NASA TECHNICAL MEMORANDUM NASA TM-76217

DESCRIPTION OF STATISTICAL METHODS AND A ROUTINE FOR DETERMINING THE PARAMETERS OF A MODEL IN PROCESSING EXPERIMENTAL RESULTS

D. A. Usikov

(NASA-TM-76217) DESCRIPTION OF STATISTICAL METHODS AND A ROUTINE FOR DETERMINING THE PARAMETERS OF A MODEL IN PROCESSING EXPERIMENTAL RESULTS (National Aeronautics and Space Administration) 42 p N81-16814

G3/65 Unclas 41242

Translation of "Opisaniye statisticheskikh metodov i programmy dlya opredeleniya parametrov modeli pri obrabotke rezul'tatov eksperimentov," Academy of Sciences USSR, Institute of Space Research, Moscow, Report Pr-383, 1977, pp 1-42

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. JUNE 1980

STANDARD TITLE PAGE

1. Report No.
NASA TM-76217

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DESCRIPTION OF STATISTICAL METHODS AND A ROUTINE FOR DETERMINING THE PARAMETERS OF A MODEL IN PROCESSING EXPERIMENTAL RESULTS

5. Report Date

6. Performing Organization Code

7. Author(s)
D. A. Usikov
USSR Academy of Sciences, Institute of Space Research

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address
Leo Kanner Associates
Redwood City, California 94063

11. Contract or Grant No.
NASW-3199

13. Type of Report and Period Covered
Translation

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration, Washington, D.C. 20546

14. Sponsoring Agency Code

15. Supplementary Notes
Translation of "Opisaniye statisticheskikh metodov i programmy dlya opredeleniya parametrov modeli pri obrabotke rezul'tatov eksperimentov," Academy of Sciences USSR, Institute of Space Research, Moscow, Report Pr-383, 1977, pp 1-42

16. Abstract

A computer routine is suggested for selecting the optimum parameters of a theoretical model and determining the errors in them due to errors in physical measurements, and for evaluating the conformity of theory with the experiment. The paper describes the specification sequence for the input data and the format of the calculation results. Sample printouts are appended.

17. Key Words (Selected by Author(s))

18. Distribution Statement
Unclassified-Unlimited

19. Security Classif. (of this report)

20. Security Classif. (of this page)

21. No. of Pages

22. Price

CONTENTS

DESCRIPTION OF STATISTICAL METHODS AND A ROUTINE FOR DETERMINING THE PARAMETERS OF A MODEL IN PROCESSING EXPERIMENTAL RESULTS

D. A. Usikov

USSR Academy of Sciences, Institute of Space Research, Moscow

The selection of the optimum parameters of a theoretical model and /2* determination of the errors in them due to errors in physical measurements is an important stage in processing an experiment. Besides, in processing experimental data it becomes necessary to evaluate the conformity of theory with the experiment. The routine described in this work solves these problems. The user wishing to process a specific experiment with its help has only to write a subroutine for calculating the function of the specific model.

In compiling this routine attention was concentrated on assuring reliability, algorithmic speed and convenience. The routine extensively utilizes formatted printing and diagnosing possible errors in the input data. This paper describes in detail the specification sequence for the input data and the format of the calculation results. Necessary information on statistics is presented in a special chapter.

The programming language is FORTRAN, and the routine has been entered as a module in the routine library of the SOFI video display processing complex.

INTRODUCTION /3

The described routine is intended for processing experiments by the method of least squares or the maximum likelihood method. The class of selected functions is arbitrary, the required parameters may enter nonlinearly. To use the routine it is necessary to program the function calculation block in each specific case.

The result of the routine is a printout of a set of tables:

*Numbers in the margin indicate pagination in the foreign text.

1. The optimum values of the parameters of the model;
2. A covariance matrix of parameter errors;
3. The theoretical curve optimally describing the experiment;
4. The minimum statistical sum, i.e., the sum of the squares of the differences between the experimental and theoretical values of the functions calculcated for the best choice of parameters.

According to the chi-square criterion, the value of the minimum statistical sum makes it possible to select competing models as well as to determine the degree of correspondence of the model and the experiment.

Chapter I. ACCESS TO THE ROUTINE

Sec. 1. Data Input

The experiment is processed by the method of least squares. The sum of the errors (statistical sum) is minimized:

$$S(X) = \frac{1}{2} \sum_{i=1}^{M} \frac{\left(f_E(Z_i) - f_T(Z_i, X)\right)^2}{\sigma_i^2}, \qquad (1)$$

where M is the number of experiments; Z_i is the coordinate of the i-th measurement, for example, the instant, length, etc.; $f_E(Z_i)$ is the experimentally obtained values at points Z_i; $f_T(Z_i,x)$ is the theoretically predicted value at point Z_i; X is the vector of the /4 selected parameters of theory. We denote the dimension of vector X as N(N = dimX). σ_i is the error of the i-th experiment (one standard error).

The routine looks for the values of X at which the statistical sum (1) is minimal. The accuracy with which the minimum is sought is given by a special parameter related to the statistical nature of the problem. Determination of the parameter is described in Chapter II. It is assumed that the statistical sum (1) has one minimum. If there are several minima, a local minimum is determined, depending on the initial approximation of the parameters.

In the description of the quantities that follows, the notation employed corresponds to the identifiers in the routine. The quantities are entered in the same sequence as they are described below.

Quantities Entered Into the Main Routine:

N - dimension of the space of the X parameters. Format 16. Restriction, $N \leq 20$.
IPE - printout label. IPE = 0 - normal printout mode. IPE = 1 - test printout mode. Format 16.

Quantities Entered by ENEXPE Subroutine:

M - number of test points. Format 16. Restriction, $M \leq 400$.
Z(M) - array of M numbers - measurement coordinates. Format 5E16.7.
EXPE(M) - in notation of (1) - $f_E(Z_i)$ - array of experimental values. Format 5E16-7.
VEXPE(M) - in notation of (1) - σ_i - array of experimental errors. Format 5 E16.7.

Quantities Entered by SPOINT Subroutine:

X(N) - array of parameter input values. The closer the input parameters approach the optimum values the faster, in general, is the minimum found. Format 5E16.7. /5
AVD(N) - array of input values of parameter errors. These quantities are required by the routine for the initial selection of the gate circuit on which the first and second derivatives are calculated. The quantities are subsequently modified by the routine as the work proceeds. It is recommended to take AVD(i) ≈ 0.1X(i), i.e., take the errors at approximately 10% of the input values of the respective parameters.
Attention: AVD(1) should not be taken equal to zero! Format 5E16.7.
MARK(N) - qualifying array. If MARK(i) = 0 the given parameter varies. If MARK(i) = 1 the given parameter is reinforced and taken equal to the input value of X(i). Format 7211.

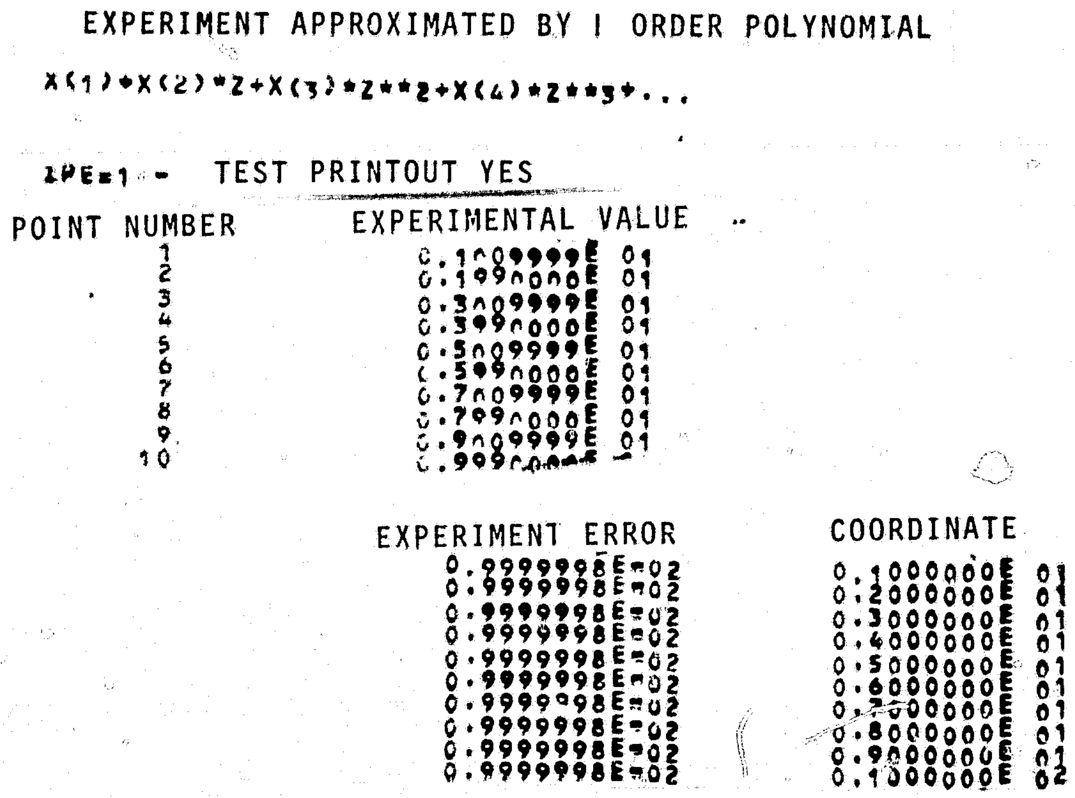
Quantities Entered Into the Main Routine:

EPS - accuracy of determination of the minimum. EPS is usually taken equal to 0.1. The precise definition of EPS is given in Chapter II. Format E16.7.

Sec. 2. Printout of Results

Below is presented the printout sequence of results for the case IPE = 0, i.e., when a test printout is not envisaged (see Sec. 1). We shall illustrate the routine printout with a concrete example.

The heading is printed indicating the model employed. This is followed by the input data: the EXPE(M) experiment array, the array of test errors VEXPE(M), the coordinates of the test points Z(M):



```
          EXPERIMENT APPROXIMATED BY I ORDER POLYNOMIAL

     X(1)+X(2)*Z+X(3)*Z**2+X(4)*Z**3+...


     IPE=1 -  TEST PRINTOUT YES
  POINT NUMBER        EXPERIMENTAL VALUE
             1             0.1009999E 01
             2             0.1990000E 01
             3             0.3009999E 01
             4             0.3990000E 01
             5             0.5009999E 01
             6             0.5990000E 01
             7             0.7009999E 01
             8             0.7990000E 01
             9             0.9009999E 01
            10             0.9990000E 01

                       EXPERIMENT ERROR            COORDINATE
                          0.9999998E-02            0.1000000E 01
                          0.9999998E-02            0.2000000E 01
                          0.9999998E-02            0.3000000E 01
                          0.9999998E-02            0.4000000E 01
                          0.9999998E-02            0.5000000E 01
                          0.9999998E-02            0.6000000E 01
                          0.9999998E-02            0.7000000E 01
                          0.9999998E-02            0.8000000E 01
                          0.9999998E-02            0.9000000E 01
                          0.9999998E-02            0.1000000E 02
```

This is followed by the input values of the parameters X(i) and the approximate errors of the parameters AVD(i):

```
  INPUT PARAMETER VALUE          APPROXIMATE PARAMETER ERROR
          0.0                          0.1000000E 01
          0.0                          0.1000000E 01
```

Some parameters may not vary (which is indicated by units in the MARK(i) array), and the routine therefore transfers to the internal list, which gives only the varied parameters. Correspondence between external and internal numeration is indicated in the following table:

```
DATA PRINTOUT EVERYWHERE ACCORDING TO INTERNAL NUMERATION
CORRESPONDENCE OF PARAMETERS
   INTERNAL PARAMETERS  EXTERNAL PARAMETERS
              1              1
              2              2
```

Next the EPS value is stated:

```
0.9999998E-02- ACCURACY OF DETERMINATION OF MINIMUM
-------------------------SCALE ONE CHI-SQUARE DISPERSION
```

With this the input data printout ends and the routine transfers to calculation:

/7

```
         1 -  THEORETICAL CURVE DEMONSTRATION
         2 -  EXPERIMENTAL VALUE
         3 -  COMPARISON WITH EXPERIMENT
              (THEORY MINUS EXPERIMENT DIVIDED BY TEST ERROR
-----------------------------------------------------------------
I  POINT  I          1          I      2        I      3         I
-----------------------------------------------------------------
I        1 I  0.0                I  0.1010E 01  I  -0.1010E 03   I
I        2 I  0.0                I  0.1990E 01  I  -0.1990E 03   I
I        3 I  0.0                I  0.3010E 01  I  -0.3010E 03   I
I        4 I  0.0                I  0.3990E 01  I  -0.3990E 03   I
I        5 I  0.0                I  0.5010E 01  I  -0.5010E 03   I
I        6 I  0.0                I  0.5990E 01  I  -0.5990E 03   I
I        7 I  0.0                I  0.7010E 01  I  -0.7010E 03   I
I        8 I  0.0                I  0.7990E 01  I  -0.7990E 03   I
I        9 I  0.0                I  0.9010E 01  I  -0.9010E 03   I
I       10 I  0.0                I  0.9990E 01  I  -0.9990E 03   I
-----------------------------------------------------------------
```

```
FIRST INPUT
   0.1924504E 07 - STATISTICAL SUM
PARAMETER    PARAMETER VALUE             LAST BASE
    1             0.0                     0.3012007E 00
    2             0.0                     0.4854292E-01
```

As the statistical sum decreases the program prints out intermediate results:

```
 INPUT FOLLOWING NEWTON METHOD STEP
     0.4849467E 01 - STATISTICAL SUM
 PARAMETER          PARAMETER VALUE           LAST BASE
     1              0.3365981E-02            0.4778767E-03
     2              0.9993422E 00            0.7764726E-04


     17 - NUMBER OF ACCESSES TO STATSUM BLOCK
```

When the optimum value of the parameters has been determined the table of theoretical and experimental values is printed out, after which the routine gives the minimum statistical sum, the Fisher matrices, and the covariance and correlation matrices:

```
          1 - THEORETICAL CURVE DEMONSTRATION
          2 - EXPERIMENTAL VALUE
          3 - COMPARISON WITH EXPERIMENT
              (THEORY MINUS EXPERIMENT DIVIDED BY TEST ERROR
-----------------------------------------------------------------
I POINT  I       1        I        2        I        3          I
-----------------------------------------------------------------
I      1 I  0.1003E 01    I   0.1010E 01    I  -0.7273E 00      I
I      2 I  0.2002E 01    I   0.1990E 01    I   0.1212E 01      I
I      3 I  0.3002E 01    I   0.3010E 01    I  -0.8486E 00      I
I      4 I  0.4001E 01    I   0.3990E 01    I   0.1091E 01      I
I      5 I  0.5000E 01    I   0.5010E 01    I  -0.9697E 00      I
I      6 I  0.6000E 01    I   0.5990E 01    I   0.9696E 00      I
I      7 I  0.6999E 01    I   0.7010E 01    I  -0.1091E 01      I
I      8 I  0.7998E 01    I   0.7990E 01    I   0.8485E 00      I
I      9 I  0.8998E 01    I   0.9010E 01    I  -0.1212E 01      I
I     10 I  0.9997E 01    I   0.9990E 01    I   0.7273E 00      I
-----------------------------------------------------------------

  MINIMUM SEARCH END
    0.4848243E 01 - STATISTICAL SUM
  PARAMETER   PARAMETER VALUE           LAST BASE
      1             0.3585961E-02        0.4778787E-03
      2             0.9993422E 00        0.7764726E-04

     24 -NUMBER OF ACCESSES TO STATSUM BLOCK

     FISHER MATRIX
  A   1   1=   0.9278250E 05
  A   1   2=   0.5100279E 06
  A   2   2=   0.3558694E 07

      COVARIANCE MATRIX
  B   1   1=   0.5079759E-04
  B   1   2=  -0.7280253E-05
  B   2   2=   0.1324398E-05

      CORRELATION MATRIX
  K   1   1=   0.1000000E 01
  K   1   2=  -0.8875962E 00
  K   2   2=   0.1000000E 01
```

/8

The routine cycle is complete. When the calculation ends control is transferred to the start, and the routine requires input of a new set of data.

Sec. 3. Function Calculation Subroutine

The function calculation subroutine FUN(i, Zi, N, X, FT) is the only subroutine which has to be changed as applied to the solution of a specific problem. The subroutine provides for calculating the function value for the given parameter values of model X(N) at point Z_i. For example, for a linear model the function is calculated according to the formula FT = X(1) + X(2)xZi + X(3)xZi↑2 + ... The FUN subroutine operates in two modes. The respective mode specifies the value of the identifier i. At i = 0, the subroutine does not perform calculations and only prints out the heading

```
EXPERIMENT APPROXIMATED BY 1 ORDER POLYNOMIAL
X(1)+X(2)*Z+X(3)*Z**2+X(4)*Z**3+...
```

At i = 1, calculation of the function takes place.

The formal parameter N in the subroutine access specifies the number of parameters of the model (varying or nonvarying in sum).

The parameters i, Zi, N, Z enter the FUN subroutine from the main routine, and the value of the function FT is transmitted back. /9

At points lying far from the minimum the search mode may generate values of the X parameters that are unusable in FUN subroutine calculations. To avoid such a situation a special message contingency is provided for. Before starting FUN subroutine calculations it is necessary to check whether the incoming values of the X parameter are permissible. If they have gone beyond the limit of permissible values the calculation is cancelled and FT is assigned the value 10^{18}. When the pilot routine receives PT equal to 10^{18} it takes steps to enter the domain of permissible values of the X parameters.

There are different ways of determining the permissibility of the X parameters. 1) X is verified prior to the calculation by means of a test consisting of a set of inequalities. 2) The permissibility of X is verified during the calculations. 3) A special FORTRAN device is employed: the possibility of transferring control to a specified place in the subroutine.

Sec. 4. Procedure for Recalling Subroutine from SOFI Library

The subroutines are cataloged in the object module library of the SOFI complex (1). The routine is generated in the following way:

```
// JOB   NAME
// PAUSE ASSGN SYSRLB,X'190' disk 102
// OPTION LINK
// EXEC FFORTRAN
      CALL MODNEU
      STOP
      END
      SUBROUTINE FUN(I,ZI,N,X,FT)                          /10
      DIMENSION  X(20)
      CALL FUN...(I,ZI,N,X,FT)
      RETURN
      END

      SUBROUTINE FUN...(I,ZI,N,X,FT)
     | subroutine body |
/*
// EXEC LNKEDT
// EXEC

     | input data |
```

The MODNEU subroutine contains the body of the main routine (texts of the routine in FORTRAN are given in the Appendix). In FUN are cataloged subroutines for calculating different functions. At present, the subroutines FUNPOL and FUNEXP, which calculate polynomials of the Nth order and the sum of exponents, respectively, have been cataloged. A special FUN subroutine is written for each of the calculated subroutines. The FUN subroutine transfers control from general access of the main MODNEU routine to the FUN function to the concrete FUN... subroutine.

If a calculation of the same type is performed for many variants according to some FUN... subroutine, an absolute model can be generated and put into the SOFI complex according to the procedures cited in work [1]. Thus, for example, at number 21000 in the SOFI complex there is located a routine with a FUNPOL subroutine, and at number 21001 in the absolute library of the SOFI complex is a routine with a FUNEXP subroutine. /11

Chapter II. ACCURACY OF DETERMINATION OF MODEL PARAMETERS

Sec. 5. Normal Limit of Likelihood Functions

In this chapter are described statistical methods of processing experiments involving normal approximations of likelihood functions. The domain of applicability of normal descriptions of an experiment is considered at the end of the chapter.

The likelihood function l(T/E) of a model T with respect to a given experiment E is determined according to Bayes equation:

$$\ell(T/E) = \frac{P(E/T)\,P(T)}{\int P(E/T)\,P(T)\,dT}. \qquad (2)$$

Here, P(E/T) is the probability density of the realization of the experiment E, provided the parameters of the model are T; P(T) is the a priori probability density of the parameters of the model. The likelihood function is sometimes called the "a posteriori" probability density of the model parameters, and P(T) is the "a priori" density.

It is assumed that for sufficiently representative experiments variations of the a priori density P(T) are insignificant in comparison with the peak of the likelihood function at the maximum point l(T/E). In these assumptions it is proved that, as the number of experiments increases, the function l(T/E) tends towards a normal distribution

[2, p. 217]. A multidimensional normal distribution has the form:

$$P(X) = \frac{|A|^{1/2}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(X-X_0)^T A(X-X_0)\right). \tag{3}$$

where $|A|$ is the determinant of A; $X^TAX = \sum_{ij} A_{ij}X_iX_j$; n = dimX is the dimension of the parameter space.

The matrix A is called the "information" matrix (or the Fisher /12 matrix) of the experiment; its inverse $B = \mathrm{cov}X = A^{-1}$ is the covariance matrix. Matrices A and B are symmetrical and positively determinate, i.e., $X^TAX > 0$ for all X.

The normal distribution is given by two parameters: the mean value

$$X_0 = \int XP(X)\,dX \tag{4}$$

and the covariance matrix

$$B = \mathrm{cov}X = \int (X - X_0)(X - X_0)^T P(X)\,dX. \tag{5}$$

It is thus assumed that the experiment is fully defined if X_0 and B are known. Calculation of X_0 and B according to equations (4) and (5) is inconvenient algorithmically, and differential methods are usually employed. The mean values of X_0 are usually found from the maximum condition of the likelihood function:

$$\frac{\partial \ln P(x)}{\partial x_i} = 0. \tag{6}$$

The Fisher matrix is obtained as the second derivative of the logarithm of the likelihood function:

$$A_{ij} = \frac{\partial^2 \ln P(x)}{\partial x_i \partial x_j}. \tag{7}$$

It is not hard to see that definitions (4), (5) and (6), (7) are equivalent. Determination of the parameters X_0 and B (or A) from equations (6) and (7) naturally involves numerical methods of looking for the maximum of the likelihood function. This relationship will be investigated in greater detail in Chapter III in describing the algorithm.

Description of an experiment by stating the parameters X_0 and B is called "normal" description of the experiment. Normal description forms a sufficient statistic for any linear combination of input parameters. Let us determine

$$Y = KX, \tag{8}$$

where X and Y are vectors, dimY = r, dimX = n, and K is the matrix /13 of the dimension coefficient rxn. If vector X is distributed normally with the parameters X_0, B, then Y is also distributed normally with the parameters:

$$Y_0 = KX_0, \quad \text{cov}Y = KBK^T. \tag{9}$$

The relationships (9) are also frequently employed in the case of a nonlinear relationship Y = f(X), the matrix K being determined as the factors of the linear term in a Taylor expansion of the function f(X) at point X_0:

$$K_{ij} = \frac{\partial y_i}{\partial x_j}\Big|_{X=X_0}. \tag{10}$$

Sec. 6. Distribution of the Statistical Sum. The Chi-Square Criterion

The most common case is when the errors of an experiment are distributed according to a normal law. The function P(E/T) (see equation (2)) has the form:

$$P(E/T) = c\,\exp(-\tfrac{1}{2}f_E - f_T(X))^T \textstyle\sum^{-1}(f_E - f_T(X)). \tag{11}$$

Here, c is the normalization constant; $\sum$ is the covariance matrix of the errors of the experiment;

$f_э = \begin{pmatrix} f_{э1} \\ \vdots \\ f_{эM} \end{pmatrix}$ is a vector compounded of experimentally measured values;

$$\begin{pmatrix} f_T^1(X) \\ \vdots \\ f_T^M(X) \end{pmatrix} = f_T(X)$$ is a vector compounded of the theoretical values of the function in the i-th experiment, when the values of the theoretical parameters are X;

M is the number of experimental points.

Note that the X parameters in the model $f_T(X)$ may enter nonlinearly. The function

$$S(X) = \frac{1}{2}\left(f_э - f_T(X)\right)^T \Sigma^{-1}\left(f_э - f_T(X)\right) \qquad (12)$$ /14

is called the "statistical sum" of the experiment. Function (12) does not differ essentially from the function (1) introduced before. Function (12) generalizes (1) for the case of dependent experiments.

Expanding S(X) in a Taylor series of X in the neighborhood of X_0 and neglecting terms higher than the second order of smallness, we arrive at the normal description of the experiment:

$$S(X) = S(X_0) + (X - X_0)^T \nabla S(X_0) + \tfrac{1}{2}(X - X_0)^T A(X_0)(X - X_o). \qquad (13)$$

Here, $\nabla S(X_0)$ is the gradient vector:

$$\nabla S = \begin{pmatrix} \frac{\partial S}{\partial X_1} \\ \vdots \\ \frac{\partial S}{\partial X_N} \end{pmatrix};$$

A is the matrix of second derivatives: $A_{ij} = \frac{\partial^2 S}{\partial x_i \partial x_j}$

It is assumed that S(X) has one extremum. Point X_0 is found from the condition

$$\nabla S(X_0) = 0. \qquad (14)$$

In particular, if the parameters enter the model linearly, then

$$f_T(X) = FX, \qquad (15)$$

where F is the dimension matrix dimf x dimX (M x N). In a linear model the expansion (13) is exact, and

$$A = F^T \Sigma^{-1} F. \tag{16}$$

The linearization procedure makes it possible to formulate a very convenient accuracy criterion for numerical methods of solving sets of equations. Suppose we have to solve a set of equations

$$F_i(X_1, X_2, \ldots, X_N) = 0; \quad i = 1, 2, \ldots, N, \tag{17}$$ /15

or in vector notation, $F(X)X = 0$.

Specific requirements are imposed on the accuracy of determination of the parameters, namely,

$$\sum_{i=1}^{N} \frac{(x_i - x_{io})^2}{\sigma_i^2} \le \varepsilon, \tag{18}$$

i.e., the approximate solutions of X should differ from the exact X by no more than a specified quantity characterized by the given error. More precisely, condition (18) is equivalent to the requirement that the deviations of the approximate values must lie within the ellipsoid given by equation (18). The parameter ε states the degree of approximation to the exact solution. If it is necessary to take into account the paired relationships of the accuracies, then condition (18) is replace by the condition

$$(X - X_0)^T \Sigma^{-1} (X - X_0), \tag{19}$$

where is Σ is a positively given definite matrix,

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_N \end{pmatrix}; \qquad X_{i0} = \begin{pmatrix} X_{10} \\ \vdots \\ X_{N0} \end{pmatrix}.$$

The solution of equation (17) is apparently equivalent to the solution of the problem of finding the minimum,

$$\min_X \sum_{i=1}^{N} F_i^2(X) = \min_X F^T F. \tag{21}$$

This procedure, which is extensively employed in numerical methods, makes it possible to solve the initial problem by developed methods of looking for the extremum of a function of many variables. However, if

$$F^T F \leq \varepsilon \tag{22}$$

is adopted as a condition for attaining the required accuracy, the values of X which satisfy (22) will not, generally speaking, satisfy /16 the inequality (19). Let us now show that, proceeding from the inequality (19), we can formulate an equivalent inequality in terms of F. To the accuracy of the linear term, we have

$$F(X) = F(X_0) + K(X - X_0), \tag{23}$$

where

$$K_{ij} = \left.\frac{\partial F_i}{\partial X_j}\right|_{X=X_0}$$

By definition, $F(X_0) = 0$, hence (23) involves only the linear term

$$F(X) = K(X - X_0). \tag{24}$$

If solution (17) is unique, then K can be inverted, and there exists a matrix K^{-1}. Substituting $X - X_0 = K^{-1}F$ into (19), we obtain

$$(K^{-1}F)^T \Sigma^{-1} K^{-1} F \leq \varepsilon$$

or

$$F^T (K \Sigma K^T)^{-1} F \leq \varepsilon \tag{25}$$

Matrix $K \Sigma K^T$ is positively definite, because Σ is positively definite. Matrix $(K \Sigma K^T)^{-1}$ is also positively definite. Finally, the solution of equation (17) is equivalent to the solution of the problem in finding the minimum:

$$\min_X F^T (K \Sigma K^T)^{-1} F, \tag{26}$$

and the condition of attaining the minimum $(X - X_0)^T \Sigma^{-1} (X - X_0)$ is equivalent to the condition $F^T (K \Sigma K^T)^{-1} F \leq \varepsilon$

Let us return to an examination of the properties of the statistical sum (12). Since it is impossible to find such an X_0 that (14) would vanish by numerical methods, it is necessary to formulate a criterion that would characterize the degree of approximation to the exact solution of the extremum problem. It is natural to require that the numerical method should be the more precise the /17 higher the accuracy of the experiment. In other words, the accuracy of the search for the extremum should be related to the covariance error matrix. We recall that X_o is the evaluation of the parameters of the model. The covariance matrix $B = A^{-1}$ expresses the probability of deviation of the true value of the parameters X_{true} from the value of X_0 obtained from the condition of the likelihood function maximum. The main statistical criteria are linked with the distribution instants of the statistical sum (12). It is not hard to show that the statistical sum possesses a chi-square distribution, provided $f(X)$ is linearly dependent on X. The mean value $\bar{S} = \frac{M-N}{2}$, the second central instant:

$$\sigma^2 = \int_0^\infty (S-\bar{S})^2 P(S)\,dS = \frac{M-N}{2}.$$

At M - N of the order of 10 or more, the chi-square distribution close to its maximum point can be considered close to normal. Therefore, if $\left|S(X_0) - \frac{M-N}{2}\right| \leq \sqrt{\frac{M-N}{2}}$, then the chi-square criterion is satisfied to a 68% probability. With the help of this criterion we can judge of the correspondence of theory and experiment, for example, correspondence of the experiment errors Σ^{-1} indicated by the experimenter to the true experiment errors. For example, if it seems certain that the experiment is correctly described theoretically, while as a result of the search for X_0 it has been found that $S(X_0) < \bar{S} - \sigma$, this can be interpreted as indicating that the experiment errors have been assumed too high.

According to the chi-square criterion, the solution of equation (14) is considered satisfactory if

$$\tfrac{1}{2}(X_0 - \hat{X})^T A^{-1} (X_0 - X) \leq \epsilon \bar{\alpha}. \qquad (27)$$

Here, $\hat{X}$ is the approximate value of X_0, ε is a certain constant expressing the degree of confidence. At $\varepsilon = 1$, as noted before, the confidence is 68%. At $\varepsilon > 1$, the confidence is accordingly higher. Usually ε is accepted equal to 0.1. /18

Since X_0 is not known, criterion (27) is not constructive but, taking into account that $S = A(X - X_0)$, we can obtain from (27) an easily computable criterion:

$$\tfrac{1}{2} \nabla S^T A^{-1} \nabla S \leq \varepsilon\sigma \qquad (28)$$

The gradient ∇S is computed at point X in the search for X_o analytically, if the explicit form $f_T(X)$ is known, but more often approximately. The matrix of second derivatives is also calculated in the course of the computations. Criterion (28) is especially convenient because calculation of ∇S and A are essential in the Newton method described further on.

Sec. 7. Condition for Ending the Minimization Process

The experimenter often does not know the absolute values of the experimental errors Σ^{-1}. All he knows is their relative course from experiment to experiment. Consequently, the quantity σ in (28) is indeterminate. In that case we can make use of the fact that $\sigma = \sqrt{S}$ and, dividing both parts of (28) by $\sqrt{\bar{S}}$, make use of the criterion

$$\tfrac{1}{2} \nabla S^T A^{-1} \nabla S \leq \varepsilon \sqrt{\bar{S}}. \qquad (29)$$

The unknown quantity $\hat{S}$ in (29) is substituted by the current statistical sum $S(\hat{X})$:

$$\tfrac{1}{2} \nabla S^T A-1 \nabla S \leq \varepsilon \sqrt{S}. \qquad (30)$$

Criterion (30) is, obviously, weaker than (29) at $S(\hat{X}) \gg \bar{S}$. However, in practice calculations show that in the overwhelming number of cases the quantity $\tfrac{1}{2} \nabla S^T A^{-1} \nabla S$ when it is far away from X_0 increases faster than $\sqrt{S}$, and criterion (30) does not lead to false stopping far from the extremum point. The criterion of stopping the minimization process (30) is employed in the program described further on.

Sec. 8. Statistical Description of Parameters of "Poorly Conditioned" Models and Experiments /19

There are at least two cases when normal description of an experiment is unsatisfactory. The first, which is frequently encountered in practice, is when it is impossible to reconstruct the parameters of the model from the experiment owing to low accuracy or inadequate statistic of the experiment. Such a situation arises, for example, in attempting to determine a great number of parameters of the model from a small number of experiments. The problem of describing the experiment that arises in this case will be discussed in more detail further on.

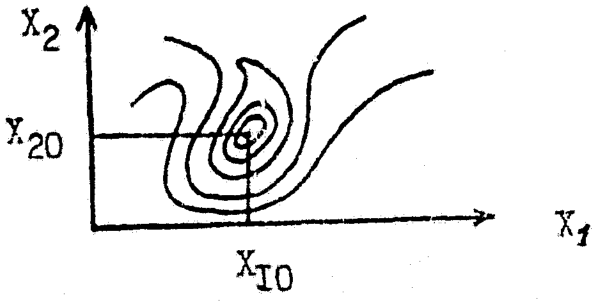
The second case, which is in a sense diametrically opposite to the first, is encountered in processing a great number of highly accurate experiments. In this case it is usually found that either the model does not adequately describe the experiment or that the experiment errors are stated imprecisely. For example, the correlation between individual measurements acquire major significance, or small systematic experimental errors become decisive, i.e., the "trifles" usually ignored, but which in rich statistics restrict the attainable accuracy of reconstruction of parameters. This case can be detected from the chi-square criterion. When the models of theory and experiment differ from the real-life experiment the minimum of the statistical sum differs substantially from the theoretical value, which can be a "trouble" indicator. The situation is described in detail in work [3].

We shall assume that the model is accurate and the errors of the experiment are given correctly. Let us consider the effect of the nonlinearity of function f(X). According to Bayes' approach [2], confidence that the theoretical parameters lie within the X domain is found from the formula

$$D(X) = \int_X l(T/E)\,dT, \qquad (31)$$ /20

where $l(T/E)$ is the likelihood function of the experiment (see equation (2)).

If the likelihood function is normal, then to calculate the integral (31) over any domain X, it is sufficient to know the normal distribution parameters: the vector of means and the covariance matrix. These parameters form a sufficient statistic. But if the likelihood function is not expressed by a normal distribution a situation arises which is conveniently illustrated with the help of curves of the levels of the likelihood function:



X_1 and X_2 are parameters of the model. Suppose that the likelihood function has only one maximum, and X_{10} and X_{20} are the parameters at which it attains that maximum. The curves show the solutions $l(X_1X_2)$ = const for various constants. The level lines always form ellipses in the neighborhood of the maximum, but farther away from the maximum point, when the model $f(X)$ is nonlinear, they are no longer ellipses. It is not hard to show that, for each confidence D there is such a d that in integrating over the X domain defined by the condition

$$l(X) \leq d, \qquad (32)$$

we obtain

$$D = \int_X l(X)\,dX.$$

Usually some meaningful confidence is assigned, for example, 0.68 or 0.99 /21

Let us denote a normal likelihood function with parameters X_0

and B as $l(X)$, The fundamental thesis of normal description of experiment is formulated as follows. If

$$\left| \int_X l(X)\,dX - \int_X \hat{l}(X)\,dX \right| \ll D, \qquad (33)$$

where D is a significant confidence, and the integration domain X is found according to (32), normal description of the experiment is assumed satisfactory.

In practice condition (33) can be obtained by the Monte Carlo method. The integral $\int_X l(X)\,dX$ in (33) can be determined according to the formula

$$\int_X l(X)\,dX = \int_X \frac{l(X)}{\hat{l}(X)} \hat{l}(X)\,dX. \qquad (34)$$

X_i is played according to the density of $\hat{l}(X)$ (an algorithm for modelling a normal distribution is described in work [11]). The quantities $\frac{lX_i}{\hat{l}X_i}$, which are an evaluation of the integral (34), are are introduced into the summator.

Chapter III. DESCRIPTION OF THE ALGORITHM

Sec. 9. The Modified Newton Method

The main task of the experiment processing algorithm is to find the minimum of the statistical sum $S(X)$ (12). The modified Newton method [4, 5, 6] used for this is based on local quadratic interpolation. Suppose the statistical sum depends upon the parameters in the following way:

$$S(X) = S_0 + (\nabla S)^T (X - X_0) + \tfrac{1}{2}(X - X_0)^T A (X - X_0). \qquad (35)$$

Here, X_0 is the only minimum point, and the gradient ∇S is taken at point X_0, i.e., it is zero:

$$S(X) - S_0 + \tfrac{1}{2}(X - X_0)^T A (X - X_0). \qquad (36)$$ /22

We take the gradient of both sides of (36):

$$\nabla S(X) = A(X - X_0). \qquad (37)$$

We multiply both sides of (37) by A^{-1} and obtain the formula of the Newton algorithm:

$$X_0 = X - A^{-1} \nabla S. \qquad (38)$$

If the representation S(X) (35) is exact, then the minimum point X_0 is found, starting from point X, in one step according to equation (38). Besides the Newton Method, the minimum can be found in a finite number of steps by means of the algorithm of the conjugated gradient method [7, 8] . The simple method of gradients, or quickest descent, does not generally speaking, converge at X_0 in a finite number of steps [4].

If the expansion (35)is not precise, we may find that the function at the new point after a step by the Newton method is greater than in the preceding point:

$$S(X - A^{-1} \nabla S) > S(X). \qquad (39)$$

To avoid such a situation and assure that the minimization process yields a monotonous decrease of function S(X), equation (38) is modified:

$$X_0 = X - \alpha A^{-1} \nabla S. \qquad (40)$$

If the simple Newton method fails to work on some step, α is selected in such a way that

$$S(X - \alpha A^{-1} \nabla S) \leq S(X). \qquad (41)$$

This device is known as the "modified" Newton method.

Sec. 10. Internal Scaling

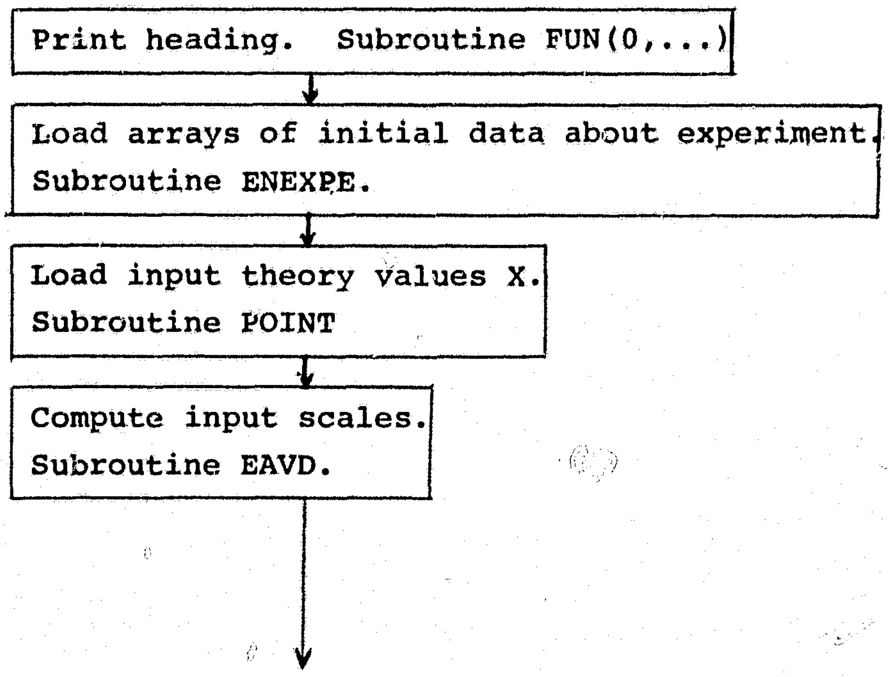
It is well known that a programmer, besides selecting a good algorithm, should take account of the specific features of computers. Algorithms are reliable and sufficiently universal only when account is taken of cases going beyond the machine's digital grid. One of the methods of avoiding overflow during calculation is described in Sec. 3. Here we shall examine scaling as related to the fact that the theory parameters X can be expressed in arbitrary physical units.

The scale adopted for each variable X_i is the evaluation of the error, or more precisely, the quantity:

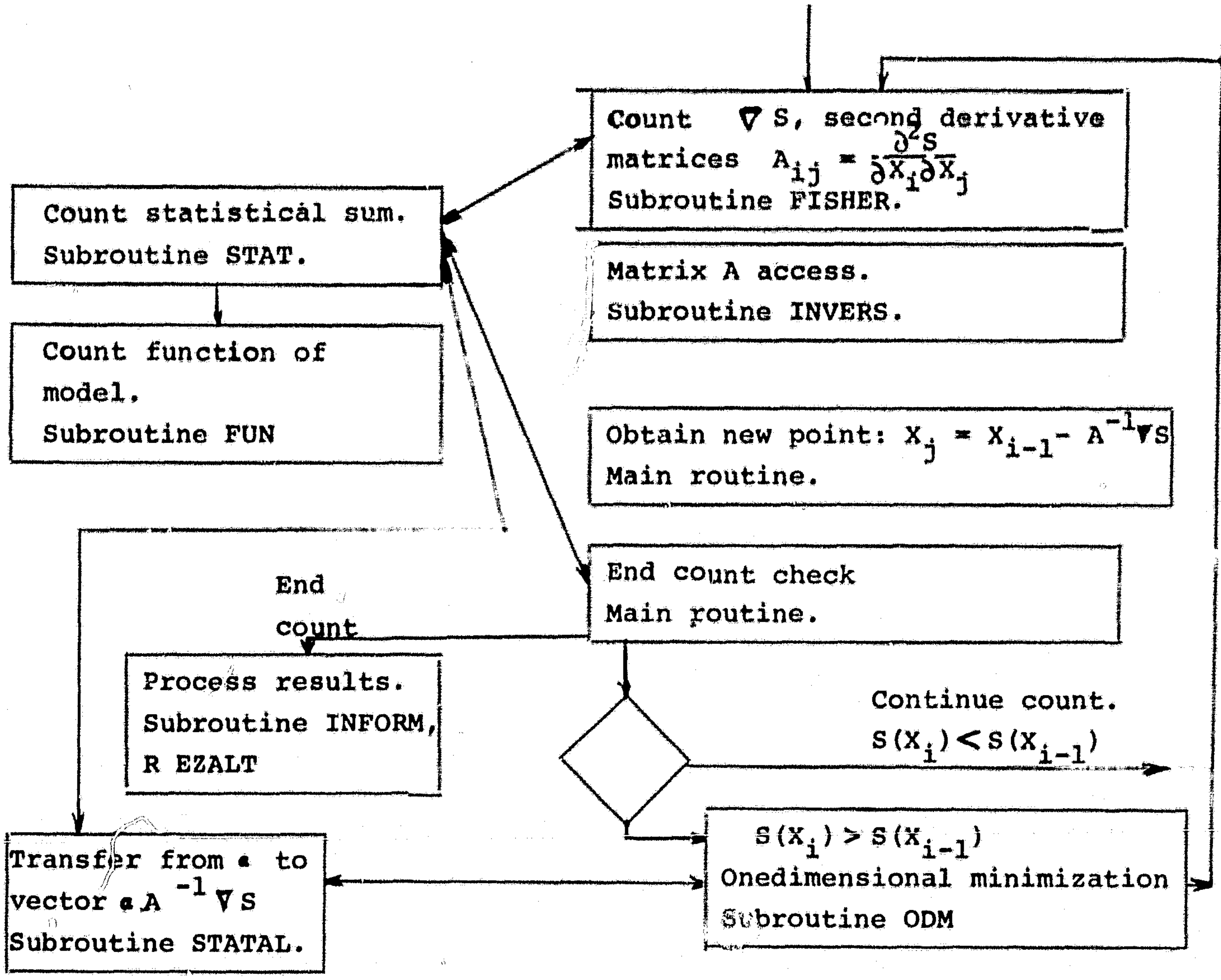
$$V_i = \frac{1}{\sqrt{\frac{\partial^2 S}{\partial X_i}}} \qquad (42)$$

This device makes it possible to make all the variables X_i dimensionless. The more precisely the quantity X_i is known the larger its value in the dimensionless form (42). The scales of $\sqrt{i}$ are verified in each step of the Newton method. The transfer to the true measurement units is carried out at the end of the calculation, that is, after the minimum has been found.

Sec. 11. Block Diagram of the Routine


Print heading. Subroutine FUN(0,...)
Load arrays of initial data about experiment.
Subroutine ENEXPE.
Load input theory values X.
Subroutine POINT
Compute input scales.
Subroutine EAVD.

/24


Count ∇S, second derivative matrices $A_{ij} = \frac{\partial^2 S}{\partial x_i \partial x_j}$ Subroutine FISHER.
Count statistical sum. Subroutine STAT.
Matrix A access. Subroutine INVERS.
Count function of model. Subroutine FUN
Obtain new point: $x_j = x_{i-1} - A^{-1}\nabla S$ Main routine.
End count
End count check Main routine.
Process results. Subroutine INFORM, R EZALT
Continue count. $S(x_i) < S(x_{i-1})$
Transfer from a to vector $a A^{-1} \nabla S$ Subroutine STATAL.
$S(x_i) > S(x_{i-1})$ Onedimensional minimization Subroutine ODM

The routine and subroutine texts are presented in the Appendix. The symmetric matrix inversion subroutine was taken from the collection of algorithms [9], algorithm N. 66b, and translated into FORTRAN.

Sec. 12. Algorithm of Approximate Calculation of Gradients and matrix of Second Derivatives

To reduce the user's preparatory work to the minimum, the routine is so devised that it does not require special programming of the first and second derivatives of the function with respect to the parameters of the model. These quantities are found in the FISHER subroutine by forming finite differences. /25

The following formula is used to compute the gradient:

$$\frac{\partial S}{\partial x_i} \approx \frac{S(x_i + \Delta x_i) - S(x_i - \Delta x_i)}{2\,\Delta x_i}.$$

The second derivatives are calculated this way:

$$(i \neq j) \quad \frac{\partial^2 S}{\partial x_i \partial x_j} = \frac{S(x_i + \Delta x_i, x_j + \Delta x_j) - S(x_i + \Delta x_i) - S(x_j + \Delta x_j) + S(x)}{\Delta x_i\ \Delta x_k};$$

$$(i = j) \quad \frac{\partial^2 S}{\partial x_i^2} = \frac{S(x_i + \Delta x_i) - 2S(x_i) + S(x_i - \Delta x_i)}{(\Delta x_i)^2}$$

The problem of selecting the base ΔX_i is resolved by taking as ΔX_i the evaluations of the parameter errors (42).

Sec. 13. The Algorithm of One-Dimensional Minimization

In passing from calculations by the Newton method to the modified method it is necessary to determine the minimum of function $S(X - \alpha A^{-1} \nabla S)$ as a function of the parameter α. The algorithm of the corresponding one-dimensional minimization is constructed as follows. By the time of reference to subroutine ODM the values of the function $S(\alpha)$ are known at two points α:

1) $S_2(\alpha_2 = 0) = S(X\quad)$;

2) $S_3(\alpha_3 = 1) = S(X\quad - A^{-1} \nabla S)$.

If $S_1(\alpha_1 = -1) < S(\alpha_2)$, a new point

$$\alpha = \alpha_1 - (\alpha_2 - \alpha_1)2 \tag{43}$$

is chosen. If $S(\alpha) < S_1(\alpha_1)$, the points are redesignated, which cahn be written using ALGOL notation:

$$\alpha_2 := \alpha_1\ ;\ \alpha_1 := \alpha\ ,$$

and the routine transfers to executing the operator (43).

This occurs until $S(\alpha) > S(\alpha_1)$. The redesignation occurs: /26

$$\alpha_3 := \alpha_2 \quad ; \quad \alpha_2 := \alpha_1 \; ; \quad \alpha_1 := \alpha \quad ;$$
$$S_3 := S_2 \quad ; \quad S_2 := S_1 \; ; \quad S_1 := S.$$

The aggregate of points α_1, α_2, α_3 such tat $S(\alpha_1) > S(\alpha_2)$, $S(\alpha_3) > S(\alpha_2)$, is called "canonical".

After obtaining the canonical triplet of numbers the algorithm transfers to looking for the extremum by the quadratic interpolation method according to three points. The minimum point α is determined from the formula

$$\alpha = \frac{a_1 b_1 - a_2 b_2 + a_3 b_3}{a_1 - a_2 + a_3},$$

where

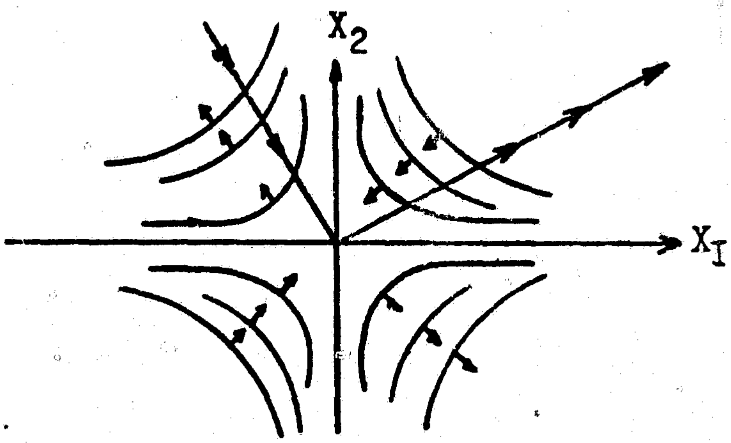
$$a_1 = S_1 (\alpha_2 - \alpha_3);$$
$$a_2 = S_2 (\alpha_1 - \alpha_3);$$
$$a_3 = S_3 (\alpha_1 - \alpha_2);$$
$$b_1 = (\alpha_2 + \alpha_3)/2;$$
$$b_2 = (\alpha_1 + \alpha_3)/2;$$
$$b_3 = (\alpha_1 + \alpha_2)/2.$$

It is not hard to show that point α always lies within the interval($\alpha_1 < \alpha < \alpha_3$) From the aggegate of four points α, α_1, α_2, α_3, S, S_1, S_2, S_3, three such points α'_1, α'_2, α'_3, S'_1, S'_2, S'_3 are selected so that the requirement $\alpha'_1 < \alpha'_2 < \alpha'_3$, $S'_2 < S'_1$, $S'_2 < S'_3$. The canonical triplet thus obtained is used to compute the next approximation of α according to the procedure described above. The condition for halting the one-dimensional minimization process is a one-dimensional case of the general criterion described in Sec. 7.

The described algorithm is highly effective. As numerous calculations show, the average number of selections of canonical triplets of points is about five. /27

Sec. 14. Drawbacks of the Algorithm of the Modified Newton Method. The Gradient Line Method.

Formula $X_0 = X - A^{-1}\nabla S$, on which the Newton method is based, is obtained from the condition $\nabla S(X_0) = 0$. However, the condition is satisfied not only by the minimum points, but by the saddle points as well. If in the iteration process point X approaches such a saddle point the direction $A^{-1}\nabla S$ will point to the saddle point. In some configurations of point X, going away from the saddle point with the help of the one-dimensional minimization process may require a lot of calculations. Let us explain this with the example of two variables, X_1 and X_2. Suppose X = 0 is a saddle point:



The lines of level S(X) = const are marked with arrows indicating the direction of increase of function S(X). If point X is located in the octants $(X_2 > 0;\ X_1 < 0)$ or $(X_2 < 0;\ X_1 > 0)$, the direction of one-dimensional minimization is towards point X = 0, and the X iteration will remain close to the saddle point as long as the next point does not move to octants $(X_1 > 0;\ X_2 < 0)$ or $(X_1 < 0;\ X_2 < 0)$ as a result of errors in the computation. The greater the number of variables the higher the probability of X approaching the domain of the saddle point. To get out of the saddle point the routine must carry out many iterations, which is precisely the principal /28 drawback of the modified Netwon method. Work [6] suggests ways of preventing X from occurring in the saddle point domain. The idea of the method is for matrix A, which is not positively determinate in the saddle point, to be replaced by a positively determinate matrix, for example, $A^* = (A^TA)^{\frac{1}{2}}$. Matrix A can also be reduced to diagonal form by similarity transformation, and all negative elements be substituted by positive ones. Work [6] presents the results of test computations which show that substantial acceleration of the operation of the algorithm can be achieved in this way. It is worth noting in passing that an algorithm in which the statistical sum is in the linear approximation (16) always has a positively definite matrix. This, apparently, explains the relatively high effectiveness of this method.

The second shortcoming of the modified Newton method is due to the need to invert the A matrix. If two parameters of a theory correlate strongly, or if the number of experiments is smaller than the number of parameters of the theory, inversion of the A matrix may prove difficult. In general, the following device can be employed: instead of the A matrix obtain an allied matrix of A*, i.e., a matrix made of algebraic complements A_{ik}, and compute the determinant A separately. The structure of the allied matrix is:

$$A^{\#}= \begin{Vmatrix} A_{II} & A_{2I} & \dots & A_{nI} \\ A_{I2} & A_{22} & \dots & A_{n2} \\ \dots & \dots & \dots & \dots \\ A_{In} & A_{2n} & \dots & A_{nn} \end{Vmatrix}$$

An allied matrix always exists. It is related to the inverse matrix by the formula $A^* = A^{-1}|A|$. The direction of $A^{-1}\nabla S$ will, apparently coincide with the direction of $A^*\nabla S$. /29

The two mentioned shortcomings can be overcome by further modifying the Newton method. In particular, in the following version of the described routine it is suggested to use a new algorithm of minimization along the gradient line, which is also based on local quadratic approximation of the function. The idea of the algorithm is to proceed from the given point X along the gradient line and thus find the one-dimensional minimum. Moving along the gradient line, unlike moving in the direction of the gradient, makes it possible to attain the minimum in a finite number of arithmetical operations. The path of motion is described by the equation:

$$\frac{\partial X}{\partial t} = \nabla S \tag{44}$$

for the initial condition $X(0) = X_{in}$. t is the parameter. Since $\nabla S = A(X - X_0)$, from (44) we obtain the equation

$$\frac{\partial X}{\partial t} = A(X - X_0), \tag{45}$$

which has the solution:

$$X(t) = X_{in} + (e^{At} - I)A^{-1}\nabla S(X_{in}). \tag{46}$$

At $t = 0$, $X(0) = X_{in}$. If the A matrix is positively definite, then $X(-\infty) = X_{in} - A^{-1}\nabla S$, which coincides with expression (38) for X_0 obtained by the Newton method. Thus, the parameter α used in one-dimensional minimization of the modified Newton method, $X(\alpha) = X_{in} - \alpha A^{-1}\nabla S(X_{in})$, is replaced by the matrix $e^{AT} - I$.

The algorithm based on equation (46) does not require inversion of the matrix A. Indeed, writing e^{AT} in the form of a Taylor series, we obtain from (46) /30

$$X(t) = X_{in} + tE(t)\nabla S(X_{in}); \tag{47}$$

the notation

$$E(t) = I + \frac{At}{2!} + \frac{A^2t^2}{3!} + \dots \tag{48}$$

is introduced. To compute the matrix E(t) we can use an algorithm based on application of the Cayeley-Hamilton theorem [10]. The A matrix is a root of its characteristic equation:

$$A^{n} + p_{1} A^{n-1} + \ldots p_{n} I = 0; \qquad (49)$$
$$P(t) = 1 + p_{1} t^{n-1} + \ldots p_{n} .$$

The coefficients p_i are obtained with the help of recurrent relationships:

$$p_I = - T_I ;$$
$$p_2 = - I/2 \; (p_I T_I + T_2) ; \qquad (50)$$
$$p_n = - 1/n \; (p_{n-1} \, T_I + p_{n-2} \, T_2 + \ldots + p_I \, T_{n-1} + T_n),$$

where $T_k = tr(A^k)$.

In (48) we limit ourselves to m terms of the expansion. Let us call the respective matrix $E_m(t)$. Since, according to (49), $P(t) = 0$, then

$$E_m \, t) = Q(t)P(t) + R(t) = R(t). \qquad (51)$$

The polynomial R(t) is found as the residue of the division of the polynomial $E_m(t)$ by the polynomial P(t). The algorithm makes it possible to find the exponential (48) to any degree of accuracy using exponents of the A matrix not higher than n.

Conclusion. The routine described in this work has been used to process experiments since 1972. Since that time various improvements have been made resulting in virtually flawless operation. Counting time under the routine is proportional to the cube of the number of varied parameters; it also strongly depends on the accuracy of the initial approximation of the parameters. The time of search for the statistical sum minimum for an essentially non-linear function f(X) comprising 5 parameters and 50 experimental points (of the sum of exponents type) with an EC 1040 machine is some 15 to 20 minutes if the initial approximation was poorly given (the statistical sum has to be reduced by a factor of more than 10,000), and two or three minutes if the statistical sum of the initial approximation differs from the minimum statistical sum by a factor of less than 100. If the parameters of the model enter linearly, the minimum is found in two iterations of the Newton method, and the counting time decreases considerably in comparison with nonlinear models.

In the near future it is planned to adjust a new version of the program, the idea of which is set forth in the last paragraph.

The author expresses his great appreciation to V. M. Dmitriyev for his support and critical remarks when the program was being elaboratred.

REFERENCES

1. Zolutukhin, V.G., B.I.Kolosov, D.A. Usikov, V.I. Borisenko, S.T. Mosin, V.N. Gorokohov. Opisaniye sistemy porgramm matematicheskoi obrabotki na EVM Yedinoi Serii (YeS) fotograficheskikh izobrazhenii Zemli, poluchennykh s kosmicheskikh apparatov. (Kompleks "SOFI") [Description of a System of Programs of Mathematical Computer Processing of the Unified Series (US) of Photographs of the Earth OBtained from Space Vehicles. (The "SOFI" Complex)] . Preprint. USSR Academy of Sciences, Institute of Space Research, 1977, Pr-336.

2. de Groot, M. Optimial'niye statisticheskiye resheniya [Optimum Statistical Solutions] . Mir, Moscow, 1974.

3. Elyasberg, P.Ye. Opredeleniye dvizheniya po resul'tatam izmerenii [Determination of Motion From Measurement Results] . Nauka, Moscow, 1976.

4. Vasilyev, F.P. Lektsii po metodam resheniya ekstremal'nykh zadach [Lectures on Methods of Solving Extremal Problems] . Moscow State University Press, Moscow, 1974.

5. Bakhvalov, N.S. Chislennyye metody [Numerical Methods].Vol I. Nauka, Moscow, 1973.

6. Fiacco, A and G. McCormic. Nelineinoye programmirovaniye [Nonlinear Programming] . Mir, Moscow, 1972.

7. Zangvill, U.I. Nelineinoye programmirovaniye [Nonlinear Programming]. Sovetskoye radio, Moscow, 1973.

8. Usikov, D.A. Statisticheskiye kriterii tochnosti chislennykh metodove resheniya sistem lineinykh uravnenii [Statistical Criteria of the Accuracy of Numerical Methods of Solving Sets of Linear Equations]. Preprint FEI-629, Obninsk, 1975.

9. Ageyev, M.I., V.P. Alik and Yu.I. Markov. Biblioteka algoritmov. 51b - 100b. [Library of Algorithms. 51b - 100b]. Sovetskoye radio, Moscow, 1976.

10. Mastascusa, E.J. A Method for Calculating e^{At} Based on the Cayley-Hamilton Theorem. JEEE 7/57, 1328-1329 (1969).

11. Sobol, I.M. Chislennyye metody Monte-Karlo Numerical Monte Carlo Methods] . Nauka, Moscow, 1973.

APPENDIX. Text of Routine in FORTRAN-IV.

/33

```
      SUBROUTINE MOnNEU
C     PROCESS EXPERIMENTS BY MODIFIED NEWTON METHOD
      DIMENSION X(20),X1(20),AVD(20),MARK(20),
     *GR(20),XH1(20),
     *Z(400),EXPE(400),VEXPE(400),
     *A(20,20),A1(20,20)
      COMMON/TM/ M,Z,EXPE,VEXPE
      COMMON/TN/ N,MARK;X
      COMMON/TN1/ N1,X1;AVD
      COMMON/HELP1/ XH1
      COMMON/A/ A
      COMMON/GR/ GR
      COMMON/IC/ ICOUN
99    READ(1,109) N
      IF(N.GT.0.AND.N.LT.20) GOTO 1
      WRITE(3,110)
      STOP
C     FIRST ACCESS PRINT FUNCTION HEADING
C     DO NOT COUNT FUNCTION VALUE
1     CALL FUN(0,ZI,N,X,FT)
      READ(1,109) IPE
      IF (IPE.EQ.0) WRITE(3,101)
      IF(IPE.EQ 1) WRITE(3,1C2)
C     LOAD INPUT EXPERIMENT DATA ARRAY
      CALL ENEXPE(M,Z,EQPE,VEXPE)
C     LOAD INITIAL PARAMETER APPROXIMATION ARRAYS
      CALL SPOINT(N,X,MARK,N1,X1,AVD,XH1)
      READ(1,106) EPS
      WRITE(3,107) EPS
      ICOUN=0
C     COUNT INITIAL SCALES AVD
      CALL STAT(1,S0)
      PR=EPS*S0/SQRT(FLOAT(M-N+1)/2.)
      CALL EAVD(S0,PR,N1,X1,AVD)
      MM=0
8     CALL INFORM(MM,S0;ICOUN,N1,XH1,AVD)
C     COUNT GRADIENTS AND SECOND DERIVATIVES MATRIX
      CALL FISHER(IPE,PR,S0)
      DO 31 I=1,N1
      DO 30 J=1,N1
30    A1(I,J)=A(I,J)
31    CONTINUE
C     ACCESS SYMMETRIC MATRIX A  DIMENSION                  N1*N1
      CALL INVERS(IPE,N1,A)
C     COUNT NEXT POINT BY NEWTON METHOD
      DIS1=0.
      DO 10 I=1,N1
      F=0.
      DO 9 K=1,N1
9     F=F+A(I,K)*GR(K)
      DIS1=DIS1+F*GR(I)
10    X1(I)=X1(I)-F
      DIS1=DIS1/2.
      I=0
C     COUNT STATISTICAL SUM AT PREDICTED POINT
21    CALL STAT(0,SS)
      IF(SS.LT.10.E18) GOTO 19
      DO 22 K=1,N1
22    X1(K)=X1(K)/100.
      I=I+1
      GOTO 21
      IF(I.EQ.0) GOTO 12
      IF(SS.GT.S0) GOTO 15
      MM=7
      GOTO 13
C     CHECK END OF MINIMUM SEARCH
12    R=S0-SS
      WRITE(3,108) DIS1,R
      IF(DIS1.GT.ABS(R)) R=DIS1
      R1=S0
      IF(SS.LT.S0) R1=SS
      IF(DIS1.GT.0..AND.ABS(R).LT.EPS*SQRT(R1)) GOTO 20
      IF(SS.GE.S0) GOTO 15
      MM=4
13    S0=SS
      PR=EPS*S0/SQRT(FLOAT(M-N+1)/2.)
      DO 14 K=1,N1
      XH1(K)=XH1(K)+X1(K)*AVD(K)
      AVD(K)=AVD(K)*SQRT(PR/ABS(A1(K,K)))
      X1(K)=0.
14    CONTINUE
C     TRANSFER TO NEXT COUNT STEP BY NEWTON METHOD
      GOTO 8
C     ONEDIMENSIONAL MINIMIZATION IN DIRECTION OF COVARIANCE MATRIX
15    DO 16 K=1,N1
16    GR(K)=X1(K)
      AL2=0.
      AL3=1
      S2=S.
      S3=SS
C     ONEDIMENSIONAL MINIMIZATION SUBROUTINE
      CALL ODM(IPE,EPS,S2,S3,AL2,AL3,AL,SS)
      MM=2
      GOTO 13
C     PRINT OPTIMUM THEORETICAL FUNCTION TABLE
20    CALL STAT(1,SS)
C     PRINT OPTIMUM PARAMETERS
      CALL INFORM(3,SS,ICOUN,N1,XH1,AVD)
C     PRINT  FISHER MATRICES,
C     COVARIANCE AND CORRELATION MATRICES,
C     ACCURACIES NORMALIZED TO STATISTICAL SUM VALUE AT
C     MINIMUM
      CALL REZULT(ICOUN,M,N1,SS,A,A1,X1,AVD,XH1)
101   FORMAT(1X///' IPE=0 - NO CHECK PRINTOUT
102   FORMAT(1X///' IPE=1 - YES CHECK PRINTOUT
103   FORMAT(1X,100X,'NBR =',I6)
105   FORMAT(19X,I4,I4)
106   FORMAT(E16.7)
107   FORMAT(1X,E16.7,'- ACCURACY OF DETERMINATION OF MINIMUM',
     *' TO SCALE OF ONE CHI-SQUARE DISPERSION')
108   FORMAT('+',E16.7,' - PREDICTED CHANGE OF STATSUM',
     *E16.7,' = REAL CHANGE OF STATSUM')
109   FORMAT(I6)
110   FORMAT(1X,' IMPERMISSIBLE NUMBER OF PARAMETERS')
      GOTO 99
      RETURN
      END
```

ENEXPE

```
         SUBROUTINE ENEXPE(M,Z,EXPE,VEXPE)
         DIMENSION Z(400),EXPE(400),VEXPE(400)
         READ(1,100) M
         READ(1,101) (Z(I),I=1,M)
         READ(1,101) (EXPE(I),I=1,M)
         READ(1,101) (VEXPE(I),I=1,M)
         WRITE(3,104)
         DO 6 I=1,M
         A=EXPE(I)
         B=VEXPE(I)
         IF(B.NE.0.) GOTO 5
         WRITE(3,105) I
         STOP
5        C=Z(I)
6        WRITE(3,106) I,A,B,C
100      FORMAT(I6)
101      FORMAT(5E16.7)
104      FORMAT(//1X,'POINT NUMBER',3X,'EXPERIMENTAL VALUE              ',3X,
        *'ERROR OF EXPERIMENT    ',6X,'COORDINATE  ')
105      FORMAT(1X,'B',I6,'ZERO ERROR ASSIGNED IN EXPERIMENT       ')
106      FORMAT(1X,I8,11X,E15.7,10X,E15.7,6X,E15.7)
         RETURN
         END
```

EAVD

```
      SUBROUTINE EAVD(S0,PR,N1,X1,AVD)
      DIMENSION X1(20),AVD(20)
      EPSU=10.E18
      C=100.
      DO 7 I=1,N1
2     X1(I)=1.
      CALL STAT(0,SPI)
      X1(I)=-1.
      CALL STAT(0,SN)
      B=(SPI-S0)+(SN-S0)
      IF(ABS(B).GT.PR) GOTO 6
      IF(ABS(AVD(I)).LT.EPSU) GOTO 5
      WRITE(3,108) I
      STOP
5     AVD(I)=AVD(I)*C
      GOTO 2
6     B=ABS(B)
      AVD(I)=AVD(I)*SQRT(PR/B)
7     X1(I)=0.
108   FORMAT(1X,'STATISTICAL SUM FOUND TO BE
     *'INDEPENDENT OF ',I5,'PARAMETER')
      RETURN
      END
```

SPOINT

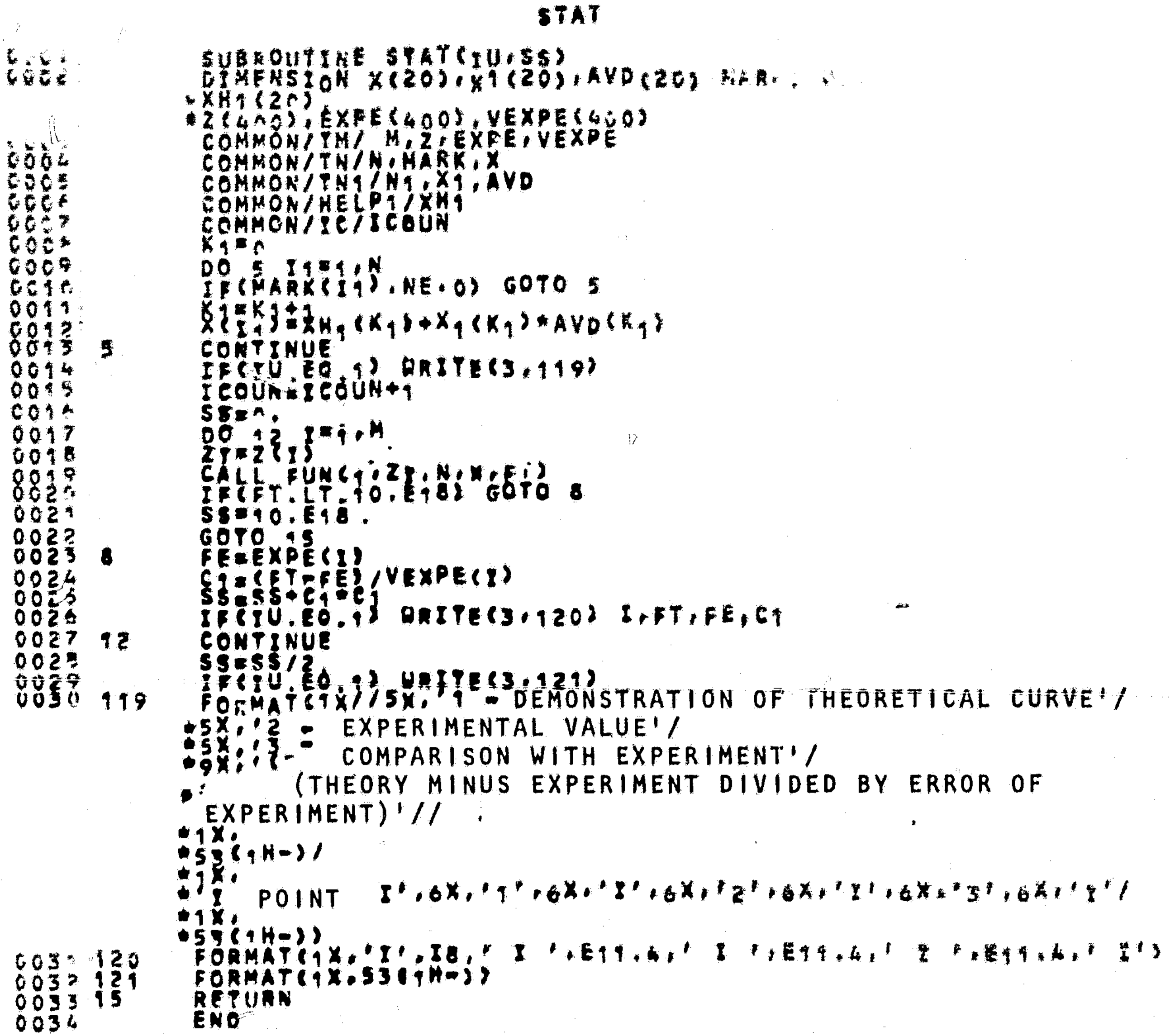
```
      SUBROUTINE SPOINT(N,X,MARK,N1,X1,AVD,XH1)
      DIMENSION X(20),AVD(20),MARK(20),XH1(20),X1(20)
      READ(1,4) (X(I),I=1,N)
      READ(1,4) (AVD(I), I=1,N)
      WRITE(3,6)
      DO 8 I=1,N
      A=X(I)
      B=AVD(I)
      IF(B.NE.0.) GOTO 7
      WRITE(3,10) I
      STOP
7     WRITE(3,9) I,A,B
8     CONTINUE
      READ(1,103) (MARK(I),I=1,N)
      WRITE(3,104)
      N1=0
      DO 17 I=1,N
      IF(MARK(I).EQ.1) GOTO 17
      N1=N1+1
      WRITE(3,105) N1,I
      X1(N1)=0.
      XH1(N1)=X(I)
      AVD(N1)=AVD(I)
17    CONTINUE
4     FORMAT(5E16,7)
6     FORMAT(1X//'PARAMETER',5X,'INITIAL VALUE OF
     *PARAMETER

     *5X,'APPROXIMATE ERROR OF PARAMETER')
9     FORMAT(11X,I5,E15.7,16X,E15.7)
10    FORMAT(1X//'APPROXIMATE ERROR',
     *I6,'OF PARAMETER STATED EQUAL TO ZERO')
103   FORMAT(72I1)
104   FORMAT(1X//'ALL FOLLOWING DATA PRINTOUT
     *'ACCORDING TO INTERNAL NUMERATION'/
     *1X,
     *'CORRESPONDENCE OF PARAMETERS'//
     *1X,
     *'INTERNAL PARAMETERS',4X,'EXTERNAL PARAMETERS')
105   FORMAT(10X,I8,2X,I8)
      RETURN
      END..
```

FISHER

```
      SUBROUTINE FISHER(IPE,PR,S0)
      DIMENSION X1(20),AVD(20),GR(20),SPB(20),A(20,20)
      COMMON/TN1/N1,X1,AVD
      COMMON/A/ A
      COMMON/GR/ GR
      DO 3 I=1,N1
2     X1(I)=-1.
      CALL STAT(0,SN)
      X1(I)=1.
      CALL STAT(0,SP)
      A(I,I)=(SP-S0)+(SN-S0)
      IF(ABS(A(I,I)).GT.PR/4.) GOTO 6
      AVD(I)=AVD(I)*2.
      GOTO 2
6     GR(I)=(SP-SN)/2.
      SPB(I)=SP
      IF(I.EQ.1) GOTO 1
      I1=I-1
      DO 4 K=1,I1
      X1(K)=1.
      CALL STAT(0,SS)
      A(I,K)=(SS-SPB(K))-(SPB(I)-S0)
      A(K,I)=A(I,K)
4     X1(K)=0.
1     X1(I)=0.
3     CONTINUE
      IF(IPE.EQ.0) GOTO 116
      WRITE(3,112) (AVD(I),I=1,N1)
      WRITE(3,113) (GR(I),I=1,N1)
112   FORMAT(/' AVD'/1X,(8E14.6))
113   FORMAT(/' GR'/1X,(8E14.6))
116   RETURN
      END
```

INVERS

```
      SUBROUTINE INVERS(IPE,N,A)
      DIMENSION A(20,20),V(20)
      N1=N-1
      IF(IPE.EQ.0) GOTO 3
      WRITE(3,115) ((A(I,J),I=1,N),J=1,N)
3     DO 9 K=1,N
      P=1./A(1,1)
      DO 4 I=2,N
4     V(I-1)=A(1,I)
      IF(N.EQ.1) GOTO 9
      DO 8 I=1,N1
      Y=-V(I)*P
      A(I,N)=Y
      DO 7 J=I,N1
7     A(I,J)=A(I+1,J+1)+V(J)*Y
8     CONTINUE
9     A(N,N)=-P
      DO 11 I=1,N
      DO 10 J=I,N
      A(I,J)=-A(I,J)
10    A(J,I)=A(I,J)
11    CONTINUE
      IF(IPE.EQ.0) GOTO 12
      WRITE(3,116) ((A(I,J),I=1,N),J=1,N)
115   FORMAT(1X,'A  BEFORE ACCESS'/(8E14.6))
116   FORMAT(1X,'A  AFTER ACCESS   '//(8E14.6))
12    RETURN
      END
```



STAT

```
      SUBROUTINE STAT(IU,SS)
      DIMENSION X(20),X1(20),AVD(20),MAR
     *XH1(20)
     *Z(400),EXPE(400),VEXPE(400)
      COMMON/TM/ M,Z,EXPE,VEXPE
      COMMON/TN/N,MARK,X
      COMMON/TN1/N1,X1,AVD
      COMMON/HELP1/XH1
      COMMON/IC/ICOUN
      K1=0
      DO 5 I1=1,N
      IF(MARK(I1).NE.0) GOTO 5
      K1=K1+1
      X(I1)=XH1(K1)+X1(K1)*AVD(K1)
    5 CONTINUE
      IF(IU.EQ.1) WRITE(3,119)
      ICOUN=ICOUN+1
      SS=0.
      DO 12 I=1,M
      ZI=Z(I)
      CALL FUN(1,ZI,N,X,FT)
      IF(FT.LT.10.E18) GOTO 8
      SS=10.E18
      GOTO 15
    8 FE=EXPE(I)
      C1=(FT-FE)/VEXPE(I)
      SS=SS+C1*C1
      IF(IU.EQ.1) WRITE(3,120) I,FT,FE,C1
   12 CONTINUE
      SS=SS/2
      IF(IU.EQ.1) WRITE(3,121)
  119 FORMAT(1X//5X,'1 - DEMONSTRATION OF THEORETICAL CURVE'/
     *5X,'2 -  EXPERIMENTAL VALUE'/
     *5X,'3 -  COMPARISON WITH EXPERIMENT'/
     *9X,'(-
     *'      (THEORY MINUS EXPERIMENT DIVIDED BY ERROR OF
      EXPERIMENT)'//
     *1X,
     *53(1H-)/
     *1X,
     *'I  POINT  I',6X,'1',6X,'I',6X,'2',6X,'I',6X,'3',6X,'I'/
     *1X,
     *53(1H-))
  120 FORMAT(1X,'I',I8,' I ',E11.4,' I ',E11.4,' I ',E11.4,' I')
  121 FORMAT(1X,53(1H-))
   15 RETURN
      END
```

ODM

```
        SUBROUTINE ODM(IPE,EPS,S2,S3,AL2,AL3,AL,SS)
        IF(AL3.GT.AL2) GOTO 2
 1      WRITE(3,30) S2,S3,AL2,AL3
        GOTO 28
 2      IF(S3.GT.S2) GOTO 3
        GOTO 1
 3      AL1=-AL3
        SSP=S2
C       SUBROUTINE   STATAL(AL,SS) BY AL  FINDS  SS
 5      CALl STATAL(AL1,S1)
        IF(IPE.EQ.1) WRITE(3,31) AL1,S1
 7      IF(S1.GT.S2) GOTO 9
        AL3=AL2
        AL2=AL1
        AL1=AL2-(AL3-AL2)*2.
        SSP=S1
        S3=S2
        S2=S1
        GOTO 5
 9      A1=S1*(AL2-AL3)
        A2=S2*(AL1-AL3)
        A3=S3*(AL1-AL2)
        B1=(AL2+AL3)/2.
        B2=(AL1+AL3)/2.
        B3=(AL1+AL2)/2.
        AL=(A1*B1-A2*B2+A3*B3)/(A1-A2+A3)
        CALL STATAL(AL,SS)
        IF(IPE.EQ.1) WRITE (3,32) AL,AL1,AL2,AL3,SS,S1,S2,S3
        IF(ABS(SS-SSP).LT.EPS*SQRT(SS).AND.SS.LE.SSP) GOTO 29
        IF(SS.LE.S2) GOTO 15
        IF(AL.LE.AL2) GOTO 18
        S3=SS
        AL3=AL
        GOTO 9
 18     S1=SS
        AL1=AL
        GOTO 9
 15     IF(AL.LE.AL2) GOTO 20
        S1=S2
        AL1=AL2
        GOTO 21
 20     S3=S2
        AL3=AL2
 21     AL2=AL
        S2=SS
        SSP=SS
        GOTO 9
 30     FORMAT(1X,'INCORRECT INPUT INTO SUBROUTINE FOR
       *'ONEDIMENSIONAL MINIMIZATION'/
       *1X,
       *'S2 =',E13.5/
       *1X,
       *'S3 =',E13.5/
       *1X,
       *'AL2 =',E13.5/
       *1X,
       *'AL3 =',E13.5)
 31     FORMAT(1X,'SEARCH FOR CANONICAL CONFIGURATION'/
       *1X,
       *'AL1 =',E13.5,' S1 =',E13.5)
 32     FORMAT(1X,'ARRAY OF POINTS'/
       *1X,
       *'AL =',E13.5,'AL1 =',E13.5,'AL2 =',E13.5,'AL3 =',E13.5/
       *1X,
       *'SS =',E13.5,' S1 =',E13.5,' S2 =',E13.5,' S3= ',E13.5)
 28     STOP
 29     RETURN
        END
```

INFORM

```
      SUBROUTINE INFORM(MM,SS,ICOUN,A1,XH1,AVD)
      DIMENSION XH1(20),AVD(20)
      IF(MM.EQ.0) WRITE(3,100)
      IF(MM.EQ.1) WRITE(3,101)
      IF(MM.EQ.2) WRITE(3,102)
      IF(MM.EQ.3) WRITE(3,103)
      WRITE(3,104) SS,ICOUN
      WRITE(3,105)
      DO 14 I=1,N1
      G=XH1(I)
      B=AVD(I)
      WRITE(3,106) I,G,B
   14 CONTINUE
  100 FORMAT(//,'FIRST INPUT')
  101 FORMAT(1X,'INPUT AFTER STEP BY NEWTON METHOD')
  102 FORMAT(1X,'INPUT AFTER ONEDIMENSIONAL MINIMIZATION')
  103 FORMAT(///'END SEARCH FOR MINIMUM')
  104 FORMAT(1X,E16.7,' -STATISTICAL SUM',
     *I12,' -NUMBER OF ACCESSES TO STATSUM BLOCK')
  105 FORMAT(1X,'PARAMETER ',4X,'VALUE OF PARAMETER     ',4X,
     *'LAST BASE')
  106 FORMAT(1X,I5,9X,E15.7,6X,E15.7)
      RETURN
      END
```

REZULT

```
      SUBROUTINE REZULT(ICOUN,M,N,SS,A,A1,X1,AVD,XH1)
      DIMENSION A(20,20),A1(20,20),AVD(20),X1(20),XH1(20)
      WRITE(3,125)
      DO 4 I=1,N
      DO 3 J=I,N
      AB=A1(I,J)/(AVD(I)*AVD(J))
      AB=AB*FLOAT(M-N+1)/(SS*2.)
    3 WRITE(3,126) I,J,AB
    4 CONTINUE
      WRITE(3,127)
      DO 6 I=1,N
      DO 5 J=I,N
      B=A(I,J)*(AVD(I)*AVD(J))
      B=B*2.*SS/FLOAT(M-N+1)
    5 WRITE(3,128) I,J,B
    6 CONTINUE
      WRITE(3,129)
      DO 9 I=1,N
      DO 8 J=I,N
      R=A(I,J)/SQRT(ABS(A(I,I)*A(J,J)))
    8 WRITE(3,130) I,J,R
    9 CONTINUE
  125 FORMAT(1X//5X,'FISHER MATRIX   '//1X)
  126 FORMAT(1X,'A',I3,I3,'=',E15.7)
  127 FORMAT(1X//5X,'COVARIANCE MATRIX     '/1X)
  128 FORMAT(1X,'B',I3,I3,'=',E15.7)
  129 FORMAT(1X//5X,'CORRELATION MATRIX     '/1X)
  130 FORMAT(1X,'R',I3,I3,'=',E15.7)
      RETURN
      END
```

STATAL

```
      SUBROUTINE STATAL(AL,SS)
      DIMENSION X1(20),GR(20),AVD(20)
      COMMON/GR/ GR
      COMMON/TN1/ N1,X1,AVD
      DO 1 K=1,N1
    1 X1(K)=GR(K)*AL
      CALL STAT(0,SS)
      RETURN
      END
```

An example of organizing a counting program with MODNEU subroutine call from the object module library. The FUNPOL (i,Zi,N,X,FT) subroutine is used, which approximates the experiment with a polynomial of the N-1 degree (N ⩾ 2). The degree of the polynomial is N-1, the number of selected parameters is N. /41

```
// JOB 072NEUT2                 USIKOV  TEL.  73-23
// PAUSE ASSGN SYSRLB,X'190' DISK  102
// OPTION LINK
// EXEC FFORTRAN
```

```
                      MAINPGM

        CALL MODNEU
        STOP
        END


                        FUN

         SUBROUTINE FUN(I,ZI,N,X,FT)
        DIMENSION X(20)
        CALL FUNPOL(I,ZI,N,X,FT)
        RETURN
        END


                        FUNPOL

        SUBROUTINE FUNPOL(I,ZI,N,X,FT)
        DIMENSION X(20)
        IF(I.EQ.1) GOTO 2
        N1=N-1
        WRITE(3,100) N1
        WRITE(3,101)
        GOTO 20
2       FT=X(1)
        C=ZI
        DO 3 I1=2,N
        FT=FT+C*X(I1)
3       C=C*ZI
100     FORMAT(///'EXPERIMENT APPROXIMATED BY POLYNOMIAL OF'     ,I4,
       *' DEGREE')
101     FORMAT(1X//' X(1)+X(2)*Z+X(3)*Z**2+X(4)*Z**3+...')
20      RETURN
        END
```