SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

132003

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA

(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina USA 27709

SPLA CONTROL NUMBER: SHR00113

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the SHARE Reference Manual, Section 6.

(1) Program Number (to be filled by SPLA) 3600-13.2.003

(2) Title of Program . . . . . NLIN: Least-Squares Estimation of Nonlinear Parameters

(3) System Type(s) (Machine) 360/370

(4) Search Key(s) . . . . . Nonlinear, parameters, regression, estimation, least-squares

(5) Programming Systems/Languages . . . . PL/I, Optional ALP Routines

(6) Primary Subject Code 13

(7) Minimum System Requirements OS PL/I Optimizing or F-Level Compiler

(8) New (N) or Revision (R) (if revision, show prior Program Number in Item 1) R

(9) Date of Submittal . . . . . August 3, 1982

(10) Documentation (number of original pages submitted) 45

(11) Author's Name and Address . . . . . R. A. Usanis
J. A. Middleton
J. H. Fulton

(12) Direct Technical Inquiries to Name & Address (if different than Author)
J. H. Fulton
N.C.S.U. Computing Center
Box 5445
Raleigh, NC 27650

(13) Submitter's Installation Membership Code . . . . . NCS

(14) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

The following changes have been made in this new version of NLIN:

1. Two new output control parameters, PARM and OBS, are available. PARM allows suppression of the listing of problem parameters, and OBS provides control over the number of observed, predicted and residual values to be printed or plotted. See Section IV.

2. Approximate coefficient values are now printed with a precision determined by problem parameter NUMDIGT.

3. Floating point underflow exceptions are suppressed during the computation of cross-products of partial derivatives.

Programmer's Guide
for
NLIN
Nonlinear least squares estimation of parameters

by

Richard A. Usanis
Computing Center
North Carolina State University
May, 1974

Contents

I. Introduction

NLIN is a PL/I version of D. W. Marquardt's main program titled 'Least Squares Estimation of Nonlinear Parameters' [3] from which much of this document has been exerpted. The original program was written in FORTRAN IV. It was converted to PL/I by J. A. Middleton of the Computing Center at North Carolina State University, Raleigh, North Carolina and submitted to the SHARE Program Library [4].

Since the original version in PL/I, several versions have been passed. The present version (Version 3) has considerably greater input-output flexibility and input data screening than earlier versions. It is in double precision.

A. The Method

NLIN is a main program to fit the model

$$Y=f\ (X_1,X_2,\ldots,X_m;b_1,b_2,\ldots,b_k)$$

where Y_i is the value predicted by the model (f) for the ith observation after estimation of the parameters (coefficients, b's) from the n sets of observed values:

$$(y_i,x_{i,1},x_{i,2},\ldots,x_{i,\ell},\ldots,x_{i,m}) \qquad i=1,2,\ldots,n$$

y_i is the observed value of the dependent variable, and the $x_{i,\ell}$ are the observed values of the m independent variables at the ith data point.

The program computes the least squares estimates of the b coefficients using the maximum neighbor method; i.e., the program minimizes

$$PHI=\sum_{i=1}^{n}\ (y_i-Y_i)^2$$

by adjusting the b_j.

A general outline of the algorithm is as follows:

Let $\underline{A}^{[k \times k]} = \underline{P}^T\underline{P}$

$\underline{P}^{[n \times k]}$ has elements

$$p_{i,j} = \frac{\partial Y_i}{\partial b_j} \qquad i=1,\ldots,n;\ j=1,\ldots,k$$

$$\underline{g}^{[k \times 1]} = \left(\sum_{i=1}^{n} (y_i - Y_i)\frac{\partial Y_i}{\partial b_j}\right) \quad j=1,\ldots,k$$

where the underlining designates an array.

Now $\underline{A}$ and $\underline{g}$ are scaled so that

$$\underline{A}' = (a'_{ij}) = \left(\frac{a_{ij}}{\sqrt{a_{ii}}\ \sqrt{a_{jj}}}\right), \text{ and}$$

$$\underline{g}' = (g'_j) = \left(g_j / \sqrt{a_{jj}}\right).$$

At the r-th iteration construct

$$(\underline{A}'^r + \lambda^r \underline{I})\underline{d}'^r = \underline{g}'^r,$$

where the superscript r denotes the iteration, and solve for d^r where $\underline{d}^r = \left(d'^r_j / \sqrt{a_{jj}}\right)$. The vector $\underline{d}$ is a direction vector and shows the general direction of the minimum. Now a vector $\underline{b}^s = \underline{b}^r + \underline{d}^r$, where the b_j are the parameters to be estimated and $s=r+1$, can be obtained.

The λ^r are selected so that

$$PHI^s < PHI^r \qquad (1)$$

until PHI is minimized. Large values of λ are used only when necessary to satisfy (1), in which case a steepest descent method of solution is approached. This method provides for rapid progress at the beginning of a problem but gives a slower later progress. When equation (1) can be satisifed by the use of a small λ the method of solution approaches an unmodified Taylor series which converges rapidly when the maximum neighborhood can be adequately represented by a linear function. Thus Marquardt's method combines both steepest-descent and Taylor's linearization method. Marquardt also found that in most cases the angle between the direction vector

produced by the steepest-descent method and that given by Taylor's series method was between 80° and 90° because of elongation of the error surface.

Marquardt's (2) strategy of choosing λ is, in general, where ϕ=PHI above:

"Let $v>1$.

Let λ^q denote the value of λ from the previous
($q=r-1$) iteration

Initially let $\lambda^0=10^{-2}$, say.

Compute $\phi(\lambda^q)$ and $\phi(\lambda^q/v)$.

i. If $\phi(\lambda^q/v)\leq\phi^r$, let $\lambda^r=\lambda^q/v$.

ii. If $\phi(\lambda^q/v)>\phi^r$, and $\phi(\lambda^q)\leq\phi^r$, let $\lambda^r=\lambda^q$.

iii. If $\phi(\lambda^q/v)>\phi^r$, and $\phi(\lambda^q)>\phi^r$
increase λ by successive multiplication by v until for some smallest w, $\phi(\lambda^q v^w)\leq\phi^r$. Let $\lambda^r=\lambda^q v^w$.

When intraparameter correlations are very high, λ can become extremely large, in which case, test iii can be altered as explained in Marquardt's paper.

Convergence has occurred when one of the tests in section I.B. are met.

The user must supply a routine named FCODE to evaluate Y_i for each $x_{i,\ell}$ and b_j combination. In addition, the user may optionally supply a routine called PCODE to evaluate the partial derivatives of Y with respect to b_j for $j=1,\ldots,k$. These routines may be either PL/I subprocedures or FORTRAN subroutines.

If PCODE (analytic derivatives) is not supplied, then NLIN will estimate the derivatives by finite difference approximations. The user should be aware that using estimated derivatives usually increases the computation time since there will be (k+1) evaluations of the function

for the k parameters to obtain the finite difference expressions; i.e.,

$$\frac{\partial Y_i}{\partial b_j} \cong \frac{f(x_{i,1},\ldots,b_j+\Delta,\ldots,b_k)-f(x_{i,1},\ldots,b_k)}{\Delta}$$

where $\Delta=\begin{cases}\text{DEL} & \text{if } b_j=0\\ \text{DEL}*b_j & \text{otherwise}\end{cases}$

and DEL is specified by the user or takes the default value 10^{-5}. When the model is insensitive to one or more of its parameters either a large value of DEL or analytical derivatives may be needed to obtain non-zero increments.

B. Convergence Criteria

The program may take one of three routes to convergence.

1. Epsilon test - This test is passed when, for all j,

   $|\Delta b_j| <= \text{CRIT}$

   where CRIT = 10.E0**-NUMDIGT if $|b_j| <= \text{TAU}$

   10.E0**-NUMDIGT*$|b_j|$ if $|b_j| > \text{TAU}$

   and Δb_j is the change in b_j from the last iteration.

2. Gamma Epsilon Test - This test is passed when all parameters have passed the epsilon test and in addition GAMMA<GAMCR, where GAMMA is the angle between the direction given by the gradient method and that given by Taylor's series method. The sum of squares (SS) should be minimized within rounding error at this point.

3. Gamma-Lambda Test - This test is passed when LAMDBA>1.0 and GAMMA>90° and indicates that the parameter estimates are highly dependent on rounding error, probably due to high correlations among the estimates of the parameters. The SS are not necessarily minimized.

C. Confidence Limits

NLIN calculates three types of confidence limits based on the problem parameters: T (student's t at $(1-\alpha)$ level) and FF (the F value at $(1-\alpha)$ level). All assume that the errors are normally and independently distributed random errors. Let $SE=(PHI/(n-k))^{1/2}$ with PHI being the minimum sums of squares; and let $\underline{C}=\underline{A}^{-1}$.

1. Single parameter confidence limits are computed as:

$$CL=b_j \pm (t_{1-\alpha,(n-k)})*SE*\sqrt{c_{jj}}$$

where t is the two tailed $(1-\alpha)$ point with $(n-k)$ degrees of freedom from Student's t distribution and b_j is the final parameter estimate. These limits provide a minimum length interval by assuming the other parameters are equal to the populational values. Note that the elements of C can be used to express the extent of the correlations (r_{ij}) among the parameter estimates as

$$\underline{R}=(r_{ij})=\left(\frac{c_{ij}}{\sqrt{c_{ii}}\ \sqrt{c_{jj}}}\right)$$

This matrix points out the parameters in a model which are highly correlated and can be used for evaluating the experimental design.

2. Support plane confidence limits are computed as

$$SP = b_j \pm (kF_{1-\alpha(k,n-k)}\ \ SE^2\ c_{jj})^{1/2}$$

where F is the variance ratio statistic at the upper $(1-\alpha)$ point of the distribution for k and $(n-k)$ degrees of freedom. These intervals are conservatively wide and considered by Marquardt (3) " to be the most realistic portrayal (within the applicability of the linear theory) of the precision of the parameter estimates individually."

3. Nonlinear confidence limits are also computed; however, since the model is nonlinear, the representation by the linear terms in a Taylor's

expansion is only an approximation in the vicinity of the minimum. This approximation will be inadequate outside some region around the minimum and thus the nonlinear confidence ellipsoids based on linear theory will not always be calculatable. When this occurs NLIN notes that the limit cannot be found in the area around the minimum.

Marquardt's (3) calculation of these limits can be briefly described as follows:

If the model is correct and the parameter estimates vary from their truevalues only because of random errors, then

$$\frac{(PHI_c - PHI)/k}{(PHI/(n-k))} \quad \text{is distrbuted as } F_{1-\alpha(n,n-k)}$$

and confidence limits can be obtained by trial and error for each parameter individually by varying PHI_c until the ratio is less than or equal to the F value. The parameters are varied one at a time. This procedure will thus give approximate (approximate in the sense of the α level) nonlinear confidence limits for the parameters.

II. Modules for NLIN

Letters in parentheses indicate the identification that is coded in columns 73-76 of the record for the source modules.

1. NLIN (NLIN) consists of two basic modules. Four sets of sample routines and test data are also available. The main program and a FORTRAN-PL/I link routine are the basic modules. The user supplies the FCODE (and optionally PCODE) routine, and the problem input data.

The REGION requirement for NLIN can be calculated approximately as the sum of the following core requirements:

1. 85 K for NLIN load module including the FORTRAN-PL/I link routine
2. 10K for transient requirements.

3. fK for the FCODE-PCODE routines. f will generally be less than 4.
4. bK for buffers. b will generally be less than 16, and is the amount of core required for buffers.
5. 10K for automatic storage plus cK for controlled storage

where: c=(4n(m+1)+16k(k+6)+8(IP+2·NTABLE+NCONS))/1024

k is the no. of parameters being estimated
n is the no. of observations
m is the no. of independent variables
NTABLE is the no. of auxillary values
NCONS is the no. of constraint residuals
IP is the no. of constant parameters

2. The FORTRAN-PL/I link routine (FTLK) is only necessary if FORTRAN is used to write FCODE (and optionally PCODE). In the main program, the link routine is called from card numbers NLIN0399,0400,0411, and 0412. The entry points are declared in card numbers NLIN0009 to NLIN0012. This routine requires about 220_{10} bytes of storage.

3. A third module (PLFP) is a PL/I source routine for testing NLIN. The model is Y=A*EXP(B*X). It contains FCODE and PCODE separated by a '*PROCESS;' card and can be compiled from disk or tape.

4. A fourth module (PLDA) is the data for the model in the third module. The correct results are:

	COEFFICIENTS	
PROBLEMS	A	B
1,4,5,6	1.03714	0.959716
2	1.0	0.980914
3	0.977976	1.0

5. This module (FOFP) is the same model as in 3, but written in FORTRAN and can be compiled directly by FORTRAN G or H.

6. Data (FODA) for module 5 which should give the same results as from module 4.

7. A PL/I test program (SKFP) using the model $y=a/x + b/x^c$. Both FCODE and PCODE are included and separated by a '*PROCESS;' card so that they can be compiled directly.

8. Data (SKDA) appropriate for the model in module 7. The estimated parameter values should be approximately:

		COEFFICIENTS	
PROBLEM	A	B	C
1	7.9982E+01	2.00066E+01	2.50075E-01
2-40	4.98125E+01	5.00812E+01	2.50331E-01

The second problem is run 40 times for timing comparisons in testing PL/I compilers.

9. FORTRAN subroutines (MQFP) FCODE and PCODE using Marquardt's test model, Y=A*EXP(B*X)+C*EXP(D*X). The subroutines are directly compilable.

10. Data (MQDA) for Marquardt's test model. The approximate parameter estimates are:

    5.90710E+00, -2.02158E-01, 2.49243E+00, -5.48951-02

III. How to put NLIN on disk in load module form.

Example Job Control Language to store NLIN on disk in load module form is given below. Procedure ASFC assembles the PL/I-FORTRAN link routine (FTLK) and stores the resulting object module in a temporary data set. Procedure PLOCL compiles the PL/I main program (NLIN), adds the resulting object module to the temporary data set and executes the Linkage-Editor which processes the combined object modules as primary input. The Assembler F-level and PL/I Optimizing compilers are used in this example, and the storage device is a 3330 disk drive.

```
     (TIME used here is CPU time plus I/O EXCP time, on a S/370/165.)
//jobname  JOB  account,programmername,TIME=(3,15),PAGES=65,M=1
//STP1  EXEC  ASFC
//C.SYSIN  DD  define data set containing FORTRAN-PL/I interface source routine
//STP2  EXEC PLOCL,PARM.C='OPTIMIZE(TIME)',REGION.C=300K
//C.SYSIN  DD  define data set containing PL/I source code for NLIN
//L.SYSLMOD  DD  define data set to contain NLIN Load Module, UNIT=DISK,
// DISP=(NEW,CATLG),SPACE=(TRK,(25,,1),RLSE),VOL=SER=NCSON1
//L.SYSIN  DD  *
   LIBRARY (FCODE)
/*
//
```

IV. How to use the NLIN load module

A. With PL/I FCODE, PCODE routines:

```
//jobname  JOB  account,programmername
//A  EXEC  PLOCLG,PARM.C='OPTIMIZE(TIME)',REGION.G=125K
//*  PL/I OPTIMIZER COMPILE,LINK-EDIT AND EXECUTE
//C.SYSIN  DD  define dataset containing:
                FCODE
               *PROCESS;  } if analytic derivatives are used.
                PCODE     }
//L.MYLIB  DD  DISP=SHR,data set name of load module library
//L.SYSIN  DD *
       INCLUDE MYLIB(programname)
       ENTRY PLISTART
/*
//G.SYSIN  DD  define data set containing problem data
//
```

B. With FORTRAN FCODE, PCODE subroutines:

```
//jobname  JOB  account,programmername
//A  EXEC  FTGCLG,REGION.G=130K
//*  FORTRAN-G COMPILE, LINK-EDIT AND EXECUTE
//C.SYSIN  DD  define data set containing:
         SUBROUTINE FCODE
         SUBROUTINE PCODE - if analytic derivatives are used.
//L.MYLIB  DD  DISP=SHR,data set name of load module library
//L.SYSIN  DD  *
       INCLUDE MYLIB(programname)
       ENTRY PLISTART
/*
//G.SYSPRINT  DD  SYSOUT=A
//G.SYSIN DD  define data set containing problem data
//
```

The example JCL assumes the use of the PL/I Optimizing compiler and the FORTRAN IV G-level compiler. The PL/I F-level and FORTRAN IV H-level compilers can also be used.

V. REFERENCES

1. DRAPER, and SMITH. 1966. Applied Regression Analysis. Wiley and Sons. N. Y.
2. Marquardt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Indust. Appl. Math: pp. 431-441.
3. Marquardt, D. W. 1966. Least-squares estimation of nonlinear parameters. IBM SHARE Program Library No. SDA 3094.01.
4. Middleton, J. A. 1969. Least-squares estimation of non-linear parameters - NLIN. IBM SHARE Program Library No. 360D-13.2.003.

Appendix A

Magnetic Tape Key

The tape volume contains 10 files and 11 tape marks (TM) as shown below. The DCB information for all files can be summarized by saying:

(RECFM=FB,LRECL=80,BLKSIZE=1680)

All source programs have an alphabetic code in card columns 73-76 and are sequence numbered in card columns 77-80. The data decks contain no identification. The files are arranged as follows:

File 1
PL/I Source Deck (NLIN main program)
EBCDIC
NLIN in cc 73-76; all cards; sequence no. 0001 thru 1079 in cc 77-80
1079 card images
TM

File 2
Assembler (OS/360) subroutine FORTLNK
EBCDIC
FTLK in cc 73-76, all cards; sequence no. 0001 thru 0064
in cc 77-80.
64 card images
TM

File 3
PL/I test problem source routines FCODE, PCODE separated with a '*PROCESS;' card.
EBCDIC
PLFP in 73-76, all cards; sequence no. 0001 thru 0015
in cc 77-80.
15 card images
TM

File 4
Test problem data for PL/I routines
EBCDIC
No identification
21 card images
TM

File 5
FORTRAN test problem source subroutines FCODE,PCODE.
EBCDIC
FOFP in cc 73-76, all cards; sequence no. 0001
thru 0014 in cc 77-80;
14 card images
TM

File 6 Test problem data for FORTRAN subroutines.
EBCDIC
No identification
21 card images
TM

File 7 Shrikhande's model; PL/I problem source routines
FCODE,PCODE separated with a '*PROCESS;' card.
EBCDIC
SKFP in cc 73-76, all cards; sequence no. 0001 thru 0027
in cc 77-80;
27 card images
TM

File 8 Shrikhande's model test data
EBCDIC
No identification
10 card images
TM

File 9 Marquardt's model; FORTRAN problem source subroutines
FCODE,PCODE
EBCDIC
MQFP in cc 73-76, all cards; sequence no. 0001 thru 0024
in cc 77-80;
24 card images
TM

File 10 Marquardt's model test data
EBCDIC
No identification
8 card images
TM
TM

June 1982

Library Services Series
Document No. LSR-089-2

NLIN -- LEAST SQUARES ESTIMATION OF NONLINEAR PARAMETERS

Documentation Prepared
by
User Services
Computing Center
North Carolina State University
Raleigh, NC 27650

SUPPORT TYPE: C
VERSION: 3.3

Triangle Universities Computation Center
Research Triangle Park, NC 27709

TABLE OF CONTENTS

I. FUNCTION

NLIN is a main program to fit the model

$$Y = f(X_1, \ldots, X_m; b_1, \ldots, b_k)$$

to the n sets of observed values

$$(y_i, x_{i,1}, \ldots, x_{i,m}) \qquad i = 1, 2, \ldots, n$$

by adjusting the coefficients b to minimize

$$PHI = \sum_{i=1}^{n} (y_i - Y_i)^2$$

Y is the value of the dependent variable Y predicted by the model f for the ith observation. y_i is the ith observed value of the dependent variable, and $x_{i,j}$ is the ith observed value of the independent variable X_j. For a complete discussion of the method, see the references in Section III.

Any number of problems can be processed in one run as long as the basic model remains unchanged. A problem is defined as one set of observations and the associated parameters. Options are provided to use either estimated or analytic partial derivatives of f with respect to the coefficients, to control the amount and detail of printing, to omit parameters (that is, to introduce constant coefficients into the model), to plot (on the printer) the observed and predicted values of the dependent variable, to introduce constraints on any coefficient value, to obtain nonlinear confidence limits, to limit the amount of new data to be read for a new problem, to allow the use of values used or computed for the previous problem, and to use either FORTRAN or PL/I external routines. Observed values can be read from disk, tape or cards. Any number of observations may be used within the limits of main memory.

II. USE

A. Job Control Language

The program may be executed at TUCC using the following JCL, where NCS.NLIN.LOAD is the name of the data set containing the NLIN program. The PL/I Optimizing and FORTRAN level G or H compilers can be used to compile user-supplied routines FCODE and PCODE.

1. For PL/I routines FCODE and PCODE:

```
//jobname  JOB  xxx.yyy.zzz,username
//     EXEC  PLOCLG
//C.SYSIN  DD  *
          (procedure FCODE)
*PROCESS;
          (procedure PCODE)
//L.LIB     DD  DSN=NCS.NLIN.LOAD,DISP=SHR
//L.SYSIN   DD  *
  INCLUDE   LIB(NLIN)
  ENTRY   PLISTART
//G.SYSIN   DD  *
      (data)
//
```

2. For FORTRAN routines FCODE and PCODE:

```
//jobname  JOB  xxx.yyy.zzz,username
//  EXEC  FTHC
//C.SYSIN  DD  *
          (subroutine FCODE)
          (subroutine PCODE)
//  EXEC  PLOLG
//L.SYSLIB   DD  DSN=SYS1.FORTLIB,DISP=SHR
//L.LIB      DD  DSN=NCS.NLIN.LOAD,DISP=SHR
//L.SYSIN    DD  *
  INCLUDE   LIB(NLIN)
  ENTRY   PLISTART
//G.FT03F001   DD  SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)
//G.SYSIN   DD  *
    (data)
//
```

B. Input

Input to the program consists of five data groups in the following order:

i. Problem parameters;

ii. Subscripts of constant or omitted coefficients (optional, see Section II.B.2.);

iii. Observed values of the independent variables X (optional after the first problem and can be in a separate data set);

iv. Observed values of the dependent variable Y (optional after the first problem, and can be in a separate data set with the independent variable values); and

v. Initial subjective estimates for the k coefficients (optional after the first problem).

These data groups can be repeated as necessary for subsequent problems. The problem parameters are prepared as PL/I data-directed input (see B.1. below); the remaining data groups are prepared as PL/I list-directed input (see Section II.B.3).

1. Problem Parameters

All problem parameters receive a default value when execution is initiated. Changes to these values are introduced by including a parameter in a problem parameter input list.

The problem parameters appear in the input stream as a series of items of the form:

parameter-name = value

and each is separated from the preceding item by a comma and any number of blanks. The entire group is ended with a semicolon. This format of input is called data-directed input. See the examples below. Only those parameters whose values are to be changed from the value used in the previous problem (or from the default value in the case of the first problem) need be included in the list; the semicolon must be present even if no parameters are specified. The parameters may be in any order; they may begin in any position and may be continued from one record to the next, as long as no blanks are introduced either in the parameter name or in the value by so doing. Parameters are read from file SYSIN (default input file for PL/I). Except for parameter ID, all parameters should have integer values if they begin with one of the letters I through N; and may have either a decimal point, an E scale factor, or both, in all other cases. Arithmetic values may be signed.

A complete list of all problem parameters and their defaults is in Appendix B. The most widely used problem parameters are:

ID — A user identification for the problem, up to 100 characters long, enclosed in apostrophes. Example: ID='TEST 1'. Default is ID=' '. Apostrophes to be included in the identification value must be coded as two apostrophes, e.g., ID='SMITH''S TEST'.

N	Number of observations. N=0 indicates the end of the input data. Default is zero.
K	Total number of coefficients to be estimated. Default is one.
M	Number of independent variables. Default is one.
IP	Number of constant coefficients (see Section B.2). Default is zero.
ITOFF	Maximum number of iterations allowed to search for a solution. If this number of iterations is reached, all information obtained to this point is printed, and the message "FORCE OFF" is printed. Default is 50.
NOPCODE	Determines use of analytic or estimated (finite difference) partial derivatives. If NOPCODE > 0, estimated derivatives will be used (PCODE will not be called). If NOPCODE <= 0, analytic derivatives will be used (PCODE will be called). Default is one and estimated derivatives will be used. PCODE is described in Section II.D.
TAU	Determines whether convergence is based on significant digits or decimal places (see Epsilon Test in Section III.A). If the absolute value of a coefficient is greater than TAU, then significant digits are used in testing for convergence; otherwise, convergence will be to a specified number of decimal places. Default is zero.
NUMDIGT	Number of decimal digits to be used in testing for convergence. The parameter value must be in the range 0 < NUMDIGT < 16. Default is four. (See Section III.A.3)
DEL	Multiplier for estimated derivative calculations. DEL must be in the range 0 < DEL < 1. Default is 1.E-5. It is recommended, although not essential, that the exponent of DEL be less than -NUMDIGT.
GAMCR	Criterion angle used in the gamma-epsilon test. Default is 45.0 degrees. See Section III.A.4.

These parameters allow for the solution of most problems. Observations and coefficients would be included for each problem, and a limited amount of printout would be obtained for each iteration with detailed information given upon the completion of each problem. Optional parameters controlling the amount and form of output (including plot output), limiting problem input, controlling confidence limit calculations, and other parameters available are explained in Section IV.

Examples of problem parameter input lists:

```
ID='FIRST DEGREE', N=20, K=2, M=3 ;

N=16, K=3, GAMCR=30., ITOFF=10, NUMDIGT=6 ;

N=0 ;                    (End of problem group.)
```

2. Constant Coefficient Subscripts

Constant coefficients are useful for making changes to a model. For example, the model

$$y = b_1\left(1-b_2e^{-b_3x}\right)^{1-b_4}$$

can represent $y = b_1(1-b_2e^{-b_3x})$

if b_4 is held constant with the value zero. Coefficients b_2 and b_3 could be held constant at one to produce another variant of the model.

If coefficients are to be held constant during the solution of a problem, then specify problem parameter IP to be the number of such coefficients and provide a list of the coefficient subscripts (corresponding to the subscripts appearing in the model equation) in the input data. These should appear as IP integers separated by commas and any number of blanks (list-directed format). The list may begin at any point in file SYSIN after the semicolon which terminates the problem parameters. Values for constant coefficients are specified along with the estimated values of nonconstant coefficients. See B.5 below.

3. Independent Variables

Observed values of the independent variables must be included for the first problem; they can be omitted for subsequent problems (see use of parameter NOX in

Section IV). These values must be listed in order by variable, one observation at a time. For example,

where n is the number of observations, and m is the number of independent variables. This is list-directed input where the values are separated by a comma and any number of blanks. There must be n*m of these values, if they are required.

These values are normally obtained from file SYSIN, but they may be included in a data set (disk, card or tape) that is separate from the other program input and has ddname DATA. Observed values of the dependent variable must be in the same data set as these values. Thus, if file DATA is defined, it is used as the input file for observed values. To use file DATA, include a DD statement of the form:

```
//G.DATA  DD  DSN=datasetname,DISP=SHR
```

preceding the usual //G.SYSIN DD * statement. See also problem parameter NVY in Section IV.A.1.

4. Dependent Variables

Observed values of the dependent variable must be included for the first problem, but they may be omitted for subsequent problems (see use of parameter NOY in Section IV). Values for the dependent variable must be given in order by observation. There must be n of these values, if they are required. The format of the values is the same as is indicated for the independent variables. Optional use of file DATA is as described above and must be used if the independent variables are read from file DATA.

5. Initial Coefficient Values

The initial estimates of the coefficients must be included for the first problem and may be omitted for subsequent problems (see use of problem parameter NOCOEFF in Section IV.A.1). The order of the coefficients is determined by the user-written FCODE and PCODE subroutines (see Sections II.D and VI.A). The format of the coefficients is the same as that of the independent variables (list-directed). These values are read from file SYSIN.

Record boundaries may be ignored when entering values as long as blanks are not introduced into a value itself. Sample input is included in Section VI.

C. Default Output

Output for each problem will begin on a new page (an example of the output is given in Section VI.A.3). Program and problem identification are printed, followed by a table of the problem parameters. For each iteration until problem termination, the following information is printed and labeled: number of iterations completed, approximate coefficient values, value of PHI (sum of squares of residuals), standard error of PHI (SE=SQRT(PHI/(N-K+IP))), LENGTH of increment vector, value of GAMMA (angle between the negative gradient and the Taylor series vectors), value of LAMBDA (a parameter which facilitates convergence), and a message indicating whether analytic or finite-difference partial derivatives were used. When one of the criteria used to terminate iteration is met (see Section III), an appropriate message is printed at the top of a new page, followed by the number of iterations completed, approximate coefficient values, a table of the observed and predicted values of the dependent variable, and the residuals at each data point. The values of PHI, SE and LAMBDA are then printed, as are the inverse of the normal equation matrix, the parameter correlation matrix, and a table of the standard error, one-parameter confidence limits and support-plane confidence limits for each coefficient. Last, the non-linear confidence limits are printed.

All standard output is to file SYSPRINT (normally the printer); output record length is 133 characters including carriage control. The user can also output information to other files from the user-written routines FCODE or PCODE. All computations are double-precision. Six significant digits are printed for most output values. Approximate coefficient values are printed with NUMDIGT+1 significant digits.

D. User-Written Routines

The user must supply an external routine named FCODE which evaluates the model equation and calculates the residual, given estimates of the coefficients. In addition, if analytic derivatives are to be used (NOPCODE =0), a routine named PCODE is required which evaluates the partial derivatives of the equation with respect to the coefficients. These routines may have either PL/I-compatible calling sequences or, if problem parameter NOPL1 > 0, FORTRAN-compatible calling sequences (see Section IV.A.4).

1. PL/I ROUTINES

a. FCODE calling sequence:

```
CALL FCODE (Y,X,B,PRNT,F,I,RES,M,N,K);
```

M, N and K are problem parameters as defined in Section II.B.1. Y is a vector containing N observed values of the dependent variable. X is a two-dimensional (N,M) array containing N sets of observed values of the M independent variables. B is a vector containing K values of the current coefficients. PRNT is a vector which may contain any user-defined output values, and is also useful as a means of passing values from FCODE to PCODE (see use of problem parameter NTABLE in Section IV.2). FCODE should compute the predicted value of the model at the Ith observation, given X and B, and store the value in argument F. The residual should then be calculated and stored in argument RES (RES=Y(I)-F) subject to considerations of constraints and weights as discussed in Section IV. A sample subroutine is included in Section VI. Y, X, B and PRNT must be declared as arrays with dimensions as follows:

```
DCL Y(*), X(*,*), (B(*), F, RES, PRNT(*)) FLOAT (16);
```

Note that X and Y are the only single-precision arguments; B, PRNT, F and RES being double-precision.

b. PCODE calling sequence:

```
CALL PCODE (P,X,B,PRNT,F,I,M,N,K);
```

M, N and K are problem parameters as defined in Section II.B.1. X, B, PRNT and I are as used in procedure FCODE, and F is the model value as computed by FCODE for the Ith observation. P is a vector which PCODE must use to store the values of the K partial derivatives of the model equation with respect to the coefficients B(1),...,B(K), evaluated at the Ith observation. See Section VI.A.1 for an example. The arguments should be declared with dimensions as follows:

```
DCL X(*,*), (P(*),B(*),PRNT(*),F) FLOAT (16);
```

2. FORTRAN Routines

a. FCODE calling Sequence:

```
CALL FCODE (Y,X,B,PRNT,F,I,RES,M,N,K)
```

M, N and K are problem parameters as defined in Section II.B.1. Y is a vector containing the N observed values of the dependent variable. X is a two-dimensional (M,N) array containing the N sets of observed values of the M independent variables. B is a vector containing K values of the current coefficients. PRNT is a vector which FCODE can use to store user-defined output values, and is also a means of passing values from FCODE to PCODE (see problem parameter NTABLE in Section IV.2).

FCODE should compute the predicted value of the model for the Ith observation and store the value in argument F, given X and B. The residual should then be calculated and stored in argument RES (RES=Y(I)-F) subject to considerations of constraints and weights as discussed in Section IV. A sample routine is included in Section VI. The variables and arrays should be declared as follows:

```
REAL*4 Y(N),X(M,N)
REAL*8 B(K),PRNT(1),F,RES
```

Note the order of subscripting on X: the first subscript is the independent variable index; the second subscript is the observation index. B, PRNT, F and RES are double-precision; X and Y are single-precision.

b. PCODE calling sequence:

```
CALL PCODE (P,X,B,PRNT,F,I,M,N,K)
```

M, N and K are problem parameters as defined in Section II.B.1. X, B, PRNT and I are values as used in FCODE, and F is the value of the model as computed by FCODE for the Ith observation. P is a vector which PCODE must use to store the values of the K partial derivatives of the model equation with respect to the coefficients evaluated at the Ith observation. See Section VI.A for an example.

The variables and arrays should be declared as follows:

```
REAL*4 X(M,N)
REAL*8 P(K),B(K),PRNT(1),F
```

III. ANALYSIS

The algorithm selects an optimized correction vector for the coefficients by interpolation between the vector obtained by the gradient method and the vector obtained by a Taylor series expansion truncated after the first derivative. Iteration is applied to this vector according to the least-squares method of estimating parameters until one of the criteria listed below is met. For a detailed discussion of the algorithm, see D. W. Marquardt's article, "An Algorithm for Least Squares Estimation of Non-linear Parameters," Journal of the Soc. of Industrial Applied Math., Vol II (1963), pp. 431-441.

A. Criteria for Problem Termination

1. Force Off: the specified maximum number of iterations were completed before satisfying another termination criterion. See problem parameter ITOFF in Section II.B.1. PHI is not minimized.

2. Gamma-Lambda test: the Marquardt parameter lambda has become > 1, while the angle, gamma, between the negative gradient vector and the Taylor series vector is > 90 degrees. This usually indicates the presence of very high correlations among the coefficient estimates. In this situation, the coefficient corrections may be dependent upon rounding error and PHI is not necessarily minimized.

3. Epsilon Test: This test is passed when two successive approximations of each coefficient round to either:

   (a) the same set of NUMDIGT significant digits, or

   (b) the same set of NUMDIGT decimal places.

   Which criterion is used depends on the magnitude of the coefficient being tested and the problem parameter TAU. If the absolute value of the coefficient is greater than TAU, the relative criterion (a) is used. When the absolute value of the coefficient is less than or equal to TAU,

criterion (b), an absolute test, is made. Thus if TAU is equal to zero, its default value, criterion (a) will always be used; however, if TAU is set larger than the largest possible value of the coefficients, an absolute test will always be made. The user can tell which criterion was used for each coefficient by comparing the magnitude of the coefficient to the value of TAU.

4. Gamma-Epsilon test: The epsilon test is satisfied, and at the same time the angle, gamma, between the negative gradient vector and the Taylor series vector is less than problem parameter GAMCR. PHI is assumed to be minimized within rounding error.

5. User termination: During the execution of either FCODE or PCODE, the user may elect to terminate the solution of the current problem by executing the statement CALL FEXIT in FORTRAN, or SIGNAL CONDITION (PEXIT); in PL/I. NLIN then prints the message *****USER EXIT followed by current values of the coefficients and proceeds to the next problem.

6. PCODE missing: If NOPCODE = 0 and the user fails to supply a PCODE subroutine, NLIN terminates execution abnormally with User Completion Code 001.

7. Singular matrix: If a singular A = P'P matrix (where P is the vector of partial derivatives) is generated, NLIN prints the message *****SINGULAR MATRIX, followed by the row and column indices of the pivot and its value, then proceeds to the next problem. When very high correlations exist among the coefficients, the matrix A can become singular due to rounding error alone. The program uses a perturbed version of A during iteration; thus the matrix is usually positive definite and the singularity will not occur until the actual A matrix is inverted at convergence.

B. Confidence Limits

Three types of confidence limits can be computed:

1. Conventional one-parameter confidence limits based on Student's t distribution (the T problem parameter) at the (1-alpha) level with (N-K+IP) degrees of freedom; these are minimum length intervals.

2. Support plane intervals for each individual parameter. These intervals are conservatively wide

and are based on the F statistic (the FF problem parameter) with (K-IP) and (N-K+IP) degrees of freedom.

3. Non-linear confidence ellipsoids calculated using the F statistic.

For a discussion of the theory and usefulness of the various confidence intervals, see Exhibit B (Confidence Region Calculations) in the appendix to the documentation of the SHARE program, "Least Squares Estimation of Nonlinear Parameters" (SDA 3094-01), by D. W. Marquardt. (This document is no longer available from SPLA.) Another source of information concerning confidence intervals is Applied Regression Analysis, by N. R. Drapper and H. Smith (John Wiley & Sons, Inc., 1966). Chapter 10 of this book includes a summary of D. W. Marquardt's method of parameter estimation as used in this program, as well as discussion of confidence intervals for parameters in nonlinear models.

IV. OTHER OPTIONS AVAILABLE

A. Optional Problem Parameters

The following problem parameters may be used to apply the program to a variety of specific uses. These parameters, if used, are included in the problem parameter input data described earlier. Arranged by type of application (input control, output control, confidence limits, language options, computation aids and constraints), they are:

1. Input Control:

NOX — provides for the use of the previous values of the independent variables, as defined in the just completed problem. If NOX <= 0, new values will be read from file DATA; if DATA is not defined, from file SYSIN. If NOX >= 1, the X values from the previous problem will be used. Default is zero. NOX must be <= 0 if either M or N is changed to a new value.

NOY — provides for use of the previous values of the dependent variable. If NOY <= 0, new values of the dependent variable will be read from file DATA, or from file SYSIN if DATA is not defined. If NOY >= 1, the Y values from the previous problem will be used. Default is zero. NOY must be <= 0 if N is changed to a new value.

NOCOEFF provides for the use of previously defined initial coefficient values. If NOCOEFF <= 0, initial values of the coefficients will be read from file SYSIN. NOCOEFF must be <= 0 for the first problem. If NOCOEFF = 1, the initial values of the coefficients will be the initial values used in the preceding problem. If NOCOEFF >= 2, the initial values will be the final coefficient values obtained in the preceding problem. The default value is zero. NOCOEFF must be <= 0 if K is changed.

NONUCOE is to be used in conjunction with constant or omitted coefficients; that is, NONUCOE is meaningful only if IP > 0. If NONUCOE <= 0, subscripts of the constant coefficients will be read from SYSIN. If NONUCOE >= 1, the constant coefficients will be those used in the previous problem. NONUCOE must be <= 0 if IP is changed. The default value is zero.

NVY is the order or position of the value of the Y (dependent) variable within an observation. If the X and Y values are to be read as described in Section II.B (all X values followed by all Y values), then NVY = 0, the default value, is required. If the data is to be read one observation at a time; i.e., in (X,Y) groups, then NVY should be set equal to the position of the Y variable within the observation. Thus, if the data input is coded as:

```
X11 X12 X13 Y1 X14
X21 X22 X23 Y2 X24
    ...
```

then NVY=4; if coded as X1 Y1 X2 Y2 ..., where there is only one independent variable per observation, then NVY=2. NVY must be in the range 1 to M+1 if this form of input is used. When NVY > 0 for a problem, NOX and NOY must both be zero or > 0. The data must still be in list-directed input form; that is, each element must be separated by a comma and any number of blanks. There must be (M+1)*N values.

2. Output Control

IFP controls plotting on the printer. IFP <= 0 indicates that no plot is to be produced. If IFP > 0, a plot of observed and predicted values of the dependent variable at each observation is produced. The position of a plotted point is correct to within two percent of the range of values plotted. The plot is made one line per observation and does not consider the values of the independent variables. The predicted values and corresponding residuals are printed in the margin to the right of the plot. Once a problem termination criterion has been met, the plot, preceded by a statement of the number of iterations completed and a list of approximate coefficient values, is inserted in the output stream preceding the usual output of problem results as described in Section II.C. Symbols used in the plot are: P, predicted value of the dependent variable; O, observed value; Y, observed and predicted values assigned to the same plot position; and X, value is beyond the range of the plot. The default value of IFP is zero.

YMN is the value of the dependent variable to be assigned to the left margin of the plot. The default value of YMN is zero.

SPRD is the range of values (not largest value) of the dependent variable to be covered by the plot. The right margin of the plot corresponds to the value YMN + SPRD. The default value of SPRD is 100.

ITPRNT is the number of iterations for which a detailed printout of computations is desired. This detailed printout includes the number of completed iterations; the approximate coefficient values; either the plot (if IFP > 0) or a table of observed, predicted, and residual values; the P'P correlation matrix (where P is the vector of partial derivatives); the value of PHI (sum of squares of residuals); the standard error of PHI (SE=SQRT(PHI/(N-K+IP))); the value of LAMBDA (parameter to facilitate convergence); a message indicating whether

analytic or partial derivatives were used; and a series of increments added to the coefficients, with corresponding values of PHI, LAMBDA, GAMMA (angle between the negative gradient vector and the Taylor series vector) and lengths of increment vectors. The default value of ITPRNT is zero.

NOPRNT is the intermediate printout suppressor. NOPRNT <= 0 provides detailed printout for ITPRNT iterations. For subsequent iterations, output is as described in Section II.C. NOPRNT > 0 suppresses output after ITPRNT iterations until a problem termination criterion has been met, then prints the solution. The default value of NOPRNT is zero.

NTABLE allows printing of values defined by the user, but not normally included in the output. These values are printed in up to four columns which appear to the right of the table of observed, predicted, and residual values. NTABLE is the length of the vector PRNT, into which the user puts the values to be printed. These values may be defined in either FCODE or PCODE, and stored in the argument vector. If more than four values are to be printed, the fifth will be placed under the first value on the following line, and similarly for the ninth, thirteenth, etc. These values are printed for each observation, and must be defined each time FCODE or PCODE is called, using current argument values. (Note that PCODE is called after FCODE, and only if NOPCODE <= 0.) Note that PRNT values computed in FCODE are available for use in PCODE computations.

PARM controls printing of the table of problem parameter values at the beginning of a problem. If PARM <= 0, then the table is omitted. The default value is one.

OBS selects observations for which observed, predicted, and residual values will be printed or plotted. OBS <= 0 suppresses all such output. If OBS = n, where n is some positive integer, then output will appear for observation numbers n, 2n, 3n,

etc. The default value is one.

3. Confidence Limits:

FF — is the variable ratio statistic with (K-IP,N+IP-K) degrees of freedom, used in calculations of confidence limits. The default value is four. FF must be positive.

T — is the two-tailed (1-alpha) point of Student's t distribution with (N+IP-K) degrees of freedom, to be used in calculations of confidence limits. The default value is two. T must be positive.

NOLIM — determines whether or not nonlinear confidence limits are calculated and printed. NOLIM <= 0 provides the calculation; NOLIM > 0 suppresses them. The default value is zero. Printout includes the value of PHI-CRITICAL and confidence limits on each individual coefficient such that PHI is approximately equal to PHI-CRITICAL at these limits (all other coefficients are unchanged).

4. Language options:

NOPL1 — provides for use of either FORTRAN-compatible or PL/I-compatible external procedures FCODE and PCODE. If NOPL1 <= 0, then FCODE and PCODE will be called with PL/I-compatible calling sequences. If NOPL1 >= 1, then FCODE and PCODE will be called with FORTRAN-compatible calling sequences. The default value is zero.

5. Computation aids:

XLAMBDA — is the initial value of lambda, the Marquart parameter used to facilitate convergence. If XLAMBDA <= 0, then lambda is first used with a value of 0.01. The default value of XLAMBDA is zero.

ZETA — is the singularity criterion for matrix inversion, used in the routine to invert the matrix (P'P - lambda I). The default value is 1.0E-31. ZETA must be non-negative.

6. Constraints:

NCONS is the number of constraints imposed on the model. These are introduced by means of additional residuals computed in FCODE. Once the residuals have been obtained for each observation, FCODE is called once more for each constraint applied, with argument I = N+c, where c=1, 2, ..., NCONS. Thus, for I=N+c, FCODE should return the value of the cth constraint residual. The constraint residual functions should be continuous in the vicinity of the bounds imposed. NCONS=0 by default. The introduction of constraints in this way may not be satisfactory, as the path to convergence is affected. If use of constraints appears necessary, a test run should be made to determine whether or not satisfactory results can be obtained. The need for constraints can often be eliminated by replacing a variable coefficient with a constant, plus (or minus) a positive function. For example, the coefficient B in the model Y=A*X+B*EXP(X) can be constrained to a value >= 1 by letting B = 1+C**2, or B = 1+EXP(C), and solving for C rather than B.

B. Application Options:

1. Weighted residuals:

A weighted residual can be easily obtained since the residuals are defined in FCODE. For example, the user can compute a weighted residual by multiplying the unweighted residual by the square root of its corresponding weight factor.

2. Initial problem computation - NEW:

An external variable called NEW is set equal to one (1) at the initiation of each problem and is not otherwise used by the main program. PL/I users can access its value by declaring NEW BIN FIXED (30) EXTERNAL. FORTRAN users can access this value by referring to an INTEGER*4 variable in a named COMMON block called NEW: COMMON/NEW/variable-name. If this value is changed by the user, it will be reset to one in the main program when the next problem is initiated. See the example in Section IV.C.1.

3. Iteration check - ICT:

This option allows the user to determine when the last calls to FCODE or PCODE are being made. The variable ICT is set by NLIN as follows:

ICT=0	iteration is continuing.
ICT=1	iteration terminated by FORCE OFF.
ICT=2	gamma-lambda test satisfied.
ICT=3	epsilon test satisfied.
ICT=4	gamma-epsilon test satisfied.

Thus, this variable can be used to determine when to output to tape or disk the final coefficient values or other information that is available to FCODE or PCODE. See the examples in Section IV.C.1. At any given iteration, FCODE (and PCODE) will be called at least once for each observation so that any output will be done N times or more unless precautions are taken to output the information only once. If changed by the user, the value of ICT will be reset to zero at the beginning of the next problem.

In PL/I procdures, the variable can be accessed by declaring ICT as FIXED BINARY (30) EXTERNAL. FORTRAN users may obtain the value by referring to an INTEGER*4 variable in a named COMMON block called ICT; that is, COMMON/ICT/variable.

FORTRAN USER NOTE: When using a FORTRAN subprogram, the user must close all files he opened by use of REWIND statements at the end of his last problem, so he must know how many problems he has and keep track of how many are completed. This can cause some problems when a terminal error occurs, since FCODE will not be called. The end result will be the loss of some information.

C. Use of Options and Inputting Other Types of Data

1. Use of NEW and ICT

```
FCODE: PROCEDURE (Y,X,B,PRNT,F,I,RES,M,,N,K);
  DCL X(*,*),Y(*),(B(*),PRNT(*),F,RES) FLOAT (16),
    (NEW,ICT) FIXED BIN (30) EXTERNAL,
    PBL FIXED BIN INITIAL (0) STATIC;
  IF NEW  = 1 THEN DO;
    /* TRANSFORM OBSERVATIONS */
    GET LIST (TRANS);
    DO II = 1 TO N;
      Y(II)=Y(II)*TRANS+5;
      X(II,1)=EXP(X(II,1));
```

```
      END;
    NEW=0;
    END;    /* END OF TRANSFORM */
...calculate F and RES according to model equation...
  IF I = N & ICT > 0 THEN DO;
    /* END OF PROBLEM */
    PBL=PBL + 1
    IF ICT > 1 THEN PUT FILE(MYDATA) LIST(PBL,ICT,B);
    ICT=-1;
    END;
  RETURN;
END FCODE;
```

This example illustrates several uses of FCODE other than simply for model evaluation (see also 2.b below);

a. The variable NEW is used to control those operations that are done only when a new problem is being started. In this case, a variable, TRANS, is read from the standard system input device. The value would be placed right after the initial coefficient values in the input deck (it could have been in some other file). The variable is then used to transform the dependent variable. I, M, N, and K should not be changed in either FCODE or PCODE. The first X variable is also transformed. These transformations are done for all observations on the first entry of FCODE for each problem. Note that NEW is set to a value other than one so that transformation will only be done on the first entry of each problem.

b. The variable PBL is initialized to zero and declared static. Thus its value is only initialized once during the job. It is then incremented at the completion of every problem, that is, when I=N and ICT is greater than zero. PBL acts as a problem counter.

c. At the completion of a problem, when FCODE is being entered for the last obsevation after convergence, the problem number, type of test passed, and the K coefficients are written out to the user file MYDATA. The user must include the proper JCL to define this file.

2. Other types of data decks

If the data is not in list-directed format (that is, has unused numbers, no blanks between numbers, or such), then several techniques can be used to enter the data:

a. If the data is all numeric and the variables to be used are separated from each other and the rest of the data by blanks or commas, then M can be specified as being the number of X variables plus the number of other data fields on the card. Only the appropriate variables are then used in FCODE.

b. If the data must be read with a format, then the variables can be read by the user routine, FCODE, by use of:

    1. some dummy data,
    2. the NEW option, and
    3. a formatted READ in FCODE.

The dummy data is required because NLIN always reads data for the initial problem. It could consist of (M+1)*N constants, such as zeros, separated by blanks or a comma. The NEW option is used so that FCODE can read the formatted data on first entry for a problem. Dummy data need not be entered for succeeding problems if NOY and NOX are then set greater than zero.

A FORTRAN example:

```
      SUBROUTINE FCODE (Y,X,B,PRNT,F,I,RES,M,N,K)
      REAL*4 X(M,N),Y(N)
      REAL*8 B(K),PRNT(1),F,RES
      COMMON/NEW/ NPBL
      IF (NPBL.NE.1) GO TO 100
C*** BEGINNING A PROBLEM ***
      DO 1 II=1,N
 1    READ(1,2) (X(J,II),J=1,M), Y(II)
 2    FORMAT (10X,F10.5,6X,F5.3,10X,F10.1)
      NPBL = 0
C*** NOW COMPUTE F AND RES ***
 100  F = B(1)*X(1,I)*DEXP(-B(2)*X(2,I))
      RES = Y(I)-F
      RETURN
      END
```

This example reads N sets of observations at the beginning of each problem. Remember not to use

argument I in the subroutine statement as a DO-loop variable. Transformations could have been performed after the data was read. Note that the variable NPBL in COMMON/NEW/ is changed in value to prevent reading of data on subsequent calls to FCODE for the current problem.

V. ERROR CONDITIONS AND RESPONSES:

All program error messages are preceded by asterisks and printed in the left margin.

A. Screening Input Parameters

Fairly extensive testing of problem parameters is made after input. For the first problem, NOCOEFF, NOX, and NOY are tested; if any one or more of these is > 0, a message is printed indicating their value, and a zero value assumed. The remaining problem data are then read, and execution proceeds.

Parameter GAMCR, the criterion angle used in the gamma-epsilon test, is adjusted to the range 0 <= GAMCR <= 180. If GAMCR has an absolute value > 1.4742E+6, then it assumes the default value. A message describing the action is printed any time the value is changed after input.

If N-K+IP <= 0, a message is printed indicating that more observations are needed. Execution then proceeds to the next problem. The following parameter relations are tested (see Sections II.B.1 and IV.A):

0 < DEL < 1	0 < NUMDIGT < 16
0 < ITOFF	0 <= ZETA
0 < FF	0 < T
0 <= NCONS	0 < SPRD

An invalid relation causes the erroneous parameter value to be printed with a message indicating that the parameter assumes the default value.

B. User Routine Missing

If routine PCODE is missing and NOPCODE <= 0, then NLIN terminates execution abnormally with User Completion Code 001.

C. PL/I ON-Units

Actions taken by ON-units for errors detected in PL/I routines are as follows:

1. ON UNDEFINED FILE: if file SYSPRINT or file SYSIN is not defined, execution is terminated with job step return code 1000.

2. ON ENDFILE: if an end-of-file condition arises during input, a message is printed and execution is terminated. If the condition arises during problem parameter input, the message indicates *****END OF PROBLEM GROUP REACHED. Otherwise, the message indicates *****END OF FILE REACHED WHILE ATTEMPTING TO READ DATA FOR CURRENT PROBLEM.

3. ON NAME: if an unrecognizable variable name appears in the input stream in the problem parameter list, a message is printed indicating the contents of the field and stating that this field is being ignored. Execution continues.

4. ON CONVERSION: if a conversion error arises during input, a message is printed indicating the contents of the field. An attempt is then made to begin input for a new problem.

5. Any other error results in the printing of a system message, followed by a program error message indicating the type of error or the PL/I ONCODE number. Processing then continues with the next problem in the group.

D. FORTRAN Routines

Action taken as a result of an error arising in FORTRAN routines FCODE and PCODE is as follows:

If either a floating-point or fixed-point divide exception, or an exponent overflow occurs, a message is printed indicating both the type of error and the FORTRAN routine in which it occurred (either FCODE or PCODE). Underflow, fixed-point overflow, and significance exceptions are masked off during execution of the FORTRAN routines.

If any other error is detected, a system message or a FORTRAN error message (prefix IHO for FORTRAN H Extended) is printed, and the job is terminated abnormally. Most FORTRAN errors are trapped by the NLIN-FORTRAN interface and are not serviced by FORTRAN.

VI. PROGRAM TESTS AND EXAMPLES

Several sample problems were run to test program performance. Most available options were used in one or another of the runs. A summary of the results of these tests is given in this section, as well as examples of FCODE, PCODE, and input data.

A. Description of Models Tested

For the primary series of tests, the observations and coefficient values were taken from the SHARE program SDA-3094.01 (see SLUR, serial number U-67).

Model equation: Y=A*EXP(B*X)

1. FCODE and PCODE routines for this function:

   a. PL/I routines:

```
FCODE: PROCEDURE (Y,X,B,PRNT,F,I,RES,M,N,K);
DCL Y(*),X(*,*),(B(*),PRNT(*),F,RES) FLOAT (16);
PRNT(1) = X(I,1);
PRNT(2) = EXP(B(2)*X(I,1));
F = B(1)*PRNT(2);
RES = Y(I)-F;
RETURN;
END FCODE;

PCODE: PROCEDURE (P,X,B,PRNT,F,I,M,N,K);
DCL X(*,*),(B(*),PRNT(*),P(*),F) FLOAT (16);
P(1) = PRNT(2);
P(2) = P(1)*B(1)*X(I,1);
RETURN;
END PCODE;
```

   b. FORTRAN routines:

```
      SUBROUTINE FCODE (Y,X,B,PRNT,F,I,RES,M,N,K)
      REAL*4 X(M,N),Y(N)
      REAL*8 B(K),PRNT(2),F,RES
      PRNT(1) = X(1,I)
      PRNT(2) = DEXP(B(2)*X(1,I))
      F = B(1)*PRNT(2)
      RES = Y(I)-F
      RETURN
      END

      SUBROUTINE PCODE (P,X,B,PRNT,F,I,M,N,K)
      REAL*4 X(M,N)
      REAL*8 P(K),B(K),PRNT(2),F
      P(1) = PRNT(2)
```

```
      P(2) = P(1)*B(1)*X(1,I)
      RETURN
      END
```

2. Input Data

Sample input data for a group of six problems using PL/I procedures FCODE and PCODE follows:

```
Record #
1   ID='PL/1 TEST 1', N=14, K=2, M=1, NTABLE=2, NUMDIGT=6
2   IFP=1, SPRD=8;
3   .01 .05 .1 .2 .3 .4 .5 .6 .7 .8 .9 1. 1.5 2.
4   1. 1.05 1.13 1.2 1.35 1.5 1.62 1.8 2. 2.4 2.5 2.7 4.5 7.
5   1.0 1.0
6   ID='PL/1 TEST 2', NOX=1, NOY=1, IP=1, NOCOEFF=1, ITPRINT=2;
7   1
8   ID='PL/1 TEST 3' ;
9   2
10  ID='PL/1 TEST 4', IP=0, NOCOEFF=0;
11  1.0 0.1
12  ID='PL/1 TEST 5' ;
13  2.0 2.0
14  ID='PL/1 TEST 6', NOX=0, NOY=0, NVY=1;
15  1. .01 1.05 .05
16  1.13 0.1 1.2 .2
17  1.35 .3 1.5 .4 1.62 .5
18  1.8 .6 2. .7 2.4 .8 2.5 .9
19  2.7 1. 4.5 1.5 7. 2.
20  1.0 1.0
21  N=0;
```

This input could have been put on about eight statements, but is on 21 for clarity. The only change necessary to run this data with FORTRAN subroutines would be addition of the problem parameter NOPL1=1 on statement 1 or between statements 1 and 2.

The final coefficients for problems 1, 4, 5, and 6 are approximately 1.03714 and 0.959716. For problem 2 they are 1.0 and 0.980914, and for problem 3 they should be close to 0.977976 and 1.0.

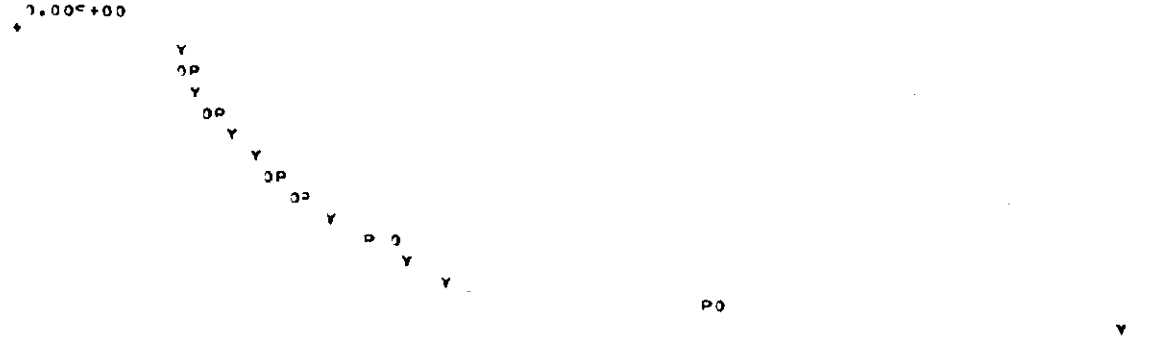
3. Output

The output produced by NLIN using the routines in Section VI.A.1.a and the first five input statements given in Section VI.A.2 is presented on the following pages.

```
PROBLEM NUMBER   1                    *****  NCSU COMPUTING CENTER  *****
         ***  NONLINEAR LEAST SQUARES FIT. VERSION 3.3  ***

USER IDENTIFICATION: PL/1 TEST 1

      N=          14          NO. DATA PTS.                                  NTABLE=      2          NO. AUX. VALUES PRINTED.
      K=           2          NO. COEFFS.                                    NOLIM=       0          NON-LIN. CONF. LIM. SUPPRESSOR.
      IP=          0          NO. CONSTANT COEFFS.                           NOPCODE=     0          USE OF EST. DERIV. INDICATOR.
      M=           1          NO. INDEP.VARS.                                NOX=         0          USE OF PREV. X-ARRAY INDICATOR.
      IFP=         1          PLOTTER OPTION.                                NOY=         0          USE OF PREV. Y VECTOR.
      ITOFF=      50          MAX. ITERATIONS.                               NDCOEFF=     0          USE OF A PREV. COEFF. VECTOR.
      IPRNT=       0          DETAIL PRINTOUT INDICATOR.                     NOPL1=       0          USE OF FORTRAN SUBROUTINES.
      NOPRNT=      0          INTERM. PRINTOUT SUPPRESSOR.                   NCONS=       0          NO. CONSTRAINTS.
      NUMDIGT=     6          NO. DEC. DIGITS USED IN CONV. TEST.            NONUCOE=     0          USE OF PREV. CONST. COEFF. SUBSCR.
      NVY=         0          NO. OF VARIABLE Y. OBSERVATIONWISE INPUT       OBS=         1          OBSERVATION PRINT CONTROL.
      TAU=      0.00000E+00   CONSTANT FOR CONVERGENCE.                      YMN=      0.00000E+00   VALUE OF LEFT PLOT MARGIN.
      GAMCR=    4.50000E+01   CRITERION ANGLE.                               SPRD=     8.00000E+00   WIDTH OF PLOT.
      T=        2.00000E+00   STUDENT'S 'T'.                                 ZETA=     1.00000E-31   SINGULARITY CRITERION FOR MATRIX INV.
      FF=       4.00000E+00   VARIANCE RATIO STATISTIC.                      DEL=      1.00000E-05   MULTIPLIER FOR EST. DERIV.
      X_LAMBDA= 0.00000E+00   LAMBDA.

      NO. ITERATIONS COMPLETED:      0
      APPROX. VALUES OF COEFFS.:       COEFF(  1) =  1.000000E+00
                                       COEFF(  2) =  1.000000E+00

                PHI                 S E              LAMBDA          ANALYTIC PARTIALS USED
            1.86849E-01         1.24783E-01          1.00E-02

      NO. ITERATIONS COMPLETED:      1
      APPROX. VALUES OF COEFFS.:       COEFF(  1) =  1.036392E+00
                                       COEFF(  2) =  9.601180E-01

                PHI                 S E              LENGTH          GAMMA           LAMBDA          ANALYTIC PARTIALS USED
            6.39689E-02         7.30119E-02          7.74ZE-01       6.615E+01       1.000E-03

      NO. ITERATIONS COMPLETED:      2
      APPROX. VALUES OF COEFFS.:       COEFF(  1) =  1.037144E+00
                                       COEFF(  2) =  9.597112E-01

                PHI                 S E              LENGTH          GAMMA           LAMBDA          ANALYTIC PARTIALS USED
            6.39598E-02         7.30067E-02          9.998E-03       5.212E+01       1.000E-04

      NO. ITERATIONS COMPLETED:      3
      APPROX. VALUES OF COEFFS.:       COEFF(  1) =  1.037137E+00
                                       COEFF(  2) =  9.597162E-01

                PHI                 S E              LENGTH          GAMMA           LAMBDA          ANALYTIC PARTIALS USED
            6.39598E-02         7.30067E-02          1.076E-04       5.213E+01       1.000E-05
```

PROBLEM NUMBER 1 EPSILON TEST SATISFIED
USER IDENTIFICATION: PL/1 TEST 1

NO. ITERATIONS COMPLETED: 3
APPROX. VALUES OF COEFFS.: COEFF(1) = 1.037137E+00
COEFF(2) = 9.597162E-01



PRED	RES
1.04714E+00	-4.71385E-02
1.08812E+00	-3.81190E-02
1.14161E+00	-1.16064E-02
1.25660E+00	-5.65972E-02
1.38317E+00	-3.31718E-02
1.52250E+00	-2.24951E-02
1.67585E+00	-5.58529E-02
1.84466E+00	-4.46585E-02
2.03047E+00	-3.04661E-02
2.23499E+00	1.65009E-01
2.46012E+00	3.98838E-02
2.70792E+00	-7.91861E-03
4.37558E+00	1.24423E-01
7.07025E+00	-7.02540E-02

OBS	PRED	DIFF		
1.00300E+00	1.04714E+00	-4.71385E-02	1.00000E-02	1.00964E+00
1.05300E+00	1.08812E+00	-3.81190E-02	5.00000E-02	1.04916E+00
1.13000E+00	1.14161E+00	-1.16064E-02	1.00000E-01	1.10073E+00
1.20000E+00	1.25660E+00	-5.65972E-02	2.00000E-01	1.21160E+00
1.35000E+00	1.38317E+00	-3.31718E-02	3.00000E-01	1.33364E+00
1.50000E+00	1.52250E+00	-2.24951E-02	4.00000E-01	1.46798E+00
1.62000E+00	1.67585E+00	-5.58529E-02	5.00000E-01	1.61585E+00
1.80000E+00	1.84466E+00	-4.46585E-02	6.00000E-01	1.77861E+00
2.00000E+00	2.03047E+00	-3.04661E-02	7.00000E-01	1.95776E+00
2.40000E+00	2.23499E+00	1.65009E-01	8.00000E-01	2.15496E+00
2.50000E+00	2.46012E+00	3.98838E-02	9.00000E-01	2.37203E+00
2.70000E+00	2.70792E+00	-7.91861E-03	1.00000E+00	2.61096E+00
4.50000E+00	4.37558E+00	1.24423E-01	1.50000E+00	4.21890E+00
7.00000E+00	7.07025E+00	-7.02540E-02	2.00000E+00	6.81709E+00

PHI	S E	LAMBDA	ANALYTIC PARTIALS USED
6.39598E-02	7.30067E-02	1.000E-05	

PTP INVERSE

1	6.22172E-02	-3.50815E-02
2	-3.50815E-02	2.35827E-02

PARAMETER CORRELATION MATRIX

1	1.0000E+00	-9.1585E-01
2	-9.1585E-01	1.0000E+00

COEFF	STD ERROR	ONE - PARAMETER LOWER	ONE - PARAMETER UPPER	SUPPORT PLANE LOWER	SUPPORT PLANE UPPER
1	1.82103E-02	1.00072E+00	1.07356E+00	9.85630E-01	1.08864E+00
2	1.12114E-02	9.37293E-01	9.82139E-01	9.28006E-01	9.91427E-01

NONLINEAR CONFIDENCE LIMITS

PHI CRITICAL = 1.06600E-01

COEFF	LOWER B	LOWER PHI	UPPER B	UPPER PHI
1	1.01646E+00	1.06600E-01	1.05782E+00	1.06600E-01
2	9.46856E-01	1.06545E-01	9.72277E-01	1.06515E-01

FOR PROBLEM NO. 1 THE ELASPED CPU TIME WAS : 0.18 SECONDS.

B. Test Results

Various combinations of problem parameters were used, and the results were quite satisfactory. Some details of interest are:

1. The use of analytic rather than estimated partials causes convergence to the same coefficient estimates (for NUMDIGT digits) with greater consistency for different starting values of the coefficients, as well as slightly reducing both the number of iterations required and the run time.

2. If estimated derivatives are used, the choice of DEL can significantly affect performance. The smaller the value of DEL, the better is the accuracy of the derivative estimates; however, if DEL is too small (less than 1.0E-15), all significant digits are lost in the computation of the derivative and the resulting derivative is zero. Recall that DEL is used in the derivative calculations as a multiplier. Thus, if a model is to be used for repeated runs with estimated derivatives, it could be helpful for the user to test various values of DEL to find an optimum.

3. Since FORTRAN and PL/I produce slightly different object codes, the use of one language will cause the path of convergence to differ slightly from the path of convergence if the other were used to write FCODE and PCODE. "Path of convergence" implies the successive coefficient approximations. However, neither language was consistently preferable to the other, and the variations in the path of convergence became negligible when analytic partials were used.

C. Other Models Tested:

1. The model

```
Y = A/X + B/(X**2)
```

was fit to 40 sets of 10 data points, totaling 200 iterations in 19.2 seconds of CPU time and 17.2 seconds of I/O time using analytical derivatives.

2. The sample data of Marquardt (SHARE SDA 3094.01) was also fit with this program. The answers were the same as he obtained with his program. The equation was fit in 7 iterations totaling 5.4 seconds (3.5 CPU seconds; 1.9 I/O seconds).

VII. ACKNOWLEDGEMENTS

This program is a modified and extended version of the now obsolete SHARE program SDA 3094.01 (NLIN) by D. W. Marquardt. The initial coding, debugging, and preparation of the write-up of the original version was performed by Joseph A. Middleton, formerly of the Computing Center staff. Subsequent revisions were contributed by R. A. Usanis and J. H. Fulton.

VIII. APPENDIX

A. Program Information

SOURCE LANGUAGE: The main program is written in PL/I and was compiled and tested at TUCC using the PL/I Optimizing compiler.

REGION REQUIREMENTS: For most applications, necessary region is under 145K. If FCODE and PCODE are coded in FORTRAN, this estimate should be increased by about 20K. The additional storage is required for FORTRAN library routines.

B. Parameter Default Values: Summary

Following is a table of all problem parameters, their default values and their meaning (see Sections II.B.1 and IV.A):

Parameter	Default Value	Meaning (in parentheses, meaning of default value)
DEL	1.0E-5	Multiplier for estimated derivatives
FF	4.	Variance ratio statistic
GAMCR	45.	Gamma criterion angle
ID	' '	User identification of problem (no identification)
IFP	0	Plot option (plot suppressed)
ITOFF	50	Max. no. of iterations
IP	0	Number of constant coefficients
ITPRNT	0	Number of iterations of detailed printout
K	1	Number of coefficients to be estimated

M	1	Number of independent variables
N	0	Number of observations (end of problem group)
NCONS	0	Number of constraint residuals
NOCOEFF	0	Input of initial coefficients (will be read)
NOLIM	0	Nonlinear confidence limits (will be computed)
NONUCOE	0	If IP > 0, input of constant coefficient subscripts (will be read)
NOPCODE	1	Estimated or analytic partials (estimated)
NOPL1	0	PL/I or FORTRAN calling sequence (PL/I)
NOPRNT	0	Intermediate printout suppression (not suppressed)
NOX	0	Input of independent variable data (will be read)
NOY	0	Input of dependent variable data (will be read)
NTABLE	0	Number of auxiliary output values
NUMDIGT	4	Number of digits used in convergence test
NVY	0	Position of variable Y when reading one observation at a time (read all X values, then all Y values)
PARM	1	List table of problem parameters (will be listed)
OBS	1	Select observations to be printed or plotted (all)
SPRD	100.	Data range of plot
T	2.	Students's t
TAU	0.	Constant to determine type of convergence test
XLAMBDA	0.	Initial value of lambda (will be corrected to 0.01)
YMN	0.	Left margin value for plotting
ZETA	1.0E-31	Singularity criterion for matrix inversion

I. FUNCTION:

NLIN is a main program to fit the model

$$Y_i=f(x_{i,1},x_{i,2},\ldots,x_{i,m};b_1,b_2,\ldots,b_k)$$

to the set of N observed values

$$(Y_i,x_{i,1},x_{i,2},\ldots,x_{i,m}) \qquad i=1,2,\ldots,N$$

where Y_i is the observed dependent variable and the $x_{i,1}$ are the m independent variables for the i^{th} data point.

The program computes the least squares estimates of the b_j coefficients (parameters) using the maximum neighbor procedures; i.e., the program minimizes

$$PHI=\sum_{i=1}^{N} (Y_i-f(x_{i,1},\ldots,x_{i,m};b_1,\ldots,b_r))^2$$

by adjusting the b_j. For a complete discussion of the method see the references in Section III.

Any number of problems, where a problem is defined as one set of observations and the associated parameters, can be processed in one run as long as the basic model (see Section II.B.2) remains unchanged. Options are provided to use either estimated or analytic partial derivatives of Y with respect to b_j; to control the amount and detail of printing; to omit parameters (i.e., to introduce constant coefficients into the model); to plot, on the printer, the observed and predicted values of the dependent variable; to introduce constraints on any coefficient value; to obtain nonlinear confidence limits; to limit the amount of new data to be read for a new problem; to allow the use of values used or computed for the previous problem; and to use either FORTRAN or PL/I external routines. Input data points can be read from disk, tape or cards. Any number of data points may be used.

II. USE:

A. Job Control Language

The program may be executed by the following JCL (at TUCC) where NCS.LOADLIB is the location of the NLIN program. PL/I F and FORTRAN level G or H can be used for these routines.

1. For PL/I routines FCODE and PCODE:

```
//jname  JOB  xxx.yyy.zzz,username
//      EXEC  PLFCLG,REGION.G=135K
//C.SYSIN DD *
            (FCODE deck)
```

```
*PROCESS;
         (PCODE deck)
/*
//L.MYLIB  DD  DSNAME=NCS.LOADLIB,DISP=SHR
//L.SYSIN DD *
       INCLUDE MYLIB(NLIN)
       ENTRY   IHENTRY
/*
//G.SYSIN DD *
          (data)
/*
//
```

2. For FORTRAN routines FCODE and PCODE:

```
//JOBNAME  JOB  xxx.yyy.zzz,username
//       EXEC   FTGC
//C.SYSIN  DD  *
            (FCODE deck)
            (PCODE deck)
/*
//       EXEC  PLFLG,REGION.G=135K
//L.SYSLIB  DD  DSNAME=SYS1.FORTLIB,DISP=SHR
//L.MYLIB  DD  DSNAME=NCS.LOADLIB,DISP=SHR
//L.SYSIN  DD  *
       INCLUDE MYLIB(NLIN)
       ENTRY   IHENTRY
/*
//G.FTØ3FØØ1  DD  SYSOUT=A
//G.SYSIN  DD  *
            (data)
/*
//
```

B. Input

Input to the program consists of five data groups, in the following order:

i. Problem parameters;

ii. Subscripts of constant or omitted coefficients (optional, see Sec. B.2.);

iii. Values of the independent variables ($X_{i,j}$) for the N data points of the M independent variables (optional after the first problem, and can be in a separate data set);

iv. Values of the dependent variables (Y_i) observed for the N data points (optional after the first problem, and can be in separate data set with the independent variable values); and

v. Initial guesses (estimates) for the K coefficients (optional after the first problem).

These data groups can be repeated as necessary for subsequent problems. The problem parameters are prepared as PL/I data-directed input (see B.1. below); the remaining data groups are prepared as PL/I list-directed input (see Sec. B. 3 below).

1. Problem Parameters

All problem parameters receive a default value when execution is initiated. Changes to these values are introduced by including a parameter in a problem parameter input list.

The problem parameters appear in the input stream as a series of items of the form:

parametername=value

and each item is separated from the preceding items by a comma and any number of blanks. The entire group is ended with a semi-colon. (This format of input is called data-directed input. See examples at end of B.1.) Only those parameters whose values are to be changed from the value used in the previous problem (or from the default value in the case of the first problem) need be included in the list; the semi-colon must be present even if no parameters are specified. The parameters may be in any order; they may begin in any position and may be continued from one record to the next, as long as no blanks are introduced either in the parameter name or in the value by so doing. Parameters are read from file SYSIN (default input file for PL/I, normally the card reader). Except for parameter ID, all parameters should have integer values if they begin with letter I through N, and may have either a decimal point, E scale factor or both, in all other cases. The value may be preceded by a '+' or '-'.

The most widely used problem parameters are:

ID A user identification for the problem, up to one-hundred characters long, enclosed in single quotes. (Example: ID='TEST 1'). Default is ID=' '. Apostrophes to be included in the identification itself must be coded as two apostrophes, e.g., ID='SMITH''S TEST.

N	Number of data points. N<=0 indicates the end of the input data. Default is zero.
K	Total number of coefficients. Default is one.
M	Number of independent variables. Default is one.
IP	Number of constant coefficients (see Sec. B.2.). Default is zero.
ITOFF	Maximum number of iterations allowed to search for a solution. If this number of iterations is reached, all information obtained to this point is printed and a message, 'FORCE OFF', printed on the output. Default is 50.
NOPCODE	Determines use of analytic or estimated (finite difference) partial derivatives. If NOPCODE>0, estimated derivatives will be used (PCODE will not be called); if NOPCODE<=0, analytic derivatives will be used (PCODE will be called). Default is 1, estimated derivatives will be used; for PCODE description see Section D.
TAU	Determines whether convergence is based on significant digits or decimal digits (see Epsilon Test in Section III). If the absolute value of the coefficient is greater than TAU, then significant digits are used in testing for convergence; otherwise, decimal digits will be used for testing; default is zero.
NUMDIGT	Number of significant digits to be used in testing for convergence if the absolute value of the coefficient is greater than TAU. The parameter value must be in the range 0<NUMDIGT<16; default is 4.
DEL	Multiplier for estimated derivative calculations. DEL must be in the range 0<DEL<1. Default is 1.E-5. It is recommended, although not essential, that the exponent of DEL (5 in the default value) be smaller than -NUMDIGT.
GAMCR	Criterion angle used in gamma-epsilon test. Default is 45.0 degrees.

These parameters and their default values allow for the solution of most problems; all data points and coefficients must be included for each problem, and a limited amount of printout will be obtained for each iteration with detailed information given upon the completion of each problem. Optional parameters controlling amount and form of output (including plot output), limiting problem input, controlling confidence limit calculations, and other parameters available are explained in Section IV, "Other Options Available."

Examples of problem parameter input lists:

```
ID='FIRST DEGREE',N=20,K=2,M=3;
N=16,K=3,GAMCR=30.,ITOFF=10,NUMDIGT=6;
N=0; (End of problem group.)
```

2. Constant Coefficient Subscripts

A list of the subscripts of the coefficients (corresponding to the subscripts as used in the function) to be held constant during the iterations must be included, if and only if, IP>0, i.e. the number of constant coefficient is greater than zero. In this case IP subscripts should be listed, in any order. The subscripts appear in the input stream as integers separated by a comma and any number of blanks (list-directed form). The list may begin at any point in the input stream after the semi-colon ending the problem parameter list. These subscripts are read from file SYSIN (normally the card reader).

Constant coefficients are particularly useful for easily making changes to the basic model. For example, the model

$$y=b_1(1-b_2e^{-b_3x})^{\frac{1}{1-b_4}}$$

can represent $y=b_1(1-b_2e^{-b_3x})$ if $b_4=0$, IP=1

or $y=b_1(1-e^{-x})^{\frac{1}{1-b_4}}$ if $b_2=b_3=1$, IP=2

or many other possible models. For the last model above, the constant coefficient subscripts punched on the card would be: 2, 3 and the initial coefficient estimates would be 1 for both parameters.

3. Independent Variables

Values of the independent variables must be included for the first problem; they can be omitted for subsequent problems (see use of parameter NOX in Section IV, "Other Options Available"). These values

must be listed, in order, for one data point at a time, followed by those for the next

$$x_{1,1}, x_{1,2}, \ldots, x_{1,m}, x_{2,1}, \ldots, x_{n,m}$$

where n is the number of data points and m is the number of independent variables. This is list-directed input where the data points are separated by a comma and any number of blanks. There must be N*M of these values, if they are present.

These values are normally obtained from file SYSIN (usually the card reader); however, they may be included in a data set (disk, card or tape) that is separate from the other program input and has ddname=DATA. The dependent variables must be in the same data set as these values. Thus, if file DATA is defined, it is used as the input file for the data points. To use file DATA, include a DD card of the form:

```
//G.DATA DD DSNAME=dsname,etc.
```

preceding the usual //G.SYSIN DD * card. See also the NVY option in Section IV.A, Input Control.

4. Dependent Variables

Values of the dependent variable must be included for the first problem; they can be omitted for subsequent problems (see use of parameter NOY in Section IV, "Other Options Available"). The order of these values must be the same as that of the first subscript of the independent variables. There must be N of these values, if they are present. The format of the values is the same as is indicated for the independent variables. Optional use of file DATA is as described above, and must be used if the independent variables are read from file DATA.

5. Initial Coefficient Values

The initial guesses of the coefficients must be included for the first problem, and may be omitted for subsequent problems (see use of problem parameter NOCOEFF in Section IV, "Other Options Available"). The order of the coefficients is determined by the user written FCODE and PCODE subroutines (see Section II.D.). The format of the coefficients is the same as that of the independent variables (list-directed). These values are read from file SYSIN.

Record boundaries may be ignored for the data groups, as long as blanks are not introduced into a value itself. Sample input is included with the description of "Program Tests and Examples", Section VI.

C. Default Output

Output for each problem will begin on a new page (an example of the output is given in Section VI.A.3). Program and problem identification are printed, followed by a table of the problem parameters. All output tables are labeled, as are values such as those mentioned below. For each iteration until problem termination, the following information is printed: number of iterations completed; approximate coefficient values; value of PHI (sum of squares of residuals); standard error of PHI (SE=SQRT(PHI/ (N-K+IP))); LENGTH of increment vector; value of GAMMA (angle between the negative gradient and the Taylor series vectors); value of LAMBDA(a parameter which facilitates convergence); and a message indicating whether analytic or finite-difference partial derivatives were used. When one of the criteria used to terminate computation is met, (see "Analysis", Section III) an appropriate message is printed at the top right of a new page. Next is printed the number of iterations completed and approximate coefficient values. Following this is a table of the observed-predicted values of the dependent variable and the residuals at each data point. The values of PHI, SE and LAMBDA, as described earlier, are then printed; as are the inverse of the normal equations matrix, the parameter correlation matrix, and a table of the standard error, one-parameter confidence limits and support-plane confidence limits for each coefficient. Lastly the non-linear confidence limits are printed.

All standard output is to file SYSPRINT (normally the printer); output record length is 133 characters including carriage control. The user can also output information to other files from the user written routines. All computations are double precision, but only six significant digits are printed for most output values.

D. User Written Routines

The user must supply an external routine named FCODE which evaluates the function and calculates the residual given the newest estimates of the coefficients. In addition, if analytic derivatives are to be used (NOPCODE<=0), a routine named PCODE, which evaluates the partial derivatives of the function with respect to the coefficients, is required. These routines may have either PL/I compatible calling sequences or, if NOPL1<=0, FORTRAN compatible calling sequences (see Section IV.A.4).

1. PL/I ROUTINES

a. FCODE calling sequence:

```
CALL FCODE (Y,X,B,PRNT,F,I,RES,M,N,K);
```

M, N and K are problem parameters as defined in Section II.B.1. Y is the vector of length N containing the observed values of the dependent variable. X is the two-dimenstional (N,M) array containing the values of the M independent variables at the N points. B is a vector of length K containing the current coefficient values. PRNT is a vector containing any user defined output values, and is also useful as a means of passing values from FCODE to PCODE (see use of problem parameter NTABLE in Section IV, "Other Options Available").

FCODE should compute the (predicted) value of the function at the I^{th} data point, given X and B, and store the value in F. The residual should then be calculated and stored in RES (RES=Y(I)-F) subject to considerations of constraints and weights as discussed in Section IV, "Other Options Available". A sample subroutine is included in Section VI, "Program Test and Examples." Y, X, B and PRNT must be declared arrays with dimensions as follow:

```
DCL Y(*), X(*,*), (B(*), F, RES, PRNT(*))FLOAT(16);
```

Note: That X and Y are the only single precision real arguments; B, PRNT, F, RES being double precision.

b. PCODE calling sequence:

```
CALL PCODE (P,X,B,PRNT,F,I,M,N,K);
```

M, N and K are problem parameters as defined in Section II.B.1. X, B, PRNT and I are values just used in FCODE, and F is the function value just computed by FCODE for the I^{th} data point. P is the vector of length K to be used to store the values of the K partial derivatives of F with respect to b_j, j=1,2,...,K (where b_j is the j^{th} coefficient) computed at data point I. The arguments should be declared with dimensions as follows:

```
DCL X(*,*),(P(*),B(*),PRNT(*),F) FLOAT(16);
```

2. FORTRAN Routines

a. FCODE calling sequence:

```
CALL FCODE (Y,X,B,PRNT,F,I,RES,M,N,K)
```

M, N and K are problem parameters as defined in Section II.B.1. Y is the vector of length N containing the observed values of the dependent variable. X is the two-dimensional (,MN) array containing the values of the M independent variables at the N data points. B is the vector of length K containing the current coefficient values. PRNT is a vector containing any user defined output values, and is also a means of passing values from FCODE to PCODE (see use of problem parameter NTABLE in Section IV, "Other Options Available").

FCODE should compute the (predicted) value of the function at the I^{th} data point and store the value in F, given X and B. The residual should then be calculated and stored in RES (RES= Y(I)-F)subject to considerations of constraints and weights as discussed in Section IV, "Other Options Available". A sample routine is included in Section VI, "Program Tests and Examples". The variables and arrays should be declared as follows:

```
REAL*8 B(K),PRNT(1),F,RES,Y*4(N),X*4(M,N)
```

Note the order of subscripting on X: the first subscript is the independent variable index; the second subscript is the data point index. Also B,PRNT,F,RES are double precision, where as X and Y are single precision.

b. PCODE calling sequence:

```
CALL PCODE (P,X,B,PRNT,F,I,M,N,K)
```

M, N and K are problem parameters as defined in Section II.B.1. X, B, PRNT and I are values just used in FCODE, and F is the function value just computed in FCODE for the I^{th} data point. P is the vector of length K to be used to store the values of the K partial derivatives of F with respect to b_j, j=1,2,...,K (where b_j is the j^{th} coefficient) computed at data point I. The variables and arrays should be declared as follows:

```
REAL*8 P(K), B(K), PRNT(1),F,X*4(M,N)
```

Note that P, B, PRNT and F are double precision.

III. ANALYSIS

The algorithm selects an optimized correction vector for the coefficients by interpolation between the vector obtained by the gradient method and that obtained by a Taylor series expansion truncated after the 1^{st} derivative. Iteration is applied to this vector according to the least-squares method of estimating parameters until one of the criteria listed below is met. For a detailed discussion of the algorithm, see D. W. Marquardt's article, "An Algorithm for Least Squares Estimation of Non-linear Parameters," J. Soc. Indust. Appl. Math., Vol II (1963), pp. 431-441.

A. Criteria for Problem Termination

1. Force Off: ITOFF iterations were reached. (PHI not minimized.)

2. Gamma-lambda test: LAMBDA becomes larger than 1.0, when gamma is greater than 90 degrees. This usually indicates the presence of very high correlations among the parameter estimates. In this situation the parameter corrections may be dependent upon very high rounding error and PHI is not necessarily minimized.

3. Epsilon Test: This test is passed when two successive approximations of each coefficient round to either:

   a. The same set of NUMDIGT significant digits, or
   b. The same set of NUMDIGT decimal digits.

   Which criterion is used depends on the magnitude of the coefficient being tested and a problem parameter, TAU. If the absolute value of the coefficient is greater than TAU, the relative criterion 'a' is used. When the absolute value of the coefficient is less than or equal to TAU criterion 'b', an absolute test, is made. Thus if TAU is equal to zero, its default value, criterion 'a', will always be used; however, if TAU is set larger than the largest values of the coefficients, an absolute test will always be made. The user can tell which criterion was used for each coefficient by comparing the magnitude of the coefficient to TAU.

4. Gamma-epsilon test: The epsilon test is satisfied, and at the same time the angle, gamma, between the negative gradient vector and the Taylor series vector is less than GAMCR. PHI is assumed to be minimized within rounding error.

5. User termination: During the execution of either FCODE or PCODE, the user may elect to terminate the solution of the current problem by executing the statement "CALL FEXIT" in FORTRAN, or "SIGNAL CONDITION (PEXIT);" in PL/I. NLIN then prints the message "*****USER EXIT", and proceeds to the next problem.

6. PCODE missing: If NOPCODE<=0 and the user fails to supply a PCODE subroutine, NLIN abends with User Completion Code 001.

7. Singular matrix: If a singular $A=P^TP$ matrix (where P is the vector of partial derivatives) is generated during iteration, NLIN prints the message '*****SINGULAR MATRIX', followed by the row and column indicies of the pivot and its value, and then proceeds to the next problem. When very high correlations exist among the parameters, the matrix A may become singular due to the rounding error alone. However, the program uses $(A+\lambda I)$ instead of A during iterations; thus the matrix is usually positive definite and the singularity will not occur until the actual A matrix is inverted at convergence.

B. Confidence Limits

Three types of confidence limits are output by this program:

1. Conventional one parameter confidence limits based on Student's t distribution (the T problem parameter) at the $(1-\alpha)$ level with (N-K+IP) degrees of freedom; these are minimum length intervals.

2. Support plane intervals for each individual parameter. These intervals are conservatively wide and are based on the F statistic (the FF problem parameter) with (K-IP) and (N-K+IP) degrees of freedom.

3. Non-linear confidence ellipsoids calculated using the F statistic.

For a discussion of the theory and usefulness of the various confidence intervals, see Exhibit B (Confidence Region Calculations) in the Appendix to the documentation of the SHARE distributed program, "Least Squares Estimation of Nonlinear Parameters" (SDA 3094-01), by D. W. Marquardt. Another source of information concerning the confidence intervals is Applied Regression Analysis, by N. R. Drapper and H. Smith (John Wiley & Sons, Inc., 1966). Chapter 10 of this book includes a summary of D. W. Marquardt's method of parameter estimation as used in this program, as well as discussion of confidence intervals for parameters in nonlinear models.

IV. OTHER OPTIONS AVAILABLE

A. Optional Problem Parameters

The following problem parameters may be used to enable the application of the program to a variety of specific uses. These parameters, if used, are included in the problem parameter input data described earlier. Arranged by type of application (input control, output control, confidence limits, language options, computation aids and constraints), these are:

1. Input control:

NOX — provides for the use of the previous values of the independent variables, as defined in the just completed problem. If NOX<=0, new values will be read from file DATA; if DATA is not defined, from file SYSIN. If NOX>=1, the X values from the previous problem will be used. Default is zero. NOX must be <=0, if either M or N is changed to a new value. NOX is set to zero after input for the first problem has been completed; value >0 is in error at this point. (See "Error Conditions and Responses," Section V.)

NOY — provides for use of the previous values of the dependent variable. If NOY <=0, new values of the dependent variable will be read from file DATA or from file SYSIN if DATA is not defined. If NOY>=1, the Y values from the previous problem will be used. Default is zero. NOY must be <=0 if N is changed to a new value. NOY is set to zero after input for the first problem has been completed. A value >0 is in error at this point. (See "Error Conditions and Responses.")

NOCOEFF — provides for the use of previously defined initial coefficient values. If NOCOEFF<=0, initial values of the coefficients will be read from file SYSIN; NOCOEFF must be <=0 for the first problem in a group, and is always set to zero before input for the second problem is begun. If NOCOEFF=1, the initial values of the coefficients will be the initial values used in the preceding problem. If NOCOEFF>=2, the initial values will be the final coefficients obtained in the preceding problem. Default is zero. NOCOEFF must be <=0 if K is changed.

NONUCOE — to be used in conjunction with constant (or omitted) coefficients; i.e., NONUCOE is meaningful only if IP>0. If NONUCOE<=0, the subscripts of the constant coefficients will be read from file SYSIN. If NONUCOE>=1, the omitted coefficients will be those used in the previous problem. NONUCOE must be <=0 if IP is changed. Default is zero.

NVY — the number of the Y (dependent) variable. This is zero (0) by default and if the X, Y values are to be read as described in Section II.B. If the data is to be read an observation at a time; i.e., in X, Y groups, then NVY should be set equal to the position of the Y variable within the observation. Thus if the data input is coded as:

$X_{11}\ X_{12}\ X_{13}\ Y_1\ X_{14}$

$X_{21}\ X_{22}\ X_{23}\ Y_2\ X_{24}$ etc.

then NVY=4; if coded as 'X_1,Y_1,X_2,Y_2,X_3,Y_3' where there is only 1 independent variable per observation, NVY=2. NVY must be in the range 1 to M+1 if this form of input is used.
Note: When NVY>0 for a given problem, NOX and NOY must be zero.
The data must still be in list-directed input form; that is, each element must be separated by a comma (followed by any number of blanks) or by one or more blanks. There must be (M+1)*N values.

2. Output Control

IFP — controls plotting on the printer. IFP<=0 indicates that no plot is to be produced on the printer. If IFP>0, a plot of observed and predicted values of the dependent variable at each data point is produced on the printer. The position of the plotted point is correct to within two percent of the range of values plotted. The plot is made one line per data point and does not consider the value of the independent variables. The predicted values and corresponding residuals are printed in the margin to the right of the plot (see discussion of problem parameter ITPRNT, below). Once a problem termination criterion has been met, the plot, preceded by a statement of the number of iterations completed and a list of approximate coefficient values, is inserted in the output stream preceding the usual output of problem results as described under Default Output in Section II. Symbols used in the plot are: 'P', predicted value of the dependent variable; '0', observed value; 'Y', observed and predicted values are assigned to the same plot position; and 'X', value is beyond the range of the plot. Default value of IFP is zero.

YMN — value of the dependent variable to be assigned to the left margin of the plot. Default value of YMN is zero.

SPRD — range of values (not largest value) of the dependent variable to be covered by the plot. Default value of SPRD is 100.

ITPRNT — number of iterations for which a detailed printout of computations is desired. This detailed printout includes: the number of completed iterations; the approximate coefficient values; either the plot (if IFP>0) or a table of observed, predicted, and residual values; the P^TP correlation matrix (where P is the vector or partial derivatives); the value of PHI (sum of squares of residuals); the standard error of PHI (SE=SQRT(PHI/(N-K+1P))); the value of LAMBDA (parameter to facilitate convergence); a message indicating whether analytic or partial derivatives were used; and a series of increments added to the coefficients, with corresponding values of PHI, LAMBDA, GAMMA (angle between negative gradient vector and Taylor series vector) and LENGTH of increment vector. Default value of ITPRNT is zero.

NOPRNT — intermediate printout suppressor. NOPRNT<=0 provides detailed printout for ITPRNT iterations; for subsequent iterations, output is as described under Default Output in Section II. NOPRNT>0 suppresses subsequent output after ITPRNT iterations until a problem termination criterion has been met. The output of the solution then follows, as described under "Default Output". Default value of NOPRNT is zero.

NTABLE — allows printing of values selected by the user, but not normally included in the output. These values are printed in up to four columns to the right of the table of observed, predicted, and residual values. NTABLE is the length of the vector PRNT, into which the user puts the values to be printed. These values may be defined in either FCODE or PCODE and stored in the argument vector. If more than four values are to be printed, the fifth will be placed under the first value on the following line, and similarly for the ninth, thirteenth, etc. These values are printed for each data point, and must thus be defined each time FCODE or PCODE is called, using current argument values. (Note that PCODE is called after FCODE, and only if NOPCODE<=0.) Note that PRNT values computed in FCODE are available for use in PCODE computations.

3. Confidence limits:

FF — variable ratio statistic with (K-IP,N+IP-K) degrees of freedom, used in confidence limits calculations. Default is 4. FF must be positive.

T — the two tailed $(1-\alpha)$ point of Student's t distribution with (N+IP-K) degrees of freedom, to be used in confidence limit calculations. Default is 2. T must be positive.

NOLIM — determines whether or not the nonlinear confidence limits are calculated and printed. NOLIM<=0 provides the calculations; NOLIM>=1 suppresses them. Default is zero. Printout includes the value of PHI-CRITICAL and confidence limits on each individual coefficient such that PHI is approximately equal to PHI-CRITICAL at these limits (all other coefficients are unchanged).

$$\text{PHI-CRITICAL}=\text{PHI}*\left(1+\frac{K-IP}{N+IP-K}\,FF\right)$$

4. Language options:

NOPL1 — provides for use of either Fortran compatible or PL/I compatible external procedures FCODE and PCODE. If NOPL1<=0, FCODE and PCODE will be called with PL/I compatible calling sequences. If NOPL1>=1, FCODE and PCODE will be called with FORTRAN compatible calling sequences. Default is zero.

5. Computation aids:

XLAMBDA — initial values of lambda, used to facilitate convergence. If XLAMBDA<=0, lambda is first used with a value of 0.01. Default is zero.

ZETA — singularity criterion for matrix inversion, used in the routine to invert the matrix of normal equations. Default is 1.0E-31. ZETA must be non-negative.

6. Constraints:

NCONS — number of constraints imposed on the model. These are introduced by means of additional residuals in FCODE. Once the residuals have been obtained for each data point, FCODE is called once more for each constraint applied, with I=N+c, where c=1, 2, 3, ... , NCONS. Thus, for I=N+c, FCODE should return the value of the c^{th} constraint residual. The constraint residual functions should be continuous in the vicinity of the bounds imposed. Default is zero constraints. The introduction of constraints in this way may not be satisfactory, as the path to convergence is affected. Hence, if use of

constraints is felt necessary, a test run should be made to determine whether or not satisfactory results can be obtained. Frequently, the need for these constraints can be eliminated by replacing a variable coefficient with a constant, plus (or minus) a positive function. For example, the coefficient B in the model Y=A*X+B*EXP(X) can be constrained to a value >=1 by letting $B=1+B_1**2$, or $B=1+EXP(B_1)$, and solving for B_1 rather than B.

B. Application Options:

1. Weighted residuals:

A weighted residual can be easily obtained since the residuals are defined in FCODE. For example, the user can compute a weighted residual by multiplying the unweighted residual by the square root of its corresponding weight factor.

2. Initial problem computation - NEW:

An external variable called NEW is set equal to one (1) at the initiation of each problem and is not otherwise used by the main program. PL/I users may obtain its value by declaring NEW BIN FIXED (30) EXTERNAL; FORTRAN users may obtain this value by referring to an INTEGER*4 variable in a named COMMON block called NEW: COMMON/NEW/Variable-Name. If this value is changed by the user, it will be reset to 1 in the main program when a new problem is initiated. See example below.

3. Iteration check -ICT:

This option allows the user to determine when the last calls to FCODE or PCODE are being made. This variable, ICT, is set by NLIN as follows:

ICT=0	if the iterations are continuing.
ICT=1	if termination of the iterations is due to Force Off.
ICT=2	if the gamma lambda test was passed.
ICT=3	if the epsilon test was passed.
ICT=4	if the gamma epsilon test was passed.

Thus, this variable can be used to determine when to output, to tape or disk the final parameter values or any other information that is passed to FCODE (or PCODE) from NLIN. See examples in Section C. Note however, that at any given iteration, FCODE (and PCODE) will be called once for each observation point so that any output will be done N times unless precautions are taken to output the information only once. If changed by the user, the value will be reset to zero at the beginning of the next problem.

USE: In PL/I procedures, the variable can be accessed by declaring ICT as FIXED BINARY (30) EXTERNAL. FORTRAN users may obtain the value by referring to an INTEGER*4 variable in a named COMMON Block called ICT; i.e., COMMON/ICT/variable.

FORTRAN USER NOTE: When using a FORTRAN subprogram, the user must close all files opened by him by use of the REWIND statement at the end of his last problem; which means he must know how many problems he has and keep track of how many are completed. This can cause some problems since when a terminal error occurs, FCODE will not be called. The end result will be the loss of some of the previous parameter estimates.

C. Use of Options and Inputting Other Types of Data

1. Use of NEW and ICT

```
FCODE:  PROCEDURE(Y,X,B,PRNT,F,I,RES,M,N,K);
   DCL X(*,*),Y(*),(B(*),PRNT(*),F,RES) FLOAT(16),
   (NEW,ICT) FIXED BIN (3Ø) EXTERNAL,
   PBL FIXED BIN INITIAL(Ø) STATIC;
   IF NEW¬=1 THEN GO TO FUNEVAL;
   GET EDIT (TRANS) (X(LØ),F(7,3));

TF:  DO II=1 TO N;
   Y(II)=Y(II)*TRANS+5.0;
   X(II,1)=EXP(X(II,1));
   END TF;
   NEW=Ø;
FUNEVAL:  (Code for function evaluation - calculate F,RES)
TERMIN:  IF I¬= N | ICT = 0 THEN GO TO RET;
   PBL=PBL+1;
   IF ICT>3 THEN DO;
   PUT FILE(CURPARM) EDIT (PBL,ICT,B(1),B(5))((2)F(3),(2)F(12,6));
   END;
RET:    RETURN;
END FCODE;
```

This example illustrates several uses of FCODE other than simply for function evaluation (see also Section C.2.b):

a. The variable NEW is used to control those operations that are done only when a new problem is being started. In this case, a variable is read from the standard system input device. This would be placed right after the initial coefficient values in the input deck (it could have been on another file.) The variable is then used to transform the dependent variable (I,M,N, and K should not be changed in either FCODE or PCODE). The first X variable is also transformed. These transformations are done for all observations on the first entry of FCODE for each problem.

b. PBL is initialized to zero and declared static. Thus it's value is only initialized once during the job. It is then incremented at the completion of every problem, that is, when I=N and ICT is greater than zero.

c. At the completion of a problem when FCODE is being entered for the last observation after conversion, the problem number, type of test passed, and two of the K parameters are written out to a user file 'CURPARM'. The user must include the proper JCL to define this file.

2. Other types of data decks.

If the data is not in list-directed format; i.e., has unused numbers, no blanks between numbers, or such, then several techniques can be used to enter the data.

a. If the data is all numeric and the variables to be used are separated from each other and the rest of the data by blanks or commas, then M can be specified as being the number of X variables plus the number of other data fields on the card. Only the appropriate variables are then used in FCODE.

b. If the data must be read with a format, then the variables can be read by the user routine, FCODE, by use of:

1. some dummy data, and
2. the NEW option, and
3. a formatted read in FCODE.

The dummy data is required because NLIN always reads data for the initial problem and should consist of (M+1)*N observations, such as punched zeros, separated by blanks or a comma. The NEW option is used so that the data can be read on the first entry to FCODE; the data is then read with format control for the first problem. Dummy data need not be entered for succeeding problems if NOY and NOX are set greater than zero (>0) for succeeding problems.

A FORTRAN example:

```
      SUBROUTINE FCODE(Y,X,B,PRNT,F,I,RES,M,N,K)
      READ*8 B(K),PRNT(1),F,RES,X*4(M,N),Y*4(N)
      COMMON/NEW/NPBL
      IF (NPBL.LT.1) GO TO 100
      DO 1 II = 1,N
    1 READ(1,10) (X(J,II),J=1,M),Y(II)
   10 FORMAT (10X, F10.5,6X,F5.3,10X,F10.1)
      NPBL=0
  100 F=B(1)*X(I,1)*(DEXP(-B(2)*(X(I,2))
      RES=Y(I)-F
      RETURN
      END
```

This example reads, M X variables and the associated Y variable at the beginning of each problem. Remember not to use the I parameter in the subroutine statement as the DO parameter. Transformations could have been performed after the data was read.

V. ERROR CONDITIONS AND RESPONSES:

All program error messages are preceded by five asterisks, and are printed by the left margin.

A. Screening Input Parameters

Fairly extensive testing of problem parameters is made after input. For the first problem, NOCOEFF, NOX, and NOY are tested; if any one or more of these is <=0, a message is printed indicating their value. The remaining problem data are then read, and execution proceeds. NOCOEFF, NOX, and NOY are set to zero after input is completed for the first problem regardless of their previous value.

Parameter GAMCR, the criterion angle used in the gamma-epsilon test, is adjusted to the range 0=<GAMCR=<180. If GAMCR has an absolute value >1.4742E6, it assumes its default value. A message describing the action is printed anytime the value is changed after input.

If N-K+IP<=0, a message is printed indicating that more data points are needed. Execution then proceeds to the next problem. The following parameter relations are tested (see Sections II.B.1 and IV.A):

0<DEL<1,	0<NUMDIGT<7,
0<ITOFF,	0<=ZETA,
0<FF,	0<T,
0<=NCONS,	0<SPRD.

An invalid relation causes the erroneous parameter value to be printed with a message indicating that the parameter assumes a default value.

B. User Routine Missing

If Routine PCODE is missing and NOPCODE<=0, NLIN abends with User Completion Code 001.

C. PL/I ON-Units

Action taken by ON-units for errors detected in PL/I routines is as follows:

1. ON UNDEFINED FILE: if file SYSPRINT or file SYSIN is not defined, execution is terminated with return code 1000.

2. ON ENDFILE: if an end-of-file condition arises during input, a message is printed and execution is terminated. If the condition arises during problem parameter input, the message indicates "*****END OF PROBLEM GROUP REACHED."; otherwise, the message indicates "****END OF FILE REACHED WHILE ATTEMPTING TO READ DATA FOR CURRENT PROBLEM."

3. ON NAME: if an unrecognizable variable name appears in the input stream in the problem parameter list, a message is printed indicating the contents of the field and stating that this field is being ignored. Execution continues.

4. ON CONVERSION: if a conversion error arises during input, a message is printed indicating the contents of the field. An attempt is then made to begin input for a new problem.

5. Any other error results in the printing of a system's message, followed by a program error message indicating the type of error or the PL/I ONCODE number. Processing then continues with the next problem in the group.

D. FORTRAN routines

Action taken as a result of an error arising in FORTRAN routines FCODE and PCODE is as follows:

If either a floating-point or fixed-point divide exception or an exponent overflow occurs, a program message is printed indicating both the type of error and the FORTRAN routine in which it occurred (either FCODE or PCODE). Underflow, fixed-point overflow, and significant exceptions are masked off during execution of the FORTRAN routines.

If any other error is detected, a systems message or a FORTRAN (IHC type error number in FORTRAN G) error message is printed, and the job is terminated abnormally. If a FORTRAN error message is printed, subsequent information (traceback) may be erroneous, as the FORTRAN environment is not completely initialized.

VI. PROGRAM TESTS AND EXAMPLES:

Several sample problems were run to test program performance. Most available options were used in one or another of the runs. A summary of the results of these tests is given in this section, as well as examples of FCODE, PCODE, and input data.

A. Description of models tested.

For the primary series of tests, the data points and coefficient values were taken from the SHARE program SDA-3094.01 (see SLUR, serial number U-67).

FORM of the function: Y=A*EXP(B*X)

1. FCODE and PCODE routines for this function:

   a. PL/I routines:

```
FCODE:  PROCEDURE (Y,X,B,PRNT,F,I,RES,M,N,K);
DCL Y(*),X(*,*),(B(*),PRNT(*),F,RES)FLOAT(16);
PRNT(2)=EXP(B(2)*X(I,1));
F=B(1)*PRNT(2);
RES=Y(I)-F;
PRNT(1)=X(I,1);
END FCODE;
```

```
PCODE:  PROCEDURE (P,X,B,PRNT,F,I,M,N,K);
DCL X(*,*),(B(*),PRNT(*),F,P(*))FLOAT(16);
P(1)=PRNT(2);
P(2)=P(1)*B(1)*X(I,1);
END PCODE;
```

b. FORTRAN routines:

```
      SUBROUTINE FCODE (Y,X,B,PRNT,F,I,RES,M,N,K)
      REAL*8 B(K),PRNT(2),F,RES,Y*4(N),X*4(M,N)
      PRNT(2)=DEXP(B(2)*X(1,I))
      F=B(1)*PRNT(2)
      RES=Y(I)-F
      PRNT(1)=X(1,I)
      RETURN
      END

      SUBROUTINE PCODE (P,X,B,PRNT,F,I,M,N,K)
      REAL*8 P(K),B(K),PRNT(2),F,X*4(M,N)
      P(1)=PRNT(2)
      P(2)=P(1)*B(1)*X(1,I)
      RETURN
      END
```

2. Input
   Example of input for a group of six problems using PL/I subroutines and analytic derivatives:

```
Card
 1    NOPCODE=0, NUMDIGT=6, IFP=1, SPRD=8.0, NTABLE=2, ID='PL/I TEST 1',
 2    N=14, K=2;
 3     .01  .05  .10  .20  .30  .40  .50  .60  .70  .80  .90 1.00 1.50 2.00
 4    1.00 1.05 1.13 1.20 1.35 1.50 1.62 1.80 2.00 2.40 2.50 2.70 4.50 7.00
 5    1.00 1.00
 6    NOCOEFF=1, IP=1, ITPRNT=2,NOX=1, NOY=1, ID='PL/1 TEST 2';
 7    1
 8    ID=' PL/1 TEST 3';
 9    2
 10   ID='PL/1 TEST 4'; IP=0,NOCOEFF=0;
 11   1.00 0.10
 12   ID='PL/1 TEST 5';
 13   2.00 2.00
 14   ID='PL/1 TEST 6'; NOX=0, NOY=0, NVY=1;
 15   1.00 0.01
 16   1.05 0.05 1.13 0.10
 17   1.20 0.20 1.35 0.30 1.50 0.40
 18   1.62 0.50 1.80 0.60 2.00 0.70 2.40 0.80 2.50 0.90 2.70 1.00 4.50 1.50
 19   7.00 2.00
 20   1.00 1.00
 21   N=0;
```

This input could have been put on about 8 cards, but is on 21 for clarity in seeing the order. Also the only necessary change to run this data with FORTRAN subroutines would be addition of the problem parameter NOPL1=1 on card 1 or between card 1 and card 2.

The final coefficients for problems 1, 4, 5, and 6 are approximately 1.03714 and 0.959716. For problem 2 they are 1.0 and 0.980914, and for problem 3 they should be close to 0.977976 and 1.0.

3. Output

The output produced by NLIN using the routines in Section VI.A.1.a and the first five input cards given in Section VI.A.2 is presented below.

PROBLEM NUMBER 1 ***** NCSU COMPUTING CENTER *****

*** NON LINEAR LEAST SQUARES FIT ***

USER IDENTIFICATION: PL/1 TEST 1

N = 14	NO. DATA PTS.	NTABLE = 2	NO. AUX. VALUES PRINTED.
K = 2	NO. COEFFS.	NOLIM = 0	NON-LIN. CONF. LIM. SUPPRESSOR.
IP = 0	NO. CONSTANT COEFFS.	NOPCODE= 0	USE OF EST. DERIV. INDICATOR.
M = 1	NO. INDEP. VARS.	NOX = 0	USE OF PREV. X-ARRAY INDICATOR.
IFP = 1	PLOTTER OPTION.	NOY = 0	USE OF PREV. Y VECTOR.
ITOFF = 50	MAX. ITERATIONS.	NOCOEFF= 0	USE OF A PREV. COEFF. VECTOR.
ITPRNT = 0	DETAIL PRINTOUT INDICATOR.	NOPL1 = 0	USE OF FORTRAN SUBROUTINES.
NOPRNT = 0	INTERM. PRINTOUT SUPPRESSOR.	NCONS = 0	NO. CONSTRAINTS.
NUMDIGI= 6	NO. SIG. DIGITS USED IN CONV. TEST.	NONUCOE= 0	USE OF PREV. CONST. COEFF. SUBSCR.
NVY = 0	NO. OF VARIABLE Y, OBSERVATIONWISE INPUT	TAU = 0.00000E+00	CONSTANT FOR CONVERGENCE.
YMN = 0.00000E+00	VALUE OF LEFT PLOT MARGIN.	GAMCR = 4.50000E+01	CRITERION ANGLE.
SPRD = 8.0000CE+00	WIDTH OF PLOT.	T = 2.00000E+00	STUDENT'S 'T'.
ZETA = 1.00000E-31	SINGULARITY CRITERION FOR MATRIX INV.	FF = 4.00000E+00	VARIANCE RATIO STATISTIC.
DEL = 1.00000E-05	MULTIPLIER FOR EST. DERIV.	XLAMBDA= 0.00000E+00	LAMBDA.

NO. ITERATIONS COMPLETED: 0

APPROX. VALUES OF COEFFS.: COEFF(1) = 1.00000E+00

COEFF(2) = 1.00000E+00

PHI	S E	LENGTH	GAMMA	LAMBDA	ANALYTIC PARTIALS USED
1.86849E-01	1.24783E-01	0.000E+00	-1.235E+42	1.000E-02	

NO. ITERATIONS COMPLETED: 1

APPROX. VALUES OF COEFFS.: COEFF(1) = 1.03639E+00

COEFF(2) = 9.60118E-01

PHI	S E	LENGTH	GAMMA	LAMBDA	ANALYTIC PARTIALS USED
6.39689E-02	7.30119E-02	7.742E-01	6.619E+01	1.000E-03	

NO. ITERATIONS COMPLETED: 2

APPROX. VALUES OF COEFFS.: COEFF(1) = 1.03714E+00

COEFF(2) = 9.59711E-01

PHI	S E	LENGTH	GAMMA	LAMBDA	ANALYTIC PARTIALS USED
6.39598E-02	7.30067E-02	9.999E-03	5.210E+01	1.000E-04	

NO. ITERATIONS COMPLETED: 3

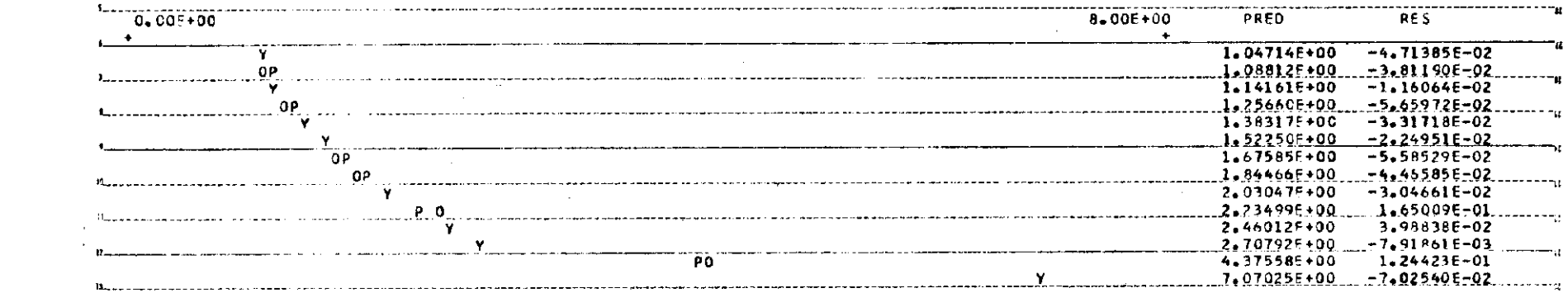
APPROX. VALUES OF COEFFS.: COEFF(1) = 1.03714E+00

COEFF(2) = 9.59716E-01

PHI	S E	LENGTH	GAMMA	LAMBDA	ANALYTIC PARTIALS USED
6.39598E-02	7.30067E-02	1.076E-04	5.214E+01	1.000E-05	

EPSILON TEST SATISFIED

NO. ITERATIONS COMPLETED: 3
APPROX. VALUES OF COEFFS.: COEFF(1) = 1.03714E+00
COEFF(2) = 9.59716E-01



PRED	RES
1.04714E+00	-4.71385E-02
1.08812E+00	-3.81190E-02
1.14161E+00	-1.16064E-02
1.25660E+00	-5.65972E-02
1.38317E+00	-3.31718E-02
1.52250E+00	-2.24951E-02
1.67585E+00	-5.58529E-02
1.84466E+00	-4.46585E-02
2.03047E+00	-3.04661E-02
2.23499E+00	1.65009E-01
2.46012E+00	3.98838E-02
2.70792E+00	-7.91861E-03
4.37558E+00	1.24423E-01
7.07025E+00	-7.02540E-02

OBS	PRED	DIFF		
1.00000E+00	1.04714E+00	-4.71385E-02	1.00000E-02	1.00964E+00
1.05000E+00	1.08812E+00	-3.81190E-02	5.00000E-02	1.04916E+00
1.13000E+00	1.14161E+00	-1.16064E-02	1.00000E-01	1.10073E+00
1.20000E+00	1.25660E+00	-5.65972E-02	2.00000E-01	1.21160E+00
1.35000E+00	1.38317E+00	-3.31718E-02	3.00000E-01	1.33364E+00
1.50000E+00	1.52250E+00	-2.24951E-02	4.00000E-01	1.46798E+00
1.62000E+00	1.67585E+00	-5.58529E-02	5.00000E-01	1.61585E+00
1.80000E+00	1.84466E+00	-4.46585E-02	6.00000E-01	1.77861E+00
2.00000E+00	2.03047E+00	-3.04661E-02	7.00000E-01	1.95776E+00
2.40000E+00	2.23499E+00	1.65009E-01	8.00000E-01	2.15496E+00
2.50000E+00	2.46012E+00	3.98838E-02	9.00000E-01	2.37203E+00
2.70000E+00	2.70792E+00	-7.91861E-03	1.00000E+00	2.61096E+00
4.50000E+00	4.37558E+00	1.24423E-01	1.50000E+00	4.21890E+00
7.00000E+00	7.07025E+00	-7.02540E-02	2.00000E+00	6.81709E+00

PHI	S E	LAMBDA	ANALYTIC PARTIALS USED
6.39598E-02	7.30067E-02	1.000E-05	

PTP INVERSE

1	6.22172E-02	-3.50815E-02
2	-3.50815E-02	2.35827E-02

PARAMETER CORRELATION MATRIX

1	1.0000E+00	-9.1585E-01
2	-9.1585E-01	1.0000E+00

	STD	ONE - PARAMETER		SUPPORT PLANE	
COEFF	ERROR	LOWER	UPPER	LOWER	UPPER
1	1.92103E-02	1.00072E+00	1.07356E+00	9.85630E-01	1.08864E+00
2	1.12114E-02	9.37293E-01	9.82139E-01	9.28006E-01	9.91427E-01

NONLINEAR CONFIDENCE LIMITS

PHI CRITICAL = 1.06600E-01

COEFF	LOWER B	LOWER PHI	UPPER B	UPPER PHI
1	1.01646E+00	1.06600E-01	1.05782E+00	1.06600E-01
2	9.46856E-01	1.06545E-01	9.72277E-01	1.06515E-01

FOR PROBLEM NO. 1 THE ELASPED CPU TIME WAS : 0.47 SECONDS.

VI. Test Results

Various combinations of problem parameters were used, and the results were quite satisfactory. Some details of interest are the following:

1. The use of analytic rather than estimated partials causes convergence to the same coefficient estimates (for NUMDIGT digits) with greater consistency for different starting values of the coefficients, as well as slightly reducing both the number of iterations required and the run time.

2. If estimated derivatives are used, the choice of DEL can significantly affect performance. The smaller the value of DEL, the better is the accuracy of the derivative estimates; however, if DEL is too small (less than 1.0E-15), all significant digits are lost in the computation of the derivative and the resulting derivative is zero. Recall that DEL is used in the derivative calculations as a multiplier. Thus, if a function is to be used for repeated runs with estimated derivatives, it could be helpful for the user to test various values of DEL to find an optimum value for his function.

3. Since FORTRAN and PL/I produce slightly different object codes, the use of one language will cause the path of convergence to be slightly different than it would be using the other to write FCODE and PCODE. ("Path of convergence" implies the successive coefficient approximations.) However, neither language was consistently preferable to the other, and the variations in the path of convergence became negligible when analytic partials were used.

4. Execution time (GO-time):
On an IBM 370/165 the 6 problems, totaling 21 iterations, ran in a total of 14.0 and 9.3 seconds using the FORTRAN G and PL/I(F) subroutines respectively. With the FORTRAN routines, the CPU time was 8.6 seconds and the I/O time was 5.4 seconds. PL/I(F) had a CPU time of 4.1 seconds and an I/O time of 5.2 seconds.

C. Other Models Tested:

1. The model

$$Y=(a/x)+(b/x^2)$$

was fit to 40 sets of 10 data points, totaling 200 iterations, in 19.2 seconds of CPU time and 17.2 seconds of I/O time, PL/I(F) routines and analytical derivatives.

2. The sample data of Marquardt (SHARE SDA 3094.01) was also fit with this program. The answers were the same as he obtained with his program. The equation was fit in 7 iterations totaling 5.4 seconds (3.5 CPU seconds; 1.9 I/O seconds).

VII. ACKNOWLEDGEMENTS

This program is a rewrite and modification of SHARE SDA 3094.01 (NLIN) by D. W. Marquardt. The initial coding, debugging, and preparation of the write-up of the original version was performed by Joseph A. Middleton, formerly of the Computing Center staff. This revision is by Richard A. Usanis.

J. H. Fulton of the NCSU Computing Center wrote the PL/I - FORTRAN interface subroutine and suggested some of the enhancements in this version.

VIII. APPENDIX

A. Program Information

SOURCE LANGUAGE: The main program is written in PL/I and was compiled and tested under Version 5.2C of the PL/I(F) compiler.

CORE REQUIREMENTS: For most applications, necessary core is under 135K. For a more accurate estimate, use the following relation: 114 K + space for FCODE and PCODE (typically about 3K) + array space. Array space, upper limit, in terms of the problem parameters, is

```
(N*(M+1))*4+(K*(12+2*K)+IP+2*NTABLE+NCONS)*8.
```

If FCODE and PCODE are coded in FORTRAN, this estimate should be increased by about 20K; this additional storage is required for FORTRAN error handling and I/O routines.

B. Parameter Default Values - Summary

Following is a table of all problem parameters, their default values and their meaning:

Parameter	Default Value	Meaning (in parentheses: meaning of default value)
DEL	1.0E-5	Multiplier for estimated derivatives
FF	4.	Variance ratio statistic
GAMCR	45.	Gamma criterion
ID	' '	User identification of problem
IFP	0	Plot option (suppressed)
ITOFF	50	Max. no. iterations
IP	0	No. constant coefficients
ITPRNT	0	No. iterations of detailed printout
K	1	No. coefficients
M	1	No. independent variables
N	0	No. data points (end of problem group)
NCONS	0	No. of constraint residuals
NOCOEFF	0	Initial coefficient estimates (will be read)
NOLIM	0	Nonlinear confidence limits (not suppressed)
NONUCOE	0	If IP>0, where constant coefficient subscripts are obtained from (will be read)
NOPCODE	1	Estimated or analytic partials (estimated partials used)
NOPL1	0	PL/I or FORTRAN calling sequence (PL/I will be used)
NOPRNT	0	Intermediate printout suppressor (not suppressed)
NOX	0	Independent variable data (will be read)
NOY	0	Dependent variable data (will be read)
NTABLE	0	No. auxiliary output values - length of PRNT array
NUMDIGT	4	No. digits used in conv. test
NVY	0	Number of variable Y when reading one observation at a time (read all X values, then all Y values)
SPRD	100.	Range of plot
T	2.	Student's t
TAU	0.	Constant for convergence
XLAMBDA	0.	Lambda (will be corrected to 0.01)
YMN	0.	Left margin value for plotting
ZETA	1.0E-31	Singularity criterion for matrix inversion

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina
27709 USA

SPLA CONTROL NUMBER:

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the "SHARE Reference Manual".

(1) Program Number (to be filled in by SPLA)..... 360D-13.2.003

(2) System Type (machine)..... 360/370

(3) Search Key..... Non-linear, parameters, regression, estimation, least-squares

(4) Programming Systems/Languages..... PL/I, optional ALP modules

(5) Author's Name and Address..... R. A. Usanis, J. A. Middleton

(6) Direct Technical Inquiries to Name & Address (if different than Author) J. H. Fulton (919) 737-2518
Computing Center, Box 5445
N. C. State University
Raleigh, North Carolina 27607

(7) Title of Program..... NLIN: Least-Squares Estimation of Non-Linear Parameters

(8) Submitter's Installation Membership Code..... NCS

(9) Submitter's Own Program Identification and Suffix(Optional).. NLIN3 2

(10) Primary Subject Code..... 13

(11) Minimum System Requirements OS PL/I-F or Optimizing Compilers

(12) New or Revision Code (if revision, show prior Program Number in Item 1) R

(13) Year Completed..... 1976

(14) Date of Submittal..... June 23, 1976

(15) Documentation (number of original pages submitted)..... 43

(16) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

Revised 4/74

SHARE PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide:

- **a. Purpose**
- **b. Programming Language used**
- **c. Version and modification level or release number**
- **d. Field of application**
- **e. Type of routine (main program, subroutine, etc.)**
- **f. Specific description of machine requirements**

ABSTRACT

NLIN is a PL/I main program which fits the model y=f(X,B) to the set of observations (Y_i, X_i) i=1(1)N using the maximum neighborhood method developed by D. W. Marquardt. X is a vector of independent variables and B is a vector of parameters estimated to minimize the sum of squares of (Y-y).

Any number of problems can be processed in one run. Required partial derivatives can be estimated or defined in a user supplied routine. Options are provided to control the detail of printed results; to omit parameters; to obtain nonlinear confidence limits; to allow use of values from a previous problem; to use either PL/I or FORTRAN external routines; and to constrain selected parameters.

Storage required is problem dependent with small problems running in 114K.

The current release is Version 3.2 containing several changes in source code which correct problems encountered when implementing earlier Versions with the PL/I Optimizing Compiler.

(Please attach additional pages if necessary).....Total pages attached 0

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program."

(17) Signature of Submitter and Date John W. Fulton 6-23-76

(18) Signature of Installation Addressee [illegible] 6-23-76

Programmer's Guide
for
NLIN
Nonlinear least squares estimation of parameters

by

Richard A. Usanis
Computing Center
North Carolina State University
May, 1974

Contents

I. Introduction

NLIN is a PL/I version of D. W. Marquardt's main program titled 'Least Squares Estimation of Nonlinear Parameters' [3] from which much of this document has been exerpted. The original program was written in FORTRAN IV. It was converted to PL/I by J. A. Middleton of the Computing Center at North Carolina State University, Raleigh, North Carolina and submitted to the SHARE Program Library [4].

Since the original version in PL/I, several versions have been passed. The present version (Version 3) has considerably greater input-output flexibility and input data screening than earlier versions. It is in double precision.

A. The Method

NLIN is a main program to fit the model

$$Y=f\ (X_1,X_2,\ldots,X_m;b_1,b_2,\ldots,b_k)$$

where Y_i is the value predicted by the model (f) for the ith observation after estimation of the parameters (coefficients, b's) from the n sets of observed values:

$$(y_i,x_{i,1},x_{i,2},\ldots,x_{i,\ell},\ldots,x_{i,m}) \qquad i=1,2,\ldots,n,$$

where y_i is the observed value of the dependent variable and the $x_{i,\ell}$ are the observed values of the m independent variables at the ith data point.

The program computes the least squares estimates of the b coefficients using the maximum neighbor method; i.e., the program minimizes

$$PHI=\sum_{i=1}^{n}\ (y_i-Y_i)^2$$

by adjusting the b_j.

A general outline of the algorithm is as follows:

Let $\underline{A}^{[k \times k]} = \underline{P}^T\underline{P}$

$\underline{P}^{[n \times k]}$ has elements

$$p_{i,j} = \frac{\partial Y_i}{\partial b_j} \qquad i=1,\ldots,n;\ j=1,\ldots,k$$

$$\underline{g}^{[k \times 1]} = \left(\sum_{i=1}^{n} (y_i - Y_i)\ \frac{\partial Y_i}{\partial b_j}\right) \quad j=1,\ldots,k$$

where the underlining designates an array.

Now $\underline{A}$ and $\underline{g}$ are scaled so that

$$\underline{A}' = (a'_{ij}) = \left(\frac{a_{ij}}{\sqrt{a_{ii}}\ \sqrt{a_{jj}}}\right), \text{ and}$$

$$\underline{g}' = (g'_i) = \left(g_i \,/\, \sqrt{a_{ii}}\right) .$$

At the rth iteration construct

$$(\underline{A}'^{r} + \lambda^r \underline{I})\underline{d}'^{r} = \underline{g}'^{r} ,$$

where the superscript r denotes the iteration, and solve for d^r where $\underline{d}^r = \left(d'^r_i / \sqrt{a_{ii}}\right)$. The vector $\underline{d}$ is a direction vector and shows the general direction of the minimum. Now a vector $\underline{b}^s = \underline{b}^r + \underline{d}^r$, where the b_i are the parameters to be estimated and s=r+1, can be obtained.

The λ^r are selected so that

$$PHI^s < PHI^r \qquad (1)$$

until PHI is minimized. Large values of λ are used only when necessary to satisfy (1), in which case a steepest descent method of solution is approached. This method provides for rapid progress at the beginning of a problem but gives a slower later progress. When equation (1) can be satisifed by the use of a small λ the method of solution approaches an unmodified Taylor series which converges rapidly when the maximum neighborhood can be adequately represented by a linear function. Thus Marquardt's method combines both steepest-descent and Taylor's linearization method. Marquart also found that in most cases the angle between the direction vector

produced by the steepest-descent method and that given by Taylor's series method was between 80° and 90° because of elongation of the error surface.

Marquardt's (2) strategy of choosing λ is, in general, where ϕ=PHI above:

"Let v>1.

Letλ^q denote the value of λ from the previous (q=r-1) iteration

Initially let $\lambda^0=10^{-2}$, say.

Compute $\phi(\lambda^q)$ and $\phi(\lambda^q/v)$.

i. If $\phi(\lambda^q/v)\leq\phi^r$, let $\lambda^r=\lambda^q/v$.

ii. If $\phi(\lambda^q/v)>\phi^r$, and $\phi(\lambda^q)\leq\phi^r$, let $\lambda^r=\lambda^q$.

iii. If $\phi(\lambda^q/v)>\phi^r$, and $\phi(\lambda^q)>\phi^r$ increase λ by successive multiplication by v until for some smallest w, $\phi(\lambda^q v^w)\leq\phi^r$. Let $\lambda^r=\lambda^q v^w$.

When intraparameter correlations are very high, λ can become extremely large, in which case, test iii can be altered as explained in Marquardt's paper.

Convergence has occurred when one of the tests in section I.B. are met.

The user must supply a routine named FCODE to evaluate Y_i for each $x_{i,\ell}$ and b_j combination. In addition, the user may optionally supply a routine called PCODE to evaluate the partial derivatives of Y with respect to b_j for j=1,...,k. These routines may be either PL/I subprocedures or FORTRAN subroutines.

If PCODE (analytic derivatives) is not supplied, then NLIN will estimate the derivatives by finite difference approximations. The user should be aware that using estimated derivatives usually increases the computation time since there will be (k+1) evaluations of the function

for the k parameters to obtain the finite difference expressions; i.e.,

$$\frac{\partial Y_i}{\partial b_j} \simeq \frac{f(x_{i,1},\ldots,b_j+\Delta,\ldots,b_k)-f(x_{i,1},\ldots,b_k)}{\Delta}$$

where $\Delta=\begin{cases}\text{DEL} & \text{if } b_j=0\\ \text{DEL}*b_j & \text{otherwise}\end{cases}$

and DEL is specified by the user or takes the default value 10^{-5}. When the model is insensitive to one or more of its parameters either a large value of DEL or analytical derivatives may be needed to obtain non-zero increments.

B. Convergence Criteria

The program may take one of three routes to convergence.

1. Epsilon test - This test is passed when, for all j,

   $|\Delta b_j| <= \text{CRIT}$

   where CRIT = 10.E0**-NUMDIGT if $|b_j| <= \text{TAU}$

   10.E0**-NUMDIGT*$|b_j|$ if $|b_j| > \text{TAU}$

   and Δb_j is the change in b_j from the last iteration.

2. Gamma Epsilon Test - This test is passed when all parameters have passed the epsilon test and in addition GAMMA<GAMCR, where GAMMA is the angle between the direction given by the gradient method and that given by Taylor's series method. The sum of squares (SS) should be minimized within rounding error at this point.

3. Gamma-Lambda Test - This test is passed when LAMDBA>1.0 and GAMMA>90° and indicates that the parameter estimates are highly dependent on rounding error, probably due to high correlations among the estimates of the parameters. The SS are not necessarily minimized.

C. Confidence Limits

NLIN calculates three types of confidence limits based on the problem parameters: T (student's t at $(1-\alpha)$ level) and FF (the F value at $(1-\alpha)$ level). All assume that the errors are normally and independently distributed random errors. Let $SE=(PHI/(n-k))^{1/2}$ with PHI being the minimum sums of squares; and let $\underline{C}=\underline{A}^{-1}$.

1. Single parameter confidence limits are computed as:

$$CL=b_j \pm (t_{1-\alpha,(n-k)})*SE*\sqrt{c_{jj}}$$

where t is the two tailed $(1-\alpha)$ point with $(n-k)$ degrees of freedom from Student's t distribution and b_j is the final parameter estimate. These limits provide a minimum length interval by assuming the other parameters are equal to the populational values. Note that the elements of C can be used to express the extent of the correlations (r_{ij}) among the parameter estimates as

$$\underline{R}=(r_{ij})=\left(\frac{c_{ij}}{\sqrt{c_{ii}}\ \sqrt{c_{jj}}}\right)$$

This matrix points out the parameters in a model which are highly correlated and can be used for evaluating the experimental design.

2. Support plane confidence limits are computed as

$$SP = b_j \pm (kF_{1-\alpha(k,n-k)}\ SE^2\ C_{jj})^{1/2}$$

where F is the variance ratio statistic at the upper $(1-\alpha)$ point of the distribution for k and $(n-k)$ degrees of freedom. These intervals are conservatively wide and considered by Marquardt (3) " to be the most realistic portrayal (within the applicability of the linear theory) of the precision of the parameter estimates individually."

3. Nonlinear confidence limits are also computed; however, since the model is nonlinear, the representation by the linear terms in a Taylor's

expansion is only an approximation in the vicinity of the minimum. This approximation will be inadequate outside some region around the minimum and thus the nonlinear confidence ellipsoids based on linear theory will not always be calculatable. When this occurs NLIN notes that the limit cannot be found in the area around the minimum.

Marquardt's (3) calculation of these limits can be briefly described as follows:

If the model is correct and the parameter estimates vary from their true values only because of random errors, then

$$\frac{(PHI_c - PHI)/k}{(PHI/(n-k))} \quad \text{is distrbuted as } F_{1-\alpha(n,n-k)}$$

and confidence limits can be obtained by trial and error for each parameter individually by varying PHI_c until the ratio is less than or equal to the F value. The parameters are varied one at a time. This procedure will thus give approximate (approximate in the sense of the α level) nonlinear confidence limits for the parameters.

II. Modules for NLIN

Letters in parentheses indicate the identification that is coded in columns 73-76 of the record for the source modules.

1. NLIN (NLIN) consists of two basic modules. Four sets of sample routines and test data are also available. The main program and a FORTRAN-PL/I link routine are the basic modules. The user supplies the FCODE (and optionally PCODE) routine, and the problem input data.

The REGION requirement for NLIN can be calculated approximately as the sum of the following core requirements:

1. 85 K for NLIN load module including the FORTRAN-PL/I link routine
2. 10K for transient requirements.

3. fK for the FCODE-PCODE routines. f will generally be less than 4
4. bK for buffers. b will generally be less than 16, and is the amount of core required for buffers.
5. 10K for automatic storage plus cK for controlled storage

where: $c=(4n(m+1)+16k(k+6)+8(IP+2\cdot NTABLE+NCONS))/1024$

k is the no. of parameters being estimated
n is the no. of observations
m is the no. of independent variables
NTABLE is the no. of auxillary values
NCONS is the no. of constraint residuals
IP is the no. of constant parameters

2. The FORTRAN-PL/I link routine (FTLK) is only necessary if FORTRAN is used to write FCODE (and optionally PCODE). In the main program the link routine is called from card numbers NLIN0399,0400,0411, and 0412. The entry points are declared in card numbers NLIN0009 to NLIN0012. This routine requires about 220 bytes of storage.

3. A third module (PLFP) is a PL/I source routine for testing NLIN. The model is Y=A*EXP(B*X). It contains FCODE and PCODE separated by a '*PROCESS;' card and can be compiled from disk or tape.

4. A fourth module (PLDA) is the data for the model in the third module. The correct results are:

	COEFFICIENTS	
PROBLEMS	A	B
1,4,5,6	1.03714	0.959716
2	1.0	0.980914
3	0.977976	1.0

5. This module (FOFP) is the same model as in 3, but written in FORTRAN and can be compiled directly by FORTRAN G or H.

6. Data (FODA) for module 5 which should give the same results as from module 4.

7. A PL/I test program (SKFP) using the model $y=a/x + b/x^c$. Both FCODE and PCODE are included and separated by a '*PROCESS;' card so that they can be compiled directly.

8. Data (SKDA) appropriate for the model in module 7. The estimated parameter values should be approximately:

	COEFFICIENTS		
PROBLEM	A	B	C
1	7.9982E+01	2.00066E+01	2.50075E-01
2-40	4.98125E+01	5.00812E+01	2.50331E-01

The second problem is run 40 times for timing comparisons in testing PL/I compilers.

9. FORTRAN subroutines (MQFP) FCODE and PCODE using Marquardt's test model, Y=A*EXP(B*X)+C*EXP(D*X). The subroutines are directly compilable.

10. Data (MQDA) for Marquardt's test model. The approximate parameter estimates are:

    5.90710E+00, -2.02158E-01, 2.49243E+00, -5.48951-02

III. How to put NLIN on disk in load module form.

Example Job Control Language to store NLIN on disk in load module form is given below. Procedure ASFC assembles the PL/I-FORTRAN link routine (FTLK) and stores the resulting object module in a temporary data set. Procedure PLOCL compiles the PL/I main program (NLIN), adds the resulting object module to the temporary data set and executes the Linkage-Editor which processes the combined object modules as primary input. The Assembler F-level and PL/I Optimizing compilers are used in this example, and the storage device is a 3330 disk drive.

```
     (TIME used here is CPU time plus I/O EXCP time, on a S/360/165.)
//jobname  JOB  account,programmername,TIME=(3,15),PAGES=65,M=1
//STP1  EXEC  ASFC
//C.SYSIN  DD  define data set containing FORTRAN-PL/I interface source routine
//STP2  EXEC PLOCL,PARM.C='OPTIMIZE(TIME)',REGION.C=300K
//C.SYSIN  DD  define data set containing PL/I source code for NLIN
//L.SYSLMOD  DD  define data set to contain NLIN Load Module, UNIT=DISK,
// DISP=(NEW,CATLG),SPACE=(TRK,(25,,1),RLSE),VOL=SER=NCSON1
//L.SYSIN  DD  *
   LIBRARY (FCODE)
/*
//
```

IV. How to use the NLIN load module

A. With PL/I FCODE, PCODE routines

```
//jobname  JOB  account,programmername
//A  EXEC  PLOCLG,PARM.C='OPTIMIZE(TIME)',REGION.G=125K
//*  PL/I OPTIMIZER COMPILE,LINK-EDIT AND EXECUTE
//C.SYSIN  DD  define dataset containing:
                FCODE
               *PROCESS;  } if analytic derivatives are used.
                PCODE     }
//L.MYLIB  DD  DISP=SHR,data set name of load module library
//L.SYSIN  DD *
       INCLUDE MYLIB(programname)
       ENTRY PLISTART
/*
//G.SYSIN  DD  define data set containing problem data
//
```

B. With FORTRAN FCODE, PCODE subroutines

```
//jobname  JOB  account,programmername
//A  EXEC  FTGCLG,REGION.G=130K
//*  FORTRAN-G COMPILE, LINK-EDIT AND EXECUTE
//C.SYSIN  DD  define data set containing:
         SUBROUTINE FCODE
         SUBROUTINE PCODE - if analytic derivatives are used.
//L.MYLIB  DD  DISP=SHR,data set name of load module library
//L.SYSIN  DD  *
       INCLUDE MYLIB(programname)
       ENTRY PLISTART
/*
//G.SYSPRINT  DD  SYSOUT=A
//G.SYSIN DD  define data set containing problem data
//
```

The example JCL assumes the use of the PL/I Optimizing compiler and the FORTRAN IV G-level compiler. The PL/I F-level and FORTRAN IV H-level compilers can also be used.

V. REFERENCES

1. DRAPER, and SMITH. 1966. Applied Regression Analysis. Wiley and Sons. N. Y.
2. Marquardt D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Indust. Appl. Math: p. 431-441.
3. Marquardt D. W. 1966. Least-squares estimation of nonlinear parameters. IBM SHARE Program Library No. SDA 3094.01.
4. Middleton, J. A. 1969. Least-squares estimation of non-linear parameters - NLIN. IBM SHARE Program Library No. 360D-13.2.003.

Appendix A

Magnetic Tape Key

The tape volume contains 10 files and 11 tape marks (TM) as shown below. The DCB information for all files can be summarized by saying:

(RECFM=FB,LRECL=80,BLKSIZE=1680)

All source programs have an alphabetic code in card columns 73-76 and are sequence numbered in card columns 77-80. The data decks contain no identification. The files are arranged as follows:

File 1 PL/I Source Deck (NLIN main program)
EBCDIC
NLIN in cc 73-76; SEQ. NO. IN CC 77-80
1079 card images
TM

File 2 Assembler (OS/360) subroutine FORTLNK
EBCDIC
FTLK in cc 73-76. all cards; sequence no. 0001 thru 0062 in cc 77-80.
62 card images
TM

File 3 PL/I test problem source routines FCODE, PCODE separated with a '*PROCESS;' card.
EBCDIC
PLFP in 73-76, all cards; sequence no. 0001 thru 0015 in cc 77-80.
15 card images
TM

File 4 Test problem data for PL/I routines
EBCDIC
No identification
21 card images
TM

File 5 FORTRAN test problem source subroutines FCODE,PCODE.
EBCDIC
FOFP in cc 73-76, all cards; sequence no. 0001 thru 0014 in cc 77-80;
14 card images
TM

File 6 Test problem data for FORTRAN subroutines.
EBCDIC
No identification
21 card images
TM

File 7 Shrikhande's model; PL/I problem source routines
FCODE,PCODE separated with a '*PROCESS;' card.
EBCDIC
SKFP in cc 73-76, all cards; sequence no. 0001 thru 0027
in cc 77-80;
27 card images
TM

File 8 Shrikhande's model test data
EBCDIC
No identification
10 card images
TM

File 9 Marquardt's model; FORTRAN problem source subroutines
FCODE,PCODE
EBCDIC
MQFP in cc 73-76, all cards; sequence no. 0001 thru 0024
in cc 77-80;
24 card images
TM

File 10 Marquardt's model test data
EBCDIC
No identification
8 card images
TM
TM

August, 1972

Library Services Series
Document No. LSR-089-1

NLIN -- NONLINEAR LEAST SQUARES ESTIMATION OF PARAMETERS

Documentation Prepared
by
R. A. Usanis
Computing Center
North Carolina State University
Raleigh, North Carolina 27607

SUPPORT TYPE: C

Triangle Universities Computation Center
Research Triangle Park, North Carolina 27709

Changes in this version of NLIN

Major changes in this version of NLIN include:

1. Increased input data checking;
2. Inclusion of a problem parameter (NVY) that allows the data to be read an observation at a time;
3. Changing all floating point variables except the input observations to double precision; and
4. More detail and examples in the manual.

The third change requires some changes to the user's subroutines FCODE and PCODE. In particular the DECLARE (PL/I) or REAL (FORTRAN) statements must reflect the precision of the variables being passed from NLIN. Thus these statements must be as follows:

A. For PL/I
   1. In FCODE:

```
DCL Y(*),X(*,*),(B(*),PRNT(*),F,RES)FLOAT(16);
```

   2. In PCODE:

```
DCL X(*,*),(P(*),B(*),PRNT(*),F)FLOAT(16);
```

B. For FORTRAN
   1. In FCODE:

```
REAL*8 B(K),PRNT(1),F,RES,Y*4(N),X*4(M,N)
```

   2. In PCODE:

```
REAL*8 P(K),B(K),PRNT(1),F,X*4(M,N)
```

TABLE OF CONTENTS