      RETURN
    2 G = .....
      RETURN
    3 G = .....
      RETURN
      .........
      .........
      .........
      etc.
      RETURN
      END
```

Usually, generators of random-values and system routines for calculating the CPU, turn-around or elapsed time are available in computer center libraries. Such routines should be used, however, they are system dependent. In the following it is described how procedures of the UNIX-operating system can be applied. Other operating systems require one to put in different but analogous system routines instead of those used here.

The elapsed-time is calculated by:

```
      REAL FUNCTION ELATIM ( D )
      DIMENSION TARRAY (2)
      ELATIM = ETIME ( TARRAY )
      RETURN
      END
```

Equally distributed pseudo-random values are generated by:

```
      REAL FUNCTION UNIRAN ( D )
    1 RH = RAND ( 0 )
      IF ( RH .EQ. FLOAT (0) ) GOTO 1
      UNIRAN = RH
      RETURN
      END
```

Gauss-normally distributed values are generated by the following function which makes use of the already defined function UNIRAN.

```
      REAL FUNCTION GAUNOR(S,UNIRAN)
      COMMON/EVZ/LZ
      DATA ZP/6.28318531/
      GOTO(1,2),LZ
1     A=SQRT(-2.*ALOG(UNIRAN(D)))
      B=ZP*UNIRAN(D)
      GAUNOR=S*A*SIN(B)
      LZ=2
      RETURN
2     GAUNOR=S*A*COS(B)
      LZ=1
      RETURN
      END
```

1.6. FORTRAN-Code of the (1+1) - strategy EVOL

The following FORTRAN - subroutine represents the simpliest version of the evolution strategies; the meaning of the formal (dummy) parameters in the parameter list corresponds with that of the actual parameters described in table 1. The association between the actual and dummy arguments is made by their positional correspondence which is represented by the consecutive number in the first column.

```
      SUBROUTINE EVOL(N,M,LF,LR,LS,TM,EA,EB,EC,ED,
     *                SN,FB,XB,SM,X,F,G,T,R,Z)
      DIMENSION XB(1),SM(1),X(1),L(10)
      COMMON/EVZ/LZ
      EXTERNAL R
      TN=TM+T(D)
      LZ=1
      IF(M)4,4,1
1     LF=-1
      FB=0.
      DO 3 J=1,M
      FG=G(J,N,XB)
      IF(FG)2,3,3
2     FB=FB-FG
3     CONTINUE
      IF(FB)4,4,5
4     LF=1
      FB=F(N,XB)
5     DO 6 K=1,10
6     L(K)=N*K/5
      LE=N+N
      LM=0
      LC=0
      FC=FB
7     DO 8 I=1,N
8     X(I)=XB(I)+Z(SM(I),R)
      IF(LF)9,9,12
9     FF=0.
      DO 11 J=1,M
      FG=G(J,N,X)
      IF(FG)10,11,11
10    FF=FF-FG
```

```
11      CONTINUE
        IF(FF)32,32,16
12      IF(M)15,15,13
13      DO 14 J=1,M
        IF(G(J,N,X))19,14,14
14      CONTINUE
15      FF=F(N,X)
16      IF(FF-FB)17,17,19
17      LE=LE+1
        FB=FF
        DO 18 I=1,N
18      XB(I)=X(I)
19      LM=LM+1
        IF(LM-N*LR)7,20,20
20      K=1
        IF(LE-L(1)-N-N)23,22,21
21      K=K-1
22      K=K-1
23      DO 24 I=1,N
24      SM(I)=AMAX1(SM(I)*SN**K,ABS(XB(I))*EB,EA)
        DO 25 K=1,9
25      L(K)=L(K+1)
        L(10)=LE
        LM=0
        LC=LC+1
        IF(LC-10*LS)31,26,26
26      IF(FC-FB-EC)28,28,27
27      IF((FC-FB)/ED-ABS(FC))28,28,30
28      LF=ISIGN(2,LF)
29      RETURN
30      LC=0
        FC=FB
31      IF(T(D)-TN)7,29,29
32      DO 33 I=1,N
33      XB(I)=X(I)
        FB=F(N,XB)
        LF=0
        GOTO 29
        END
```

1.7. Numerical examples

Since there is no optimization problem available which could serve as a test sample generally acknowledged, it is a good idea to check the solution behaviour by using a representative set of distinct optimization problems. For this reason, here 4 appropriate test instances are to be considered by which the efficiency and the reliability of the evolution strategies are demonstrated.

test case no. 1 (MATYAS , 1965)

The test function

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \right\} \tag{1.50}$$

where

$$f(\mathbf{x}) = 0.26(x_1^2+x_2^2)-0.48x_1 \tag{1.51}$$

leads to a malfunction of the popular relaxation methods. These methods cyclically perform linear searches in the direction of the design variables x_i , $i=1,2,3,...,n$. Thus, within the solution space S^n only a limited number of directions is available. Basically, this restriction results in a premature termination of the relaxation methods. After a certain number of iteration steps, no improvement of the objective function is possible by using solely steps in the direction of the design variables. By contrast, the evolution strategies converge properly. Obviously, the solution of the above problem is

$$\mathbf{x}^* = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} ; \; f(\mathbf{x}^*)=0 \tag{1.52}$$

The solution process and the relevant details are represented in Appendix A.

test case no. 2 (SCHWEFEL , 1975)

The second test function is to represent multidimensional and highly nonlinear optimization problems, which may often occur in structural optimization. The objective function is defined by

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \right\} \tag{1.53}$$

where

$$f(\mathbf{x}) = \sum_{i=1}^{i=n} x_i^{10} \tag{1.54}$$

Due to the high order of nonlinearity, the popular conjugate gradient and similar methods diverge. This type of divergence results from the fact that the gradient methods are all based on internal models which assume a quadratic nature of the present optimization problem. If this assumption is not valid, difficulties are encountered because such methods are extremely sensitive to problems of higher than second or third order. The evolution strategies appropriately converge to the optimum

$$\mathbf{x}^* = \begin{Bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} ; \; f(\mathbf{x}^*)=0 \tag{1.55}$$

Again the computational results and details are shown in Appendix A.

test case no. 3 (UEING , 1971)

The third test case is to simulate a typical but elementary structural optimization problem which is distinguished by a set of linear and/or nonlinear constraints. In addition, multiple optima are involved. Also a non coherent feasible domain occurs. The problem can be described as follows:

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \;\middle|\; \mathbf{g}(\mathbf{x})\geq 0\right\} \tag{1.56}$$

where

$$f(\mathbf{x}) = -(x_1^2+x_2^2) \tag{1.57}$$

and

$$\begin{aligned} g_1(\mathbf{x}) &= x_1\geq 0\\ g_2(\mathbf{x}) &= x_2\geq 0\\ g_3(\mathbf{x}) &= -x_1+x_2+4\geq 0\\ g_4(\mathbf{x}) &= x_1/3-x_2+4\geq 0\\ g_5(\mathbf{x}) &= x_1^2+x_2^2-10x_1-10x_2+41\geq 0 \end{aligned} \tag{1.58}$$

Apart from the global minimum

$$\mathbf{x}^* = \begin{Bmatrix}12\\8\end{Bmatrix};\ f(\mathbf{x}^*)=-208 \tag{1.59}$$

there are two further local minima

$$\mathbf{x}' = \begin{Bmatrix}2.02\\4.67\end{Bmatrix};\ f(\mathbf{x}')=-25.91 \tag{1.60}$$

and

$$\mathbf{x}'' = \begin{Bmatrix}6.29\\2.29\end{Bmatrix};\ f(\mathbf{x}'')=-44.86 \tag{1.61}$$

Again, the iteration process is given in Appendix A.

test case no. 4 (ROSEN and SUZUKI , 1965)

The fourth and last test case has the purpose of drawing the user's attention to the fact that, even though the evolution strategies are powerful and robust compared with other candidate methods, they are not capable of solving every type of problem with the highest degree of accuracy. Therefore, in the test case no. 4 an example is choosen which can only be approximatively solved. All popular methods known which solve the restrained problem , however, yield worse solutions. The problem is given by the following relationships:

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \;\middle|\; \mathbf{g}(\mathbf{x})\geq 0\right\} \tag{1.62}$$

where

$$f(\mathbf{x}) = x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4 \tag{1.63}$$

and

$$\begin{aligned} g_1(\mathbf{x}) &= -2x_1^2-x_2^2-x_3^2-2x_1+x_4+5\geq 0\\ g_2(\mathbf{x}) &= -x_1^2-x_2^2-x_3^2-x_4^2-x_1+x_2-x_3+x_4+8\geq 0\\ g_3(\mathbf{x}) &= -x_1^2-2x_2^2-x_3^2-2x_4^2+x_1+x_4+10\geq 0 \end{aligned} \tag{1.64}$$

The solution of the problem is

$$\mathbf{x}^* = \begin{pmatrix} 0 \\ 1 \\ 2 \\ -1 \end{pmatrix}; \; f(\mathbf{x}^*) = -44 \tag{1.65}$$

The results of the corresponding computer program are shown in Appendix A.

1.8. Causing the execution of the (μ,λ)-strategy

In order to invoke the (μ,λ)-strategy GRUP the following FORTRAN - statement, which is ananlogous to the CALL - statement for the subroutine EVOL, must be enbedded into the main progam:

```
      CALL GRUP ( RECOM,NBASV,NPOLV,NOPTV,NICON,
     *            ITYPE,TLIMO,ABACC,REACC,ACOBJ,RCOBJ,RISTP,
     *            FOCUR,FOPTC,XOPTV,SDOPT,XAUXV,
     *            FHIST,XHIST,SHIST,
     *              F   ,  G   ,
     *            ELATIM,UNIRAN,GAUNOR  )
```

Again, the significance of the individual actual parameters is described in a corresponding table (see table no.2 in the next two pages).

no.	argument	type	description	comments
1	**RECOM**	boolean	switch for recombination; if 'true' recombination mode is activated; if 'false' no recombination is carried out.	input by user
2	**NBASV**	integer	total number of basic vectors	input by user
3	**NPOLV**	integer	total number of vectors within the vector pool generated from NBASV basic vectors. It is required that NPOLV > 5*NBASV.	input by user
4	**NOPTV**	integer	total number of optimization (design) variables	input by user
5	**NICON**	integer	total number of inequality constraints (within the constraints the structural analysis procedure must be enbedded)	input by user
6	**ITYPE**	integer	identifier indicating the type of result of optimization process; The following incidences are possible: ITYPE = -2:::no feasible point available ITYPE = -1:::no feasible point within time limit ITYPE = +0:::infeasible initial point, feasible found ITYPE = +1:::optimization time limit exceeded ITYPE = +2:::optimization process converged	output re-define problem check input restart restart check solution
7	**TLIMO**	integer	time limit parameter for optimization according to the time units used in the corresponding system CPU-time routine ELATIM (D).	input by user
8	**ABACC**	real	lower bound for the standard deviations in order to ensure at least a minimum variation of the individual design variables; corresponds with the absolute computer accuracy;	set default
9	**REACC**	real	lower bound of the standard deviations in relation to the design variables to ensure that the last digit of the design varibles is at least been varied; corresponds with the relative accuracy of the applied computer.	set default
10	**ACOBJ**	real	absolute convergence tolerance for the objective criterion	input by user
11	**RCOBJ**	real	relative convergence tolerance for the objective criterion	input by user
12	**RISTP**	real	parameter for the reduction or increment of the steplengths. Standard value assumed is : CONSTANT/ SQRT (NOPTV), where Constant > 0, e.g. in the case of NBASV = 10 and NPOLV = 100 the value of CONSTANT is to be equal to 1.0; if NPOLV is increased the value of CONSTANT should be also (superlinearly) increased;here, (10,100)-strategy is assumed; therefore, CONSTANT = 1 is used; if NBASV is increased the value of CONSTANT should be reduced. In other cases, CONSTANT ought be appropriately estimated.	calculated
13	**FOCUR**	real	best value of objective function or optimization criterion within the total optimization process at all. If the difference between FOPTC and FOCUR is great the solution may be not unique.Two cases are possible: 1. during the optimization a potential minimum was abandoned ; 2. numerical instabilities occur in the vicinity of the minimum due to the finite accuracy of the computer.	output

Table No. 2 : Arguments of evolution-strategy GRUP

no.	argument	type	description	comments
14	**FOPTC**	real	best value of the objective function or optimization criterion, respectively, with respect to the NBASV vectors of current basic vectors	output
15	**XOPTV**	array	one dimensional array of NOPTV components (optimization variables) for a design vector **x** . Initially, this array represents the starting values of **x**. Finally, the best or optimum values of **x** are stored according to the variable ITYPE.	input, also output
16	**SDOPT**	array	one dimensional array of NOPTV components for the standard deviations. By means of the standard deviations, the step-lengths are generated through a random generator which generates GAUSS distributed step-lengths.	input by user
17	**XAUXV**	array	to store new generated optimization vectors $\mathbf{x}^{new}$, temporarily, an auxiliary array with NOPTV optimization variables is used.	dummy array
18	**FHIST**	array	array of the length 2*NBASV in order to store the best values of the optimization criterion of the two last sets of basic vectors; history of the best values of the optimization criterion. (array length must fit dimension statement !).	output
19	**XHIST**	array	array of the length 2*NOPTV*NBASV;represents the history of the design variables of the last two sets of basic vectors; the first NOPTV locations of the array correspond with the first basic vector,the second ones with the second, etc. (array length must fit dimension statement !).	output
20	**SHIST**	array	array for the standard deviations corresponding to the organisation of the array XHIST.	output
21	**F**	function	objective function or criterion; minimization is assumed; if maximization is required the negative function F must be minimized. The real function F must have the required formal structure.	defined by user
22	**G**	function	name for the set of constraints Constraints are assumed to be feasible if the right hand side is ≥ 0. The individual constraint, for example, the J - th constraint, has to be activated by a switch . The structure of the the function G has to fit the formal requirements.	defined by user
18	**ELATIM**	function	system routine to determine the CPU-time. The structure of the invoked real function ELATIM must correspond with the formal requirements.	given
19	**UNIRAN**	function	generator of uniformly distributed pseudo-random values of the intervall (0,1).The structure assumed is UNIRAN (D) where D is a dummy; see formal requirements.	given
20	**GAUNOR**	function	generator of Gauss-normally distributed random values using equally distributed random values generated by UNIRAN. The structure of GAUNOR must be GAUNOR (S , UNIRAN) where the formal argument S represents the corresponding component of the vector of the standard deviations , actual name SDOPT (I), I=1,2,3,...,NOPTV;	given

Continuation Table No. 2 : Arguments of evolution-strategy GRUP

1.9. Required functions and subroutines

All details which were already discussed in chapter 1.5 hold for the subroutine GRUP. The reader is requested to re-read the comments.

1.10. FORTRAN - Code of the (μ,λ)-strategy

The following FORTRAN - subroutine represents the higher order (μ,λ)-strategy. The meaning of the formal parameters in the parameter list corresponds with that of the actual parameters described in table 2 .

```
      SUBROUTINE GRUP(REKO,L,LL,N,M,LF,TM,EA,EB,EC,ED,
     *SN,FA,FB,XB,SM,X,FK,XK,SK,F,G,T,R,Z)
      LOGICAL REKO
      DIMENSION XB(1),SM(1),X(1),FK(1),XK(1),SK(1)
      COMMON/GRZ/LZ
      EXTERNAL R
      KK(RR)=(LA+IFIX(FLOAT(L)*RR))*N
      TN=TM+T(D)
      LZ=1
      IF(M)4,4,1
1     LF=-1
      FB=0.
      DO 3 J=1,M
      FG=G(J,N,XB)
      IF(FG)2,3,3
2     FB=FB-FG
3     CONTINUE
      IF(FB)4,4,5
4     LF=1
      FB=F(N,XB)
5     DO 6 I=1,N
      SK(I)=AMAX1(SM(I),ABS(XB(I))*EB,EA)
6     XK(I)=XB(I)
      FK(1)=FB
      KA=N
      KB=0
      DO 21 K=2,L
      SA=1.
7     DO 8 I=1,N
8     X(I)=XB(I)+Z(SM(I)*SA,B)
      IF(LF)9,9,12
9     FF=0.
      DO 11 J=1,M
      FG=G(J,N,X)
      IF(FG)10,11,11
10    FF=FF-FG
11    CONTINUE
      IF(FF)60,60,17
12    IF(M)16,16,13
13    DO 15 J=1,M
      IF(G(J,N,X))14,15,15
14    SA=SA*.5
      GOTO 7
15    CONTINUE
```

```
16      FF=F(N,X)
17      IF(FF-FB)18,19,19
18      FB=FF
        KB=K
19      DO 20 I=1,N
        KA=KA+1
        SK(KA)=AMAX1(SM(I)*SA,ABS(X(I))*EB,EA)
20      XK(KA)=X(I)
21      FK(K)=FF
        IF(KB)24,24,22
22      KB=(KB-1)*N
        DO 23 I=1,N
23      XB(I)=XK(KB+I)
24      LA=L
        LB=0
25      LC=LB
        LB=LA
        LA=LC
        LC=0
        LD=0
26      SA=EXP(Z(SN,R))
        IF(REKO)GOTO 28
        KI=KK(R(D))
        DO 27 I=1,N
        KI=KI+1
        SM(I)=SK(KI)*SA
27      X(I)=XK(KI)+Z(SM(I),R)
        GOTO 30
28      SA=SA*0.5
        DO 29 I=1,N
        SM(I)=(SK(KK(R(D))+I)+SK(KK(R(D))+I))*SA
29      X(I)=XK(KK(R(D))+I)+Z(SM(I),R)
30      IF(LF)31,31,34
31      FF=0.
        DO 33 J=1,M
        FG=G(J,N,X)
        IF(FG)32,33,33
32      FF=FF-FG
33      CONTINUE
        IF(FF)60,60,38
34      IF(M)37,37,35
35      DO 36 J=1,M
        IF(G(J,N,X))46,36,36
36      CONTINUE
37      FF=F(N,X)
38      LD=LD+1
        IF(LD-L)40,40,39
39      IF(FF-FS)41,41,46
40      KS=LB+LD
41      FK(KS)=FF
        KS=(KS-1)*N
        DO 42 I=1,N
        KS=KS+1
        SK(KS)=AMAX1(SM(I),ABS(X(I))*EB,EA)
```

```
42      XK(KS)=X(I)
        IF(LD-L)46,43,43
43      KS=LB+1
        FS=FK(KS)
        DO 45 K=2,L
        KA=LB+K
        FF=FK(KA)
        IF(FF-FS)45,45,44
44      FS=FF
        KS=KA
45      CONTINUE
46      LC=LC+1
        IF(LC-LL)26,47,47
47      IF(LD-L)26,48,48
48      KA=LB+1
        FA=FK(KA)
        FC=FA
        DO 50 K=2,L
        KB=LB+K
        FF=FK(KB)
        FC=FC+FF
        IF(FF-FA)49,50,50
49      FA=FF
        KA=KB
50      CONTINUE
        IF(FA-FB)51,51,53
51      FB=FA
        KB=(KA-1)*N
        DO 52 I=1,N
52      XB(I)=XK(KB+I)
53      IF(FS-FA-EC)55,55,54
54      IF((FS-FA)*FLOAT(L)/ED-ABS(FC))55,55,59
55      LF=ISIGN(2,LF)
56      KB=(KA-1)*N
        DO 57 I=1,N
57      X(I)=XK(KB+I)
58      RETURN
59      IF(T(D)-TN)25,56,56
60      DO 61 I=1,N
61      XB(I)=X(I)
        FB=F(N,XB)
        FA=FB
        LF=0
        GOTO 58
        END
```

1.11. Numerical examples

Here the same test samples are discussed as previously (see paragraph 1.7.) in order to provide the opportunity to compare both the (1+1) and the (μ,λ)-strategy with each other. For the sake of convenience, all test samples are again summarized.

test case no. 1 (MATYAS , 1965)

The test function

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \right\} \tag{1.66}$$

where

$$f(\mathbf{x}) = 0.26(x_1^2+x_2^2)-0.48x_1 \tag{1.67}$$

leads to a malfunction of the popular relaxation methods. These methods cyclically perform linear searches in the direction of the design variables x_i , $i=1,2,3,...,n$. Thus, within the solution space S^n only a limited number of directions is available. Basically, this restriction results in a premature termination of the relaxation methods. After a certain number of iteration steps, no improvement of the objective function is possible by using solely steps in the direction of the design variables. By contrast, the evolution strategies converge properly. Obviously, the solution of the above problem is

$$\mathbf{x}^* = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} ; \; f(\mathbf{x}^*)=0 \tag{1.68}$$

The solution process and the relevant details are represented in the Appendix A.

test case no. 2 (SCHWEFEL , 1975)

The second test function is to represent multidimensional and highly nonlinear optimization problems, that may often occur in structural optimization. The objective function is defined by

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \right\} \tag{1.69}$$

where

$$f(\mathbf{x}) = \sum_{i=1}^{i=n} x_i^{10} \tag{1.70}$$

Due to the high order of nonlinearity, the popular conjugate gradient and similar methods diverge. This type of divergence results from the fact that the gradient methods are all based on internal models which assume a quadratic nature of the present optimization problem. If this assumption is not valid, difficulties are encountered because such methods are extremely sensitive towards problems of higher than second or third order. The evolution strategies appropriately converge to the optimum

$$\mathbf{x}^* = \begin{Bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} ; \; f(\mathbf{x}^*)=0 \tag{1.71}$$

Again the computational results and details are shown in Appendix A.

test case no. 3 (UEING , 1971)

The third test case is to simulate a typical but elementary structural optimization problem which is distinguished by a set of linear and/or nonlinear constraints. In addition, multiple optima are involved. Also a non coherent feasible domain occurs. The problem can be described as follows:

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \;\middle|\; \mathbf{g}(\mathbf{x})\geqslant 0\right\} \tag{1.72}$$

where

$$f(\mathbf{x}) = -(x_1^2+x_2^2) \tag{1.73}$$

and

$$\begin{aligned} g_1(\mathbf{x}) &= x_1\geqslant 0\\ g_2(\mathbf{x}) &= x_2\geqslant 0\\ g_3(\mathbf{x}) &= -x_1+x_2+4\geqslant 0\\ g_4(\mathbf{x}) &= x_1/3-x_2+4\geqslant 0\\ g_5(\mathbf{x}) &= x_1^2+x_2^2-10x_1-10x_2+41\geqslant 0 \end{aligned} \tag{1.74}$$

Apart from the global minimum

$$\mathbf{x}^* = \begin{Bmatrix}12\\8\end{Bmatrix};\ f(\mathbf{x}^*)=-208 \tag{1.75}$$

there are two further local minima

$$\mathbf{x}' = \begin{Bmatrix}2.02\\4.67\end{Bmatrix};\ f(\mathbf{x}')=-25.91 \tag{1.76}$$

and

$$\mathbf{x}'' = \begin{Bmatrix}6.29\\2.29\end{Bmatrix};\ f(\mathbf{x}'')=-44.86 \tag{1.77}$$

Again, the iteration process is given in Appendix A.

test case no. 4 (ROSEN and SUZUKI , 1965)

The fourth and last test case has the purpose of drawing the user's attention to the fact that, even though the evolution strategies are powerful and robust compared with other candidate methods, they are not capable of solving every type of problem with the highest degree of accuracy. Therefore, in the test case no. 4 an example is choosen which can only be approximatively solved. All popular methods known which solve the restrained problem , however, yield worse solutions. The problem is given by the following relationships:

$$\min_{\mathbf{x}\in S^n}\left\{ f(\mathbf{x}) \;\middle|\; \mathbf{g}(\mathbf{x})\geqslant 0\right\} \tag{1.78}$$

where

$$f(\mathbf{x}) = x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4 \tag{1.79}$$

and

$$\begin{aligned} g_1(\mathbf{x}) &= -2x_1^2-x_2^2-x_3^2-2x_1+x_4+5\geqslant 0\\ g_2(\mathbf{x}) &= -x_1^2-x_2^2-x_3^2-x_4^2-x_1+x_2-x_3+x_4+8\geqslant 0\\ g_3(\mathbf{x}) &= -x_1^2-2x_2^2-x_3^2-2x_4^2+x_1+x_4+10\geqslant 0 \end{aligned} \tag{1.80}$$

The solution of the problem is

$$\mathbf{x}^* = \begin{Bmatrix} 0 \\ 1 \\ 2 \\ -1 \end{Bmatrix}; \; f(\mathbf{x}^*) = -44 \tag{1.81}$$

The results of the corresponding computer program are shown in Appendix A.

2. References

[1] I. RECHENBERG, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Fromann-Holzboog Verlag, Stuttgart, 1977

[2] D. HARTMANN, *Optimierung balkenartiger Zylinderschalen aus Stahlbeton mit elastischem und plastischem Werkstoffverhalten.* Dissertation,Universitaet Dortmund, D290 , 1974

[3] D. HARTMANN, *Zur Systematik und Methodik in der Strukturoptimierung.* Habilitationsschrift, Universitaet Dortmund, 1978

[4] D. HARTMANN, *A generally applicable structural optimization method and its application to cylindrical shells,* Engineering Optimization, Vol.3, (1978), pp. 201 -213

[5] D. HARTMANN, *Optimization of hyperbolic paraboloidal shells - A problem of algorithmically nonlinear programming,* 20.IASS-World Congress, Madrid, September 1979, pp. 5.265 - 5.277

[6] D. HARTMANN, *Optimization in CAD - On the applicability of nonlinear evolution strategies for optimization strategies in CAD,* IFIP-Conference on Optimization in CAD, Lyon, France, 1983

[7] D.HARTMANN, *Computer aided-structural optimization - Enbedding of the finite element method into numerical nonlinear optimization,* ASME, Design Automation Conference, September 11 -14, Detroit, Michigan, USA, Paper No. 83 - DET - 59, 1983

[8] H.P. SCHWEFEL, *Numerical optimization of computer models,* John Wiley & Sons , New York , 1982

[9] M. LAWO, *Automatische Bemessung von dynamisch belasteten Stabtragwerken.* Dissertation, Universitaet Essen, 1981

- APPENDIX A

- EVOL
- EXAMPLE 1
- EXAMPLE 2
- EXAMPLE 3
- EXAMPLE 4
- GRUP
- EXAMPLE 1
- EXAMPLE 2
- EXAMPLE 3
- EXAMPLE 3.1
- EXAMPLE 3.2
- EXAMPLE 3.3
- EXAMPLE 4

=== EVOL ===

=== test program for the (1 + 1) - strategy ===

==========

```
C     ( 1 , 1 ) - OPTIMIZATION
C     ================================================================
C     BERKELEY, U.S.A., APRIL 84
C     ================================================================
C     DECLARATIONS
C     ================================================================
      COMMON /GRZ/LZ
      COMMON /TF/ TARRAY (2)
      COMMON /IO/JINP,JOUT
      EXTERNAL F4,G4,ELATIM,GAUNOR,UNIRAN    ! F4 and G4 have to be replaced !
      DIMENSION XOPTV(20), SDOPT(20), XAUXV(20)
      INTEGER*2 TITLE(40)
      DATA JINP, JOUT /5,6/
C     ================================================================
C     INPUT / OUTPUT OF CONTROL PARAMETERS
C     ================================================================
      READ (JINP,5000) (TITLE(I),I=1,40)
5000  FORMAT(40A2)
      WRITE(JOUT,6000) (TITLE(I),I=1,40)
6000  FORMAT(1X,40A2)
      READ (JINP,*) NOPTV,NICON,TLIMO
      PRINT * ,'NOPTV = NUMBER OF OPTIMIZATION VARIABLES    =',NOPTV
      PRINT * ,'NICON = NUMBER OF CONSTRAINTS               =',NICON
      PRINT * ,'TLIMO = TIME - LIMIT                        =',TLIMO
      READ (JINP,*) JSTEP,NCYCT,RISTP
      PRINT * ,'JSTEP = STEP-ADJUSTMENT PARAMETER           =',JSTEP
      PRINT * ,'NCYCT = NUMBER OF CYCLES FOR TOL.CONVERGENCE=',NCYCT
      PRINT * ,'RISTP = REDUCTION/INCREMENT PARAMETER       =',RISTP
      READ(JINP,*) ABACC,REACC,ACOBJ,RCOBJ
      PRINT * ,'ABACC = ABSOLUTE COMPUTER ACCURACY          =',ABACC
      PRINT * ,'REACC = RELATIVE COMPUTER ACCURACY          =',REACC
      PRINT * ,'ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE      =',ACOBJ
      PRINT * ,'RCOBJ = RELATIVE CONVERGENCE TOLERANCE      =',RCOBJ
C     ================================================================
C     INPUT AND OUTPUT INITIAL VECTOR
C     ================================================================
      DO 5 I=1,NOPTV
      READ (JINP,*) XOPTV(I)
5     CONTINUE
      CALL SCRIPT (1,XOPTV,NOPTV,JOUT)
      DO 6 I=1,NOPTV
      READ(JINP,*) SDOPT(I)
6     CONTINUE
      CALL SCRIPT (2,SDOPT,NOPTV,JOUT)
C     ================================================================
C     PRE-CALCULATIONS
C     ================================================================
      RANDOM=RAND(1)
      PRINT * ,'RANDOM = INITIAL VALUE OF RANDOM SEQUENCE   =',RANDOM
      IF ( NICON .EQ. 0 ) GOTO 8
      DO 7 J=1,NICON
      CONSTRAINT = G4 ( J,NOPTV,XOPTV )
      PRINT * ,'INITIAL CONSTRAINT NO. ',J,' =',CONSTRAINT
7     CONTINUE
8     FINIT= F4 (NOPTV,XOPTV)
      PRINT * ,'FINIT = INITIAL VALUE OF OPT. CRITERION     =',FINIT
```

```
C     ================================================================
C     INVOKE OPTIMIZATION ROUTINE
C     ================================================================
      TIME=ELATIM(D)
      CALL EVOL (NOPTV,NICON,ITYPE,JSTEP,NCYCT,
     1           TLIMO,ABACC,REACC,ACOBJ,RCOBJ,RISTP,
     2           FOPTC,XOPTV,SDOPT,XAUXV,
     3           F4,G4,ELATIM,UNIRAN,GAUNOR)
      TIME=ELATIM(D)-TIME
C     ================================================================
C     RESULTS
C     ================================================================
      PRINT * ,'TIME  = ELAPSED TIME IN SEC                    =',TIME
      PRINT * ,'ITYPE = TYPE OF RESULT                         =',ITYPE
      PRINT * ,'FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL    =',FOPTC
      CALL SCRIPT (3,XOPTV,NOPTV,JOUT)
      IF ( NICON .EQ. 0 ) GOTO 10
      DO 9 J=1,NICON
      CONSTRAINT = G4 ( J,NOPTV,XOPTV )
      PRINT * ,'FINAL  CONSTRAINT NO. ',J,' =',CONSTRAINT
9     CONTINUE
10    CALL SCRIPT (4,SDOPT,NOPTV,JOUT)
      STOP
      END
C     ================================================================
C     ================================================================
      SUBROUTINE EVOL(N,M,LF,LR,LS,TM,EA,EB,EC,ED,SN,FB,XB,
     1SM,X,F,G,T,R,Z)
      DIMENSION XB(1),SM(1),X(1),L(10)
      COMMON/EVZ/LZ
      EXTERNAL R
      TN=TM+T(D)
      LZ=1
      IF(M)4,4,1
1     LF=-1
      FB=0.
      DO 3 J=1,M
      FG=G(J,N,XB)
      IF(FG)2,3,3
2     FB=FB-FG
3     CONTINUE
      IF(FB)4,4,5
4     LF=1
      FB=F(N,XB)
5     DO 6 K=1,10
6     L(K)=N*K/5
      LE=N+N
      LM=0
      LC=0
      FC=FB
7     DO 8 I=1,N
8     X(I)=XB(I)+Z(SM(I),R)
      IF(LF)9,9,12
9     FF=0.
      DO 11 J=1,M
      FG=G(J,N,X)
```

```
      IF(FG)10,11,11
10    FF=FF-FG
11    CONTINUE
      IF(FF)32,32,16
12    IF(M)15,15,13
13    DO 14 J=1,M
      IF(G(J,N,X))19,14,14
14    CONTINUE
15    FF=F(N,X)
16    IF(FF-FB)17,17,19
17    LE=LE+1
      FB=FF
      DO 18 I=1,N
18    XB(I)=X(I)
19    LM=LM+1
      IF(LM-N*LR)7,20,20
20    K=1
      IF(LE-L(1)-N-N)23,22,21
21    K=K-1
22    K=K-1
23    DO 24 I=1,N
24    SM(I)=AMAX1(SM(I)*SN**K,ABS(XB(I))*EB,EA)
      DO 25 K=1,9
25    L(K)=L(K+1)
      L(10)=LE
      LM=0
      LC=LC+1
      IF(LC-10*LS)31,26,26
26    IF(FC-FB-EC)28,28,27
27    IF((FC-FB)/ED-ABS(FC))28,28,30
28    LF=ISIGN(2,LF)
29    RETURN
30    LC=0
      FC=FB
31    IF(T(D)-TN)7,29,29
32    DO 33 I=1,N
33    XB(I)=X(I)
      FB=F(N,XB)
      LF=0
      GOTO 29
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION UNIRAN (D)
1     RH=RAND(0)
      IF (RH.EQ.FLOAT(0)) GO TO 1
      UNIRAN=RH
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION GAUNOR(S,R)
      COMMON/GRZ/LZ
      DATA ZP/6.28318531/
      GOTO(1,2),LZ
1     A=SQRT(-2.*ALOG(R(D)))
```

```
      B=ZP*R(D)
      GAUNOR=S*A*SIN(B)
      LZ=2
      RETURN
2     GAUNOR=S*A*COS(B)
      LZ=1
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION ELATIM (D)
      DIMENSION TARRAY (2)
C     ================================================================
C     ELATIM   YIELDS  ELAPSED TIME IN SECONDS
C     ================================================================
      ELATIM=ETIME (TARRAY)
      PRINT * , '           "TIME" =  ' , ELATIM
      RETURN
      END
C     ================================================================
C     ================================================================
      SUBROUTINE SCRIPT ( IPARM, VEKT,LVEKT,JOUT )
      DIMENSION VEKT ( LVEKT )
      GOTO ( 10 , 20 , 30 , 40 , 50 , 60 , 70 ), IPARM
10    WRITE (JOUT,100)
      DO 15 I=1,LVEKT
      WRITE (JOUT,1000) I,VEKT(I)
15    CONTINUE
      GOTO 99
20    WRITE (JOUT,200)
      DO 25 I=1,LVEKT
      WRITE (JOUT,2000) I,VEKT(I)
25    CONTINUE
      GOTO 99
30    WRITE (JOUT,300)
      DO 35 I=1,LVEKT
      WRITE (JOUT,3000) I,VEKT(I)
35    CONTINUE
      GOTO 99
40    WRITE (JOUT,400)
      DO 45 I=1,LVEKT
      WRITE (JOUT,4000) I,VEKT(I)
45    CONTINUE
      GOTO 99
50    WRITE (JOUT,500)
      DO 55 I=1,LVEKT
      WRITE (JOUT,5000) I,VEKT(I)
55    CONTINUE
      GOTO 99
60    WRITE (JOUT,600)
      DO 65 I=1,LVEKT
      WRITE (JOUT,6000) I,VEKT(I)
65    CONTINUE
      GOTO 99
70    WRITE (JOUT,700)
      DO 75 I=1,LVEKT
```

```
      WRITE (JOUT,7000) I,VEKT(I)
75    CONTINUE
99    RETURN
100   FORMAT (/,'INITIAL CONSTELLATION: ',/,22('-'),/,
     *         'STARTING VECTOR: ',/,16('-'),/)
200   FORMAT (/,'STANDARD DEVIATIONS OF THE',/,26('-'),/,
     *         'INITIAL VECTOR COMPONENTS: ',/,26('-'),/)
300   FORMAT (/,'OPTIMUM VECTOR: ',/,15('-'),/)
400   FORMAT (/,'STANDARD DEVIATIONS OF THE',/,26('-'),/,
     *         'FINAL VECTOR COMPONENTS: ',/,24('-'),/)
500   FORMAT (/,'VALUES OF THE LAST TWO BASIC',/,28('-'),/,
     *         'VECTOR SETS ( HISTORY ): ',/,24('-'),/)
600   FORMAT (/,'CORRESPONDING VARIABLES: ',/,24('-'),/)
700   FORMAT (/,'CORRESPONDING DEVIATIONS: ',/,25('-'),/)
1000  FORMAT (6X,'XOPTV(',I2,')',10X,E15.8)
2000  FORMAT (6X,'SDOPT(',I2,')',10X,E15.8)
3000  FORMAT (6X,'XOPTV(',I2,')',10X,E15.8)
4000  FORMAT (6X,'SDOPT(',I2,')',10X,E15.8)
5000  FORMAT (6X,'FHIST(',I3,')',10X,E15.8)
6000  FORMAT (6X,'XHIST(',I3,')',10X,E15.8)
7000  FORMAT (6X,'SHIST(',I3,')',10X,E15.8)
      END
C     ================================================================
      REAL FUNCTION F1 ( N,X )
      DIMENSION X ( N )
C     ================================================================
C     TEST-FUNCTION ACCORDING TO MATYAS ( 1965 )
C     ================================================================
      H1=X(1)*X(1)
      H2=X(2)*X(2)
      H12=X(1)*X(2)
      F1 = 0.26 * (H1+H2) - 0.48 * H12
      RETURN
      END
C     ================================================================
C     ================================================================
C     ================================================================
      REAL FUNCTION G1 (J,N,X)
      DIMENSION X(N)
      GOTO (1,2),J
1     G=0.
      RETURN
2     G=0.
      RETURN
      END
C     ================================================================
      REAL FUNCTION F2 ( N,X )
      DIMENSION X (N)
C     ================================================================
C     TEST-FUNCTION ACCORDING TO SCHWEFEL ( 1978 )
C     ================================================================
      SUM=0.0
      DO 5 I=1,N
      SUM=SUM + X  (I) ** 10
5     CONTINUE
      F2 = SUM
```

```
      RETURN
      END
C     ================================================================
C     ================================================================
C     ================================================================
      REAL FUNCTION G2 (J,N,X)
      DIMENSION X(N)
      GOTO (1,2),J
1     G=0.
      RETURN
2     G=0.
      RETURN
      END
C     ================================================================
      REAL FUNCTION F3 ( N,X )
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO UEING
C     ================================================================
      H1=X(1)*X(2)
      H2=X(2)*X(2)
      F3 = -H1 - H2
      RETURN
      END
C     ================================================================
      REAL FUNCTION G3 (J,N,X)
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO UEING ( 1967 ) , MODIFIED
C     ================================================================
      GOTO ( 1,2,3,4,5,6,7 ) , J
1     G3 = X(1)
      RETURN
2     G3 = X(2)
      RETURN
3     G3 = - X(1) + X(2) + 4.0
      RETURN
4     G3 = X(1)/3.0 - X(2) + 4.0
      RETURN
5     G3 = X(1)*X(1) + X(2)*X(2) - 10.*X(1) - 10.*X(2) + 41.0
      RETURN
6     G3 = -X(1) + 7.0
      RETURN
7     G3 = -X(2) + 7.
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION G4 (J,N,X)
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO ROSEN AND SUZUKI
C     ================================================================
      GOTO ( 1,2,3 ) , J
1     G4 = -2.*X(1)*X(1) - X(2)*X(2) - X(3)*X(3) - 2.*X(1) + X
      RETURN
```

```
2       G4 = -X(1)*X(1) - X(2)*X(2)
     *      -   X(1)    +     X(2)    -
        RETURN
3       G4 = -X(1)*X(1) - 2.*X(2)*X(2) - X(
     *      +   X(1)
        RETURN
        END
C       ================================================
```

=== test sample # 1 for EVOL ===

```
**************************  INPUT - FILE  ************************
TEST SAMPLE NO. 1
2, 0, 1000. 00,
1, 2, 0. 85,      ,
1. 7E-37, 1. 0E-7, 1. 7E-15, 1. 0E-7,
15. 0
30. 0,
5. 0,
5. 0,
**************************  OUTPUT-FILE  *************************

 TEST SAMPLE NO. 1
 ----------------

  NOPTV = NUMBER OF OPTIMIZATION VARIABLES     =  2
  NICON = NUMBER OF CONSTRAINTS                =  0
  TLIMO = TIME - LIMIT                         =   1000. 00
  JSTEP = STEP-ADJUSTMENT PARAMETER            =  1
  NCYCT = NUMBER OF CYCLES FOR TOL. CONVERGENCE=  2
  RISTP = REDUCTION/INCREMENT PARAMETER        =  0. 850000
  ABACC = ABSOLUTE COMPUTER ACCURACY           =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY           =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE       =   1. 70000e-15
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE       =   1. 00000e-07

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0. 15000000e+02
      XOPTV( 2)            0. 30000000e+02

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 50000000e+01
      SDOPT( 2)            0. 50000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE    =  0. 513870
  FINIT = INITIAL VALUE OF OPT. CRITERION      =   76. 5000
  TIME  = ELAPSED TIME IN SEC                  =   2. 36667
  ITYPE = TYPE OF RESULT                       =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL  =   4. 92918e-19

OPTIMUM VECTOR:
---------------

      XOPTV( 1)            0. 34603100e-08
      XOPTV( 2)            0. 28411822e-08

STANDARD DEVIATIONS OF THE
--------------------------
FINAL VECTOR COMPONENTS:
```

SDOPT(1)	0.63937531e-08
SDOPT(2)	0.63937531e-08

=== test sample # 2 for EVOL ===

```
**************************  INPUT - FILE  **************************
TEST SAMPLE NO. 2
3, 0, 1000. 00,
1, 2, 0. 85,
1. 7E-37, 1. 0E-7, 1. 7E-35, 1. 0E-15,
-1. 0,
2. 0
1. 0,
1. 0,
1. 0,
1. 0,
**************************  OUTPUT - FILE  **************************

 TEST SAMPLE NO. 2
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES     =  3
  NICON = NUMBER OF CONSTRAINTS                =  0
  TLIMO = TIME - LIMIT                         =   1000. 00
  JSTEP = STEP-ADJUSTMENT PARAMETER            =  1
  NCYCT = NUMBER OF CYCLES FOR TOL. CONVERGENCE=  2
  RISTP = REDUCTION/INCREMENT PARAMETER        =  0. 850000
  ABACC = ABSOLUTE COMPUTER ACCURACY           =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY           =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE       =   1. 70000e-35
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE       =   1. 00000e-15

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)          -0. 10000000e+01
      XOPTV( 2)           0. 20000000e+01
      XOPTV( 3)           0. 10000000e+01

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)           0. 10000000e+01
      SDOPT( 2)           0. 10000000e+01
      SDOPT( 3)           0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE    =  0. 513870
  FINIT = INITIAL VALUE OF OPT. CRITERION      =   1026. 00
  TIME  = ELAPSED TIME IN SEC                  =   1. 91667
  ITYPE = TYPE OF RESULT                       =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL  =  0.

OPTIMUM VECTOR:
---------------

      XOPTV( 1)          -0. 22439162e-05
      XOPTV( 2)           0. 90434056e-04
      XOPTV( 3)           0. 88489018e-04
```

STANDARD DEVIATIONS OF THE
FINAL VECTOR COMPONENTS:

SDOPT(1)	0.11154764e-03
SDOPT(2)	0.11154764e-03
SDOPT(3)	0.11154764e-03

=== test sample # 3 for EVOL ===

```
**************************  INPUT = FILE  **************************
TEST SAMPLE NO. 3
2, 7, 1000. 00,
1, 2, 0. 85,
1. 7E-37, 1. 0E-7, 1. 7E-35, 1. 0E-25,
0. 0
0. 0,
1. 0,
1. 0,
**************************  OUPUT-FILE  **************************

 TEST SAMPLE NO. 3
 ----------------

  NOPTV = NUMBER OF OPTIMIZATION VARIABLES     =  2
  NICON = NUMBER OF CONSTRAINTS                =  7
  TLIMO = TIME - LIMIT                         =   1000. 00
  JSTEP = STEP-ADJUSTMENT PARAMETER            =  1
  NCYCT = NUMBER OF CYCLES FOR TOL. CONVERGENCE=  2
  RISTP = REDUCTION/INCREMENT PARAMETER        =  0. 850000
  ABACC = ABSOLUTE COMPUTER ACCURACY           =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY           =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE       =   1. 70000e-35
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE       =   1. 00000e-25

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0.          e+00
      XOPTV( 2)            0.          e+00

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 10000000e+01
      SDOPT( 2)            0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE   =  0. 513870

  INITIAL CONSTRAINT NO.  1   =  0.
  INITIAL CONSTRAINT NO.  2   =  0.
  INITIAL CONSTRAINT NO.  3   =   4. 00000
  INITIAL CONSTRAINT NO.  4   =   4. 00000
  INITIAL CONSTRAINT NO.  5   =   41. 0000
  INITIAL CONSTRAINT NO.  6   =  0.
  INITIAL CONSTRAINT NO.  7   =  0.

  FINIT = INITIAL VALUE OF OPT. CRITERION     =  0.
  TIME  = ELAPSED TIME IN SEC                 =   3. 50000
  ITYPE = TYPE OF RESULT                      =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL  =  -44. 8579
```

OPTIMUM VECTOR:

XOPTV(1)	0.62928929e+01
XOPTV(2)	0.22928936e+01

FINAL	CONSTRAINT NO.	1	=	6.29289
FINAL	CONSTRAINT NO.	2	=	2.29289
FINAL	CONSTRAINT NO.	3	=	7.15256e-07
FINAL	CONSTRAINT NO.	4	=	3.80474
FINAL	CONSTRAINT NO.	5	=	0.
FINAL	CONSTRAINT NO.	6	=	6.29289
FINAL	CONSTRAINT NO.	7	=	6.29289

STANDARD DEVIATIONS OF THE
FINAL VECTOR COMPONENTS:

SDOPT(1)	0.62928927e-06
SDOPT(2)	0.22928937e-06

=== test sample # 4 for EVOL ===

```
**************************  INPUT - FILE  **************************
TEST SAMPLE NO. 4
4, 3, 1000. 00,
1, 2, 0. 85,
1. 7E-35, 1. 0E-7, 1. 7E-7, 1. 0E-15,
0. 0
0. 0,
0. 0,
0. 0,
1. 0,
1. 0,
1. 0,
1. 0,
**************************  OUPUT-FILE  ****************************

 TEST SAMPLE NO. 4
 -----------------

  NOPTV = NUMBER OF OPTIMIZATION VARIABLES     =  4
  NICON = NUMBER OF CONSTRAINTS                =  3
  TLIMO = TIME - LIMIT                         =   1000. 00
  JSTEP = STEP-ADJUSTMENT PARAMETER            =  1
  NCYCT = NUMBER OF CYCLES FOR TOL. CONVERGENCE=  2
  RISTP = REDUCTION/INCREMENT PARAMETER        =  0. 850000
  ABACC = ABSOLUTE COMPUTER ACCURACY           =   1. 70000e-35
  REACC = RELATIVE COMPUTER ACCURACY           =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE       =   1. 70000e-07
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE       =   1. 00000e-15

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0.          e+00
      XOPTV( 2)            0.          e+00
      XOPTV( 3)            0.          e+00
      XOPTV( 4)            0.          e+00

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)          0. 10000000e+01
      SDOPT( 2)          0. 10000000e+01
      SDOPT( 3)          0. 10000000e+01
      SDOPT( 4)          0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE    =  0. 513870

  INITIAL CONSTRAINT NO.  1   =   5. 00000
  INITIAL CONSTRAINT NO.  2   =   8. 00000
  INITIAL CONSTRAINT NO.  3   =   10. 0000

  FINIT = INITIAL VALUE OF OPT. CRITERION      =  0.
```

```
  TIME  = ELAPSED TIME IN SEC                        =   3.85000
  ITYPE = TYPE OF RESULT                             =   2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL        =  -41.0542

OPTIMUM VECTOR:
---------------

      XOPTV( 1)          -0.34392156e-01
      XOPTV( 2)           0.76766777e+00
      XOPTV( 3)           0.19029341e+01
      XOPTV( 4)          -0.85594648e+00

  FINAL  CONSTRAINT NO.  1   =  0.
  FINAL  CONSTRAINT NO.  2   =   1.09888
  FINAL  CONSTRAINT NO.  3   =   2.84340

STANDARD DEVIATIONS OF THE
--------------------------
FINAL VECTOR COMPONENTS:
------------------------

      SDOPT( 1)           0.63202066e-07
      SDOPT( 2)           0.76766781e-07
      SDOPT( 3)           0.19029341e-06
      SDOPT( 4)           0.85594650e-07
```

=== GRUP ===

=== test program for the (μ,λ) - strategy ===

==========

```
C       ( MU , LAMBDA ) - OPTIMIZATION
C       ==============================================================
C       BERKELEY, U.S.A., APRIL 84
C       ==============================================================
C       DECLARATIONS
C       ==============================================================
        COMMON /GRZ/LZ
        COMMON /TF/ TARRAY (2)
        COMMON /IO/JINP,JOUT
        EXTERNAL F*,G*,ELATIM,GAUNOR,UNIRAN  !! REPLACE * BY CORRESPONDING NO.  !!
        LOGICAL RECOM
        DIMENSION XOPTV(20), SDOPT(20), XAUXV(20)
        DIMENSION FHIST(200), XHIST(400), SHIST(400)
        INTEGER*2 TITLE(40)
        DATA JINP, JOUT /5,6/
        DATA NFTOT, NXTOT /200,400/
C       ==============================================================
C       INPUT / OUTPUT OF CONTROL PARAMETERS
C       ==============================================================
        READ (JINP,5000) (TITLE(I), I=1,40)
5000    FORMAT(40A2)
        WRITE(JOUT,6000) (TITLE(I), I=1,40)
6000    FORMAT(1X,40A2)
        READ (JINP,5010) RECOM
5010    FORMAT(L1)
        WRITE(JOUT,6010) RECOM
6010    FORMAT(2X,'RECOM = RECOMBINATION - OPTION',14X,'=',1X,L2)
        READ (JINP,*) NBASV,NPOLV,NOPTV,NICON,TLIMO
        PRINT * ,'NBASV = NUMBER OF BASIC VECTORS                =',NB
        PRINT * ,'NPOLV = NUMBER OF POOL VECTORS                 =',
        PRINT * ,'NOPTV = NUMBER OF OPTIMIZATION VARIABLES       =',
        PRINT * ,'NICON = NUMBER OF CONSTRAINTS
        PRINT * ,'TLIMO = TIME - LIMIT
C       ***************************************************
        IF ( 2*NBASV .GT. NFTOT ) STOP
        IF ( 2*NBASV*NOPTV .GT. NXTOT ) STOP
C       ***************************************************
        READ(JINP,*) ABACC,REACC,ACOBJ,RCOBJ
        PRINT * ,'ABACC = ABSOLUTE COMPUTER ACCURACY
        PRINT * ,'REACC = RELATIVE COMPUTER ACCURACY
        PRINT * ,'ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE
        PRINT * ,'RCOBJ = RELATIVE CONVERGENCE TOLERANCE
C       ==============================================================
C       INPUT AND OUTPUT INITIAL VECTOR
C       ==============================================================
        DO 5 I=1,NOPTV
        READ (JINP,*) XOPTV(I)
5       CONTINUE
        CALL SCRIPT (1,XOPTV,NOPTV,JOUT)
        DO 6 I=1,NOPTV
        READ(JINP,*) SDOPT(I)
6       CONTINUE
        CALL SCRIPT (2,SDOPT,NOPTV,JOUT)
C       ==============================================================
C       PRE-CALCULATIONS
C       ==============================================================
```

```
      COSW=1.
      RANDOM=RAND(1)
      PRINT * , 'RANDOM = INITIAL VALUE OF RANDOM SEQUENCE    =', RANDOM
      XNOPTV=FLOAT(NOPTV)
      RISTP= COSW/SQRT(XNOPTV)
      PRINT * , 'RISTP = REDUCTION/INCREMENT PARAMETER        =', RISTP
      IF ( NICON .EQ. 0 ) GOTO 8
      DO 7 J=1,NICON
      CONSTRAINT = G* ( J,NOPTV,XOPTV )
      PRINT * , 'INITIAL CONSTRAINT NO. ', J, ' =', CONSTRAINT
7     CONTINUE
8     FINIT= F* (NOPTV,XOPTV)
      PRINT * , 'FINIT = INITIAL VALUE OF OPT. CRITERION      =', FINIT
C     ================================================================
C     INVOKE OPTIMIZATION ROUTINE
C     ================================================================
      TIME=ELATIM(D)
      CALL GRUP (RECOM,NBASV,NPOLV,NOPTV,NICON,
     1           ITYPE,TLIMO,ABACC,REACC,ACOBJ,RCOBJ,RISTP,
     2           FOCUR,FOPTC,XOPTV,SDOPT,XAUXV,FHIST,XHIST,SHIST,
     3           F*,G*,ELATIM,UNIRAN,GAUNOR)
      TIME=ELATIM(D)-TIME
C     ================================================================
C     RESULTS
C     ================================================================
      PRINT * , 'TIME  = ELAPSED TIME IN SEC                  =', TIME
      PRINT * , 'ITYPE = TYPE OF RESULT                       =', ITYPE
      PRINT * , 'FOCUR = BEST VALUE OF OPT. CRIT.-CURRENT SET=', FOCUR
      PRINT * , 'FOPTC = BEST VALUE OF OPT. CRIT.- AT ALL     =', FOPTC
      CALL SCRIPT (3,XOPTV,NOPTV,JOUT)
      IF ( NICON .EQ. 0 ) GOTO 10
      DO 9 J=1,NICON
      CONSTRAINT = G* ( J,NOPTV,XOPTV )
      PRINT * , 'FINAL  CONSTRAINT NO. ', J, ' =', CONSTRAINT
9     CONTINUE
10    CALL SCRIPT (4,SDOPT,NOPTV,JOUT)
      NN=2*NBASV
      NNN=NN*NOPTV
      CALL SCRIPT (5,FHIST,NN  ,JOUT)
      CALL SCRIPT (6,XHIST,NNN ,JOUT)
      CALL SCRIPT (7,SHIST,NNN ,JOUT)
      STOP
      END
C     ================================================================
C     ================================================================
      SUBROUTINE GRUP(REKO,L,LL,N,M,LF,TM,EA,EB,EC,ED,SN,FA,
     1FB,XB,SM,X,FK,XK,SK,F,G,T,R,Z)
      LOGICAL REKO
      DIMENSION XB(1),SM(1),X(1),FK(1),XK(1),SK(1)
      COMMON/GRZ/LZ
      EXTERNAL R
      KK(RR)=(LA+IFIX(FLOAT(L)*RR))*N
      TN=TM+T(D)
      LZ=1
      IF(M)4,4,1
1     LF=-1
```

```
      FB=0.
      DO 3 J=1,M
      FG=G(J,N,XB)
      IF(FG)2,3,3
2     FB=FB-FG
3     CONTINUE
      IF(FB)4,4,5
4     LF=1
      FB=F(N,XB)
5     DO 6 I=1,N
      SK(I)=AMAX1(SM(I),ABS(XB(I))*EB,EA)
6     XK(I)=XB(I)
      FK(1)=FB
      KA=N
      KB=0
      DO 21 K=2,L
      SA=1.
7     DO 8 I=1,N
8     X(I)=XB(I)+Z(SM(I)*SA,R)
      IF(LF)9,9,12
9     FF=0.
      DO 11 J=1,M
      FG=G(J,N,X)
      IF(FG)10,11,11
10    FF=FF-FG
11    CONTINUE
      IF(FF)60,60,17
12    IF(M)16,16,13
13    DO 15 J=1,M
      IF(G(J,N,X))14,15,15
14    SA=SA*.5
      GOTO 7
15    CONTINUE
16    FF=F(N,X)
17    IF(FF-FB)18,19,19
18    FB=FF
      KB=K
19    DO 20 I=1,N
      KA=KA+1
      SK(KA)=AMAX1(SM(I)*SA,ABS(X(I))*EB,EA)
20    XK(KA)=X(I)
21    FK(K)=FF
      IF(KB)24,24,22
22    KB=(KB-1)*N
      DO 23 I=1,N
23    XB(I)=XK(KB+I)
24    LA=L
      LB=0
25    LC=LB
      LB=LA
      LA=LC
      LC=0
      LD=0
26    SA=EXP(Z(SN,R))
      IF(REKO)GOTO 28
      KI=KK(R(D))
```

```
      DO 27 I=1,N
      KI=KI+1
      SM(I)=SK(KI)*SA
27    X(I)=XK(KI)+Z(SM(I),R)
      GOTO 30
28    SA=SA*0.5
      DO 29 I=1,N
      SM(I)=(SK(KK(R(D))+I)+SK(KK(R(D))+I))*SA
29    X(I)=XK(KK(R(D))+I)+Z(SM(I),R)
30    IF(LF)31,31,34
31    FF=0.
      DO 33 J=1,M
      FG=G(J,N,X)
      IF(FG)32,33,33
32    FF=FF-FG
33    CONTINUE
      IF(FF)60,60,38
34    IF(M)37,37,35
35    DO 36 J=1,M
      IF(G(J,N,X))46,36,36
36    CONTINUE
37    FF=F(N,X)
38    LD=LD+1
      IF(LD-L)40,40,39
39    IF(FF-FS)41,41,46
40    KS=LB+LD
41    FK(KS)=FF
      KS=(KS-1)*N
      DO 42 I=1,N
      KS=KS+1
      SK(KS)=AMAX1(SM(I),ABS(X(I))*EB,EA)
42    XK(KS)=X(I)
      IF(LD-L)46,43,43
43    KS=LB+1
      FS=FK(KS)
      DO 45 K=2,L
      KA=LB+K
      FF=FK(KA)
      IF(FF-FS)45,45,44
44    FS=FF
      KS=KA
45    CONTINUE
46    LC=LC+1
      IF(LC-LL)26,47,47
47    IF(LD-L)26,48,48
48    KA=LB+1
      FA=FK(KA)
      FC=FA
      DO 50 K=2,L
      KB=LB+K
      FF=FK(KB)
      FC=FC+FF
      IF(FF-FA)49,50,50
49    FA=FF
      KA=KB
50    CONTINUE
```

```
      IF(FA-FB)51,51,53
51    FB=FA
      KB=(KA-1)*N
      DO 52 I=1,N
52    XB(I)=XK(KB+I)
53    IF(FS-FA-EC)55,55,54
54    IF((FS-FA)*FLOAT(L)/ED-ABS(FC))55,55,59
55    LF=ISIGN(2,LF)
56    KB=(KA-1)*N
      DO 57 I=1,N
57    X(I)=XK(KB+I)
58    RETURN
59    IF(T(D)-TN)25,56,56
60    DO 61 I=1,N
61    XB(I)=X(I)
      FB=F(N,XB)
      FA=FB
      LF=0
      GOTO 58
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION UNIRAN (D)
1     RH=RAND(0)
      IF (RH.EQ.FLOAT(0)) GO TO 1
      UNIRAN=RH
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION GAUNOR(S,R)
      COMMON/GRZ/LZ
      DATA ZP/6.28318531/
      GOTO(1,2),LZ
1     A=SQRT(-2.*ALOG(R(D)))
      B=ZP*R(D)
      GAUNOR=S*A*SIN(B)
      LZ=2
      RETURN
2     GAUNOR=S*A*COS(B)
      LZ=1
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION ELATIM (D)
      DIMENSION TARRAY (2)
C     ================================================================
C     ELATIM   YIELDS  ELAPSED TIME IN SECONDS
C     ================================================================
      ELATIM=ETIME (TARRAY)
      PRINT * , '          "TIME" =  ' , ELATIM
      RETURN
      END
C     ================================================================
C     ================================================================
```

```
      SUBROUTINE SCRIPT ( IPARM, VEKT,LVEKT,JOUT )
      DIMENSION VEKT ( LVEKT )
      GOTO ( 10 , 20 , 30 , 40 , 50 , 60 , 70 ), IPARM
10    WRITE (JOUT,100)
      DO 15 I=1,LVEKT
      WRITE (JOUT,1000) I,VEKT(I)
15    CONTINUE
      GOTO 99
20    WRITE (JOUT,200)
      DO 25 I=1,LVEKT
      WRITE (JOUT,2000) I,VEKT(I)
25    CONTINUE
      GOTO 99
30    WRITE (JOUT,300)
      DO 35 I=1,LVEKT
      WRITE (JOUT,3000) I,VEKT(I)
35    CONTINUE
      GOTO 99
40    WRITE (JOUT,400)
      DO 45 I=1,LVEKT
      WRITE (JOUT,4000) I,VEKT(I)
45    CONTINUE
      GOTO 99
50    WRITE (JOUT,500)
      DO 55 I=1,LVEKT
      WRITE (JOUT,5000) I,VEKT(I)
55    CONTINUE
      GOTO 99
60    WRITE (JOUT,600)
      DO 65 I=1,LVEKT
      WRITE (JOUT,6000) I,VEKT(I)
65    CONTINUE
      GOTO 99
70    WRITE (JOUT,700)
      DO 75 I=1,LVEKT
      WRITE (JOUT,7000) I,VEKT(I)
75    CONTINUE
99    RETURN
100   FORMAT (/,'INITIAL CONSTELLATION:',/,22('-'),/,
     *          'STARTING VECTOR:',/,16('-'),/)
200   FORMAT (/,'STANDARD DEVIATIONS OF THE',/,26('-'),/,
     *          'INITIAL VECTOR COMPONENTS:',/,26('-'),/)
300   FORMAT (/,'OPTIMUM VECTOR:',/,15('-'),/)
400   FORMAT (/,'STANDARD DEVIATIONS OF THE',/,26('-'),/,
     *          'FINAL VECTOR COMPONENTS:',/,24('-'),/)
500   FORMAT (/,'VALUES OF THE LAST TWO BASIC',/,28('-'),/,
     *          'VECTOR SETS ( HISTORY ):',/,24('-'),/)
600   FORMAT (/,'CORRESPONDING VARIABLES:',/,24('-'),/)
700   FORMAT (/,'CORRESPONDING DEVIATIONS:',/,25('-'),/)
1000  FORMAT (6X,'XOPTV(',I2,')',10X,E15.8)
2000  FORMAT (6X,'SDOPT(',I2,')',10X,E15.8)
3000  FORMAT (6X,'XOPTV(',I2,')',10X,E15.8)
4000  FORMAT (6X,'SDOPT(',I2,')',10X,E15.8)
5000  FORMAT (6X,'FHIST(',I3,')',10X,E15.8)
6000  FORMAT (6X,'XHIST(',I3,')',10X,E15.8)
7000  FORMAT (6X,'SHIST(',I3,')',10X,E15.8)
```

```
      END
C     ================================================================
      REAL FUNCTION F1 ( N,X )
      DIMENSION X ( N )
C     ================================================================
C     TEST-FUNCTION ACCORDING TO MATYAS ( 1965 )
C     ================================================================
      H1=X(1)*X(1)
      H2=X(2)*X(2)
      H12=X(1)*X(2)
      F1 = 0.26 * (H1+H2) - 0.48 * H12
      RETURN
      END
C     ================================================================
C     ================================================================
C     ================================================================
      REAL FUNCTION G1 (J,N,X)
      DIMENSION X(N)
      GOTO (1,2),J
1     G=0.
      RETURN
2     G=0.
      RETURN
      END
C     ================================================================
      REAL FUNCTION F2 ( N,X )
      DIMENSION X (N)
C     ================================================================
C     TEST-FUNCTION ACCORDING TO SCHWEFEL ( 1978 )
C     ================================================================
      SUM=0.0
      DO 5 I=1,N
      SUM=SUM + X  (I) ** 10
5     CONTINUE
      F2 = SUM
      RETURN
      END
C     ================================================================
C     ================================================================
C     ================================================================
      REAL FUNCTION G2 (J,N,X)
      DIMENSION X(N)
      GOTO (1,2),J
1     G=0.
      RETURN
2     G=0.
      RETURN
      END
C     ================================================================
      REAL FUNCTION F3 ( N,X )
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO UEING
C     ================================================================
      H1=X(1)*X(1)
      H2=X(2)*X(2)
```

```
      F3 = -H1 - H2
      RETURN
      END
C     ================================================================
      REAL FUNCTION G3 (J,N,X)
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO UEING ( 1967 ) , MODIFIED
C     ================================================================
      GOTO ( 1,2,3,4,5 ) , J
1     G3 = X(1)
      RETURN
2     G3 = X(2)
      RETURN
3     G3 = - X(1) + X(2) + 4.0
      RETURN
4     G3 = X(1)/3.0 - X(2) + 4.0
      RETURN
5     G3 = X(1)*X(1) + X(2)*X(2) - 10.*X(1) - 10.*X(2) + 41.0
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION F4 ( N,X )
      DIMENSION X (N)
C     ================================================================
C     TEST-FUNCTION ACCORDING TO ROSEN AND SUZUKI ( 1965 )
C     ================================================================
      H1=X(1)*X(1)
      H2=X(2)*X(2)
      H3=X(3)*X(3)
      H4=X(4)*X(4)
      F4 = H1+H2+2.*H3+H4 - 5.*X(1) - 5.*X(2) - 21.*X(3) + 7.*X(4)
      RETURN
      END
C     ================================================================
C     ================================================================
      REAL FUNCTION G4 (J,N,X)
      DIMENSION X ( N )
C     ================================================================
C     TESTPROBLEM ACCORDING TO ROSEN AND SUZUKI
C     ================================================================
      GOTO ( 1,2,3 ) , J
1     G4 = -2.*X(1)*X(1) - X(2)*X(2) - X(3)*X(3) - 2.*X(1) + X(4) + 5.
      RETURN
2     G4 = -X(1)*X(1) - X(2)*X(2) - X(3)*X(3) - X(4)*X(4)
     *     -   X(1)   +    X(2)   -    X(3)   +    X(4)   + 8.0
      RETURN
3     G4 = -X(1)*X(1) - 2.*X(2)*X(2) - X(3)*X(3) - 2.*X(4)*X(4)
     *     +   X(1)                                +     X(4)     + 10.
      RETURN
      END
C     ================================================================
```

=== test sample # 1 for GRUP ===

```
**************************  INPUT - FILE  ************************
TEST SAMPLE NO. 1
T
10, 100, 2, 0, 1000. 00,
1. 7E-37, 1. 0E-7, 1. 7E-15, 1. 0E-7,
15. 0
30. 0,
5. 0,
5. 0,
************************** OUTPUT - FILE  **************************

 TEST SAMPLE NO. 1
 -----------------

  RECOM = RECOMBINATION - OPTION                =  t
  NBASV = NUMBER OF BASIC VECTORS               =  10
  NPOLV = NUMBER OF POOL VECTORS                =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES      =  2
  NICON = NUMBER OF CONSTRAINTS                 =  0
  TLIMO = TIME - LIMIT                          =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY            =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY            =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE        =   1. 70000e-15
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE        =   1. 00000e-07

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0. 15000000e+02
      XOPTV( 2)            0. 30000000e+02

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 50000000e+01
      SDOPT( 2)            0. 50000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE     =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER.        =  0. 707107
  FINIT = INITIAL VALUE OF OPT. CRITERION       =   76. 5000
  TIME  = ELAPSED TIME IN SEC                   =   24. 6500
  ITYPE = TYPE OF RESULT                        =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL   =   1. 19798e-16
  FOCUR = BEST VALUE OF OPT.  CRITERION-LOCAL   =   1. 79280e-16

OPTIMUM VECTOR:
---------------

      XOPTV( 1)           -0. 52540983e-07
      XOPTV( 2)           -0. 41260776e-07

STANDARD DEVIATIONS OF THE
```

FINAL VECTOR COMPONENTS:

SDOPT(1)	0.15262837e-06
SDOPT(2)	0.24800039e-06

VALUES OF THE LAST TWO BASIC VECTOR SETS (HISTORY):

FHIST(1)	0.17927982e-15
FHIST(2)	0.13630492e-14
FHIST(3)	0.13664580e-14
FHIST(4)	0.42981078e-15
FHIST(5)	0.72375281e-15
FHIST(6)	0.28032958e-15
FHIST(7)	0.13282121e-14
FHIST(8)	0.67262124e-15
FHIST(9)	0.79457366e-15
FHIST(10)	0.40724950e-15
FHIST(11)	0.23680079e-14
FHIST(12)	0.19846907e-15
FHIST(13)	0.28337419e-14
FHIST(14)	0.10212547e-14
FHIST(15)	0.29767421e-14
FHIST(16)	0.46038083e-15
FHIST(17)	0.22264888e-14
FHIST(18)	0.13389101e-14
FHIST(19)	0.19437407e-14
FHIST(20)	0.16823836e-14

CORRESPONDING VARIABLES:

XHIST(1)	-0.66676876e-07
XHIST(2)	-0.55902206e-07
XHIST(3)	0.68479260e-07
XHIST(4)	0.13065640e-06
XHIST(5)	0.48224386e-07
XHIST(6)	-0.25567886e-07
XHIST(7)	0.75817695e-07
XHIST(8)	0.41652296e-07
XHIST(9)	0.55884811e-07
XHIST(10)	0.99769643e-07
XHIST(11)	-0.83456371e-07
XHIST(12)	-0.83955456e-07
XHIST(13)	0.49467022e-07
XHIST(14)	-0.23233184e-07
XHIST(15)	-0.13185098e-06
XHIST(16)	-0.12562090e-06
XHIST(17)	-0.13246014e-06
XHIST(18)	-0.10081081e-06
XHIST(19)	-0.77757676e-08
XHIST(20)	-0.46641542e-07

XHIST(21)	0.11796858e-06
XHIST(22)	0.24935405e-07
XHIST(23)	0.23925423e-07
XHIST(24)	-0.39661785e-08
XHIST(25)	0.27138304e-06
XHIST(26)	0.25255923e-06
XHIST(27)	0.21045246e-08
XHIST(28)	-0.60725121e-07
XHIST(29)	-0.87645944e-07
XHIST(30)	0.20647247e-07
XHIST(31)	0.50226490e-07
XHIST(32)	0.83746272e-07
XHIST(33)	-0.24059548e-06
XHIST(34)	-0.22148087e-06
XHIST(35)	0.18567232e-06
XHIST(36)	0.17845493e-06
XHIST(37)	-0.21259203e-06
XHIST(38)	-0.22434978e-06
XHIST(39)	-0.19329670e-06
XHIST(40)	-0.20914494e-06

CORRESPONDING DEVIATIONS:

SHIST(1)	0.37416675e-06
SHIST(2)	0.33419423e-06
SHIST(3)	0.27326439e-06
SHIST(4)	0.86065036e-07
SHIST(5)	0.16313980e-06
SHIST(6)	0.56216788e-06
SHIST(7)	0.55335261e-06
SHIST(8)	0.14390504e-06
SHIST(9)	0.77337617e-07
SHIST(10)	0.27208671e-06
SHIST(11)	0.22244707e-06
SHIST(12)	0.15020740e-06
SHIST(13)	0.13206065e-06
SHIST(14)	0.26514002e-06
SHIST(15)	0.14575002e-05
SHIST(16)	0.57274576e-06
SHIST(17)	0.12396376e-06
SHIST(18)	0.21669429e-06
SHIST(19)	0.92335412e-07
SHIST(20)	0.25301259e-06
SHIST(21)	0.85078725e-07
SHIST(22)	0.83062915e-07
SHIST(23)	0.38240844e-06
SHIST(24)	0.10496666e-06
SHIST(25)	0.16740698e-06
SHIST(26)	0.18553352e-06
SHIST(27)	0.10013137e-06
SHIST(28)	0.21563454e-06
SHIST(29)	0.35952940e-07
SHIST(30)	0.14876318e-06
SHIST(31)	0.14341863e-06
SHIST(32)	0.89901853e-07

```
SHIST( 33)          0.20486887e-06
SHIST( 34)          0.23201162e-06
SHIST( 35)          0.92197342e-06
SHIST( 36)          0.18010115e-06
SHIST( 37)          0.44778215e-07
SHIST( 38)          0.96965032e-07
SHIST( 39)          0.70063344e-07
SHIST( 40)          0.16779873e-06
```

=== test sample # 2 for GRUP ===

```
**************************  INPUT - FILE  **************************
TEST SAMPLE NO. 2
T
10, 100, 3, 0, 1000. 00,
1. 7E-37, 1. 0E-7, 1. 7E-35, 1. 0E-15,
-1. 0,
2. 0
1. 0,
1. 0,
1. 0,
1. 0,
**************************  OUTPUT - FILE  **************************

 TEST SAMPLE NO. 2
 ----------------

  RECOM = RECOMBINATION - OPTION                    =  t
  NBASV = NUMBER OF BASIC VECTORS                   =  10
  NPOLV = NUMBER OF POOL VECTORS                    =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES          =  3
  NICON = NUMBER OF CONSTRAINTS                     =  0
  TLIMO = TIME - LIMIT                              =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY                =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY                =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE            =   1. 70000e-35
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE            =   1. 00000e-15

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)          -0. 10000000e+01
      XOPTV( 2)           0. 20000000e+01
      XOPTV( 3)           0. 10000000e+01

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)           0. 10000000e+01
      SDOPT( 2)           0. 10000000e+01
      SDOPT( 3)           0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE         =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER             =  0. 577350
  FINIT = INITIAL VALUE OF OPT. CRITERION           =   1026. 00
  TIME  = ELAPSED TIME IN SEC                       =   13. 2000
  ITYPE = TYPE OF RESULT                            =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL       =  0.
  FOCUR = BEST VALUE OF OPT.  CRITERION-LOCAL       =  0.

OPTIMUM VECTOR:
---------------
```

XOPTV(1)	0.13932062e-04
XOPTV(2)	0.12443652e-04
XOPTV(3)	0.12579307e-03

STANDARD DEVIATIONS OF THE
FINAL VECTOR COMPONENTS:

SDOPT(1)	0.46523320e-03
SDOPT(2)	0.38848715e-03
SDOPT(3)	0.83406182e-03

VALUES OF THE LAST TWO BASIC
VECTOR SETS (HISTORY):

FHIST(1)	0. e+00
FHIST(2)	0.16592443e-37
FHIST(3)	0. e+00
FHIST(4)	0. e+00
FHIST(5)	0.25609483e-37
FHIST(6)	0.69352720e-38
FHIST(7)	0. e+00
FHIST(8)	0.30264687e-38
FHIST(9)	0.41674193e-37
FHIST(10)	0. e+00
FHIST(11)	0.46843949e-35
FHIST(12)	0.54821192e-35
FHIST(13)	0. e+00
FHIST(14)	0.60603474e-37
FHIST(15)	0.57748553e-35
FHIST(16)	0.44462480e-34
FHIST(17)	0.15135659e-34
FHIST(18)	0.62289146e-37
FHIST(19)	0.47837058e-34
FHIST(20)	0.43466704e-36

CORRESPONDING VARIABLES:

XHIST(1)	0.13932062e-04
XHIST(2)	0.12443652e-04
XHIST(3)	0.12579307e-03
XHIST(4)	-0.99333993e-04
XHIST(5)	0.16672128e-03
XHIST(6)	-0.11268884e-03
XHIST(7)	0.31396878e-04
XHIST(8)	0.56424673e-04
XHIST(9)	0.67551286e-04
XHIST(10)	-0.33154072e-04
XHIST(11)	0.12799095e-03
XHIST(12)	0.21640561e-04
XHIST(13)	0.15554461e-03
XHIST(14)	0.31191943e-04

XHIST(15)	-0.16743789e-03
XHIST(16)	0.10511528e-03
XHIST(17)	-0.15279402e-03
XHIST(18)	-0.12320795e-03
XHIST(19)	0.11760974e-03
XHIST(20)	0.77192548e-04
XHIST(21)	0.31151911e-04
XHIST(22)	-0.64085216e-04
XHIST(23)	0.12063665e-04
XHIST(24)	-0.14063506e-03
XHIST(25)	0.18280443e-03
XHIST(26)	0.41259831e-04
XHIST(27)	0.70345675e-04
XHIST(28)	-0.13776371e-03
XHIST(29)	-0.10736816e-03
XHIST(30)	-0.10556674e-04
XHIST(31)	0.13804920e-03
XHIST(32)	-0.27448390e-03
XHIST(33)	-0.27248971e-03
XHIST(34)	-0.12543490e-03
XHIST(35)	0.13916491e-03
XHIST(36)	-0.29777954e-03
XHIST(37)	0.13342759e-03
XHIST(38)	0.74882206e-04
XHIST(39)	0.12950534e-03
XHIST(40)	0.12185700e-03
XHIST(41)	0.18977970e-03
XHIST(42)	0.12005676e-03
XHIST(43)	0.29933266e-03
XHIST(44)	0.43952256e-04
XHIST(45)	-0.30875977e-04
XHIST(46)	-0.79156176e-04
XHIST(47)	0.36711266e-03
XHIST(48)	-0.13391487e-04
XHIST(49)	0.23300560e-03
XHIST(50)	0.32856790e-03
XHIST(51)	0.37609920e-04
XHIST(52)	-0.78335404e-04
XHIST(53)	0.44188382e-04
XHIST(54)	-0.19030107e-03
XHIST(55)	-0.36980811e-03
XHIST(56)	0.12568707e-03
XHIST(57)	0.13467303e-04
XHIST(58)	0.23110837e-03
XHIST(59)	0.73372299e-04
XHIST(60)	-0.48707137e-04

CORRESPONDING DEVIATIONS:

SHIST(1)	0.32119956e-03
SHIST(2)	0.21709198e-03
SHIST(3)	0.33716636e-03
SHIST(4)	0.27063868e-04
SHIST(5)	0.10323704e-03
SHIST(6)	0.18168738e-03

```
SHIST(  7)          0.20948536e-03
SHIST(  8)          0.20541575e-03
SHIST(  9)          0.16471888e-03
SHIST( 10)          0.81771213e-04
SHIST( 11)          0.28030030e-03
SHIST( 12)          0.77865925e-03
SHIST( 13)          0.21891666e-03
SHIST( 14)          0.11072608e-03
SHIST( 15)          0.29565766e-03
SHIST( 16)          0.93986338e-04
SHIST( 17)          0.86953027e-04
SHIST( 18)          0.83113096e-04
SHIST( 19)          0.17885536e-03
SHIST( 20)          0.18364764e-03
SHIST( 21)          0.38967319e-03
SHIST( 22)          0.37769237e-03
SHIST( 23)          0.19136375e-03
SHIST( 24)          0.45727511e-03
SHIST( 25)          0.12267371e-03
SHIST( 26)          0.27112506e-03
SHIST( 27)          0.52501808e-03
SHIST( 28)          0.10137582e-03
SHIST( 29)          0.83348728e-04
SHIST( 30)          0.22123873e-03
SHIST( 31)          0.10013265e-03
SHIST( 32)          0.71312475e-04
SHIST( 33)          0.24704618e-03
SHIST( 34)          0.34249283e-03
SHIST( 35)          0.16550790e-03
SHIST( 36)          0.89946965e-03
SHIST( 37)          0.46969293e-04
SHIST( 38)          0.56763086e-03
SHIST( 39)          0.95827185e-03
SHIST( 40)          0.30451256e-03
SHIST( 41)          0.95381227e-04
SHIST( 42)          0.61360892e-03
SHIST( 43)          0.11374384e-03
SHIST( 44)          0.37847180e-03
SHIST( 45)          0.19855618e-03
SHIST( 46)          0.39843741e-03
SHIST( 47)          0.25596441e-03
SHIST( 48)          0.51161833e-03
SHIST( 49)          0.11168580e-03
SHIST( 50)          0.11647214e-03
SHIST( 51)          0.25437100e-03
SHIST( 52)          0.32429988e-03
SHIST( 53)          0.23457981e-03
SHIST( 54)          0.14230114e-02
SHIST( 55)          0.55705983e-03
SHIST( 56)          0.25390298e-03
SHIST( 57)          0.83186640e-03
SHIST( 58)          0.89634719e-04
SHIST( 59)          0.24205883e-03
SHIST( 60)          0.49966463e-03
```

=== test sample # 3 for GRUP ===

```
************************** INPUT - FILE ******************************
TEST SAMPLE NO. 3
T
10, 100, 2, 5, 1000.00,
1.7E-37, 1.0E-7, 1.7E-35, 1.0E-25,
0.0
0.0,
1.0,
1.0,
************************** OUTPUT - FILE *****************************

 TEST SAMPLE NO. 3
 -----------------

  RECOM = RECOMBINATION - OPTION                =  t
  NBASV = NUMBER OF BASIC VECTORS               =  10
  NPOLV = NUMBER OF POOL VECTORS                =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES      =  2
  NICON = NUMBER OF CONSTRAINTS                 =  5
  TLIMO = TIME - LIMIT                          =   1000.00
  ABACC = ABSOLUTE COMPUTER ACCURACY            =   1.70000e-37
  REACC = RELATIVE COMPUTER ACCURACY            =   1.00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE        =   1.70000e-35
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE        =   1.00000e-25

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0.          e+00
      XOPTV( 2)            0.          e+00

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0.10000000e+01
      SDOPT( 2)            0.10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE     =  0.513870
  RISTP = REDUCTION/INCREMENT PARAMETER         =  0.707107

  INITIAL CONSTRAINT NO.  1   =  0.
  INITIAL CONSTRAINT NO.  2   =  0.
  INITIAL CONSTRAINT NO.  3   =   4.00000
  INITIAL CONSTRAINT NO.  4   =   4.00000
  INITIAL CONSTRAINT NO.  5   =   41.0000

  FINIT = INITIAL VALUE OF OPT. CRITERION       =  0.
  TIME  = ELAPSED TIME IN SEC                   =   19.0000
  ITYPE = TYPE OF RESULT                        =  2
  FOCUR = BEST VALUE OF OPT. CRIT.-CURRENT SET=  -25.9055
  FOPTC = BEST VALUE OF OPT. CRIT.- AT ALL      =  -25.9055
```

OPTIMUM VECTOR:

XOPTV(1)	0.20179145e+01
XOPTV(2)	0.46726379e+01

FINAL	CONSTRAINT NO.	1	=	2.01791
FINAL	CONSTRAINT NO.	2	=	4.67264
FINAL	CONSTRAINT NO.	3	=	6.65472
FINAL	CONSTRAINT NO.	4	=	2.38419e-07
FINAL	CONSTRAINT NO.	5	=	0.

STANDARD DEVIATIONS OF THE
FINAL VECTOR COMPONENTS:

SDOPT(1)	0.49516945e-06
SDOPT(2)	0.82091532e-06

VALUES OF THE LAST TWO BASIC
VECTOR SETS (HISTORY):

FHIST(1)	-0.25905525e+02
FHIST(2)	-0.25905525e+02
FHIST(3)	-0.25905525e+02
FHIST(4)	-0.25905525e+02
FHIST(5)	-0.25905525e+02
FHIST(6)	-0.25905525e+02
FHIST(7)	-0.25905525e+02
FHIST(8)	-0.25905525e+02
FHIST(9)	-0.25905525e+02
FHIST(10)	-0.25905525e+02
FHIST(11)	-0.25905525e+02
FHIST(12)	-0.25905525e+02
FHIST(13)	-0.25905525e+02
FHIST(14)	-0.25905525e+02
FHIST(15)	-0.25905525e+02
FHIST(16)	-0.25905525e+02
FHIST(17)	-0.25905525e+02
FHIST(18)	-0.25905525e+02
FHIST(19)	-0.25905523e+02
FHIST(20)	-0.25905523e+02

CORRESPONDING VARIABLES:

XHIST(1)	0.20179145e+01
XHIST(2)	0.46726379e+01
XHIST(3)	0.20179145e+01
XHIST(4)	0.46726379e+01
XHIST(5)	0.20179145e+01
XHIST(6)	0.46726379e+01
XHIST(7)	0.20179145e+01

XHIST(8)	0.46726379e+01
XHIST(9)	0.20179145e+01
XHIST(10)	0.46726379e+01
XHIST(11)	0.20179145e+01
XHIST(12)	0.46726379e+01
XHIST(13)	0.20179145e+01
XHIST(14)	0.46726379e+01
XHIST(15)	0.20179145e+01
XHIST(16)	0.46726379e+01
XHIST(17)	0.20179145e+01
XHIST(18)	0.46726379e+01
XHIST(19)	0.20179145e+01
XHIST(20)	0.46726379e+01
XHIST(21)	0.20179145e+01
XHIST(22)	0.46726379e+01
XHIST(23)	0.20179145e+01
XHIST(24)	0.46726379e+01
XHIST(25)	0.20179145e+01
XHIST(26)	0.46726379e+01
XHIST(27)	0.20179145e+01
XHIST(28)	0.46726379e+01
XHIST(29)	0.20179145e+01
XHIST(30)	0.46726379e+01
XHIST(31)	0.20179145e+01
XHIST(32)	0.46726379e+01
XHIST(33)	0.20179145e+01
XHIST(34)	0.46726379e+01
XHIST(35)	0.20179145e+01
XHIST(36)	0.46726379e+01
XHIST(37)	0.20179141e+01
XHIST(38)	0.46726379e+01
XHIST(39)	0.20179143e+01
XHIST(40)	0.46726379e+01

CORRESPONDING DEVIATIONS:

SHIST(1)	0.39161182e-06
SHIST(2)	0.56492962e-06
SHIST(3)	0.20179145e-06
SHIST(4)	0.46726379e-06
SHIST(5)	0.37152392e-06
SHIST(6)	0.74690377e-06
SHIST(7)	0.32885936e-06
SHIST(8)	0.74202524e-06
SHIST(9)	0.20179145e-06
SHIST(10)	0.46726379e-06
SHIST(11)	0.20179145e-06
SHIST(12)	0.46726379e-06
SHIST(13)	0.20179145e-06
SHIST(14)	0.46726379e-06
SHIST(15)	0.51663682e-06
SHIST(16)	0.55523174e-06
SHIST(17)	0.27517299e-06
SHIST(18)	0.54216764e-06
SHIST(19)	0.20671450e-06

```
SHIST( 20)          0.49137077e-06
SHIST( 21)          0.20822776e-06
SHIST( 22)          0.63523260e-06
SHIST( 23)          0.35777211e-06
SHIST( 24)          0.51109618e-06
SHIST( 25)          0.20179145e-06
SHIST( 26)          0.46726379e-06
SHIST( 27)          0.22005223e-06
SHIST( 28)          0.46726379e-06
SHIST( 29)          0.20179145e-06
SHIST( 30)          0.46726379e-06
SHIST( 31)          0.46322538e-06
SHIST( 32)          0.10726343e-05
SHIST( 33)          0.24479931e-06
SHIST( 34)          0.46726379e-06
SHIST( 35)          0.28923935e-06
SHIST( 36)          0.46726379e-06
SHIST( 37)          0.51179512e-06
SHIST( 38)          0.46726379e-06
SHIST( 39)          0.44602743e-06
SHIST( 40)          0.46726379e-06
```

=== test sample # 3.1 for GRUP ===
=== restart from sample # 3 ===

```
**************************  INPUT - FILE  ***************************
TEST SAMPLE NO. 3. 1
T
10, 100, 2, 5, 1000. 00,
1. 7E-37, 1. 0E-7, 1. 7E-35, 1. 0E-25,
0. 0
0. 0,
10. 0,
10. 0,
************************** OUTPUT - FILE  **************************

 TEST SAMPLE NO. 3. 1
 --------------------

  RECOM = RECOMBINATION - OPTION                    =  t
  NBASV = NUMBER OF BASIC VECTORS                   =  10
  NPOLV = NUMBER OF POOL VECTORS                    =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES          =  2
  NICON = NUMBER OF CONSTRAINTS                     =  5
  TLIMO = TIME - LIMIT                              =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY                =   1. 70000e-37
  REACC = RELATIVE COMPUTER ACCURACY                =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE            =   1. 70000e-35
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE            =   1. 00000e-25

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0.          e+00
      XOPTV( 2)            0.          e+00

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 10000000e+02
      SDOPT( 2)            0. 10000000e+02

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE    =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER        =  0. 707107

  INITIAL CONSTRAINT NO.  1   =  0.
  INITIAL CONSTRAINT NO.  2   =  0.
  INITIAL CONSTRAINT NO.  3   =   4. 00000
  INITIAL CONSTRAINT NO.  4   =   4. 00000
  INITIAL CONSTRAINT NO.  5   =   41. 0000

  FINIT = INITIAL VALUE OF OPT. CRITERION      =  0.
  TIME  = ELAPSED TIME IN SEC                  =   35. 1833
  ITYPE = TYPE OF RESULT                       =  2
  FOCUR = BEST VALUE OF OPT. CRIT. -CURRENT SET=  -44. 8579
  FOPTC = BEST VALUE OF OPT. CRIT. - AT ALL    =  -101, 927
```

OPTIMUM VECTOR:

XOPTV(1)	0.84524336e+01
XOPTV(2)	0.55211496e+01

FINAL	CONSTRAINT NO.	1	=	8.45243
FINAL	CONSTRAINT NO.	2	=	5.52115
FINAL	CONSTRAINT NO.	3	=	1.06872
FINAL	CONSTRAINT NO.	4	=	1.29633
FINAL	CONSTRAINT NO.	5	=	3.19089

STANDARD DEVIATIONS OF THE
FINAL VECTOR COMPONENTS:

SDOPT(1)	0.20424385e-06
SDOPT(2)	0.86143352e-07

VALUES OF THE LAST TWO BASIC
VECTOR SETS (HISTORY):

FHIST(1)	-0.44857864e+02
FHIST(2)	-0.44857864e+02
FHIST(3)	-0.44857864e+02
FHIST(4)	-0.44857861e+02
FHIST(5)	-0.44857861e+02
FHIST(6)	-0.44857861e+02
FHIST(7)	-0.44857861e+02
FHIST(8)	-0.44857864e+02
FHIST(9)	-0.44857864e+02
FHIST(10)	-0.44857861e+02
FHIST(11)	-0.44857864e+02
FHIST(12)	-0.44857864e+02
FHIST(13)	-0.44857864e+02
FHIST(14)	-0.44857864e+02
FHIST(15)	-0.44857864e+02
FHIST(16)	-0.44857864e+02
FHIST(17)	-0.44857864e+02
FHIST(18)	-0.44857864e+02
FHIST(19)	-0.44857864e+02
FHIST(20)	-0.44857864e+02

CORRESPONDING VARIABLES:

XHIST(1)	0.62928929e+01
XHIST(2)	0.22928936e+01
XHIST(3)	0.62928929e+01
XHIST(4)	0.22928936e+01
XHIST(5)	0.62928929e+01
XHIST(6)	0.22928936e+01
XHIST(7)	0.62928929e+01

XHIST(8)	0.22928929e+01
XHIST(9)	0.62928929e+01
XHIST(10)	0.22928929e+01
XHIST(11)	0.62928929e+01
XHIST(12)	0.22928932e+01
XHIST(13)	0.62928929e+01
XHIST(14)	0.22928932e+01
XHIST(15)	0.62928929e+01
XHIST(16)	0.22928936e+01
XHIST(17)	0.62928929e+01
XHIST(18)	0.22928936e+01
XHIST(19)	0.62928929e+01
XHIST(20)	0.22928932e+01
XHIST(21)	0.62928929e+01
XHIST(22)	0.22928936e+01
XHIST(23)	0.62928929e+01
XHIST(24)	0.22928936e+01
XHIST(25)	0.62928929e+01
XHIST(26)	0.22928936e+01
XHIST(27)	0.62928929e+01
XHIST(28)	0.22928936e+01
XHIST(29)	0.62928929e+01
XHIST(30)	0.22928936e+01
XHIST(31)	0.62928929e+01
XHIST(32)	0.22928936e+01
XHIST(33)	0.62928929e+01
XHIST(34)	0.22928936e+01
XHIST(35)	0.62928929e+01
XHIST(36)	0.22928936e+01
XHIST(37)	0.62928929e+01
XHIST(38)	0.22928936e+01
XHIST(39)	0.62928929e+01
XHIST(40)	0.22928936e+01

CORRESPONDING DEVIATIONS:

SHIST(1)	0.62928927e-06
SHIST(2)	0.22928937e-06
SHIST(3)	0.62928927e-06
SHIST(4)	0.22928937e-06
SHIST(5)	0.62928927e-06
SHIST(6)	0.22928937e-06
SHIST(7)	0.62928927e-06
SHIST(8)	0.22928930e-06
SHIST(9)	0.62928927e-06
SHIST(10)	0.22928930e-06
SHIST(11)	0.62928927e-06
SHIST(12)	0.22928931e-06
SHIST(13)	0.62928927e-06
SHIST(14)	0.22928931e-06
SHIST(15)	0.62928927e-06
SHIST(16)	0.22928937e-06
SHIST(17)	0.82757663e-06
SHIST(18)	0.30153780e-06
SHIST(19)	0.62928927e-06

```
SHIST( 20)          0.22928931e-06
SHIST( 21)          0.62928927e-06
SHIST( 22)          0.22928937e-06
SHIST( 23)          0.62928927e-06
SHIST( 24)          0.22928937e-06
SHIST( 25)          0.62928927e-06
SHIST( 26)          0.22928937e-06
SHIST( 27)          0.62928927e-06
SHIST( 28)          0.22928937e-06
SHIST( 29)          0.62928927e-06
SHIST( 30)          0.22928937e-06
SHIST( 31)          0.62928927e-06
SHIST( 32)          0.22928937e-06
SHIST( 33)          0.74543010e-06
SHIST( 34)          0.27160669e-06
SHIST( 35)          0.62928927e-06
SHIST( 36)          0.22928937e-06
SHIST( 37)          0.62928927e-06
SHIST( 38)          0.22928937e-06
SHIST( 39)          0.62928927e-06
SHIST( 40)          0.22928937e-06
```

=== test sample # 3.2 for GRUP ===
=== restart from sample # 3.1 ===

```
**************************  INPUT - FILE  **************************
TEST SAMPLE NO. 3. 2
T
10, 100, 2, 5, 1000. 00,
1. 7E-15, 1. 0E-7, 1. 7E-30, 1. 0E-25,
100. 0,
100. 0,
1. 0,
1. 0,
**************************  OUTPUT-FILE  **************************

 TEST SAMPLE NO. 3. 2
 --------------------

  RECOM = RECOMBINATION - OPTION                    =  t
  NBASV = NUMBER OF BASIC VECTORS                   =  10
  NPOLV = NUMBER OF POOL VECTORS                    =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES          =  2
  NICON = NUMBER OF CONSTRAINTS                     =  5
  TLIMO = TIME - LIMIT                              =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY                =   1. 70000e-15
  REACC = RELATIVE COMPUTER ACCURACY                =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE            =   1. 70000e-30
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE            =   1. 00000e-25

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0. 10000000e+03
      XOPTV( 2)            0. 10000000e+03

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 10000000e+01
      SDOPT( 2)            0. 10000000e+01
  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE   =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER       =  0. 707107

  INITIAL CONSTRAINT NO.  1  =   100. 000
  INITIAL CONSTRAINT NO.  2  =   100. 000
  INITIAL CONSTRAINT NO.  3  =   4. 00000
  INITIAL CONSTRAINT NO.  4  =  -62. 6667
  INITIAL CONSTRAINT NO.  5  =   18041. 0

  FINIT = INITIAL VALUE OF OPT. CRITERION     =  -20000. 0
  TIME  = ELAPSED TIME IN SEC                 =   5. 11667
  ITYPE = TYPE OF RESULT                      =  0
  FOCUR = BEST VALUE OF OPT. CRIT. -CURRENT SET=  -139. 382
  FOPTC = BEST VALUE OF OPT. CRIT. - AT ALL   =  -139. 382

OPTIMUM VECTOR:
```

XOPTV(1) 0.95885630e+01
XOPTV(2) 0.68877983e+01

FINAL CONSTRAINT NO. 1 = 9.58856
FINAL CONSTRAINT NO. 2 = 6.88780
FINAL CONSTRAINT NO. 3 = 1.29924
FINAL CONSTRAINT NO. 4 = 0.308389
FINAL CONSTRAINT NO. 5 = 15.6187

STANDARD DEVIATIONS OF THE

FINAL VECTOR COMPONENTS:

SDOPT(1) 0.42966986e+01
SDOPT(2) 0.25562286e+01

VALUES OF THE LAST TWO BASIC

VECTOR SETS (HISTORY):

FHIST(1) 0.68618164e+01
FHIST(2) 0.76382494e+00
FHIST(3) 0.69209871e+01
FHIST(4) 0.98889732e+01
FHIST(5) 0.66645555e+01
FHIST(6) 0.55805001e+01
FHIST(7) 0.84594955e+01
FHIST(8) 0.93096533e+01
FHIST(9) 0.99218140e+01
FHIST(10) 0.90019016e+01
FHIST(11) 0.74180145e+01
FHIST(12) 0.38998752e+01
FHIST(13) 0.28830090e+01
FHIST(14) 0.60893755e+01
FHIST(15) 0.78106909e+01
FHIST(16) 0.16628046e+01
FHIST(17) 0.82734251e+01
FHIST(18) 0.60241375e+01
FHIST(19) 0.73548870e+01
FHIST(20) 0.26052365e+01

CORRESPONDING VARIABLES:

XHIST(1) 0.20094868e+02
XHIST(2) 0.92330513e+01
XHIST(3) 0.13145738e+02
XHIST(4) 0.84937296e+01
XHIST(5) 0.16560005e+02
XHIST(6) 0.56390181e+01
XHIST(7) 0.14194315e+02
XHIST(8) 0.30534172e+00

XHIST(9)	0.11967516e+02
XHIST(10)	0.14653728e+02
XHIST(11)	0.20370750e+02
XHIST(12)	0.11466395e+02
XHIST(13)	0.19831617e+02
XHIST(14)	0.19070034e+02
XHIST(15)	0.25964479e+02
XHIST(16)	0.17234438e+02
XHIST(17)	0.26882721e+02
XHIST(18)	0.22487324e+02
XHIST(19)	0.21128099e+02
XHIST(20)	0.81261978e+01
XHIST(21)	0.13221402e+02
XHIST(22)	0.15825149e+02
XHIST(23)	0.15851079e+02
XHIST(24)	0.13183568e+02
XHIST(25)	0.10264261e+02
XHIST(26)	0.10304429e+02
XHIST(27)	0.14899837e+02
XHIST(28)	0.48104610e+01
XHIST(29)	0.23271782e+02
XHIST(30)	0.19567951e+02
XHIST(31)	0.94321280e+01
XHIST(32)	0.37693233e+01
XHIST(33)	0.18894159e+02
XHIST(34)	0.18571478e+02
XHIST(35)	0.20439495e+02
XHIST(36)	0.10415358e+02
XHIST(37)	0.14231995e+02
XHIST(38)	0.28771076e+01
XHIST(39)	0.13188107e+02
XHIST(40)	0.11001272e+02

CORRESPONDING DEVIATIONS:

SHIST(1)	0.22866812e+02
SHIST(2)	0.22530378e+02
SHIST(3)	0.13769862e+02
SHIST(4)	0.13132776e+02
SHIST(5)	0.20649583e+01
SHIST(6)	0.21319036e+01
SHIST(7)	0.17787144e+02
SHIST(8)	0.15932670e+02
SHIST(9)	0.95659655e+00
SHIST(10)	0.12414961e+02
SHIST(11)	0.24748602e+01
SHIST(12)	0.63296504e+01
SHIST(13)	0.20178658e+00
SHIST(14)	0.64311752e+01
SHIST(15)	0.11214477e+01
SHIST(16)	0.10441768e+02
SHIST(17)	0.43620224e+02
SHIST(18)	0.93430424e+01
SHIST(19)	0.34915471e+01
SHIST(20)	0.28184717e+01

```
SHIST( 21)          0.14421588e+02
SHIST( 22)          0.13171772e+02
SHIST( 23)          0.90818577e+01
SHIST( 24)          0.47603412e+01
SHIST( 25)          0.65677876e+01
SHIST( 26)          0.31704514e+01
SHIST( 27)          0.84204620e+00
SHIST( 28)          0.58181062e+01
SHIST( 29)          0.10012677e+02
SHIST( 30)          0.63251615e+01
SHIST( 31)          0.12298533e+02
SHIST( 32)          0.10547853e+02
SHIST( 33)          0.91606388e+01
SHIST( 34)          0.63879600e+01
SHIST( 35)          0.80181284e+01
SHIST( 36)          0.31671243e+01
SHIST( 37)          0.66900223e+02
SHIST( 38)          0.14176805e+02
SHIST( 39)          0.28466063e+01
SHIST( 40)          0.74523807e+01
```

=== test sample # 3.3 for GRUP ===
=== restart from sample 3.2 ===

```
**************************** INUT - FILE ****************************
TEST SAMPLE NO. 3. 3
T
10, 100, 2, 5, 1000. 00,
1. 7E-15, 1. 0E-7, 1. 7E-30, 1. 0E-25,
9. 58,
6. 89,
1. 0,
1. 0,
**************************** OUTPUT - FILE ****************************

 TEST SAMPLE NO. 3. 3

  RECOM = RECOMBINATION - OPTION                =  t
  NBASV = NUMBER OF BASIC VECTORS               =  10
  NPOLV = NUMBER OF POOL VECTORS                =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES      =  2
  NICON = NUMBER OF CONSTRAINTS                 =  5
  TLIMO = TIME - LIMIT                          =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY            =   1. 70000e-15
  REACC = RELATIVE COMPUTER ACCURACY            =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE        =   1. 70000e-30
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE        =   1. 00000e-25

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0. 95799999e+01
      XOPTV( 2)            0. 68899999e+01

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 10000000e+01
      SDOPT( 2)            0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE   =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER       =  0. 707107

  INITIAL CONSTRAINT NO.  1   =   9. 58000
  INITIAL CONSTRAINT NO.  2   =   6. 89000
  INITIAL CONSTRAINT NO.  3   =   1. 31000
  INITIAL CONSTRAINT NO.  4   =  0. 303334
  INITIAL CONSTRAINT NO.  5   =   15. 5485

  FINIT = INITIAL VALUE OF OPT. CRITERION     =  -139. 249
  TIME  = ELAPSED TIME IN SEC                 =   25. 2667
  ITYPE = TYPE OF RESULT                      =  2
  FOCUR = BEST VALUE OF OPT. CRIT. -CURRENT SET=  -208. 000
  FOPTC = BEST VALUE OF OPT. CRIT. - AT ALL   =  -208. 000

OPTIMUM VECTOR:
```

XOPTV(1) 0.12000000e+02
XOPTV(2) 0.80000000e+01

FINAL CONSTRAINT NO. 1 = 12.0000
FINAL CONSTRAINT NO. 2 = 8.00000
FINAL CONSTRAINT NO. 3 = 0.
FINAL CONSTRAINT NO. 4 = 0.
FINAL CONSTRAINT NO. 5 = 49.0000

STANDARD DEVIATIONS OF THE

FINAL VECTOR COMPONENTS:

SDOPT(1) 0.13134381e-05
SDOPT(2) 0.77600788e-06

VALUES OF THE LAST TWO BASIC

VECTOR SETS (HISTORY):

FHIST(1) -0.20800000e+03
FHIST(2) -0.20800000e+03
FHIST(3) -0.20800000e+03
FHIST(4) -0.20800000e+03
FHIST(5) -0.20800000e+03
FHIST(6) -0.20800000e+03
FHIST(7) -0.20800000e+03
FHIST(8) -0.20800000e+03
FHIST(9) -0.20800000e+03
FHIST(10) -0.20800000e+03
FHIST(11) -0.20800000e+03
FHIST(12) -0.20800000e+03
FHIST(13) -0.20800000e+03
FHIST(14) -0.20800000e+03
FHIST(15) -0.20800000e+03
FHIST(16) -0.20800000e+03
FHIST(17) -0.20799998e+03
FHIST(18) -0.20799998e+03
FHIST(19) -0.20800000e+03
FHIST(20) -0.20799998e+03

CORRESPONDING VARIABLES:

XHIST(1) 0.12000000e+02
XHIST(2) 0.80000000e+01
XHIST(3) 0.12000000e+02
XHIST(4) 0.80000000e+01
XHIST(5) 0.12000000e+02
XHIST(6) 0.80000000e+01
XHIST(7) 0.12000000e+02
XHIST(8) 0.80000000e+01

```
      XHIST(  9)            0.12000000e+02
      XHIST( 10)            0.80000000e+01
      XHIST( 11)            0.12000000e+02
      XHIST( 12)            0.80000000e+01
      XHIST( 13)            0.12000000e+02
      XHIST( 14)            0.80000000e+01
      XHIST( 15)            0.12000000e+02
      XHIST( 16)            0.80000000e+01
      XHIST( 17)            0.12000000e+02
      XHIST( 18)            0.80000000e+01
      XHIST( 19)            0.12000000e+02
      XHIST( 20)            0.80000000e+01
      XHIST( 21)            0.12000000e+02
      XHIST( 22)            0.80000000e+01
      XHIST( 23)            0.12000000e+02
      XHIST( 24)            0.80000000e+01
      XHIST( 25)            0.12000000e+02
      XHIST( 26)            0.80000000e+01
      XHIST( 27)            0.12000000e+02
      XHIST( 28)            0.80000000e+01
      XHIST( 29)            0.12000000e+02
      XHIST( 30)            0.80000000e+01
      XHIST( 31)            0.12000000e+02
      XHIST( 32)            0.80000000e+01
      XHIST( 33)            0.11999999e+02
      XHIST( 34)            0.79999995e+01
      XHIST( 35)            0.11999999e+02
      XHIST( 36)            0.79999995e+01
      XHIST( 37)            0.12000000e+02
      XHIST( 38)            0.80000000e+01
      XHIST( 39)            0.11999999e+02
      XHIST( 40)            0.79999995e+01

CORRESPONDING DEVIATIONS:
-------------------------

      SHIST(  1)            0.12000000e-05
      SHIST(  2)            0.80000001e-06
      SHIST(  3)            0.12000000e-05
      SHIST(  4)            0.80000001e-06
      SHIST(  5)            0.12000000e-05
      SHIST(  6)            0.80000001e-06
      SHIST(  7)            0.12000000e-05
      SHIST(  8)            0.80000001e-06
      SHIST(  9)            0.12000000e-05
      SHIST( 10)            0.80000001e-06
      SHIST( 11)            0.12000000e-05
      SHIST( 12)            0.80000001e-06
      SHIST( 13)            0.12000000e-05
      SHIST( 14)            0.80000001e-06
      SHIST( 15)            0.12000000e-05
      SHIST( 16)            0.80000001e-06
      SHIST( 17)            0.12000000e-05
      SHIST( 18)            0.80000001e-06
      SHIST( 19)            0.15180561e-05
      SHIST( 20)            0.10120374e-05
```

```
SHIST( 21)          0.15375596e-05
SHIST( 22)          0.10136092e-05
SHIST( 23)          0.12000000e-05
SHIST( 24)          0.80000001e-06
SHIST( 25)          0.12000000e-05
SHIST( 26)          0.80000001e-06
SHIST( 27)          0.15080921e-05
SHIST( 28)          0.10053947e-05
SHIST( 29)          0.12000000e-05
SHIST( 30)          0.80000001e-06
SHIST( 31)          0.12000000e-05
SHIST( 32)          0.80000001e-06
SHIST( 33)          0.11999999e-05
SHIST( 34)          0.79999995e-06
SHIST( 35)          0.11999999e-05
SHIST( 36)          0.79999995e-06
SHIST( 37)          0.12307664e-05
SHIST( 38)          0.81136119e-06
SHIST( 39)          0.11999999e-05
SHIST( 40)          0.79999995e-06
```

=== test sample # 4 for GRUP ===

```
**************************  INPUT - FILE  ************************
TEST SAMPLE NO. 4
T
10, 100, 4, 3, 1000. 00,
1. 7E-35, 1. 0E-7, 1. 7E-7, 1. 0E-15,
0. 0
0. 0,
0. 0,
0. 0,
1. 0,
1. 0,
1. 0,
1. 0,
************************** OUTPUT - FILE  **************************

 TEST SAMPLE NO. 4
 ----------------

  RECOM = RECOMBINATION - OPTION                  =  t
  NBASV = NUMBER OF BASIC VECTORS                 =  10
  NPOLV = NUMBER OF POOL VECTORS                  =  100
  NOPTV = NUMBER OF OPTIMIZATION VARIABLES        =  4
  NICON = NUMBER OF CONSTRAINTS                   =  3
  TLIMO = TIME - LIMIT                            =   1000. 00
  ABACC = ABSOLUTE COMPUTER ACCURACY              =   1. 70000e-35
  REACC = RELATIVE COMPUTER ACCURACY              =   1. 00000e-07
  ACOBJ = ABSOLUTE CONVERGENCE TOLERANCE          =   1. 70000e-07
  RCOBJ = RELATIVE CONVERGENCE TOLERANCE          =   1. 00000e-15

INITIAL CONSTELLATION:
----------------------
STARTING VECTOR:
----------------

      XOPTV( 1)            0.          e+00
      XOPTV( 2)            0.          e+00
      XOPTV( 3)            0.          e+00
      XOPTV( 4)            0.          e+00

STANDARD DEVIATIONS OF THE
--------------------------
INITIAL VECTOR COMPONENTS:
--------------------------

      SDOPT( 1)            0. 10000000e+01
      SDOPT( 2)            0. 10000000e+01
      SDOPT( 3)            0. 10000000e+01
      SDOPT( 4)            0. 10000000e+01

  RANDOM = INITIAL VALUE OF RANDOM SEQUENCE   =  0. 513870
  RISTP = REDUCTION/INCREMENT PARAMETER       =  0. 500000

  INITIAL CONSTRAINT NO.  1  =   5. 00000
  INITIAL CONSTRAINT NO.  2  =   8. 00000
  INITIAL CONSTRAINT NO.  3  =   10. 0000
```

```
  FINIT = INITIAL VALUE OF OPT. CRITERION      =  0.
  TIME  = ELAPSED TIME IN SEC                  =   164.333
  ITYPE = TYPE OF RESULT                       =  2
  FOPTC = BEST VALUE OF OPT. CRITERION-GLOBAL  =  -41.8676
  FOCUR = BEST VALUE OF OPT. CRITERION-LOCAL   =  -41.8676

OPTIMUM VECTOR:
---------------

      XOPTV( 1)          -0.13582385e+00
      XOPTV( 2)           0.59290773e+00
      XOPTV( 3)           0.19052869e+01
      XOPTV( 4)          -0.12530940e+01

  FINAL  CONSTRAINT NO.  1   =  0.
  FINAL  CONSTRAINT NO.  2   =   4.76837e-07
  FINAL  CONSTRAINT NO.  3   =   1.11895

STANDARD DEVIATIONS OF THE
--------------------------
FINAL VECTOR COMPONENTS:
------------------------

      SDOPT( 1)           0.50644263e-06
      SDOPT( 2)           0.72712299e-06
      SDOPT( 3)           0.19255525e-06
      SDOPT( 4)           0.14985424e-06

VALUES OF THE LAST TWO BASIC
----------------------------
VECTOR SETS ( HISTORY ):
------------------------

      FHIST(  1)          -0.41867634e+02
      FHIST(  2)          -0.41867634e+02
      FHIST(  3)          -0.41867634e+02
      FHIST(  4)          -0.41867634e+02
      FHIST(  5)          -0.41867634e+02
      FHIST(  6)          -0.41867634e+02
      FHIST(  7)          -0.41867634e+02
      FHIST(  8)          -0.41867634e+02
      FHIST(  9)          -0.41867634e+02
      FHIST( 10)          -0.41867634e+02
      FHIST( 11)          -0.41867634e+02
      FHIST( 12)          -0.41867634e+02
      FHIST( 13)          -0.41867634e+02
      FHIST( 14)          -0.41867634e+02
      FHIST( 15)          -0.41867634e+02
      FHIST( 16)          -0.41867634e+02
      FHIST( 17)          -0.41867630e+02
      FHIST( 18)          -0.41867630e+02
      FHIST( 19)          -0.41867630e+02
      FHIST( 20)          -0.41867630e+02

CORRESPONDING VARIABLES:
------------------------
```

```
XHIST(  1)          -0.13582437e+00
XHIST(  2)           0.59290838e+00
XHIST(  3)           0.19052867e+01
XHIST(  4)          -0.12530942e+01
XHIST(  5)          -0.13582464e+00
XHIST(  6)           0.59290797e+00
XHIST(  7)           0.19052870e+01
XHIST(  8)          -0.12530935e+01
XHIST(  9)          -0.13582467e+00
XHIST( 10)           0.59290725e+00
XHIST( 11)           0.19052868e+01
XHIST( 12)          -0.12530946e+01
XHIST( 13)          -0.13582350e+00
XHIST( 14)           0.59290802e+00
XHIST( 15)           0.19052868e+01
XHIST( 16)          -0.12530932e+01
XHIST( 17)          -0.13582382e+00
XHIST( 18)           0.59290886e+00
XHIST( 19)           0.19052868e+01
XHIST( 20)          -0.12530928e+01
XHIST( 21)          -0.13582473e+00
XHIST( 22)           0.59290832e+00
XHIST( 23)           0.19052868e+01
XHIST( 24)          -0.12530943e+01
XHIST( 25)          -0.13582432e+00
XHIST( 26)           0.59290862e+00
XHIST( 27)           0.19052867e+01
XHIST( 28)          -0.12530942e+01
XHIST( 29)          -0.13582286e+00
XHIST( 30)           0.59290814e+00
XHIST( 31)           0.19052868e+01
XHIST( 32)          -0.12530918e+01
XHIST( 33)          -0.13582358e+00
XHIST( 34)           0.59290731e+00
XHIST( 35)           0.19052873e+01
XHIST( 36)          -0.12530925e+01
XHIST( 37)          -0.13582444e+00
XHIST( 38)           0.59290832e+00
XHIST( 39)           0.19052868e+01
XHIST( 40)          -0.12530942e+01
XHIST( 41)          -0.13582450e+00
XHIST( 42)           0.59290904e+00
XHIST( 43)           0.19052870e+01
XHIST( 44)          -0.12530923e+01
XHIST( 45)          -0.13582402e+00
XHIST( 46)           0.59290832e+00
XHIST( 47)           0.19052864e+01
XHIST( 48)          -0.12530942e+01
XHIST( 49)          -0.13582411e+00
XHIST( 50)           0.59290808e+00
XHIST( 51)           0.19052867e+01
XHIST( 52)          -0.12530942e+01
XHIST( 53)          -0.13582391e+00
XHIST( 54)           0.59290731e+00
XHIST( 55)           0.19052868e+01
```

XHIST(56)	-0.12530942e+01
XHIST(57)	-0.13582426e+00
XHIST(58)	0.59290868e+00
XHIST(59)	0.19052868e+01
XHIST(60)	-0.12530931e+01
XHIST(61)	-0.13582414e+00
XHIST(62)	0.59290910e+00
XHIST(63)	0.19052868e+01
XHIST(64)	-0.12530930e+01
XHIST(65)	-0.13582465e+00
XHIST(66)	0.59290802e+00
XHIST(67)	0.19052873e+01
XHIST(68)	-0.12530924e+01
XHIST(69)	-0.13582373e+00
XHIST(70)	0.59290868e+00
XHIST(71)	0.19052869e+01
XHIST(72)	-0.12530922e+01
XHIST(73)	-0.13582323e+00
XHIST(74)	0.59290820e+00
XHIST(75)	0.19052867e+01
XHIST(76)	-0.12530924e+01
XHIST(77)	-0.13582437e+00
XHIST(78)	0.59290844e+00
XHIST(79)	0.19052870e+01
XHIST(80)	-0.12530926e+01

CORRESPONDING DEVIATIONS:

SHIST(1)	0.46187230e-06
SHIST(2)	0.94132922e-07
SHIST(3)	0.19052867e-06
SHIST(4)	0.16615201e-06
SHIST(5)	0.48644102e-06
SHIST(6)	0.28444029e-06
SHIST(7)	0.19052871e-06
SHIST(8)	0.18661275e-06
SHIST(9)	0.57834819e-06
SHIST(10)	0.10028856e-06
SHIST(11)	0.28627724e-06
SHIST(12)	0.17489279e-06
SHIST(13)	0.10709175e-05
SHIST(14)	0.37638731e-06
SHIST(15)	0.44841852e-06
SHIST(16)	0.48156852e-06
SHIST(17)	0.50259786e-06
SHIST(18)	0.22494862e-06
SHIST(19)	0.19052868e-06
SHIST(20)	0.17827963e-06
SHIST(21)	0.33279443e-06
SHIST(22)	0.97762232e-07
SHIST(23)	0.19052868e-06
SHIST(24)	0.13454860e-06
SHIST(25)	0.22234296e-06
SHIST(26)	0.96619473e-07
SHIST(27)	0.19052867e-06

SHIST(28)	0.12530943e-06
SHIST(29)	0.88724482e-06
SHIST(30)	0.69098098e-06
SHIST(31)	0.36762916e-06
SHIST(32)	0.27093967e-06
SHIST(33)	0.34980363e-06
SHIST(34)	0.16672351e-06
SHIST(35)	0.19052872e-06
SHIST(36)	0.24803231e-06
SHIST(37)	0.61030806e-06
SHIST(38)	0.19188047e-06
SHIST(39)	0.19504914e-06
SHIST(40)	0.23649231e-06
SHIST(41)	0.81810299e-06
SHIST(42)	0.35757790e-06
SHIST(43)	0.21556320e-06
SHIST(44)	0.21209364e-06
SHIST(45)	0.82096807e-06
SHIST(46)	0.95815415e-06
SHIST(47)	0.30261342e-06
SHIST(48)	0.36083847e-06
SHIST(49)	0.58117132e-06
SHIST(50)	0.22199315e-06
SHIST(51)	0.19052867e-06
SHIST(52)	0.12530943e-06
SHIST(53)	0.39789865e-06
SHIST(54)	0.75107543e-07
SHIST(55)	0.19052868e-06
SHIST(56)	0.13079031e-06
SHIST(57)	0.86476263e-06
SHIST(58)	0.35468292e-06
SHIST(59)	0.37154572e-06
SHIST(60)	0.73142974e-06
SHIST(61)	0.42105378e-06
SHIST(62)	0.39074715e-06
SHIST(63)	0.19052868e-06
SHIST(64)	0.17124472e-06
SHIST(65)	0.40597300e-06
SHIST(66)	0.17752534e-06
SHIST(67)	0.19052872e-06
SHIST(68)	0.12530924e-06
SHIST(69)	0.81706446e-06
SHIST(70)	0.13522147e-06
SHIST(71)	0.21558735e-06
SHIST(72)	0.24620289e-06
SHIST(73)	0.11139175e-05
SHIST(74)	0.48078641e-06
SHIST(75)	0.41853599e-06
SHIST(76)	0.31525593e-06
SHIST(77)	0.63879060e-06
SHIST(78)	0.10107816e-06
SHIST(79)	0.27994514e-06
SHIST(80)	0.15393638e-06