P.R.L.

TECHNICAL NOTE

TN-83-35

A PROGRAMMING SYSTEM FOR ESTIMATING COEFFICIENTS IN NONLINEAR MODELS

by

C.S.R.MURTY

June 1983

Physical Research Laboratory
Navrangpura,Ahmedabad-380009

C O N T E N T S

A PROGRAMMING SYSTEM FOR ESTIMATING COEFFICIENTS IN NONLINEAR MODELS

C.S.R.MURTY
Physical Research Laboratory
Ahmedabad - 380009

ABSTRACT

This note briefly describes the "derivative-free simplex method" for estimating coefficients of a model which is nonlinear in parameters. A Programming system has been developed in FORTRAN IV language on IBM 360/44 for fitting such models. It provides usually required statistical results inaddition to the estimation of coefficients. Detailed control card information for executing the system is provided. A sample problem is worked out using this program. The input and output for the sample problem is given for the purpose of illustration.

Key Words : Nonlinear curve fitting - Nonlinear models - Nonlinear regression - Derivative-free methods - Flexible geometric simplex method.

..2

I. Introduction:

Fitting of models which are non-linear in variables can be carried out using Ordinary Least Squares (OLS) analysis with appropriate transformations on variables. It is, however, not possible to use OLS for estimating a model which is non-linear in parameters. Quite a good number of methods are available for fitting nonlinear models. They can be broadly divided into two types (a) methods which involve the use of first and second order partial derivatives and (b) the methods which do not depend upon derivatives (derivative-free methods). The latter has a distinct advantage in that the model builder need not spend considerable time in partial differentiation. Among the derivative free methods, Simplex method is found to be computationally compact and efficient.

This note is organised into five sections including the introduction (Section I). Section II briefly describes simplex procedure. A FORTRAN IV programming system has been developed based on this procedure and tested on IBM 360/44. Section III contains the details of programming system. Section IV contains control card information, while section V provides the listing of 'SIMPLX' subroutine alongwith a sample problem.

II. Simplex Method:

The simplex procedure of estimating the parameters in non-linear models is a derivative-free method. It was originally suggested by Spendley et.al (1962). A modified version of this, so as to change the shape of the simplex was provided by

..3

Nelder et.al (1965) and is found more effective. The procedure requires construction of a simplex consisting of m+1 vertices with trial parameters, where m denotes the number of parameters to be estimated. For example in the model

$$Y = \frac{A}{1 + BR^X} , \quad \ldots\ldots \qquad \ldots(1)$$

A, B, R are the parameters to be estimated and X and Y are a given set of 'n' data points. The model is non-linear in parameters and has 3 of them. Hence to fit the above model one has to construct a simplex consisting of 4 vertices in 3 parameter space. One of the ways of the constructing the simplex is as follows*. Origin is taken as a vertex. The co-ordinates of the other 3 vertices are given by (K_1, K_2, K_2); (K_2, K_1, K_2); (K_2, K_2, K_1).

Here $K_1 = \frac{a}{m\sqrt{2}} \left(\sqrt{m+1} + m-1 \right)$ and

$K_2 = \frac{a}{m\sqrt{2}} \left(\sqrt{m+1} - 1 \right)$; 'a' being length of the path between two vertices (to be supplied by the user). A value of 1.0 is generally assigned to 'a'.

Given the initial simplex, the procedure begins by evaluating the error sum of squares (Z) at each of the vertices. The example given in equation (1), requires the evaluation of Z at 4 vertices. It is given by

$$Z = \sum_{i=1}^{n} \left(Y_i - \frac{A}{1+BR^{X_i}} \right)^2$$

* For other ways of constructing initial simplex see section III

..4

After evaluation the procedure iteratively replaces the vertex M corresponding to maximum Z with a new vertex by one of the operations of (a) reflection, (b) expansion, (c) contraction or (d) replacing all the vertices by new vertices till convergence is achieved. The iteration is performed according to the following steps.

Step 1:

The sample standard deviation (SD) of the functional values (Z's) is calculated at the (m+1) vertices. If $SD \leq \epsilon$. where ϵ is an arbitrarily small positive number (to be supplied by the user), then convergence is achieved (according to epsilon criterion) and the procedure terminates.* If not, it goes to next stop.

Step 2:

Let Z_M be the maximum Z corresponding to vertex M and Z_L, the minimum Z corresponding to vertex L. Let G be the centroid of all the points of simplex excluding the vertex M where the functional value is maximum.

In this step, the program reflects the vertex M to obtain the vertex M1, whose co-ordinates are given by

$$M1_i = (1+r)G_i - r\,M_i$$

where 'r', a positive constant is the reflection coefficient to be supplied by the user, A value of 1.0 is generally assigned to 'r'. $M1_i$, G_i, M_i stand for the i^{th} co-ordinates of the reflected point M1, the centroid G, and the point to be reflected M. It calculates Z_{M1} and goes to step 3.

* For alternate ways of terminating the procedure see section III

..5

Step 3:

At this stage the program compares the value of Z_{M1} with that of Z_L. If $Z_{M1} < Z_L$, it goes to step 4, otherwise to step 5.

Step 4:

In this step the program expands M1 to M2 whose co-ordinates are given by

$$M2_i = (e)\, M1_i + (1-e)\, G_i, \quad e > 1 \text{ and } e > r$$

where 'e' is the expansion coefficient, its value to be supplied by the user subject to the above mentioned conditions, A value of 2.0 is generally used. Evaluate Z_{M2}. If $Z_{M2} \leqslant Z_L$, M is replaced by M2 and the program goes back to step 1. If however, $Z_{M2} > Z_L$ expansion is said to be failed and M is replaced by M1 and the program restarts from step 1.

Step 5:

If $Z_{M1} > Z_i$ for all $i \neq M$ it goes to step 6, otherwise M is replaced by M1 and then it goes to step 1.

Step 6:

If $Z_{M1} < Z_M$, M is replaced by M1, and the program goes to step 7.

Step 7:

At this stage the program contracts the point M to M2, whose co-ordinates are given by

$$M2_i = (C)\, M_i + (1-C)\, G_i, \quad 0 < C < 1 \text{ and } C < r$$

..6

where 'C' is the contraction coefficient, its value to be supplied by the user subject to the above mentioned conditions. A value of 0.5 is generally used for 'C'. Evaluate Z_{M2}. If $Z_{M2} < Z_M$ M is replaced by M2 and the program goes back to step 1. If $Z_{M2} \geqslant Z_M$ contraction is said to be failed and the program goes to step 8.

Step 8:

At this stage all vertices are replaced by new vertices whose co-ordinates are given by

$$VN_i = 0.5\ (V_i + V_L)$$

where VN_i, V_i and V_L are respectively the co-ordinates of new vertices, old vertices and the vertex with minimum Z. The function Z is evaluated at the new vertices and the procedure restarts from step 1.

When the procedure terminates, the size of simplex is too small, in the sense that the co-ordinates of all vertices are same to the desired level of accuracy. Hence the co-ordinates of any one of them are the required parameters.

III. Programming System:

The system contains a main program which calls the subroutine 'SIMCSR'*. The subroutine 'SIMCSR' inturn calls another subroutine 'SIMPLX', to be supplied by the user.

* The subprogram 'SIMCSR' requires a number of program control cards to be supplied by the user. They are described in the next section.

..7

The subroutine 'SIMPLX' reads the input data using the object time format. It evaluates the function Z at each vertex and sends the results to the calling program. The subprogram 'SIMCSR' decides whether to stop or continue the iterative procedure. Further it decides whether to replace all vertices or only the maximum vertex. It performs the full iterative procedure described in steps 1 to 8 except evaluation of Z. It calls the 'SIMPLX', subroutine for evaluation of Z. When convergence is achieved, 'SIMPLX' subroutine also calculates the required statistical results like R^2, Mean error sum of squares, calculated values of the dependent variable using the model, difference between observed and calculated values and percentage differences. 'SIMPLX' also prints the above results inaddition to the values of the parameters.

The subroutine 'SIMPLX' requires few instructions to be modified depending on the model to be fitted. They are instructions which are meant to 'READ' and 'PRINT' the input data and instructions to calculate the right hand side of the function to be fitted. They are marked '*****' in the subprogram given below. The 5 stars can be punched from column No.76 of a standard 80 column card.

There are 4 alternative ways of forming the initial simplex.

1) The co-ordinates of all the vertices are given by trial and error.

..8

2) The co-ordinates of one of the vertex are OLS* estimates. The co-ordinates of other vertices of the simplex are given by trial and error.

3) The co-ordinates of one of the vertices are given as in 2nd way and let them be (b1, b2, b3). Now the co-ordinates of the other vertices are given by (b_1+k, b_2, b_3); (b_1, b_2+k, b_3); (b_1, b_2, b_3+k), where 'k' is a constant to be given by the user.

4) The co-ordinates of one of the vertices are given as in 3rd way. Now the co-ordinates of other vertices are given by $(b_1+k_1\ b_2, b_3)$; (b_1, b_2+k_2, b_3); (b_1, b_2, b_3+k_3). Where k_1, k_2, k_3 are constants to be given by the user.

Two more criterion are used for stopping the iterative procedure. (1) If the maximum difference between the co-ordinates of all the vertices of the simplex, in two consecutive iterations is less than epsilon (ϵ) the iteration stops. (2) If the number of iterations are greater than some prespecified value(by the user) the iterative procedure stops.

The main program for calling 'SIMCSR' subroutine is given below.

```
CALL SIMCSR
END
```

The subroutine 'SIMCSR' calls 'SIMPLX' subroutine as shown below:

```
CALL SIMPLX(B,BS,SQDEV,PS,IUNIF,ITRR,ITRE,ITRC,IFE,IFC)
```

* Perform an OLS analysis by linearising the model. Linearisation may require values to be given to some parameters. The assumed parameter values and OLS estimates constitute the co-ordinates of one of the vertex.

..9

IV. Control Card Information:

The subroutine 'SIMCSR' reads two control cards. Depending on the values taken by the control variables of these cards, 1 to 3 additional control cards will be read. The control cards and their control variables are discussed below.

Input:-

1st control card:

9 control variables are read in 2I3, 4F3.0, 3I4 format.

Control variable 1:

Specifies the number of parameters to be estimated in the non linear regression model. It should not be more than 50.

Control variable 2:

Reads the number of observations (data points)

Control variable 3:

Specifies a value for evaluating trial values of the methods. Usually a value of 1.0 is given*

Control variable 4:

Reads the reflection coefficient (r), usually a value of 1.0 is provided.

Control variable 5:

Specifies the expansion coefficient(e). The value should be greater than 1.0 and the reflection coefficient given by control variable No.4. Usually a value of 2.0 is given.

* This is the value given for 'a' to evaluate K_1 and K_2 described in section II.

..10

Control variable 6:

This specifies the contraction coefficient (C). Its value should be less than the value given to control variable No.4 (of control card No.1) and 1.0. Usually a value of 0.5 is given.

Control variable 7:

This takes 0 or 1. If 1 simplex is formed according to the 3rd or 4th way given in section III. When 1 control card No.2 will be supplied. If 0, initial simplex is formed according to the procedure described in section II.

Control variable 8:

This reads the iteration number, at which intermediate results are required. This may be required to see whether convergence is proceeding properly or not.

Control variable 9:

This reads the increment by which the value of control variable No.8 (of control card No.1) is to be augmented. The intermediate results are printed at the augmented value of the iterations.

<u>Control card 2:</u>

This control card(s) will be present if control variable No.7 (of control card No.1) takes one. The control variables of this control card(s) are read in 8F10.4 format. Number of control

..11

variables read must be equal to the number of parameters to be estimated. The values of these control variables determine the shape of the initial simplex. These are the values required if one wishes to form the simplex by the 3rd or 4th way described in section III. The values of K_1, K_2, K_3 are provided through this card(s). If $K_1 = K_2 = K_3 = K$ then simplex is formed according to 3rd way and if they are different it is formed according to 4th way given in section III.

Control Card 3:

This control card reads 5 control variables in 4I5, E14.8 format.

Control variable 1:

This takes 0 or 1, If 0, then origin is taken as one of the vertex. If 1, a format control card (control card No.4) will be read. When 1, co-ordinates of one of the vertices of the simplex will be supplied in the format specified by control card No.4.

Control variable 2:

This again takes 0 or 1. If 0, co-ordinates of the other vertices of the simplex are calculated. If 1, then a format control card (control card No.5) will be read. When 1, co-ordinates of the remaining N vertices of the simplex will be read. They will be read according to the format specified by control card No.5.

..12

Control variable 3:

This specifies the maximum number of iterations to which one wishes to iterate when convergence does not take place according to epssilon criterion.

Control variable 4:

This reads 0 or 1. If 1, co-ordinates of the vertices of the simplex will be punched by the system in 5E14.8 format. The punching of the co-ordinates of each vertex will be on separate card(s). The punching will be done at the end of the processing. If 0, no punching will be done.

Control variable 5:

This reads the epsilon ϵ value for terminating the iterative process. By default it takes the value 0.0001.

Control card 4:-

This control card will be present if control variable 1 (of control card No.2) takes the value one. This control card reads the object time format. Following this control card co-ordinates of one of the simplex are supplied in the format specified by this control card. This card is necessary when one requires to form the initial simplex by any of the 4 alternative ways given in section III.

..13

Control card 5:-

This control card will be present if the value taken by control variable 2 (of control card No.2) is 1. This control card reads the object time format. Following this control card the co-ordinates of each vertex of the remaining N vertices of the simplex are given in the format specified by this control card. The co-ordinates of each of them are to be supplied on a separate card(s). This is required when initial simplex is to be formed by way 1 or 2 as given in section III.

The subroutine 'SIMPLX' to be supplied by the user contains a control card to read the object time format. The input data is read using this object time format.

V. Subroutine 'SIMPLX' and output:

A listing of the subroutine 'SIMPLX' for the model given in equation (1) is given below. The program prints the input and the results. The results are the coefficients, R^2, Mean error sum of squares etc, and can be understood from the output and going through the program listing of the subrouting 'SIMPLX'.

```
C       SUBROUTINE SIMPLX.
        IMPLICIT REAL*8(A-H,O-Z)
        SUBROUTINE S[illegible](B,BS,SQDEV,BS,IUNIF,ITRR,ITRE,ITRC,IFE,IFC)
        DIMENSION [illegible](5[illegible],180),B(50,50),SQDEV(50),FMT(10),X(180),BS(50)
        COMMON NOB,M,IFP,IMET
C       IMET=2 IF INITIAL PARAMETERS HAVE TO BE CHANGED IF PROCEDURE
C       FAILS
        M1=M+1
        IF (IUNIF.EQ.1.AND.IFP.EQ.1) GO TO 13
        IF (IUNIF.GT.1) GO TO 4
        READ 300, FMT
        PRINT 210, FMT
   210  FORMAT (1X,'INPUT DATA READ IN FORMAT',1X,10A8)
     2  DO 1 I= 1,NOB
        READ (5,FMT) X (1,I),Y(I)                                    *****
        PRINT 650,X(1,I),Y(I)                                        *****
     1  CONTINUE
     4  ANOB = NOB
        IF (IUNIF.EQ.1) GOTO 101
        M1=1
    13  IF (IUNIF.NE.3) GOTO 101
        TSS=0.
        TOTAL=0.
        TOBAL=0.
        PRINT 1100
  1100  FORMAT (1H2)
   101  DO 51 I=1,M1
        RSS=0.
        ERRSQ=0.
        IF(IUNIF.EQ.3)PRINT  900,(B(K,K),K=1,3)                      *****
   900  FORMAT ([illegible]E14.8)
```

```
      IF(IUNIF.EQ.3) PRINT 990  11
 990  FORMAT (2X,"OBSERVED Y",7X,"CALCULATED Y",5X,
     *"DIFFERENCE",7X,"%DIFFERENCE"//)
 230  DO 640 I=1,NOB
      AI=I
      IF (IUNIF.NE.2) GOTO 634
      RHS=1.0/(BS(1)+BS(2)*(BS(3)**X(1,I)))          *****
      SUM=RHS
      GOTO 635
 634  RHS=1.0/(B(L,1)+B(L,2)*(B(L,3)**X(1,I)))     *****
      SUM=RHS
 635  DIF=Y(I)-SUM
      ERRSQ=ERRSQ+DIF*DIF
      IF(IUNIF.NE.3) GOTO 640
      PDIF=(DIF*100.0)/Y(I)
      TOBAL=TOBAL+SUM
      TOTAL=TOTAL+Y(I)
      TSS=TSS+Y(I)*Y(I)
      RSS=RSS+SUM*SUM
      YPRO=YPRO+Y(I)*SUM
      PRINT 650,Y(I),SUM,DIF,PDIF
 650  FORMAT(7(2X,E14.8,2X))
 640  CONTINUE
      IF (IUNIF.EQ.1) SQDEV(L)=ERRSQ
      IF (IUNIF.EQ.2) PS=ERRSQ
  51  CONTINUE
      IF(IUNIF.NE.3) GOTO 310
      YPRO=YPRO/ANOB
      TOTAL=TOTAL/ANOB
      TOBAL=TOBAL/ANOB
      YPRO=YPRO-TOTAL*TOBAL
      TSS1=TSS/ANOB-TOTAL*TOTAL
      TSS2=RSS/ANOB-TOBAL*TOBAL
      STS1=DSQRT(TSS1)
```

..16

```
      STS2=DSQRT(TSS2)
      RF=YPRO/(STS1*STS2)
      RSQ=RF*RF
      ESS2=TSS1-TSS2
      R2=RSS/TSS
      R21=TSS2/TSS1
      ERRSL=ERRSQ/(ANOB-M)
      PRINT 650,TOTAL,TOBAL,TSS,RSS,TSS1,TSS2,
     *ESS2,ERRSQ,ERRSL,RF,R2,R21,RSQ
    5 FORMAT(25I3)
  300 FORMAT(10A8)
  310 RETURN
      END
```

<u>Output from Computer:</u>

```
FIRST CONTROL CARD
9 PARAMETERS ARE READ IN FORMAT 2I3,4F3.0,3I4
1) NUMBER OF PARAMETERS TO BE ESTIMATED = 3
2) NUMBER OF OBSERVATIONS = 6
3) VALUE FOR EVALUATING TRAIL VALUES OF THE PARAMETERS = 1.00
4) VALUE OF REFLECTION COEFFICIENT = 1.00
5) VALUE OF EXPANTION COEFFICIENT = 2.00
6) VALUE OF CONTRACTION COEFFICIENT = 0.50
7) VALUE OF 7TH PARAMETER = 0
IF ONE READS VALUES FOR CALCULATING TRAIL VALUES
8) INTERMEDIATE RESULTS STARTS GIVING FROM 500
9) INTERMEDIATE RESULTS WILL BE GIVEN AT AN INTERVAL OF 50
```

..17

SECOND CONTROL CARD

5 PARAMETERS ARE READ IN 4I5,E14.8 FORMAT

1) VALUE OF FIRST PARAMETER = 1

IF 0 ASSUMES ORIGIN AS ONE OF THE VERTEX

IF 1 READS THE INTIAL VALUES OF THE PARAMETERS

IF 1 READS CONTROL CARD 3

WHICH IS A FORMAT CONTROL CARD

2) VALUE OF SECOND PARAMETER = 0

IF 1 READS TRAIL VALUES OF ALL VERTICES

IF 1 READS CONTROL CARD 4

WHICH IS A FORMAT CONTROL CARD

IF 0 CALCULATES TRAIL VALUES OF REMAINING VERTICES

3) MAXIMUM NUMBER OF ITERATIONS AT WHICH CONVERGENCE TOBE STOPPED = 500

4) VALUE OF 4TH PARAMETER = 0

IF VALUE IS 1 PUNCHES THE COEFFICIENTS IN 5E14.8 FORMAT

5) CRITERION VALUE AT WHICH COVERGENCE VALUE IS TO BE ACHIVED = 0.10000000D-08

CONTROL CARD 3

FORMAT CONTROL CARD

TRAIL VALUES OF ONE OF THE VERTICES IS READ IN FORMAT

(3F1.0)

```
0.20000000D  01     0.10000000D  01    0.20000000D 01
0.94280930D  00     0.23570232D  00    0.23570232D 00
0.23570232D  00     0.94280930D  00    0.23570232D 00
0.23570232D  00     0.23570232D  00    0.94280930D 00
```

INPUT DATA READ IN FORMAT (F1.0,F4.2)

```
0.10000000D  01    0.25000000D  00
0.20000000D  01    0.17000000D  00
0.30000000D  01    0.10000000D  00
0.40000000D  01    0.60000000D -01
0.50000000D  01    0.30000000D -01
0.60000000D  01    0.20000000D -01
```

..18

```
INITIAL 'Z' VALUES
0.54718024D-04  0.53932185D 01  0.82607456D 02  0.80140867D 02
'Z' VALUES AFTER CONVERGENCE
0.25977602D-04  0.25966498D-04  0.25969300D-04  0.25978494D-04
CO-ORDINATES OF VERTICES AT CONVERGENCE
0.18949365D 01    0.10853094D 01    0.19338039D 01
0.18913964D 01    0.10860682D 01    0.19336560D 01
0.18926749D 01    0.10864817D 01    0.19325636D 01
0.19026862D 01    0.10792093D 01    0.19360375D 01
 NO.OF REFLECTIONS,EXPANSIONS,CONTRACTIONS,FAILED EXPANSIONS AND
 FAILED CONTRACTIONS
   51       6       32      1      0
COEFFICIENTS
0.18949365D 01    0.10853094D 01    0.19338039D 01
OBSERVED  Y       CALCULATED  Y   DIFFERENCE       %DIFFERENCE
0.25000000D 00  0.25039362D 00  -.39361734D-03  -.15744694D 00
0.17000000D 00  0.16796682D 00  0.20331832D-02  0.11959901D 01
0.10000000D 00  0.10263239D 00  -.26323915D-02  -.26323915D 01
0.60000000D-01  0.58573579D-01  0.14264207D-02  0.23773673D 01
0.30000000D-01  0.32004657D-01  -.20046566D-02  -.66821886D 01
0.20000000D-01  0.17049403D-01  0.29505973D-02  0.14752986D 02
OTHER STATISTICS
0.10500000D 00  0.10477008D 00  0.10630000D 00  0.10618907D 00
0.66916667D-02  0.67214088D-02  -.29742104D-04  0.25977602D-04
0.43296003D-05  0.99968361D 00  0.99895642D 00  0.10044446D 01
0.99936372D 00
```

VI) Execution of the program:-

```
Job card
//CURVE EXEC FORTRAN(MAP)
Main Program
Subroutine 'SIMPLX'
/*
//SYS005 ACCESS PDSREL
//EXEC RLNKEDT(SYS005)
//EXEC CURVE
Program Control Cards
Data
/*
/&
```

VII) Acknowledgements:

I am thankful to Prof.D.Lal and Prof.P.V.Kulkarni for encouraging me to prepare this note. I am also thankful to Mr.A.V.L.Narayana for going through the manuscript and for his helpful suggestions. Thanks are due to Mrs.J.J.Trivedi and Mr.B.H.Patel who helped scientists in executing this program for scientist's specific problems. Thanks are also due to Miss Meena Shah for typing and stenciling the manuscript.

..20

VIII) References & Bibliography:

1. Spendley W.; Hext G.R. and Himsworth F.R. (1962)
'Sequential Application of Simplex designs in
optimisation and evolutionary operation'
Technometrics, Vol.4, pp.441

2. J.A.Nelder and R.Mead (1965)
'A Simplex method for function minimization'
Compt. Journal, Vol.7, pp.308

3. C.S.R.Murty (1980)
'Determination of frequencies and amplitudes
using a mathematical model'
Paper presented at the 15th CSI Annual Convention
held at Bombay

4. David M Himmelblau
'Process Analysis by Statistical methods'
John Wiley & Sons, INC, New York, 1970.