

USER'S GUIDE
TO
STARPAC

THE STANDARDS TIME SERIES AND REGRESSION PACKAGE

National Institute of Standards and Technology
(formerly the National Bureau of Standards)
Internal Report NBSIR 86-3448
(text revised 3/15/90)
(STARPAC sample output revised 03/15/90)

Janet R. Donaldson
Peter V. Tryon

U.S. Department of Commerce
Center for Computing and Applied Mathematics
National Institute of Standards and Technology
Boulder, Colorado 80303-3328

This online document contains only the portions of the above-referenced publication that are relevant to OTIS-P. Parties interested in STARPAC and related statistical software should contact:

Janet E. Rogers
National Institute of Standards and Technology
Mail Code 719
325 Broadway
Boulder, CO 80303-3328
Internet: jrd@cam.nist.gov

Assistance provided by Janet Rogers and the National Institute of Standards and Technology is gratefully acknowledged.

Disclaimer

No warranties, express or implied, are made by the distributors or developers that STARPAC or its constituent parts are free of error. They should not be relied upon as the sole basis for solving a problem whose incorrect solution could result in injury to person or property. If the programs are employed in such a manner, it is at the user's own risk and the distributors and developers disclaim all liability for such misuse.

Computers have been identified in this paper in order to adequately specify the sample programs and test results. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

Preface

STARPAC, the Standards Time Series and Regression Package, is a library of Fortran subroutines for statistical data analysis developed by the Statistical Engineering Division (SED) of the National Institute of Standards and Technology (NIST), formerly the National Bureau of Standards (NBS), Boulder, Colorado. Earlier versions of this library were distributed by the SED under the name STATLIB [Tryon and Donaldson, 1978]. Chapter 1 and chapter 9 of this document were previously distributed as NBS Technical Notes 1068-1 and 1068-2, respectively [Donaldson and Tryon, 1983a and 1983b]. STARPAC incorporates many changes to STATLIB, including additional statistical techniques, improved algorithms and enhanced portability.

STARPAC consists of families of subroutines for nonlinear least squares regression, time series analysis (in both time and frequency domains), line printer graphics, basic statistical analysis, and linear least squares regression. These subroutines feature:

- * ease of use, alone and with other Fortran subroutine libraries;
- * extensive error handling facilities;
- * comprehensive printed reports;
- * no problem size restrictions other than effective machine size; and
- * portability.

Notation, format and naming conventions are constant throughout the STARPAC documentation, allowing the documentation for each family of subroutines to be used alone or in conjunction with the documentation for another.

STARPAC is written in ANSI Fortran 77 [American National Standards Institute, 1977]. Workspace and machine-dependent constants are supplied using subroutines based on the Bell Laboratories "Framework for a Portable Library" [Fox et al., 1978a]. We have also used subroutines from LINPACK [Dongarra et al., 1979], from the "Basic Linear Algebra Subprograms for Fortran Usage" [Lawson et al., 1979], from DATAPAC [Filliben, 1977] and from the portable special function subroutines of Fullerton [1977]. The analyses generated by several of the subroutine families have been adapted from OMNITAB II [Hogben et al., 1971]; users are directed to Peavy et al. [1985] for information about OMNITAB 80, the current version of OMNITAB.

Computer facilities for the STARPAC project have been provided in part by the National Oceanic and Atmospheric Administration Mountain Administrative Support Center Computer Division, Boulder, Colorado, and we gratefully acknowledge their support. The STARPAC subroutine library is the result of the programming efforts of Janet R. Donaldson and John E. Koontz, with assistance from Ginger A. Caldwell, Steven M. Keefer, and Linda L. Mitchell. Valuable contributions have also been made by each of the members of the Statistical Engineering Division in Boulder, and from many within the STARPAC user community. We are grateful for the many valuable comments that we have received on early drafts of the STARPAC documentation; we wish especially to thank Paul T. Boggs, Ginger A. Caldwell, Sally E. Howe, John E. Koontz, James T. Ringland, Ralph J. Slutz, and Dominic F. Vecchia. Finally, we wish to thank Lorna Buhse for excellent manuscript support.

Janet R. Donaldson
Peter V. Tryon (deceased)
October 1985
Revised September 1987
Revised February 1990

Contents

Preface

1. INTRODUCTION TO USING STARPAC
 - A. Overview of STARPAC and Its Contents
 - B. Documentation Conventions
 - C. A Sample Program
 - D. Using STARPAC
 - D.1 The PROGRAM Statement
 - D.2 The Dimension Statements
 - D.3 The CALL Statements
 - D.4 STARPAC Output
 - D.5 STARPAC Error Handling
 - D.6 Common Programming Errors When Using STARPAC
2. LINE PRINTER GRAPHICS
3. NORMAL RANDOM NUMBER GENERATION
4. HISTOGRAMS
5. STATISTICAL ANALYSIS OF A UNIVARIATE SAMPLE
6. ONE-WAY ANALYSIS OF VARIANCE
7. CORRELATION ANALYSIS
8. LINEAR LEAST SQUARES
9. NONLINEAR LEAST SQUARES

- A. Introduction
- B. Subroutine Descriptions
 - B.1 Nonlinear Least Squares Estimation Subroutines
 - B.2 Derivative Step Size Selection Subroutines
 - B.3 Derivative Checking Subroutines
- C. Subroutine Declaration and CALL Statements
- D. Dictionary of Subroutine Arguments and COMMON Variables
- E. Computational Methods
 - E.1 Algorithms
 - E.1.a Nonlinear Least Squares Estimation
 - E.1.b Derivative Step Size Selection
 - E.1.c Derivative Checking
 - E.2 Computed Results and Printed Output
 - E.2.a The Nonlinear Least Squares Estimation Subroutines
 - E.2.b The Derivative Step Size Selection Subroutines
 - E.2.c The Derivative Checking Subroutines
- F. Examples
- G. Acknowledgments
- 10. DIGITAL FILTERING
- 11. COMPLEX DEMODULATION
- 12. CORRELATION AND SPECTRUM ANALYSIS
- 13. ARIMA MODELING
- Appendix A. CONTINUITY OF VERTICAL PLOTS ON THE CDC CYBER 840 AND 855
- Appendix B. WEIGHTED LEAST SQUARES
- Appendix C. ESTIMATING THE NUMBER OF RELIABLE DIGITS IN THE RESULTS OF A FUNCTION
- Appendix D. LIST OF STARPAC SUBPROGRAM NAMES
 - D.1 Subprograms specifically written for STARPAC
 - D.2 Subprograms from NL2SOL
 - D.3 Subprograms from miscellaneous public domain sources
 - D.4 Subprograms from LINPACK and BLAS
 - D.5 Subprograms specifying machine dependent constants
- Appendix E. LIST OF STARPAC LABELED COMMON NAMES
- References.

The Standards Time Series and Regression Package

Janet R. Donaldson and Peter V. Tryon

U.S. Department of Commerce
Center for Computing and Applied Mathematics
National Institute of Standards and Technology
Boulder, Colorado 80303-3328

STARPAC, the Standards Time Series and Regression Package, is a library of Fortran subroutines for statistical data analysis developed by the Statistical Engineering Division of the National Institute of Standards and Technology (formerly the National Bureau of Standards), Boulder, Colorado. Earlier versions of this library were distributed by the SED under the name STATLIB [Tryon and Donaldson, 1978]. STARPAC incorporates many changes to STATLIB, including additional statistical techniques, improved algorithms and enhanced portability.

STARPAC emphasizes the statistical interpretation of results, and, for this reason, comprehensive printed reports of auxiliary statistical information, often in graphical form, are automatically provided to augment the basic statistical computations performed by each user-callable STARPAC subroutine. STARPAC thus provides the best features of many stand-alone statistical software programs within the flexible environment of a subroutine library.

Key words: data analysis; nonlinear least squares; STARPAC; statistical computing; statistical subroutine library; statistics; STATLIB; time series analysis.

CHAPTER 1

INTRODUCTION TO USING STARPAC

A. Overview of STARPAC and Its Contents

STARPAC is a portable library of approximately 150 user-callable ANSI 77 Fortran subroutines for statistical data analysis. Designed primarily for time series analysis and for nonlinear least squares regression, STARPAC also includes subroutines for normal random number generation, line printer plots, basic statistical analyses and linear least squares. Emphasis has been placed on facilitating the interpretation of statistical analyses, and, for this reason, comprehensive printed reports of auxiliary statistical information, often in graphical form, are automatically provided to augment the basic statistical computations performed by each user-callable STARPAC subroutine. STARPAC thus provides the best features of many stand-alone statistical software programs within the flexible environment of a subroutine library.

STARPAC is designed to be easy to use; in many situations, only a few lines of elementary Fortran code are required for the users' main programs. A fundamental STARPAC philosophy is to provide two or more user-callable subroutines for each method of analysis: one which minimizes the complexity of the CALL statement, automatically producing a comprehensive printed report of the results; and one or more others which provide user control of the computations, allow suppression of all or part of the printed reports, and/or provide storage of computed results for further analyses.

STARPAC was developed and is maintained by the Center for Computing and Applied Mathematics of the National Institute of Standards and Technology (NIST), Boulder, Colorado. Users' comments and suggestions, which have had significant impact already, are highly valued and always welcomed. Comments and suggestions should be directed to:

Janet R. Donaldson
NIST Center for Computing and Applied Mathematics
Mail Code 719
325 Broadway
Boulder, CO 80303-3328.

B. Documentation Conventions

The documentation for the various STARPAC subroutine families uses a standard format description of the information needed to use a STARPAC subroutine, including one or more examples.

References to chapter sections within the STARPAC documentation refer to the identified section within the current chapter unless explicitly stated otherwise. Figures are identified by the section in which they occur. For example, figure B-1 refers to the first figure in section B of this chapter (chapter 1).

Names of INTEGER and REAL STARPAC subroutine arguments are consistent with the implicit Fortran convention for specifying variable type. That is, variable names beginning with I through N are type INTEGER while all others are type REAL unless otherwise explicitly typed DOUBLE PRECISION or COMPLEX. All dimensioned variables are explicitly declared in STARPAC documentation by means of INTEGER, REAL, DOUBLE PRECISION, or COMPLEX statements, as appropriate. The convention used to specify the dimension statements is discussed below in section D.2.

The precision of the STARPAC library is indicated in the printed reports generated by STARPAC: an S following the STARPAC version number in the output heading indicates the single precision version is being used, while a D indicates the double precision version. The STARPAC documentation is designed for use with both single and double precision versions. Subroutine arguments which are double precision in both versions are declared DOUBLE PRECISION; similarly, arguments which are single precision in both versions are declared REAL. Arguments whose precision is dependent upon whether the single or double precision version of STARPAC is being used are declared <real>. If the double precision version of the STARPAC library is being used, then the user should substitute DOUBLE PRECISION for <real>; if the single precision version is being used, then the user should substitute REAL for <real>. Other precision-dependent features are discussed as they occur.

C. A Sample Program

The sample program shown below illustrates the use of STARPAC subroutines. The code shown is portable ANSI 77 Fortran. Section D below uses this example to discuss Fortran programming as it relates to STARPAC.

The data used in this example are 84 relative humidity measurements taken at Pikes Peak, Colorado. STARPAC subroutine PP, documented in chapter 2, plots the data versus time-order and STARPAC subroutine STAT, documented in chapter 5, prints a comprehensive statistical analysis of the data.

Program:

```
PROGRAM EXAMPL
C
C DEMONSTRATE STAT AND PP USING SINGLE PRECISION VERSION OF STARPAC
C
C N.B. DECLARATION OF Y AND X MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL Y(100), X(100)
DOUBLE PRECISION DSTAK(100)
C
COMMON /CSTAK/ DSTAK
COMMON /ERRCHK/ IERR
C
C SET UP INPUT AND OUTPUT FILES
C [CHAPTER 1, SECTION D.4, DESCRIBES HOW TO CHANGE OUTPUT UNIT.]
C
CALL IPRINT(IPRT)
OPEN (UNIT=IPRT, FILE='FILENM')
OPEN (UNIT=5, FILE='DATA')
C
C DEFINE LDSTAK, THE LENGTH OF DSTAK
C
LDSTAK = 100
C
C READ NUMBER OF OBSERVATIONS INTO N AND
C DATA INTO VECTOR Y
C
READ (5,100) N
READ (5,101) (Y(I), I=1,N)
C
C CREATE A VECTOR OF ORDER INDICES IN X
C
DO 10 I=1,N
X(I) = I
10 CONTINUE
C
C PRINT TITLE, PLOT OF DATA AND ERROR INDICATOR
C
WRITE (IPRT,102)
CALL PP (Y, X, N)
WRITE (IPRT,103) IERR
C
```

```

C     PRINT TITLE, STATISTICAL ANALYSIS OF DATA AND ERROR INDICATOR
C
C     WRITE (IPRT,102)
C     CALL STAT (Y, N, LDSTAK)
C     WRITE (IPRT,103) IERR
C
C     FORMAT STATEMENTS
C
C     100 FORMAT (I5)
C     101 FORMAT (11F7.4)
C     102 FORMAT ('1DAVIS-HARRISON PIKES PEAK RELATIVE HUMIDITY DATA')
C     103 FORMAT (' IERR = ', I1)
C     END

```

<1-3>

Data:

```

      84
0.6067 0.6087 0.6086 0.6134 0.6108 0.6138 0.6125 0.6122 0.6110 0.6104 0.7213
0.7078 0.7021 0.7004 0.6981 0.7242 0.7268 0.7418 0.7407 0.7199 0.6225 0.6254
0.6252 0.6267 0.6218 0.6178 0.6216 0.6192 0.6191 0.6250 0.6188 0.6233 0.6225
0.6204 0.6207 0.6168 0.6141 0.6291 0.6231 0.6222 0.6252 0.6308 0.6376 0.6330
0.6303 0.6301 0.6390 0.6423 0.6300 0.6260 0.6292 0.6298 0.6290 0.6262 0.5952
0.5951 0.6314 0.6440 0.6439 0.6326 0.6392 0.6417 0.6412 0.6530 0.6411 0.6355
0.6344 0.6623 0.6276 0.6307 0.6354 0.6197 0.6153 0.6340 0.6338 0.6284 0.6162
0.6252 0.6349 0.6344 0.6361 0.6373 0.6337 0.6383

```

D. Using STARPAC

The following subsections provide general information needed when using STARPAC, including a discussion of Fortran programming as it relates to STARPAC usage. Although only elementary knowledge of Fortran is required to use STARPAC, users may still have to consult with a Fortran text and/or their Computing Center staff when questions arise.

D.1 The PROGRAM Statement

The PROGRAM statement is used to name the user's main program. The name EXAMPL is assigned to the main program in this example. The program name cannot be the name of any variable in the user's main program and, in addition, cannot be the name of any other subroutine or function called during execution of the user's code. Specifically, it cannot be the name of any subroutine within STARPAC. To ensure that the name of a STARPAC subroutine is not inadvertently chosen for the name of the main program, users should consult with the local installer of STARPAC to obtain a list of the STARPAC subroutine names.

D.2 The Dimension Statements

The user's program must include dimension statements to define the sizes and types of the vectors, matrices and three-dimensional arrays required by

each STARPAC subroutine used; STARPAC itself has no inherent upper limit on problem size.

Within the STARPAC documentation for the subroutine declaration and CALL statements, lowercase identifiers in the dimension statements represent integer constants which must equal or exceed the value of the identically-spelled uppercase argument. For example, if the documentation specifies the minimum dimension of a variable as <real> XM(n,m), and if the number of observations N is 15, and the number of columns of data M is 3, then (assuming the single precision version of STARPAC is being used) the minimum array size is given by the dimension statement REAL XM(15,3).

The exact dimensions assigned to some vectors and matrices must be supplied in the CALL statements to some STARPAC subroutines. For example, the argument IXM is defined as "the exact value of the first dimension of the matrix XM as declared in the calling program." Continuing the example from the preceding paragraph, if the statement REAL XM(20,5) is used to dimension the matrix XM for a particular subroutine, and IXM is an argument in the CALL statement, then IXM must have the value 20 regardless of the value assigned to the variable N.

Many STARPAC subroutines require a work area for internal computations. This work area is provided by the DOUBLE PRECISION vector DSTAK. The rules for defining DSTAK are as follows.

1. Programs which call subroutines requiring the work vector DSTAK must include the statements

```
DOUBLE PRECISION DSTAK (ldstak)
COMMON /CSTAK/ DSTAK
```

where ldstak indicates the integer constant used to dimension DSTAK.

2. Since all STARPAC subroutines use the same work vector, the length of DSTAK must equal or exceed the longest length required by any of the individual STARPAC subroutines called by the user's program.
3. The length, LDSTAK, of the work vector DSTAK must be specified in the CALL statement of any STARPAC subroutine using DSTAK to enable STARPAC to verify that there will be sufficient work area for the problem.

It is recommended that a variable LDSTAK be set to the length of DSTAK, and that this variable be used in each CALL statement requiring the length of DSTAK to be specified. Then, if a future modification to the user's program requires the length of DSTAK to be changed, the only alterations required in the existing code would be to the DOUBLE PRECISION dimension statement and to the statement which assigns the length of DSTAK to LDSTAK.

STARPAC manages its work area using subroutines modeled after those in ACM Algorithm 528: Framework for a Portable Library [Fox et al. 1978a]. Although STARPAC and the Framework share the same COMMON for their work areas, there are differences between the STARPAC management subroutines and those of the Framework. In particular, the STARPAC management subroutines re-initialize DSTAK each time the user invokes a STARPAC subroutine requiring work area, destroying all data previously stored in DSTAK; the Framework only initializes DSTAK the first time any of its management subroutines are

invoked, preserving work area allocations still in use. Thus, users must be cautious when utilizing STARPAC with other libraries which employ the Framework, such as PORT [Fox et al., 1978b].

The sample program shown in figure C-1 provides an example of the use of dimensioned variables with STARPAC. The REAL vector Y, used by both subroutines PP and STAT, contains the 84 relative humidity measurements; its minimum length, N (the number of observations), is 84. The REAL vector X used by subroutine PP contains the corresponding time order indices of the data; its minimum length is also 84. The DOUBLE PRECISION vector DSTAK contains the work area needed by STAT for intermediate computations; its minimum length, 49 in this case, is defined in section D of chapter 4. In this example, the dimensions of Y, X, and DSTAK, are each 100, exceeding the required minimum values.

D.3 The CALL Statements

The STARPAC CALL statement arguments provide the interface for specifying the data to be used, controlling the computations, and providing space for any returned results. The CALL statements used in the example (fig. C-1) are CALL PP(Y, X, N) and CALL STAT(Y, N, LDSTAK). Note that scalar arguments may be specified either by a variable preset to the desired value, as was done in the example, or by the actual numerical values. For example, CALL PP(Y, X, 84) and CALL STAT(Y, 84, 100) could have been used instead of the forms shown. We recommend using variables rather than the actual numerical values in order to simplify future changes in the program. When variables are used, changes need to be made in only one place; numerical values have to be changed every place they occur. The use of variables can also clarify the meaning of the program.

D.4 STARPAC Output

Most STARPAC subroutines produce extensive printed reports, freeing the user from formatting and printing all statistics of interest. The standard output device is used for these reports. The user has the options of titling the reports and changing the output device.

The first page of the report from each STARPAC subroutine does not start on a new page. This allows the user to supply titles. For example,

```
WRITE (6, 100)
100 FORMAT ('1DAVIS-HARRISON PIKES PEAK RELATIVE HUMIDITY DATA')
CALL PP (Y, X, N)
```

will print the title DAVIS-HARRISON PIKES PEAK RELATIVE HUMIDITY DATA on the top line of a new page, immediately preceding the output produced by the call to subroutine PP. Users should note that titles more than one line in length can cause a printed report designed for one page to extend beyond the bottom of the page.

The unit number, IPRT, of the output device used by STARPAC is returned by STARPAC subroutine IPRINT. Users can change the output device unit number by including with their program a subroutine IPRINT which will supersede the STARPAC subroutine of the same name. The subroutine must have the form

```
SUBROUTINE IPRINT(IPRT)
  IPRT = u
  RETURN
END
```

where `u` is an integer value specifying the output unit to which all STARPAC output will be written.

D.5 STARPAC Error Handling

STARPAC provides extensive error-checking facilities which include both printed reports and a program-accessible error flag variable. There are essentially two types of errors STARPAC can detect.

The first type of error involves incorrect problem specification, i.e., one or more of the input arguments in the subroutine statement has an improper value. For example, the number of observations, `N`, might have an obviously meaningless non-positive value. In the case of improper problem specification STARPAC generates a printed report identifying the subroutine involved, the error detected, and the proper form of the subroutine `CALL` statement. The latter is provided because improper input is often the result of an incorrectly specified subroutine argument list.

A second type of error can be thought of as a computation error: either the initiated calculation cannot be completed or the results from the called subroutine are questionable. For example, when the least squares model and data are found to be singular, the desired computations cannot be completed; when one or more of the standardized residuals from a least squares fit cannot be computed because the standard deviation of the residual is zero, the results of the error estimates from the least squares regression may be questionable. If a computation error is detected, STARPAC generates a report which identifies the error, and, to aid the user in determining the cause of

<1-9>

the error, summarizes the completed results in a printed report.

STARPAC error reports cannot be suppressed, even when the normal output from the STARPAC subroutine has been suppressed. (STARPAC output must be directed to a separate output device [see section D.4] when users do not want any STARPAC reports displayed under any conditions.) Because of this, users seldom have to consciously handle STARPAC error conditions in their code.

When proper execution of the user's program depends on knowing whether or not an error has been detected, the error flag can be examined from within the user's code. When access to the error flag is desired, the statement

```
COMMON /ERRCHK/ IERR
```

must be placed with the Fortran declaration statements in the user's program. Following the execution of a STARPAC subroutine, the variable `IERR` will be set to zero if no errors were detected, and to a nonzero value otherwise; the value of `IERR` may indicate the type of error [e.g., see chapter 9, section D, argument `IERR`]. If the `CALL` statement is followed with a statement of the form

```
IF (IERR .NE. 0) STOP
```

then the program will stop when an error is detected. (In figure C-1, the value of IERR is printed following each CALL statement to show the value returned.)

D.6 Common Programming Errors When Using STARPAC

STARPAC error-checking procedures catch many programming errors and print informative diagnostics when such errors are detected. However, there are some errors which STARPAC cannot detect. The more common of these are discussed below.

1. The most common error involves array dimensions which are too small. Although certain arguments are checked by STARPAC to verify that array dimensions are adequate, if incorrect information is supplied to STARPAC, or if the dimension of an array which is not checked is too small, the program will produce erroneous results and/or will stop prematurely. Users should check the dimension statements in their program whenever difficulties are encountered in using STARPAC.
2. The second most common error involves incorrect CALL statements, that is, CALL statements in which the STARPAC subroutine name is misspelled, the arguments are incorrectly ordered, one or more arguments are omitted, or the argument types (INTEGER, REAL, DOUBLE PRECISION, and COMPLEX) are incorrect. Users having problems using STARPAC should carefully check their declaration and CALL statements to verify that they agree with the documentation.
3. The third most common error involves incorrect specification of the work vector DSTAK. Programs which call STARPAC subroutines requiring work area must include both the DOUBLE PRECISION statement dimension DSTAK and the COMMON /CSTAK/ DSTAK statement.
4. The final common error involves user-supplied subroutines which have the same name as a subroutine in the STARPAC library. Users should

<1-10>

consult with the local installer of STARPAC to obtain a list of all STARPAC subroutine names. This list can then be used to ensure that a STARPAC subroutine name has not been duplicated.

Users who have not found the cause of a problem after checking the possibilities mentioned above should consult with their Computing Center advisers.

CHAPTER 9

NONLINEAR LEAST SQUARES

A. Introduction

STARPAC contains 16 user-callable subroutines for nonlinear least squares regression. Twelve of these are estimation subroutines that compute the least

squares solution as described below, performing either weighted or unweighted regression with either numerically approximated or user-supplied (analytic) derivatives. The estimation subroutines allow three levels of control of the computations and printed output, and allow the user to specify a subset of the parameters to be treated as constants, with their values held fixed at their input values. This last feature allows the user to examine the results obtained estimating various subsets of the parameters of a general model without rewriting the model subroutine for each subset. The other four subroutines described in this chapter are utility procedures which choose optimum step sizes for numerically approximating the derivative and which verify the correctness of user-supplied (analytic) derivatives.

Each of the subroutines described in this chapter assumes that the observations of the dependent variable, $y(i)$, are modeled by

$$y(i) = f(x(i),PAR) + e(i) \quad \text{for } i = 1, \dots, N,$$

where

N is the number of observations;

f is the function (nonlinear in its parameters) that models the i th observation;

$x(i)$ is the vector of the M independent variables at the i th observation;

PAR is the vector of the $NPAR$ model parameters; and

$e(i)$ is the unobservable random error in the i th observation, which is estimated by the i th residual.

The least squares estimates of the parameters, PAR , are obtained using an iterative procedure that requires the matrix of partial derivatives of the model with respect to each parameter,

$$D(i,k) = \text{partial} [f(x(i),PAR) \text{ wrt } PAR(k)]$$

for $i = 1, \dots, N$ and $k = 1, \dots, NPAR$.

The derivative matrix may be supplied analytically or approximated numerically.

The least squares solution is that which minimizes (with respect to PAR) the residual sum of squares function,

$$RSS(PAR) = \sum_{i=1}^N wt(i) * (y(i) - f(x(i),PAR)) ** 2 = \sum_{i=1}^N wt(i) * e(i) ** 2$$

<9-1>

here

$wt(i)$ is the weight assigned to the i th observation ($wt(i) = 1.0$ in the 'unweighted' case). Appendix B discusses several common applications for weighted least squares.

The user must supply both initial values for the parameters and the subroutine NLSMDL (described in section D) used to compute $f(x(i), PAR)$, $i = 1, \dots, N$, i.e., the predicted values of the dependent variable given the independent variables and the parameter values from each iteration. Initial parameter values should be chosen with care, since good values can significantly reduce computing time.

STARPAC provides a variety of subroutines to accommodate many levels of user sophistication and problem difficulty. Users are directed to section B for a brief description of the subroutines. The declaration and CALL statements are given in section C, and the subroutine arguments are defined in section D. The algorithms used and the output produced by these subroutines are discussed in section E. Sample programs and their output are shown in section F.

B. Subroutine Descriptions

B.1 Nonlinear Least Squares Estimation Subroutines

The simplest of the 12 nonlinear least squares estimation subroutines, NLS, requires neither user-supplied weights nor analytic derivatives. The estimated results and a variety of statistics are automatically summarized in a five-part printed report, and the estimated parameters and residuals are returned to the user via the subroutine argument list (level one control, described below). Most nonlinear least squares problems can be solved using NLS.

The other 11 estimation subroutines add the weighting, derivative and level two and three control features both singly and in combination, providing greater flexibility to the user at the price of less simplicity. These features are indicated by the suffix letter(s) on the subroutine name (e.g., NLSS and NLSWDC).

- * Suffix W indicates user-supplied weights.
- * Suffix D indicates user-supplied (analytic) derivatives.
- * Suffix C indicates level two control of the computations.
- * Suffix S indicates level three control of the computations.

The three levels of computation and printed output control are as follows.

- * In level one, a five-part printed report, discussed in detail in section E.2.a, is automatically provided and the estimated model parameters and residuals are returned to the user via the argument list.
- * Level two also returns the estimated parameters and residuals, and, in addition, allows the user to supply arguments to indicate

<9-2>

- a subset of the model parameters to be treated as constants, with their values held fixed at their input values;
- either the step sizes used to compute the numerical approximations

- to the derivative, or, when user-supplied analytic derivatives are used, whether they will be checked;
 - the maximum number of iterations allowed;
 - the convergence criteria;
 - the scale (i.e., the typical size) of each parameter;
 - the maximum change allowed in the parameters at the first iteration;
 - how the variance-covariance matrix is to be approximated; and
 - the amount of printed output desired.
- * Level three has all the features of level two, and, in addition returns the following estimated values via the argument list:
- the number of nonzero weighted observations (only when a weighted analysis is performed);
 - the number of parameters actually estimated;
 - the residual standard deviation;
 - the predicted values;
 - the standard deviations of the predicted values;
 - the standardized residuals; and
 - the variance-covariance matrix of the estimated parameters.

B.2 Derivative Step Size Selection Subroutines

When the partial derivatives used in the nonlinear least squares solution are not available analytically, STARPAC subroutines approximate them numerically. In this case, the subroutines can select optimum step sizes for approximating the derivatives [see section E.1.b]. The user also has the option of computing these step sizes independently of the estimation process by calling either of the two step size selection subroutines directly. For example, when planning to use the parameter fixing capability [argument IFIXED] to examine several subsets of the parameters of a general model, computing the step sizes first and passing them to the estimation subroutine is more efficient than recomputing them each time the estimation subroutine is called.

The simplest of the two user-callable step size selection subroutines, STPLS, summarizes the step size selection information for each parameter in a printed report and returns the step sizes to the user via the subroutine argument list.

The second step size selection subroutine, STPLSC, differs from STPLS only in that it enables the user to supply arguments to indicate

- the number of reliable digits in the model results;
- the number of exemptions allowed by the acceptance criteria, specified as a proportion of the total number of observations (see section E.1.b);
- the scale (i.e., the typical size) of each parameter; and
- the amount of printed output desired.

B.3 Derivative Checking Subroutines

When the partial derivatives used in the nonlinear least squares solution are available analytically, the user can code them for use by the estimation

subroutines [see section D, argument NLSDRV]. Because coding errors are a common problem with user-supplied derivatives, the STARPAC estimation subroutines automatically check the validity of the user-supplied derivative code by comparing its results to numerically approximated values for the derivative. When the results are questionable, the checking procedure attempts to determine whether the problem lies with the user's code or with the accuracy of the numerical approximation [see section E.1.c]. Although the checking procedure is automatically available to the estimation subroutines which accept user-supplied derivatives, the user may want to check the derivative code independently of the estimation process. In these cases, the user can call either of the two derivative checking subroutines directly, and suppress checking by the estimation subroutines [see section D, argument IDRVCK].

The simplest of the two derivative checking subroutines, DCKLS, summarizes the results of the check in a printed report.

The second of the derivative checking subroutine, DCKLSC, differs from DCKLS only in that it enables the user to supply arguments to indicate

- the number of reliable digits in the model results;
- the agreement tolerance;
- the scale (i.e., the typical size) of each parameter;
- the row at which the derivative is to be checked; and
- the amount of printed output desired.

C. Subroutine Declaration and CALL Statements

NOTE: Argument definitions and sample programs are given in sections D and F, respectively. The conventions used to present the following declaration and CALL statements are given in chapter 1, sections B and D.

Nonlinear Least Squares Estimation Subroutines

The <basic declaration block> identifies declaration statements that are needed by all of the nonlinear least squares estimation subroutines. The user should substitute the following four statements for each occurrence of <basic declaration block> given below.

```
<real> Y(n), XM(n,m), PAR(npar), RES(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
EXTERNAL NLSMDL
```

===

NLS: Compute and print a five-part weighted nonlinear least squares analysis with numerically approximated derivatives; return parameter estimates and residuals

```
<basic declaration block>
:
:
CALL NLS (Y, XM, N, M, IXM, NLSMDL,
+       PAR, NPAR, RES, LDSTAK)
```

===

<9-4>

NLSC: Compute and optionally print a five-part unweighted nonlinear least squares analysis with numerically approximated derivatives using user-supplied control values; return parameter estimates and residuals

```
<basic declaration block>
INTEGER IFIXED(npar)
<real> STP(npar), STOPSS, STOPP, SCALE(npar), DELTA
:
:
CALL NLSC (Y, XM, N, M, IXM, NLSMDL,
+         PAR, NPAR, RES, LDSTAK,
+         IFIXED, STP, MIT, STOPSS, STOPP,
+         SCALE, DELTA, IVAPRX, NPRT)
```

===

NLSS: Compute and optionally print a five-part unweighted nonlinear least squares analysis with numerically approximated derivatives using user-supplied control values; return parameter estimates, residuals, number of nonzero weights, number of parameters estimated, residual standard deviation, predicted values, standard deviations of the predicted values and variance-covariance matrix of the estimated parameters

```
<basic declaration block>
INTEGER IFIXED(npar)
<real> STP(npar), STOPSS, STOPP, SCALE(npar), DELTA
<real> RSD, PV(n), SDPV(n), SDRES(n), VCV(npars,npars)
:
:
CALL NLSS (Y, XM, N, M, IXM, NLSMDL,
+         PAR, NPAR, RES, LDSTAK,
+         IFIXED, STP, MIT, STOPSS, STOPP,
+         SCALE, DELTA, IVAPRX, NPRT,
+         NPARE, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

NLSW: Compute and print a five-part weighted nonlinear least squares analysis with numerically approximated derivatives; return parameter estimates and residuals

```
<basic declaration block>
<real> WT(n)
:
:
CALL NLSW (Y, WT, XM, N, M, IXM, NLSMDL,
+         PAR, NPAR, RES, LDSTAK)
```

NLSWC: Compute and optionally print a five-part weighted nonlinear least squares analysis with numerically approximated derivatives using

user-supplied control values; return parameter estimates and residuals

```
<basic declaration block>
INTEGER IFIXED(npar)
<real> WT(n)
<real> STP(npar), STOPSS, STOPP, SCALE(npar), DELTA
:
:
CALL NLSWC (Y, WT, XM, N, M, IXM, NLSMDL,
+          PAR, NPAR, RES, LDSTAK,
+          IFIXED, STP, MIT, STOPSS, STOPP,
+          SCALE, DELTA, IVAPRX, NPRT)
```

===

NLSWS: Compute and optionally print a five-part weighted nonlinear least squares analysis with numerically approximated derivatives using user-supplied control values; return parameter estimates, residuals, number of nonzero weights, number of parameters estimated, residual standard deviation, predicted values, standard deviations of the predicted values and variance-covariance matrix of the estimated parameters

```
<basic declaration block>
INTEGER IFIXED(npar)
<real> WT(n)
<real> STP(npar), STOPSS, STOPP, SCALE(npar), DELTA
<real> RSD, PV(n), SDPV(n), SDRES(n), VCV(npars,npars)
:
:
CALL NLSWS (Y, WT, XM, N, M, IXM, NLSMDL,
+          PAR, NPAR, RES, LDSTAK,
+          IFIXED, STP, MIT, STOPSS, STOPP,
+          SCALE, DELTA, IVAPRX, NPRT,
+          NNZW, NPARE, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

NLSD: Compute and print a five-part unweighted nonlinear least squares analysis with user-supplied derivatives; return parameter estimates and residuals

```
<basic declaration block>
EXTERNAL NLSDRV
:
:
CALL NLSD (Y, XM, N, M, IXM, NLSMDL, NLSDRV,
+          PAR, NPAR, RES, LDSTAK)
```

NLSDC: Compute and optionally print a five-part unweighted nonlinear least squares analysis with user-supplied derivatives using user-supplied control values; return parameter estimates and residuals

```
<basic declaration block>
EXTERNAL NLSDRV
INTEGER IFIXED(npar)
```

```

<real> STOPSS, STOPP, SCALE(npar), DELTA
:
:
CALL NLSDC (Y, XM, N, M, IXM, NLSMDL, NLSDRV,
+          PAR, NPAR, RES, LDSTAK,
+          IFIXED, IDRVCCK, MIT, STOPSS, STOPP,
+          SCALE, DELTA, IVAPRX, NPRT)

```

===

NLSDS: Compute and optionally print a five-part unweighted nonlinear least squares analysis with user-supplied derivatives using user-supplied control values; return parameter estimates, residuals, number of parameters estimated, residual standard deviation, predicted values, standard deviations of the predicted values and variance-covariance matrix of the estimated parameters

```

<basic declaration block>
EXTERNAL NLSDRV
INTEGER IFIXED(npar)
<real> STOPSS, STOPP, SCALE(npar), DELTA
<real> RSD, PV(n), SDPV(n), SDRES(n), VCV(npars,npars)
:
:
CALL NLSDS (Y, XM, N, M, IXM, NLSMDL, NLSDRV,
+          PAR, NPAR, RES, LDSTAK,
+          IFIXED, IDRVCCK, MIT, STOPSS, STOPP,
+          SCALE, DELTA, IVAPRX, NPRT,
+          NPARE, RSD, PV, SDPV, SDRES, VCV, IVCV)

```

===

NLSWD: Compute and print a five-part weighted nonlinear least squares analysis with user-supplied derivatives; return parameter estimates and residuals

```

<basic declaration block>
EXTERNAL NLSDRV
<real> WT(n)
:
:
CALL NLSWD (Y, WT, XM, N, M, IXM, NLSMDL, NLSDRV,
+          PAR, NPAR, RES, LDSTAK)

```

===

NLSWDC: Compute and optionally print a five-part weighted nonlinear least squares analysis with user-supplied derivatives using user-supplied control values; return parameter estimates and residuals

```

<basic declaration block>
EXTERNAL NLSDRV
INTEGER IFIXED(npar)
<real> WT(n)
<real> STOPSS, STOPP, SCALE(npar), DELTA
:
:

```

```

CALL NLSWDC (Y, WT, XM, N, M, IXM, NLSMDL, NLSDRV,
+           PAR, NPAR, RES, LDSTAK,
+           IFIXED, IDRVCCK, MIT, STOPSS, STOPP,
+           SCALE, DELTA, IVAPRX, NPRT)

```

===

NLSWDS: Compute and optionally print a five-part weighted nonlinear least squares analysis with user-supplied derivatives using user-supplied control values; return parameter estimates, residuals, number of nonzero weights, number of parameters estimated, residual standard deviation, predicted values, standard deviations of the predicted values and variance-covariance matrix of the estimated parameters

```

<basic declaration block>
EXTERNAL NLSDRV
INTEGER IFIXED(npar)
<real> WT(n)
<real> STOPSS, STOPP, SCALE(npar), DELTA
<real> RSD, PV(n), SDPV(n), SDRES(n), VCV(npars,npars)
:
:
CALL NLSWDS (Y, WT, XM, N, M, IXM, NLSMDL, NLSDRV,
+           PAR, NPAR, RES, LDSTAK,
+           IFIXED, IDRVCCK, MIT, STOPSS, STOPP,
+           SCALE, DELTA, IVAPRX, NPRT,
+           NNZW, NPARE, RSD, PV, SDPV, SDRES, VCV, IVCV)

```

===

Step Size Selection Subroutines

STPLS: Compute and print optimum step sizes for numerically approximating derivatives; return selected step sizes

```

<real> XM(n,m), PAR(npar), STP(npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
EXTERNAL NLSMDL
:
:
CALL STPLS (XM, N, M, IXM, NLSMDL, PAR, NPAR, LDSTAK, STP)

```

===

<9-8>

STPLSC: Compute and optionally print optimum step sizes for numerically approximating derivatives using user-supplied control values; return selected step sizes

```

<real> XM(n,m), PAR(npar), STP(npar)
<real> EXMPT, SCALE(npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
EXTERNAL NLSMDL

```

```

:
:
CALL STPLSC (XM, N, M, IXM, NLSMDL, PAR, NPAR, LDSTAK, STP,
+          NETA, EXMPT, SCALE, NPRT)

```

===

Derivative Checking Subroutines

DCKLS: Perform and print derivative checking analysis; return error code

```

<real> XM(n,m), PAR(npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
EXTERNAL NLSMDL, NLSDRV
:
:
CALL DCKLS (XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, LDSTAK)

```

===

DCKLSC: Perform and optionally print derivative checking analysis using user-supplied control values; return error code

```

<real> XM(n,m), PAR(npar)
<real> SCALE(npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
EXTERNAL NLSMDL, NLSDRV
:
:
CALL DCKLSC (XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, LDSTAK,
+          NETA, NTAU, SCALE, NROW, NPRT)

```

===

D. Dictionary of Subroutine Arguments and COMMON Variables

NOTE: --> indicates that the argument is input to the subroutine and that the input value is preserved;
 <-- indicates that the argument is returned by the subroutine;
 <-> indicates that the argument is input to the subroutine and that the input value is overwritten by the subroutine;
 --- indicates that the argument is input to some subroutines and is returned by others;
 *** indicates that the argument is a subroutine name;
 ... indicates that the variable is passed via COMMON.

D <-- The matrix of exact dimension N by NPAR that contains the partial derivatives of the model with respect to each parameter, PAR(k), k = 1, ..., NPAR. This argument is used within derivative subroutine NLSDRV [see argument NLSDRV below].

DELTA --> The maximum scaled change allowed in the parameters at the first

iteration [see section E.1.a]. The default value is 100.0. When DELTA <= 0.0 or when DELTA is not an argument of the subroutine CALL statement the default value is used. A smaller value of DELTA may be appropriate if, at the first iteration, the computation of the predicted values from the user's model subroutine produces an arithmetic overflow or the parameters leave the region of interest in parameter space. A reasonable alternative to the default value of DELTA is an upper bound to the scaled change that the estimated parameters should be allowed to make on the first iteration,

$$\text{DELTA} = \min(|\text{del}(\text{PAR}(k))|/\text{SCALE}(k), \text{ for } k = 1, \dots, \text{NPAR})$$

where del(PAR(k)) is the maximum change allowed for the kth parameter at the first iteration.

DSTAK ... The DOUBLE PRECISION vector in COMMON /CSTAK/ of dimension at least LDSTAK. DSTAK provides workspace for the computations. The first LDSTAK locations of DSTAK will be overwritten during subroutine execution.

EXMPT --> The proportion used to compute the number of observations, $a = \text{EXMPT} * N$, for which the forward difference quotient derivative with respect to a given parameter is exempted from meeting the acceptance criteria for step size selection [see section E.1.b]. The default value for EXMPT is 0.1 (10 percent). When the user-supplied value is outside the range [0.0, 1.0], or when EXMPT is not an argument of the subroutine CALL statement, the default value is used.

IDRVCK --> The indicator variable used to designate whether or not the user-supplied derivative subroutine is to be checked. When IDRVCK <> 0 the derivative is checked, and when IDRVCK = 0 it is not. The default value is IDRVCK <> 0. When IDRVCK is not an argument of the subroutine CALL statement the default value is used.

IERR ... An error flag returned in COMMON /ERRCHK/ [see chapter 1, section D.5]. Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided.

For the estimation subroutines:

IERR = 0 indicates that no errors were detected, and that the iterations converged satisfactorily.

IERR = 1 indicates that improper input was detected.

IERR = 2 indicates that the computation of the residual sum of squares using the initial parameter values produced an arithmetic overflow. The user should reduce the size of DELTA or should supply new starting values.

IERR = 3 indicates that the model is computationally singular, which means the model has too many parameters near the solution. The user should examine the model and data

to determine and remove the cause of the singularity.

IERR = 4 indicates that at least one of the standardized residuals could not be computed because its standard deviation was zero. The validity of the covariance matrix is questionable.

IERR = 5 indicates false convergence [see section E.1.a].

IERR = 6 indicates that convergence was not reached in the allowed number of iterations or model subroutine calls [see argument MIT].

IERR = 7 indicates that the variance-covariance matrix could not be computed.

For the step size selection subroutines:

IERR = 0 indicates that no errors were detected, and that all the step sizes satisfied the selection criteria.

IERR = 1 indicates that improper input was detected.

IERR = 2 indicates that one or more of the step sizes did not satisfy the selection criteria.

For the derivative checking subroutines:

IERR = 0 indicates that no errors were detected, and that the user-supplied derivative code appears to be correct.

IERR = 1 indicates that improper input was detected.

IERR = 2 indicates that the user-supplied derivative code and numerical derivatives do not agree for at least one parameter, but that in each case of disagreement the accuracy of the numerical derivatives is questionable. Further testing is suggested.

IERR = 3 indicates that the user-supplied derivative code and numerical derivatives do not agree for at least one parameter, and in at least one instance of disagreement there is no reason to doubt the numerical derivatives.

IFIXED --> The vector of dimension at least NPAR that contains values used to indicate whether the corresponding parameter in PAR is to be treated as a fixed constant or is to be estimated. If IFIXED(I) > 0, PAR(I) will be held fixed at its input value; if IFIXED(I) = 0, PAR(I) will be estimated using the least squares procedure described in section A. The default values are

<9-11>

IFIXED(I) = 0, I = 1, ..., NPAR, i.e., all parameters are estimated. When IFIXED(1) <= -1, or when IFIXED is not an argument of the subroutine CALL statement, the default value will be used.

IVCV --> The exact value of the first dimension of the matrix VCV as specified in the calling program.

IVAPRX --> The indicator variable used to specify how the variance-covariance matrix, VCV, is to be approximated. Three approximations are available:

$$(1) \text{ VCV} = \text{RSD}^{**2} * \text{inv}(\text{trans}(\text{Dhat}) * \text{W} * \text{Dhat})$$

$$(2) \text{ VCV} = \text{RSD}^{**2} * \text{inv}(\text{Hhat})$$

$$(3) \text{ VCV} = \text{RSD}^{**2} * \text{inv}(\text{Hhat}) * (\text{trans}(\text{Dhat}) * \text{W} * \text{Dhat}) * \text{inv}(\text{Hhat})$$

where

trans(.) indicates the transpose of the designated matrix;

inv(.) indicates the inverse of the designated matrix;

Hhat is the matrix of second partial derivatives of the model with respect to each parameter (the Hessian matrix), evaluated at the solution

$$\begin{aligned} &= \text{trans}(\text{Dhat}) * \text{W} * \text{Dhat} + \\ &\quad \sum_{i=1}^N \text{e}(i) * \text{wt}(i) * (\text{second partial of e}(i) \text{ wrt PAR}(j) \ \& \ \text{PAR}(k) \\ &\quad \quad \quad \text{for } j = 1, \dots, \text{NPAR} \ \& \ k = 1, \dots, \text{NPAR}); \end{aligned}$$

W is an N by N diagonal matrix of weights,

$$\text{W} = \text{diag}(\text{wt}(i), i = 1, \dots, N),$$

when a weighted analysis is performed, and is the identity matrix otherwise, and

Dhat is the matrix that contains the partial derivatives of the model with respect to each parameter (the Jacobian matrix), evaluated at the solution.

Approximation (1) is based on the assumption that H is approximately equal to trans(D)*W*D because the residuals are sufficiently small at the solution; approximation (2) is based on the assumption that the necessary conditions for asymptotic maximum likelihood theory have been met; and approximation (3) is based on the assumption that the necessary conditions for asymptotic maximum likelihood theory may be violated. The results of a study by Donaldson and Schnabel [1987] indicate that approximation (1) is preferable because it is simple, less expensive, more numerically stable and at least as accurate as approximations (2) and (3). However, all approximations to the variance-covariance matrix are subject to sampling variation because they are computed using the estimated parameter values.

<9-12>

The variance-covariance matrix computed for any particular nonlinear least squares solution should thus be regarded as only a

rough estimate [Bard, 1974; Donaldson and Schnabel, 1987].

If IVAPRX = 1 or 4 then approximation (1) is used;
= 2 or 5 then approximation (2) is used; and
= 3 or 6 then approximation (3) is used.

If IVAPRX = 1, 2, or 3, then, when user-supplied analytic derivatives are available [see argument NLSDRV], they are used to compute VCV; if IVAPRX = 4, 5, or 6, then only the predicted values from the model subroutine are used to compute VCV. When analytic derivatives are available, options 1, 2, or 3, will generally result in a faster, more accurate computation of VCV.

The default value for IVAPRX is 1. When argument IVAPRX is outside the range [1, 6], or when IVAPRX is not an argument of the subroutine CALL statement, then the default value will be used.

IXM --> The exact value of the first dimension of the matrix XM as specified in the calling program.

LDSTAK --> The length of the DOUBLE PRECISION workspace vector DSTAK. LDSTAK must equal or exceed the appropriate value given below, where if the single precision version of STARPAC is being used P = 0.5, otherwise P = 1.0 [see chapter 1, section B].

For NLS, NLSC, NLSS, NLSW, NLSWC and NLSWS:

$$\text{LDSTAK} \geq 27 + \max(\text{IS} * (\text{N} + \text{NPAR}), 30 + \text{NPARE}) + \max(\text{IS} * 10 * \text{N}, 94 + \text{N} * (3 + \text{NPAR}) + (3 * \text{NPARE} ** 2 + 37 * \text{NPARE}) / 2) * \text{P}$$

with IS = 1 if default values are used for the derivative step sizes, and IS = 0 otherwise.

For NLSD, NLSDC, NLSDS, NLSWD, NLSWDC and NLSWDS:

$$\text{LDSTAK} \geq 45 + \text{NPAR} + (94 + \text{N} * (3 + \text{NPAR}) + (3 * \text{NPARE} ** 2 + 35 * \text{NPARE}) / 2) * \text{P}$$

For STPLS and STPLSC:

$$\text{LDSTAK} \geq 27 + (\text{N} + \text{NPAR}) + 10 * \text{N} * \text{P}$$

For DCKLS and DCKLSC:

$$\text{LDSTAK} \geq 14 + \text{NPAR} + (\text{N} * \text{NPAR} + \text{N} + \text{NPAR}) * \text{P}$$

M --> The number of independent variables, i.e., the number of columns of data in XM.

MIT --> The maximum number of iterations allowed. This argument is also used to compute the maximum number of model subroutine calls, (2*MIT). The iterations will stop if either limit is reached, although, as a rule, the maximum number of iterations will be reached first. The default value for the maximum number of iterations is 21. When MIT <= 0 or when MIT is not an argument of the subroutine CALL statement the default value will be used.

N --> The number of observations.

NETA --> The number of reliable decimal digits in the predicted values (PV) computed by the user's model subroutine. The default value for NETA is experimentally determined by the procedure described in Appendix C. The default value will be used when NETA is not an argument in the subroutine CALL statement, or when the user-supplied value of NETA is outside the range [1, DIGITS], where DIGITS is the number of decimal digits carried by the user's computer for a single precision value when the single precision version of STARPAC is being used and is the number carried for a double precision value otherwise.

NLSDRV *** The name of the user-supplied subroutine that computes the partial derivative matrix (Jacobian). This argument must be listed in an EXTERNAL statement in the program which calls the STARPAC estimation or derivative checking subroutine. The form of the derivative subroutine argument list and dimensioning statements must be exactly as shown below, although if there is only one independent variable ($M = 1$), XM may be declared to be a vector with dimension IXM.

```

SUBROUTINE NLSDRV (PAR, NPAR, XM, N, M, IXM, D)
<real> PAR(NPAR), XM(IXM,M), D(N,NPAR)

< Computations for D(I,J), I = 1, ..., N and J = 1, ..., NPAR >

RETURN
END

```

NLSMDL *** The name of the user-supplied subroutine that computes the predicted value of the dependent variable given the independent variables and the current values of the model parameters. This argument must be listed in an EXTERNAL statement in the program which calls the STARPAC estimation, step size selection, and/or derivative checking subroutines. The form of the model subroutine argument list and dimensioning statements must be exactly as shown below, although if there is only one independent variable ($M = 1$), XM may be declared to be a vector with dimension IXM.

```

SUBROUTINE NLSMDL (PAR, NPAR, XM, N, M, IXM, PV)
<real> PAR(NPAR), XM(IXM,M), PV(N)

< Computations for PV(I), I = 1, ..., N >

RETURN
END

```

NNZW <-- The number of observations with nonzero weights. N.B., this value is returned by the estimation subroutines.

NPAR --> The number of parameters in the model, including both those held fixed at their starting values and those which are to be estimated.

NPARE <-- The number of parameters actually estimated, i.e., the number of zero elements in IFIXED. N.B., this value is returned by the

estimation subroutines.

NPRT --> The argument controlling printed output.

For the estimation subroutines:

NPRT is a five-digit integer, in which the value of the Ith digit (counting from left to right) is used to control the Ith section of the output.

If the Ith digit = 0 the output from the Ith section is suppressed;
= 1 the brief form of the Ith section is given;
>=2 the full form of the Ith section is given.

The default value for NPRT is 11112. When NPRT <= -1, or when NPRT is not an argument in the subroutine CALL statement, the default value will be used. If the convergence criteria are not satisfied the subroutine gives a suitable warning and provides a printed report even if NPRT = 0. A full discussion of the printed output is given in section E.2.a and is summarized as follows.

Section 1 lists the starting estimates and control values. Brief output and full output are the same for this section.

Section 2 reports the results of the iterations. Brief output includes information only about the first and last iteration while full output includes information about all of the iterations.

Section 3 provides information for each observation based on the final solution. Brief output includes information for the first 40 observations while full output provides the information for all of the data.

Section 4 is a set of four residual plots. Brief output and full output are the same for this section.

Section 5 is the final summary of the estimated parameters. Brief output does not include printing the variance-covariance matrix while full output does.

For the step size selection and derivative checking subroutines:

If NPRT = 0 the printed output is suppressed.

If NPRT <> 0 the printed output is provided.

When the acceptance criteria are not met a printed report is provided even if NPRT = 0.

NROW --> The row of the independent variable matrix at which the user-supplied derivative code is to be checked. The default value is the first row with no independent variables equal to zero; when

all rows have one or more independent variables equal to zero, row one will be used for the default value. When the user-supplied value is outside the range [1, N] or when NROW is not an argument of the subroutine CALL statement the default value will be used.

NTAU --> The agreement tolerance, i.e., the number of digits of agreement required between the user-supplied derivatives and the derivatives numerically approximated by the derivative checking subroutine. The default value is NETA/4. When the user-supplied value of NTAU is outside the range [1, NETA/2] or when NTAU is not an argument of the subroutine CALL statement the default value will be used.

PAR --- The vector of dimension at least NPAR that contains the parameter values. For all estimation subroutines it must contain initial values for the parameters on input and will contain the final values on return. For the step size and derivative checking subroutines it must contain the parameter values at which the operations are to be performed.

PV <-- The vector of dimension at least N that contains the predicted values of the dependent variable at the solution,

$$PV(i) = f(x(i), PAR) \quad \text{for } i = 1, \dots, N.$$

RES <-- The vector of dimension at least N that contains the residuals at the solution,

$$RES(i) = y(i) - f(x(i), PAR) = e(i) \quad \text{for } i = 1, \dots, N.$$

RSD <-- The residual standard deviation at the solution,

$$RSD = \sqrt{RSS(PAR)/(NNZW-NPARE)}.$$

SCALE --> The vector of dimension at least NPAR that contains the scale, or typical size, of each parameter. The vector SCALE is used to normalize the size of each parameter so that

$$|PAR(j)/SCALE(j)| \text{ approximates } |PAR(k)/SCALE(k)|$$

for $k = 1, \dots, NPAR$ and $j = 1, \dots, NPAR$.

Values of $|SCALE(k)| > |PAR(k)|$ can be used to increase the step size in cases where the model function is known to be insensitive to small changes in the value $PAR(k)$.

For the estimation subroutines:

The default values for SCALE are selected by the NL2SOL algorithm [Dennis et al., 1981a,b] and are updated at each iteration. When SCALE is not an argument in the subroutine CALL statement or when the user-supplied value for SCALE(1) ≤ 0 the default procedure will be used to select scale values. When SCALE(1) > 0 , values of SCALE(k) ≤ 0 for $k = 2, \dots, NPAR$ will be interpreted as an input error. User-supplied scale values may be either a vector of the typical size of each parameter or a vector of ones if the typical sizes of the parameters are

roughly equal; user-supplied scale values can sometimes result in reduced computing time since these values are not updated at each iteration.

For the derivative checking and step size selection subroutines:

The default values of SCALE are defined for $k = 1, \dots, \text{NPAR}$ as:

$$\text{SCALE}(k) = 1.0 \quad \text{if } \text{PAR}(k) = 0.0$$
$$\text{SCALE}(k) = |\text{PAR}(k)| \quad \text{otherwise}$$

where $\text{PAR}(k)$ is the input value of the k -th parameter.

When SCALE is not an argument in the subroutine CALL statement or when the user-supplied value of $|\text{SCALE}(k)| \leq |\text{PAR}(k)|$ the default value for $\text{SCALE}(k)$ is used. When $\text{SCALE}(1) \leq 0$, the default values will be used for $\text{SCALE}(k)$, $k = 1, \dots, \text{NPAR}$. When $\text{SCALE}(1) > 0$, values of $\text{SCALE}(k) \leq 0$ for $k = 2, \dots, \text{NPAR}$ will be interpreted as an input error.

SDPV <-- The vector of dimension at least N that contains an approximation to the standard deviation of each predicted value at the solution,

$\text{SDPV}(i)$ = the i th diagonal element of $\text{sqrt}(\text{Dhat} * \text{VCV} * \text{trans}(\text{Dhat}))$

for $i = 1, \dots, N$, where

$\text{Dhat}(i,j)$ = partial [$f(x(i), \text{PAR})$ wrt $\text{PAR}(j)$]

for $i = 1, \dots, N$ and $j = 1, \dots, \text{NPAR}$, evaluated at the solution, and $\text{trans}(\text{Dhat})$ is the transpose of Dhat.

This approximation is based on a linearization of the model in the neighborhood of the solution; the validity of the approximation depends on the nonlinearity of the model. This approximation may be extremely inaccurate for a problem with a highly nonlinear model.

SDRES <-- The vector of dimension at least N that contains an approximation to the standardized residuals at the solution,

$\text{SDRES}(i) = \text{RES}(i) / \text{sqrt}[(\text{RSD} ** 2 / \text{WT}(i)) - \text{SDPV}(i) ** 2]$

for $i = 1, \dots, N$, which is the i th residual divided by its individual estimated standard deviation. This approximation is based on a linearization of the model in the neighborhood of the solution; the validity of the approximation depends on the nonlinearity of the model. This approximation may be extremely inaccurate for a problem with a highly nonlinear model.

STOPP --> The stopping value for the convergence test based on the maximum scaled relative change in the parameters at the most recent iteration. The convergence criterion is satisfied if the current step is a Newton step and

$$\max[|\text{PARc}(k) - \text{PARp}(k)| / \text{SCALE}(k) \text{ for } k = 1, \dots, \text{NPAR}]$$

$$\max[(|\text{PARc}(k)| + |\text{PARp}(k)|) / \text{SCALE}(k) \text{ for } k = 1, \dots, \text{NPAR}] < \text{STOPP}.$$

where PARc(k) and PARp(k) indicate the current value and the value from the previous iteration, respectively, of the kth parameter [see Dennis et al. 1981a]. This convergence test is roughly equivalent to the test based on the maximum relative change in each parameter as measured by

$$\max(|\text{PARc}(k) - \text{PARp}(k)| / |\text{PARp}(k)| \text{ for } k = 1, \dots, \text{NPAR}).$$

STOPP is not a scale-dependent value; if its value is $10^{*(-4)}$ then this criteria will be met when the first four digits of each parameter are the same at two successive iterations regardless of the size of the parameter values.

The default value is approximately $10^{*(-\text{DIGITS}/2)}$, where DIGITS is the number of decimal digits carried by the user's computer for a single precision value when the single precision version of STARPAC is being used and is the number carried for a double precision value otherwise. When the user-supplied value for STOPP is outside the interval [0.0, 1.0] or when STOPP is not an argument of the subroutine CALL statement the default value will be used.

STOPSS --> The stopping value for the convergence test based on the ratio of the forecasted change in the residual sum of squares, $\text{fcst}(\text{RSS}(\text{PAR}))$, to the residual sum of squares from the previous iteration. The convergence criterion is satisfied if certain conditions are met and

$$\text{fcst}(\text{RSS}(\text{PAR})) / \text{RSS}(\text{PARp}) < \text{STOPSS},$$

where the notation is described in the description of argument STOPP [see Dennis et al., 1981a]. This convergence test is roughly equivalent to the test based on the relative change in the residual standard deviation between two iterations as measured by $(\text{RSDc} - \text{RSDl}) / \text{RSDc}$. STOPSS is not a scale-dependent value; if its value is $10^{*(-5)}$ this criteria will be met when the first five digits of the residual sum of squares are the same at two successive iterations regardless of the size of the residual sum of squares.

The default value is approximately the maximum of $10^{*(-10)}$ and $10^{*(-2*\text{DIGITS}/3)}$, where DIGITS is the number of decimal digits carried by the user's computer for a single precision value when the single precision version of STARPAC is being used and is the number carried for a double precision value otherwise. When the user-supplied value for STOPSS is outside the interval $[10^{*(-\text{DIGITS})}, 0.1]$ or when STOPSS is not an argument of the subroutine CALL statement the default value will be used.

STP --- The vector of dimension at least NPAR that contains the relative step sizes used to approximate the derivative matrix numerically. It is input to the estimation subroutines and returned from the step size selection subroutines. The procedure used to select the

default values is described in section E.1.b. For the estimation

<9-18>

subroutines, when STP is not an argument of the subroutine CALL statement or when STP(1) ≤ 0 the default values will be used for all of the step sizes, and when STP(1) > 0 values of STP(k) ≤ 0 for $k = 2, \dots, NPAR$ will be interpreted as an input error.

VCV <-- The matrix of dimension at least NPARE by NPARE that contains the variance-covariance matrix of the estimated parameters, approximated as designated by argument IVAPRX. The parameters which are held fixed [see argument IFIXED] are not included in the variance-covariance matrix.

The approximation of the variance-covariance matrix is based on a linearization of the model in the neighborhood of the solution; the validity of the approximation depends on the nonlinearity of the model. This approximation may be extremely inaccurate for a problem with a highly nonlinear model.

WT --> The vector of dimension at least N that contains the weights. Negative weights are not allowed and the number of nonzero weights must equal or exceed the number of parameters being estimated. A zero weight eliminates the corresponding observation from the analysis, although the residual, the predicted value and the standard deviation of the predicted value are still computed [see Appendix B].

XM --> The matrix of dimension at least N by M whose jth column contains the N values of the jth independent variable, $j = 1, \dots, M$.

Y --> The vector of dimension at least N that contains the dependent variable.

E. Computational Methods

E.1 Algorithms

E.1.a Nonlinear Least Squares Estimation

The nonlinear least squares estimation subroutines use the NL2SOL software package written by Dennis et al., [1981a,b]. The observations of the dependent variable, which are measured with error, are iteratively fit to a nonlinear model by minimizing the sums of squares of the errors as described in section A. The iterations continue until the convergence criteria based on the change in the parameter values or in the residual sum of squares are satisfied [see section D, arguments STOPP and STOPSS], the maximum number of iterations (or model subroutine calls) is reached [see section D, argument MIT], or the iterations are terminated due to singularity in the model or false convergence. All but the first of these stopping conditions may indicate computational problems and will produce an error report [see chapter 1, section D.5].

Singular convergence means that the model contains too many parameters, at least near the solution, while false convergence can indicate that either STOPSS or STOPP is set too small for the accuracy to which the model and its

derivatives are being computed or that there is an error or discontinuity in the derivative. Users should examine their models to determine and correct the underlying cause of singular or false convergence.

Iterative procedures for solving nonlinear least squares problems are discussed in Dennis and Schnabel [1983], Draper and Smith [1981] and Kennedy and Gentle [1980]. The specific procedure used in STARPAC is as follows. At the current iteration the values of the parameter vector PARc are given by

$$\text{PARc} = \text{PARp} - \text{inv}(\text{trans}(\text{Dp}) * \text{W} * \text{Dp} + \text{Sp} + \text{Gp}) * \text{trans}(\text{Dp}) * \text{W} * \text{trans}(\text{ep})$$

subject to the restriction that

$$\sqrt{\sum_{k=1}^{\text{NPAR}} [(\text{PARc}(k) - \text{PARp}(k)) / \text{SCALE}(k)]^2} \leq \text{dp},$$

where

trans(.) is the transpose of the designated matrix.

PARp is the vector of the NPAR estimated parameter values from the previous iteration.

Dp is the N by NPAR matrix of the partial derivatives evaluated at PARp,

$$D(i,k) = \text{partial}[f(x(i), \text{PAR}) \text{ wrt } \text{PAR}(k)]$$

for $i = 1, \dots, N$ and $k = 1, \dots, \text{NPAR}$.

W is an N by N diagonal matrix of user-supplied weights,

$$W = \text{diag}(\text{wt}(i), i = 1, \dots, N)$$

when a weighted analysis is performed and is the identity matrix otherwise.

Sp is an APPROXIMATION to the exact term Sp^* in the matrix of second order terms (Hessian) of the Taylor series expansion of the residual sum of squares function,

$$\text{Sp}^*(j,k) = \sum_{i=1}^N [\text{ep}(i) * \text{wt}(i) *$$

$$(\text{second partial } \text{ep}(i) \text{ wrt } \text{PARp}(j) \ \& \ \text{PARp}(k))],$$

for $j = 1, \dots, \text{NPAR}$ and $k = 1, \dots, \text{NPAR}$.

ep is the vector of the N residuals from the previous iteration.

dp is the adaptively chosen diameter of the trust region, i.e., the region in which the local approximation to the user's model function is reliable. At each iteration, dp is computed based on information from the previous iteration. At the first iteration, the initial value, d0, is supplied by argument DELTA which can be used to control the change in the parameters permitted at the first iteration.

G_p is an NPAR by NPAR diagonal matrix,

$$G_p = \text{diag}(g_p/\text{SCALE}(k), k = 1, \dots, \text{NPAR}),$$

where g_p is chosen to approximate the smallest non-negative number such that the restriction given above on the size of the change in the parameters is satisfied.

The second order term Sp^* , which is expensive and difficult to compute accurately, is important only if it is large compared to the term $\text{trans}(D_p) * W * D_p$, that is, when the residuals are large or the model is highly nonlinear. When Sp^* is large compared to $\text{trans}(D_p) * W * D_p$, algorithms which ignore it, such as Levenberg-Marquardt or Gauss-Newton, may converge slowly. The NL2SOL algorithm used by STARPAC, however, adaptively decides when inclusion of this term is necessary for reliable results and uses an inexpensive approximation to Sp^* in those cases.

The matrix, D , of partial derivatives of the model with respect to each parameter is either computed analytically using a user-supplied subroutine, NLSDRV, or is numerically approximated using forward difference quotients as described in section E.1.b. When the derivatives are approximated numerically, the least squares solution, especially the variance-covariance matrix, can be sensitive to the step sizes used for the approximation. The user may want to use STARPAC subroutines STPLS or STPLSC to recompute the step sizes at the solution provided by the estimation subroutines to assure that the step sizes which were used are still acceptable. If there is a significant change in the step size the least squares solution should be recomputed with the new step sizes from the current point. In addition, if the estimation subroutine has convergence problems the user may want to recompute the step sizes with the most recent parameter values to see if a change in the curvature of the model, which will be reflected as a change in the optimum step sizes, is causing the problem.

Dennis et al. [1981a] provides a detailed description of the algorithm used in STARPAC. STARPAC also includes the subroutines NL2SOL, NL2SNO, and NL2ITR, which they reference, and which can be used as documented by them [see Dennis et al., 1981b].

E.1.b Derivative Step Size Selection

The STARPAC step size selection subroutines use an algorithm developed by Schnabel [1982] to compute optimum step sizes for approximating the partial derivatives of the model with respect to each parameter. Briefly, the relative step sizes selected by these subroutines are those which produce forward difference quotient approximations to the derivative, D_{fd} , that agree reasonably well with the central difference quotient approximations, D_{cd} . The central difference quotient approximations are twice as accurate but also twice as expensive to compute. Since the additional accuracy is not usually needed, central difference quotient approximations are not used by the estimation subroutines.

The number of reliable digits in these derivatives is a function of the step sizes used to compute them. Given properly chosen step sizes, the number of reliable digits in D_{fd} and D_{cd} will be approximately $h/2$ and h , respectively, where h is the number of reliable digits in the predicted

values, PV, from the user's model subroutine. For example, if the predicted values are computed using an iterative procedure (such as quadrature or a solution of partial differential equations) which is expected to provide five good digits, then h would be five; if the predicted values are calculated from a simple algebraic expression translated directly into Fortran code, then h would (usually) be the number of decimal digits carried by the user's computer for the results.

The relative step size for PAR(k), k = 1, ..., NPAR, is initially

$$STP(k) = 2*\text{sqrt}(10**(-NETA)/q) \quad \text{for } k = 1, \dots, NPAR,$$

where

q is the average curvature (estimated by STARPAC) of the model with respect to PAR(k).

The forward difference quotient approximations with respect to PAR(k), k = 1, ..., NPAR are then

$$Dfd(i,k) = \frac{f(x(i),PARk) - f(x(i),PAR)}{STP(k)*SCALE(k)*SIGN(PAR(k))} \quad \text{for } i = 1, \dots, N,$$

where

f is the function which models the ith observation.

x(i) is the vector of the values of the M independent variables at the ith observation.

PAR is the vector of the NPAR parameter values.

PARk is a vector which has the same values as PAR except that the kth parameter is equal to

$$PAR(k) + STP(k)*SCALE(k)*SIGN(PAR(k)).$$

SIGN is a function which returns the sign of its argument.

The central difference approximations to the model derivative with respect to PAR(k), k = 1, ..., NPAR, are

$$Dcd(i,k) = \frac{f(x(i),PARpk) - f(x(i),PARmk)}{(3*10**(-NETA))**(1/3)*SCALE(k)*SIGN(PAR(k))} \quad \text{for } i = 1, \dots, N,$$

where

PARpk is a vector which has the same values as PAR except that the kth parameter is equal to

$$PAR(k) + (3*10**(-NETA))**(1/3)*SCALE(k)*SIGN(PAR(k)).$$

PARmk is a vector which has the same values as PAR except that the kth parameter is equal to

$$\text{PAR}(k) - (3 \cdot 10^{-(\text{NETA})})^{1/3} \cdot \text{SCALE}(k) \cdot \text{SIGN}(\text{PAR}(k)).$$

The relative step size is considered acceptable if, for at least N-a observations,

$$|\text{Dfd}(i,k) - \text{Dcd}(i,k)| \leq \min(10^{-(\text{NETA}/4)}, .02) \quad \text{for } i = 1, \dots, N,$$

<9-22>

where a is the number of observations exempted from meeting the above acceptance criterion [see section D, argument EXMPT]. If the step size is not acceptable, it is adjusted by factors of 10 until the condition is met or until no further decrease in the number of failures can be made, although in no case will the selected relative step size be greater than 1.0 or less than $10^{-(\text{NETA})}$.

Note that the step size selection subroutines will return the selected step sizes even when the number of failures exceeds the allowed value; this condition will be noted by the value of IERR. The detailed printed output should always be examined for problems discovered by the step size selection subroutines.

E.1.c Derivative Checking

The STARPAC derivative checking subroutines use an algorithm developed by Schnabel [1982] to determine the validity of the user-supplied derivative subroutine. The user-supplied derivative subroutine is considered correct for a given row i, $i = 1, \dots, N$, and parameter $\text{PAR}(k)$, $k = 1, \dots, \text{NPAR}$, if

$$|\text{Dfd}(i,k) - \text{D}(i,k)| \leq 10^{-(t)} \cdot |\text{D}(i,k)|$$

where

D is the derivative computed by the user's subroutine.

Dfd is the forward difference quotient approximation to the derivative described in section E.1.b.

t is the agreement tolerance, i.e., number of digits of agreement required between D and Dfd, which must be less than or equal to the number of good digits in Dfd [see section D, argument NTAU].

When the agreement tolerance is not satisfied the checking subroutine attempts to determine whether the disagreement is due to an error in the user's code or is due to the inaccuracy of the difference quotient approximation, caused either by high curvature in the user's model or by significant roundoff error.

The derivative checking subroutines each check only one row of the derivative matrix. The user should examine the row at which the derivatives were checked to ensure that some relation between the parameters and independent variables, such as a zero-valued independent variable or a factor $(x(i) - \text{PAR}(k))$ when $x(i) = \text{PAR}(k)$, is not hiding the effect of an incorrectly computed derivative. Checking only one row is appropriate since the same code is frequently used to compute the model function and derivatives at each row $i = 1, \dots, N$, as is the case in the examples shown in section F. If the code used to express the model function and derivatives is not the same for each

row, then each distinct section of the code should be checked by making multiple calls to DCKLSC with argument NROW set to a row within each section.

E.2 Computed Results and Printed Output

E.2.a The Nonlinear Least Squares Estimation Subroutines

The argument controlling the printed output, NPRT, is discussed in section D.

The output from the nonlinear least squares estimation subroutines consists of five sections, several of which include tables summarizing the results. In the following descriptions, the actual table headings are given by the uppercase phrases enclosed in angle braces (<...>). Results which correspond to input or returned subroutine CALL statement arguments are identified by the argument name in uppercase (not enclosed in angle braces).

Section 1 provides a summary of the initial estimates and control values. It lists the following information.

- * The initial values of the parameters, PAR, and whether they are to be held fixed or not as specified by IFIXED.
- * The scale values, SCALE.
- * Either the step sizes used to approximate the derivatives numerically, or, when user-supplied (analytic) derivatives are used, the results of the checking procedure; and the control values used in these computations as applicable [see section E.1.b and section E.1.c].
- * The number of observations, N.
- * The number of observations with nonzero weights, NNZW.
- * The number of independent variables, M.
- * The maximum number of iterations allowed, MIT.
- * The maximum number of model subroutine calls allowed.
- * The two convergence criteria, STOPSS and STOPP.
- * The maximum change in the parameters allowed at the first iteration, DELTA.
- * The residual sum of squares computed using the starting parameter values.
- * The residual standard deviation, RSD, computed using the starting parameter values.

Section 2 lists selected information about each iteration and includes the reason the iterations were terminated. The information provided for each iteration includes the following.

- * The iteration number.
- * <MODEL CALLS>: the total number of times since execution began that the user's model subroutine has been called, not including calls required to approximate the derivatives numerically.
- * <RSD>: the residual standard deviation computed using the parameter values from the current iteration.
- * <RSS>: the residual sum of squares computed using the parameter values from the current iteration.
- * <REL CHNG RSS>: the relative change in the residual sum of squares caused by the current iteration.
- * <FORECASTED REL CHNG RSS>: the forecasted relative change in the residual sum of squares at the current iteration, and whether this value was checked against STOPSS (<CHKD> = Y) or not (<CHKD> = N).
- * <REL CHNG PAR>: the maximum scaled relative change in the parameters at the current iteration, and whether this value was checked against STOPP (<CHKD> = Y) or not (<CHKD> = N).
- * <CURRENT PARAMETER VALUES>: the estimated parameter values resulting from the current iteration.

Section 3 provides the following information for each observation, $i = 1, \dots, N$, based on the final solution.

- * <ROW>: the row number of the observations.
- * <PREDICTOR VALUES>: the values for up to the first three columns of the independent variable matrix, XM, not including the first column if it is constant.
- * <DEPENDENT VARIABLE>: the values of the dependent variable, Y.
- * <PREDICTED VALUE>: the estimated predicted values, PV, from the fit.
- * <STD DEV OF PRED VALUE>: the standard deviations of the predicted values, SDPV.
- * <RESIDUAL>: the error estimates, RES.
- * <STD RES>: the standardized residuals, SDRES.
- * <WEIGHT>: the user-supplied weights, WT, printed only when weighted analysis is performed.

Section 4 displays the following plots of the standardized residuals.

- * The standardized residuals versus row numbers.
- * The standardized residuals versus predicted values.

- * The autocorrelation function of the (non-standardized) residuals.
- * The normal probability plot of the standardized residuals.

Section 5 summarizes the following information about the final parameter estimates and their variances.

- * The variance-covariance matrix, VCV, of the estimated (unfixed) parameters, and the corresponding correlation matrix,

$$r_{jk} = \text{VCV}(j,k) / \sqrt{\text{VCV}(j,j)\text{VCV}(k,k)} \quad \text{for } j = 1, \dots, \text{NPARE} \\ \text{and } k = 1, \dots, \text{NPARE}.$$

- * <PARAMETER>: the final value of each parameter, PAR(k), k = 1, ..., NPAR.

- * <SD OF PAR>: the standard deviation of each estimated parameter, $\sqrt{\text{VCV}(k,k)}$ for k = 1, ..., NPAR.

- * <RATIO>: the ratio of each estimated parameter to its standard deviation,

$$\text{RATIO}(k) = \text{PAR}(k) / \sqrt{\text{VCV}(k,k)} \quad \text{for } k = 1, \dots, \text{NPAR}.$$

- * <APPROXIMATE 95-PERCENT CONFIDENCE LIMITS>: the lower and upper 95-percent confidence limits for each parameter, computed using the appropriate value of the Student's t distribution with NNZW-NPARE degrees of freedom.

- * the residual sum of squares, RSS(PAR).

- * the residual standard deviation at the solution, RSD.

- * the residual degrees of freedom, NNZW-NPARE.

- * an approximation to the condition number of the derivative matrix, D (the Jacobian), under the assumption that the absolute error in each column of D is roughly equal. The approximation will be meaningless if this assumption is not valid; otherwise it will usually underestimate the actual condition number by a factor of from 2 to 10 [see Dongarra et al., 1979, p. 9.5]. (Note that the condition number returned by the nonlinear least squares subroutines is not exactly the same as that returned by the linear least subroutines because of differences in the computational procedures used by the two families of subroutines.)

NOTE: The standard deviation of the predicted values, the standardized residuals, the variance-covariance matrix, the standard deviations of the parameters and the 95-percent confidence limits on the parameters are all based on a linear approximation to the model in a neighborhood of the solution; the validity of this approximation depends on the nonlinearity of the model. The statistics based on this approximation may be extremely inaccurate for a problem with a highly nonlinear model.

E.2.b The Derivative Step Size Selection Subroutines

The argument controlling the printed output, NPRT, is discussed in section D.

The output from the step size selection subroutines consists of a summary of the input and control values and, for each parameter, the selected relative step size, the number of observations at which this step size failed the step size selection criteria and the row numbers at which the failures occurred.

E.2.c The Derivative Checking Subroutines

The argument controlling the printed output, NPRT, is discussed in section D.

The output for the derivative checking subroutines consists of a summary of the input and control values and the results of the derivative checking test with respect to each of the model parameters, PAR(k), $k = 1, \dots, \text{NPAR}$. The possible test results are:

OK -

- * The user-supplied derivative and the numerical derivative agree to the required number of digits.

QUESTIONABLE -

- * The user-supplied derivative and the approximated derivative agree to the required number of digits but both are equal to zero. The user should recheck the derivative at another row.
- * The user-supplied derivative and the approximated derivative do not agree to the required number of digits but the user-supplied derivative is identically zero and the approximated derivative is nearly zero. The user should recheck the derivative at another row.
- * The user-supplied derivative and the approximated derivative disagree but the user-supplied derivative is identically zero. The user should recheck the derivative at another row.
- * The user-supplied derivative and the approximated derivative disagree but the validity of the approximated derivative is questionable because either the ratio of the relative curvature of the model to the slope of the model is too high or SCALE(k) is wrong.
- * The user-supplied derivative and the approximated derivative disagree but the validity of the estimated derivative is questionable because the ratio of the relative curvature of the model to the slope of the model is too high.

INCORRECT -

- * The user-supplied derivative and the approximated derivative disagree, and there is no reason to question the accuracy of the approximated

derivative.

F. Examples

The sample programs of this section use the model and data given in example one, pages 428 to 441 of Daniel and Wood [1980]; the model is

$$f(x(i),b) = \text{PAR}(1)*x(i,1)**\text{PAR}(2) \quad \text{for } i = 1, \dots, N.$$

Nonlinear Least Squares Estimation. In the first example program below, NLS is used to compute the least squares solution using numerically approximated derivatives. In the second example program, NLS D is used to compute the least squares solution given analytic derivatives.

Derivative Step Size Selection. In the third example program, STPLS is used to compute the optimum step sizes for numerically approximating the derivatives with respect to each of the parameters, $\text{PAR}(k)$, $k = 1, 2$.

Derivative Checking. In the fourth example program below, DCKLS is used to check the validity of a user-supplied derivative subroutine. The derivative subroutine has been intentionally coded incorrectly in order to display the report obtained when the derivative checking subroutine determines the derivatives are incorrect, and the starting parameter values have been chosen in order to display the report obtained when the test results are questionable.

Program:

```
PROGRAM EXAMPL
C
C DEMONSTRATE NLS USING SINGLE PRECISION VERSION OF STARPAC
C
C N.B. DECLARATION OF Y, XM, PAR AND RES MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL Y(10), XM(10,5), PAR(5), RES(10)
DOUBLE PRECISION DSTAK(200)
C
EXTERNAL NLSMDL
C
COMMON /CSTAK/ DSTAK
C
SET UP INPUT AND OUTPUT FILES
[CHAPTER 1, SECTION D.4, DESCRIBES HOW TO CHANGE OUTPUT UNIT.]
C
CALL IPRINT(IPRT)
OPEN (UNIT=IPRT, FILE='FILENM')
OPEN (UNIT=5, FILE='DATA')
C
C SPECIFY NECESSARY DIMENSIONS
C
LDSTAK = 200
IXM = 10
C
```

FAILING STEP SIZE SELECTION CRITERIA				STEP SIZE FOR	OBSERVATIONS
NOTES	PARAMETER	STARTING VALUE	SCALE	APPROXIMATING DERIVATIVE	COUNT
INDEX	FIXED	(PAR)	(SCALE)	(STP)	F C
1	NO	.72500000	DEFAULT	.46415888E-04	0
2	NO	4.0000000	DEFAULT	.38782913E-06	0

* NOTES. A PLUS (+) IN THE COLUMNS HEADED F OR C HAS THE FOLLOWING MEANING.

F - NUMBER OF OBSERVATIONS FAILING STEP SIZE SELECTION CRITERIA EXCEEDS NUMBER OF EXEMPTIONS ALLOWED.

C - HIGH CURVATURE IN THE MODEL IS SUSPECTED AS THE CAUSE OF ALL FAILURES NOTED.

NUMBER OF RELIABLE DIGITS IN MODEL RESULTS	(NETA)	13
PROPORTION OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA	(EXMPT)	.1000
NUMBER OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA		1
NUMBER OF OBSERVATIONS	(N)	6
NUMBER OF INDEPENDENT VARIABLES	(M)	1
MAXIMUM NUMBER OF ITERATIONS ALLOWED	(MIT)	21
MAXIMUM NUMBER OF MODEL SUBROUTINE CALLS ALLOWED		42
CONVERGENCE CRITERION FOR TEST BASED ON THE		
FORECASTED RELATIVE CHANGE IN RESIDUAL SUM OF SQUARES (STOPSS)		.3696E-09
MAXIMUM SCALED RELATIVE CHANGE IN THE PARAMETERS (STOPP)		.8425E-07
MAXIMUM CHANGE ALLOWED IN THE PARAMETERS AT THE FIRST ITERATION (DELTA)		100.0
RESIDUAL SUM OF SQUARES FOR INPUT PARAMETER VALUES		.1472E-01
RESIDUAL STANDARD DEVIATION FOR INPUT PARAMETER VALUES	(RSD)	.6067E-01

STARPAC 2.08S (03/15/90)

NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES

ITERATION NUMBER 1

MODEL

FORECASTED

CALLS	RSD	RSS	REL CHNG RSS	REL CHNG RSS	REL
CHNG PAR				VALUE	CHKD
VALUE	CHKD				
2	.3390E-01	.4597E-02	.6877	.7109	Y
.1790E-01	Y				

CURRENT PARAMETER VALUES

INDEX	1	2
VALUE	.7679852	3.859309

ITERATION NUMBER 4

MODEL	RSD	RSS	REL CHNG RSS	FORECASTED	REL
CALLS				REL CHNG RSS	
CHNG PAR				VALUE	CHKD
VALUE	CHKD				
5	.3285E-01	.4317E-02	.8936E-12	.7028E-12	Y
.1123E-07	Y				

CURRENT PARAMETER VALUES

INDEX	1	2
VALUE	.7688623	3.860406

***** PARAMETER AND RESIDUAL SUM OF SQUARES CONVERGENCE *****

STARPAC 2.08S (03/15/90)

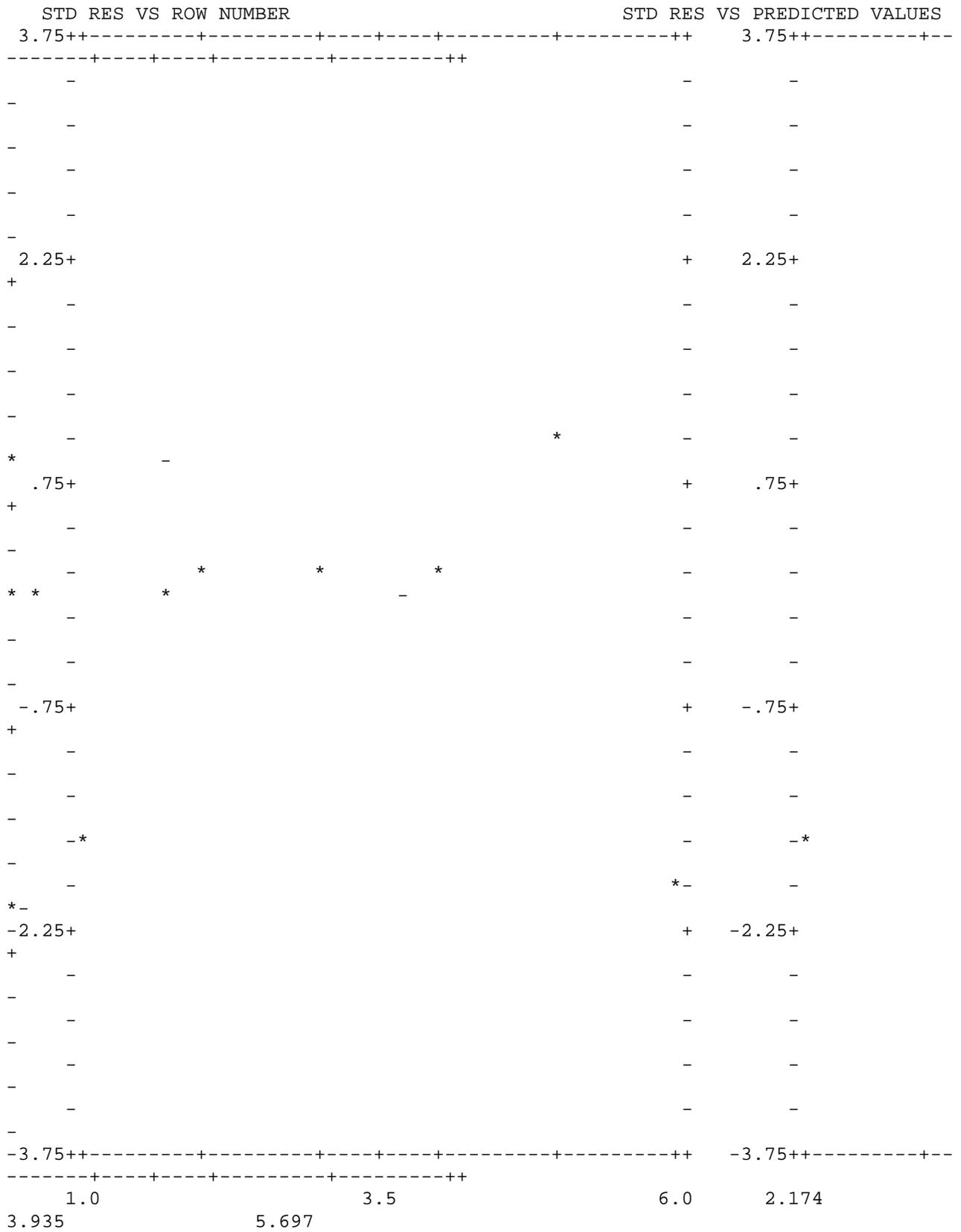
NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES

RESULTS FROM LEAST SQUARES FIT

STD DEV OF	PREDICTOR VALUES	STD	DEPENDENT	PREDICTED
ROW	RESIDUAL	RES	VARIABLE	VALUE
PRED VALUE				
1	1.3090000		2.1380000	2.1741175
.22079043E-01	-.36117492E-01	-1.48		
2	1.4710000		3.4210000	3.4111549
.16469586E-01	.98450831E-02	.35		
3	1.4900000		3.5970000	3.5844108
.15615321E-01	.12589151E-01	.44		
4	1.5650000		4.3400000	4.3326419
.14065814E-01	.73580833E-02	.25		
5	1.6110000		4.8820000	4.8453073
.16512112E-01	.36692701E-01	1.29		
6	1.6800000		5.6600000	5.6968365
.26183728E-01	-.36836492E-01	-1.86		

STARPAC 2.08S (03/15/90)

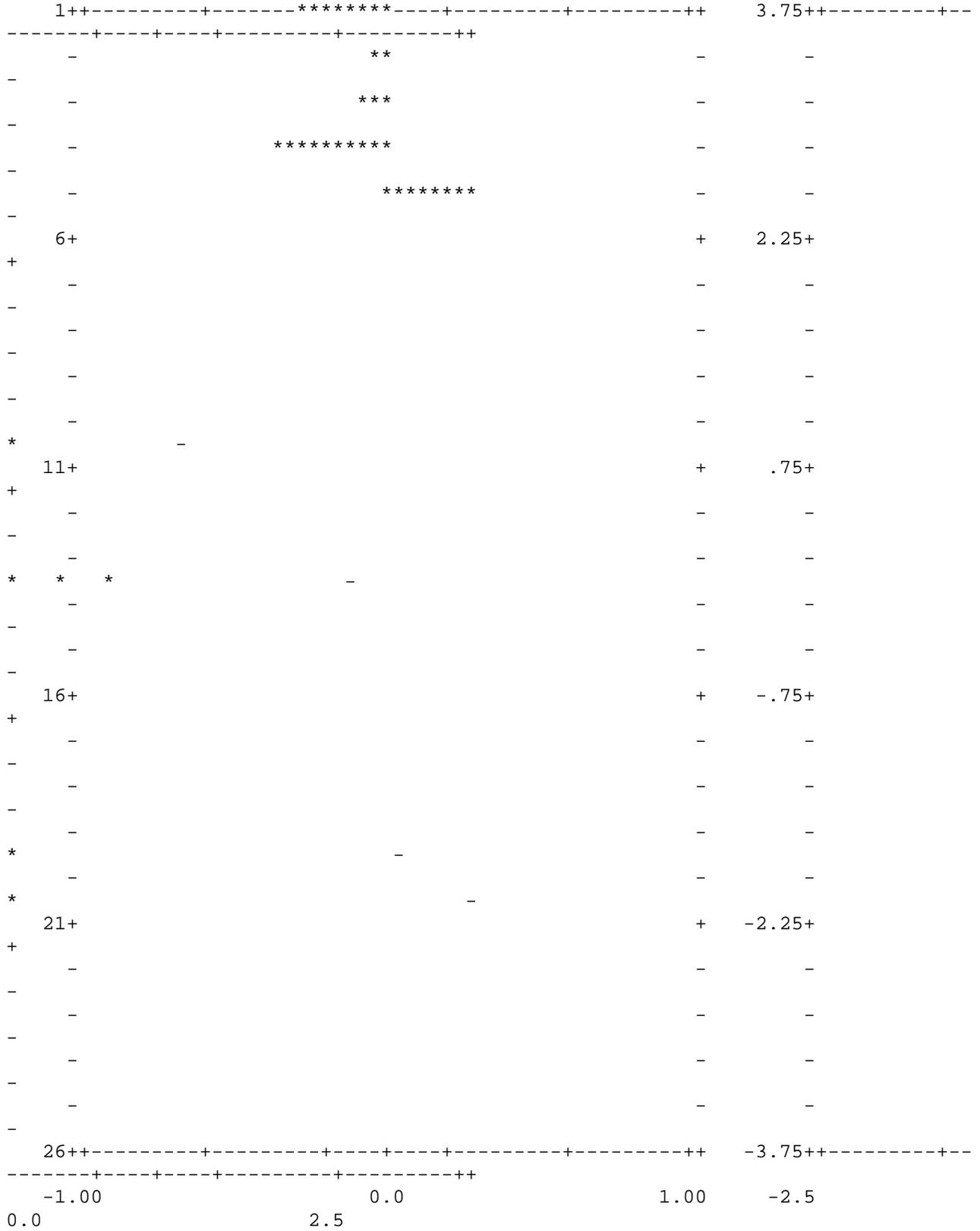
NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES



AUTOCORRELATION FUNCTION OF RESIDUALS

NORMAL PROBABILITY PLOT OF

STD RES



STARPAC 2.08S (03/15/90)
NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES

VARIANCE-COVARIANCE AND CORRELATION MATRICES OF THE ESTIMATED (UNFIXED) PARAMS

- APPROXIMATION BASED ON ASSUMPTION THAT RESIDUALS ARE SMALL
- COVARIANCES ARE ABOVE THE DIAGONAL
- VARIANCES ARE ON THE DIAGONAL
- CORRELATION COEFFICIENTS ARE BELOW THE DIAGONAL

COLUMN	1	2
1	.3342304E-03	-.9369370E-03
2	-.9907719	.2675639E-02

ESTIMATES FROM LEAST SQUARES FIT

INDEX	FIXED	PARAMETER	SD OF PAR	95 PERCENT CONFIDENCE RATIO	APPROXIMATE LOWER LIMITS
1	NO	.76886226	.18281968E-01	42.06	.71810338
2	NO	3.8604056	.51726577E-01	74.63	3.7167896

RESIDUAL SUM OF SQUARES .4317308E-02

RESIDUAL STANDARD DEVIATION .3285311E-01
BASED ON DEGREES OF FREEDOM 6 - 2 = 4

APPROXIMATE CONDITION NUMBER 20.87491

Program:

```

PROGRAM EXAMPL
C
C DEMONSTRATE NLSD USING SINGLE PRECISION VERSION OF STARPAC
C
C N.B. DECLARATION OF Y, XM, PAR AND RES MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL Y(10), XM(10,5), PAR(5), RES(10)
DOUBLE PRECISION DSTAK(200)
C
EXTERNAL NLSMDL, NLSDRV

```



```

C
C   SUBROUTINE TO COMPUTE THE PARTIAL DERIVATIVE (JACOBIAN) MATRIX
C
C   N.B. DECLARATION OF PAR, XM AND D MUST BE CHANGED TO DOUBLE
C       PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C   REAL PAR(NPAR), XM(IXM,M), D(N,NPAR)
C
C   DO 10 I = 1, N
C       D(I,1) = XM(I,1) ** PAR(2)
C       D(I,2) = PAR(1) * XM(I,1) ** PAR(2) * ALOG(XM(I,1))
10  CONTINUE
C
C   RETURN
C   END

```

Data:

```

      2
0.725 4.000
      6      1
1.309 1.471 1.490 1.565 1.611 1.680
2.138 3.421 3.597 4.340 4.882 5.660

```

RESULTS OF STARPAC NONLINEAR LEAST SQUARES SUBROUTINE NLSD
 STARPAC 2.08S (03/15/90)

 * NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES *

SUMMARY OF INITIAL CONDITIONS

INDEX	PARAMETER FIXED	STARTING VALUE (PAR)	SCALE (SCALE)	DERIVATIVE ASSESSMENT
1	NO	.72500000	DEFAULT	OK
2	NO	4.0000000	DEFAULT	OK

NUMBER OF RELIABLE DIGITS IN MODEL RESULTS	(NETA)	13
NUMBER OF DIGITS IN DERIVATIVE CHECKING AGREEMENT TOLERANCE	(NTAU)	4
ROW NUMBER AT WHICH DERIVATIVES WERE CHECKED	(NROW)	1
-VALUES OF THE INDEPENDENT VARIABLES AT THIS ROW		
INDEX	1	
VALUE	1.309000	
NUMBER OF OBSERVATIONS	(N)	6
NUMBER OF INDEPENDENT VARIABLES	(M)	1
MAXIMUM NUMBER OF ITERATIONS ALLOWED	(MIT)	21

MAXIMUM NUMBER OF MODEL SUBROUTINE CALLS ALLOWED

42

CONVERGENCE CRITERION FOR TEST BASED ON THE

FORECASTED RELATIVE CHANGE IN RESIDUAL SUM OF SQUARES (STOPSS)	.3696E-09
MAXIMUM SCALED RELATIVE CHANGE IN THE PARAMETERS (STOPP)	.8425E-07

MAXIMUM CHANGE ALLOWED IN THE PARAMETERS AT THE FIRST ITERATION (DELTA) 100.0

RESIDUAL SUM OF SQUARES FOR INPUT PARAMETER VALUES .1472E-01

RESIDUAL STANDARD DEVIATION FOR INPUT PARAMETER VALUES (RSD) .6067E-01

STARPAC 2.08S (03/15/90)

NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED

ITERATION NUMBER 1

MODEL					FORECASTED		
CALLS	RSD	RSS	REL CHNG	RSS	REL CHNG	RSS	REL
CHNG PAR					VALUE	CHKD	
VALUE	CHKD						
2	.3390E-01	.4597E-02	.6877		.7109	Y	
.1790E-01	Y						

CURRENT PARAMETER VALUES

INDEX	1	2
VALUE	.7679852	3.859309

ITERATION NUMBER 4

MODEL					FORECASTED		
CALLS	RSD	RSS	REL CHNG	RSS	REL CHNG	RSS	REL
CHNG PAR					VALUE	CHKD	
VALUE	CHKD						
5	.3285E-01	.4317E-02	-.3214E-13		.6352E-12	Y	
.1068E-07	Y						

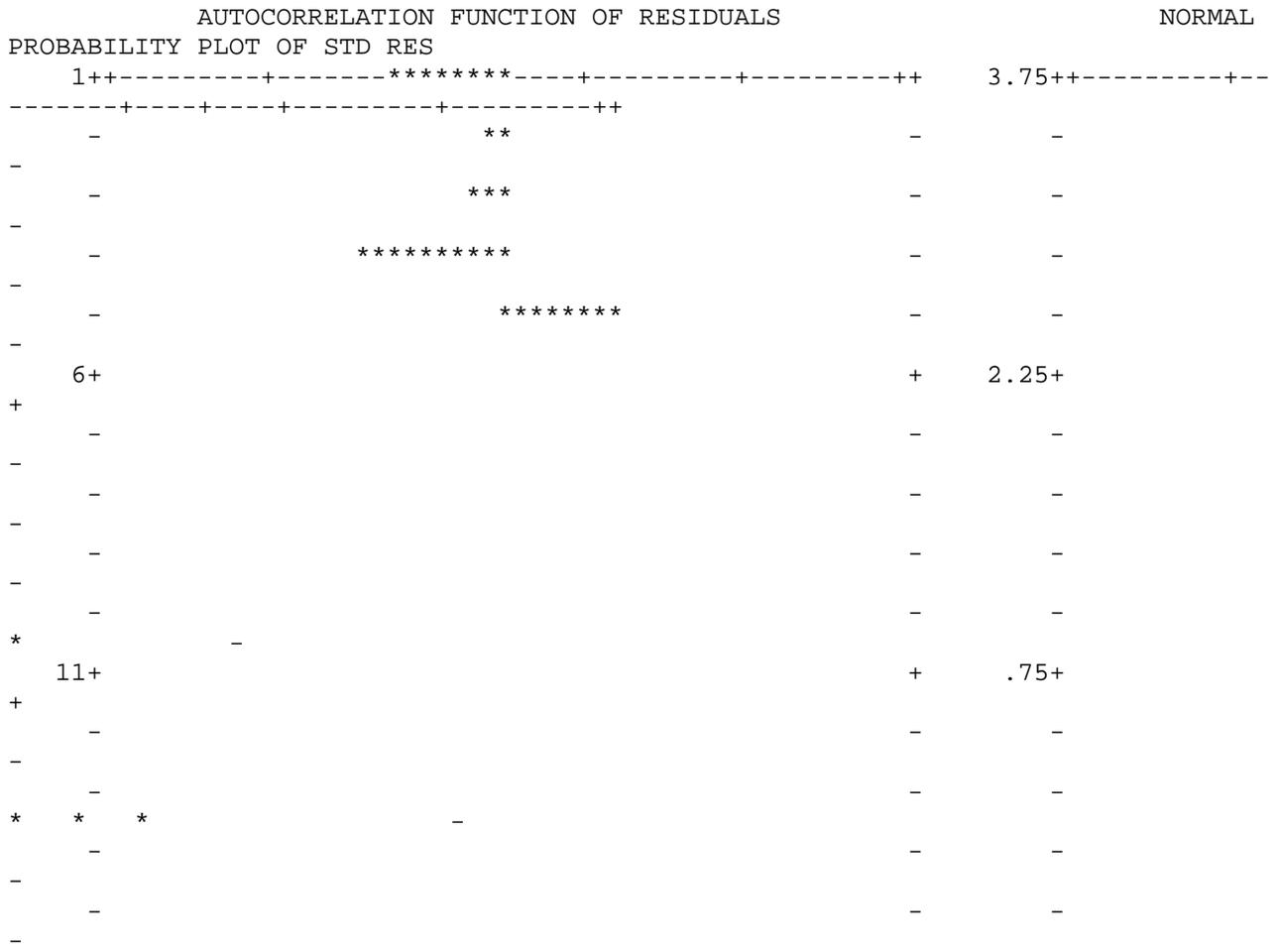
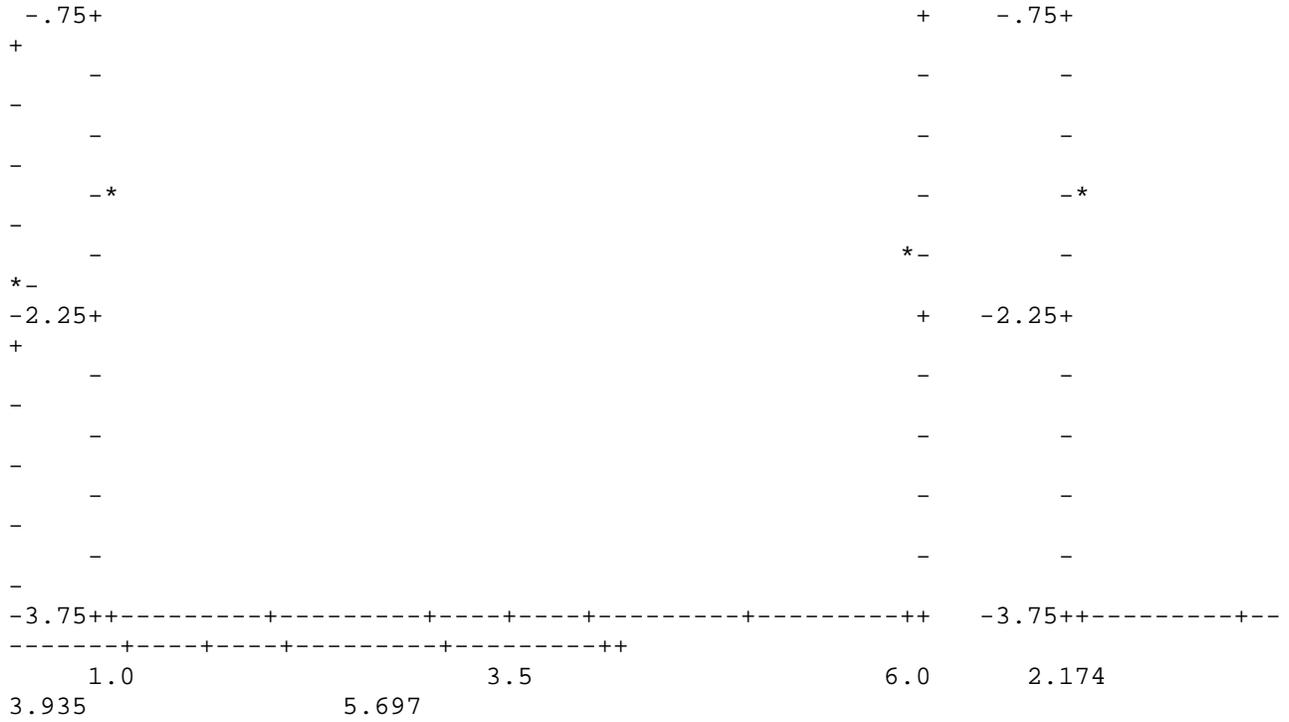
CURRENT PARAMETER VALUES

INDEX	1	2
VALUE	.7688623	3.860406

***** PARAMETER AND RESIDUAL SUM OF SQUARES CONVERGENCE *****

STARPAC 2.08S (03/15/90)

NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED



```

16+                                     +   -.75+
+
-                                     -   -
-                                     -   -
-                                     -   -
*                                     -   -
*                                     -   -
21+                                     +   -2.25+
+
-                                     -   -
-                                     -   -
-                                     -   -
-                                     -   -
-                                     -   -
26+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+
-1.00                                     0.0                                     1.00   -2.5
0.0                                     2.5
<9-42>

```

STARPAC 2.08S (03/15/90)
NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED

VARIANCE-COVARIANCE AND CORRELATION MATRICES OF THE ESTIMATED (UNFIXED) PARAMS

- APPROXIMATION BASED ON ASSUMPTION THAT RESIDUALS ARE SMALL
- COVARIANCES ARE ABOVE THE DIAGONAL
- VARIANCES ARE ON THE DIAGONAL
- CORRELATION COEFFICIENTS ARE BELOW THE DIAGONAL

COLUMN	1	2
1	.3342306E-03	-.9369379E-03
2	-.9907719	.2675642E-02

ESTIMATES FROM LEAST SQUARES FIT

					APPROXIMATE	
					95 PERCENT	
CONFIDENCE LIMITS	INDEX	FIXED	PARAMETER	SD OF PAR	RATIO	LOWER
UPPER						

1	NO	.76886229	.18281974E-01	42.06	.71810339
.81962119					
2	NO	3.8604055	.51726611E-01	74.63	3.7167894
4.0040216					

RESIDUAL SUM OF SQUARES .4317308E-02

RESIDUAL STANDARD DEVIATION .3285311E-01
 BASED ON DEGREES OF FREEDOM 6 - 2 = 4

APPROXIMATE CONDITION NUMBER 20.87492

Program:

```

PROGRAM EXAMPL
C
C DEMONSTRATE STPLS USING SINGLE PRECISION VERSION OF STARPAC
C
C N.B. DECLARATION OF XM, PAR AND STP MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL XM(10,5), PAR(5), STP(5)
DOUBLE PRECISION DSTAK(200)
C
EXTERNAL NLSMDL, DERIV
C
COMMON /CSTAK/ DSTAK
C
SET UP INPUT AND OUTPUT FILES
C [CHAPTER 1, SECTION D.4, DESCRIBES HOW TO CHANGE OUTPUT UNIT.]
C
CALL IPRINT(IPRT)
OPEN (UNIT=IPRT, FILE='FILENM')
OPEN (UNIT=5, FILE='DATA')
C
C SPECIFY NECESSARY DIMENSIONS
C
LDSTAK = 200
IXM = 10
C
C READ NUMBER OF PARAMETERS
C STARTING VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS AND NUMBER OF INDEPENDENT VARIABLES
C INDEPENDENT VARIABLES
C
READ (5,100) NPAR
READ (5,101) (PAR(I), I=1,NPAR)
READ (5,100) N, M
READ (5,101) ((XM(I,J), I=1,N), J=1,M)
C
C PRINT TITLE AND CALL STPLS TO SELECT STEP SIZES FOR
C APPROXIMATING DERIVATIVES
C
WRITE (IPRT,102)
CALL STPLS (XM, N, M, IXM, NLSMDL, PAR, NPAR, LDSTAK, STP)
C

```


NUMBER OF EXEMPTIONS ALLOWED.

C - HIGH CURVATURE IN THE MODEL IS SUSPECTED AS THE CAUSE OF ALL FAILURES NOTED.

NUMBER OF RELIABLE DIGITS IN MODEL RESULTS	(NETA)	13
PROPORTION OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA	(EXMPT)	.1000
NUMBER OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA		1
NUMBER OF OBSERVATIONS	(N)	6

Program:

```
PROGRAM EXAMPL
C
C DEMONSTRATE DCKLS USING SINGLE PRECISION VERSION OF STARPAC
C
C N.B. DECLARATION OF XM AND PAR MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL XM(10,5), PAR(5)
DOUBLE PRECISION DSTAK(200)
C
EXTERNAL NLSMDL, NLSDRV
C
COMMON /CSTAK/ DSTAK
C
SET UP INPUT AND OUTPUT FILES
C [CHAPTER 1, SECTION D.4, DESCRIBES HOW TO CHANGE OUTPUT UNIT.]
C
CALL IPRINT(IPRT)
OPEN (UNIT=IPRT, FILE='FILENM')
OPEN (UNIT=5, FILE='DATA')
C
C SPECIFY NECESSARY DIMENSIONS
C
LDSTAK = 200
IXM = 10
C
C READ NUMBER OF PARAMETERS
C STARTING VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS AND NUMBER OF INDEPENDENT VARIABLES
C INDEPENDENT VARIABLES
C
READ (5,100) NPAR
READ (5,101) (PAR(I), I=1,NPAR)
READ (5,100) N, M
READ (5,101) ((XM(I,J), I=1,N), J=1,M)
C
C PRINT TITLE AND CALL DCKLS TO PERFORM DERIVATIVE CHECKING
C
WRITE (IPRT,102)
CALL DCKLS (XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, LDSTAK)
C
C FORMAT STATEMENTS
```

```

C
100 FORMAT (2I5)
101 FORMAT (6F6.3)
102 FORMAT ('1RESULTS OF STARPAC',
* ' DERIVATIVE CHECKING SUBROUTINE DCKLS')
END
SUBROUTINE NLSMDL (PAR, NPAR, XM, N, M, IXM, PV)
C
C SUBROUTINE TO COMPUTE PREDICTED VALUES OF DEPENDENT VARIABLE
C
C N.B. DECLARATION OF PAR, XM AND PV MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL PAR(NPAR), XM(IXM,M), PV(N)
C
C <9-47>
C
DO 10 I = 1, N
PV(I) = PAR(1) * XM(I, 1) ** PAR(2)
10 CONTINUE
C
RETURN
END
SUBROUTINE NLSDRV (PAR, NPAR, XM, N, M, IXM, D)
C
C SUBROUTINE TO COMPUTE THE PARTIAL DERIVATIVE (JACOBIAN) MATRIX
C
C DERIVATIVE WITH RESPECT TO FIRST PARAMETER HAS BEEN CODED
C INCORRECTLY TO DEMONSTRATE ERROR DETECTION CAPABILITIES
C
C N.B. DECLARATION OF PAR, XM AND D MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
REAL PAR(NPAR), XM(IXM,M), D(N,NPAR)
C
DO 10 I = 1, N
D(I,1) = XM(I,1) * PAR(2)
D(I,2) = PAR(1) * XM(I,1) ** PAR(1) * ALOG(XM(I,1))
10 CONTINUE
C
RETURN
END

```

Data:

```

2
0.000 4.000
6 1
1.309 1.471 1.490 1.565 1.611 1.680

```

RESULTS OF STARPAC DERIVATIVE CHECKING SUBROUTINE DCKLS

STARPAC 2.08S (03/15/90)

```

*****
* DERIVATIVE CHECKING *

```

INDEX	PARAMETER STARTING VALUE (PAR)	SCALE (SCALE)	DERIVATIVE ASSESSMENT
1	0.	DEFAULT	INCORRECT
2	4.0000000	DEFAULT	QUESTIONABLE (1)

* NUMBERS IN PARENTHESES REFER TO THE FOLLOWING NOTES.

(1) USER-SUPPLIED AND APPROXIMATED DERIVATIVES AGREE, BUT BOTH ARE ZERO. RECHECK AT ANOTHER ROW.

NUMBER OF RELIABLE DIGITS IN MODEL RESULTS	(NETA)	14
NUMBER OF DIGITS IN DERIVATIVE CHECKING AGREEMENT TOLERANCE	(NTAU)	4
ROW NUMBER AT WHICH DERIVATIVES WERE CHECKED	(NROW)	1
-VALUES OF THE INDEPENDENT VARIABLES AT THIS ROW		
INDEX	1	
VALUE	1.309000	
NUMBER OF OBSERVATIONS	(N)	6

<9-49>

G. Acknowledgments

The subroutines used to compute the nonlinear least squares solution are those referenced in Dennis et al. [1981]. The algorithms used to select optimum step sizes for numerical derivatives, and to check analytic derivatives were developed by Schnabel [1982]. The printed output for the nonlinear least squares subroutines has been modeled on the linear least squares output used by OMNITAB II [Hogben et al., 1971].

 Appendix B

WEIGHTED LEAST SQUARES

Weighted least squares can be used to eliminate observations from the analysis and to compensate for unequal variances in the observational errors.

Observations can be eliminated from the analysis by using weight values consisting only of zeros and ones. This will produce the same results as performing an unweighted analysis with the zero-weighted values removed except that the predicted values, the standard deviations of the predicted values, and the residuals of the zero-weighted data are computed. There are two main reasons for weighting observations zero. The first is to obtain the predicted values and their standard deviations for a set of independent variables not

included in the observed data. (This is done by assigning any arbitrary value to the dependent variable of the desired set of independent variables, and then weighting these values zero.) The second reason is to allow easy examination of the effect of outliers and influential data points. Outliers often appear as large values in residual plots. Careful checking of the data often leads to confirmation that the data are in error, and sometimes to a correction. When a cause for suspicious data cannot be found, it may be advisable to compare the analysis with and without the questionable data. Caution is in order if the estimates or conclusions are highly sensitive to a small amount of suspicious data. Data that have a very high influence on a fitted curve may not result in large residuals, however, even if they are in error. In fact, extremely influential observations may force the fitted curve to be very close, leading to very small residuals. It is therefore desirable to identify influential observations and to compare the results obtained with and without these points. Several methods for detecting influential observations are discussed in Bement and Williams [1969], Cook [1977], Hoaglin and Welsch [1978], and Belsley et al. [1980].

Using weights to compensate for unequal observational error variances is not as straightforward as using zero weights to eliminate observations from the analysis. When the variances of the observational errors, $e(i)$, are not equal, the unweighted least squares estimates remain unbiased but do not have minimum variance. Minimum variance estimates are obtained by using weights $w_i = 1/\text{Variance}[e(i)]$ when the error variances are known. If weights must be estimated, they should be based on at least 10 degrees of freedom [see Bement and Williams, 1969]. In practice, however, weights are derived from theory, or obtained from the data being fit, and either of these methods can do more harm than good. When the need for weights is suspected and the error variances are not known, first fit the data using unweighted least squares; analysis of the residuals may confirm the need for weighting and may also provide estimates for the weights themselves. If the need for weights is confirmed, then a statistician should be consulted to assist in selecting the weights and in interpreting the results.

Appendix C

ESTIMATING THE NUMBER OF RELIABLE DIGITS
IN THE RESULTS OF A FUNCTION

The number of reliable digits, h , in the results of a real valued function, $g(b)$, can be estimated in most cases by evaluating

$$h = -\log_{10} \left(\max_{j=-2, \dots, 2} \frac{(|g(b_j) - [a+j*b]|)}{|g(b)|} \right)$$

where

bj is the vector of the NPAR parameters of the function given by,

$$bj(k) = b(k) + b(k)*j*10**(-DIGITS/2) \quad \text{for } j = -2, \dots, 2, \\ \text{and } k = 1, \dots, \text{NPAR},$$

where

DIGITS is the number of decimal digits carried by the user's computer for a single precision value when the single precision version of STARPAC is being used and is the number carried for a double precision value otherwise.

$$a = (0.20) * \sum_{j=-2}^2 g(bj).$$

$$b = (0.10) * \sum_{j=-2}^2 j*g(bj).$$

This procedure may underestimate the number of reliable digits if g(b) is extremely nonlinear. A more elaborate and more robust procedure is described in Gill et al. [1981].

 Appendix D

LIST OF STARPAC SUBPROGRAM NAMES

D.1 Subprograms specifically written for STARPAC

ABSCOM	ACCDIG	ACF	ACFD	ACFDTL	ACFER	ACFF	ACFFS	ACFLST
ACFM	ACFMN	ACFMNF	ACFMNM	ACFMS	ACFOUT	ACFS	ACFSD	ACFSDM
ACVF	ACVFF	ACVFM	ADJLMT	AIME	AIMEC	AIMES	AIMF	AIMFS
AIMX1	AMDRV	AMEAN	AMEANM	AMECNT	AMEDRV	AMEER	AMEFIN	AMEHDR
AMEISM	AMEMN	AMEOUT	AMEPT1	AMEPT2	AMESTP	AMFCNT	AMFER	AMFHDR
AMFMN	AMFOUT	AMLST	AMLST1	AOS	AOSLST	AOV1	AOV1ER	AOV1HD
AOV1MN	AOV1S	AOV1XP	ARCOEF	ARFLT	AXPBY	BACKOP	BFS	BFSDRV
BFSER	BFSF	BFSFS	BFSLAG	BFSM	BFSMN	BFSMS	BFSMV	BFSMVS
BFSS	BFSV	BFSVS	CCF	CCFER	CCFF	CCFFS	CCFLST	CCFM
CCFMN	CCFMNF	CCFMNM	CCFMS	CCFOUT	CCFS	CCFSD	CCFSDM	CCFXP
CCVF	CCVFF	CCVFM	CDFCHI	CDFF	CDFNML	CDFT	CENTER	CHIRHO
CMPFD	CNTR	CORR	CORRER	CORRHD	CORRMN	CORRS	CORRXP	CPYASF
CPYMSS	CPYVII	DCKCNT	DCKCRV	DCKDRV	DCKER	DCKFPA	DCKHDR	DCKLS
DCKLSC	DCKLS1	DCKMN	DCKOUT	DCKZRO	DCEOEF	DEMDRV	DEMOM	DEMOMDS
DEMOMU	DEMORD	DEMOUT	DFBW	DFBWM	DIF	DIFC	DIFM	DIFMC
DIFSER	DOTC	DOTCM	DRV	DRV1A	DRV1B	DRV2	DRV3	DRV4A
DRV4B	ECVF	EHDR	EIAGE	EIAGEP	EISEQ	EISGE	EISII	EISLE
EISRNG	EIVEO	EIVEQ	EIVII	ENFFT	ERAGT	ERAGTM	ERAGTP	ERDF
ERIODD	ERSEI	ERSGE	ERSGT	ERSIE	ERSII	ERSLF	ERSLFS	ERVGT
ERVGTM	ERVGTP	ERVII	ERVWT	ETAMD	EXTEND	FACTOR	FFT	FFTCT
FFTLEN	FFTR	FITEXT	FITPT1	FITPT2	FITSXP	FITXSP	FIXPRT	FLTAR
FLTARM	FLTMA	FLTMD	FLTSL	GENI	GENR	GETPI	GFAEST	GFARF
GFARFS	GFORD	GFOUT	GFSEST	GFSLF	GFSLFS	GMEAN	HIPASS	HIST
HISTC	HPCOEF	HPFLT	HSTER	HSTMN	ICNTI	ICOPY	INPERL	IPGDV

IPGM	IPGMN	IPGMP	IPGMPS	IPGMS	IPGORD	IPGOUT	IPRINT	LDSCMP
LLCNT	LLCNTG	LLCNTP	LLER	LLHDRG	LLHDRP	LLS	LLSMN	LLSP
LLSPS	LLSPW	LLSPWS	LLSS	LLSW	LLSWS	LOGLMT	LOPASS	LPCOEF
LPFLT	LSTLAG	LSTVCF	LSTVEC	MAFLT	MATPRF	MATPRT	MDFLT	MDLTS1
MDLTS2	MDLTS3	MDL1	MDL2	MDL3	MDL4	MGS	MODSUM	MPP
MPPC	MPPL	MPPM	MPPMC	MPPML	MSGX	MULTBP	MVCHK	MVP
MVPC	MVPL	MVPM	MVPMC	MVPML	NCHOSE	NLCMP	NLCNT	NLCNTA
NLCNTN	NLDRVA	NLDRVN	NLER	NLERR	NLFIN	NLHDRA	NLHDRN	NLINIT
NLISM	NLITRP	NLMN	NLOUT	NLS	NLSC	NLSD	NLSDC	NLSDS
NLSKL	NLSPK	NLSS	NLSUPK	NLSW	NLSWC	NLSWD	NLSWDC	NLSWDS
NLSWS	NLSX1	NLSX2	NRAND	NRANDC	OANOVA	OBSSM2	OBSSUM	PARENZ
PGM	PGMEST	PGMMN	PGMS	PGORD	PGOUT	PLINE	PLTCHK	PLTPLX
PLTSYM	POLAR	PP	PPC	PPCNT	PPFCHS	PPFF	PPFNML	PPFT
PPL	PPLMT	PPM	PPMC	PPML	PPMN	PRTCNT	RANDN	RANDU
RANKO	REALTR	RELCOM	REPCK	SAMPLE	SETESL	SETFRQ	SETIV	SETLAG
SETRA	SETROW	SETRV	SLFLT	SMPLY	SPCCK	SPP	SPPC	SPPL
SPPLTC	SPPLTD	SPPLTL	SPPM	SPPMC	SPPML	SRTIR	SRTIRR	SRTRI
SRTRRI	STAT	STATER	STATS	STATW	STATWS	STAT1	STAT1W	STAT2
STAT2W	STKCLR	STKGET	STKREL	STKSET	STKST	STPADJ	STPAMO	STPCNT
STPDRV	STPER	STPHDR	STPLS	STPLSC	STPLS1	STPLS2	STPMN	STPOUT
STPSEL	SUMBS	SUMDS	SUMID	SUMIDW	SUMOT	SUMSS	SUMTS	SUMWDS
SUMWSS	SUMWTS	SVP	SVPC	SVPL	SVPM	SVPMC	SVPML	TAPER
UAS	UASCFT	UASDV	UASER	UASEST	UASF	UASFS	UASORD	UASOUT
UASS	UASV	UASVAR	UASVS	UFS	UFSDRV	UFSEST	UFSEST	UFSF
UFSFS	UFSLAG	UFSM	UFSMN	UFSMS	UFSMV	UFSMVS	UFSOUT	UFSPCV
UFSS	UFSV	UFSVS	VCVOTF	VCVOUT	VERSP	VP	VPC	VPCNT
VPHEAD	VPL	VPLMT	VPM	VPMC	VPML	VPMN	XACF	XAIMD
XAIMT	XAOV1	XBFS	XCCF	XCORR	XDCKLD	XDCKLE	XDCKLT	XDEMOD
XDFLT	XHIST	XLLS	XNLS	XNLSE	XNLST	XNRAND	XPGM	XPP
XSTAT	XSTPLD	XSTPLE	XSTPLT	XUAS	XUFS	XVP	XXCH1	XXCH2
XXCH3	XXCH4	XXCH5	XXCH6	XXCH7	XXCH8	XXCH9	XXCH10	XXCH11
XXCH12	XXCH13							

D.2 Subprograms from NL2SOL

ASSESS	COVCLC	DFAULT	DOTPRD	DUPDAT	GQTSTP	IMDCON	ITSMRY	LINVRT
LITVMU	LIVMUL	LMSTEP	LSQRT	LSVMIN	LTSQAR	MADJ	MADR	NL2ITR
NL2SNO	NL2SOL	NL2X	PARCHK	QAPPLY	QRFACT	RELDST	RMDCON	RPTMUL
SLUPDT	SLVMUL	STOPX	UFPARM	VAXPY	VCOPY	VSCOPY	V2NORM	

D.3 Subprograms from miscellaneous public domain sources

ALBETA	ALGAMS	ALNGAM	ALNREL	BETAI	CSEVL	DBETAI	DCSEVL	DERF
DERFC	DGAMI	DGAMIT	DGAMLM	DGAMMA	DGAMR	DLBETA	DLGAMS	DLNGAM
DLNREL	D9GMIT	D9LGIC	D9LGIT	D9LGMC	EPRINT	ERF	ERFC	E9RINT
FDUMP	GAMI	GAMIT	GAMLIM	GAMMA	GAMR	INITDS	INITS	I8SAVE
J4SAVE	R9GMIT	R9LGIC	R9LGIT	R9LGMC	SETERR	S88FMT	XERABT	XERCLR
XERCTL	XERPRT	XERROR	XERRWV	XERSAV	XGETF	XGETUA	XSETF	

D.4 Subprograms from LINPACK and BLAS

DASUM	DAXPY	DCOPY	DDOT	DNRM2	DSCAL	DSIDI	DSIFA	DSWAP
DTRCO	DTRDI	IDAMAX	ISAMAX	SASUM	SAXPY	SCOPY	SDOT	SNRM2
SSCAL	SSIDI	SSIFA	SSWAP	STRCO	STRDI			

D.5 Subprograms specifying machine dependent constants

D1MACH I1MACH R1MACH

Appendix E

LIST OF STARPAC LABELED COMMON NAMES

CSTAK ERRCHK NOTOPT

<E-1>

REFERENCES

Abramowitz, M.; Stegun, I. (1964). Handbook of mathematical functions. Nat. Bur. Stand. (U.S.) Appl. Math. Ser. 55.

Akaike, H. (1974). A new look at statistical model identification. IEEE Transactions on Automatic Control, AC-19.

American National Standards Institute (1977). ANS FORTRAN X3.9-1977. New York, NY: American National Standards Institute.

Anderson, T. W. (1958). An introduction to multivariate statistical analysis. New York, NY: John Wiley and Sons.

Anscombe, F. J.; Tukey, J. W. (1963). The examination and analysis of residuals. Technometrics, 5: 141-160.

Bard, Y. (1974). Nonlinear parameter estimation. New York, NY: Academic Press.

Belsley, D. A.; Kuh, E.; Welsch, R. E. (1980). Regression diagnostics. New York, NY: John Wiley and Sons.

Bement, T. R.; Williams, J. S. (1969). Variance of weighted regression estimators when sampling errors are independent and heteroscedastic. J. Amer. Statists. Assoc. 64: 1369-1382.

Bloomfield, P. (1976). Fourier analysis of time series: an introduction. New York, NY: John Wiley and Sons.

Box, G. E. P.; Jenkins, G. M. (1976). Time series analysis forecasting and control. San Francisco, CA: Holden-Day.

Bradley, J. V. (1968). Distribution-free statistical tests. Englewood Cliffs, NJ: Prentice-Hall.

Brownlee, K. A. (1965). Statistical theory and methodology in science and technology. Second edition. New York, NY: John Wiley and Sons.

Cook, D. R. (1977). Detection of influential observations in linear regression. *Technometrics*, 19: 15.

Crow, E. L.; Siddiqui, M. M. (1967). Robust estimation of locations. *J. Amer. Stat. Assoc.* 62: 353-389.

Daniel, C.; Wood, F. S. (1980). *Fitting equations to data*. Second edition. New York, NY: John Wiley and Sons.

Davis, P. J. (1962). Orthonormalization codes in numerical analysis. In *Survey of Numerical Analysis*, J. Todd, ed. New York, NY: McGraw-Hill.

Dennis, J. E. Jr.; Gay, D. M.; Welsch, R. E. (1981a). An adaptive nonlinear least-squares algorithm. *ACM Trans. Math. Software*, 7(3): 348-368.

Dennis, J. E. Jr.; Gay, D. M.; Welsch, R. E. (1981b). Algorithm 573: NL2SOL - an adaptive nonlinear least squares algorithm [E4]. *ACM Trans. Math. Software*, 7(3): 369-383.

<REF-1>

Dennis, J. E. Jr.; Schnabel, R. B. (1983). *Numerical methods for unconstrained optimization and nonlinear equations*. Englewood Cliffs, NJ: Prentice-Hall.

Dixon, W. J.; Massey, F. J. Jr. (1957). *Introduction to statistical analysis*. Second edition. New York, NY: McGraw-Hill.

Donaldson, J. R.; Schnabel, R. B. (1987). Computational experience with confidence regions and confidence intervals for nonlinear least squares. *Technometrics*, 29: 67-82.

Donaldson, J. R.; Tryon, P. V. (1983a). Introduction to STARPAC, the Standards time series and regression package. *Nat. Bur. Stand. (U.S.) Tech. Note* 1068-1.

Donaldson, J. R.; Tryon, P. V. (1983b). Nonlinear Least Squares Regression Using STARPAC, the Standards time series and regression package. *Nat. Bur. Stand. (U.S.) Tech. Note* 1068-2.

Dongarra, J. J.; Moler, C. B.; Bunch, J. R.; Stewart, G. W. (1979). *LINPACK users' guide*. Philadelphia, PA: SIAM.

Draper, N. R.; Smith, H. (1981). *Applied regression analysis*. Second edition. New York, NY: John Wiley and Sons.

Duncan, A. J. (1965). *Quality control and industrial statistics*. Third edition. Richard D. Irwin.

Eisenhart, C. (1947). Significance of the largest of a set of sample estimates of variance. In *Techniques of Statistical Analysis*. C. Eisenhart, M. W. Hastay and W. A. Wallis, eds. New York, NY: McGraw-Hill.

Filliben, J. J. (1977). User's guide to the DATAPAC data analysis package. (Unpublished - available from NIST Statistical Engineering Division/Gaithersburg.)

- Fisher, R. A. (1950). Statistical methods for research workers. Eleventh edition. Edenburg: Oliver and Boyd.
- Fox, P. A.; Hall, A. D.; Schryer, N. L. (1978a). Algorithm 528: framework for a portable library [Z]. ACM Trans. Math. Software, 4(2): 177-188.
- Fox, P. A.; Hall, A. D.; Schryer, N. L. (1978b). The PORT mathematical subroutine library. ACM Trans. Math. Software, 4(2): 104-126.
- Freund, J. E.; Williams, F. J. (1958). Modern business statistics. Englewood Cliffs, NJ: Prentice-Hall.
- Fullerton, L. W. (1977). Portable special function routines. In Portability of Numerical Software: Proceedings. Lecture Notes in Computer Science: Vol. 57, W. Crowell, editor. Oak Brook, IL: Springer-Verlag.
- Gill, P. E.; Murray, W.; Wright, M. H. (1981). Practical optimization. New York, NY: Academic Press.
- Hald, A. (1952). Statistical theory with engineering applications. New York, NY: John Wiley and Sons.
- Hoaglin, D. C.; Welsh, R. E. (1978). The hat matrix in regression and ANOVA. American Statistician, 32: 17.
- Hogben, D.; Peavy, S. T.; Varner, R. N. (1971). OMNITAB II user's reference manual. Nat. Bur. Stand. (U.S.) Tech. Note 552.
- Jenkins, G. M.; Watts, D. C. (1968). Spectral analysis and its applications. San Francisco, CA: Holden-Day.
- Jones, R. H. (1971). Spectrum estimation with missing observations. Annals of the Institute of Statistical Mathematics, 23: 3.
- Kendall, M. G. (1948). Rank correlation methods. London: Charles Griffen and Co.
- Kendall, M. G.; Stuart, A. (1973). The advanced theory of statistics. Inference and Relationship, Vol. 2: London: Charles Griffin and Co.
- Kennedy, W. J. Jr.; Gentle, J. E. (1980). Statistical computing. New York, NY: Marcel-Dekker Inc.
- Ku, H. H. (1973). A user's guide to the OMNITAB command "STATISTICAL ANALYSIS." Nat. Bur. Stand. (U.S.) Tech. Note 756.
- Lawson, C.; Hanson, R.; Kincaid, D.; Krogh, F. (1979). Basic linear algebra subprograms for FORTRAN usage. ACM Trans. Math. Software, 5(3): 308-371.
- Mandel, J. (1964). The statistical analysis of experimental data. New York, NY: Interscience.
- Marsaglia, G.; Tsang, W. W. (1984). A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions. SIAM J. Sci. Stat. Comput. 5 (2): 349-358.

Miller, I.; Freund, J. E. (1977). Probability and statistics for engineers. Second edition. Englewood Cliffs, NJ: Prentice-Hall.

Morrison, D. F. (1967). Multivariate statistical methods. New York, NY: McGraw-Hill.

Natrella, M. G. (1966). Experimental statistics. Nat. Bur. Stand. (U.S.) Handb. 91.

Owen, D. B. (1962). Handbook of statistical tables. Reading, MA: Addison-Wesley Publishing Co.

Ryan, T. A.; Joiner, B. L.; Ryan, B. F. (1974). Student handbook for the MINITAB statistical computing system. The Pennsylvania State University.

Schnabel, R. B. (1982). Finite difference derivatives - theory and practice. (Unpublished - available from NIST Statistical Engineering Division/Boulder.)

Singleton, R. C. (1969). An algorithm for computing the mixed radix fast Fourier transform. IEEE Transactions on Audio and Electroacoustics, 17: 2.

Snedecor, G. W. (1956). Statistical methods. Fifth edition. Ames, IA: Iowa State University Press.

Snedecor, G. W.; Cochran, W. G. (1967). Statistical methods. Sixth edition. Ames, IA: Iowa State University Press.

Tryon, P. V.; Donaldson, J. R. (1978). STATLIB: A library of Fortran subroutines for statistical analysis of experimental data. (Unpublished - available from NIST Statistical Engineering Division/Boulder.)

Waldmeier, M. (1961). The Sunspot activity in the years 1610-1960. Zurich: Schulthess.

Walsh, P. J. (1962). Algorithm 127 - ORTHO. Comm. Assoc. Comp. Mach. 5: 511-513.