

REFERENCE

NBS  
PUBLICATIONS

NBSIR 86-3448

A11102 631417

NATL INST OF STANDARDS & TECH R.I.C.



A11102631417

Donaldson, Janet R/STARPAC : the standa  
QC100 .U56 NO.86-3448 1987 V19 C.1 NBS-P

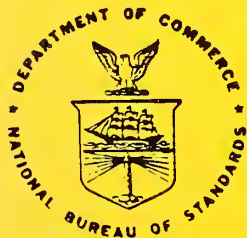
# The Standards Times Series and Regression Package

---

Janet R. Donaldson  
Peter V. Tryon

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
Boulder, Colorado 80303

January 1987



U.S. DEPARTMENT OF COMMERCE

N-QC ——— U OF STANDARDS

100

.U56

86-3448

1987



NBS  
RESEARCH  
INFORMATION  
CENTER

NBSR

QC100

.US6

NO. 86-3448

1987

NBSIR 86-3448

**STARPAC**

**THE STANDARDS TIMES SERIES AND  
REGRESSION PACKAGE**

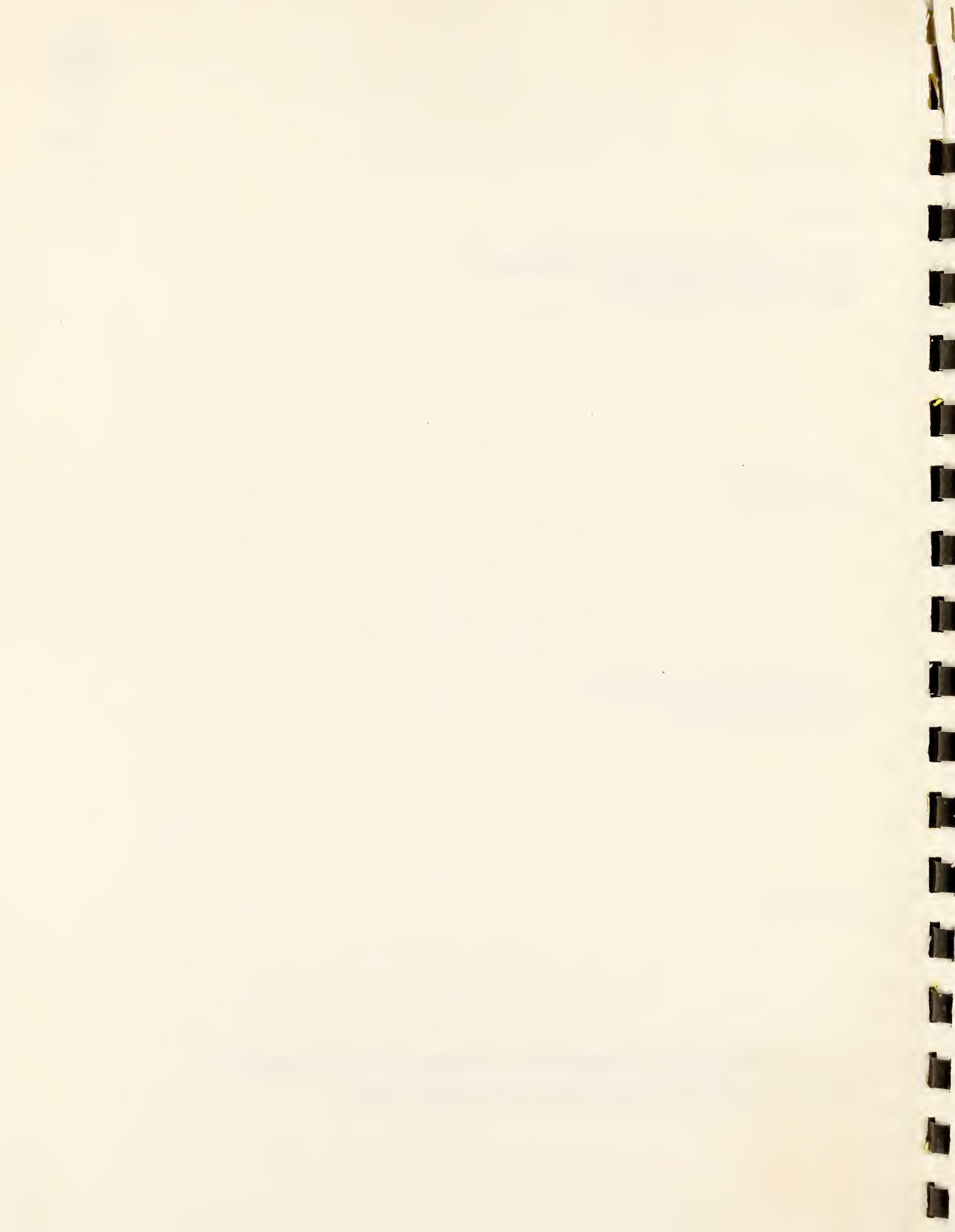
---

Janet R. Donaldson  
Peter V. Tryon

U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards  
Boulder, Colorado 80303

January 1987

**U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary***  
**NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director***





In memory of  
Peter V. Tryon  
1941 - 1982

### 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 Bureau of Standards, 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 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.

The code for STARPAC was developed at the U.S. Department of Commerce Boulder Laboratories on a CDC CYBER 180/840 under NOS 2.3. All examples presented in this documentation were executed in this environment using the FTN 5.1+617 compiler with rounding.

STARPAC is written in ANSI Fortran '77 [American National Standards Institute, 1977] using the Hollerith Extension. 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 M. 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

# Contents

Preface.....	v
1. INTRODUCTION TO USING STARPAC.....	1-1
A. Overview of STARPAC and Its Contents.....	1-1
B. Documentation Conventions.....	1-1
C. A Sample Program.....	1-2
D. Using STARPAC.....	1-6
D.1 The PROGRAM Statement.....	1-6
D.2 The Dimension Statements.....	1-6
D.3 The CALL Statements.....	1-8
D.4 STARPAC Output.....	1-8
D.5 STARPAC Error Handling.....	1-9
D.6 Common Programming Errors When Using STARPAC.....	1-10
2. LINE PRINTER GRAPHICS.....	2-1
A. Introduction.....	2-1
B. Subroutine Descriptions.....	2-1
C. Subroutine Declaration and CALL Statements.....	2-2
D. Dictionary of Subroutine Arguments and COMMON Variables....	2-11
E. Computational Details.....	2-14
F. Examples.....	2-15
3. NORMAL RANDOM NUMBER GENERATION.....	3-1
A. Introduction.....	3-1
B. Subroutine Descriptions.....	3-1
C. Subroutine Declaration and CALL Statements.....	3-1
D. Dictionary of Subroutine Arguments.....	3-2
E. Computational Methods.....	3-3
F. Example.....	3-3
G. Acknowledgments.....	3-6
4. HISTOGRAMS.....	4-1
A. Introduction.....	4-1
B. Subroutine Descriptions.....	4-1
C. Subroutine Declaration and CALL Statements.....	4-1
D. Dictionary of Subroutine Arguments.....	4-2
E. Computational Methods.....	4-3
E.1 Algorithms.....	4-3
E.2 Computed Results and Printed Output.....	4-3
F. Example.....	4-5
G. Acknowledgments.....	4-8



5.	STATISTICAL ANALYSIS OF A UNIVARIATE SAMPLE.....	5-1
	A. Introduction.....	5-1
	B. Subroutine Descriptions.....	5-1
	C. Subroutine Declaration and CALL Statements.....	5-1
	D. Dictionary of Subroutine Arguments and COMMON Variables....	5-3
	E. Computational Methods.....	5-8
	E.1 Algorithms.....	5-8
	E.2 Computed Results and Printed Output.....	5-8
	F. Example.....	5-8
	G. Acknowledgments.....	5-11
6.	ONE-WAY ANALYSIS OF VARIANCE.....	6-1
	A. Introduction.....	6-1
	B. Subroutine Descriptions.....	6-1
	C. Subroutine Declaration and CALL Statements.....	6-1
	D. Dictionary of Subroutine Arguments and COMMON Variables....	6-2
	E. Computational Methods.....	6-3
	E.1 Algorithms.....	6-3
	E.2 Computed Results and Printed Output.....	6-3
	F. Example.....	6-6
	G. Acknowledgments.....	6-9
7.	CORRELATION ANALYSIS.....	7-1
	A. Introduction.....	7-1
	B. Subroutine Descriptions.....	7-1
	C. Subroutine Declaration and CALL Statements.....	7-1
	D. Dictionary of Subroutine Arguments and COMMON Variables....	7-2
	E. Computational Methods.....	7-3
	E.1 Algorithms.....	7-3
	E.2 Computed Results and Printed Output.....	7-3
	F. Example .....	7-7
	G. Acknowledgments.....	7-11
8.	LINEAR LEAST SQUARES.....	8-1
	A. Introduction.....	8-1
	B. Subroutine Descriptions.....	8-2
	C. Subroutine Declaration and CALL Statements.....	8-2
	D. Dictionary of Subroutine Arguments and COMMON Variables....	8-5
	E. Computational Methods.....	8-9
	E.1 The Linear Least Squares Algorithm.....	8-9
	E.2 Computed Results and Printed Output.....	8-9
	F. Examples.....	8-12
	G. Acknowledgments.....	8-23

9.	NONLINEAR LEAST SQUARES.....	9-1
	A. Introduction.....	9-1
	B. Subroutine Descriptions.....	9-2
	B.1 Nonlinear Least Squares Estimation Subroutines.....	9-2
	B.2 Derivative Step Size Selection Subroutines.....	9-3
	B.3 Derivative Checking Subroutines.....	9-4
	C. Subroutine Declaration and CALL Statements.....	9-4
	D. Dictionary of Subroutine Arguments and COMMON Variables....	9-10
	E. Computational Methods.....	9-20
	E.1 Algorithms.....	9-20
	E.1.a Nonlinear Least Squares Estimation.....	9-20
	E.1.b Derivative Step Size Selection.....	9-22
	E.1.c Derivative Checking.....	9-24
	E.2 Computed Results and Printed Output.....	9-25
	E.2.a The Nonlinear Least Squares Estimation	
	Subroutines.....	9-25
	E.2.b The Derivative Step Size Selection Subroutines.	9-28
	E.2.c The Derivative Checking Subroutines.....	9-28
	F. Examples .....	9-29
	G. Acknowledgments.....	9-49
10.	DIGITAL FILTERING.....	10-1
	A. Introduction.....	10-1
	B. Subroutine Descriptions.....	10-1
	B.1 Symmetric Linear Filter Subroutines.....	10-1
	B.2 Autoregressive or Difference Linear Filter	
	Subroutines.....	10-3
	B.3 Gain and Phase Function Subroutines.....	10-4
	C. Subroutine Declaration and CALL Statements.....	10-4
	D. Dictionary of Subroutine Arguments and COMMON Variables....	10-8
	E. Computational Methods.....	10-14
	E.1 Algorithms.....	10-14
	E.2 Computed Results and Printed Output.....	10-14
	F. Examples.....	10-14
	G. Acknowledgments.....	10-20
11.	COMPLEX DEMODULATION.....	11-1
	A. Introduction.....	11-1
	B. Subroutine Descriptions.....	11-1
	C. Subroutine Declaration and CALL Statements.....	11-2
	D. Dictionary of Subroutine Arguments and COMMON Variables....	11-2
	E. Computational Methods.....	11-4
	E.1 Algorithms.....	11-4
	E.2 Computed Results and Printed Output.....	11-5
	F. Example.....	11-5
	G. Acknowledgments.....	11-9

12.	CORRELATION AND SPECTRUM ANALYSIS.....	12-1
	A. Introduction.....	12-1
	B. Subroutine Descriptions.....	12-1
	B.1 Correlation Analysis.....	12-2
	B.1.a Univariate Series.....	12-2
	B.1.b Bivariate Series.....	12-3
	B.2 Spectrum Estimation.....	12-4
	B.2.a Univariate Series.....	12-4
	B.2.b Bivariate Series.....	12-7
	C. Subroutine Declaration and CALL Statements.....	12-7
	D. Dictionary of Subroutine Arguments and COMMON Variables....	12-20
	E. Computational Methods.....	12-37
	E.1 Algorithms.....	12-37
	E.1.a Correlation Analysis.....	12-37
	E.1.b Spectrum Analysis.....	12-37
	E.2 Computed Results and Printed Output.....	12-38
	E.2.a Correlation Analysis.....	12-38
	E.2.b Spectrum Analysis.....	12-41
	F. Examples.....	12-44
	G. Acknowledgments.....	12-77
13.	ARIMA MODELING.....	13-1
	A. Introduction.....	13-1
	B. Subroutine Descriptions.....	13-3
	B.1 ARIMA Estimation Subroutines.....	13-3
	B.2 ARIMA Forecasting Subroutines.....	13-3
	C. Subroutine Declaration and CALL Statements.....	13-4
	D. Dictionary of Subroutine Arguments and COMMON Variables....	13-5
	E. Computational Methods.....	13-14
	E.1 Algorithms.....	13-14
	E.1.a ARIMA Estimation.....	13-14
	E.1.b ARIMA Forecasting.....	13-15
	E.2 Computed Results and Printed Output.....	13-15
	E.2.a The ARIMA Estimation Subroutines.....	13-15
	E.2.b The ARIMA Forecasting Subroutines.....	13-18
	F. Examples.....	13-18
	G. Acknowledgments.....	13-29
	Appendix A.....	A-1
	Appendix B.....	B-1
	Appendix C.....	C-1
	References.....	R-1



## STARPAC

### The Standards Time Series and Regression Package

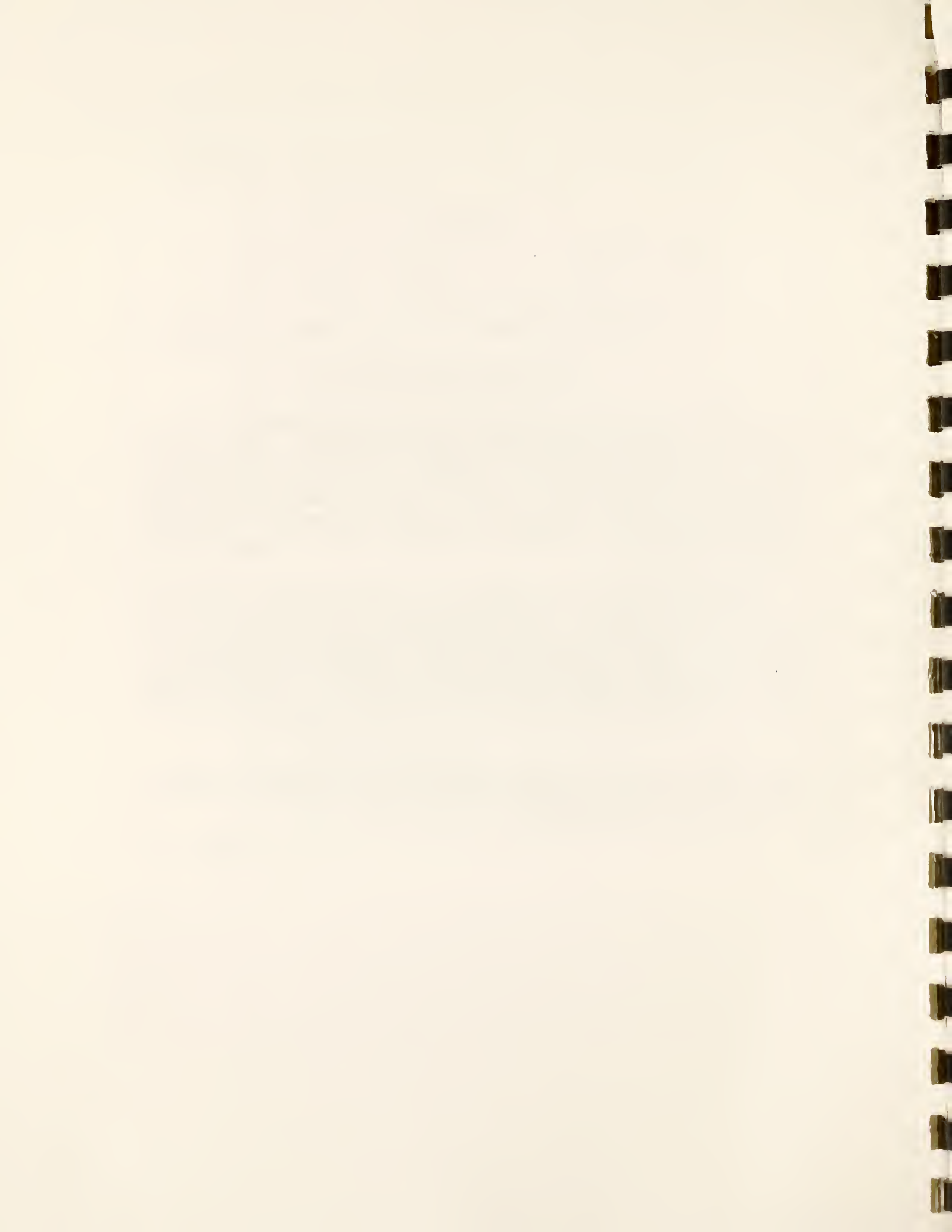
Janet R. Donaldson and Peter V. Tryon

National Bureau of Standards  
Washington, DC 20234

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 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 has been developed and is maintained by the Statistical Engineering Division of the National Bureau of Standards, 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  
National Bureau of Standards  
Mail Code 714  
325 Broadway  
Boulder, CO 80303.

#### 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 are made using the symbol §, and 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 C-2 refers to the second figure in §C 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 §D.2.

Currently, only the single precision version of the STARPAC library is supported by the Statistical Engineering Division. However, a double precision version can be generated relatively easily. 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 in figure C-1 illustrates the use of STARPAC subroutines. This example was executed on the CDC CYBER 180/840 at the U.S. Department of Commerce Boulder Laboratories under the NOS 2.3 operating system using a single precision version of the STARPAC library. 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 (fig. C-2) and STARPAC subroutine STAT, documented in chapter 4, prints a comprehensive statistical analysis of the data (fig. C-3).



```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE STAT AND PP USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION 0.4]
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
C REAL Y(100), X(100)
C DOUBLE PRECISION DSTAK(100)
C
C COMMON /CSTAK/ DSTAK
C COMMON /ERRCHK/ IERR
C
C DEFINE LDSTAK, THE LENGTH OF DSTAK
C
C LDSTAK = 100
C
C READ NUMBER OF OBSERVATIONS INTO N AND
C DATA INTO VECTOR Y
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C CREATE A VECTOR OF ORDER INDICES IN X
C
C DO 10 I=1,N
C   X(I) = I
C 10 CONTINUE
C
C PRINT TITLE, PLOT OF DATA AND ERROR INDICATOR
C
C WRITE (6, 102)
C CALL PP (Y, X, N)
C WRITE (6, 103) IERR
C
C PRINT TITLE, STATISTICAL ANALYSIS OF DATA AND ERROR INDICATOR
C
C WRITE (6, 102)
C CALL STAT (Y, N, LDSTAK)
C WRITE (6, 103) IERR
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I5)
C 101 FORMAT (12F7.4)
C 102 FORMAT ('DAVIS-HARRISON PIKES PEAK RELATIVE HUMIDITY DATA')
C 103 FORMAT (' IERR = ', I1)
C
C ENO
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 0.6300 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 0.6330 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

```

Figure C-1  
STARPAC sample program and data



STATISTICAL ANALYSIS

N = 04

FREQUENCY DISTRIBUTION (1-6) 5 25 35 0 1 0 0 4 4 2

MEASURES OF LOCATION (2-2)

UNWEIGHTED MEAN = 6.3734048E-01  
 WEIGHTED MEAN = 6.3734048E-01  
 MEDIAN = 6.2915000E-01  
 MID-RANGE = 6.6845000E-01  
 25 PCT UNWTO TRIMMED MEAN = 6.288992E-01  
 25 PCT WTO TRIMMED MEAN = 6.288992E-01

MEASURES OF DISPERSION (2-6)

WTD STANDARD DEVIATION = 3.240213E-02  
 WEIGHTED S.O. OF MEAN = 3.5356987E-03  
 RANGE = 1.4670000E-01  
 MEAN DEVIATION = 2.1076417E-02  
 VARIANCE = 1.0500979E-03  
 COEF. OF. VAR. (PERCENT) = 5.0844330E+00

A TWO-SIDED 95 PCT CONFIDENCE INTERVAL FOR MEAN IS 6.3030811E-01 TO 6.4437284E-01 (2-2)  
 A TWO-SIDED 99 PCT CONFIDENCE INTERVAL FOR S.D. IS 2.8137113E-02 TO 3.8211736E-02 (2-7)

LINEAR TREND STATISTICS (3-1)

SLOPE = -2.4736661E-04  
 S.D. OF SLOPE = 1.4414086E-04  
 SLOPE/S.D. OF SLOPE = T = -1.7161490E+00  
 PROB EXCEEDING ABS VALUE OF OBS T = .090

OTHER STATISTICS

MINIMUM = 5.9510000E-01  
 MAXIMUM = 7.4180000E-01  
 BETA ONE = 3.7288258E+00  
 BETA TWO = 5.9283926E+00  
 WTD SUM OF VALUES = 5.3536600E+01  
 WTD SUM OF SQUARES = 3.4208200E+01  
 WTD SUM OF DEV SQUARED = 8.7158122E-02  
 STUDENTS T = 1.8029871E+02  
 WTD SUM ABSOLUTE VALUES = 5.3536600E+01  
 WTD AVE ABSOLUTE VALUES = 6.3734048E-01

TESTS FOR NON-RANDOMNESS

NO. OF RUNS UP AND DOWN = 47  
 EXPECTED NO. OF RUNS = 99.7  
 S.D. OF NO. OF RUNS = 3.82  
 MEAN SQ SUCCESSIVE DIFF = 3.6382337E-04  
 MEAN SQ SUCC DIFF/VAR = .346

DEVIATIONS FROM WTD MEAN

NO. OF + SIGNS = 22  
 NO. OF - SIGNS = 62  
 NO. OF RUNS = 14  
 EXPECTED NO. OF RUNS = 33.5  
 S.D. OF RUNS = 3.51  
 DIFF./S.D. OF RUNS = -5.950

NOTE - ITEMS IN PARENTHESES REFER TO PAGE NUMBER IN NDS HANDBOOK 91 (MATRELLA, 1966)  
 IERR = 0

Figure C-3

Example of STAT output

## 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, italicized 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  $\langle \text{real} \rangle XM(n,m)$ , 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 §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 plot as shown in figure C-2. 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 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.<sup>†</sup> 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, §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.)

---

<sup>†</sup>STARPAC output must be directed to a separate output device [see §D.4] when users do not want any STARPAC reports displayed under any conditions.



## 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 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 2

### LINE PRINTER GRAPHICS

#### A. Introduction

STARPAC contains 36 subroutines for producing 2 basic styles of line printer plots.

The first, called a page plot, uses a single 11 x 14 inch page of line printer paper.

The second, called a vertical plot, is designed for plotting time series. The user specifies only the y-axis values since the x-axis values (independent variable) are assumed to be equally spaced and ordered consecutively. The independent variable in the resulting plot is oriented vertically and the plot will continue over as many pages as necessary to plot one point per line.

Within these two basic styles the user has many options, including controlling the plot symbol, plotting multivariate values, designating missing data, using log scales and specifying plot limits and plot size.

Users are directed to §B for a brief description of the subroutines. The actual declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

#### B. Subroutine Descriptions

PP (Page Plot) and VP (Vertical Plot) are the simplest of the STARPAC line printer plot subroutines. For both, plot limits are automatically set by the range of the data and other control parameters are set to the default values given in §D. The remaining plotting subroutines are named by adding letters to the beginning and/or end of PP and VP.

- Prefix S (e.g., SPP) indicates the user controls the plot symbol for each point.
- Prefix M indicates the subroutine will accept multivariate y-axis values (e.g., MPP).
- Suffix M subroutines allow data with missing observations to be plotted (e.g., VPM).
- Suffix L indicates log scales can be used (e.g., PPL).
- Suffix C subroutines allow control of the parameters which specify the plot limits, size, scale, etc. (e.g., VPC).

The following table, which indicates the capabilities of each of the STARPAC plotting subroutines, can be used to select from the available subroutines. Subroutine declaration and CALL statements are given in §C, listed in the same order as in the table.

Name	Plot						
	Symbol Control (S)	Multiple Y-Axis (M)	Page Plot (PP)	Vertical Plot (VP)	Missing Data (M)	Log Scale (L)	Control Parameters (C)
PP			✓				
PPL			✓			✓	
PPC			✓			✓	✓
PPM			✓		✓		
PPML			✓		✓	✓	
PPMC			✓		✓	✓	✓
SPP	✓		✓				
SPPL	✓		✓			✓	
SPPC	✓		✓			✓	✓
SPPM	✓		✓		✓		
SPPML	✓		✓		✓	✓	
SPPMC	✓		✓		✓	✓	✓
MPP		✓	✓				
MPPL		✓	✓			✓	
MPPC		✓	✓			✓	✓
MPPM		✓	✓		✓		
MPPML		✓	✓		✓	✓	
MPPMC		✓	✓		✓	✓	✓
VP				✓			
VPL				✓		✓	
VPC				✓		✓	✓
VPM				✓	✓		
VPML				✓	✓	✓	
VPMC				✓	✓	✓	✓
SVP	✓			✓			
SVPL	✓			✓		✓	
SVPC	✓			✓		✓	✓
SVPM	✓			✓	✓		
SVPML	✓			✓	✓	✓	
SVPMC	✓			✓	✓	✓	✓
MVP		✓		✓			
MVPL		✓		✓		✓	
MVPC		✓		✓		✓	✓
MVPM		✓		✓	✓		
MVPML		✓		✓	✓	✓	
MVPMC		✓		✓	✓	✓	✓

### C. Subroutine Declaration and CALL Statements

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

Page Plots

*PP: Print Y versus X scatterplot; linear axes; default control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n)
:
CALL PP (Y, X, N)
```

===

*PPL: Print Y versus X scatterplot; log or linear axes; default control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n)
:
CALL PPL (Y, X, N, ILOG)
```

===

*PPC: Print Y versus X scatterplot; log or linear axes; user-supplied control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n), YLB, YUB, XLB, XUB
:
CALL PPC (Y, X, N, ILOG, ISIZE, NOUT, YLB, YUB, XLB, XUB)
```

===

*PPM: Print Y versus X scatterplot; linear axes; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS
:
CALL PPM (Y, YMISS, X, XMISS, N)
```

===

*PPML: Print Y versus X scatterplot; log or linear axes; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS
:
CALL PPML (Y, YMISS, X, XMISS, N, ILOG)
```

===



*PPMC: Print Y versus X scatterplot; log or linear axes; user-supplied control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS, YLB, YUB, XLB, XUB
:
CALL PPMC (Y, YMISS, X, XMISS, N, ILOG, ISIZE, NOUT, YLB, YUB,
1 XLB, XUB)
```

===

*SPP: Print Y versus X scatterplot with individual plot symbols specified by user; linear axes; default control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n)
INTEGER ISYM(n)
:
CALL SPP (Y, X, N, ISYM)
```

===

*SPPL: Print Y versus X scatterplot with individual plot symbols specified by user; log or linear axes; default control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n)
INTEGER ISYM(n)
:
CALL SPPL (Y, X, N, ISYM, ILOG)
```

===

*SPPC: Print Y versus X scatterplot with individual plot symbols specified by user; log or linear axes; user-supplied control values and axis limits; no missing values allowed*

```
<real> Y(n), X(n), YLB, YUB, XLB, XUB
INTEGER ISYM(n)
:
CALL SPPC (Y, X, N, ISYM, ILOG, ISIZE, NOUT, YLB, YUB, XLB, XUB)
```

===



*SPPM: Print Y versus X scatterplot with individual plot symbols specified by user; linear axes; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS  
INTEGER ISYM(n)  
:  
CALL SPPM (Y, YMISS, X, XMISS, N, ISYM)
```

===

*SPPML: Print Y versus X scatterplot with individual plot symbols specified by user; log or linear axis; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS  
INTEGER ISYM(n)  
:  
CALL SPPML (Y, YMISS, X, XMISS, N, ISYM, ILOG)
```

===

*SPPMC: Print Y versus X scatterplot with individual plot symbols specified by user; log or linear axes; user-supplied control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, X(n), XMISS, YLB, YUB, XLB, XUB  
INTEGER ISYM(n)  
:  
CALL SPPMC (Y, YMISS, X, XMISS, N, ISYM, ILOG, ISIZE, NOUT, YLB,  
1 YUB, XLB, XUB)
```

===

*MPP: Print plot of multiple Y vectors versus a common X vector; linear axes; default control values and axis limits; no missing values allowed*

```
<real> YM(n,m), X(n)  
:  
CALL MPP (YM, X, N, M, IYM)
```

===

*MPPL: Print plot of multiple Y vectors versus a common X vector; log or linear axes; default control values and axis limits; no missing values allowed*

```
<real> YM(n,m), X(n)
:
CALL MPPL (YM, X, N, M, IYM, ILOG)
===
```

*MPPC: Print plot of multiple Y vectors versus a common X vector; log or linear axes; user-supplied control values and axis limits; no missing values allowed*

```
<real> YM(n,m), X(n), YLB, YUB, XLB, XUB
:
CALL MPPC (YM, X, N, M, IYM, ILOG, ISIZE, NOUT, YLB, YUB, XLB,
1 XUB)
===
```

*MPPM: Print plot of multiple Y vectors versus a common X vector; linear axes; default control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m), X(n), XMISS
:
CALL MPPM (YM, YMMISS, X, XMISS, N, M, IYM)
===
```

*MPPML: Print plot of multiple Y vectors versus a common X vector; log or linear axes; default control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m), X(n), XMISS
:
CALL MPPML (YM, YMMISS, X, XMISS, N, M, IYM, ILOG)
===
```

*MPPMC: Print plot of multiple Y vectors versus a common X vector; log or linear axes; user-supplied control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m), X(n), XMISS, YLB, YUB, XLB, XUB
:
CALL MPPMC (YM, YMMISS, X, XMISS, N, M, IYM, ILOG, ISIZE, NOUT,
1 YLB, YUB, XLB, XUB)
===
```

## Vertical Plots

*VP: Print vertical plot of Y versus input order; linear axes; default control values and axis limits; no missing values allowed*

```
<real> Y(n)
:
CALL VP (Y, N, NS)
```

===

*VPL: Print vertical plot of Y versus input order; log or linear horizontal (Y) axis; default control values and axis limits; no missing values allowed*

```
<real> Y(n)
:
CALL VPL (Y, N, NS, ILOG)
```

===

*VPC: Print vertical plot of Y versus input order; log or linear horizontal (Y) axis; user-supplied control values and axis limits; no missing values allowed*

```
<real> Y(n), YLB, YUB, XLB, XINC
:
CALL VPC (Y, N, NS, ILOG, ISIZE, IRLIN, IBAR, YLB, YUB, XLB,
1 XINC)
```

===

*VPM: Print vertical plot of Y versus input order; linear axis; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS
:
CALL VPM (Y, YMISS, N, NS)
```

===

*VPML: Print vertical plot of Y versus input order; log or linear horizontal (Y) axis; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS
:
CALL VPML (Y, YMISS, N, NS, ILOG)
```

===

*VPMC: Print vertical plot of Y versus input order; log or linear horizontal (Y) axis; user-supplied control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, YLB, YUB, XLB, XINC
:
CALL VPMC (Y, YMISS, N, NS, ILOG, ISIZE, IRLIN, IBAR, YLB, YUB,
1  XLB, XINC)
===
```

*SVP: Print vertical plot of Y versus input order with individual plot symbols specified by user; linear axis; default control values and axis limits; no missing values allowed*

```
<real> Y(n)
INTEGER ISYM(n)
:
CALL SVP (Y, N, NS, ISYM)
===
```

*SVPL: Print vertical plot of Y versus input order with individual plot symbols specified by user; log or linear horizontal (Y) axis; default control values and axis limits; no missing values allowed*

```
<real> Y(n)
INTEGER ISYM(n)
:
CALL SVPL (Y, N, NS, ISYM, ILOG)
===
```

*SVPC: Print vertical plot of Y versus input order with individual plot symbols specified by user; log or linear horizontal (Y) axis; user-supplied control values and axis limits; no missing values allowed*

```
<real> Y(n), YLB, YUB, XLB, XINC
INTEGER ISYM(n)
:
CALL SVPC (Y, N, NS, ISYM, ILOG, ISIZE, IREFLN, IBAR, YLB, YUB,
1  XLB, XINC)
===
```

*SVPM: Print vertical plot of Y versus input order with individual plot symbols specified by user; linear axis; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS
INTEGER ISYM(n)
:
CALL SVPM (Y, YMISS, N, NS, ISYM)
```

===

*SVPML: Print vertical plot of Y versus input order with individual plot symbols specified by user; log or linear horizontal (Y) axis; default control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS
INTEGER ISYM(n)
:
CALL SVPML (Y, YMISS, N, NS, ISYM, ILOG)
```

===

*SVPMC: Print vertical plot of Y versus input order with individual plot symbols specified by user; log or linear horizontal (Y) axis; user-supplied control values and axis limits; missing values allowed*

```
<real> Y(n), YMISS, YLB, YUB, XLB, XINC
INTEGER ISYM(n)
:
CALL SVPMC (Y, YMISS, N, NS, ISYM, ILOG, ISIZE, IRLIN, IBAR, YLB,
1 YUB, XLB, XINC)
```

===

*MVP: Print vertical plot of multiple Y vectors versus input order; linear axis; default control values and horizontal (Y) axis limits; no missing values allowed*

```
<real> YM(n,m)
:
CALL MVP (YM, N, M, IYM, NS)
```

===



*MVPL: Print vertical plot of multiple Y vectors versus input order; log or linear horizontal (Y) axis; default control values and axis limits; no missing values allowed*

```
<real> YM(n,m)
:
CALL MVPL (YM, N, M, IYM, NS, ILOG)
```

===

*MVPC: Print vertical plot of multiple Y vectors versus input order; log or linear horizontal (Y) axis; user-supplied control values and axis limits; no missing values allowed*

```
<real> YM(n,m), YLB, YUB, XLB, XINC
:
CALL MVPC (YM, N, M, IYM, NS, ILOG, ISIZE, YLB, YUB, XLB, XINC)
```

===

*MVPM: Print vertical plot of multiple Y vectors versus input order; linear axis; default control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m)
:
CALL MVPM (YM, YMMISS, N, M, IYM, NS)
```

===

*MVPML: Print vertical plot of multiple Y vectors versus input order; log or linear horizontal (Y) axis; default control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m)
:
CALL MVPML (YM, YMMISS, N, M, IYM, NS, ILOG)
```

===

*MVPMC: Print vertical plot of multiple Y vectors versus input order; log or linear horizontal (Y) axis; user-supplied control values and axis limits; missing values allowed*

```
<real> YM(n,m), YMMISS(m), YLB, YUB, XLB, XINC
:
CALL MVPMC (YM, YMMISS, N, M, IYM, NS, ILOG, ISIZE, YLB, YUB,
1 XLB, XINC)
```

===

## 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.

IBAR --> The indicator variable used to designate whether a vertical plot is to be a bar plot or not. Bar plots connect the plotted points to the reference line [see argument IRLIN] with a string of plot symbols. [See, e.g., chapter 12, figure F-1c.] If  $IBAR \geq 1$ , the plot is a bar plot. If  $IBAR < 0$ , it is not. The default value is  $IBAR = 0$ . When IBAR is not an argument in the subroutine CALL statement the default value is used.

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

IERR = 0 indicates that no errors were detected and that the plot was completed satisfactorily.

IERR = 1 indicates that improper input was detected or that some error prevented the plot from being completed.

ILOG --> The indicator variable used to designate whether the axes are to be on a log or linear scale. ILOG is a two-digit integer, pq, where the value of p is used to designate the scale of the x-axis and the value of q is used to designate the scale of the y-axis. If  $p = 0$  ( $q = 0$ ) the x-axis (y-axis) is on a linear scale; if  $p \neq 0$  ( $q \neq 0$ ) the x-axis (y-axis) is on a log scale. For vertical plots, the value of q is used to specify the scale on the horizontal-axis and the value of p is ignored. The default value is  $ILOG = 0$ , corresponding to linear scale for both the x-axis and the y-axis. When ILOG is not an argument in the subroutine CALL statement the default value is used.

IRLIN --> The indicator variable used to designate whether zero or the series mean is to be plotted as a reference line on the vertical plots or whether no reference line should be used. If  $IRLIN \leq -1$ , no reference line is plotted. If  $IRLIN = 0$ , a reference line is plotted showing the location of zero on the plot. If  $IRLIN > 1$ , a reference line is plotted showing the series mean. The default value is  $IRLIN = -1$ . When IRLIN is not an argument in the subroutine CALL statement the default value is used.

ISIZE --> The indicator variable used to designate the size of a page plot. ISIZE is a two-digit integer, pq, where the value of p is used to designate the size of the x-axis and the value of q is used to designate the size of the y-axis. If p = 0 (q = 0) the x-axis (y-axis) is the maximum possible, 101 columns (51 rows), i.e., 101 (51) plot positions. If p ≠ 0 (q ≠ 0) the x-axis (y-axis) is half the maximum, or 51 columns (26 rows). For vertical plots, the value of q is used to specify the size of the horizontal-axis and the value of p is ignored. The default value is ISIZE = 0, corresponding to a plot of 51 rows by 101 columns. When ISIZE is not an argument in the subroutine CALL statement the default value is used.

ISYM --> The vector of dimension at least N that contains the values designating the plotting symbol to be used for each point. The plot symbols designated by each possible integer value are given below.

ISYM(•) ≤ 1 → +	ISYM(•) = 9 → E	ISYM(•) = 16 → L	ISYM(•) = 23 → S
ISYM(•) = 2 → •	ISYM(•) = 10 → F	ISYM(•) = 17 → M	ISYM(•) = 24 → T
ISYM(•) = 3 → *	ISYM(•) = 11 → G	ISYM(•) = 18 → N	ISYM(•) = 25 → U
ISYM(•) = 4 → -	ISYM(•) = 12 → H	ISYM(•) = 19 → O	ISYM(•) = 26 → V
ISYM(•) = 5 → A	ISYM(•) = 13 → I	ISYM(•) = 20 → P	ISYM(•) = 27 → W
ISYM(•) = 6 → B	ISYM(•) = 14 → J	ISYM(•) = 21 → Q	ISYM(•) = 28 → Y
ISYM(•) = 7 → C	ISYM(•) = 15 → K	ISYM(•) = 22 → R	ISYM(•) ≥ 29 → Z
ISYM(•) = 8 → D			

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

M --> The number of columns of data in YM.

N --> The number of observations.

NOUT --> The number of points falling outside the plot limits that are to be listed following the plot. If NOUT ≥ 1, a message giving the total number of points falling outside the plot limits and a list of the coordinates of these points (up to a maximum of NOUT or 50, whichever is smaller) is printed. If NOUT = 0, only a message listing the number of points falling outside the limits is printed. If NOUT < 0, no points are listed and no message is given. The default value is NOUT = 0. When NOUT is not an argument in the subroutine CALL statement the default value is used.

NS --> The sampling frequency of the points plotted on a vertical plot. If NS = 1, every point is plotted; if NS = 2, every second point is plotted; if NS = 3, every third point is plotted, etc. The default value is NS = 1. When NOUT ≤ 0 or NS is not an argument in the subroutine CALL statement the default value is used.



- X --> The vector of dimension at least N that contains the x-axis values.
- XINC --> The increment to be used for labeling the x-axis (i.e., the vertical-axis) on vertical plots. The x-axis labels are XLB, XLB + NS·XINC, XLB + 2·NS·XINC, etc. The default value is XINC = 1.0. When XINC is not an argument in the subroutine CALL statement the default value is used.
- XLB --> The lower bound for the x-axis.
- For page plots:
- The default value is the smallest x-axis value within the range of the y-axis values to be plotted. If XLB  $\geq$  XUB, the default value is used.
- For vertical plots:
- The default value is 1.0.
- For both page and vertical plots, when XLB is not an argument in the subroutine CALL statements the default value is used. (The plot limits may be adjusted slightly from the user-supplied values when the plotting subroutine uses a log scale.)
- XMISS --> The missing value code used within the vector X to indicate that a value is missing. The user must indicate missing observations by putting the missing value code in place of each missing observation. Missing data are not indicated on page plots in any way.
- XUB --> The upper bound for the x-axis. The default value is the largest x-axis value within the range of the y-axis values to be plotted. If XLB  $\geq$  XUB or XUB is not an argument in the subroutine CALL statement the default value is used. (The plot limits may be adjusted slightly from the user-supplied value when the plotting subroutines use a log scale.)
- Y --> The vector of dimension at least N that contains the y-axis values.
- YLB --> The lower bound for the y-axis. The default value is the smallest y-axis value within the range of the x-axis values to be plotted. If YLB  $\geq$  YUB or YLB is not an argument in the subroutine CALL statement the default value is used. (The plot limits may be adjusted slightly from the user-supplied value when the plotting subroutines use a log scale.)
- YM --> The matrix of dimension at least N by M whose columns each contain one of the M sets of N observations to be plotted against a common X vector.

- YMISS --> The missing value code used within the vector Y to indicate a value is missing. The user must indicate missing observations by putting the missing value code in place of each missing observation. Missing data are indicated on vertical plots by the word "MISSING" next to the right axis of the appropriate line. Missing data are not indicated on page plots in any way.
- YMMISS --> The vector of dimension at least M that contains the codes used within each of the M columns of YM to indicate a value is missing, where the first element of YMMISS is the missing value code for the first column of YM, etc. The user must indicate missing observations by putting the appropriate missing value code in place of each missing observation. Missing data are indicated on vertical plots by the word "MISSING" next to the right axis of the appropriate line. Missing data are not indicated on page plots in any way.
- YUB --> The upper bound for the y-axis. The default value is the largest y-axis value within the range of the x-axis values to be plotted. If YLB  $\geq$  YUB or YUB is not an argument in the subroutine CALL statement the default value is used. (The plot limits may be adjusted slightly from the user-supplied value when the plotting subroutines use a log scale.)

#### E. Computational Details

Plotting Symbols. The plotting symbol used depends on the type of plot and whether or not more than one point falls on a given plot position. If two to nine points fall on a single plot position, the integer corresponding to the number of points is used as the plotting symbol. When 10 or more values fall on a single position the plotting symbol X is used. This is the only way that integers or X are used as plot symbols.

Subroutines without an S or M prefix use the plotting symbol + to indicate one point on a single printer position.

For subroutines with an S prefix the user-supplied vector ISYM of integer values is used to specify the plotting symbol for each data point. The Fourier spectrum plot shown in chapter 12, figure F-3b, is an example of this option.

Subroutines with an M prefix use a different letter as the plot symbol for each column of the matrix of the dependent variables (y-axis): A for the first, B for the second, ..., Z for columns 25 and beyond, with X still only used to indicate that 10 or more points fell on a single plot position.

Continuity of Vertical Plots. Normally, a line printer will automatically provide margins at the top and bottom of each page, causing a break in the continuity of a vertical plot or any other output continuing over two or more pages. However, these automatic page-ejects can be suppressed by the user on many systems. Appendix A gives the control sequence necessary to suppress automatic page-ejects on a Cyber computer. Users of other systems should consult their Computer Center staff for any equivalent method available.

#### F. Examples

The example program of figure F-1a uses MPP to display the 12 years of monthly airline data listed on page 531 of Box and Jenkins [1976] versus month. The year is indicated by the plotting symbol (A = 1949, B = 1950, etc.). Figure F-1b shows the output from this program.

Other examples of STARPAC plots can be found in the output of many of the subroutines discussed elsewhere. The output from the complex demodulation subroutines includes a simple vertical plot (VP) (chapter 11, figure F-1b) and a vertical plot of multivariate data (MVPC) (chapter 11, figure F-1c); the output from the autocorrelation and cross correlation subroutines includes vertical plots using the bar plot option (VPC) (chapter 12, figure F-1c); and the output from the Fourier spectrum subroutines (chapter 12, figure F-3b) is produced using a symbol plot (SPPC).

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE MPP USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF X AND YM MUST BE CHANGED TO DOUBLE PRECISION IF
C DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL X(20), YM(20,20)
C
C SPECIFY NECESSARY DIMENSIONS
C
C IYM = 20
C
C READ NUMBER OF OBSERVATIONS AND NUMBER OF COLUMNS OF DATA
C X-AXIS VALUES
C Y-AXIS VALUES
C
C READ 100, N, M
C READ 101, (X(I), I=1,M)
C READ 101, ((YM(I,J), I=1,N), J=1,M)
C
C PRINT TITLE AND CALL MPP FOR PLOT
C
C WRITE (6, 102)
C CALL MPP (YM, X, N, M, IYM)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I5)
C 101 FORMAT (12F6.1)
C 102 FORMAT ('!RESULTS OF STARPAC PLOT SUBROUTINE MPP')
C END

DATA:      12  12
           1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  10.0  11.0  12.0
112.0 118.0 132.0 129.0 121.0 135.0 148.0 148.0 136.0 119.0 104.0 118.0
115.0 126.0 141.0 135.0 125.0 149.0 170.0 170.0 158.0 133.0 114.0 140.0
145.0 150.0 178.0 163.0 172.0 178.0 199.0 199.0 184.0 162.0 146.0 166.0
171.0 180.0 193.0 181.0 183.0 218.0 230.0 242.0 209.0 191.0 172.0 194.0
196.0 196.0 236.0 235.0 229.0 243.0 264.0 272.0 237.0 211.0 180.0 201.0
204.0 186.0 235.0 227.0 234.0 264.0 302.0 293.0 259.0 229.0 203.0 229.0
242.0 233.0 267.0 269.0 270.0 315.0 364.0 347.0 312.0 274.0 237.0 278.0
284.0 277.0 317.0 313.0 318.0 374.0 413.0 405.0 355.0 306.0 271.0 306.0
319.0 301.0 356.0 348.0 355.0 422.0 465.0 467.0 404.0 347.0 305.0 336.0
340.0 318.0 362.0 348.0 363.0 435.0 491.0 505.0 404.0 359.0 310.0 337.0
360.0 342.0 406.0 396.0 420.0 472.0 548.0 559.0 463.0 407.0 362.0 405.0
417.0 391.0 419.0 461.0 472.0 535.0 622.0 606.0 508.0 461.0 390.0 432.0

```

Figure F-1a

Example program using MPP



RESULTS OF STARPAC PLOT SUBROUTINE MPP

STARPAC 3.03S - DECEMBER 2, 1985

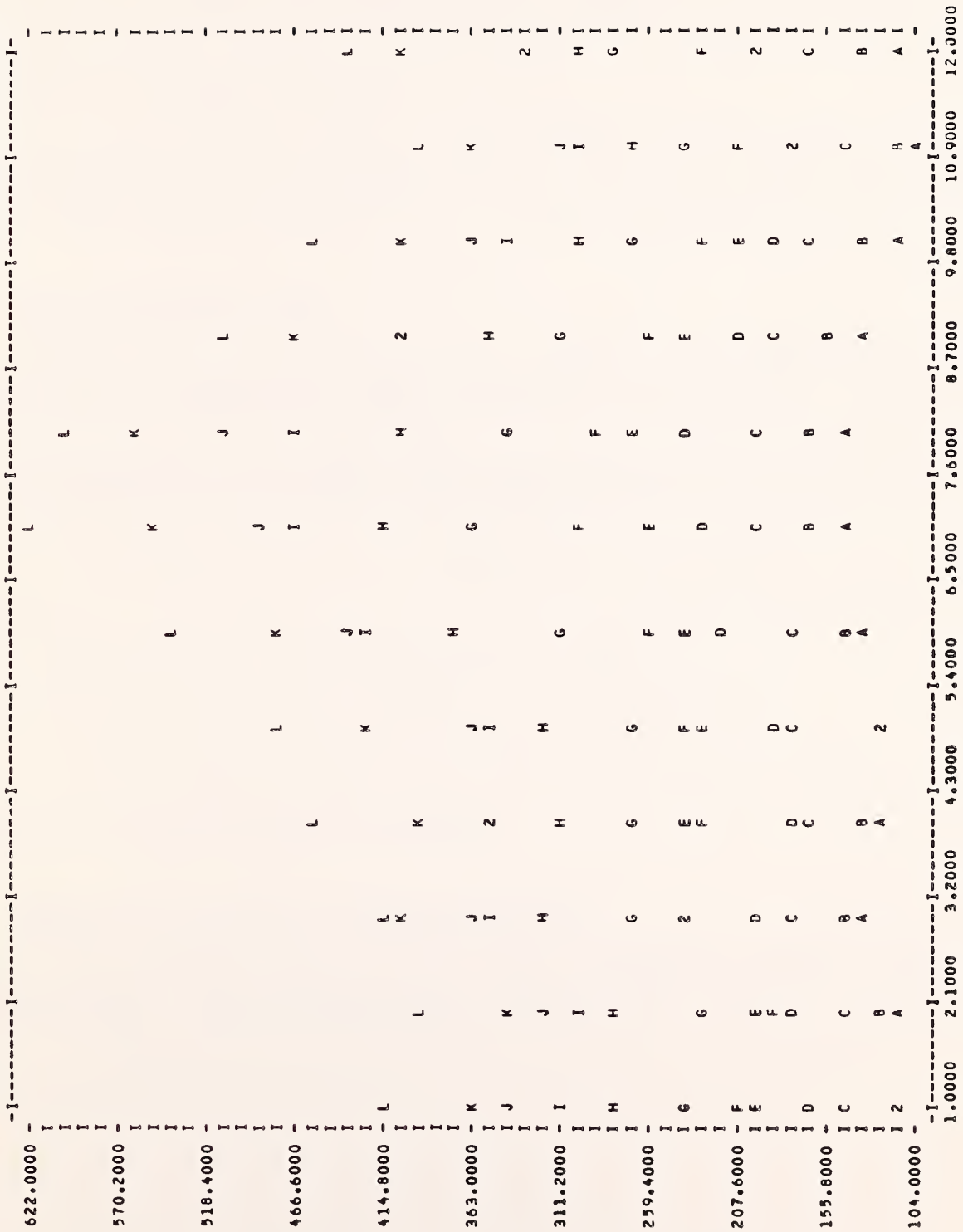
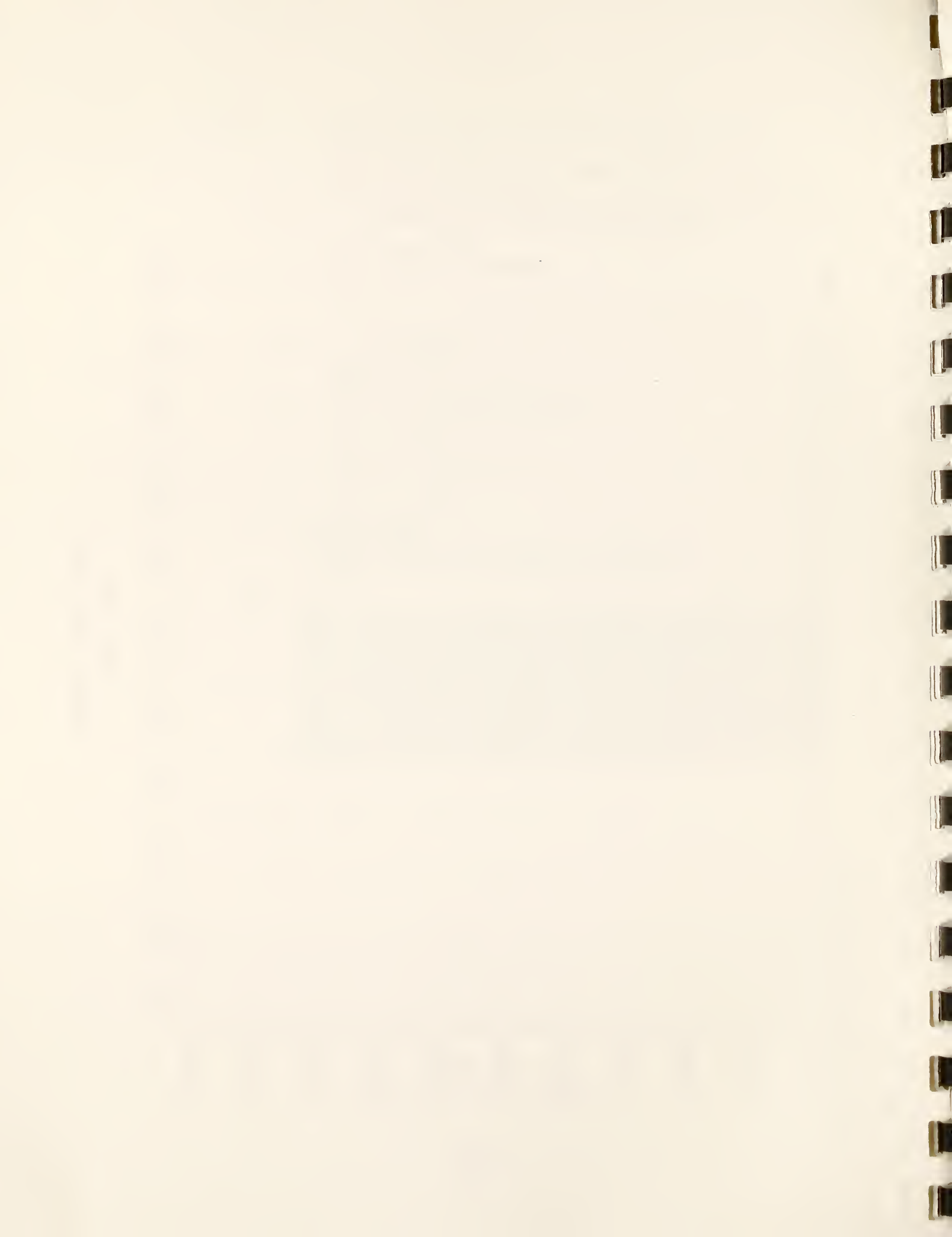


Figure F-1b  
Example of MPP output



## CHAPTER 3

### NORMAL RANDOM NUMBER GENERATION

#### A. Introduction

STARPAC contains two subroutines for generating pseudo-random numbers (noise) which obey a normal probability law with mean  $\mu$  and standard deviation  $\sigma$ . Such random numbers are often useful for evaluating data analysis procedures or computer programs.

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithm used by these subroutines is discussed in §E. A sample program showing the use of these subroutines is given in §F.

#### B. Subroutine Descriptions

STARPAC subroutine NRAND generates a vector of standard (zero mean and unit standard deviation) normal (Gaussian) random numbers. There is no printed output from this subroutine.

STARPAC subroutine NRANDC generates Gaussian noise with mean  $\mu$  and standard deviation  $\sigma$  using the transformation

$$z = \sigma y + \mu$$

where

$y$  is a standard normal psuedo-random number;

$\mu$  is the desired mean (see §D, argument YMEAN); and

$\sigma$  is the desired standard deviation (see §D, argument SIGMA).

There is no printed output from NRANDC.

#### C. Subroutine Declaration and CALL Statements

NOTE: Argument definitions and a sample program are given in §D and §F, respectively. The conventions used to present the following declaration and CALL statments are given in chapter 1, §B and D.

*NRAND: Generate a vector of normal pseudo-random numbers with zero mean and unit standard deviation*

```
<real> Y(n)
:
CALL NRAND (Y, N, ISEED)
```

===

*NRANDC: Generate a vector of normal pseudo-random numbers with mean YMEAN and standard deviation SIGMA*

```
<real> Y(n)
:
CALL NRANDC (Y, N, ISEED, YMEAN, SIGMA)
```

#### D. Dictionary of Subroutine Arguments

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.

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

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

ISEED --> The basis for the pseudo-random number generation.

For ISEED  $\neq$  0, use of the same value of ISEED will always generate the same data set.

For ISEED = 0, a different data set will be produced by each call to NRAND or NRANDC in the user's program, although the numbers generated will not differ from run to run.

N --> The number of random numbers to be generated.



SIGMA --> The standard deviation of the generated random numbers.

Y <-- The vector of dimension at least N that contains the generated normal pseudo-random numbers.

YMEAN --> The mean of the generated random numbers.

#### E. Computational Methods

The normal pseudo-random number generation procedure is that of Marsaglia and Tsang [1984]. The same pseudo-random numbers (to within final round-off error) will be generated on any computer. The code was written by Boisvert and Kahanar of the National Bureau of Standards Scientific Computing Division.

#### F. Example

The sample program shown in figure F-1a illustrates the use of both NRAND and NRANDC. NRAND is used to generate a standard normal pseudo-random sample of size 50 from a normal population with zero mean and unit standard deviation. NRANDC is then used to generate a sequence of normal pseudo-random numbers with a mean of 4 and standard deviation 0.5. The same seed is used for both NRAND and NRANDC. Therefore, the values generated by NRANDC are YMEAN plus SIGMA times the values generated by NRAND, i.e.,

$$YMEAN(I,2) = YMEAN + SIGMA*YMEAN(I,1) \text{ for } I = 1, \dots, N.$$

The generated random numbers are displayed using STARPAC plot subroutine MVP. Figure F-1b shows the output from MVP. There is no output from NRAND and NRANDC.

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE NRAND AND NRANDC AND DISPLAY RESULTS WITH MVP USING
C SINGLE PRECISION VERSION OF STARPAC RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF YM MUST BE CHANGED TO DOUBLE PRECISION IF
C DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL YM(100,2)
C
C SPECIFY NECESSARY DIMENSIONS
C
C IYM = 100
C
C SET THE SEED
C THE NUMBER OF VALUES TO BE GENERATED
C THE NUMBER OF SETS OF DATA TO BE GENERATED
C
C ISEED = 931
C N = 50
C M = 2
C
C GENERATE STANOARD NORMAL PSEUDO-RANDOM NUMBERS INTO COLUMN 1 OF YM
C
C CALL NRAND (YM(1,1), N, ISEED)
C
C GENERATE NORMAL PSEUDO-RANDOM NUMBERS WITH MEAN 4.0 AND
C STANOARD DEVIATION 0.5 INTO COLUMN 2 OF YM
C
C YMEAN = 4.0
C SIGMA = 0.5
C CALL NRANDC (YM(1,2), N, ISEED, YMEAN, SIGMA)
C
C PRINT TITLE AND CALL MVP TO PLOT RESULTS
C
C WRITE (6, 100)
C CALL MVP (YM, N, M, IYM)
C
C STOP
C
C FORMAT STATEMENTS
C
100 FORMAT ('RESULTS OF STARPAC NORMAL PSUEDO-RANDOM NUMBER',
1 ' GENERATION SUBROUTINES',
2 ' DISPLAYED WITH STARPAC PLOT SUBROUTINE MVP')
END

```

Figure F-1a

Example program using NRAND, NRANDC and MVP

RESULTS OF STARPAC NORMAL PSUE00-RANDOM NUMBER GENERATION SUBROUTINES DISPLAYED WITH STARPAC PLOT SURROUTINE MVP  
 STARPAC 2.03S - DECEMBER 2, 1965  
 -2.7388 -1.9718 -1.2049 -.4380 .3289 1.0959 1.8628 2.6297 3.3967 4.1636 4.9305



Figure F-1b

Example of NRAND and NRANDC results displayed using MVP

G. Acknowledgments

The code used to generate the pseudo-random numbers was written by Boisvert and Kahanar of the National Bureau of Standards Scientific Computing Division.



## CHAPTER 4

### HISTOGRAMS

#### A. Introduction

STARPAC contains two subroutines for producing histograms. Both subroutines produce a one-page printout which includes, in addition to the histogram, a number of summary statistics (mean, median, standard deviation, cell fractions, etc.) and several tests for normality.

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

#### B. Subroutine Descriptions

HIST provides the analysis described in §A using a preset procedure for choosing the number of cells. The lower and upper bounds of the histogram are chosen from the range of the observations.

HISTC provides the same analysis as HIST but allows the user to specify the number of cells and the upper and lower cell boundaries. Statistics are based only on the data within the user-supplied bounds.

#### C. Subroutine Declaration and CALL Statements

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

*HIST: Compute and print a histogram and summary statistics, with automatic selection of number of cells and cell boundaries*

```
<real> Y(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL HIST (Y, N, LDSTAK)
```

===

*HISTC: Compute and print a histogram and summary statistics with user control of number of cells and cell boundaries*

```
<real> Y(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL HISTC (Y, N, NCELL, YLB, YUB, LDSTAK)
```

#### D. Dictionary of Subroutine Arguments

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.

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.

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

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

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, §B.]

For HIST:  $LDSTAK \geq (17+N)/2 + 26 \cdot P$

For HISTC:  $LDSTAK \geq (17+N)/2 + \max(NCELL, 26) \cdot P$

N --> The number of observations.

NCELL --> The number of cells in the histogram. If  $NCELL < 0$  or NCELL is not an argument in the subroutine CALL statement the subroutine will choose the number of cells.

- Y --> The vector of dimension at least N that contains the observed data.
- YLB --> The lower bound for constructing the histogram. The interval [YLB, YUB] is divided into NCELL increments. If YLB  $\geq$  YUB, the lower and upper bounds for constructing the histogram will be the minimum and maximum observations.
- YUB --> The upper bound for constructing the histogram. The interval [YLB, YUB] is divided into NCELL increments. If YLB  $\geq$  YUB, the lower and upper bounds for constructing the histogram will be the minimum and maximum observations.

## E. Computational Methods

### E.1 Algorithms

The code and output for the histogram subroutines are modeled after early versions of MINITAB [Ryan et al. 1974].

### E.2 Computed Results and Printed Output

The output from the histogram family of subroutines includes a summary of the input data in addition to the actual histogram. This summary includes the following information, where the actual output headings are given by the underlined, uppercase phrases. Results which correspond to subroutine CALL statements arguments are identified by the argument name in uppercase (not underlined). In the formulas,  $x_1, x_2, \dots, x_k$  denotes the ordered observations of Y such that  $YLB < Y(i) < YUB, i = 1, \dots, N$ , i.e.,  $x_1$  is the smallest observation of Y such that  $YLB < Y(i)$ ,  $x_k$  is the largest observation of Y such that  $Y(i) < YUB$ , etc. The value of expressions enclosed in square brackets, e.g.,  $[(k/2) + 1]$ , is the largest integer less than or equal to the value of the expression.

- NUMBER OF OBSERVATIONS, N
- MINIMUM OBSERVATION
- MAXIMUM OBSERVATION
- HISTOGRAM LOWER BOUND, YLB
- HISTOGRAM UPPER BOUND, YUB
- NUMBER OF CELLS, NCELL
- OBSERVATIONS USED, k, where

k = the number of observations for which  
 $YLB < Y(i) < YUB, i = 1, \dots, N$

- MIN. OBSERVATION USED,  $x_1$ , where

$x_1 =$  the smallest observation such that  $YLB \leq Y(i)$ ,  $i = 1, \dots, N$

- MAX. OBSERVATION USED,  $x_k$ , where

$x_k =$  the largest observation such that  $Y(i) \leq YUB$ ,  $i = 1, \dots, N$

- MEAN VALUE,  $x_{\text{mean}}$ , where

$$x_{\text{mean}} = k^{-1} \sum_{i=1}^k x_i$$

- MEDIAN VALUE,  $x_{\text{median}}$ , where

$x_{\text{median}} = x_{[(k+1)/2]}$  if  $k$  is odd

$x_{\text{median}} = 0.5 \cdot (x_{[k/2]} + x_{[(k/2) + 1]})$  if  $k$  is even

- 25 PCT TRIMMED MEAN,  $x_{\text{trim}}$ , where

$$x_{\text{trim}} = [k - (2 \cdot [k/4])]^{-1} \sum_{i=1+[k/4]}^{k-[k/4]} x_i$$

- STANDARD DEVIATION,  $s$ , where

$$s = \left( (k-1)^{-1} \sum_{i=1}^k (x_i - x_{\text{mean}})^2 \right)^{1/2}$$

- MEAN DEV./STD. DEV.,  $r$ , where

$$r = (s \cdot k)^{-1} \sum_{i=1}^k |x_i - x_{\text{mean}}|$$

- SQRT(BETA ONE),  $\beta_1^{1/2}$ , where

$$\beta_1 = \left( (k-1)^3 \cdot s^6 \right)^{-1} \left( \sum_{i=1}^k (x_i - x_{\text{mean}})^3 \right)^2 \cdot k$$

- BETA TWO,  $\beta_2$ , where,

$$\beta_2 = \left( (k-1)^2 \cdot s^4 \right)^{-1} \sum_{i=1}^k (x_i - x_{\text{mean}})^4 \cdot k$$



Information provided for each cell,  $\xi = 1, \dots, \text{NCELL}$ , of the histogram includes the following.

- INTERVAL MID POINT,  $c_{\text{mid}}$ , where
$$c_{\text{mid}} = \text{YLB} + (\text{YUB} - \text{YLB}) / (2 \cdot \text{NCELL})$$
- CUM. FRACT., the cumulative fraction of the observations which are in cells 1 through  $\xi$
- 1-CUM. FRACT., the cumulative fraction of the observations which are in cells  $\xi$  through NCELL
- CELL FRACT., the fraction of the observations which are in cell  $\xi$
- NO. OBS., the actual number of observations which are in cell  $\xi$ .

The histogram itself displays the actual number of observations in each cell when the largest number of observations per cell does not exceed 50. When the largest number of observations per cell does exceed 50 then the histogram displays the cell fraction.

#### F. Example

The example program of figure F-1a uses HIST to analyze the 39 measurements of the velocity of light shown on page 81 of Mandel [1964]. The output from HIST is shown in figure F-1b.

```

MAIN PROGRAM:      PROGRAM EXAMPL
C                 DEMONSTRATE HIST USING SINGLE PRECISION VERSION OF STARPAC
C                 RUN ON CYBER 180/840.
C                 OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C                 [SEE CHAPTER 1, SECTION D.4]
C                 N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION IF
C                 DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C                 REAL Y(200)
C                 DOUBLE PRECISION DSTAK(200)
C
C                 COMMON /CSTAK/ DSTAK
C
C                 SPECIFY NECESSARY DIMENSIONS
C
C                 LDSTAK = 200
C
C                 READ NUMBER OF OBSERVATIONS
C                 OBSERVED DATA
C
C                 READ 100, N
C                 READ 101, (Y(I), I=1,N)
C
C                 PRINT TITLE AND CALL HIST TO ANALYZE RESULTS
C
C                 WRITE (6, 102)
C                 CALL HIST (Y, N, LDSTAK)
C
C                 STOP
C
C                 FORMAT STATEMENTS
C
100 FORMAT (I5)
101 FORMAT (13F5.1)
102 FORMAT ('RESULTS OF STARPAC HISTOGRAM SUBROUTINE HIST')
END
DATA:             39
0.4  0.6  1.0  1.0  1.0  0.5  0.6  0.7  1.0  0.6  0.2  1.9  0.2
0.4  0.0 -0.4 -0.3  0.0 -0.4 -0.3  0.1 -0.1  0.2 -0.5  0.3 -0.1
0.2 -0.2  0.8  0.5  0.6  0.8  0.7  0.7  0.2  0.5  0.7  0.8  1.1

```

Figure F-1a

Example program using HIST

RESULTS OF STARPAC HISTOGRAM SUBROUTINE HIST

HISTOGRAM

NUMBER OF OBSERVATIONS = 39  
 MINIMUM OBSERVATION = -5.0000000E-01  
 MAXIMUM OBSERVATION = 1.9000000E+00  
 HISTOGRAM LOWER BOUND = -5.0000000E-01  
 HISTOGRAM UPPER BOUND = 1.9000000E+00  
 NUMBER OF CELLS = 7  
 OBSERVATIONS USED = 39  
 MIN. OBSERVATION USED = -5.0000000E-01  
 MAX. OBSERVATION USED = 1.9000000E+00  
 MEAN VALUE = 4.10296410E-01  
 MEDIAN VALUE = 5.0000000E-01

25 PCT TRIMMED MEAN = 4.23809524E-01  
 STANDARD DEVIATION = 5.06689399E-01  
 MEAN DEV./STD. DEV. = 7.99040249E-01  
 SQRT(BETA ONE) = 3.10646614E-01  
 BETA TWO = 3.33793260E+00

FOR A NORMAL DISTRIBUTION, THE VALUES (MEAN DEVIATION/STANDARD DEVIATION), SQRT(BETA ONE), AND BETA TWO ARE APPROXIMATELY 0.8, 0.0 AND 3.0, RESPECTIVELY. TO TEST THE NULL HYPOTHESIS OF NORMALITY, SEE TABLES OF CRITICAL VALUES PP. 207-208, BIONOMETRIKA TABLES FOR STATISTICIANS, VOL. 1. SEE PP. 67-68 FOR A DISCUSSION OF THESE TESTS.

INTERVAL MID POINT	CUM. FRACT.	1-CUM. FRACT.	CELL FRACT.	NO. OBS.	0	10	20	30	40	50
-3.285714E-01	.154	1.000	.154	6	+++++					
1.428571E-02	.282	.846	.128	5	++++					
3.571429E-01	.564	.718	.282	11	+++++	+++++				
7.000000E-01	.846	.436	.282	11	+++++	+++++				
1.042857E+00	.974	.154	.128	5	++++					
1.385714E+00	.974	.026	.000	0						
1.728571E+00	1.000	.026	.026	1	+					

Figure F-1b  
Example of HIST output

G. Acknowledgments

The code and output for the histogram subroutines is modeled on that in early versions of MINITAB [Ryan et al. 1974].

## CHAPTER 5

### STATISTICAL ANALYSIS OF A UNIVARIATE SAMPLE

#### A. Introduction

STARPAC contains 4 subroutines for performing a comprehensive statistical analysis of a univariate sample. They each compute 53 different statistics which summarize the sample through measures of location (mean, median, etc), measures of dispersion (standard deviation, mean deviation, etc.) and diagnostic features such as tests for outliers, non-normality, trends and non-randomness (assuming the input order of the data is a meaningful time sequence). Common statistics such as Student's t and confidence intervals for the mean and standard deviation are also included. NBS Technical Note 756, A User's Guide to the OMNITAB Command "STATISTICAL ANALYSIS," by H. H. Ku [1973] provides a complete discussion of the output of these subroutines, which is the same output as that provided by the OMNITAB II Command STATISTICAL [Hogben et al. 1971].

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

#### B. Subroutine Descriptions

STAT computes and prints the 53 descriptive statistics described in §A.

STATS provides the same analysis as STAT but allows the user to suppress the printed output and store the computed statistics for further use.

STATW and STATWS perform a weighted analysis and are otherwise identical to STAT and STATS, respectively.

#### C. Subroutine Declaration and CALL Statements

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



*STAT: Compute and print 53 statistics describing the input data*

```
<real> Y(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL STAT (Y, N, LDSTAK)
```

===

*STATS: Compute and optionally print 53 statistics describing input data; return statistics*

```
<real> Y(n), STS(53)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL STATS (Y, N, LDSTAK, STS, NPRT)
```

===

*STATW: Compute and print 53 statistics describing weighted input data*

```
<real> Y(n), WT(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL STATW (Y, WT, N, LDSTAK)
```

===

*STATWS: Compute and optionally print 53 statistics describing weighted input data; return statistics*

```
<real> Y(n), WT(n), STS(53)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL STATWS (Y, WT, N, LDSTAK, STS, 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.

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.

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

LDSTAK --> The length of the DOUBLE PRECISION workspace vector DSTAK. LDSTAK must equal or exceed  $(N/2) + 7$ .

N --> The number of observations.

NPRT --> The argument controlling printed output.

If NPRT = 0 the printed output is suppressed.

If NPRT  $\neq$  0 the printed output is provided.

STS --> The vector of dimension at least 53 that contains the computed statistics. The contents of STS are listed below, along with applicable references; the number in parenthesis is the location within STS that the given statistic is stored. In the formulas,  $x$  denotes the ordered observations of  $Y$  for which the weight is nonzero, i.e.,  $x_1$  is the smallest observation of  $Y$  with a nonzero weight,  $x_k$  is the largest observation of  $Y$  with a nonzero weight, etc. The weight associated with  $x_i$  is denoted  $w_i$ . Zero weighted observations are not included in the analysis. The value of expressions enclosed in square brackets, e.g.,  $[(k/2) + 1]$ , is the largest integer less than or equal to the value of the expression.

(1) NUMBER OF OBSERVATIONS,  $N$

(2) NUMBER OF NONZERO WEIGHTS,  $k$

- (3) UNWEIGHTED MEAN [Dixon and Massey, 1957, p. 14],

$$x_{\text{mean}} = k^{-1} \sum_{i=1}^k x_i$$

- (4) WEIGHTED MEAN [Brownlee, 1965, pp. 95-97],

$$x_{\text{wtdmean}} = \frac{\sum_{i=1}^k (w_i \cdot x_i)}{\sum_{i=1}^k w_i}$$

- (5) MEDIAN [Dixon and Massey, 1957, p. 70],

$$x_{\text{median}} = x_{[(k+1)/2]} \text{ if } k \text{ is odd}$$

$$x_{\text{median}} = 0.5 \cdot (x_{[k/2]} + x_{[(k/2) + 1]}) \text{ if } k \text{ is even}$$

- (6) MID-RANGE [Dixon and Massey, 1957, p. 71],

$$x_{\text{mid}} = 0.5 \cdot (x_1 + x_k)$$

- (7) 25 PCT UNWTD TRIMMED MEAN [Crow and Siddiqui, 1967],

$$x_{\text{trim}} = [k - (2 \cdot [k/4])]^{-1} \sum_{i=1+[k/4]}^{k-[k/4]} x_i$$

- (8) 25 PCT WTD TRIMMED MEAN,

$$x_{\text{wtdtrim}} = \frac{\sum_{i=1+[k/4]}^{k-[k/4]} (w_i \cdot x_i)}{\sum_{i=1+[k/4]}^{k-[k/4]} w_i}$$

- (9) WEIGHTED STANDARD DEVIATION [Snedecor and Cochran, 1967, p. 44],

$$s = \left( (k-1)^{-1} \sum_{i=1}^k w_i \cdot (x_i - x_{\text{wtdmean}})^2 \right)^{1/2}$$

- (10) WTD S.D. OF MEAN [Brownlee, 1965, p. 80],

$$s_{\text{mean}} = s / \left( \sum_{i=1}^k w_i \right)^{1/2}$$

(11) RANGE [Snedecor and Cochran, 1967, p. 39],

$$x_{\text{range}} = x_k - x_1$$

(12) MEAN DEVIATION [Duncan, 1965, p. 50],

$$x_{\text{meandev}} = k^{-1} \sum_{i=1}^k |x_i - x_{\text{wtdmean}}|$$

(13) VARIANCE [Snedecor and Cochran, 1967, p. 44],

$$s^2 = (k-1)^{-1} \sum_{i=1}^k w_i \cdot (x_i - x_{\text{wtdmean}})^2$$

(14) COEF. OF VAR. (PERCENT) [Snedecor and Cochran, 1967, p. 62],

$$cvar = |100 \cdot s / x_{\text{wtdmean}}|$$

(15) LOWER CONFIDENCE LIMIT, MEAN [Natrella, 1966, pp. 2-2, 2-3],

$$x_{\text{mean}} - t_{0.025} \cdot s_{\text{mean}}$$

where  $t_{0.025}$  is the appropriate t-statistic with (k-1) degrees of freedom

(16) UPPER CONFIDENCE LIMIT, MEAN [Natrella, 1966, pp. 2-2, 2-3],

$$x_{\text{mean}} + t_{0.025} \cdot s_{\text{mean}}$$

where  $t_{0.025}$  is the appropriate t-statistic with (k-1) degrees of freedom

(17) LOWER CONFIDENCE LIMIT, S.D. [Natrella, 1966, p. 2-7],

$$s \cdot ((k-1) / \chi^2_{0.975})^{1/2}$$

where  $\chi^2_{0.975}$  is the appropriate chi-square statistic with (k-1) degrees of freedom

(18) UPPER CONFIDENCE LIMIT, S.D. [Natrella, 1966, p. 2-7],

$$s \cdot ((k-1) / \chi^2_{0.025})^{1/2}$$

where  $\chi^2_{0.025}$  is the appropriate chi-square statistic with (k-1) degrees of freedom

(19) SLOPE [Fisher, 1950, p. 136],

$$B = (k \cdot (k^2 - 1))^{-1} \sum_{i=1}^k i \cdot (x_i - x_{\text{wtdmean}})$$

(20) S.D. OF SLOPE,

$$s_B = \frac{\left[ \left( 12 \sum_{i=1}^k (x_i - x_{\text{wtdmean}})^2 \right) - B^2 \cdot k \cdot (k^2 - 1) \right]^{1/2}}{\left( k \cdot (k^2 - 1) \cdot (k - 2) \right)^{1/2}}$$

(21) SLOPE/S.D. OF SLOPE = T,

$$t_0 = B / s_B \text{ with } (k-2) \text{ degrees of freedom}$$

(22) PROB EXCEEDING ABS VALUE OF OBS T [Brownlee, 1965, p. 344],

$$\text{Pr}\{t < -|t_0| \text{ and } t > +|t_0|\}$$

(23) NO. OF RUNS UP AND DOWN [Brownlee, 1965, p. 223], r

(24) EXPECTED NO. OF RUNS [Bradley, 1965, p. 279],

$$E(r) = (2k-1)/3$$

(25) S.D. OF NO. OF RUNS [Bradley, 1965, p. 279],

$$r_{sd} = [(16k-29)/90]^{1/2}$$

(26) MEAN SQ SUCCESSIVE DIFF [Brownlee, 1965, p. 222],

$$D^2 = (k-1)^{-1} \sum_{i=1}^{k-1} (x_{i+1} - x_i)^2$$

(27) MEAN SQ SUCC DIFF/VAR [Brownlee, 1965, p. 222],

$$D^2 / s^2$$

(28) NO. OF + SIGNS,

$$u = \text{number of times sign of } (x_i - x_{\text{wtdmean}}) \text{ is positive}$$

(29) NO. OF - SIGNS,

$$v = \text{number of times sign of } (x_i - x_{\text{wtdmean}}) \text{ is negative}$$

(30) NO. OF RUNS [Brownlee, 1965, p. 224],

$$\text{RUNS} = 1 + \text{number of changes in sign of } (x_i - x_{\text{wtdmean}})$$

(31) EXPECTED NO. OF RUNS [Brownlee, 1965, p. 227],

$$E(\text{RUNS}) = 1 + (2 \cdot u \cdot v) / k$$



(32) S.D. OF RUNS [Brownlee, 1965, p. 230],

$$\text{RUNS}_{\text{sd}} = \left( [(u+v)^2 \cdot (k-1)]^{-1} \cdot 2 \cdot u \cdot v \cdot (2 \cdot u \cdot v - u - v) \right)^{1/2}$$

(33) DIFF./S.D. OF RUNS [Brownlee, 1965, p. 230],

$$[\text{RUNS} - E(\text{RUNS})] / \text{RUNS}_{\text{sd}}$$

(34) MINIMUM [Natrella, 1966, p. 19-1],

$x_1$  = smallest value with nonzero weight

(35) MAXIMUM [Natrella, 1966, p. 19-3],

$x_k$  = largest value with nonzero weight

(36) BETA ONE [Snedecor and Cochran, 1967, p. 86],

$$\beta 1 = \left( (k-1)^3 \cdot s^6 \right)^{-1} \left( \sum_{i=1}^k (x_i - x_{\text{wtdmean}})^3 \right)^2 \cdot k$$

(37) BETA TWO [Snedecor and Cochran, 1967, p. 87],

$$\beta 2 = \left( (k-1)^2 \cdot s^4 \right)^{-1} \sum_{i=1}^k (x_i - x_{\text{wtdmean}})^4 \cdot k$$

(38) WTD SUM OF VALUES,

$$\sum_{i=1}^k w_i \cdot x_i$$

(39) WTD SUM OF SQUARES,

$$\sum_{i=1}^k w_i \cdot x_i^2$$

(40) WTD SUM OF DEVS SQUARED,

$$\sum_{i=1}^k w_i \cdot (x_i - x_{\text{wtdmean}})^2$$

(41) STUDENT'S T [Brownlee, 1965, p. 296],

$$t = \left( \sum_{i=1}^k w_i \right)^{1/2} \cdot x_{\text{wtdmean}} / s \text{ with } (k-1) \text{ degrees of freedom}$$

(42) WTD SUM ABSOLUTE VALUES,

$$\sum_{i=1}^k w_i \cdot |x_i|$$

(43) WTD AVE ABSOLUTE VALUES,

$$\frac{\sum_{i=1}^k w_i \cdot |x_i|}{\sum_{i=1}^k w_i}$$

(44-53) FREQUENCY DISTRIBUTION [Freund and Williams, 1958, p. 17].

WT <-- The vector of dimension at least N that contains the weights. A zero weight excludes the corresponding observation from the analysis. If the weights are all equal to 1.0, the resulting analysis is equivalent to an unweighted analysis.

Y --> The vector of dimension at least N that contains the observations. The tests for trend and randomness will not be meaningful unless the sample is ordered with respect to time or some other relevant variable.

## E. Computational Methods

### E.1. Algorithms

Formulas for the computed statistics are given in §D under argument STS. The code for the statistical analysis subroutines is adapted from OMNITAB II [Hogben, et al. 1971].

### E.2. Computed Results and Printed Output

The output consists of a one-page display of the 53 statistics listed for argument STS in §D. The argument, NPRT, controlling the printed output is discussed in §D.

## F. Example

The example program of figure F-1a uses STAT to analyze the 39 measurements of the velocity of light shown on page 81 of Mandel [1964]. The output from STAT is shown in figure F-1b.

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE STAT USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION IF
C DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(200)
C DOUBLE PRECISION DSTAK(200)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED DATA
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C PRINT TITLE AND CALL STAT TO PERFORM STATISTICAL ANALYSIS
C
C WRITE (6, 102)
C CALL STAT (Y, N, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I5)
C 101 FORMAT (13F5.1)
C 102 FORMAT ('RESULTS OF STARPAC STATISTICAL ANALYSIS',
C * ' SUBROUTINE STAT')
C END
DATA:      39
0.4 0.6 1.0 1.0 1.0 0.5 0.6 0.7 1.0 0.6 0.2 1.9 0.2
0.4 0.0 -0.4 -0.3 0.0 -0.4 -0.3 0.1 -0.1 0.2 -0.5 0.3 -0.1
0.2 -0.2 0.8 0.5 0.6 0.8 0.7 0.7 0.2 0.5 0.7 0.8 1.1

```

Figure F-1a

Example program using STAT

RESULTS OF STARPAC STATISTICAL ANALYSIS SUBROUTINE STAT

STAPPAC 2.005 - DECEMBER 2, 1985

STATISTICAL ANALYSIS

N = 39

FREQUENCY DISTRIBUTION (1-6) 5 3 8 3 7 7 5 0 0 1

MEASURES OF LOCATION (2-2)

UNWEIGHTED MEAN = 4.1025641E-01  
 WEIGHTED MEAN = 4.1025641E-01  
 MEDIAN = 5.0000000E-01  
 MID-RANGE = 7.0000000E-01  
 25 PCT UNWTD TRIMMED MEAN = 4.2380952E-01  
 25 PCT WTD TRIMMED MEAN = 4.2380952E-01

MEASURES OF DISPERSION (2-6)

WTD STANDARD DEVIATION = 5.0668940E-01  
 WEIGHTED S.D. OF MEAN = 8.1135237E-02  
 RANGE = 2.4000000E+00  
 MEAN DEVIATION = 4.0486522E-01  
 VARIANCE = 2.5673414E-01  
 COEF. OF. VAR. (PERCENT) = 1.2350554E+02

A TWO-SIDED 95 PCT CONFIDENCE INTERVAL FOR MEAN IS 2.4600615E-01 TO 5.7450667E-01 (2-2)  
 A TWO-SIDED 95 PCT CONFIDENCE INTERVAL FOR S.D. IS 4.1408984E-01 TO 6.5301026E-01 (2-7)

LINEAR TREND STATISTICS (5-1)

SLOPE = -4.0080972E-03  
 S.D. OF SLOPE = 7.2760495E-03  
 SLOPE/S.D. OF SLOPE = T = -5.5086172E-01  
 PROB EXCEEDING ABS VALUE OF OBS T = .505

OTHER STATISTICS

MINIMUM = -5.0000000E-01  
 MAXIMUM = 1.9000000E+00  
 BETA ONE = 9.6501319E-02  
 BETA TWO = 3.3379326E+00  
 WTD SUM OF VALUES = 1.6000000E+01  
 WTD SUM OF SQUARES = 1.6320000E+01  
 WTD SUM OF DEV SQUARED = 9.7559974E+00  
 STUDENTS T = 5.054517E+00  
 WTD SUM ABSOLUTE VALUES = 2.0600000E+01  
 WTD AVE ABSOLUTE VALUES = 5.2820513E-01

TESTS FOR NON-RANDOMNESS

NO. OF RUNS UP AND DOWN = 23  
 EXPECTED NO. OF RUNS = 25.7  
 S.D. OF NO. OF RUNS = 2.57  
 MEAN SQ SUCCESSIVE DIFF = 2.8289474E-01  
 MEAN SQ SUCC DIFF/VAR = 1.102

DEVIATIONS FROM WTD MEAN

NO. OF + SIGNS = 20  
 NO. OF - SIGNS = 19  
 NO. OF RUNS = 8  
 EXPECTED NO. OF RUNS = 20.5  
 S.D. OF RUNS = 3.08  
 DIFF./S.D. OF RUNS = -4.056

NOTE - ITEMS IN PARENTHESES REFER TO PAGE NUMBER IN NOS HANDBOOK 91 (MARELLA, 1966)

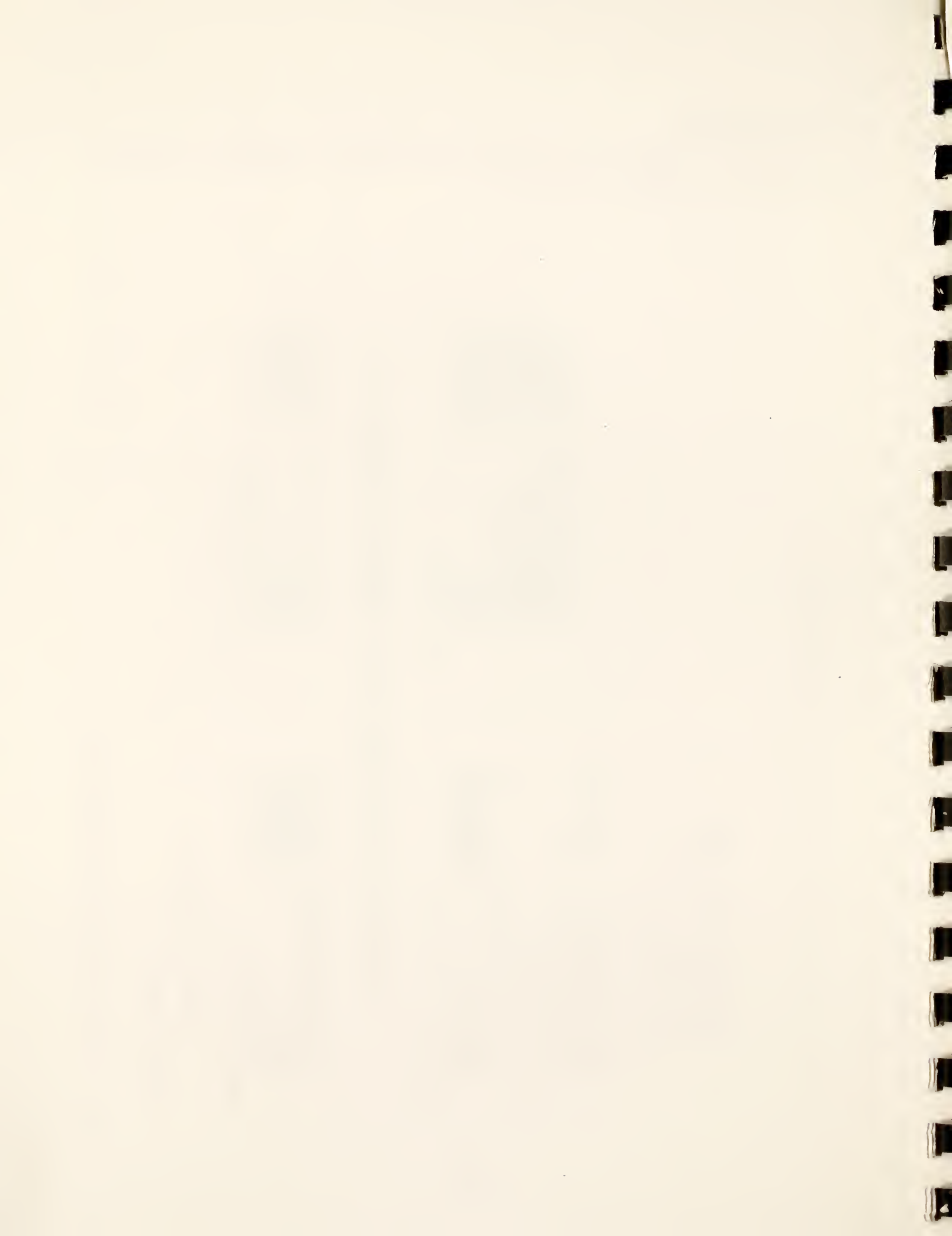
Figure F-1b

Example of STAT output

G. Acknowledgments

The code and output for the statistical analysis subroutines is adapted from OMNITAB II [Hogben et al. 1971].





## CHAPTER 6

### ONE-WAY ANALYSIS OF VARIANCE

#### A. Introduction

STARPAC contains two subroutines for one-way analysis of variance. The output from these subroutines includes the usual analysis of variance table plus the robust Kruskal-Wallis rank test. Comprehensive summary statistics are also given, including means, standard deviations, standard deviations of the mean and confidence intervals for the mean of each group. Within group standard deviations, standard deviations of the mean and 95-percent confidence intervals for the mean are given assuming fixed effects and random effects models and also assuming ungrouped data. The output also includes pair-wise multiple comparisons using the Newman-Keuls and Scheffe techniques; the Cochran's C, the Bartlett-Box F and variance ratio tests for homogeneity of variances; and the random effects model components of variance estimate. The analysis performed by these subroutines is the same as that performed by the OMNITAB II command ONEWAY [Hogben et al. 1971]. A reference for one-way analysis of variance is Brownlee [1965], chapter 10.

Users are directed to \$B for a brief description of the subroutines. The declaration and CALL statements are given in \$C and the subroutine arguments are defined in \$D. The algorithms used and output produced by these subroutines are discussed in \$E. Sample programs and their output are shown in \$F.

#### B. Subroutine Descriptions

Subroutine AOV1 computes and prints the one-way analysis of variance described in \$A.

Subroutine AOV1S provides the same analysis as AOV1 but allows the user to suppress the printed output and to store the number of observations in each group, the group means and the group standard deviations.

#### C. Subroutine Declaration and CALL Statements

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

AOV1: *Compute and print a one-way analysis of variance of the input data*

```
REAL Y(n), TAG(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL AOV1 (Y, TAG, N, LDSTAK)
```

===

AOV1S: *Compute and optionally print a one-way analysis of variance of the input data; return tag value of each group, number of observations in each group, group averages, and group standard deviations*

```
REAL Y(n), TAG(n), GSTAT(igstat,4)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL AOV1S (Y, TAG, N, LDSTAK, NPRT, GSTAT, IGSTAT, NG)
```

#### 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.

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.

GSTAT <-- The matrix of dimension at least NG by four whose columns contain, in order, the tag value of the group, number of observations in the group, group average and group standard deviation. The groups are in order of ascending tag values.

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

- IGSTAT --> The exact value of the first dimension of the matrix GSTAT 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, §B].
- For AOV1:  $LDSTAK \geq 22 + N + (8 \cdot NG + N) \cdot P$
- For AOV1S:  $LDSTAK \geq 22 + N + (4 \cdot NG + N) \cdot P$
- N --> The total number of observations (all of the groups combined).
- NG <-- The number of distinct groups, that is, the number of different positive tag values.
- NPRT --> The parameter controlling printed output.
- If NPRT = 0 the printed output is suppressed.
- If NPRT  $\neq$  0 the printed output is provided.
- TAG --> The vector of dimension at least N that contains the tag value for each observation. The tag values may be any REAL number. Groups are formed from observations having the same positive tag value. Observations having a zero or negative tag value will not be included in the analysis.
- Y --> The vector of dimension at least N that contains the observed data. The order of the observations in Y is arbitrary since groups are specified by the values in the corresponding elements of vector TAG.

## E. Computational Methods

### E.1 Algorithms

The code and output for the one-way analysis of variance subroutines are adapted from OMNITAB II [Hogben et al. 1971]. The computations performed are discussed below.

### E.2 Computed Results and Printed Output

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

Each of the five sections of automatic printing is described below under the headings which appear on the printed page as shown in example F-1b. The discussion is taken from the OMNITAB II User's Reference Manual [Hogben et al. 1971], pages 122 to 124.



ANALYSIS OF VARIANCE. The traditional analysis of variance for a one-way classification is printed. This shows the sources of variation, the degrees of freedom, the sums of squares, the mean squares, the F-ratio for testing for differences between group means and the significance level of the F-ratio. The usual assumptions of normality, independence and constant variance of measurement errors are made. A discussion of the statistical treatment of a one-way classification can be found in §10.2 of Brownlee [1965].

If the significance level of the F ratio (F PROB.) for

$$F = (\text{Between Groups Mean Square}) / (\text{Within Groups Mean Square})$$

is less than 0.10 and the number of groups exceeds two, the between groups (means) sum of squares is separated into two components: the first associated with the slope (one degree of freedom) and the second representing deviations about the straight line regression of group averages on group number. This information, which does not appear in a traditional analysis of variance, can be used to examine the effect of time. A discussion of some of the statistical aspects of this procedure are found in §11.12 of Brownlee [1965].

Following the above mentioned analysis of variance, the results for the Kruskal-Wallis non-parametric H-test for testing for differences between group means (averages) are printed. The value of H is printed along with its significance level (F PROB.). The H-test uses the ranks of the measurements and avoids any assumption about the distribution of measurement errors. Details of this test may be found in §7.7 of Brownlee [1965].

ESTIMATES. The following items are printed for each group:

- (1) group number,
- (2) number of observations in the group,
- (3) mean,
- (4) within standard deviation,
- (5) standard deviation of the mean,
- (6) minimum (i.e., smallest) observation,
- (7) maximum (i.e., largest) observation,
- (8) the sum of the ranks of the observations, and
- (9) a 95-percent confidence interval for the mean.

The results are printed with the group numbers (tags) in consecutive, increasing order regardless of the order in which the numbers were entered.

In printing the means and standard deviations of the groups, the characters + and - are put immediately after the high (largest) and low (smallest) values. If two or more values are tied for the largest value, the character + is put immediately after all of the tied values. Ties for the smallest values are handled in an analogous manner using the character -. If the number of observations in a group equals one, ESTIMATE NOT AVAILABLE is printed under WITHINS S.D. and S.D. OF MEAN. Also, \*\*\*\*\* TO \*\*\*\*\* appears under 95PCT CONF INT FOR MEAN.

The total number of observations, mean, minimum observation and maximum observation are also printed for the whole dataset combined. In addition, the within standard deviation, standard deviation of the mean and 95-percent



confidence interval for the mean are printed for three different models: a fixed effects model (Model I), a random effects model (MODEL II) and a model which assumes that all observations are from a single group. The confidence limits are formed by taking the grand mean plus (and minus) the product of the percentage point of Student's t distribution and the standard deviation of the mean. Let k be the number of groups and n be the total number of observations with positive tag. Then, the standard deviation of the mean is the square root of the variance of the mean formed as follows:

<u>Model</u>	<u>Variance</u>	<u>Variance of Mean</u>	<u>Degrees of Freedom</u>
I	$V_I =$ Within groups mean square	$V_I/n$	n-k
II	$V_{II} = \sum_{i=1}^k (Y(i) - \bar{Y})^2 / (k-1)$	$V_{II}/k$	k-1
Ungrouped	$V_u =$ Total mean square	$V_u/n$	n-1

PAIRWISE MULTIPLE COMPARISON OF MEANS. This section only appears if the significance level (value of F PROB.) of the between groups F-ratio is less than 0.10. The Newman-Keuls-Hartley procedure is not performed if the number of measurements with positive tag is less than four plus the number of groups.

The purpose of this comparison is to divide the groups in such a way that the group means within a division are not significantly different at the 0.05 significance level, whereas the group means in different divisions are significantly different at the 0.05 level. Two different procedures are used: the Newman-Keuls-Hartley method and the Scheffe' method. The two methods are similar but not identical and frequently give slightly different results. The Newman-Keuls-Hartley method is described in §10.6 of Snedecor [1956] and §10.8 of Snedecor and Cochran [1967]. The Scheffe' method is discussed in §10.3 of Brownlee [1965]. Groups are separated by a string of five asterisks. If two divisions have no group means in common, the two divisions are separated by two strings of five asterisks.

Both the Newman-Keuls-Hartley method and the Scheffe' method require percentage points of the studentized range: an approximation developed by Mandel is used to compute them. Since the Newman-Keuls-Hartley method is designed for use when the number of observations in each group is the same, the number of observations in each of the two groups is approximated by m, where

$$1/m = (1/2) \cdot (1/m_i + 1/m_j)$$

and  $m_i$  and  $m_j$  are the actual number of measurements in each of the two groups.

TEST FOR HOMOGENEITY OF VARIANCES. The usual analysis of variance for a one-way classification assumes that the variances of all groups are the same.

This section of output provides information for assessing the validity of this assumption. Small values of the significance level P indicate lack of homogeneity of variance. The Cochran's C statistic printed is discussed on page 180 of Dixon and Massey [1957] and in more detail in chapter 15 of Eisenhart et al. [1947]. The Bartlett-Box F-test is a modification of Bartlett's test which uses the F-distribution rather than the chi-squared distribution and is less sensitive to non-normality. It is discussed on pages 179 and 180 of Dixon and Massey [1957]. A table of critical values of (maximum variance)/(minimum variance) for equal sample sizes is given on pages 100 and 101 of Owen [1962].

If either P value is less than or equal to 0.10, the approximate between mean F-test in the presence of heterogeneous variance and its significance level P are also printed. This approximate F-test for testing for differences between means is described on pages 287-289 of Snedecor [1956]. This information does not appear in figure F-1b because both P values (significance levels) exceed 0.10.

MODEL II - COMPONENTS OF VARIANCE. This is the usual analysis of variance estimate for the between component in a random effects model (Model II). For a discussion of this analysis, see §10.6 and §10.7 of Brownlee [1965].

#### F. Example

The example program of figure F-1a uses AOV1 to analyze 16 determinations of the gravitational constant, grouped according to the material used to make the measurements. Figure F-1b shows the output from this example. A discussion of this example can be found on pages 314-316 of Brownlee [1965].

```

MAIN PROGRAM:   PROGRAM EXAMPL
C
C DEMONSTRATE AOV1 USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/R40.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPES ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y AND TAG MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(20), TAG(20)
C DOUBLE PRECISION OSTAK(200)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED DATA
C TAG VALUES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C READ 101, (TAG(I), I=1,N)
C
C PRINT TITLE AND CALL AOV1 FOR ANALYSIS OF VARIANCE
C
C WRITE (6, 102)
C CALL AOV1 (Y, TAG, N, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
100 FORMAT (I5)
101 FORMAT (20F5.1)
102 FORMAT ('RESULTS OF STARPAC ONE-WAY ANALYSIS OF VARIANCE',
* ' SUBROUTINE AOV1')
C ENO
DATA:          16
83.0 81.0 76.0 78.0 79.0 72.0 61.0 61.0 67.0 67.0 64.0 78.0 71.0 75.0 72.0 74.0
1.0  1.0  1.0  1.0  1.0  1.0  2.0  2.0  2.0  2.0  2.0  3.0  3.0  3.0  3.0  3.0

```

Figure F-1a

Example program using AOV1

ANALYSIS OF VARIANCE

\*GROUP NUMBERS HAVE BEEN ASSIGNED ACCORDING TO TAG VALUES GIVEN, WHERE THE SMALLEST TAG GREATER THAN ZERO HAS BEEN ASSIGNED \*  
 \*GROUP NUMBER 1, THE NEXT SMALLEST, GROUP NUMBER 2, ETC. TAGS LESS THAN OR EQUAL TO ZERO HAVE NOT BEEN INCLUDED IN ANALYSIS.\*  
 \*NUMBER OF VALUES EXCLUDED FROM ANALYSIS IS 0

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARES	F RATIO	F PROB.
BETWEEN GROUPS	2	5.651042E+02	2.825521E+02	26.082	.000
SLOPE	1	6.450893E+01	6.450893E+01	1.408	.237
DEVS. ABOUT LINE	1	5.005952E+02	5.005952E+02	46.209	.000
WITHIN GROUPS	13	1.408333E+02	1.083333E+01		
TOTAL	15	7.059375E+02			

KRUSKAL-WALLIS RANK TEST FOR DIFFERENCE BETWEEN GROUP MEANS \* H = 11.391, F PROB = .000 (APPROX.)

ESTIMATES

TAG	NO.	MEAN	WITHIN S.O.	S.O. OF MEAN	MINIMUM	MAXIMUM	SUM OF RANKS	95PCT CONF INT FOR MEAN
1	6	7.81667E+01	3.86868E+00	1.97938E+00	7.20000E+01	8.30000E+01	76.0	7.41067E+01 TO 8.22266E+01
2	5	6.40000E+01	3.00000E+00	1.34164E+00	6.10000E+01	6.70000E+01	15.0	6.02750E+01 TO 6.77250E+01
3	5	7.40000E+01	2.73861E+00	1.22474E+00	7.10000E+01	7.80000E+01	45.0	7.05996E+01 TO 7.74004E+01
TOTAL	16	7.24375E+01			6.10000E+01	8.30000E+01		

FIXED EFFECTS MODEL 3.29140E+00 8.22851E-01  
 RANDOM EFFECTS MODEL 7.29576E+00 4.21221E+00  
 UNGROUPED DATA 6.86021E+00 1.71509E+00

PAIRWISE MULTIPLE COMPARISON OF MEANS. THE MEANS ARE PUT IN INCREASING ORDER IN GROUPS SEPARATED BY \*\*\*\*\*. A MEAN IS ADJUDGED NON-SIGNIFICANTLY DIFFERENT FROM ANY MEAN IN THE SAME GROUP AND SIGNIFICANTLY DIFFERENT AT THE .05 LEVEL FROM ANY MEAN IN ANOTHER GROUP. \*\*\*\*\* INDICATES ADJACENT GROUPS HAVE NO COMMON MEAN.

NEWMAN-KEULS TECHNIQUE, HARTLEY MODIFICATION. (APPROXIMATE IF GROUP NUMBERS ARE UNEQUAL.)  
 6.40000E+01, \*\*\*\*\*  
 7.40000E+01, 7.81667E+01, \*\*\*\*\*

SCHIFFE TECHNIQUE.

6.40000E+01, \*\*\*\*\*  
 7.40000E+01, 7.81667E+01, \*\*\*\*\*

TESTS FOR HOMOGENEITY OF VARIANCES.

COCHRAN C = MAX. VARIANCE/SUM(VARIANCES) = .4756, P = .395 (APPROX.)  
 HARTLEY-BOX F = .269, P = .764  
 MAXIMUM VARIANCE / MINIMUM VARIANCE = 1.9956

MODEL II - COMPONENTS OF VARIANCE.  
 ESTIMATE OF BETWEEN COMPONENT 5.114706E+01

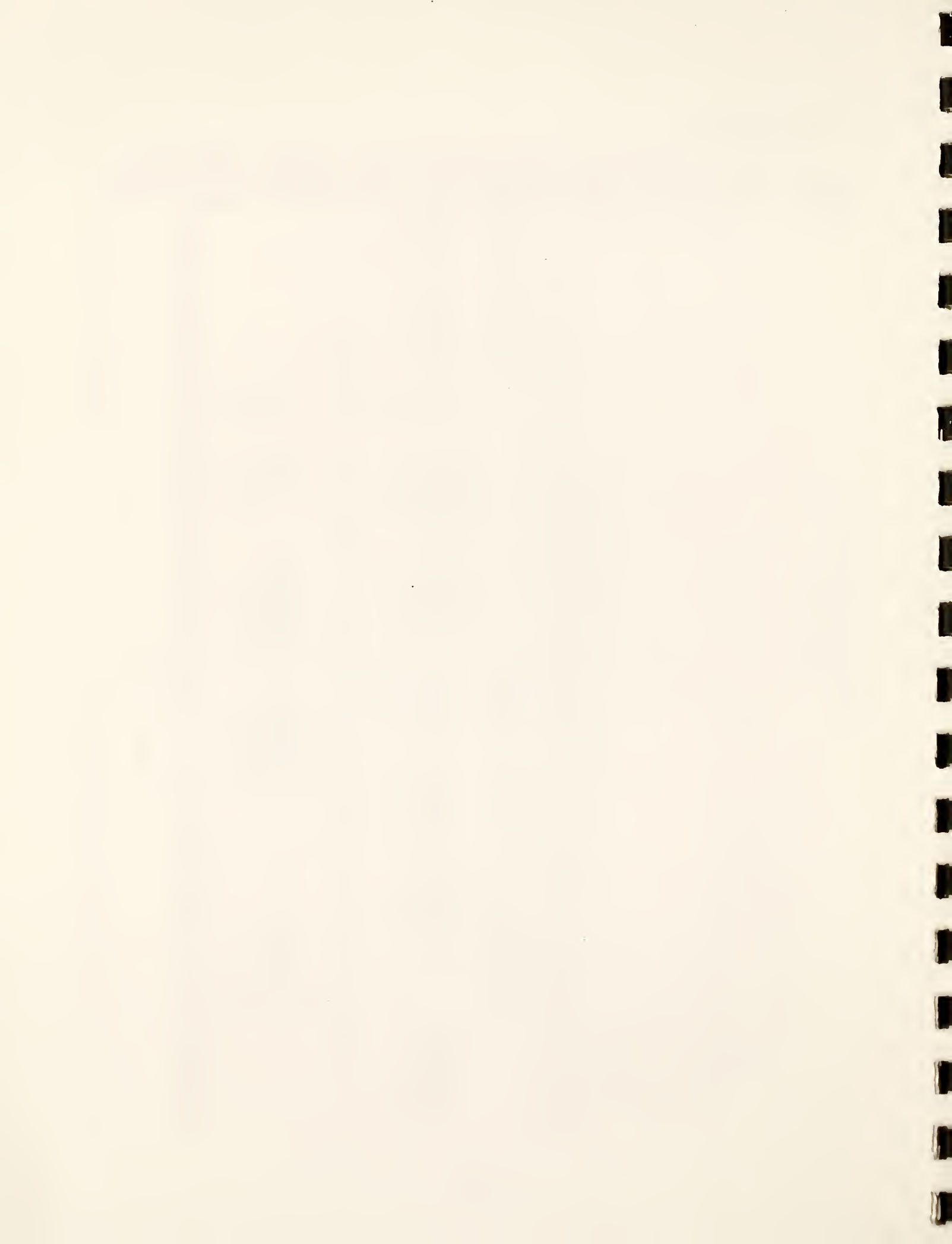
Figure F-1b

Example of AOV1 output

G. Acknowledgments

The code and output for the one-way analysis of variance subroutines is adapted from OMNITAB II [Hogben et al. 1971]. The discussion of the results (§E.2) is taken from the OMNITAB II User's Reference Manual [Hogben et al. 1971].





## CHAPTER 7

### CORRELATION ANALYSIS

#### A. Introduction

STARPAC contains two subroutines for correlation analysis of a multivariate random sample. The analysis provided by these subroutines consists of seven tables, which, when used together, aid the user in using correlation techniques effectively for prediction and model building. The analysis is the same as that provided by the OMNITAB II command CORRELATION [Hogben et al. 1971]. For further information on correlation techniques users should consult Kendall and Stuart [1973], Brownlee [1965] and Anderson [1958].

Users are directed to \$B for a brief description of the subroutines. The declaration and CALL statements are given in \$C and the subroutine arguments are defined in \$D. The algorithms used and output produced by these subroutines are discussed in \$E. Sample programs and their output are shown in \$F.

#### B. Subroutine Descriptions

Subroutine CORR computes and prints 1) the simple correlation matrix and 2) the significance levels of the simple correlation coefficients; 3) the partial correlation coefficients and 4) their significance levels; 5) the Spearman rank correlation coefficients; 6) a test for a quadratic relationship among the variables; and 7) 95-percent and 99-percent confidence intervals for the simple correlation coefficients.

Subroutine CORRS provides the same analysis as CORR but returns the variance-covariance matrix used to compute the correlation coefficients. The user can also optionally suppress the printed output.

#### C. Subroutine Declaration and CALL Statements

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

*CORR: Compute and print a correlation analysis of a multivariate random sample*

```
<real> YM(n,m)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL CORR (YM, N, M, IYM, LDSTAK)
```

===

*CORRS: Compute and optionally print a correlation analysis of a multivariate random sample; return variance-covariance matrix*

```
<real> YM(n,m), VCV(m,m)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
```

```
:
```

```
CALL CORRS (YM, N, M, IYM, LDSTAK, NPRT, VCV, IVCV)
```

#### 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.

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.

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

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

IYM --> The exact value of the first dimension of the matrix YM 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, §B.]

-- continued --

For CORR:

$$\text{LDSTAK} \geq (47 + \max(N, M)) / 2 + (\max(N, M) + N \cdot (M + 3) + M + 7 \cdot M^2) \cdot P$$

For CORRS:

$$\text{LDSTAK} \geq (47 + \text{IO} \cdot \max(N, M)) / 2 + \text{IO} \cdot (\max(N, M) + N \cdot (M + 3) + M + 6 \cdot M^2) \cdot P$$

where IO = 0 if NPRT = 0 and IO = 1 if NPRT ≠ 0

- M      --> The number of variables measured for each observation, that is, the number of columns of data in YM.
- N      --> The number of observations, that is, the number of rows of data in YM.
- NPRT   --> The argument controlling printed output.
- If NPRT = 0 the printed output is suppressed.
- If NPRT ≠ 0 the printed output is provided.
- VCV    <-- The matrix of dimension at least M by M that contains the variance-covariance matrix,

$$\text{VCV}(j, k) = (N-1)^{-1} \sum_{i=1}^N (\text{YM}(i, j) - \bar{Y}_j)(\text{YM}(i, k) - \bar{Y}_k)$$

$$\text{where } \bar{Y}_j = N^{-1} \cdot \sum_{i=1}^N \text{YM}(i, j) \text{ and } \bar{Y}_k = N^{-1} \cdot \sum_{i=1}^N \text{YM}(i, k).$$

- YM      --> The matrix of dimension at least N by M that contains the observed multivariate data. The element in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column is the  $i^{\text{th}}$  observation on the  $j^{\text{th}}$  variable.

## E. Computational Methods

### E.1 Algorithms

Formulas for the computed tables are given in §E.2. The code and output for the correlation analysis subroutines are adapted from OMNITAB II [Hogben et al. 1971].

### E.2 Computed Results and Printed Output

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



STARPAC correlation analysis subroutines compute and print a seven-part correlation analysis. The output for each part is discussed individually below; the text for this discussion is taken from the OMNITAB II User's Reference Manual [Hogben et al. 1971].

The Simple Correlation Coefficient Matrix. The  $(j,k)^{th}$  entry of this matrix is the simple (product moment) correlation coefficient,  $r_{jk}$ , between the data in columns  $j$  and  $k$  defined by

$$r_{jk} = \frac{VCV(j,k)}{(VCV(j,j) \cdot VCV(k,k))^{1/2}} .$$

Note that when more than two variables are under study, the simple correlation coefficient can be seriously distorted by the effect of other variables. The partial correlation coefficient (see below) can be used to identify such distortion.

The Significance Levels of the Simple Correlation Coefficients. The  $(j,k)^{th}$  entry of this table is the significance level,  $S_r(j,k)$  of the corresponding partial correlation coefficient,  $r_{jk}$ ,

$$S_r(j,k) = \text{probability of exceeding } F_0(1, N-2)$$

where  $F_0$  is an F-statistic with 1 and  $N-2$  degrees of freedom,

$$F_0(1, N-2) = \frac{(N-2)r_{jk}^2}{(1-r_{jk}^2)}$$

If the "true" correlation coefficient is equal to zero, then  $S_r(j,k)$  is the probability that in a random sample (of the same size) the absolute value of a sample correlation coefficient will exceed the absolute value of the observed correlation coefficient,  $r_{jk}$ .

The Partial Correlation Coefficients. The partial correlation coefficient,  $p_{jk}$ , between the data in columns  $j$  and  $k$ ,  $j \neq k$ , with the remaining variables fixed, i.e., held constant, is given by

$$p_{jk} = -c_{jk} / (c_{jj} \cdot c_{kk})^{1/2}$$

where  $c_{jk}$  denotes the  $(j,k)^{th}$  element of the inverse of the simple correlation matrix. Because the partial correlation coefficient measures the correlation between two variables after eliminating the effect of the remaining variables, it may avoid the distortion suffered by the simple correlation coefficient when more than two variables are under study. The user should therefore compare the simple correlation coefficients with the partial correlation



coefficients. Any "large" discrepancy indicates that one or more of the remaining variables is having an important effect on the relationship. [See Kendall and Stuart, 1961, §27.5, page 318.]

The Significance Levels of the Partial Correlation Coefficients. The  $(j,k)^{th}$  entry of this table is the significance level,  $S_p(j,k)$  of the corresponding partial correlation coefficient,  $p_{jk}$ ,

$$S_p(j,k) = \text{probability of exceeding } F_0(1, N-M)$$

where  $F_0$  is an F-statistic with 1 and  $N-M$  degrees of freedom,

$$F_0(1, N-M) = \frac{(N-M)p_{jk}^2}{(1-p_{jk}^2)}$$

If the "true" partial correlation coefficient is equal to zero, then  $S_p(j,k)$  is the probability that in a random sample (of the same size) the absolute value of a partial correlation coefficient will exceed the absolute value of the observed partial correlation coefficient,  $p_{jk}$ .

Spearman Rank Correlation Coefficient. The rank correlation coefficient does not require the assumption that the data have a bivariate normal distribution. The Spearman rank correlation coefficient,  $s_{jk}$ , for the data in columns  $j$  and  $k$  is computed from

$$s_{jk} = \frac{A - D_{jk}^2 - T_j - T_k}{\sqrt{(A - 2T_j)(A - 2T_k)}}$$

where

$$D_{jk}^2 = \sum_{i=1}^N [\text{rank}(YM(i,j)) - \text{rank}(YM(i,k))]^2,$$

$$A = (N-1)(N)(N+1)/6,$$

$$T_j = (1/12) \sum_j ((t_j-1)(t_j)(t_j+1)),$$

$$T_k = (1/12) \sum_k ((t_k-1)(t_k)(t_k+1)),$$

$t_j$  = number of ties in a set of tied values in column  $j$  of  $YM$ , and

$t_k$  = number of ties in a set of tied values in column  $k$  of  $YM$ .

The quantities  $T_j$  and  $T_k$  adjust for ties in the ranks. If there are no ties,  $T_j$  and  $T_k$  equal zero and

$$s_{jk} = 1 - (D_{jk}^2 / A).$$

A comparison should be made between the rank correlation coefficients and the corresponding simple and partial correlation coefficients. Again, a "large" discrepancy between two comparable coefficients is an indicator of some abnormality in the data. See Kendall [1948] for further details.

Significance Level of Quadratic Fit over Linear Fit. Underlying the use of a correlation coefficient is the assumption that the two variables are linearly related. The results in this part are useful in assessing the validity of this assumption of linearity. The variables are all assumed to be normally distributed. The numbers printed are the significance levels for a F-test of the hypothesis that the quadratic term in a quadratic model is zero. The F-statistic used is

$$F_0(1, N-3) = \frac{\text{RSS}(\text{linear model}) - \text{RSS}(\text{quadratic model})}{\text{RSS}(\text{quadratic model}) / (N-3)}$$

with 1 and (N-3) degrees of freedom, where RSS is the residual sum of squares function. The values of 1 and (N-3) are printed in the heading. The significance level,  $S_Q$ , is then computed as

$$S_Q = \Pr(F > F_0).$$

Small values of the significance level (less than 0.05, for example) indicate lack of linearity. The test results differ depending upon which variable of a pair is considered the dependent variable and which one is considered the independent (or predictor) variable. Hence, the entire table is printed, rather than just the lower triangle. The diagonal entries are always equal to one and have no particular relevance. Tests of hypotheses in linear regression are discussed in §13.8 of Brownlee [1965].

Confidence Intervals For Simple Correlation Coefficients. Both 95-percent and 99-percent confidence intervals for the simple correlation coefficients are printed in this two-way table. There are two entries in each cell of the table. The values 0.95 and 0.99 are printed along the upper left to lower right diagonal. The 95-percent confidence limits are printed below the diagonal and the 99-percent confidence limits are printed above the diagonal. The number in the lower left of each cell is the lower confidence limit and the number in the upper right is the upper confidence limit.

The confidence intervals are based on a normal approximation. [See Morrison, 1967, chapter 3, page 101.] They are computed as follows:

$$\text{Lower confidence limit: } \tanh[z - u / \sqrt{N-3}]$$

$$\text{Upper confidence limit: } \tanh[z + u / \sqrt{N-3}]$$

where

$$z = \begin{cases} \tanh^{-1}(r_{jk}) \\ 0.50 \log_e[(1+r_{jk})/(1-r_{jk})] \end{cases}$$

and

$$u = \begin{cases} 1.96 \text{ for 95-percent confidence interval} \\ 2.58 \text{ for 99-percent confidence interval.} \end{cases}$$

F. Example

The sample program shown in figure F-1a illustrates the use of CORR. The data are taken from Draper and Smith [1968], page 216. The data are part of a study to determine the effect of relative urbanization, educational level, and relative income (columns 1, 2, and 3, respectively) on product usage (column 4). The output from CORR is shown in figures F-1b and F-1c.

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE CORR USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF YM MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL YM(20,6)
C DOUBLE PRECISION DSTAK(200)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C IYM = 20
C
C READ NUMBER OF OBSERVATIONS AND NUMBER OF VARIABLES
C OBSERVED MULTIVARIATE DATA
C
C READ 100, N, M
C READ 101, ((YM(I,J), J=1,M), I=1,N)
C
C PRINT TITLE AND CALL CORR TO PERFORM CORRELATION ANALYSIS
C
C WRITE(6,102)
C CALL CORR (YM, N, M, IYM, LDSTAK)
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I5)
C 101 FORMAT (4F6.1)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' CORRELATION ANALYSIS SUBROUTINE CORR')
C END
DATA:
      9      4
42.2 11.2 31.9 167.1
48.6 10.6 13.2 174.4
42.6 10.6 28.7 160.8
39.0 10.4 26.1 162.0
34.7  9.3 30.1 140.8
44.5 10.8  8.5 174.6
39.1 10.7 24.3 163.7
40.1 10.0 18.6 174.5
45.9 12.0 20.4 185.7

```

Figure F-1a

Example program using CORR

RESULTS OF STARPAC CORRELATION ANALYSIS SUBROUTINE CORR

CORRELATION ANALYSIS FOR 4 VARIABLES WITH 9 OBSERVATIONS

CORRELATION MATRIX  
 - STANDARD DEVIATIONS ARE ON THE DIAGONAL  
 - CORRELATION COEFFICIENTS ARE BELOW THE DIAGONAL

COLUMN	1	2	3	4
1	4.1764522			
2	.69374212	.74628711		
3	-.61596989	-.17249295	7.9279218	
4	.80175221	.76795029	-.62074595	12.649157

SIGNIFICANCE LEVELS OF SIMPLE CORRELATION COEFFICIENTS (ASSUMING NORMALITY)

COLUMN	1	2	3	4
1	0.			
2	.42269734E-01	0.		
3	.77358710E-01	.65719690	0.	
4	.93530479E-02	.15660429E-01	.69711268E-01	0.

PARTIAL CORRELATION COEFFICIENTS WITH 2 REMAINING VARIABLES FIXED

COLUMN	1	2	3	4
1	1.0000000			
2	.43170939	1.0000000		
3	-.44663591	.69717000	1.0000000	
4	.10539042	.72682008	-.64778927	1.0000000

SIGNIFICANCE LEVELS OF PARTIAL CORRELATION COEFFICIENTS (ASSUMING NORMALITY)

COLUMN	1	2	3	4
1	0.			
2	.33344895	0.		
3	.30301787	.81676691E-01	0.	
4	.82207563	.64249391E-01	.11565980	0.

SPEARMAN RANK CORRELATION COEFFICIENTS (ADJUSTED FOR TIES)

COLUMN	1	2	3	4
1	1.0000000			
2	.61088401	1.0000000		
3	-.56666667	-.12552411	1.0000000	
4	.68333333	.60251573	-.71666667	1.0000000

Figure F-1b

Example of CORR output



SIGNIFICANCE LEVEL OF QUADRATIC FIT OVER LINEAR FIT BASED ON F RATIO WITH 1 AND 6 DEGREES OF FREEDOM  
 (FOR EXAMPLE, .1703 IS THE SIGNIFICANCE LEVEL OF THE QUADRATIC TERM WHEN COLUMN 2 IS FITTED TO COLUMN 1)

COLUMN	1	2	3	4
1	1.0000000	.40442827	.94936019	.85222263
2	.17034675	1.0000000	.80987340	.93773837
3	.71654009	.56763094	1.0000000	.84988526
4	.15652386	.59973987	.36810270	1.0000000

CONFIDENCE INTERVALS FOR SIMPLE CORRELATION COEFFICIENTS (USING FISHER TRANSFORMATION)  
 95 PER CENT LIMITS BELOW DIAGONAL, 99 PER CENT LIMITS ABOVE DIAGONAL

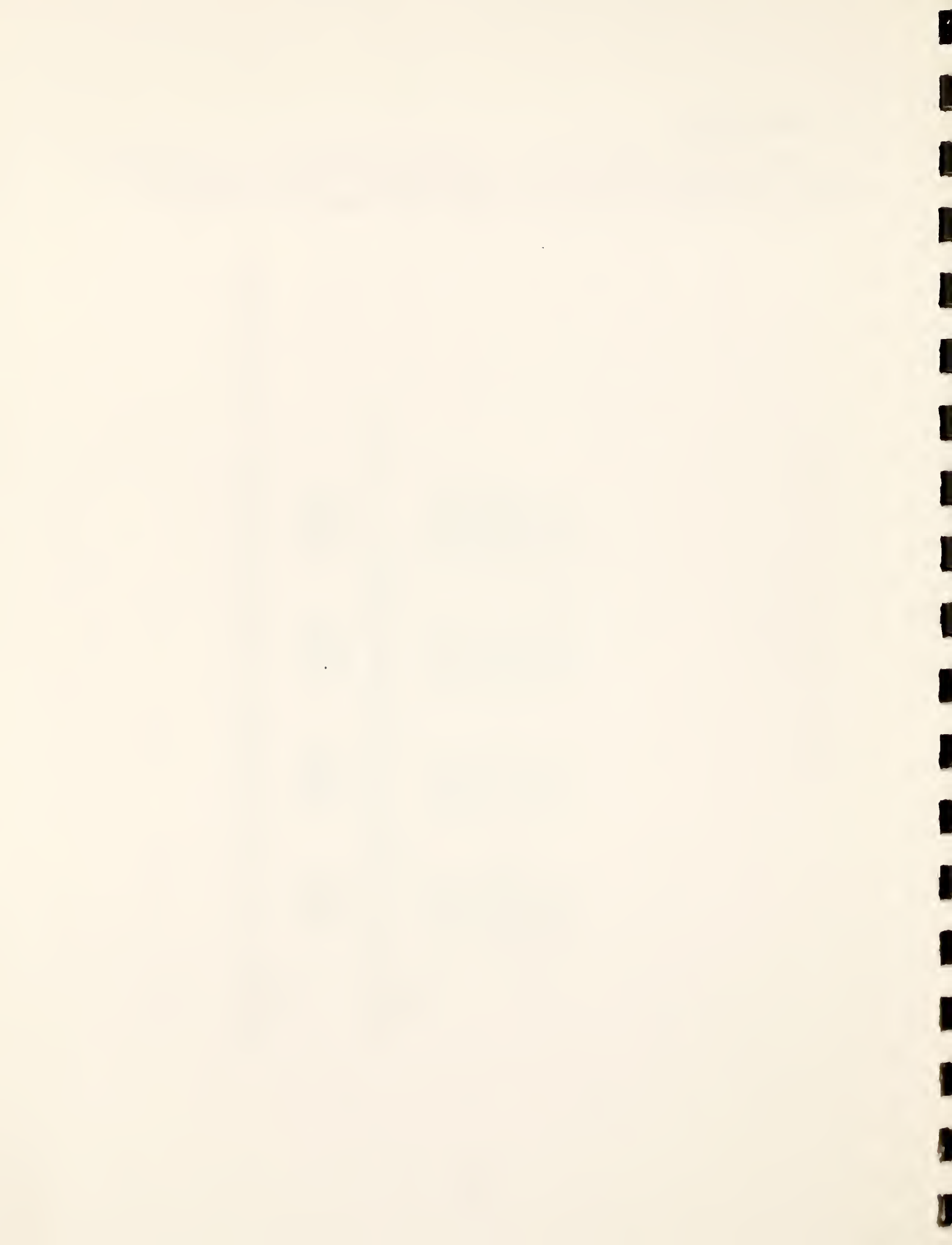
COLUMN	1	2	3	4
1	99.000000	.95517073	.32124731	.97349304
2	95.000000	-.21219610	-.94363630	.21887496E-01
3	.92694792	99.000000	.70508573	.96846093
4	.35940598E-01	95.000000	-.84136050	-.36249864E-01
5	.01486051E-01	.55523436	99.000000	.30247207
6	-.90845980	-.75062574	95.000000	-.94388730
7	.95654889	.94888422	.60737579E-01	99.000000
8	.29437227	.21190035	-.91203488	95.000000

Figure F-1c

Example of CORR output (continued)

G. Acknowledgments

The code and output for the correlation subroutines is adapted from OMNITAB II [Hogben et al. 1971]. The discussion of the results (§E.2) is taken from the OMNITAB II Users' Reference Manual [Hogben et al. 1971].



## CHAPTER 8

### LINEAR LEAST SQUARES

#### A. Introduction

STARPAC contains eight subroutines for linear least squares analysis. For four of these, the user specifies the model by supplying the design matrix (the matrix whose columns are the independent variables plus a column of ones if a constant term is being estimated). The other four perform the same analysis for the special case of a polynomial model, where the need for the user to explicitly create the design matrix is eliminated.

Each of the subroutines described in this chapter assumes that the observations of the dependent variable,  $y_i$ , which are measured with error, are modeled by

$$y_i = \left[ \sum_{j=1}^{\text{NPAR}} \beta(j) \cdot x_i(j) \right] + \epsilon_i \text{ for } i = 1, \dots, N,$$

where

$N$  is the number of observations;

$\text{NPAR}$  is the number of parameters;

$x_i(j)$  is the  $j^{\text{th}}$  element of the  $i^{\text{th}}$  row of the design matrix (for the user-specified model,  $x_i(j) = \text{XM}(i,j)$  for  $i = 1, \dots, N$  and  $j = 1, \dots, \text{NPAR}$ , while for the polynomial model,  $x_i(j) = X(i)^{j-1}$  for  $i = 1, \dots, N$  and  $j = 1, \dots, \text{NPAR}$ );

$\beta$  is the vector of the  $\text{NPAR}$  unknown parameters (coefficients); and

$\epsilon_i$  is the random error in the  $i^{\text{th}}$  observation.

The least squares solution,  $\hat{\beta}$ , is that which minimizes (with respect to  $\beta$ ) the residual sum of squares function

$$\begin{aligned} \text{RSS}(\beta) &= \sum_{i=1}^N [\epsilon_i^2 \cdot \text{wt}_i] \\ &= \sum_{i=1}^N \left[ \left( y_i - \left[ \sum_{j=1}^{\text{NPAR}} \beta(j) \cdot x_i(j) \right] \right)^2 \cdot \text{wt}_i \right] \end{aligned}$$

where carat (^) denotes the estimated quantity, and

$\text{wt}_i$  is the weight assigned to the  $i^{\text{th}}$  observation ( $\text{wt}_i = 1.0$  in the "unweighted" case). Appendix B discusses several common applications for weighted least squares.

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

## B. Subroutine Descriptions

The linear least squares estimation subroutines permit both weighted and unweighted analysis. The user has two levels of control over the computations and printed output.

- In level one, a four-part printed report is automatically provided and the residuals are returned to the user via the subroutine argument list.
- Level two also returns the residuals. It allows the user to specify the amount of printed output, and, in addition, returns the following estimated values via the argument list:
  - the estimated parameters;
  - 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.

The simplest of the linear least squares subroutines are LLS for the user-specified model and LLSP for the polynomial model. They perform unweighted analyses, provide a four-part printed report and return the residuals via the argument list (level one control). The other six subroutines provide greater flexibility by adding the weighting and/or level two control features. These features are each indicated by a different suffix letter on the subroutine name (e.g., LLSS and LLSPWS).

- Suffix W indicates user-supplied weights.
- Suffix S indicates level two control of the computations and output.

## C. Subroutine Declaration and CALL Statements

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



*LLS: Compute and print a four-part unweighted linear least squares analysis with user-specified model (design matrix); return residuals*

```
<real> Y(n), XM(n,npar), RES(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLS (Y, XM, N, IXM, NPAR, RES, LDSTAK)
```

===

*LLSS: Compute and optionally print a four-part unweighted linear least squares analysis with user-specified model (design matrix); return residuals, parameter estimates, residual standard deviation, predicted values, standard deviations of the predicted values, standardized residuals, and variance-covariance matrix of parameters*

```
<real> Y(n), XM(n,npar), RES(n)
<real> PAR(npar), PV(n), SDPV(n), SDRES(n), VCV(npar,npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSS (Y, XM, N, IXM, NPAR, RES, LDSTAK,
1 NPRT, PAR, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

*LLSW: Compute and print a four-part weighted linear least squares analysis with user-specified model (design matrix); return residuals*

```
<real> Y(n), XM(n,npar), RES(n)
<real> WT(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSW (Y, WT, XM, N, IXM, NPAR, RES, LDSTAK)
```

===

*LLSWS: Compute and optionally print a four-part weighted linear least squares analysis with user-specified model (design matrix); return residuals, parameter estimates, residual standard deviation, predicted values, standard deviations of the predicted values, standardized residuals, and variance-covariance matrix of parameters*

```
<real> Y(n), XM(n,npar), RES(n)
<real> WT(n), PAR(npar), PV(n), SDPV(n), SDRES(n), VCV(npar,npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSWS (Y, WT, XM, N, IXM, NPAR, RES, LDSTAK,
1 NPRT, PAR, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

*LLSP: Compute and print a four-part unweighted linear least squares analysis with polynomial model (design matrix); return residuals*

```
<real> Y(n), X(n), RES(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSP (Y, X, N, NDEG, RES, LDSTAK)
```

===

*LLSPS: Compute and optionally print a four-part unweighted linear least squares analysis with polynomial model (design matrix); return residuals, parameter estimates, residual standard deviation, predicted values, standard deviations of the predicted values, standardized residuals, and variance-covariance matrix of parameters*

```
<real> Y(n), X(n), RES(n)
<real> PAR(npar), PV(n), SDPV(n), SDRES(n), VCV(npar,npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSPS (Y, X, N, NDEG, RES, LDSTAK,
1 NPRT, LPAR, PAR, NPAR, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

*LLSPW: Compute and print a four-part weighted linear least squares analysis with polynomial model (design matrix); return residuals*

```
<real> Y(n), X(n), RES(n)
<real> WT(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSPW (Y, WT, X, N, NDEG, RES, LDSTAK)
```

===

*LLSPWS: Compute and optionally print a four-part weighted linear least squares analysis with polynomial model (design matrix); return residuals, parameter estimates, residual standard deviation, predicted values, standard deviations of the predicted values, standardized residuals, and variance-covariance matrix of parameters*

```
<real> Y(n), X(n), RES(n)
<real> WT(n), PAR(npar), PV(n), SDPV(n), SDRES(n), VCV(npar,npar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL LLSPWS (Y, WT, X, N, NDEG, RES, LDSTAK,
1 NPRT, LPAR, PAR, NPAR, RSD, PV, SDPV, SDRES, VCV, IVCV)
```

===

#### 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.

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.

IERR    ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.]  
 Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected and that the least squares solution was found.

IERR = 1 indicates that improper input was detected.

IERR = 2 indicates that the model is computationally singular, which may mean the model has too many parameters. The user should examine the model and data to determine and remove the cause of the singularity.

IERR = 3 indicates that at least one of the standardized residuals could not be computed because its standard deviation was zero. The validity of the variance-covariance matrix is questionable.

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

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 value given below, where if the single precision version of STARPAC is being used  $P = 0.5$ , otherwise  $P = 1.0$ . [See chapter 1, §B.]

$$LDSTAK \geq 28 + [6 \cdot N + 1 + 5 \cdot NPAR + NPAR \cdot N + 2 \cdot NPAR^2] \cdot P$$

LPAR    --> The actual length of the vector PAR as specified in the calling program.

N        --> The number of observations.

NDEG    --> The degree of the polynomial model. The number of estimated parameters is  $NPAR = NDEG + 1$ .

NPAR    --- The number of parameters to be estimated. NPAR is input to the subroutines with a user-specified model; for the subroutines with a polynomial model,  $NPAR = NDEG + 1$  is returned.



NPRT --> The argument controlling printed output. NPRT is a four-digit integer, where the value of the I<sup>th</sup> digit (counting from left to right) controls the I<sup>th</sup> section of the output.

If the I<sup>th</sup> digit = 0, the output from the I<sup>th</sup> section is suppressed;  
= 1, the brief form of the I<sup>th</sup> section is given;  
> 2, the full form of the I<sup>th</sup> section is given.

The default value for NPRT is 1112. If the user-supplied value of NPRT is less than zero or NPRT is not an argument in the subroutine CALL statement the default value will be used.

A full discussion of the printed output is given in §E.2 and is summarized as follows.

Section 1 provides information for each observation based on the solution. Brief output includes information for the first 40 observations, while full output provides the information for all of the data.

Section 2 is a set of four residual plots. Brief output and full output are the same for this section.

Section 3 is an analysis of variance. Brief output and full output are the same for this section.

Section 4 is the final summary of the estimated parameters. Brief output does not include printing the variance-covariance matrix while full output does.

PAR <-- The vector of dimension at least NPAR that contains the estimated parameter values.

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

$$PV(i) = \sum_{j=1}^{NPAR} \hat{\beta}(j) \cdot x_i(j) = \hat{y}_i \text{ for } i = 1, \dots, N.$$

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

$$\begin{aligned} RES(i) &= y_i - \sum_{j=1}^{NPAR} \hat{\beta}(j) x_i(j) \\ &= y_i - \hat{y}_i = \hat{\epsilon}_i \text{ for } i = 1, \dots, N. \end{aligned}$$



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

$$RSD = \left( \text{RSS}(\hat{\beta}) / (N_{\text{nzW}} - \text{NPAR}) \right)^{1/2}$$
 where  $N_{\text{nzW}}$  is the number of observations with nonzero weights.
- SDPV <-- The vector of dimension at least N that contains the standard deviation of each predicted value at the solution,  

$$\text{SDPV}(i) = \text{the } i^{\text{th}} \text{ diagonal element of } [D \cdot \text{VCV} \cdot D^T]^{1/2}$$
 for  $i = 1, \dots, N$ , where D is the design matrix,  $D(i,j) = x_i(j)$  with  $x_i(j)$  defined in §A, and  $D^T$  is the transpose of D.
- SDRES <-- The vector of dimension at least N that contains the standardized residuals at the solution, i.e., the  $i^{\text{th}}$  residual divided by its estimated standard deviation,  

$$\text{SDRES}(i) = \text{RES}(i) / \left[ (\text{RSD}^2 / \text{wt}_i) - \text{SDPV}(i)^2 \right]^{1/2} \text{ for } i = 1, \dots, N.$$
- VCV <-- The matrix of dimension at least NPAR by NPAR that contains the variance-covariance matrix of the estimated parameters at the solution,  

$$\text{VCV} = \text{RSD}^2 \cdot (D^T \cdot W \cdot D)^{-1}$$
 where W is the N by N diagonal matrix of weights,  

$$W = \text{diag}\{\text{wt}_i, i=1, \dots, N\},$$
 and D is the design matrix,  $D(i,j) = x_i(j)$  with  $x_i(j)$  defined in §A, and  $D^T$  is the transpose of D.
- 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 of a zero-weighted observation are still computed. [See Appendix B.]
- X --> The vector of dimension at least N that contains the independent variable used to construct the design matrix for the polynomial model.
- XM --> The matrix of dimension at least N by NPAR that contains the design matrix, i.e., the matrix whose columns are the independent variables plus a column of ones if a constant term is being estimated.
- Y --> The vector of dimension at least N that contains the dependent variable.

## E. Computational Methods

### E.1. The Linear Least Squares Algorithm

The linear least squares estimation subroutines use a modified Gram-Schmidt algorithm [Davis, 1962; Walsh, 1962]. The printed output for the linear least squares subroutines has been modeled on the linear least squares output used by OMNITAB II [Hogben et al. 1971].

### E.2 Computed Results and Printed Output

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

The output from the linear least squares estimation subroutines consists of four sections, several of which include tables summarizing the results. In the following descriptions, the actual table headings are given by the underlined, uppercase phrases. Results which correspond to input or returned arguments are identified by the argument name in uppercase (not underlined).

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

- ROW: the row number of the observation.
- PREDICTOR VALUES: the values for up to the first three columns of the independent variable (design matrix). For subroutines with a user-supplied model, this is up to the first three columns of the matrix XM (excluding the first column if it is all ones, indicating a constant term); for the polynomial model subroutines, this is the variable X.
- DEPENDENT VARIABLE: the values of the dependent variable, Y.
- PREDICTED VALUE: the predicted values, PV, from the fit.
- STD DEV OF PRED VALUE: the standard deviations of the predicted values, SDPV.
- RESIDUAL: the error estimate, RES.
- STD RES: the standardized residual, SDRES.
- WEIGHT: the user-supplied weights, WT, printed only when weighted analysis is performed.

Section 2 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 3 provides an analysis of variance. *The results of this analysis depend upon the order of the columns of the design matrix unless the columns are orthogonal.* The analysis includes the following information.

- PAR INDEX: the index,  $j$ , of the parameter being examined,  $\beta(j)$ .
- SUM OF SQUARES RED DUE TO PAR:  $SS_j$ , the reduction in the sum of squares due to fitting  $\beta(j)$  after having fit parameters  $\beta(1), \beta(2), \dots, \beta(j-1)$ .  $SS_j$  depends on the order of the parameters unless the design matrix has orthogonal columns. This is a decomposition of the total sum of squares, TSS, into  $\text{NPAR} + 1$  parts

$$\text{TSS} = \left( \sum_{j=1}^{\text{NPAR}} SS_j \right) + \text{RSS}(\hat{\beta}).$$

The residual sum of squares and total sum of squares is also listed in this column.

- CUM MS RED: the cumulative mean square reduction,

$$\text{MSRED}_j = \sum_{k=1}^j SS_k / j \text{ for } j = 1, \dots, \text{NPAR}.$$

- DF(MSRED): the degrees of freedom associated with the cumulative mean square reduction for each parameter,  $\text{DF}(\text{MSRED}_j) = j$  for  $j = 1, \dots, \text{NPAR}$ . The degrees of freedom for the residuals,  $N_{\text{nzW}} - \text{NPAR}$ , and the total degrees of freedom,  $N_{\text{nzW}}$ , where  $N_{\text{nzW}}$  is the number of observations with nonzero weights, are also listed in this column.

- CUM RES MS: the cumulative residual mean square,

$$\text{RMS}_j = \left( \text{RSS}(\hat{\beta}) + \sum_{k=j+1}^{\text{NPAR}} SS_k \right) / (N_{\text{nzW}} - j) \text{ for } j = 1, \dots, \text{NPAR}.$$

- DF(RMS): the degrees of freedom associated with the cumulative residual mean square for each parameter,  $\text{DF}(\text{RMS}_j) = N_{\text{nzW}} - j$  for  $j = 1, \dots, \text{NPAR}$ .

- PAR=0, F and PROB(F): the F-ratio and its significance level under the null hypotheses that  $\beta(j)$  is zero after allowance has been made for parameters  $\beta(1), \beta(2), \dots, \beta(j-1)$ . This F-ratio is

$$F_j = SS_j / \text{RMS}_{\text{NPAR}} \text{ for } j = 1, \dots, \text{NPAR},$$

with 1 and  $N_{\text{nzW}} - \text{NPAR}$  degrees of freedom. The significance level listed is the probability of exceeding the calculated F-ratio under the null hypothesis that the corresponding parameter,  $\beta(j)$ , is zero.

- PARS=0, F and PROB(F): the F-ratio and its significance level under the null hypothesis that parameters  $\beta(j), \beta(j+1), \dots, \beta(\text{NPAR})$ , are zero after allowance has been made for parameters  $\beta(1), \beta(2), \dots, \beta(j-1)$ . This F-ratio is

$$F_j = \left( \sum_{k=j}^{\text{NPAR}} SS_k / (\text{NPAR} - j + 1) \right) / \text{RMS}_{\text{NPAR}} \text{ for } j = 1, \dots, \text{NPAR},$$

with  $\text{NPAR} - j + 1$  and  $N_{\text{nzW}} - \text{NPAR}$  degrees of freedom. The significance level listed is the probability of exceeding the calculated F-ratio under the null hypothesis that all of the parameters  $\beta(j), \beta(j+1), \dots, \beta(\text{NPAR})$  are zero.

The numerator of this ratio is the extra sum of squares accounted for by inclusion of the terms  $\beta(j) \cdot x(j) + \beta(j+1) \cdot x(j+1) + \dots + \beta(\text{NPAR}) \cdot x(\text{NPAR})$  in the model, divided by its degrees of freedom; the denominator is the residual mean square of the full model. This ratio is a means of comparing the extra sum of squares to its expected value as estimated by the residual mean square. When the terms of the model have a logical order of entry, this series of F-tests can be used to judge how many terms should be included in the model. [See Draper and Smith, 1981, pages 97 and 98.]

Section 4 summarizes the following information about the final parameter estimates and their variances.

- The variance-covariance matrix, VCV, of the estimated parameters, and the corresponding correlation matrix,

$$r_{jk} = \text{VCV}(j,k) / (\text{VCV}(j,j) \cdot \text{VCV}(k,k))^{1/2} \text{ for } j = 1, \dots, \text{NPAR} \\ \text{and } k = 1, \dots, \text{NPAR}.$$

- ESTIMATES FROM FIT:

- ESTIMATED PARAMETER: the final estimate for each parameter,  $\text{PAR}(j)$  for  $j = 1, \dots, \text{NPAR}$ .
- SD OF PAR: the standard deviation of the estimated parameter,  $(\text{VCV}(j,j))^{1/2}$  for  $j = 1, \dots, \text{NPAR}$ .
- T(PAR=0): the Student's t statistic under the null hypothesis that  $\text{PAR}(j)$  is actually zero,



$$T(\text{PAR}=0)_j = \text{PAR}(j) / (\text{VCV}(j,j))^{1/2} \text{ for } j = 1, \dots, \text{NPAR}.$$

- PROB(T): the two-sided significance level of  $T(\text{PAR}=0)_j$ . This is the probability of exceeding the given t value under the null hypothesis that the parameter  $\text{PAR}(j)$  is actually zero.
- ACC DIG: an estimate of the number of reliable digits in the parameter estimates, i.e., an indication of the computational accuracy of the solution. A computationally accurate solution will produce values between  $\text{DIGITS}-2$  and  $\text{DIGITS}$ , where  $\text{DIGITS}$  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. Values less than  $\text{DIGITS}-4$  may indicate some computational difficulty such as poor scaling or near singularity.
- ESTIMATES FROM FIT OMITTING LAST PREDICTOR VALUE: ESTIMATED PARAMETER, SD OF PAR,  $T(\text{PAR}=0)$  and  $\text{PROB}(T)$  values for a fit omitting the last column of the design matrix and thus omitting the last parameter from the model.
- The residual standard deviation, RSD.
- The residual degrees of freedom,  $N_{\text{nzW}} - \text{NPAR}$ , where  $N_{\text{nzW}}$  is the number of observations with nonzero weights.
- The squared multiple correlation coefficient,

$$R^2 = 1.0 - \frac{\text{RSS}(\hat{\beta})}{\sum_{i=1}^N \text{wt}_i (Y(i) - \bar{Y}_w)^2} \quad \text{where} \quad \bar{Y}_w = \frac{\sum_{i=1}^N \text{wt}_i \cdot Y(i)}{\sum_{i=1}^N \text{wt}_i}.$$

$R^2$  is a measure of how well the fitted equation accounts for the total variation of the dependent variable,  $Y$ . It is only computed when the first parameter of the model is a constant, i.e., when the elements of the first column of the design matrix are all equal.

## F. Examples

User-Specified Model (Design Matrix). In the example program of figure F-1a, LLS is used to compute the least squares solution for the example given on pages 61-65 of Daniel and Wood [1971]. The results for this problem are also discussed in Draper and Smith [1981], pages 372 and 373. Figures F-1b through F-1e show the four pages of output from LLS.

Polynomial Model (Design Matrix). In the example program of figure F-2a, LLSP is used to compute the least squares solution for the example given on page 311 of Miller and Freund [1977]. Figures F-2b through F-2e show the four pages of output from LLSP.



Linear Least Squares  
With User-Specified Model (Design Matrix)

```

MAIN PROGRAM:      PROGRAM EXAMPL
                   C
                   C DEMONSTRATE LLS USING SINGLE PRECISION VERSION OF STARPAC
                   C RUN ON CYBER 180/840.
                   C
                   C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
                   C [SEE CHAPTER 1, SECTION D.4]
                   C
                   C N.B. DECLARATION OF Y, XM AND RES MUST BE CHANGED TO DOUBLE PRECISION
                   C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
                   C
                   C REAL Y(30), XM(30,9), RES(30)
                   C DOUBLE PRECISION DSTAK(500)
                   C
                   C COMMON /CSTAK/ DSTAK
                   C
                   C SPECIFY NECESSARY DIMENSIONS
                   C
                   C LDSTAK = 500
                   C IXM = 30
                   C
                   C READ NUMBER OF OBSERVATIONS AND NUMBER OF UNKNOWN PARAMETERS
                   C INDEPENDENT VARIABLES
                   C DEPENDENT VARIABLES
                   C
                   C READ 100, N, NPAR
                   C READ 101, ((XM(I,J), I=1,N), J=1,NPAR)
                   C READ 101, (Y(I), I=1,N)
                   C
                   C PRINT TITLE AND CALL LLS TO PERFORM LINEAR LEAST SQUARES ANALYSIS
                   C
                   C WRITE (6, 102)
                   C CALL LLS (Y, XM, N, IXM, NPAR, RES, LDSTAK)
                   C
                   C STOP
                   C
                   C FORMAT STATEMENTS
                   C
                   C 100 FORMAT (2I5)
                   C 101 FORMAT (21F3.0)
                   C 102 FORMAT ('!RESULTS FROM STARPAC',
                   C * ' LINEAR LEAST SQUARES SUBROUTINE LLS')
                   C END
DATA:              21      4
                   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
                   60 80 75 62 62 62 62 62 58 58 58 58 50 50 50 50 50 56 70
                   27 27 25 24 22 23 24 24 23 18 18 17 18 19 18 18 19 19 20 20 20
                   89 88 90 87 87 87 93 93 87 80 89 88 82 93 89 86 72 79 80 82 91
                   42 37 37 28 18 18 19 20 15 14 14 13 11 12  8  7  8  8  9 15 15

```

Figure F-1a

Example program using LLS

RESULTS FROM STAPPAC LINEAR LEAST SQUARES SUBROUTINE LLS  
 \* LINEAR LEAST SQUARES ESTIMATION WITH USER-SPECIFIED MODEL \*

RESULTS FROM LEAST SQUARES FIT

ROW	PREDICTOR VALUES		DEPENDENT VARIABLE	PREDICTED VALUE	STD DEV OF PRED VALUE	RESIDUAL	STD RES
1	60.000000	27.000000	42.000000	38.763363	1.7810630	3.2346372	1.19
2	60.000000	27.000000	88.000000	38.917485	1.8285236	-1.9174853	-.72
3	75.000000	25.000000	37.000000	32.444467	1.3553032	4.5553030	1.55
4	62.000000	24.000000	28.000000	22.302226	1.1626690	5.6977742	1.88
5	62.000000	22.000000	87.000000	19.711654	.74116599	-1.7116536	-.54
6	62.000000	23.000000	18.000000	21.006940	.90284068	-3.0069397	-.97
7	62.000000	24.000000	19.000000	21.389491	1.5186314	-2.3894907	-.83
8	62.000000	24.000000	20.000000	21.389491	1.5186314	-1.3894907	-.48
9	58.000000	23.000000	15.000000	18.144379	1.2143513	-3.1443789	-1.05
10	58.000000	18.000000	14.000000	12.732806	1.4506369	1.2671941	.44
11	58.000000	18.000000	14.000000	11.363703	1.2770503	2.6362968	.88
12	58.000000	17.000000	13.000000	10.220540	1.5114771	2.7794604	.97
13	58.000000	18.000000	11.000000	12.420561	1.2872987	-1.4285609	-.48
14	58.000000	19.000000	12.000000	12.050499	1.4714398	-.50499291E-01	-.02
15	50.000000	18.000000	8.000000	5.638916	1.4154780	2.3614184	.81
16	50.000000	18.000000	7.000000	6.094942	1.1742308	.90505080	.30
17	50.000000	19.000000	8.000000	9.5199506	2.0821373	-1.5199506	-.61
18	50.000000	19.000000	79.000000	8.4550930	1.2907464	-4.5509295	-.15
19	50.000000	20.000000	9.000000	9.5982566	1.3549990	-.5982566	-.20
20	56.000000	20.000000	82.000000	13.587853	.91842680	1.4121473	.45
21	70.000000	20.000000	91.000000	22.237713	1.7300647	-7.2377129	-2.64

Figure F-1b

Example of LLS output

LINEAR LEAST SQUARES ESTIMATION WITH USER-SPECIFIED MODEL, CONTINUED

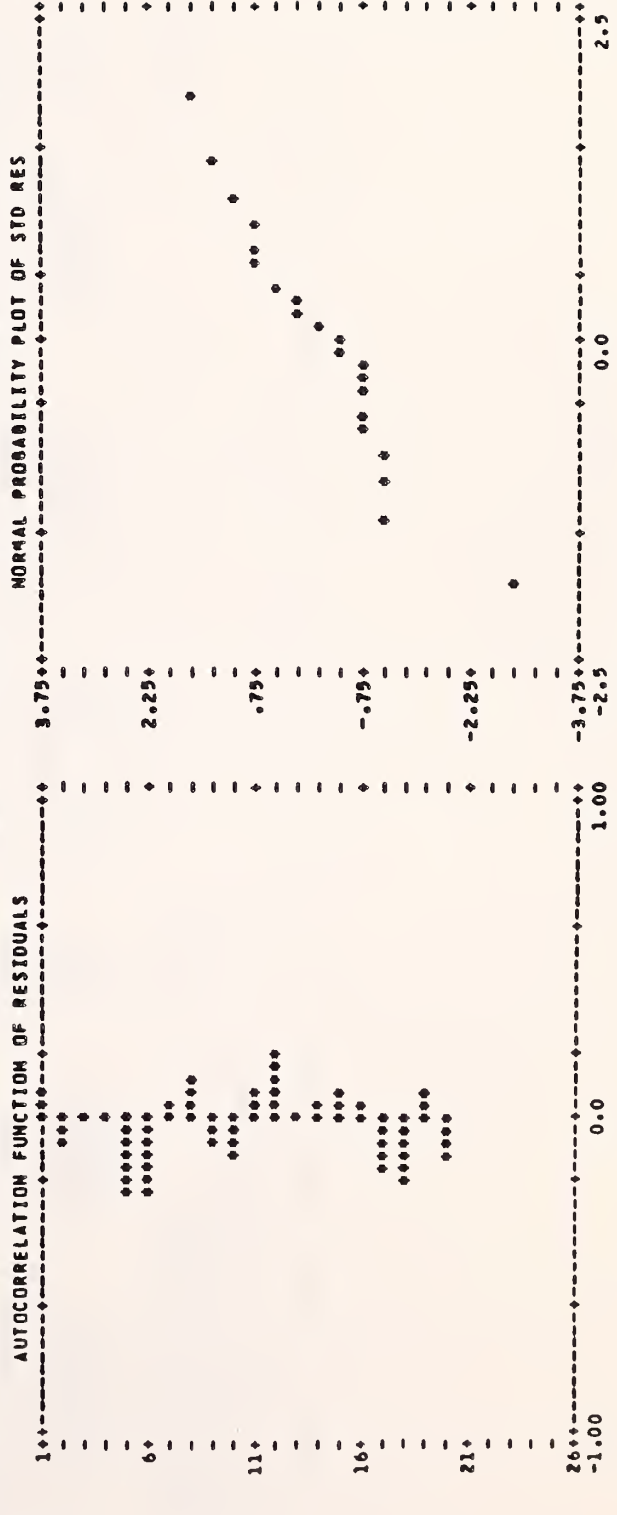
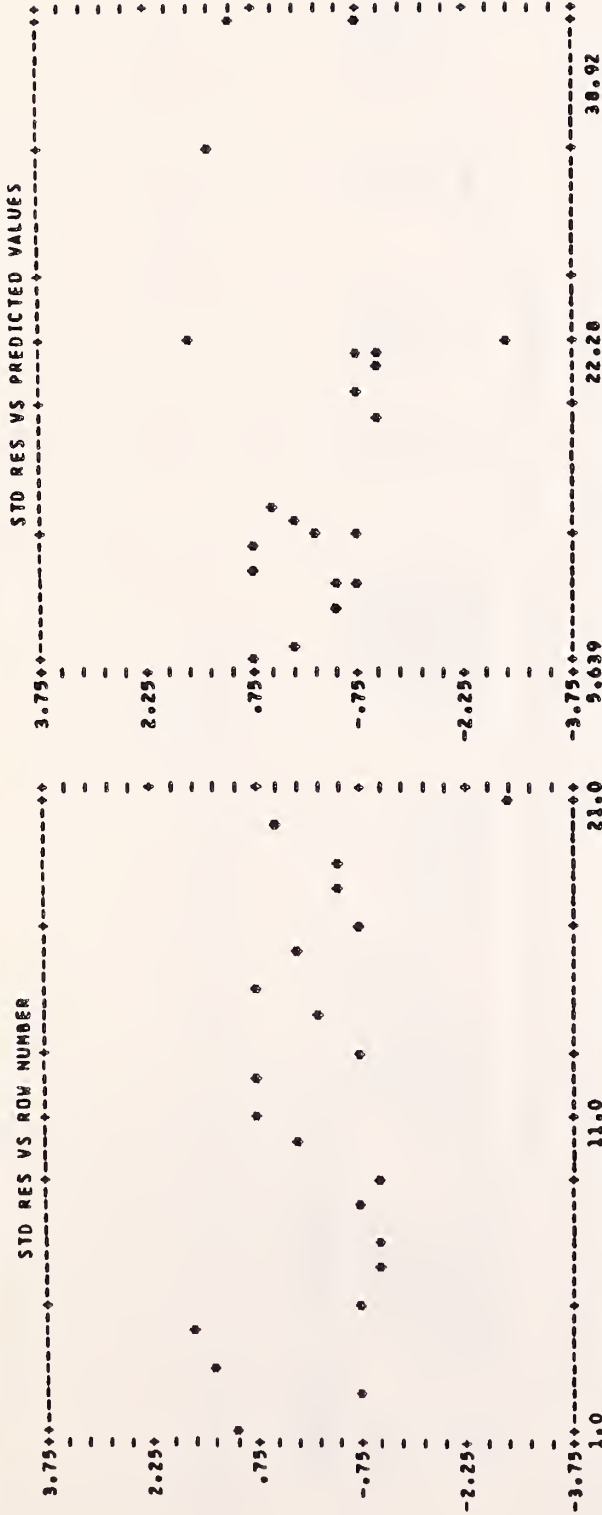


Figure F-1c  
Example of LLS output (continued)

LINEAR LEAST SQUARES ESTIMATION WITH USER-SPECIFIED MODEL, CONTINUED

ANALYSIS OF VARIANCE  
 -DEPENDENT ON ORDER VARIABLES ARE ENTERED, UNLESS VECTORS ARE ORTHOGONAL-

PAR INDEX	SUM OF SQUARES RED DUE TO PAR	CUM MS RED	DF(MSRED)	CUM RES MS	DF(RMS)	F	PROB(F)	F	PROB(F)
1	6448.76190	6448.76190	1	108.461905	20	613.035	.000	196.105	.003
2	1750.12199	4099.44195	2	16.7955845	19	166.371	.000	59.9022	.000
3	130.32072	2776.40136	3	10.4886297	18	12.3806	.003	6.66797	.007
4	9.96597226	2084.79251	4	10.5194099	17	.947332	.344	.947332	.344
RESIDUAL	178.0900		17						
TOTAL	8518.000		21						

Figure F-1d

Example of LLS output (continued)

LINEAR LEAST SQUARES ESTIMATION WITH USER-SPECIFIED MODEL, CONTINUED

VARIANCE-COVARIANCE AND CORRELATION MATRICES OF THE ESTIMATED PARAMETERS

- COVARIANCES ARE ABOVE THE DIAGONAL
- VARIANCES ARE ON THE DIAGONAL
- CORRELATION COEFFICIENTS ARE BELOW THE DIAGONAL

COLUMN	1	2	3	4
1	141.51474	.28758711	-.65179437	-1.6763208
2	.17926325	.18186730E-01	-.36510675E-01	-.71435215E-02
3	-.14887895	-.73264128	.13244186	.10476827E-04
4	-.90159992	-.33891642	.18214234E-03	.24427828E-01

----- ESTIMATES FROM FIT -----				----- ESTIMATES FROM FIT OMITTING LAST PREDICTOR VALUE -----					
ESTIMATED PARAMETER	SD OF PAR	T(PAR=0)	PROB(T)	ACC DIG <sup>9</sup>	ESTIMATED PARAMETER	SD OF PAR	T(PAR=0)	PROB(T)	
1	-39.9196744	11.8959969	-3.356	.004	14.1	5.13832806	-9.801	.000	
2	.715440200	.134858185	5.307	.000	14.1	.126621047	5.228	.000	
3	1.29528612	.368024265	3.520	.003	13.8	.367483444	3.525	.002	
4	-.152122519	.158294043	-.9733	.344	14.1				
RESIDUAL STANDARD DEVIATION		3.243364							
BASED ON DEGREES OF FREEDOM		21 - 4 =	17					21 - 3 =	18
MULTIPLE CORRELATION COEFFICIENT SQUARED		.9136							

\* THE NUMBER OF CORRECTLY COMPUTED DIGITS IN EACH PARAMETER USUALLY DIFFERS BY LESS THAN 1 FROM THE VALUE GIVEN HERE.

Figure F-1e  
Example of LLS output (continued)



Linear Least Squares  
With Polynomial Model (Design Matrix)

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE LLSP USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPES ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, X AND RES MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(30), X(30), RES(30)
C DOUBLE PRECISION DSTAK(500)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 500
C
C READ NUMBER OF OBSERVATIONS AND DEGREE OF THE POLYNOMIAL TO BE FIT
C INDEPENDENT AND DEPENDENT VARIABLES
C
C READ 100, N, NDEG
C READ 101, (X(I), Y(I), I=1,N)
C
C PRINT TITLE AND CALL LLSP TO PERFORM LINEAR LEAST SQUARES ANALYSIS
C
C WRITE (6, 102)
C CALL LLSP (Y, X, N, NDEG, RES, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I9)
C 101 FORMAT (2F9.1)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' LINEAR LEAST SQUARES SUBROUTINE LLSP')
C END
DATA:              9      2
                  0.0 12.0
                  1.0 10.5
                  2.0 10.0
                  3.0  8.0
                  4.0  7.0
                  5.0  8.0
                  6.0  7.5
                  7.0  8.5
                  8.0  9.0
```

Figure F-2a

Example program using LLSP

RESULTS OF STARPAC LINEAR LEAST SQUARES SUBROUTINE LLSP  
 \* LINEAR LEAST SQUARES ESTIMATION WITH POLYNOMIAL MODEL \*

RESULTS FROM LEAST SQUARES FIT

ROW	PREDICTOR VALUES	DEPENDENT VARIABLE	PREDICTED VALUE	STO DEV OF PREO VALUE	RESIDUAL	STD RES
1	0.	12.000000	12.104040	.41999992	-.10404040	-.61
2	1.000000	10.500000	10.521212	.27204429	-.21212121E-01	-.05
3	2.000000	10.000000	9.2233766	.23159593	.77662330	1.68
4	3.000000	8.000000	8.2913420	.24891687	-.29134199	-.64
5	4.000000	7.000000	7.7251082	.26112476	-.72510823	-1.63
6	5.000000	6.000000	7.5246753	.24891687	.47332468	1.03
7	6.000000	7.500000	7.6900493	.23159593	-.19004329	-.41
8	7.000000	8.500000	8.2212121	.27204429	.27878780	.64
9	8.000000	9.000000	9.1181818	.41999992	-.11818182	-.39

Figure F-2b

Example of LLSP output

LINEAR LEAST SQUARES ESTIMATION WITH POLYNOMIAL MODEL, CONTINUED

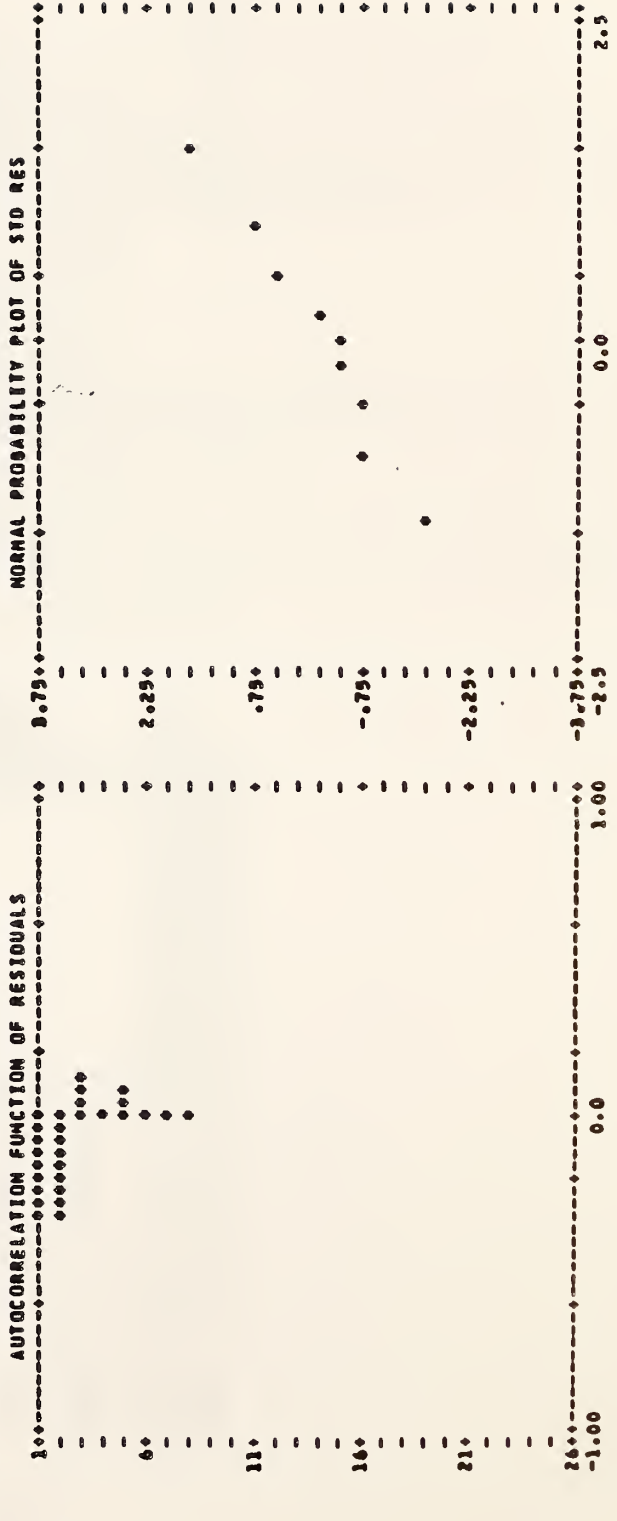
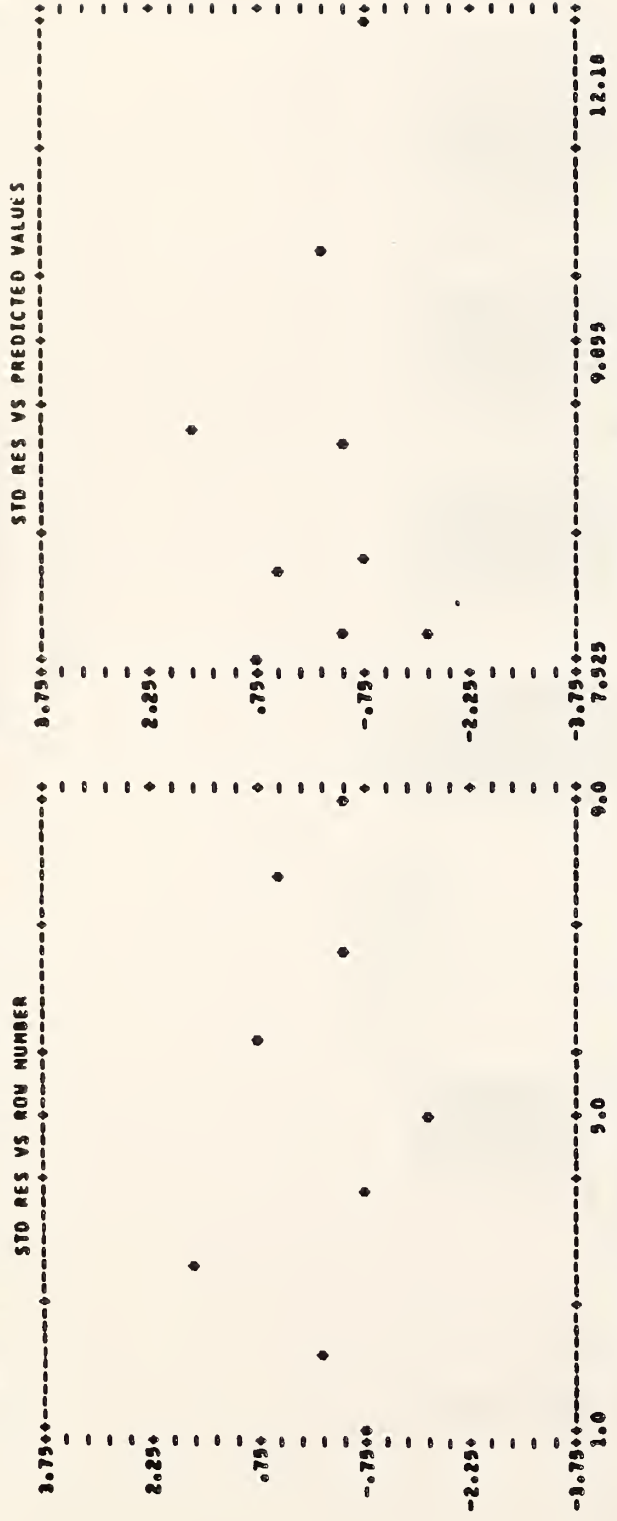


Figure F-2c  
Example of LLSP output (continued)

LINEAR LEAST SQUARES ESTIMATION WITH POLYNOMIAL MODEL, CONTINUED

ANALYSIS OF VARIANCE  
 --DEPENDENT ON ORDER VARIABLES ARE ENTERED, UNLESS VECTORS ARE ORTHOGONAL--

PAR INDEX	SUM OF SQUARES RED DUE TO PAR	CUM MS RED	OF(MSRED)	CUM RES MS	DF(RMS)	F	PAR=0 PROB(F)	F	PAR=0 PROB(F)
1	720.027778	720.027778	1	2.59027778	8	2696.46	.000	922.687	.000
2	6.81666667	364.422222	2	1.70079365	7	33.0178	.001	35.8017	.000
3	10.3033911	246.302612	3	.267027417	6	38.9855	.001	38.9855	.001
RESTOTAL	1.602165		6						
TOTAL	740.7500		9						

Figure F-2d

Example of LLSP output (continued)

LINEAR LEAST SQUARES ESTIMATION WITH POLYNOMIAL MODEL, CONTINUED  
 VARIANCE-COVARIANCE AND CORRELATION MATRICES OF THE ESTIMATED PARAMETERS

-----  
 - COVARIANCES ARE ABOVE THE DIAGONAL  
 - VARIANCES ARE ON THE DIAGONAL  
 - CORRELATION COEFFICIENTS ARE BELOW THE DIAGONAL  
 -----

COLUMN	1	2	3
1	.17639993	-.0233747E-01	.00917399E-02
2	-.00268762	.59936678E-01	-.09397771E-02
3	.65431992	-.96219769	.06697213E-03

----- ESTIMATES FROM FIT -----				----- ESTIMATES FROM FIT OMITTING LAST PREDICTOR VALUE -----				
ESTIMATED PARAMETER	SD OF PAR	T(PAR=0)	PROB(T)	ACC DIG	ESTIMATED PARAMETER	SD OF PAR	T(PAR=0)	PROB(T)
1	12.1646489	.41999917	29.01	.000	14.1	10.477770	.801574729	13.07
2	-1.64633680	.244019675	-7.542	.000	14.1	-.383333333	.168364369	-2.277
3	-.182900433	.294443905E-01	6.212	.001	14.0			.000
RESIDUAL STANDARD DEVIATION					1.304145			
BASED ON DEGREES OF FREEDOM				9 - 3 =	6			
MULTIPLE CORRELATION COEFFICIENT SQUARED					.9227			

\* THE NUMBER OF CORRECTLY COMPUTED DIGITS IN EACH PARAMETER USUALLY DIFFERS BY LESS THAN 1 FROM THE VALUE GIVEN HERE.

Figure F-2e  
 Example of LLSP output (continued)



G. Acknowledgments

The code and printed output for the linear least squares subroutines has been modeled on the linear least squares code and output used by OMNITAB II [Hogben et al. 1971].



## 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

$$y_i = f_i(x_i, \beta) + \epsilon_i \quad \text{for } i = 1, \dots, N,$$

where

$N$  is the number of observations;

$f_i$  is the function (nonlinear in its parameters) that models the  $i^{\text{th}}$  observation;

$x_i$  is the vector of the  $M$  independent variables at the  $i^{\text{th}}$  observation;

$\beta$  is the vector of the  $NPAR$  model parameters; and

$\epsilon_i$  is the unobservable random error in the  $i^{\text{th}}$  observation, which is estimated by the  $i^{\text{th}}$  residual.

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

$$D(i,k) = \partial f_i(x_i, \beta) / \partial \beta(k) \quad \text{for } i = 1, \dots, N \text{ and } k = 1, \dots, NPAR.$$

The derivative matrix may be supplied analytically or approximated numerically.

The least squares solution,  $\hat{\beta}$ , is that which minimizes (with respect to  $\beta$ ) the residual sum of squares function,

$$RSS(\beta) = \sum_{i=1}^N [(y_i - f_i(x_i, \beta))^2 \cdot wt_i] = \sum_{i=1}^N [\varepsilon_i^2 \cdot wt_i]$$

where carat (^) denotes the estimated quantity, and

wt<sub>i</sub> is the weight assigned to the i<sup>th</sup> observation (wt<sub>i</sub> = 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 §D) used to compute f<sub>i</sub>(x<sub>i</sub>, β<sub>ℓ</sub>), i = 1, ..., N, i.e., the predicted values of the dependent variable given the independent variables and the parameter values from iteration ℓ for ℓ = 1, 2, ... . 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 §B for a brief description of the subroutines. The declaration and CALL statements are given in §C, and the subroutine arguments are defined in §D. The algorithms used and the output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §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 §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
  - 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 §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 §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 §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 §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 §D, argument IDRVCCK.]

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 §D and §F, respectively. The conventions used to present the following declaration and CALL statements are given in chapter 1, §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)
```

===

*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,
1 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(npare, npare)
:
CALL NLSS (Y, XM, N, M, IXM, NLSMDL, PAR, NPAR, RES, LDSTAK,
1  IFIXED, STP, MIT, STOPSS, STOPP, SCALE, DELTA, IVAPRX, NPRT,
2  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,
1  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(np $ar$ )
<real> WT( $n$ )
<real> STP(np $ar$ ), STOPSS, STOPP, SCALE(np $ar$ ), DELTA
<real> RSD, PV( $n$ ), SDPV( $n$ ), SDRES( $n$ ), VCV(np $are$ ,np $are$ )
:
CALL NLSWS 'Y, WT, XM, N, M, IXM, NLSMDL, PAR, NPAR, RES, LDSTAK,
1  IFIXED, STP, MIT, STOPSS, STOPP, SCALE, DELTA, IVAPRX, NPRT,
2  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,
1  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(np $ar$ )
<real> STOPSS, STOPP, SCALE(np $ar$ ), DELTA
:
CALL NLSDC (Y, XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, RES,
1  LDSTAK, IFIXED, IDR $VCK$ , MIT, STOPSS, STOPP, SCALE, DELTA,
2  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(npare,npare)
:
CALL NLSDS (Y, XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, RES,
1  LDSTAK, IFIXED, IDRCK, MIT, STOPSS, STOPP, SCALE, DELTA,
2  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,
1  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,
1  LDSTAK, IFIXED, IDRCK, MIT, STOPSS, STOPP, SCALE, DELTA,
2  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,
1  LDSTAK, IFIXED, IDRCK, MIT, STOPSS, STOPP, SCALE, DELTA,
2  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)
```

===

*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,
1  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,
1 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, i.e.,  $\delta_0$ . [See §E.1.a.] The default value is 100.0. When DELTA < 0.0 or when DELTA is not an argument of the

--continued--

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,

$$\text{DELTA} = \min\{|\Delta_{\max}\beta(k)|/\text{SCALE}(k), \text{ for } k = 1, \dots, \text{NPAR}\}$$

where  $\Delta_{\max}\beta(k)$  is the maximum change allowed for the  $k^{\text{th}}$  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,  $\alpha = \text{EXMPT} \cdot N$ , 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 §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  $\neq$  0 the derivative is checked, and when IDRVCK = 0 it is not. The default value is IDRVCK  $\neq$  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, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

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.

--continued--



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 §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

--continued--

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 §A. The default values are 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:

$$(1) \text{ VCV} = \text{RSD}^2 \cdot (\hat{D}^T \cdot W \cdot \hat{D})^{-1}$$

$$(2) \text{ VCV} = \text{RSD}^2 \cdot H^{-1}$$

$$(3) \text{ VCV} = \text{RSD}^2 \cdot H^{-1} \cdot (\hat{D}^T \cdot W \cdot \hat{D}) \cdot H^{-1}$$

where

$$H = \hat{D}^T \cdot W \cdot \hat{D} + \left\{ \sum_{i=1}^N \hat{\epsilon}_i \cdot \text{wt}_i \cdot (\partial^2 \hat{\epsilon}_i / \partial \beta(j) \partial \beta(k)) \text{ for } j = 1, \dots, \text{NPAR} \right. \\ \left. \text{and } k = 1, \dots, \text{NPAR} \right\};$$

W is an N by N diagonal matrix of weights,

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

when a weighted analysis is performed, and  $\hat{D}$  is the identity matrix otherwise, and  $\hat{D}$  is the matrix that contains the partial derivatives of the model with respect to each parameter.

Approximation (1) is based on the assumption that  $H \sim \hat{D}^T \cdot W \cdot \hat{D}$  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 recent study by Donaldson and Schnabel [1985] 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. 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, 1985].

--continued--



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, §B.]

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

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

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:

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

For STPLS and STPLSC:

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

For DCKLS and DCKLSC:

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

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 I<sup>th</sup> digit (counting from left to right) is used to control the I<sup>th</sup> section of the output.

If the I<sup>th</sup> digit = 0 the output from the I<sup>th</sup> section is suppressed;  
 = 1 the brief form of the I<sup>th</sup> section is given;  
 > 2 the full form of the I<sup>th</sup> 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 §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.

--continued--



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,

$$PV(i) = f_i(x_i, \hat{\beta}) = \hat{y}_i \quad \text{for } i = 1, \dots, N.$$

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

$$RES(i) = y(i) - f_i(x_i, \hat{\beta}) = y(i) - \hat{y}(i) = \hat{\epsilon}(i) \quad \text{for } i = 1, \dots, N.$$

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

$$RSD = (RSS(\hat{\beta}) / (NNZW - NPARE))^{1/2}.$$

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

$$|\beta_\ell(j) / SCALE(j)| \approx |\beta_\ell(k) / SCALE(k)| \quad \text{for } k = 1, \dots, NPAR \\ \text{and } j = 1, \dots, NPAR.$$

Values of  $|SCALE(k)| > |\beta_\ell(k)|$  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  $\beta_\ell(k)$ .

--continued--

For the estimation subroutines:

The default values for SCALE are selected by the NL2SOL algorithm [Dennis et al. 1981a and 1981b] 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)  $\leq 0$  the default procedure will be used to select scale values. When SCALE(1)  $> 0$ , values of SCALE(k)  $\leq 0$  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 as:

$$\left. \begin{array}{l} \text{SCALE}(k) = 1.0 \quad \text{if } \text{PAR}(k) = 0.0 \\ \text{SCALE}(k) = |\text{PAR}(k)| \quad \text{otherwise} \end{array} \right\} \text{ for } k = 1, \dots, NPAR.$$

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

$$\text{SDPV}(i) = \text{the } i^{\text{th}} \text{ diagonal element of } [(\hat{D} \cdot \text{VCV} \cdot \hat{D}^T)]^{1/2}$$

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

$$\text{SDRES}(i) = \text{RES}(i) / [(\text{RSD}^2 / \text{WT}(i)) - \text{SDPV}(i)^2]^{1/2}$$

for  $i = 1, \dots, N$ , which is the  $i^{\text{th}}$  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

$$\frac{\max\{|\beta_{\ell+1}(k) - \beta_{\ell}(k)| / \text{SCALE}(k) \text{ for } k = 1, \dots, \text{NPAR}\}}{\max\{(|\beta_{\ell+1}(k)| + |\beta_{\ell}(k)|) / \text{SCALE}(k) \text{ for } k = 1, \dots, \text{NPAR}\}} < \text{STOPP}.$$

[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  $\max\{|\beta_{\ell+1}(k) - \beta_{\ell}(k)| / |\beta_{\ell}(k)| \text{ for } k = 1, \dots, \text{NPAR}\}$ . 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,  $\Delta \text{RSS}_{\ell+1}$ , to the current residual sum of squares,  $\text{RSS}(\beta_{\ell})$ . The convergence criterion is satisfied if certain conditions are met and

$$\Delta \text{RSS}_{\ell+1} / \text{RSS}(\beta_{\ell}) < \text{STOPSS}.$$

[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  $\ell$  and  $\ell+1$  as measured by  $(s_{\ell} - s_{\ell+1}) / s_{\ell}$ . 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  $10^{-2 \cdot \text{DIGITS}/3}$ , 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 §E.1.b. For the estimation 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  $j^{\text{th}}$  column contains the N values of the  $j^{\text{th}}$  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 and 1981b]. 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 §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 §D, arguments STOPP and STOPSS], the maximum number of iterations (or model subroutine calls) is reached [see §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, §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 iteration  $\ell+1$  the values of the parameter vector  $\beta_{\ell+1}$  are given by

$$\beta_{\ell+1} = \beta_{\ell} - (D_{\ell}^T \cdot W \cdot D_{\ell} + S_{\ell} + \Lambda_{\ell})^{-1} \cdot D_{\ell}^T \cdot W \cdot \epsilon_{\ell}^T$$

subject to the restriction that

$$\left\{ \sum_{k=1}^{\text{NPAR}} [(\beta_{\ell+1}(k) - \beta_{\ell}(k)) / \text{SCALE}(k)]^2 \right\}^{1/2} < \delta_{\ell},$$

where

$\beta_{\ell}$  is the vector of NPAR estimated parameter values from the  $\ell^{\text{th}}$  iteration.

$D_{\ell}$  is the N by NPAR matrix of the partial derivatives,

$$D_{\ell}(i, k) = \partial f_i(x_i, \beta_{\ell}) / \partial \beta_{\ell}(k) \quad \text{for } i = 1, \dots, N \text{ and } k = 1, \dots, \text{NPAR}.$$

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.

$S_{\ell}$  is an approximation to the exact term  $S_{\ell}^*$  in the matrix of second order terms (Hessian) of the Taylor series expansion of the residual sum of squares function,

$$S_{\ell}^*(j, k) = \sum_{i=1}^N [\epsilon_{\ell i} \cdot \text{wt}_i \cdot (\partial^2 \epsilon_{\ell}(i) / \partial \beta_{\ell}(j) \partial \beta_{\ell}(k))],$$

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

$\epsilon_{\ell}$  is the vector of the N residuals from the previous iteration.



$\delta_\ell$  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  $\ell$ ,  $\delta_\ell$  is computed based on information from the previous iteration. At the first iteration, the initial value,  $\delta_0$ , is supplied by argument DELTA which can be used to control the change in the parameters permitted at the first iteration.

$\Lambda_\ell$  is an NPAR by NPAR diagonal matrix,

$$\Lambda_\ell = \text{diag}\{\lambda_\ell/\text{SCALE}(k), k = 1, \dots, \text{NPAR}\},$$

where  $\lambda_\ell$  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  $S_\ell^*$ , which is expensive and difficult to compute accurately, is important only if it is large compared to the term  $D_\ell^T \cdot W \cdot D_\ell$ , that is, when the residuals are large or the model is highly nonlinear. When  $S_\ell^*$  is large compared to  $D_\ell^T \cdot W \cdot D_\ell$ , 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  $S_\ell^*$  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 §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  $\eta/2$  and  $\eta$ , respectively, where  $\eta$  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  $\eta$  would be five; if the predicted values are calculated from a simple algebraic expression translated directly into Fortran code, then  $\eta$  would (usually) be the number of decimal digits carried by the user's computer for the results.

The relative step size for  $\beta(k)$ ,  $k = 1, \dots, NPAR$ , is initially

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

where

$\gamma$  is the average curvature (estimated by STARPAC) of the model with respect to  $\beta(k)$ .

The forward difference quotient approximations with respect to  $\beta(k)$ ,  $k = 1, \dots, NPAR$  are then

$$D_{fd}(i,k) = \frac{f_i(x_i, \beta^k) - f_i(x_i, \beta)}{STP(k) \cdot SCALE(k) \cdot SIGN(\beta(k))} \quad \text{for } i = 1, \dots, N,$$

where

$f_i$  is the function which models the  $i^{\text{th}}$  observation.

$x_i$  is the vector of the values of the M independent variables at the  $i^{\text{th}}$  observation.

$\beta$  is the vector of the NPAR parameter values.

$\beta^k$  is a vector which has the same values as  $\beta$  except that the  $k^{\text{th}}$  parameter is equal to

$$\beta(k) + (STP(k) \cdot SCALE(k) \cdot SIGN(\beta(k))).$$

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

The central difference approximations to the model derivative with respect to  $\beta(k)$ ,  $k = 1, \dots, \text{NPAR}$ , are

$$D_{cd}(i,k) = \frac{f_i(x_i, \beta^{+k}) - f_i(x_i, \beta^{-k})}{3^{1/3} \cdot 10^{-\text{NETA}/3} \cdot \text{SCALE}(k) \cdot \text{SIGN}(\beta(k))} \quad \text{for } i = 1, \dots, N,$$

where

$\beta^{+k}$  is a vector which has the same values as  $\beta$  except that the  $k^{\text{th}}$  parameter is equal to

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

$\beta^{-k}$  is a vector which has the same values as  $\beta$  except that the  $k^{\text{th}}$  parameter is equal to

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

The relative step size is considered acceptable if, for at least  $N-\alpha$  observations,

$$|D_{fd}(i,k) - D_{cd}(i,k)| < \min\{10^{-\text{NETA}/4}, 10^{-2}\} \quad \text{for } i = 1, \dots, N,$$

where  $\alpha$  is the number of observations exempted from meeting the above acceptance criterion. [See \$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 coefficient  $\beta(k)$ ,  $k = 1, \dots, \text{NPAR}$ , if

$$|D_{fd}(i,k) - D(i,k)| < 10^{-\tau} |D(i,k)|$$

where

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

$D_{fd}$  is the forward difference quotient approximation to the derivative described in §E.1.b.

$\tau$  is the agreement tolerance, i.e., number of digits of agreement required between  $D$  and  $D_{fd}$ , which must be less than or equal to the number of good digits in  $D_{fd}$ . [See §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  $(x_i - \beta(k))$  when  $x_i = \beta(k)$ , 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 §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 §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 underlined, uppercase phrases. Results which correspond to input or returned subroutine CALL statement arguments are identified by the argument name in uppercase (not underlined).

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, 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 §E.1.b and §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,

$$r_{jk} = VCV(j,k) / (VCV(j,j) VCV(k,k))^{1/2} \quad \begin{array}{l} \text{for } j = 1, \dots, NPARE \\ \text{and } k = 1, \dots, NPARE. \end{array}$$

- PARAMETER: the final value of each parameter,  $PAR(k)$ ,  $k = 1, \dots, NPAR$ .
- SD OF PAR: the standard deviation of each estimated parameter,  
 $(VCV(k,k))^{1/2}$  for  $k = 1, \dots, NPAR$ .
- RATIO: the ratio of each estimated parameter to its standard deviation,

$$RATIO_k = PAR(k) / (VCV(k,k))^{1/2} \quad \text{for } k = 1, \dots, NPAR.$$

- 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,  $RSS(\hat{\beta})$ .
- 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: 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 §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 §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,  $\beta(k)$ ,  $k = 1, \dots, NPAR$ . 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

$$f_i(x_i, \beta) = \beta(1) \cdot x_i(1) \beta(2) \quad \text{for } i = 1, \dots, N.$$

Nonlinear Least Squares Estimation. In the example program of figure F-1a, NLS is used to compute the least squares solution using numerically approximated derivatives; figures F-1b through F-1f show the output from NLS.

In the example program of figures F-2a and F-2b, NLSD is used to compute the least squares solution given analytic derivatives; figures F-2c through F-2g show the output from NLSD.

Derivative Step Size Selection. In the example program of figure F-3a, STPLS is used to compute the optimum step sizes for numerically approximating the derivatives with respect to each of the parameters,  $\beta(k)$ ,  $k = 1, 2$ . Figure F-3b shows the output from STPLS.

Derivative Checking. In the example program of figures F-4a and F-4b, DCKLS is used to check the validity of a user-supplied derivative subroutine. In figure F-4b, 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. Figure F-4c shows the output from DCKLS.



Nonlinear Least Squares Estimation  
With Numerically Approximated Derivatives

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE NLS USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, XM, PAR AND RES MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(10), XM(10,5), PAR(5), RES(10)
C DOUBLE PRECISION DSTAK(200)
C
C EXTERNAL NLSMDL
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C IXM = 10
C
C READ NUMBER OF PARAMETERS
C STARTING VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS AND NUMBER OF INDEPENDENT VARIABLES
C INDEPENDENT AND DEPENDENT VARIABLES
C
C READ 100, NPAR
C READ 101, (PAR(I), I=1,NPAR)
C READ 100, N, M
C READ 101, ((XM(I,J), I=1,N), J=1,M), (Y(I), I=1,N)
C
C PRINT TITLE AND CALL NLS TO PERFORM NONLINEAR REGRESSION
C WITH NUMERICALLY APPROXIMATED DERIVATIVES
C
C WRITE (6, 102)
C CALL NLS (Y, XM, N, M, IXM, NLSMDL, PAR, NPAR, RES, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I5)
C 101 FORMAT (6F6.3)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' NONLINEAR LEAST SQUARES SUBROUTINE NLS')
C END
MODEL SUBROUTINE: 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 PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL PAR(NPAR), XM(IXM,M), PV(N)
C
C DO 10 I = 1, N
C   PV(I) = PAR(1) * XM(I, 1) ** PAR(2)
C 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

```

Figure F-1a

Example program using NLS

RESULTS OF STARPAC NONLINEAR LEAST SQUARES SUBROUTINE NLS  
 \* NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES \*

SUMMARY OF INITIAL CONDITIONS

INDEX	PARAMETER	STARTING VALUE	SCALE (SCALE)	STEP SIZE FOR APPROXIMATING DERIVATIVE (STP)	COUNT	NOTES F C	ROW NUMBER
1	NO	.72500000	DEFAULT	.5969982E-04	0		
2	NO	4.00000000	DEFAULT	.56513757E-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 (META) 12

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

Figure F-1b

Example of NLS output

NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES, CONTINUED

```

ITERATION NUMBER 1
-----
MODEL CALLS      RSD      RSS      REL CHNG RSS      REL CHNG PAR
                .3390E-01 .4597E-02 .6877          VALUE      CHKD
                .1790E-01 Y
CURRENT PARAMETER VALUES
INDEX 1
VALUE .7679852      2
                3.8599309
    
```

```

ITERATION NUMBER 5
-----
MODEL CALLS      RSD      RSS      REL CHNG RSS      REL CHNG PAR
                .3285E-01 .4317E-02 -.1864E-12 .3271E-15
                .2423E-09 Y
CURRENT PARAMETER VALUES
INDEX 1
VALUE .7688623      2
                3.860406
    
```

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

Figure F-1c

Example of NLS output (continued)

NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES, CONTINUED

RESULTS FROM LEAST SQUARES FIT

ROW	PREDICTOR VALUES	DEPENDENT VARIABLE	PREDICTED VALUE	STD DEV OF PRED VALUE	RESIDUAL	STD RES
1	1.3090000	2.1380000	2.174175	.22079042E-01	-.36117493E-01	-1.48
2	1.4710000	3.4210000	3.4111549	.16469587E-01	.98450824E-02	.35
3	1.4900000	3.5970000	3.5984108	.19619321E-01	.12989150E-01	.44
4	1.5650000	4.3400000	4.3326419	.14065814E-01	.73580832E-02	.25
5	1.6110000	4.8820000	4.8453073	.16912112E-01	.36692701E-01	1.29
6	1.6880000	5.6680000	5.6968369	.26183728E-01	-.36836491E-01	-1.86

Figure F-1d

Example of NLS output (continued)



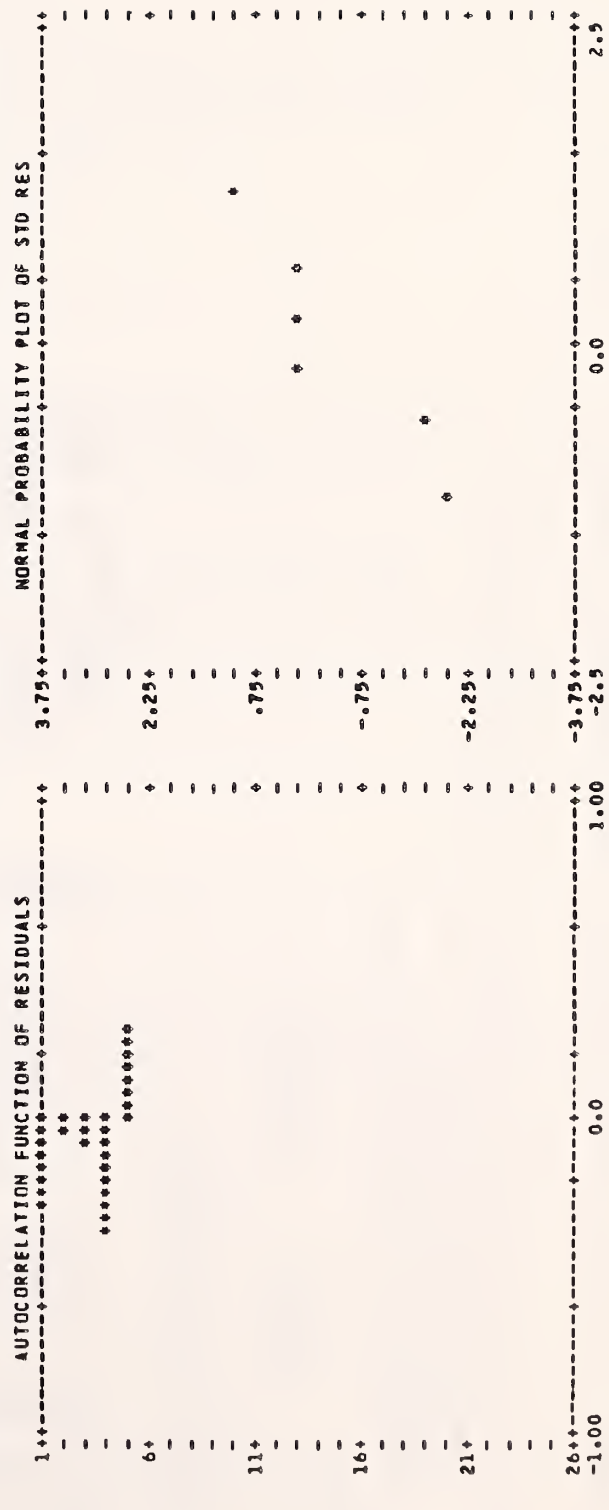
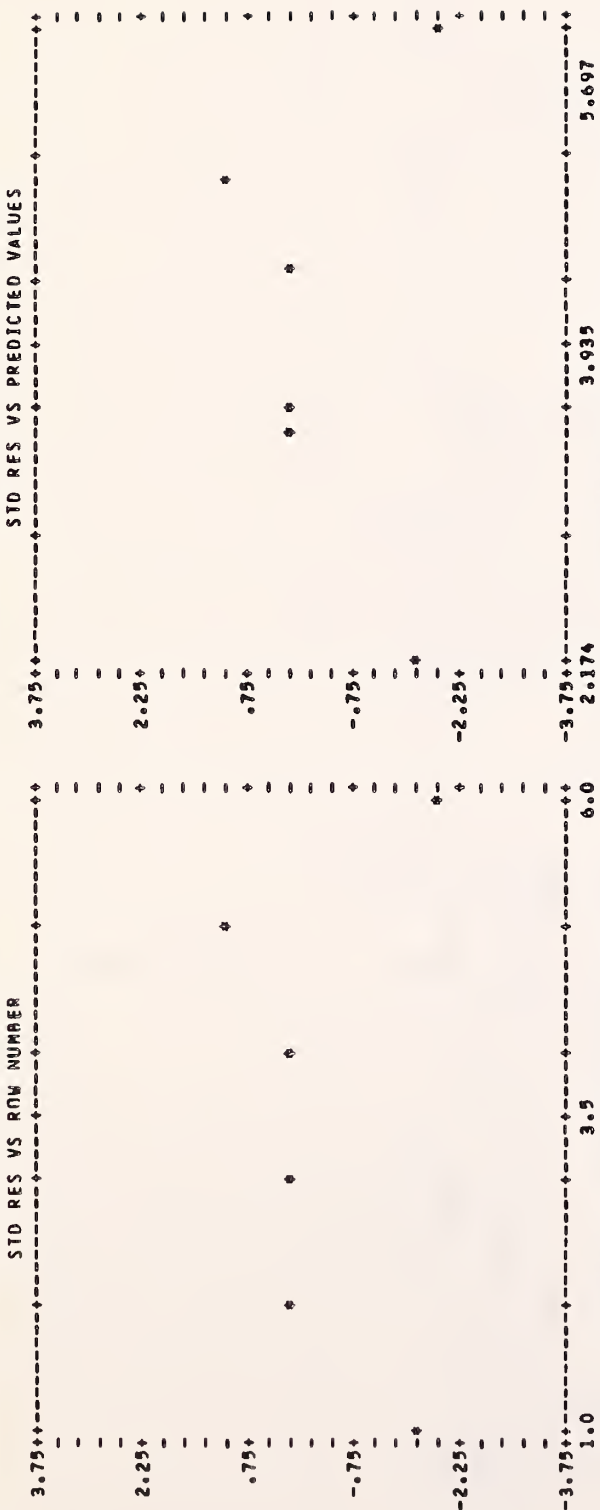


Figure F-1e  
Example of NLS output (continued)

NONLINEAR LEAST SQUARES ESTIMATION WITH NUMERICALLY APPROXIMATED DERIVATIVES, CONTINUED

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

- 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	.3342303E-03	-.9369366E-03
2	-.9907719	.2679637E-02

ESTIMATES FROM LEAST SQUARES FIT

INDEX	FIXED	PARAMETER	SD OF PAR	RATIO	95 PERCENT CONFIDENCE LOWER	APPROXIMATE UPPER LIMITS
1	NO	.76886226	.18291966E-01	42.06	.71810339	.81962114
2	NO	3.8604056	.51726563E-01	74.63	3.7167896	4.0040215

RESIDUAL SUM OF SQUARES .4317308E-02

RESIDUAL STANDARD DEVIATION .3289311E-01

BASED ON DEGREES OF FREEDOM 6 -- 2 = 4

APPROXIMATE CONDITION NUMBER 20.87491

Figure F-1f  
Example of NLS output (continued)

Nonlinear Least Squares Estimation  
With User-Supplied Analytic Derivatives

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE NLSM USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, XM, PAR AND RES MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(10), XM(10,5), PAR(5), RES(10)
C DOUBLE PRECISION DSTAK(200)
C
C EXTERNAL NLSMDL, NLSDRV
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C IXM = 10
C
C READ NUMBER OF PARAMETERS
C STARTING VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS AND NUMBER OF INDEPENDENT VARIABLES
C INDEPENDENT AND DEPENDENT VARIABLES
C
C READ 100, NPAR
C READ 101, (PAR(I), I=1,NPAR)
C READ 100, N, M
C READ 101, ((XM(I,J), I=1,N), J=1,M), (Y(I), I=1,N)
C
C PRINT TITLE AND CALL NLSM TO PERFORM NONLINEAR REGRESSION
C WITH USER-SUPPLIED DERIVATIVES
C
C WRITE (6, 102)
C CALL NLSM (Y, XM, N, M, IXM, NLSMDL, NLSDRV, PAR, NPAR, RES,
C * LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I5)
C 101 FORMAT (6F6.3)
C 102 FORMAT ('RESULTS OF STARPAC',
C * 'NONLINEAR LEAST SQUARES SUBROUTINE NLSM')
C END
MODEL SUBROUTINE: 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 PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL PAR(NPAR), XM(IXM,M), PV(N)
C
C DO 10 I = 1, N
C PV(I) = PAR(1) * XM(I, 1) ** PAR(2)
10 CONTINUE
C
C RETURN
C END

```

Figure F-2a

Example program using NLSM

```

DERIVATIVE SUBROUTINE:      SUBROUTINE NLSDRV (PAR, NPAR, XM, M, N, IXM, D)
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 PRECISION
C      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, M
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 1.965 1.611 1.680
           2.138 3.421 3.597 4.340 4.682 5.660

```

Figure F-2b

Example program using NLSD (continued)





NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED

```

ITERATION NUMBER 1
-----
MODEL          RSS          REL CHNG RSS          FORECASTED
CALLS          VALUE          CHKO          VALUE          CHKO          Y
2             .3390E-01          .4397E-02          .6877          .7109          Y          .1790E-01          Y
CURRENT PARAMETER VALUES
INDEX 1
VALUE .7679852
    
```

```

ITERATION NUMBER 4
-----
MODEL          RSS          REL CHNG RSS          FORECASTED
CALLS          VALUE          CHKO          VALUE          CHKO          Y
9             .3205E-01          .4317E-02          -.9214E-13          .6352E-12          Y          .1068E-07          Y
CURRENT PARAMETER VALUES
INDEX 1
VALUE .7688623
    
```

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

Figure F-2d  
Example of NLSD output (continued)

NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED

RESULTS FROM LEAST SQUARES FIT

ROW	PREDICTOR VALUES	DEPENDENT VARIABLE	PREDICTED VALUE	STD DEV OF PRED VALUE	RESIDUAL	STD RES
1	1.3090000	2.1380000	2.1741175	.22079044E-01	-.36117923E-01	-1.48
2	1.4710000	3.4210000	3.4111549	.18469585E-01	.98450648E-02	.35
3	1.4900000	3.9970000	3.9844109	.19619321E-01	.12589135E-01	.44
4	1.5650000	4.3400000	4.3326419	.14065814E-01	.73580808E-02	.29
5	1.6110000	4.8620000	4.8493073	.16512112E-01	.36692709E-01	1.29
6	1.6800000	5.6600000	5.6568365	.26183777E-01	-.38836464E-01	-1.86

Figure F-2e  
Example of NLSD output (continued)

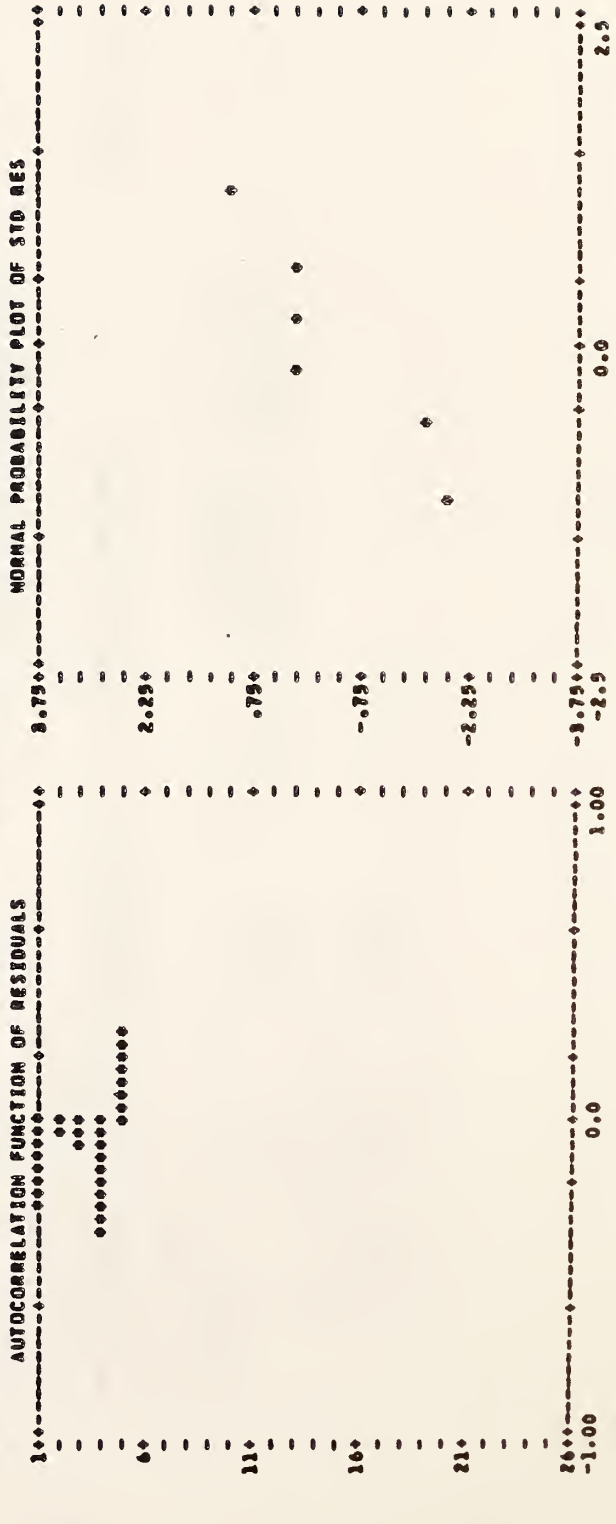
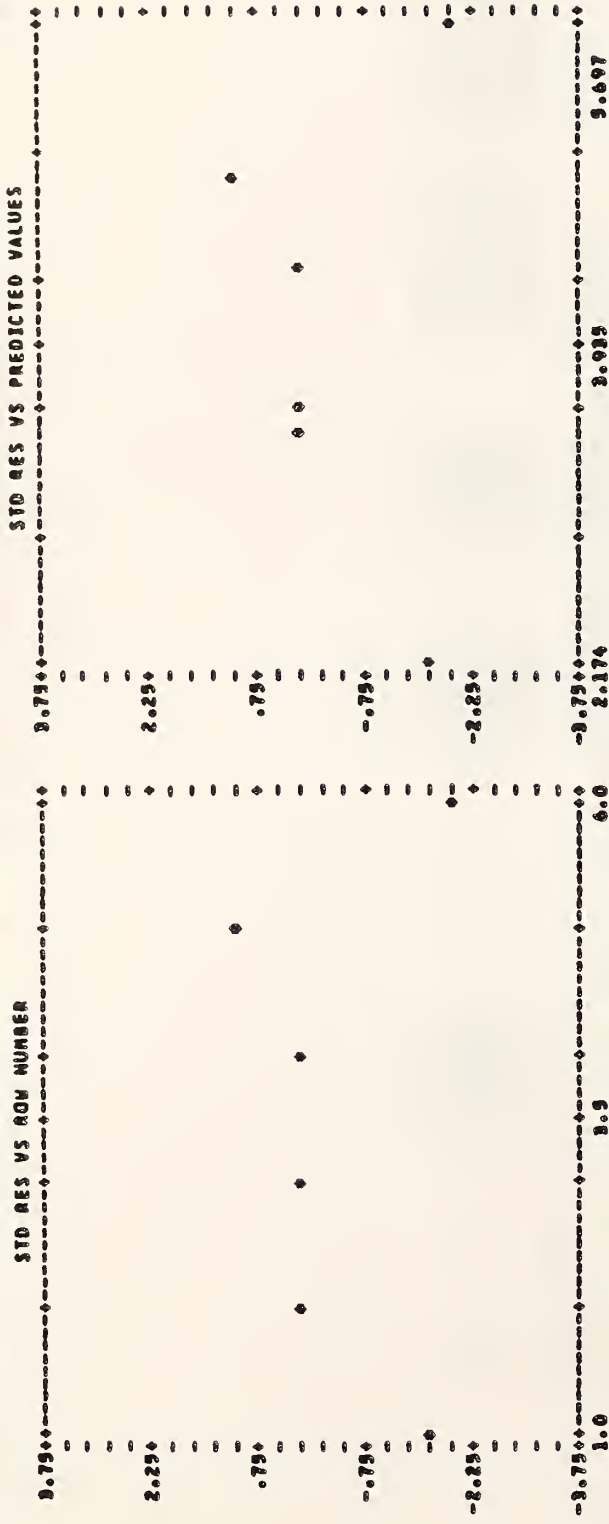


Figure F-2f

Example of NLSD output (continued)



NONLINEAR LEAST SQUARES ESTIMATION WITH USER-SUPPLIED DERIVATIVES, CONTINUED

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

- 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	.2679642E-02

ESTIMATES FROM LEAST SQUARES FIT

INDEX	FIXED	PARAMETER	SD OF PAR	RATIO	95 PERCENT CONFIDENCE LIMITS
					LOWER UPPER
1	NO	.76886229	.18281974E-01	42.06	.71810339 .81962119
2	NO	3.8604095	.91726811E-01	74.63	3.7167894 4.0090216

RESIDUAL SUM OF SQUARES .4317808E-02

RESIDUAL STANDARD DEVIATION .3289311E-01

BASED ON DEGREES OF FREEDOM 6 - 2 = 4

APPROXIMATE CONDITION NUMBER 20.87492

Figure F-2g

Example of NLS output (continued)

## Derivative Step Size Selection

```
MAIN PROGRAM:      PROGRAM EXAMPL
                   C
                   C DEMONSTRATE STPLS USING SINGLE PRECISION VERSION OF STARPAC
                   C RUN ON CYBER 180/840.
                   C
                   C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
                   C [SEE CHAPTER 1, SECTION D.4]
                   C
                   C N.B. DECLARATION OF XM, PAR AND STP MUST BE CHANGED TO DOUBLE PRECISION
                   C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
                   C
                   C REAL XM(10,5), PAR(5), STP(5)
                   C DOUBLE PRECISION DSTAK(200)
                   C
                   C EXTERNAL NLSMOL, DERIV
                   C
                   C COMMON /CSTAK/ DSTAK
                   C
                   C SPECIFY NECESSARY DIMENSIONS
                   C
                   C LDSTAK = 200
                   C 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
                   C READ 100, NPAR
                   C READ 101, (PAR(I), I=1,NPAR)
                   C READ 100, N, M
                   C READ 101, ((XM(I,J), I=1,N), J=1,M)
                   C
                   C PRINT TITLE AND CALL STPLS TO SELECT STEP SIZES FOR
                   C APPROXINATING DERIVATIVES
                   C
                   C WRITE (6, 102)
                   C CALL STPLS (XM, N, M, IXM, NLSMOL, PAR, NPAR, LDSTAK, STP)
                   C
                   C STOP
                   C
                   C FORMAT STATEMENTS
                   C
                   C 100 FORMAT (2I5)
                   C 101 FORMAT (6F6.3)
                   C 102 FORMAT ('RESULTS OF STARPAC',
                   C * ' DERIVATIVE STEP SIZE SELECTION SUBROUTINE STPLS')
                   C END
MODEL SUBROUTINE: SUBROUTINE NLSMOL (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 PRECISION
                   C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
                   C
                   C REAL PAR(NPAR), XM(IXM,M), PV(N)
                   C
                   C DO 10 I = 1, M
                   C PV(I) = PAR(1) * XM(I, 1) ** PAR(2)
                   C 10 CONTINUE
                   C
                   C RETURN
                   C END
DATA:              2
                   0.725 4.000
                   6 1
                   1.300 1.471 1.490 1.565 1.611 1.680
```

Figure F-3a

Example program using STPLS

RESULTS OF STARPAC DERIVATIVE STEP SIZE SELECTION SUBROUTINE STPLS

\*\*\*\*\*  
 \* DERIVATIVE STEP SIZE SELECTION \*  
 \*\*\*\*\*

INDEX	PARAMETER STARTING VALUE (PAR)	SCALE (SCALE)	STEP SIZE FOR APPROXIMATING DERIVATIVE (STP)	OBSERVATIONS FAILING STEP SIZE SELECTION CRITERIA COUNT	NOTES F C	ROW NUMBER(S)
1	.72500000	DEFAULT	.59658982E-04	0		
2	4.00000000	DEFAULT	.56518757E-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 (META) 12

PROPORTION OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA (EXMPT) .1000

NUMBER OF OBSERVATIONS EXEMPTED FROM SELECTION CRITERIA 1

NUMBER OF OBSERVATIONS (N) 6

Figure F-3b  
 Example of STPLS output

## Derivative Checking

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE DCKLS USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 100/040.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C (SEE CHAPTER 1, SECTION 0.4)
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
C REAL XM(10,5), PAR(5)
C DOUBLE PRECISION DSTAK(200)
C
C EXTERNAL NLSMDL, NLSDRV
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 200
C 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
C READ 100, NPAR
C READ 101, (PAR(I), I=1,NPAR)
C READ 100, M, N
C READ 101, ((XM(I,J), I=1,N), J=1,M)
C
C PRINT TITLE AND CALL DCKLS TO PERFORM DERIVATIVE CHECKING
C
C WRITE (6, 102)
C CALL DCKLS (XM, M, N, IXM, NLSMDL, NLSDRV, PAR, NPAR, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (2I5)
C 101 FORMAT (6F6.3)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' DERIVATIVE CHECKING SUBROUTINE DCKLS')
C END
MODEL SUBROUTINE: SUBROUTINE NLSMDL (PAR, NPAR, XM, M, N, 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 PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL PAR(NPAR), XM(IXM,N), PV(N)
C
C DO 10 I = 1, N
C PV(I) = PAR(1) * XM(I, 1) ** PAR(2)
C 10 CONTINUE
C
C RETURN
C END
```

Figure F-4a

Example program using DCKLS



```

DERIVATIVE SUBROUTINE:      SUBROUTINE NLSORV (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 PRECISION
C      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, M
C          D(I,1) = XM(I,1) * PAR(2)
C          D(I,2) = PAR(1) * XM(I,1) ** PAR(1) * ALOG(XM(I,1))
10 CONTINUE
C
C      RETURN
C      END
DATA:
      2
      0.000 4.000
      6 1
      1.309 1.471 1.490 1.565 1.611 1.680

```

Figure F-4b

Example program using DCKLS (continued)

RESULTS OF STARPAC DERIVATIVE CHECKING SUBROUTINE DCKLS

\*\*\*\*\*  
 \* DERIVATIVE CHECKING \*  
 \*\*\*\*\*

INDEX	PARAMETER STARTING VALUE (PAR)	SCALE (SCALE)	DERIVATIVE ASSESSMENT
1	0.	DEFAULT	INCORRECT
2	4.000000	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 (META) 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 1.209000

NUMBER OF OBSERVATIONS (N) 6

Figure F-4c

Example of DCKLS output

## 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].





## CHAPTER 10

### DIGITAL FILTERING

#### A. Introduction

STARPAC contains 16 subroutines for digital filtering time series. These include subroutines which: compute a least squares approximation to an ideal low-pass filter; perform symmetric linear filter operations; sample values from a series; perform autoregressive (or difference) filter operations; and compute the gain function of any symmetric linear filter and the gain and phase functions of any autoregressive (or difference) filter.

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and the output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

#### B. Subroutine Descriptions

##### B.1 Symmetric Linear Filter Subroutines

Subroutine LPCOEF computes symmetric linear low-pass filter coefficients using a least squares approximation to an ideal low-pass filter that has convergence factors which reduce overshoot and ripple [Bloomfield, 1976]. This low-pass filter has a transfer function which changes from approximately one to zero in a transition band about the ideal cutoff frequency  $FC$ , that is from  $(FC - 1/K)$  to  $(FC + 1/K)$ , as shown in figure B.1-1. The user must specify the cutoff frequency in cycles per sample interval and the number of filter coefficients, which must be odd. The user must also choose the number of filter terms,  $K$ , so that  $(FC - 1/K) > 0$  and  $(FC + 1/K) < 0.5$ . In addition,  $K$  must be chosen as a compromise between:

- 1) A sharp cutoff, that is,  $1/K$  small; and
- 2) Minimizing the number of data points lost by the filtering operations ( $(K - 1)/2$  data points will be lost from each end of the series).

The subroutine returns the normalized low-pass filter coefficients. There is no printed output.

For any low-pass filter there is a corresponding high-pass filter equivalent to subtracting the low-pass filtered series from the original series. Subroutine HPCOEF returns symmetric linear high-pass filter coefficients computed from user supplied low-pass symmetric linear filter coefficients. The number of filter coefficients must be odd. There is no printed output.

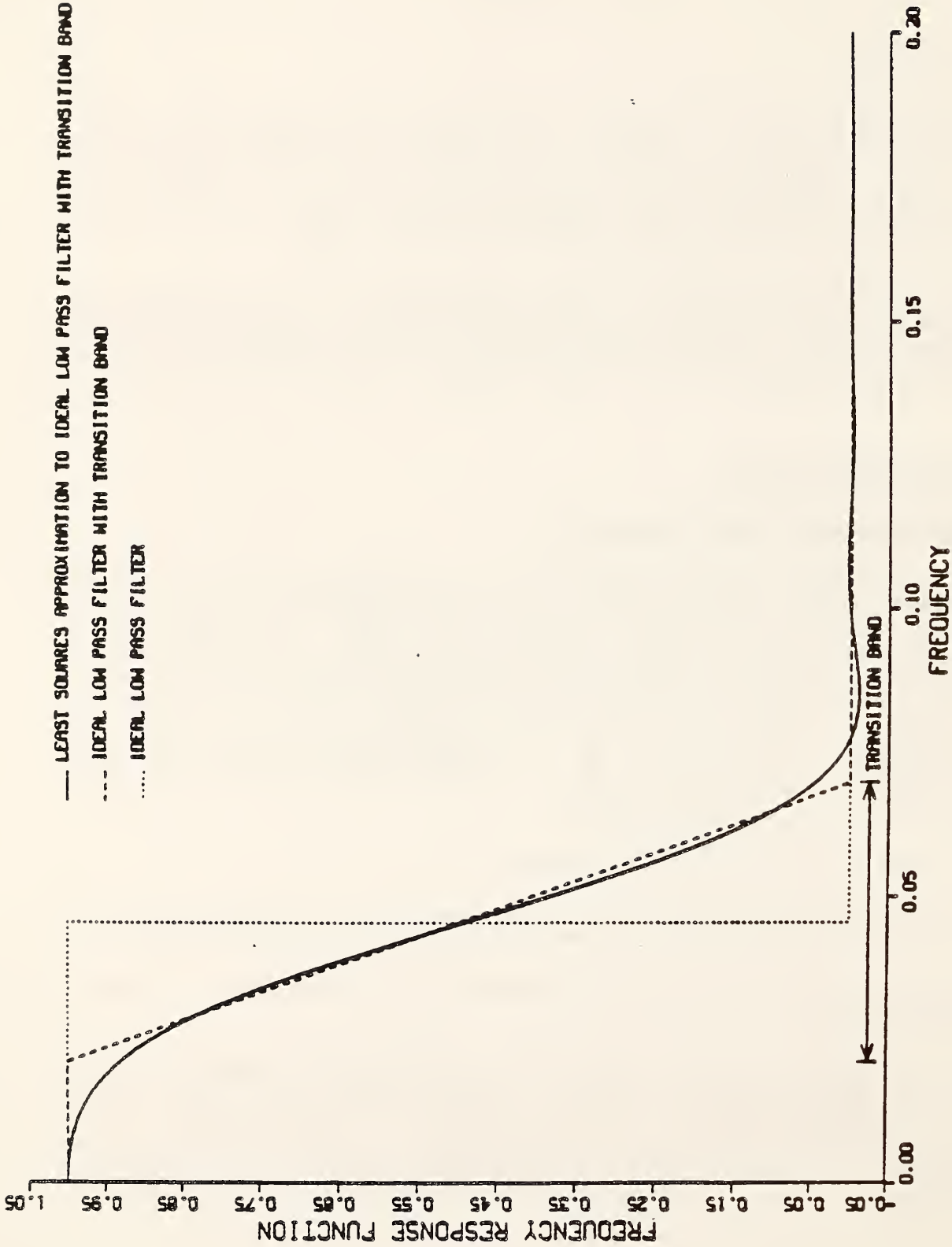


Figure B.1-1

Transfer function of 41-term least squares low-pass filter with a cutoff frequency of 1/22 cycles per sample interval

Subroutine MAFLT performs a simple moving average filter operation on the input series using the simple moving average filter defined by

$$HMA(J) = 1/K \text{ for } J = 1, \dots, K.$$

The user must specify the number of filter coefficients, K, which must be odd; the subroutine returns the filtered series and the number of observations in the filtered series. There is no printed output.

Subroutine SLFLT performs a symmetric linear filter operation with user supplied coefficients and returns the filtered series to the user. The filter coefficients must be normalized on input to SLFLT. The filtered series and the number of observations in the filtered series are returned. There is no printed output.

Subroutine SAMPLE samples every  $NS^{\text{th}}$  observation from an input series. If the input series was obtained using an NS term low-pass filter, this sampling rate removes the autocorrelation introduced by the filtering operation. This subroutine returns the series of sampled observations and the number of observations in the series. There is no printed output.

Subroutine LOPASS computes low-pass filter coefficients as described for subroutine LPCOEF and then performs the filtering operation described for subroutine SLFLT. The user must specify the cutoff frequency in cycles per sample interval and the number of filter terms, which must be odd. The subroutine returns the normalized filter coefficients, the filtered series and the number of observations in the filtered series. There is no printed output.

Subroutine HIPASS computes the high-pass filter coefficients equivalent to using HPCOEF with the input low-pass filter coefficients supplied by LPCOEF and performs the filtering operation described for subroutine SLFLT. The user must specify the cutoff frequency in cycles per sample interval and the number of filter terms, which must be odd. The subroutine returns the filter coefficients, the filtered series and the number of observations in the filtered series. There is no printed output.

## B.2 Autoregressive or Difference Linear Filter Subroutines

Subroutine ARFLT subtracts the series mean from each observation and performs an autoregressive linear filter operation with user supplied filter coefficients. This subroutine returns the filtered series and the number of observations in the filtered series. There is no printed output.

Subroutine DIF performs a first difference filter operation on the input series. It returns the differenced series and the number of observations in the differenced series. There is no printed output.

Subroutine DIFC performs a user controlled differencing operation. It returns the order of the difference filter specified, the coefficients of the difference filter, the differenced series and the number of observations in



the differenced series. This subroutine can be used as a high-pass filter or for differencing series in the style of Box and Jenkins [1976]. There is no printed output.

Subroutines DIFM and DIFMC are the same as DIF and DIFC, respectively, except that the input series may contain missing data. A missing value code must be used within the input series to specify time points without an observed value. The difference between a missing value and an observed value, or between two missing values, will result in a missing value in the differenced series; the missing value code used in the differenced series is also returned to the user. Users should note that the number of missing values may be significantly increased by the differencing operation.

### B.3 Gain and Phase Function Subroutines

Subroutine GFSLF computes the gain function of a symmetric linear filter. The printed output consists of a plot of the gain function versus frequency.

Subroutine GFSLFS is the same as GFSLF but allows the user to specify various options which are preset in GFSLF, including the frequency range, the number of frequencies for which the gain function is to be computed and the type of plot to be used. In addition, the gain function is returned to the user, permitting the use of other methods of displaying the results.

Subroutine GFARF computes the gain and phase functions of either autoregressive or difference filters. The output consists of a plot of the gain and phase functions versus frequency.

Subroutine GFARFS is the same as GFARF but provides the user with the same options as are available for subroutine GFSLFS.

### C. Subroutine Declaration and CALL Statements

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

#### Subroutines Supporting Symmetric Linear Filter Operations

*LPCOEF: Compute symmetric linear low-pass filter coefficients; return filter coefficients (no printed output)*

```
<real> HLP(k)
:
CALL LPCOEF (FC, K, HLP)
```

===



*HPCOEF: Compute symmetric linear high-pass filter coefficients; return coefficients (no printed output)*

<real> HLP(k), HHP(k)  
:  
CALL HPCOEF (HLP, K, HHP)

===

*MAFLT: Perform simple moving average; return filtered series (no printed output)*

<real> Y(n), YF(n)  
:  
CALL MAFLT (Y, N, K, YF, NYF)

===

*SLFLT: Perform symmetric linear filter operation with user-supplied filter coefficients; return filtered series (no printed output)*

<real> Y(n), H(k), YF(n)  
:  
CALL SLFLT (Y, N, K, H, YF, NYF)

===

*SAMPLE: Sample (extract) every  $NS^{th}$  observation from a series; return sampled series (no printed output)*

<real> Y(n), YS(n)  
:  
CALL SAMPLE (Y, N, NS, YS, NYS)

*LOPASS: Filter series with symmetric linear low-pass filter; return filtered series (no printed output)*

<real> Y(n), HLP(k), YF(n)  
:  
CALL LOPASS (Y, N, FC, K, HLP, YF, NYF)

===

HIPASS: *Filter series with symmetric linear high-pass filter; return filtered series (no printed output)*

```
<real> Y(n), HHP(k), YF(n)
:
CALL HIPASS (Y, N, FC, K, HHP, YF, NYF)
```

Subroutines for Autoregressive or Difference Linear Filters

ARFLT: *Perform autoregressive filter operation with user-supplied filter coefficients; return filtered series (no printed output)*

```
<real> Y(n), PHI(iar), YF(n)
:
CALL ARFLT (Y, N, IAR, PHI, YF, NYF)
```

===

DIF: *Perform first-difference filter operation; return differenced series (no printed output)*

```
<real> Y(n), YF(n)
:
CALL DIF (Y, N, YF, NYF)
```

===

DIFC: *Perform user-specified difference filter operation; return differenced series (no printed output)*

```
INTEGER ND(nfac), IOD(nfac)
<real> Y(n), YF(n), PHI(iar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL DIFC (Y, N, NFAC, ND, IOD, IAR, PHI, LPHI, YF, NYF, LDSTAK)
```

===

DIFM: *Perform first-difference filter operation on series with missing data; return differenced series (no printed output)*

```
<real> Y(n), YF(n)
:
CALL DIFM (Y, YMISS, N, YF, YFMISS, NYF)
```

===

*DIFMC: Perform user-specified difference filter operation on series with missing data; return differenced series (no printed output)*

```
INTEGER ND(nfac), IOD(nfac)
<real> Y(n), YF(n), PHI(iar)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
```

⋮

```
CALL DIFMC (Y, YMISS, N, NFAC, ND, IOD, IAR, PHI, LPHI, YF,
1 YFMISS, NYF, LDSTAK)
```

Subroutines for Computing the Gain and Phase Functions

*GFSLF: Compute and plot gain function of symmetric linear filter*

```
<real> H(k)
```

⋮

```
CALL GFSLF (H, K)
```

===

*GFSLFS: Compute and optionally plot gain function of symmetric linear filter with user-supplied control values; return gain function values and corresponding frequency values*

```
<real> H(k), GAIN(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
```

⋮

```
CALL GFSLFS (H, K, NF, FMIN, FMAX, GAIN, FREQ, NPRT, LDSTAK)
```

===

*GFARF: Compute and plot gain and phase functions of autoregressive or difference filter*

```
<real> PHI(iar)
```

⋮

```
CALL GFARF (PHI, IAR)
```

===

*GFARFS: Compute and optionally plot gain and phase functions of autoregressive or difference filter; with user-supplied control values; return gain and phase function values and corresponding frequency values*

```
<real> PHI(iar), GAIN(nf), PHAS(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL GFARFS (PHI, IAR, NF, FMIN, FMAX, GAIN, PHAS, FREQ, NPRT,
1  LDSTAK)
```

#### 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.

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 are overwritten during subroutine execution.

FC --> The cutoff frequency for the filter, in cycles per sample interval. FC must lie between 0.0 and 0.5.

FMAX --> The maximum frequency, in cycles per sample interval, at which the gain and phase functions are computed ( $0.0 < FMIN < FMAX < 0.5$ ). The default value is 0.5. If FMAX is outside the range FMIN to 0.5 or is not an argument in the CALL statement the default value is used.

FMIN --> The minimum frequency, in cycles per sample interval, at which the gain and phase functions are computed ( $0.0 < FMIN < FMAX < 0.5$ ). The default value is 0.5. If FMIN is outside the range 0.0 to FMAX or is not an argument in the CALL statement the default value is used.

FREQ <-- The vector of dimension at least NF containing the NF frequencies at which the gain and phase functions are computed.

GAIN <-- The vector of dimension at least NF containing the NF gain function values over the range FMIN to FMAX.

-- continued --



For symmetric linear filters:

The gain function of a symmetric linear filter is

$$\text{GAIN}(I) = \left| \sum_{J=1}^K \{H(J) \cos[2\pi |K_m - J| (F_{\text{MIN}} + \Delta_I)]\} \right|$$

for  $I = 1, \dots, \text{NF}$ , where

$K_m$  is the midpoint of the symmetric filter,  $K_m = (K+1)/2$ , and

$\Delta_I$  is the frequency increment, defined as

$$\Delta_I = 2(I-1)(F_{\text{MAX}} - F_{\text{MIN}}) / (\text{NF} - 1)$$

There is no phase change in a symmetric linear filter.

For autoregressive (or difference) filters:

The gain and phase functions of an autoregressive (or difference) filter are

$\text{GAIN}(I) =$

$$\left| 1 - \sum_{J=1}^{\text{IAR}} \{ \text{PHI}(J) (\cos[2\pi J(F_{\text{MIN}} + \Delta_I)] + i \sin[2\pi J(F_{\text{MIN}} + \Delta_I)]) \} \right|$$

and

$$\text{PHAS}(I) = \text{Arctan} \frac{\sum_{J=1}^{\text{IAR}} \{ \text{PHI}(J) \sin[i\pi J(F_{\text{MIN}} + \Delta_I)] \}}{1 - \sum_{J=1}^{\text{IAR}} \{ \text{PHI}(J) \cos[i\pi J(F_{\text{MIN}} + \Delta_I)] \}}$$

for  $I = 1, 2, \dots, \text{NF}$ , where

$i$  is the complex value  $(-1)^{1/2}$ ; and

$\Delta_I$  is the frequency increment, defined as

$$\Delta_I = 2(I-1)(F_{\text{MAX}} - F_{\text{MIN}}) / (\text{NF} - 1).$$

H --> The vector of dimension at least  $K$  containing filter coefficients, which must be symmetric about  $H((K+1)/2)$ .

HHP <-- The vector of dimension at least K containing the K high-pass filter coefficients, which are symmetric about  $HHP((K+1)/2)$ . The high-pass filter coefficients are computed from the low-pass coefficients by

$$HHP(J) = 1 - [HLP(J) / \sum_{L=1}^K HLP(L)] \text{ for } J = K_m,$$

$$HHP(J) = -[HLP(J) / \sum_{L=1}^K HLP(L)] \text{ for } J = 1, \dots, K_m-1, K_m+1, \dots, K.$$

HLP --- The vector of dimension at least K containing the K low-pass filter coefficients, which must be symmetric about  $HLP[(K+1)/2]$ . HLP must be input to HPCOEF; it is returned by LPCOEF and LOPASS.

For LPCOEF and LOPASS, HLP is defined by

$$HLP(J) = h_J / \sum_{I=1}^K h_I \text{ for } J = 1, \dots, K,$$

where

$h_J$  is computed by

$$h_J = 2 \cdot FC \quad \text{for } J = K_m$$

$$h_J = \frac{\sin[2\pi |K_m - J| FC]}{\pi |K_m - J|} \cdot \frac{\sin[2\pi |K_m - J| / K]}{2\pi |K_m - J| / K} \text{ for } J = 1, \dots, K_m-1, K_m+1, \dots, K$$

with  $K_m$  the midpoint of the filter,  $K_m = (K+1)/2$ .

This low-pass filter has a transfer function which changes from approximately one to zero in a transition band about the ideal cutoff frequency  $FC$ , that is from  $(FC - 1/K)$  to  $(FC + 1/K)$ , as shown in figure B.1-1.

IAR --- The number of coefficients in the autoregressive (or difference) filter, including zero coefficients. Equivalently, IAR is the maximum order of the backward shift operator. IAR must be input to ARFLT, GFARF and GFARFS; it is returned by DIFC and DIFMC. IAR is defined by

$$IAR = \sum_{J=1}^{NDF} [IOD(J) \cdot ND(J)].$$

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

IOD --> The vector of dimension at least NFAC containing the NFAC values designating the order of each difference factor.

K --> The number of coefficients in the symmetric linear filter. K must be odd. For LPCOEF, LOPASS, and HIPASS, the user must choose the number of filter terms, K, so that  $(FC - 1/K) > 0$  and  $(FC + 1/K) < 0.5$ . In addition, K must be chosen as a compromise between:

1) A sharp cutoff, that is,  $1/K$  small; and

2) Minimizing the number of data points lost by the filtering operations  $(K - 1)/2$  data points will be lost from each end of the series.

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, §B.]

For DIFC and DIFMC:

$$LDSTAK \geq 7 + 2 \cdot \sum_{J=1}^{NFAC} [ND(J) \cdot IOD(J)] \cdot P$$

For GFSLFS and GFARFS:

$$LDSTAK \geq (11 + IO \cdot (9 + NF)) / 2 + IO \cdot 2 \cdot NF \cdot P$$

where  $IO = 0$  if  $NPRT = 0$ , and  $IO = 1$  if  $NPRT \neq 0$ .

LPHI --> The length of the vector PHI. LPHI must equal or exceed IAR.

N --> The number of observations in the time series. The minimum number of observations is three.

ND --> The vector of dimension at least NFAC containing the NFAC values designating the number of times each difference factor is to be applied.

NF --> The number of frequencies at which the gain and phase functions are to be computed. The default value is 101. If NF is not an argument of the subroutine CALL statement the default value is used.

NFAC --> The number of difference factors.

NPRT --> The argument controlling printed output.

If NPRT < 0, the output consists of a plot of the gain function versus frequency, where the gain function is expressed in decibels and is adjusted so that the peak is at zero. For GFARFS only the output also includes a plot of the phase function versus frequency.

If NPRT = 0, the automatic printout is suppressed.

If NPRT > 0, the output consists of a log-linear plot of the gain function versus frequency. For GFARFS only, the output also includes a plot of the phase function versus frequency.

The default value is -1. If NPRT is not an argument of the subroutine CALL statement the default value is used.

NS --> The sample rate,  $1 \leq NS \leq N$ .

NYF <-- The number of observations in YF.

NYS <-- The number of observations in YS.

PHAS <-- The vector of dimension at least NF containing the NF phase function values over the range FMIN to FMAX.

PHI --- The vector of dimension at least NF containing IAR autoregressive or difference filter coefficients. PHI must be input to ARFLT, GFARF, and GFARFS; it is returned by DIFC and DIFMC.

For DIFC and DIFMC the difference filter coefficients are obtained by expanding the difference operator

$$\prod_{J=1}^{NFAC} (1 - B^{IOD(J)}) ND(J) =$$

$$1 - PHI(1)B^1 - PHI(2)B^2 - \dots - PHI(IAR)B^{IAR}$$

where

B is the backward shift operator, defined by  $B^i y_t = y_{t-i}$ ;

PHI(i) is the  $i^{th}$  difference filter coefficient, which will be a positive or negative integer if the  $i^{th}$  order backward shift operator  $B^i$  is used, and zero if the  $i^{th}$  order backward shift operator is unused.



Y --> The vector of dimension at least N containing the N observations of the time series.

YF <-- The vector of dimension at least N containing the NYF values of the filtered series. The filtered series will start in YF(1); YF(NYF+1) through YF(N) will be set to zero.

For symmetric linear filters:

The filtered series obtained by applying a moving average filter is defined by

$$YF(I) = \sum_{J=1}^K H(J) \cdot Y(I+K-J) \text{ for } I = 1, \dots, NYF,$$

where

NYF is the number of values in the filtered series,  $NYF = N - (K-1)$ , reflecting the  $(K-1)/2$  data points lost from each end of the original series by the filtering operation.

For autoregressive or difference filters:

The filtered series obtained using an autoregressive (or difference) filter is computed by

$$YF(I) = Z(I+IAR) - \sum_{J=1}^{IAR} [\text{PHI}(J) \cdot Z(I+IAR-J)] \text{ for } I = 1, \dots, NYF,$$

where

Z is the N observation time series being filtered which, for ARFLT is the input series Y minus its mean and, for DIF, DIFC, DIFM and DIFMC is the input series Y;

NYF is the number of observations in the filtered series,  $NYF = N - IAR$ , reflecting the IAR data points lost from the beginning of the original series by the filtering operation.

YFMISS <-- The missing value code used within the filtered series YF to indicate that a value could not be computed due to missing data.

YMISS --> The missing value code used within the input series Y to indicate that an observation is missing.

YS <-- The vector of dimension at least N containing the series formed by sampling every  $NS^{\text{th}}$  element of Y,

$$YS(J) = Y((J-1) \cdot NS + 1) \text{ for } J = 1, \dots, NYS,$$

where  $NYS$ , the number of observations in the sampled series, is returned by subroutine `SAMPLE`. The series will start in `YS(1)`; `YS(NYS+1)` through `YS(N)` will be set to zero.

## E. Computational Methods

### E.1 Algorithms

The code for computing the low-pass filter coefficients is based on subroutine `LOPASS`, listed on page 149 of Bloomfield [1976]. The transforms used to compute the gain function of symmetric filters and the gain and phase functions autoregressive (or difference) filters are based on the algorithms shown on pages 311 and 420, respectively, of Jenkins and Watts [1968].

### E.2 Computed Results and Printed Output

Except for the gain and phase function subroutines, `STARPAC` digital filtering subroutines do not produce printed output. For the gain and phase function subroutines, the argument controlling the printed results is `NPRT` and is discussed in §D; the output from the gain and phase function subroutines consists of line printer plots of the gain and phase function of the input filter.

## F. Examples

In the example program of figure F-1a, `DIF` is used to filter the input series `Y`; `VP` (documented in chapter 2) is used to display the log of the original series and the differenced series; and `GFARF` is used to plot the gain and phase functions of the first difference filter. The data used are the natural logarithm of Series G, the airline data, listed on page 531 of Box and Jenkins [1976]. The formulas for the gain and phase functions of a first difference filter and a plot of the corresponding gain function are shown on pages 296 and 9, respectively, of Jenkins and Watts [1968].

Figures F-1b and F-1c display the `VP` plot of the log of the original series and the differenced series, respectively. Both plots have been truncated so that they will fit on one page of this manual; the actual plots generated by `VP` continue for another 90 lines. Figures F-1d and F-1e show the output from `GFARF`. There is no automatic printout from `DIF`.

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE DIF AND GFARF USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPES ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, YF AND PHI MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(200), YF(200), PHI(5)
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C COMPUTE LOG OF DATA
C
C DO 10 I = 1, N
C   Y(I) = ALOG(Y(I))
10 CONTINUE
C
C CALL DIF TO PERFORM DIFFERENCING OPERATION
C
C CALL DIF (Y, N, YF, NYF)
C
C PRINT TITLE AND CALL VP TO DISPLAY LOG OF ORIGINAL SERIES
C
C WRITE (6, 102)
C CALL VP (Y, N, 1)
C
C PRINT TITLE AND CALL VP TO DISPLAY DIFFERENCED SERIES
C
C WRITE (6, 103)
C CALL VP (YF, NYF, 1)
C
C SET PARAMETERS FOR FIRST DIFFERENCE FILTER
C
C PHI(1) = 1.0
C IAR = 1
C
C PRINT TITLE AND CALL GFARF TO COMPUTE GAIN AND PHASE OF
C FIRST DIFFERENCE FILTER
C
C WRITE (6, 104)
C CALL GFARF (PHI, IAR)
C
C STOP
C
C FORMAT STATEMENTS
C
100 FORMAT (4I5)
101 FORMAT (12F6.1)
102 FORMAT ('LOG OF ORIGINAL SERIES DISPLAYED WITH STARPAC PLOT',
1 ' SUBROUTINE VP')
103 FORMAT ('RESULTS OF STARPAC FIRST DIFFERENCE DIGITAL FILTERING',
1 ' SUBROUTINE DIF DISPLAYED WITH STARPAC PLOT SUBROUTINE VP')
104 FORMAT ('RESULTS OF STARPAC',
* ' GAIN AND PHASE FUNCTION SUBROUTINE GFARF')
END
DATA: 144
112.0 118.0 132.0 129.0 121.0 135.0 148.0 148.0 135.0 119.0 104.0 118.0
115.0 126.0 141.0 135.0 125.0 149.0 170.0 170.0 158.0 133.0 114.0 146.0
145.0 150.0 178.0 163.0 172.0 178.0 199.0 199.0 194.0 162.0 146.0 166.0
171.0 180.0 193.0 181.0 183.0 218.0 230.0 242.0 209.0 191.0 172.0 194.0
196.0 196.0 236.0 235.0 229.0 243.0 264.0 272.0 237.0 211.0 180.0 201.0
204.0 188.0 235.0 227.0 234.0 264.0 302.0 293.0 259.0 229.0 203.0 229.0
242.0 233.0 267.0 269.0 270.0 315.0 364.0 347.0 312.0 274.0 237.0 278.0
284.0 277.0 317.0 313.0 318.0 374.0 413.0 405.0 355.0 306.0 271.0 306.0
315.0 301.0 356.0 348.0 355.0 422.0 465.0 467.0 404.0 347.0 305.0 336.0
340.0 318.0 362.0 348.0 363.0 435.0 491.0 505.0 404.0 359.0 310.0 337.0
360.0 342.0 406.0 396.0 420.0 472.0 548.0 559.0 463.0 407.0 362.0 405.0
417.0 391.0 419.0 461.0 472.0 535.0 622.0 606.0 508.0 461.0 390.0 432.0

```

Figure F-la

Example program using DIF, VP and GFARF

LOG OF ORIGINAL SERIES DISPLAYED WITH STARPAC PLOT SUBROUTINE VP

STARPAC 2.00S - DECEMBER 2, 1985

6.4329 6.2541 6.0752 5.8964 5.7175 5.5387 5.3596 5.1810 5.0023 4.8232 4.6444

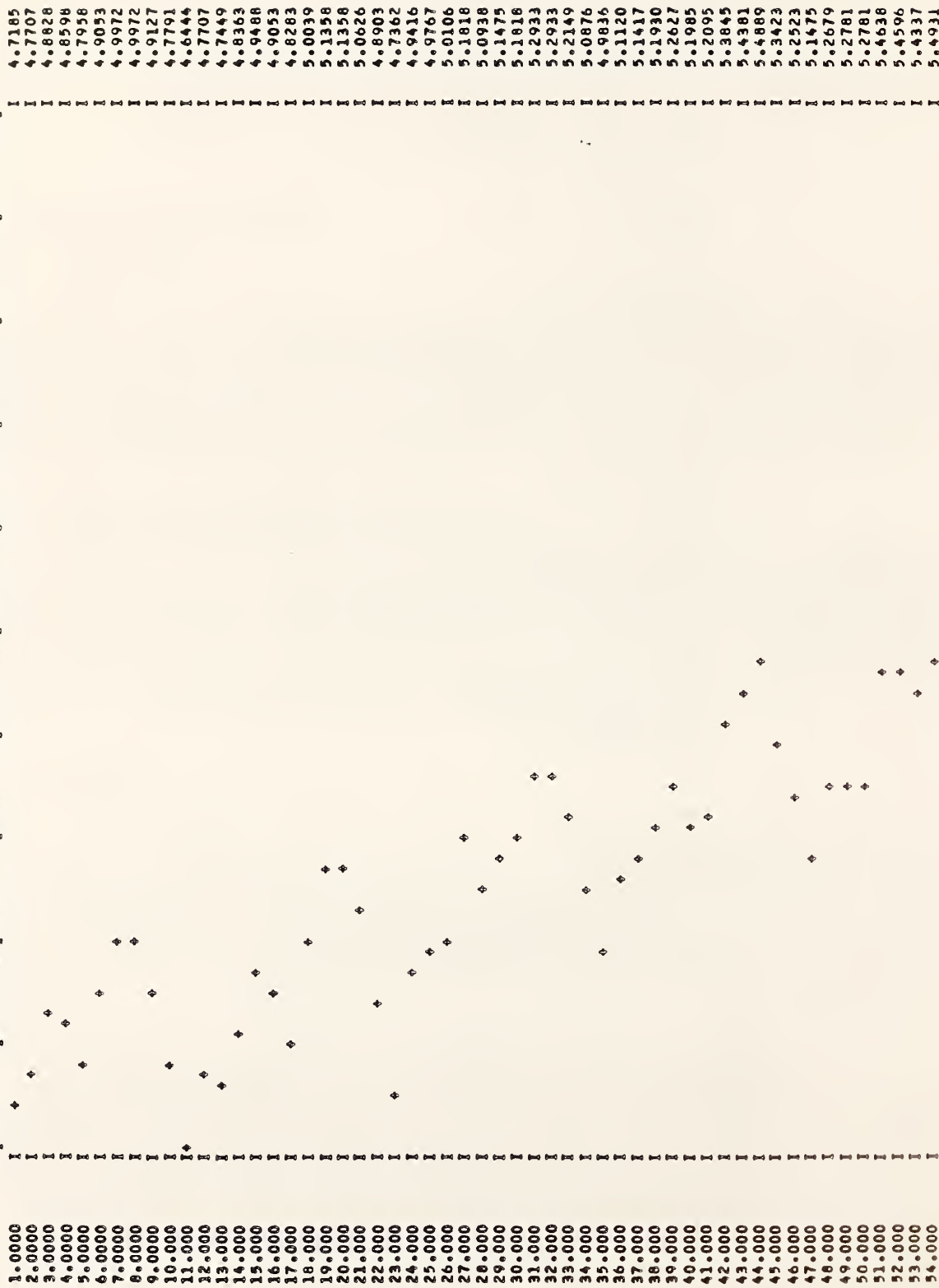


Figure F-1b

Example of partial VP output displaying the log of the original series



RESULTS OF STARPAC FIRST DIFFERENCE DIGITAL FILTERING SUBROUTINE DIF DISPLAYED WITH STARPAC PLOT SUBROUTINE VP  
 STARPAC 2.00S - DECEMBER 2, 1985  
 -223E+00 -.179E+00 -.134E+00 -.893E-01 -.446E-01 .888E-15 .446E-01 .893E-01 .134E+00 .179E+00 .223E+00



Figure F-1c

Example of partial VP output displaying the differenced series



Figure F-1d

Example of GFARF output

PHASE FUNCTION OF 1 TERM AUTOREGRESSIVE, OR DIFFERENCE, FILTER

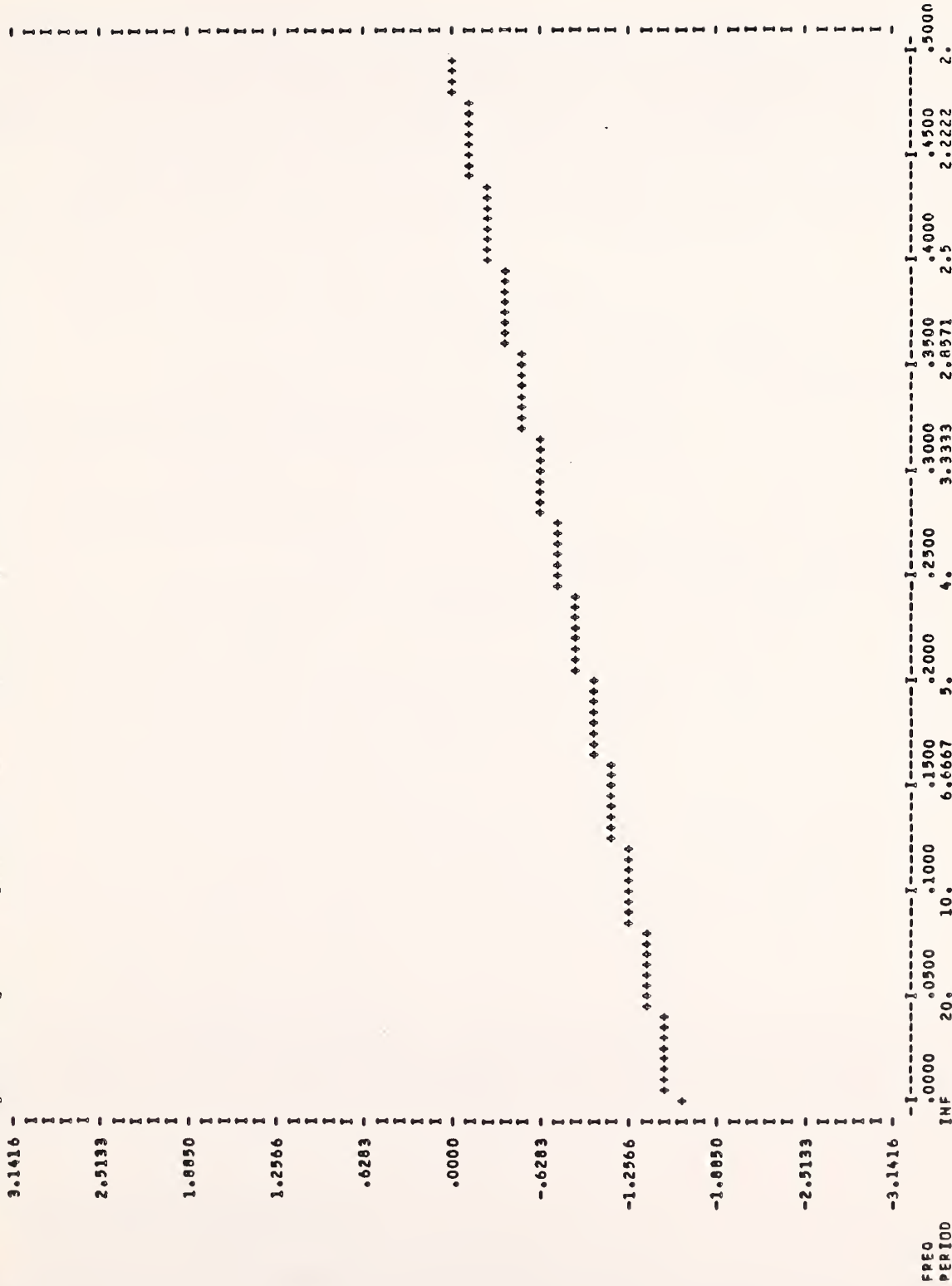


Figure F-1e

Example of GFARF output (continued)

## G. Acknowledgments

The code for computing the low-pass filter coefficients is based on subroutine LOPASS, listed on page 149 of Bloomfield [1976]. The transforms used to compute the gain function of symmetric filters and the gain and phase functions of autoregressive (or difference) filters are based on the algorithms shown on pages 311 and 420, respectively, of Jenkins and Watts [1968].



## CHAPTER 11

### COMPLEX DEMODULATION

#### A. Introduction

STARPAC contains two subroutines which find the amplitude and phase functions of a demodulated series as described in Bloomfield [1976].

The demodulated series  $\delta(t)$  is formed by multiplying the observed series by a complex sinusoid at the demodulation frequency. If the observed series  $Y$  is a sinusoid of the nominal demodulation frequency  $FD$  with varying amplitude and phase plus noise, that is,

$$\begin{aligned} Y(t) &= R(t) \cos(2\pi FDt + \phi(t)) + a(t) \\ &= 0.5 R(t) \left[ e^{i(2\pi FDt + \phi(t))} + e^{-i(2\pi FDt + \phi(t))} \right] + a(t) \end{aligned}$$

then the demodulated series may be represented by

$$\begin{aligned} \delta(t) &= e^{-i2\pi FD} Y(t) \\ &= 0.5 R(t) e^{i\phi(t)} + (0.5)R(t) e^{-i(4\pi FDt + \phi(t))} + e^{-i2\pi FD} a(t) \end{aligned}$$

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

$N$  is the number of time points in the series;

$i$  is the complex value  $(-1)^{1/2}$ ;

$FD$  is the demodulation frequency in cycles per sample interval;

$R(t)$  is the amplitude component of the observed series at time  $t$ ;

$\phi(t)$  is the phase component of the observed series at time  $t$ ;

$a(t)$  is the noise at time  $t$ .

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

#### B. Subroutine Descriptions

Subroutine DEMOD computes the smoothed amplitude and phase components of the demodulated series. The user must specify the demodulation frequency, along with the number of filter terms and the cutoff frequency which define

the low-pass filter utilized to smooth the demodulated series. Output from DEMOD consists of plots of the amplitude and phase functions. The phase function plot reduces discontinuities using the method suggested by Bloomfield [1976]. As shown in figure F-1c this method displays both the principle phase value, which is defined to lie in the range  $-\pi$  to  $\pi$ , and the principle phase value plus or minus  $2\pi$ , where the sign is chosen such that the second value lies in the range  $-2\pi$  to  $2\pi$ .

Subroutine DEMODS is the same as DEMOD except that the computed amplitude and phase functions are returned to the user and the printed output described for DEMOD is optional.

### C. Subroutine Declaration and CALL Statements

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

DEMOD: *Compute and plot the results of a complex demodulation of the input series*

```
<real> Y(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL DEMOD (Y, N, FD, FC, K, LDSTAK)
```

===

DEMOS: *Compute and optionally plot the results of a complex demodulation of the input series; return amplitude and phase functions of demodulated series*

```
<real> Y(n), AMPL(n), PHAS(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL DEMOS (Y, N, FD, FC, K, AMPL, PHAS, NDEM, NPRT, LDSTAK)
```

### 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.

AMPL <-- The vector of dimension at least  $N-(K-1)$  that contains the NDEM values of the smoothed amplitude function of the observed series,  $AMPL(I) = R(t)$ , where  $R(t)$  is defined in §A and the index  $I$  is computed as  $I = t - (K-1)/2$  for  $t = (K+1)/2$  to  $N-(K-1)/2$ . The stored values of the amplitude function will start in  $AMPL(1)$ ;  $AMPL(NDEM+1)$  to  $AMPL(N)$  will be set to zero.

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.

FC --> The cutoff frequency for the low-pass filter in cycles per sample interval. FC must lie between  $1/K$  and FD.

FD --> The demodulation frequency in cycles per sample interval. FD must lie between 0.0 and 0.5.

IERR \*\*\* An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

K --> The number of terms in the low-pass filter used to extract the amplitude and phase functions. K must be odd. The user must choose the number of filter terms, K, so that  $(FC - 1/K) > 0$  and  $(FC + 1/K) < 0.5$ . In addition, K must be chosen as a compromise between:

1) A sharp cutoff, that is,  $1/K$  small; and

2) Minimizing the number of data points lost by the filtering operations. ( $(K - 1)/2$  data points will be lost from each end of the series.)

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, §B.]

For DEMOD:

$$LDSTAK > 10 + (3 \cdot N + K) \cdot P$$

For DEMODS:

$$LDSTAK > 9 + (10 \cdot 2 \cdot N + K) \cdot P$$

where  $IO = 0$  if  $NPRT = 0$  and  $IO = 1$  if  $NPRT \neq 0$ .

- N --> The number of observations, which must equal or exceed 17.
- NDEM <-- The number of observations in AMPL and PHAS, NDEM = N - (K-1).
- NPRT --> The variable controlling printed output.
- If NPRT = 0, the automatic printout is suppressed.
- If NPRT ≠ 0, the automatic printout is provided.
- The default value is 1. If NPRT is not an argument of the subroutine CALL statement the default value is used.
- PHAS <-- The vector of dimension at least NDEM = N-(K-1) that contains the NDEM primary values of the smoothed phase function of the observed series, PHAS(I) = φ(t), where φ(t) is defined in \$A and the index I is computed as I = t - (K-1)/2 for t = (K+1)/2 to N - (K-1)/2. The stored values of the phase function will start in PHAS(1); PHAS(NDEM+1) to PHAS(N) will be set to zero.
- Y --> The vector of dimension at least N that contains the observations of the time series.

## E. Computational Methods

### E.1 Algorithms

The STARPAC code for performing complex demodulation was adapted from the subroutines given on pages 147 to 150 of Bloomfield [1976]. As noted in Bloomfield, the first term of the demodulated series defined in \$A is centered about zero frequency while the remaining two terms are centered at frequencies FD and 2FD. Thus, the first term can be separated from the others using the low-pass filter described in chapter 10 (with FC ≈ FD/2), resulting in the complex filtered series

$$\begin{aligned}
 YF(t) &= \sum_{J=1}^K \text{HLP}(J) \cdot \delta(t + K_m - J) \\
 &= \alpha(t) + i \cdot \beta(t) \\
 &\cong 0.5 \cdot R(t) \cdot e^{i\phi(t)} \quad \text{for } t = K_m, K_m+1, \dots, (N-K_m+1),
 \end{aligned}$$

where

K is the number of filter terms;

K<sub>m</sub> is the midpoint of the filter, K<sub>m</sub> = (K+1)/2;

HLP(J) is the J<sup>th</sup> low-pass filter coefficient, defined in chapter 10, \$D;



$\alpha(t)$  is the real part of the filtered series;

$\beta(t)$  is the imaginary part of the filtered series.

The smoothed estimates of the amplitude  $R$  and phase  $\phi$  functions can then be extracted from the filtered series using

$$\hat{R}(t) = 2(\alpha(t)^2 + \beta(t)^2)^{1/2}$$

and

$$\hat{\phi}(t) = \tan^{-1}(\alpha(t)/\beta(t)).$$

Note that  $(K-1)/2$  points have been lost from each end of the demodulated series by the filtering operation.

## E.2 Computed Results and Printed Output

The argument controlling the printed output, NPRT, is discussed in §D. The output consists of plots of the smoothed amplitude and phase functions, and a list of the demodulation frequency, cutoff frequency and number of terms in the low-pass filter used to smooth the demodulated series.

## F. Example

In the example program of figure F-1a, DEMOD is used to estimate the amplitude and phase function corresponding to the input series  $Y$ . The data used are the Wolf sunspot numbers for the years 1700 to 1960 as tabulated by Waldmeier [1961]. Figure F-1b shows the plot of the amplitude function and figure F-1c shows the plot of the phase function. Both plots have been truncated so that they will fit on one page of this manual; the actual plots generated by DEMOD continue for another 177 lines. The transfer function of the low-pass filter used in this example is shown in chapter 10, figure B.1-1. Further discussion of this example can be found on pages 137 to 141 of Bloomfield [1976].

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE DEMOD USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(300)
C DOUBLE PRECISION DSTAK(500)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 500
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C SET DEMODULATION FREQUENCY
C CUTOFF FREQUENCY
C NUMBER OF TERMS IN THE LOW-PASS FILTER
C
C FD = 1.0 / 11.0
C FC = 1.0 / 22.0
C K = 41
C
C PRINT TITLE AND CALL DEMOD FOR COMPLEX DEMODULATION ANALYSIS
C
C WRITE (6, 102)
C CALL DEMOD (Y, N, FD, FC, K, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
100 FORMAT (I5)
101 FORMAT (10F7.2)
102 FORMAT ('RESULTS OF STARPAC',
* ' COMPLEX DEMODULATION SUBROUTINE DEMOD')
END

DATA: 261
5.00 11.00 16.00 23.00 36.00 58.00 29.00 20.00 10.00 8.00
3.00 0.00 0.00 2.00 11.00 27.00 47.00 63.00 60.00 39.00
28.00 26.00 22.00 11.00 21.00 40.00 78.00 122.00 103.00 73.00
47.00 35.00 11.00 5.00 16.00 34.00 70.00 81.00 111.00 101.00
73.00 40.00 20.00 16.00 5.00 11.00 22.00 40.00 60.00 80.90
83.40 47.70 47.80 30.70 12.20 9.60 10.20 32.40 47.60 54.00
62.90 85.90 61.20 45.10 36.40 20.90 11.40 37.80 69.80 106.10
100.80 81.60 66.50 34.80 30.60 7.00 19.80 92.50 154.40 125.90
84.80 68.10 38.50 22.80 10.20 24.10 82.90 132.00 130.90 118.10
89.90 66.60 60.00 46.90 41.00 21.30 16.00 6.40 4.10 6.80
14.50 34.00 45.00 43.10 47.50 42.20 28.10 10.10 8.10 2.50
0.00 1.40 5.00 12.20 13.90 35.40 45.80 41.10 30.10 23.90
15.60 6.60 4.00 1.80 8.50 16.60 36.30 49.60 64.20 67.00
70.90 47.80 27.50 8.50 13.20 56.90 121.50 138.30 103.20 85.70
64.60 36.70 24.20 10.70 15.00 40.10 61.50 98.50 124.70 96.30
66.60 64.50 54.10 39.00 20.60 6.70 4.30 22.70 54.80 93.80
95.80 77.20 59.10 44.00 47.00 30.50 16.30 7.30 37.60 74.00
139.00 111.20 101.60 66.20 44.70 17.00 11.30 12.40 3.40 6.00
32.30 54.30 59.70 63.70 63.50 52.20 25.40 13.10 6.80 6.30
7.10 35.60 73.00 85.10 78.00 64.00 41.80 26.20 26.70 12.10
9.50 2.70 5.00 24.40 42.00 63.50 53.80 62.00 48.50 43.90
18.60 5.70 3.60 1.40 9.60 47.40 57.10 103.90 80.60 63.60
37.60 26.10 14.20 5.80 16.70 44.30 63.90 69.00 77.80 64.90
35.70 21.20 11.10 5.70 8.70 36.10 79.70 114.40 109.60 88.80
67.80 47.50 30.60 16.30 9.60 33.20 92.60 151.60 136.30 134.70
83.90 69.40 31.50 13.90 4.40 38.00 141.70 190.20 184.80 159.00
112.30

```

Figure F-1a

Example program using DEMOD

TIME SERIES DEMODULATION

DEMULATION FREQUENCY IS .09090909  
CUTOFF FREQUENCY IS .04545455  
THE NUMBER OF TERMS IN THE FILTER IS 41

PLOT OF AMPLITUDE OF SMOOTHED DEMODULATED SERIES

LOCATION OF MEAN IS GIVEN BY PLOT CHARACTER M

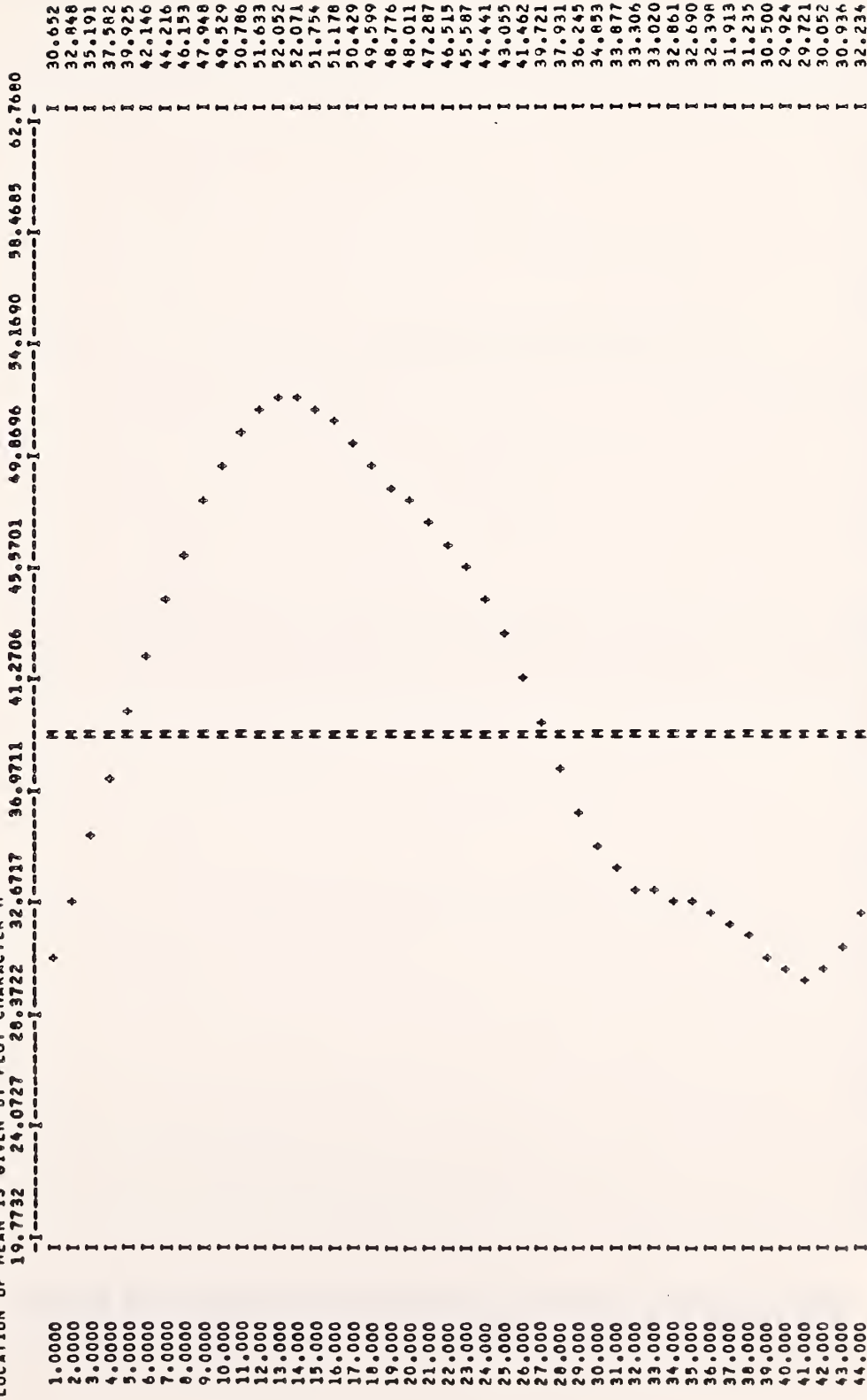


Figure F-1b

Example of DEMOD output

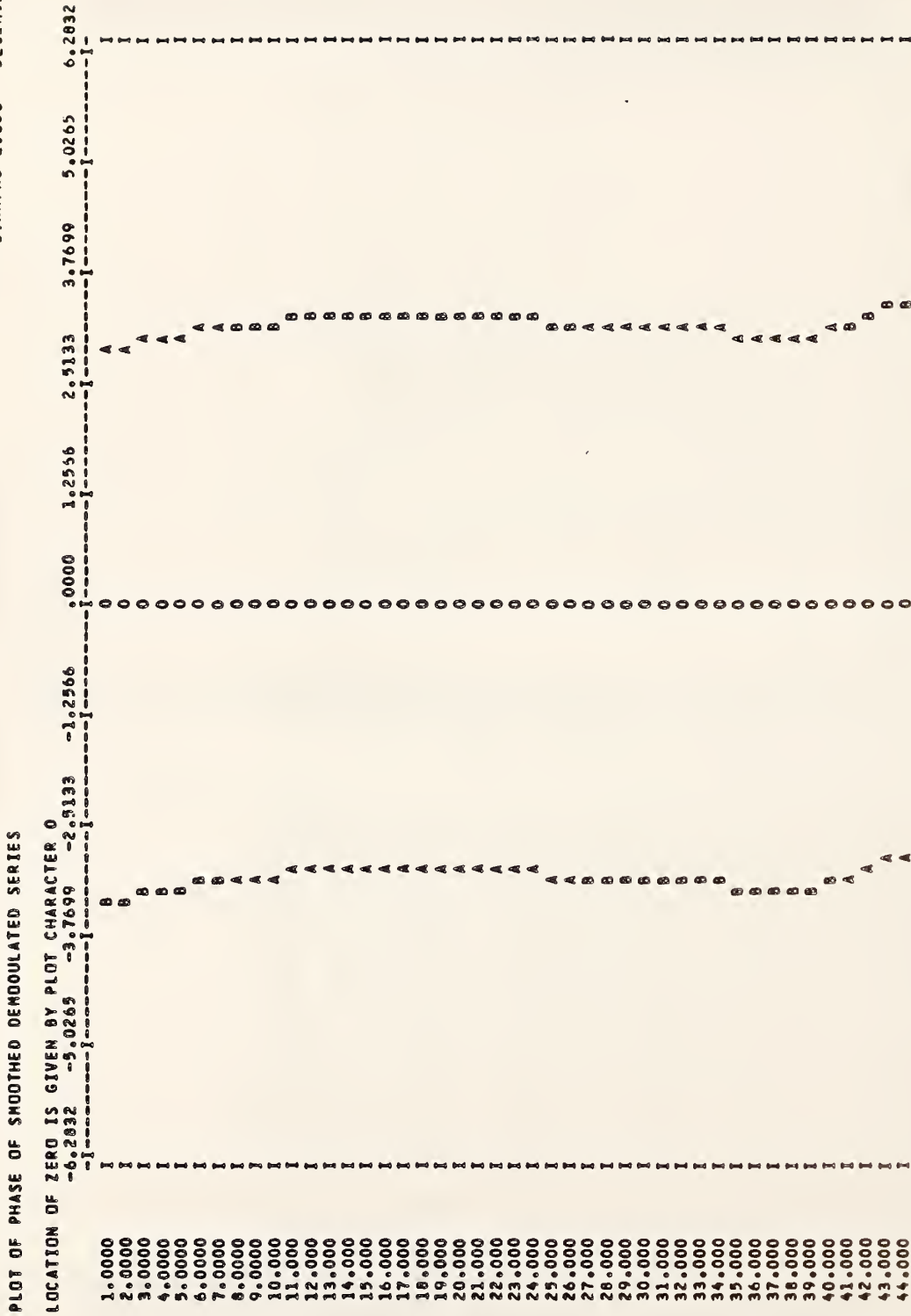
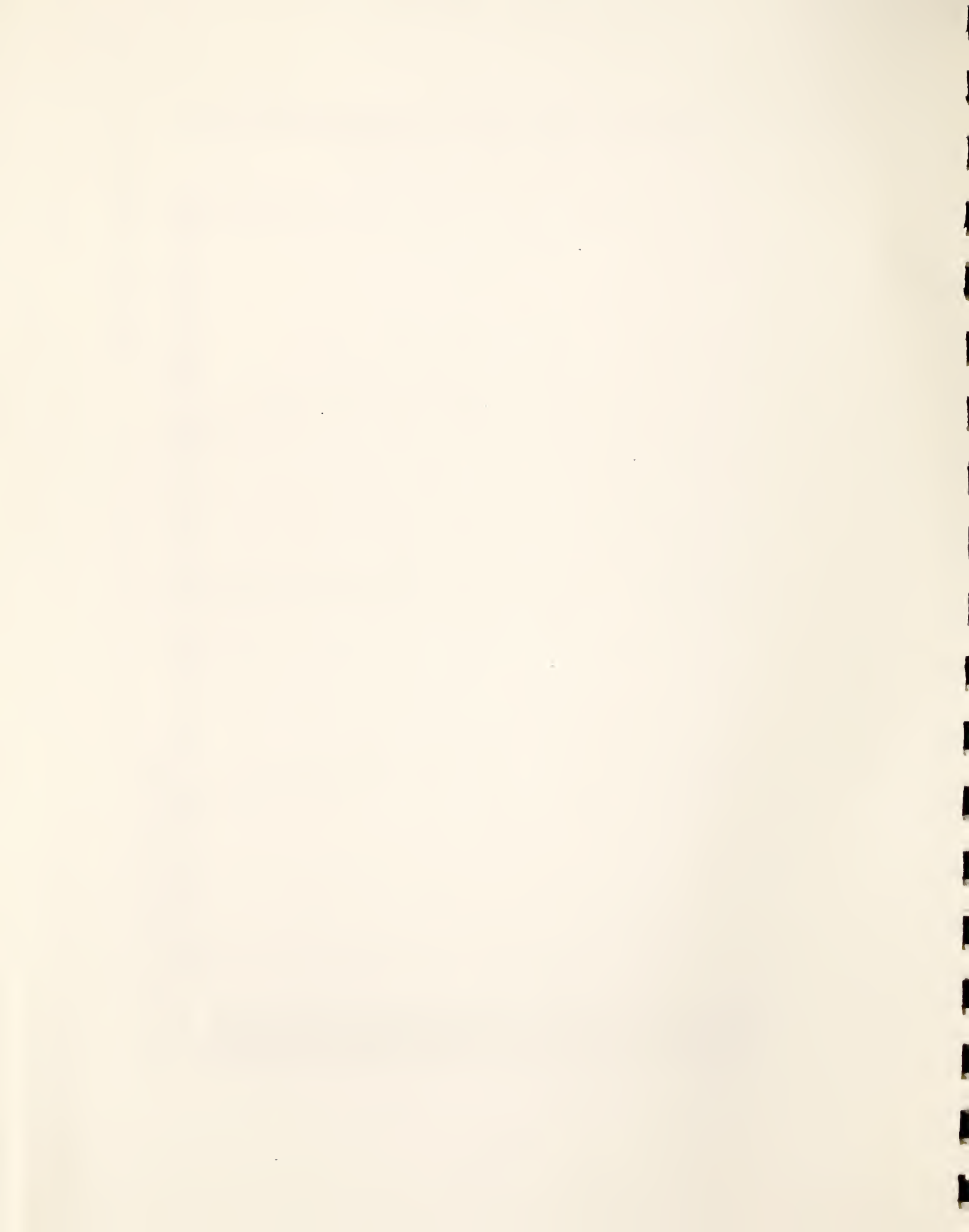


Figure F-1c  
Example of DEMOD output (continued)



G. Acknowledgments

The code for performing the complex demodulation was adapted from the subroutines given on pages 147 to 150 of Bloomfield [1976].



## CHAPTER 12

### CORRELATION AND SPECTRUM ANALYSIS

#### A. Introduction

STARPAC contains 50 subroutines for time series correlation and spectrum estimation. Both univariate and bivariate series can be analyzed. Included are subroutines that compute the correlation function using the fast Fourier transform and that accept time series with missing observations. The user may choose from spectrum analysis subroutines implementing the classical Fourier transformed covariance function techniques presented in Jenkins and Watts [1968], the autoregressive or rational spectrum techniques described by Jones [1971] or the direct Fourier transform (periodogram) techniques discussed in Bloomfield [1976].

Users are directed to \$B for a brief description of the subroutines. The declaration and CALL statements are given in \$C and the subroutine arguments are defined in \$D. The algorithms used and output produced by these subroutines are discussed in \$E. Sample programs and their output are shown in \$F.

#### B. Subroutine Descriptions

STARPAC correlation and spectrum analysis subroutines are divided into seven families. For correlation analysis of univariate and bivariate series there are two families of subroutines supporting

1. Autocorrelation Analysis and
2. Cross Correlation Analysis.

For spectrum estimation there are four families of subroutines for univariate series and one family for bivariate series supporting

3. Univariate Spectrum Estimation Using the Fourier Transform of the Autocorrelation Function,
4. Univariate Spectrum Estimation Using Autoregressive Models,
5. Univariate Spectrum Estimation Using the Direct Fourier Transform,
6. Univariate Series Utilities and
7. Bivariate Spectrum Estimation Using the Fourier Transform of the Cross Correlation Function.

In general, each family of subroutines has one basic subroutine which performs the desired computations with a minimum of user input. The other subroutines in each family provide greater flexibility to the user at the price of more input. The features of these subroutines are indicated by the suffix letter(s) on the subroutine (e.g., ACFM and BFSFS). Not all features are available for each family. Features which are common to more than one family are described here. Features which are unique to a specific family are described in the subsections below.

- Suffix S indicates that the user is allowed to specify various options which are preset in the simplest call and that certain results are

returned to the user via the subroutine CALL statement. In the subsections that follow, the specific details of this feature are discussed individually for each family of subroutines.

- Suffix M indicates that the series contains missing data. A missing value code must be used within the series to specify time points without an observed value. There is no limit on the percentage of data that can be missing. However, because the correlation matrix computed from a series with missing values is not necessarily positive definite, the partial autocorrelation function estimates and autoregressive order selection statistics are not computed and caution must be used in interpreting those results which are provided. Analysis of time series with missing values is discussed in Jones [1971].
- Suffix F indicates that the covariances are computed using the Singleton [1969] fast Fourier transform (FFT). When the number of observations in the series is large this method of computation is more efficient than the direct computation normally used by STARPAC. Subroutines with an F suffix reduce the amount of workspace needed by using the vector originally containing the data as workspace; the data must be copied into another vector prior to calling these subroutines if the data are to be preserved. These subroutines automatically extend the length of the input series by appending enough zeros to meet the requirements of this FFT code; the length of the vector used to pass the data to these subroutines must therefore equal or exceed the extended series length, NFFT, as discussed in §D.
- Suffix V indicates that the user inputs the covariances rather than the original series, thus avoiding a time-consuming recomputation of the covariance function if it is already available, for example, from subroutines ACFS, ACFFS, ACFMS, CCFS, CCFFS or CCFMS.

## B.1 Correlation Analysis

### B.1.a Univariate Series

Autocorrelation Analysis. STARPAC's autocorrelation function (acf) subroutines compute and plot the autocorrelation function estimates; compute the large lag standard error of the estimates; perform a chi-squared test of the null hypothesis that the series is white noise; compute the partial autocorrelation function coefficients estimates; and, using the modified Akaike information criterion [Akaike, 1974], select the order of an autoregressive process which models the series and estimate the parameters of this autoregressive model. The user should note that a purely autoregressive model may approximate the true structure of the model with an unnecessarily large number of terms. Such an autoregressive model must be used with discretion since the true structure might actually be more complex including moving average components, harmonic terms or some mixture of deterministic and stochastic elements. For some purposes, a purely autoregressive approximation may be useful. In other cases, careful model identification can lead to the discovery of more detailed structure of the data or to a more parsimonious model.



The simplest of the AutoCorrelation Function subroutines is ACF, which performs the basic analysis described in the preceding paragraph. The other autocorrelation analysis subroutines provide the same basic analysis as ACF while adding the features indicated above by suffixes S, M, F, MS and FS.

For the ACF family of subroutines, the suffix S feature allows the user to indicate:

- 1) the maximum lag value for which the correlation function is to be computed; and
- 2) the amount of printed output.

The acf subroutines with suffix S also return the autocovariance function estimates and the coefficients of the selected autoregressive model to the user via the subroutine CALL statement.

The ACF family of subroutines also includes subroutine ACFD, where the suffix D indicates that the autocorrelation analysis will be performed for a sequence of differenced series. The difference factors are provided by the user. If the number of difference factors, NFAC, is greater than one, difference factors beyond the first are applied to the input series Y(t) to yield a series Z(t) given by

$$Z(t) = \left\{ \prod_{J=2}^{NFAC} (1 - B^{IOD(J)})^{ND(J)} \right\} Y(t)$$

where the  $B^k$  indicates the backward shift operator defined by

$$B^k Y(t) = Y(t-k)$$

and IOD and ND are defined in \$D. If the number of difference factors is equal to one,  $Z(t) = Y(t)$ . In either case, the autocorrelation analysis is performed first on the series Z and then on series Z with the difference factor  $(1 - B^{IOD(1)})$  applied 1 to ND(1) times. This produces ND(1) + 1 passes of the basic ACF analysis.

### B.1.b Bivariate Series

Cross Correlation Analysis. STARPAC's cross correlation analysis subroutines compute and plot the cross correlation function coefficients and provide the large lag standard error of these estimates. Subroutine CCF is the simplest of the Cross Correlation Function subroutines. The other five cross correlation analysis subroutines provide the same basic analysis as CCF while adding the features indicated above by suffixes S, M, F, MS and FS.

For the CCF family of subroutines, suffix S indicates that the analysis is provided for each pair of series of a multivariate time series. The suffix S feature also allows the user to indicate:

- 1) the maximum lag value for which the correlation function is to be computed; and
- 2) the amount of printed output.

In addition, the cross covariance function estimates are returned to the user.

## B.2 Spectrum Estimation

### B.2.a Univariate Series

Univariate Spectrum Estimation Using the Fourier Transform of the Autocorrelation Function. The UFS (Univariate Fourier Spectrum) subroutine family computes the estimated spectrum from the Fourier transform of the autocovariance function (acvf) as discussed in Jenkins and Watts [1968]. The spectrum is smoothed using Parzen windows with the bandwidth controlled either by the user or a window-closing algorithm. The principal output from each of these subroutines consists of plots of the estimated spectrum.

Subroutine UFS has the simplest CALL statement of this family of subroutines. The printed output consists of four spectrum plots with successively narrower bandwidths. Each spectrum is displayed in decibels (10 times the base 10 logarithm of the power spectrum) scaled so that the maximum value plotted is zero. The length of the upper and lower 95-percent confidence intervals and the bandwidth for each spectrum are shown on the plots.

The other nine univariate Fourier spectrum estimation subroutines provide the same basic analysis as UFS while adding the features indicated above by suffixes S, M, F, V, FS, MS, VS, MV and MVS.

For the UFS family of subroutines, the suffix S feature allows the user to indicate:

- 1) the number of different window bandwidths to be used and the lag window truncation point for each;
- 2) the frequency range and the number of frequencies for which the spectrum is to be computed; and
- 3) whether the plot is to be in decibels or on a logarithmic scale.

In addition, the spectrum values are returned to the user.

Univariate Spectrum Estimation Using Autoregressive Models. STARPAC Univariate Autoregressive Spectrum estimation subroutines (UAS family) approximate an input series with an autoregressive model and compute the corresponding theoretical spectrum for that model. For comparative purposes, the plot of the autoregressive spectrum is superimposed against a Fourier spectrum plot.

Subroutine UAS is the simplest of the autoregressive spectrum estimation subroutines. It uses the modified Akaike information criterion [Akaike, 1974] to select the order of the autoregressive model to be used. The autoregressive coefficients are then computed from the autocovariance function using the Levinson-Durbin recursive method [Box and Jenkins, 1976] for solving the

Yule-Walker equations. The lag window truncation point used for the Fourier spectrum is half the maximum truncation point which would have been selected by subroutine UFS. [See \$D, argument LAGS.] The output consists of several autoregressive order selection statistics and a plot of the autoregressive and Fourier spectra in decibels (10 times the base 10 logarithm of the power spectrum) scaled such that the maximum value plotted is zero. The bandwidth and the length of the 95-percent confidence interval for the Fourier spectrum are shown on the plot. This Fourier spectrum and its confidence intervals should be used in interpreting the autoregressive spectrum since confidence intervals are not computed by STARPAC for the autoregressive spectrum. (The bandwidth is not relevant to the autoregressive spectrum.)

The other five autoregressive spectrum subroutines provide the same basic analysis as subroutine UAS while adding the features indicated above by suffixes S, F, V, FS and VS.

For the autoregressive spectrum family of subroutines, the suffix S feature allows the user to indicate:

- 1) the order of the autoregressive model to be used for the autoregressive spectrum;
- 2) the lag window truncation point to be used for the Fourier spectrum;
- 3) the frequency range and the number of frequencies within this range at which the spectrum is to be computed; and
- 4) whether the plot is to be in decibels or on a logarithmic scale.

In addition, the autoregressive and Fourier spectra are returned to the user. The user should be cautious about using high order models without checking order selection statistics since such models can produce spurious peaks in the spectrum.

Univariate Spectrum Estimation Using the Direct Fourier Transform. The STARPAC direct Fourier transform subroutines (PGM family) implement the PeriodoGram approach to time series analysis discussed in Bloomfield [1976]. Subroutines are included for computing the raw periodogram and for computing and plotting the integrated periodogram (or cumulative spectrum).

Subroutine PGM computes the periodogram of the series as described for argument PER in \$D using zeros to extend the length of the input series. Output consists of a plot of the computed periodogram in decibels (10 times the base 10 logarithm of the periodogram estimates) scaled so that the maximum value plotted is zero. The input series must be either centered or tapered by the user before PGM is called.

PGMS provides the same basic analysis as PGM but allows the user to indicate:

- 1) whether zeros or the series mean is used to extend the series;
- 2) the length of the extended series; and
- 3) the amount of printed output.

In addition, the periodogram values and their frequencies are returned to the user via the subroutine CALL statement.



The integrated periodogram subroutine IPGM first subtracts the series mean from the series, then extends the input series with zeros and finally computes the normalized integrated periodogram. Output consists of a one-page plot of the integrated periodogram, accompanied by 95-percent contours for testing the null hypothesis of white noise. The integrated periodogram is discussed in chapter 8 of Box and Jenkins [1976].

The other three integrated periodogram subroutines add the features indicated by suffixes S, P and PS. The suffix S option allows the user to control the amount of printed output; the integrated periodogram values and their corresponding frequencies are also returned to the user via the subroutine CALL statement. The suffix P option indicates that the user inputs the periodogram rather than the original series, thus avoiding a time-consuming recomputation of the periodogram if it is already available, for example, from subroutine PGMS.

Utilities. STARPAC includes utility subroutines for centering (subtracting the mean) and tapering the observed series, periodogram smoothing and for computing the Fourier coefficients of the series. These routines are particularly useful when using direct Fourier techniques such as PGM and IPGM.

Subroutine CENTER subtracts the series mean from the series and returns the centered series. There is no printed output.

Subroutine TAPER centers the input series and applies the split-cosine-bell taper described for argument YT in \$D. The user specifies the total proportion of the series to be tapered. The centered tapered series is returned to the user and can be used as input to subroutine PGM or PGMS. There is no printed output.

Subroutine FFTLEN computes for an observed series length, N, the minimum extended series length, NFFT, which will meet the requirements of the Singleton FFT code. The value of the extended series length is returned to the user. There is no printed output.

Subroutine MDFLT smooths the input periodogram by applying a sequence of modified Daniell filters as discussed in chapter 7 of Bloomfield [1976]. The filtered series is returned to the user. There is no printed output. Subroutine MDFLT takes advantage of the symmetry of the periodogram to avoid losing values from the ends of the series. It should therefore not be used for input series that are not symmetric about their end values. Other digital filtering routines, such as those described in chapter 10, may also be used for periodogram smoothing but end effect losses will be incurred.

Subroutine FFTR computes the Fourier coefficients of an input series of single precision observations. There is no printed output.



## B.2.b Bivariate Series

Bivariate Spectrum Estimation Using the Fourier Transform of the Cross Correlation Function. The BFS (Bivariate Fourier Spectrum) subroutine family computes the estimated spectrum from the Fourier transform of the covariance function as discussed in Jenkins and Watts [1968]. The spectrum is smoothed using Parzen windows with the bandwidth controlled either by the user or a window-closing algorithm. The principal output from each of these subroutines consists of plots of the squared coherency and phase components of the cross spectrum.

Subroutine BFS provides the basic analysis with a brief CALL statement. The printed output consists of four spectrum plot pairs (a squared coherency plot and a phase plot) with successively narrower bandwidths chosen by the window-closing algorithm. The upper and lower 95-percent confidence intervals and the 95-percent significance levels are shown on the coherency plots.

The other nine bivariate Fourier spectrum estimation subroutines provide the basic analysis described for BFS while adding the features indicated above by suffixes S, M, F, V, FS, MS, VS, MV and MVS.

For the BFS family of subroutines, the suffix S feature allows the user to indicate:

- 1) the number of different window bandwidths to be used and the lag window truncation point for each;
- 2) the frequency range and the number of frequencies for which the spectrum is to be computed; and
- 3) whether the plot is to be in decibels or on a logarithmic scale.

In addition, the squared coherency and phase values are returned to the user.

## C. Subroutine Declaration and CALL Statements

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

### Subroutines for Autocorrelation Analysis

ACF: *Compute and print a two-part autocorrelation analysis of a series*

```
<real> Y(n)
:
CALL ACF (Y, N)
```

===

*ACFS: Compute and optionally print a two-part autocorrelation analysis of a series using user-supplied control values; return autocovariance function and order and parameter estimates of selected autoregressive model*

```
<real> Y(n), ACOV(lagmax+1), PHI(lagmax)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL ACFS (Y, N, LAGMAX, LACOV, ACOV, IAR, PHI, NPRT, LDSTAK)
```

===

*ACFM: Compute and print a two-part autocorrelation analysis of a series with missing observations*

```
<real> Y(n)
:
CALL ACFM (Y, YMISS, N)
```

===

*ACFMS: Compute and optionally print a two-part autocorrelation analysis of a series with missing observations using user-supplied control values; return autocovariance function*

```
INTEGER NLPPA(lagmax+1)
<real> Y(n), ACOV(lagmax+1)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL ACFMS (Y, YMISS, N, LAGMAX, LACOV, ACOV, AMISS, NLPPA, NPRT,
1 LDSTAK)
```

===

*ACFF: Compute and print a two-part autocorrelation analysis of a series; use FFT for computations; return autocovariance function and order and parameter estimates of selected autoregressive model*

```
<real> YFFT(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL ACFF (YFFT, N, LYFFT, LDSTAK)
```

===

*ACFFS: Compute and optionally print a two-part autocorrelation analysis of a series using user-supplied control values; use FFT for computations; return autocovariance function and order and parameter estimates of selected autoregressive model*

```
<real> YFFT(nfft), ACOV(lagmax+1), PHI(lagmax)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL ACFFS (YFFT, N, LYFFT, LDSTAK, LAGMAX, LACOV, ACOV, IAR, PHI,
1 NPRT)
```

===

*ACFD: Compute and print a two-part autocorrelation analysis of a differenced series*

```
INTEGER ND(nfac), IOD(nfac)
<real> Y(n)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL ACFD (Y, N, LAGMAX, NFAC, ND, IOD, LDSTAK)
```

#### Subroutines for Cross Correlation Analysis

*CCF: Compute and print a two-part cross correlation analysis of a pair of series*

```
<real> Y1(n), Y2(n)
:
CALL CCF (Y1, Y2, N)
```

===

*CCFS: Compute and optionally print a two-part cross correlation analysis of a multivariate series using user-supplied control values; return cross covariance function*

```
<real> YM(n,m), CCOV(lagmax+1,m,m)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL CCFS (YM, N, M, IYM, LAGMAX, CCOV, ICCOV, JCCOV,
1 NPRT, LDSTAK)
```

===

*CCFM: Compute and print a two-part cross correlation analysis of a pair of series with missing observations*

```
<real> Y1(n), Y2(n)
:
CALL CCFM (Y1, YMISS1, Y2, YMISS2, N)
```

===

*CCFMS: Compute and optionally print a two-part cross correlation analysis of a multivariate series with missing observations using user-supplied control values; return cross covariance function*

```
INTEGER NLPPC(lagmax+1,m,m)
<real> YM(n,m), YMISS(m), CCOV(lagmax+1,m,m)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL CCFMS (YM, YMISS, N, M, IYM, LAGMAX, CCOV, CMISS,
1 ICCOV, JCCOV, NLPPC, INLPPC, JNLPPC, NPRT, LDSTAK)
```

===

*CCFF: Compute and print a two-part cross correlation analysis of a pair of series; use FFT for computations*

```
<real> YFFT1(nfft), YFFT2(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL CCFF(YFFT1, YFFT2, N, LYFFT, LDSTAK)
```

===

*CCFFS: Compute and optionally print a two-part cross correlation analysis of a multivariate series using user-supplied control values; use FFT for computations; return cross covariance function*

```
<real> YMFFT(nfft,m), CCOV(lagmax+1,m,m)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL CCFFS (YMFFT, N, M, IYMFFT, LAGMAX, CCOV,
1 ICCOV, JCCOV, NPRT, LDSTAK)
```

===



Subroutines for Univariate Spectrum Estimation  
Using the Fourier Transform of the Autocorrelation Function

*UFS: Compute and print a univariate Fourier spectrum analysis of a series*

```
<real> Y(n)
:
CALL UFS (Y, N)
```

===

*UFSS: Compute and optionally print a univariate Fourier spectrum analysis of a series using user-supplied control values; return Fourier spectrum and corresponding frequencies*

```
INTEGER LAGS(nw)
<real> Y(n), SPCF(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSS (Y, N, NW, LAGS, NF, FMIN, FMAX, NPRT, SPCF, ISPCF,
1 FREQ, LDSTAK)
```

===

*UFSF: Compute and print a univariate Fourier spectrum analysis of a series; use FFT for computations*

```
<real> YFFT(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSF (YFFT, N, LYFFT, LDSTAK)
```

===

*UFSFS: Compute and optionally print a univariate Fourier spectrum analysis of a series using user-supplied control values; use FFT for computations; return Fourier spectrum and corresponding frequencies*

```
INTEGER LAGS(nw)
<real> YFFT(nfft), SPCF(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSFS (YFFT, N, LYFFT, LDSTAK, NW, LAGS, NF, FMIN, FMAX, NPRT,
1 SPCF, ISPCF, FREQ)
```

===

*UFSM: Compute and print a univariate Fourier spectrum analysis of a series with missing observations*

```
<real> Y(n)
:
CALL UFSM (Y, YMISS, N)
```

===

*UFSMS: Compute and optionally print a univariate Fourier spectrum analysis of a series with missing observations using user-supplied control values; return Fourier spectrum and corresponding frequencies*

```
INTEGER LAGS(nw)
<real> Y(n), SPCF(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSMS (Y, YMISS, N, NW, LAGS, NF, FMIN, FMAX, NPRT, SPCF,
1 ISPCF, FREQ, LDSTAK)
```

===

*UFSV: Compute and print a univariate Fourier spectrum analysis of a series; input covariances rather than original series*

```
<real> ACOV(lagmax+1)
:
CALL UFSV (ACOV, LAGMAX, N)
```

===

*UFSVS: Compute and optionally print a univariate Fourier spectrum analysis of a series using user-supplied control values; input covariances rather than original series; return Fourier spectrum and corresponding frequencies*

```
INTEGER LAGS(nw)
<real> ACOV(lagmax+1), SPCF(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSVS (ACOV, LAGMAX, N, NW, LAGS, NF, FMIN, FMAX,
1 NPRT, SPCF, ISPCF, FREQ, LDSTAK)
```

===

*UFSMV: Compute and print a univariate Fourier spectrum analysis of a series with missing observations; input covariances rather than original series*

```
INTEGER NLPPA(lagmax+1)
<real> ACOV(lagmax+1)
:
CALL UFSMV(ACOV, NLPPA, LAGMAX, N)
```

===

*UFSMVS: Compute and optionally print a univariate Fourier spectrum analysis of a series with missing observations using user-supplied control values; input covariances rather than original series; return Fourier spectrum and corresponding frequencies*

```
INTEGER NLPPA(lagmax+1)
<real> ACOV(lagmax+1), SPCF(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UFSMVS (ACOV, NLPPA, LAGMAX, N, NW, LAGS, NF, FMIN,
1 FMAX, NPRT, SPCF, ISPCF, FREQ, LDSTAK)
```

#### Subroutines for Univariate Spectrum Estimation Using Autoregressive Models

*UAS: Compute and print a univariate autoregressive spectrum analysis of a series*

```
<real> Y(n)
:
CALL UAS (Y, N)
```

===

*UASS: Compute and optionally print a univariate autoregressive spectrum analysis of a series using user-supplied control values; return autoregressive and Fourier spectrum and corresponding frequencies*

```
<real> Y(n), PHI(lagmax), SPCF(nf), SPCA(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UASS (Y, N, IAR, PHI, LAGMAX, LAG, NF, FMIN, FMAX, NPRT,
1 SPCA, SPCF, FREQ, LDSTAK)
```

===

*UASF: Compute and print a univariate autoregressive spectrum analysis of a series; use FFT for computations*

```
<real> YFFT(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UASF(YFFT, N, LYFFT, LDSTAK)
```

===

*UASFS: Compute and optionally print a univariate autoregressive spectrum analysis of a series using user-supplied control values; use FFT for computations; return autoregressive and Fourier spectrum and corresponding frequencies*

```
<real> YFFT(nfft), PHI(lagmax), SPCA(nf), SPCF(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UASFS (YFFT, N, LYFFT, LDSTAK, IAR, PHI, LAGMAX, LAG, NF, FMIN,
1 FMAX, NPRT, SPCA, SPCF, FREQ)
```

===

*UASV: Compute and print a univariate autoregressive spectrum analysis of a series; input covariances rather than original series*

```
<real> ACOV (lagmax+1)
:
CALL UASV (ACOV, LAGMAX, N)
```

===

*UASVS: Compute and optionally print a univariate autoregressive spectrum analysis of a series using user-supplied control values; input covariances rather than original series; return autoregressive and Fourier spectrum and corresponding frequencies*

```
<real> ACOV(lagmax+1), Y(n), PHI(lagmax)
<real> SPCA(nf), SPCF(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL UASVS (ACOV, LAGMAX, Y, N, IAR, PHI, LAG, NF, FMIN,
1 FMAX, NPRT, SPCA, SPCF, FREQ, LDSTAK)
```

===



Subroutines for Univariate Spectrum Estimation Using the  
Direct Fourier Transform

*PGM: Compute and print a periodogram analysis of a series; use FFT for computations*

```
<real> YFFT(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL PGM (YFFT, N, LYFFT, LDSTAK)
```

===

*PGMS: Compute and optionally print a periodogram analysis of a series; use FFT for computations; return periodogram and corresponding frequencies*

```
<real> YFFT(nfft), PER(nf), FREQ(nf)
:
CALL PGMS (YFFT, N, NFFT, LYFFT, IEXTND, NF, PER, LPER, FREQ,
1 LFREQ, NPRT)
```

===

*IPGM: Compute and print an integrated periodogram analysis of a series; use FFT for computations*

```
<real> YFFT(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL IPGM (YFFT, N, LYFFT, LDSTAK)
```

===

*IPGMS: Compute and optionally print an integrated periodogram analysis of a series; use FFT for computations; return integrated periodogram and corresponding frequencies*

```
<real> YFFT(nfft), PERI(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL IPGMS (YFFT, N, LYFFT, LDSTAK, NF, PERI, LPERI, FREQ, LFREQ,
1 NPRT)
```

===

*IPGMP: Compute and print an integrated periodogram analysis of a series; input periodogram rather than original series*

```
<real> PER(nf), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL IPGMP (PER, FREQ, NF, N, LDSTAK)
```

===

*IPGMPS: Compute and optionally print an integrated periodogram analysis of a series; input periodogram rather than original series; return integrated periodogram and corresponding frequencies*

```
<real> PER(nf), FREQ(nf), PERI(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL IPGMPS (PER, FREQ, NF, N, LDSTAK, PERI, NPRT)
```

#### Utility Subroutines

*CENTER: Subtract the series mean from each observation of a series; return the centered series (no printed output)*

```
<real> Y(n), YC(n)
:
CALL CENTER (Y, N, YC)
```

===

*TAPER: Center a series about its mean and apply a split-cosine-bell taper; return the tapered series (no printed output)*

```
<real> Y(n), YT(n)
:
CALL TAPER (Y, N, TAPERP, YT)
```

===

*FFTLEN: Compute the minimum extended series length for using the Singleton FFT; return the extended series length (no printed output)*

```
CALL FFTLEN (N, NDIV, NFFT)
```

===

*MDFLT: Smooth a periodogram by applying a sequence of modified Daniell filters; return the smoothed periodogram (no printed output)*

```
INTEGER KMD(nk)
<real> PER(nf), PERF(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL MDFLT (PER, NF, NK, KMD, PERF, LDSTAK)
```

===

*FFTR: Compute the Fourier coefficients of an input series of REAL (single precision) observations; return the Fourier coefficients (no printed output)*

```
<real> YFFT(n), AB(nfft)
:
CALL FFTR (YFFT, N, NFFT, IEXTND, NF, AB, LAB)
```

Subroutines for Bivariate Spectrum Estimation Using the  
Fourier Transform of the Cross Correlation Function

*BFS: Compute and print a bivariate Fourier spectrum analysis of a pair of series*

```
<real> Y1(n), Y2(n)
:
CALL BFS (Y1, Y2, N)
```

===

*BFSS: Compute and optionally print a bivariate Fourier spectrum analysis of a pair of series using user-supplied control values; return squared coherency and phase components of the cross spectrum and the corresponding frequencies*

```
INTEGER LAGS(nw)
<real> Y1(n), Y2(n), CSPC2(nf,nw), PHAS(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL BFSS (Y1, Y2, N, NW, LAGS, NF, FMIN, FMAX, NPRT,
1  CSPC2, ICSPC2, PHAS, IPHAS, FREQ, LDSTAK)
```

===

*BFSF: Compute and print a bivariate Fourier spectrum analysis of a pair of series; use FFT for computations*

```
<real> YFFT1(nfft), YFFT2(nfft)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL BFSF (YFFT1, YFFT2, N, LYFFT, LDSTAK)
```

===

*BFSFS: Compute and optionally print a bivariate Fourier spectrum analysis of a pair of series using user-supplied control values; use FFT for computations; return squared coherency and phase components of the cross spectrum and the corresponding frequencies*

```
INTEGER LAGS(nw)
<real> YFFT1(nfft), YFFT2(nfft), CSPC2(nf,nw), PHAS(nf,nw),
1  FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL BFSFS (YFFT1, YFFT2, N, LYFFT, LDSTAK, NW, LAGS,
1  NF, FMIN, FMAX, NPRT, CSPC2, ICSPC2, PHAS, IPHAS, FREQ)
```

===

*BFSM: Compute and print a bivariate Fourier spectrum analysis of a pair of series with missing observations*

```
<real> Y1(n), Y2(n)
:
CALL BFSM (Y1, YMISS1, Y2, YMISS2, N)
```

===

*BFSMS: Compute and optionally print a bivariate Fourier spectrum analysis of a pair of series with missing observations using user-supplied control values; return squared coherency and phase components of the cross spectrum and the corresponding frequencies*

```
INTEGER LAGS(nw)
<real> Y1(n), Y2(n), CSPC2(nf,nw), PHAS(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL BFSMS (Y1, YMISS1, Y2, YMISS2, N, NW, LAGS, NF, FMIN,
1  FMAX, NPRT, CSPC2, ICSPC2, PHAS, IPHAS, FREQ, LDSTAK)
```

===



*BFSV: Compute and print a bivariate Fourier spectrum analysis of a pair of series; input covariances rather than original series*

<real> CCOV(lagmax+1,m,m)

⋮

CALL BFSV (CCOV, INDEX1, INDEX2, N, LAGMAX, ICCOV, JCCOV)

===

*BFSVS: Compute and optionally print a bivariate Fourier spectrum analysis of a pair of series using user-supplied control values; input covariances rather than original series; return squared coherency and phase components of the cross spectrum and the corresponding frequencies*

INTEGER LAGS(nw)

<real> CCOV(lagmax+1,m,m), CSPC2(nf,nw), PHAS(nf,nw), FREQ(nf)

DOUBLE PRECISION DSTAK(ldstak)

COMMON /CSTAK/ DSTAK

⋮

CALL BFSVS (CCOV, INDEX1, INDEX2, N, ICCOV, JCCOV,  
1 NW, LAG, NF, FMIN, FMAX, NPRT, CSPC2, ICSPC2, PHAS, IPHAS,  
2 FREQ, LDSTAK)

===

*BFSMV: Compute and print a bivariate Fourier spectrum analysis of a pair of series with missing observations; input covariances rather than original series*

INTEGER NLPPC(lagmax+1,m,m)

<real> CCOV(lagmax+1,m,m)

⋮

CALL BFSMV (CCOV, NLPPC, INDEX1, INDEX2, N,  
1 LAGMAX, ICCOV, JCCOV, INLPPC, JNLPPC)

===

*BFSMVS: Compute and optionally print a bivariate Fourier spectrum analysis of a pair of series with missing observations using user-supplied control values; input covariances rather than original series; return squared coherency and phase components of the cross spectrum and the corresponding frequencies*

```

INTEGER NLPPC(lagmax+1,m,m), LAGS(nw)
<real> CCOV(n,m,m), CSPC2(nf,nw), PHAS(nf,nw), FREQ(nf)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
:
CALL BFSMVS (CCOV, NLPPC, INDEX1, INDEX2, N, ICCOV,
1  JCCOV, INLPPC, JNLPPC, NW, LAGS, NF, FMIN, FMAX, NPRT,
2  CSPC2, ICSPC2, PHAS, IPHAS, FREQ, LDSTAK)

```

#### 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.

AB <-- The vector of dimension at least  $2 \cdot NF$  that contains the  $NF$  real and the  $NF$  imaginary components of the Fourier coefficients of the input series. The real and imaginary components of the Fourier coefficients are returned in the REAL (single precision) array AB stored such that the real and imaginary parts of the complex Fourier coefficients are

$$AB(2I-1) + i \cdot AB(2I) \text{ for } I = 1, \dots, NF,$$

where

$i$  is the complex value  $(-1)^{1/2}$ ; and

$NF$  is the number of harmonic frequencies,  $NF = NFFT/2$ .

The vector YFFT used to input the observed series can also be used in place of the vector AB to conserve storage space.

ACOV --- The vector of dimension at least  $LAGMAX + 1$  that contains the  $LAGMAX+1$  autocovariance function (acvf) estimates for lags  $i = 0, \dots, LAGMAX$ . ACOV is returned from the autocorrelation analysis subroutines; it is input to the spectrum analysis subroutines.

-- continued --

The acvf estimate for lag  $i$ ,  $c(i)$ , is defined by

$$c(i) = c(-i) = \frac{(N-i) \cdot \sum_{t=1}^{N-i} [(Y(t)-\bar{Y}) \cdot (Y(t+i)-\bar{Y}) \cdot \xi(t) \cdot \xi(t+i)]}{N \cdot \sum_{t=1}^{N-i} [\xi(t) \cdot \xi(t+i)]}$$

for lags  $i = 0$  to LAGMAX, where

$\bar{Y}$  is the average of all observed values; and

$\xi(t)$  is an indicator variable, defined by

$\xi(t) = 1$  if  $Y(t)$  is observed ( $Y(t) \neq YMISS$ ), and

$\xi(t) = 0$  if  $Y(t)$  is missing ( $Y(t) = YMISS$ ).

The acvf are stored in ACOV such that  $ACOV(I) = c(I-1)$  for  $I = 1, \dots, LAGMAX+1$ . When there are no missing observations the above formula for the acvf reduces to that of the usual positive definite acvf estimator.

- AMISS <-- The missing value code used within ACOV to indicate that the autocovariance function at a given lag could not be computed because of missing data.
- CCOV --- The three-dimensional array of dimension at least LAGMAX+1 by M by M that contains the cross covariance function (ccvf) estimates. CCOV is returned by the cross correlation analysis subroutines; it is input to the spectrum analysis subroutines.

The ccvf estimate,  $c_{jk}(i)$ , for lag  $i$  between the series stored in the  $j^{\text{th}}$  and  $k^{\text{th}}$  column of YM is defined by

$$c_{jk}(i) = c_{kj}(-i)$$

$$= \frac{(N-i) \cdot \sum_{t=1}^{N-i} [(YM(t,j)-\bar{YM}_j) \cdot (YM(t+i,k)-\bar{YM}_k) \cdot \xi_j(t) \cdot \xi_k(t+i)]}{N \cdot \sum_{t=1}^{N-i} [\xi_j(t) \cdot \xi_k(t+i)]}$$

for lags  $i = 0, \dots, LAGMAX$  and for series  $j = 1, \dots, M$  and  $k = 1, \dots, M$ , where

$\bar{YM}_j$  is the average of all the observed values for the  $j^{\text{th}}$  column of YM;

-- continued --

$\xi_j(t)$  is an indicator variable, defined by

$\xi_j(t) = 1$  if  $YM(t,j)$  is observed ( $YM(t,j) \neq YMMISS(J)$ )

$\xi_j(t) = 0$  if  $YM(t,j)$  is missing ( $YM(t,j) = YMMISS(J)$ ).

When there are no missing observations the above formula for the ccvf reduces to that of the usual positive definite ccvf estimator and when  $j = k$  the above formula is that of the autocovariance function for the  $j^{\text{th}}$  series.

The ccvf are stored in CCOV such that

$CCOV(I,J,K) = c_{jk}(i-1) = c_{kj}(-i+1)$  for  $I = 1, \dots, LAGMAX+1$ .

The appropriate formulas for the ccvf estimates computed by CCF and CCFM can be obtained by letting  $M = 2$  and by substituting  $Y1(t)$  for  $YM(t,1)$  and  $Y2(t)$  for  $YM(t,2)$  in the above.

CMISS <-- The missing value code used in CCOV to indicate that the cross covariance function at a given lag could not be computed because of missing data.

CSPC2 <-- The matrix of dimension at least NF by NW that contains the squared coherency component of the cross spectra between two series, designated j and k, respectively. CSPC2(I,L) contains the smoothed squared coherency value for the  $I^{\text{th}}$  frequency computed using the  $L^{\text{th}}$  lag point specified in LAGS. The returned values are expressed as shown below regardless of the plot option selected.

The estimated smoothed squared coherency component of the cross spectrum is

$$CSPC2(I,L) = \frac{COSPEC(I,L)^2 + QSPEC(I,L)^2}{SPCF_j(I,L) * SPCF_k(I,L)}$$

for  $I = 1, \dots, NF$  and  $L = 1, \dots, NW$ , where

$COSPEC(I,L)$  is the smoothed co-spectra for the  $L^{\text{th}}$  lag point,

$$COSPEC(I,L) = EVEN_0 + 2 \sum_{\ell=1}^{LAGS(L)-1} \{EVEN_{\ell} W_L(\ell) \cos[2\pi\ell(FMIN+\Delta_I)]\};$$

$QSPEC(I,L)$  is the smoothed quadrature spectra for the  $L^{\text{th}}$  lag point,

-- continued --



$$QSPEC(I,L) = 2 \sum_{\ell=1}^{LAGS(L)-1} \{ ODD_{\ell} W_L(\ell) \sin[2\pi\ell(FMIN+\Delta_I)] \};$$

$W_L(\ell)$  is the Parzen lag window function for the  $L^{\text{th}}$  window, defined by

$$\begin{aligned} W(\ell) &= 1 && \ell = 0 \\ W(\ell) &= 1 - 6(|\ell|/LAGS(L))^2 + 6(|\ell|/LAGS(L))^3 && 1 \leq |\ell| \leq LAGS(L)/2 \\ W(\ell) &= 2[1 - (|\ell|/LAGS(L))]^3 && LAGS(L)/2 < |\ell| \leq LAGS(L) \\ W(\ell) &= 0 && LAGS(L) < |\ell| ; \end{aligned}$$

$\Delta_I$  is the frequency increment,

$$\Delta_I = 2(I - 1)(FMAX - FMIN)/(NF - 1);$$

$EVEN_{\ell}$  and  $ODD_{\ell}$  are functions of  $CCOV$  defined by

$$EVEN_{\ell} = CCOV(\ell+1,j,k) + CCOV(\ell+1,k,j) \text{ for } \ell = 0, \dots, LAGS(L)$$

$$ODD_{\ell} = CCOV(\ell+1,j,k) - CCOV(\ell+1,k,j) \text{ for } \ell = 0, \dots, LAGS(L);$$

$SPCF_j(I,L)$  and  $SPCF_k(I,L)$  are the univariate Fourier spectrum estimates for series  $j$  and  $k$ , respectively, computed using the  $L^{\text{th}}$  lag window truncation point. [See argument  $SPCF$ .]

Note that the modifications necessary for series with missing data are included in the computation of  $CCOV$ .

- 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.
- FMAX    --> The maximum frequency, in cycles per sample interval, at which the spectrum is to be computed ( $0.0 \leq FMIN < FMAX \leq 0.5$ ). The default value is 0.5. If FMAX is outside the range FMIN to 0.5 or is not an argument in the CALL statement the default value is used.
- FMIN    --> The minimum frequency, in cycles per sample interval, at which the spectrum is to be computed ( $0.0 \leq FMIN < FMAX \leq 0.5$ ). The default value is 0.0. If FMIN is outside the range 0.0 to FMAX or is not an argument in the CALL statement the default value is used.
- FREQ    --- The vector of dimension at least NF that contains the NF frequency values at which the spectrum is computed. FREQ is an input argument to subroutines IPGMP and IPGMS. The values of FREQ are returned by all other subroutines including it in their CALL statements.

IAR --- The order of the autoregressive model chosen to approximate the series and the order of the autoregressive model to be used in computing the autoregressive spectrum. IAR is returned by the autocorrelation subroutines; it is input to the autoregressive spectrum analysis subroutines.

For the autoregressive spectrum subroutines:

The order IAR of the autoregressive model must not exceed LAGMAX. In no case may IAR exceed  $N/2$ .

If  $IAR > 0$ , the user must supply the coefficients for the order IAR autoregressive model in the vector PHI. (These coefficients are available, for example, from subroutine ACFS and ACFFS or perhaps from the ARIMA model fitting procedure discussed in chapter 13.)

If  $IAR < 0$ , the coefficients for the order  $|IAR|$  autoregressive model will be computed by the STARPAC subroutine using Durbin's recursive method.

If  $IAR = 0$ , the order and model coefficients will be chosen using the model selection criteria discussed below in §E.2.b and the input value of IAR will be overwritten by the selected value. If the  $IAR = 0$  option is used the user must specify the order with a variable, i.e.,

```
IAR = 0
CALL ASPS(..., IAR, ...)
```

NOT

```
CALL ASPS(..., 0, ...).
```

The latter will cause the value of zero to be redefined on many computers including the CYBER 840 and 855.

ICCOV --> The exact value of the first dimension of CCOV as specified in the calling program.

ICSPC2 --> The exact value of the first dimension of CSPC2 as specified in the calling program.

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

IERR = 0 indicates that no errors were detected.

IERR = 1 indicates that improper input was detected.

IEXTND --> The indicator variable used to designate whether zero or the series mean is to be used to extend the series. If IEXTND = 0, zero will be used. If IEXTND ≠ 0, the series mean will be used.

INDEX1 --> The index of the first series used in computing the cross covariance function, corresponding to the subscript j used in the definition of CCOV.

INDEX2 --> The index of the second series used in computing the cross covariance function, corresponding to the subscript k used in the definition of CCOV.

INLPPC --> The exact value of the first dimension of NLPPC as specified in the calling program.

IOD --> The vector of dimension at least NFAC that contains the NFAC values designating the order of each difference factor.

IPHAS --> The exact value of the first dimension of PHAS as specified in the calling program.

ISPCF --> The exact value of the first dimension of SPCF as specified in the calling program.

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

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

JCCOV --> The exact value of the second dimension of CCOV as specified in the calling program.

JNLPPC --> The exact value of the second dimension of NLPPC as specified in the calling program.

KMD --> The vector of dimension at least NK that contains the NK modified Daniell filter lengths. All values in KMD must be even.

LAB --> The length of the vector AB. LAB must equal or exceed NFFT.

LACOV --> The length of the vectors ACOV and NLPPA. LACOV must equal or exceed LAGMAX + 1.

LAG <-> The lag window truncation point to be used for computing the Fourier spectrum. The default value is half the maximum truncation point selected by the algorithm described for the default values of LAGS. If LAG is not an argument of the CALL statement or if LAG is outside the range [1, N-1] the default value will be used. If the user supplied value for LAG is less

-- continued --



than or equal to zero, the input value will be overwritten by the selected value. The user must therefore specify the lag window truncation point with a variable in this case, i.e.,

```
LAG = 0
CALL ASPS(..., LAG, ...)
```

NOT

```
CALL ASPS(..., 0, ...).
```

The latter will cause the value of zero to be redefined on many computers including the CYBER 840 and 855.

LAGMAX --> The maximum lag value for which the correlation coefficients are computed. The default value of LAGMAX is selected by STARPAC as follows.

```
LAGMAX = 100      if 301 < N.
LAGMAX = N/3      if 96 < N < 300.
LAGMAX = 32       if 33 < N < 95.
LAGMAX = N-1      if N < 32.
```

If LAGMAX is less than or equal to zero or if neither LAGMAX nor LAGS is an argument in the CALL statement the default value is used. When LAGS is an argument in the CALL statement, LAGMAX is the largest value in LAGS.

LAGS --> The vector of dimension at least NW that contains the NW lag window truncation points,  $1 < \text{LAGS}(i) < N-1$  for  $i = 1, \dots, \text{NW}$ . By default, four lag window truncation points are used. The smallest lag window truncation point  $T_1$  is selected by examining the covariance function. It is chosen to be  $3/16$  times the lag value beyond which the covariance function remains less than the 95-percent confidence limits for white noise, although in no case is  $T_1$  greater than  $\text{LAGMAX}/8$ . The values of the remaining lag window truncation points are then specified by  $T_2 = 2 \cdot T_1$ ,  $T_3 = 4 \cdot T_1$  and  $T_4 = 8 \cdot T_1$ , resulting in progressively narrower bandwidths. The procedure of using progressively narrower bandwidths to compute the spectrum of a given time series is called window closing. It is discussed in Jenkins and Watts [1968]. If LAGS is not an argument in the CALL statement the four default values are used.

LDSTAK --> The length of the DOUBLE PRECISION workspace vector DSTAK. LDSTAK must equal or exceed the appropriate value given below. In the following specifications of the value of LDSTAK, if the single precision version of STARPAC is being used  $P = 0.5$ , otherwise

-- continued --



P = 1.0. [See chapter 1, §B.] Also, IO = 0 if NPRT = 0 and IO = 1 if NPRT ≠ 0.

For autocorrelation subroutine

ACFS: LDSTAK > 12 + [5·LAGMAX + 1]·P  
ACFMS: LDSTAK > IO·(13 + [6·LAGMAX + 1]·P)  
ACFF: LDSTAK > 6 + NFFT·P  
ACFFS: LDSTAK > 12 + [4·LAGMAX + NFFT + 1]·P  
ACFD: LDSTAK > 16 + [7·LAGMAX + N + 2]·P

For cross correlation subroutine

CCFS: LDSTAK > 12 + [2·M + IO·(4·LAGMAX + 2)]·P  
CCFMS: LDSTAK > (26+M)/2 + [2·M + IO·(4·LAGMAX + 2)]·P  
CCFF: LDSTAK > 6 + NFFT·P  
CCFFS: LDSTAK > 13 + [NFFT + 2·M + IO·(4·LAGMAX + 2)]·P

For univariate Fourier spectrum subroutine

UFSS: LDSTAK > [26 + IO·(NF+5)]/2 +  
[2·LAGMAX + 2 + IO·(2·NF+10)]·P  
UFSF: LDSTAK > 7 + NFFT·P  
UFSFS: LDSTAK > [26 + IO·(NF+5)]/2 +  
[LAGMAX + 1 + NFFT + IO·(2·NF+10)]·P  
UFSMS: LDSTAK > [29 + LAGMAX + 1 + IO·(NF+5)]/2 +  
[2·LAGSMX + 2 + IO·(2·NF+10)]·P  
UFSVS: LDSTAK > [23 + IO·(NF+5)]/2 +  
[LAGMAX + 1 + IO·(2·NF+10)]·P  
UFSMVS: LDSTAK > [23 + IO·(NF+5)]/2 +  
[LAGMAX + 1 + IO·(2·NF+10)]·P

-- continued --

For autoregressive spectrum subroutine

$$\text{UASS: LDSTAK} \geq [32 + \text{IO} \cdot (2 \cdot \text{NF} + 5)]/2 + [2 \cdot \text{LAGMAX} + 2 + \text{IA} \cdot (3 \cdot \text{LAGMAX} + 1) + \text{IO} \cdot (4 \cdot \text{NF} + 10)] \cdot \text{P}$$

$$\text{UASF: LDSTAK} \geq 7 + \text{NFFT} \cdot \text{P}$$

$$\text{UASFS: LDSTAK} \geq [32 + \text{IO} \cdot (2 \cdot \text{NF} + 5)]/2 + [2 \cdot \text{LAGMAX} + 2 + \text{NFFT} + \text{IA} \cdot (3 \cdot \text{LAGMAX} + 1) + \text{IO} \cdot (4 \cdot \text{NF} + 10)] \cdot \text{P}$$

$$\text{UASVS: LDSTAK} \geq [29 + \text{IO} \cdot (2 \cdot \text{NF} + 5)]/2 + [\text{LAGMAX} + 1 + \text{IA} \cdot (3 \cdot \text{LAGMAX} + 1) + \text{IO} \cdot (4 \cdot \text{NF} + 10)] \cdot \text{P}$$

where IA = 0 if IAR ≠ 0 and IA = 1 if IAR = 0.

For direct Fourier spectrum subroutine

$$\text{PGM: LDSTAK} \geq 9 + \text{NFFT} \cdot \text{P}$$

$$\text{IPGM: LDSTAK} \geq [123 + \text{NFFT}/2]/2 + [2 \cdot \text{NFFT} + 206] \cdot \text{P}$$

$$\text{IPGMS: LDSTAK} \geq \text{IO} \cdot ([123 + \text{NFFT}/2]/2 + [2 \cdot \text{NFFT} + 206] \cdot \text{P})$$

$$\text{IPGMP: LDSTAK} \geq [126 + \text{NF}]/2 + [3 \cdot \text{NF} + 206] \cdot \text{P}$$

$$\text{IPGMPs: LDSTAK} \geq \text{IO} \cdot ([123 + \text{NF}]/2 + [2 \cdot \text{NF} + 206] \cdot \text{P})$$

For utility subroutine

$$\text{MDFLT: LDSTAK} \geq 7 + \text{NF} \cdot \text{P}$$

For bivariate Fourier spectrum subroutine

$$\text{BFSS: LDSTAK} \geq [38 + \text{IO} \cdot 4 \cdot \text{NF}]/2 + [7 \cdot \text{LAGMAX} + 7 + \text{IO} \cdot 8 \cdot \text{NF}] \cdot \text{P}$$

$$\text{BFSF: LDSTAK} \geq 7 + \text{NFFT} \cdot \text{P}$$

$$\text{BFSFS: LDSTAK} \geq [38 + \text{IO} \cdot 4 \cdot \text{NF}]/2 + [6 \cdot \text{LAGMAX} + 6 + \text{NFFT} + \text{IO} \cdot 8 \cdot \text{NF}] \cdot \text{P}$$

$$\text{BFSMS: LDSTAK} \geq [45 + 4 \cdot \text{LAGMAX} + \text{IO} \cdot 4 \cdot \text{NF}]/2 + [7 \cdot \text{LAGMAX} + 7 + 2 \cdot \text{NF} + \text{IO} \cdot 8 \cdot \text{NF}] \cdot \text{P}$$

$$\text{BFSVS: LDSTAK} \geq [35 + \text{IO} \cdot 4 \cdot \text{NF}]/2 + [3 \cdot \text{LAGMAX} + 3 + 2 \cdot \text{NF} + \text{IO} \cdot 8 \cdot \text{NF}] \cdot \text{P}$$

$$\text{BFSMVS: LDSTAK} \geq [35 + \text{IO} \cdot 4 \cdot \text{NF}]/2 + [3 \cdot \text{LAGMAX} + 3 + 2 \cdot \text{NF} + \text{IO} \cdot 8 \cdot \text{NF}] \cdot \text{P}$$

LFREQ --> The length of the vector FREQ. LFREQ must equal or exceed NF.

- LPER --> The length of the vector PER. LPER must equal or exceed NF.  
 LPERI --> The length of the vector PERI. LPERI must equal or exceed NF.  
 LYFFT --> The maximum length allowed for the extended series created in YFFT. LYFFT must equal or exceed the actual extended series length, NFFT. If too small a value of LYFFT is used an error message giving the correct value is generated.  
 M --> The number of columns of data in YM.  
 N --> The number of time points in each series. For the correlation analysis subroutines the minimum number of time points is 3; for the spectrum analysis subroutines the minimum number of time points is 17.  
 ND --> The vector of dimension at least NFAC that contains the values designating the number of times each difference factor is to be applied.  
 NDIV --> A required constant used by FFTLEN to determine the extended series length. (See NFFT.) NDIV must be two for a simple FFT. It must be four when the covariance function is being computed.  
 NF <-> The number of frequencies at which the spectrum is computed.

For the Fourier and autoregressive spectrum subroutines and the utility subroutines:

NF is an input argument. The default value of NF is 101. If NF is not an argument of the CALL statement the default value will be used.

For the direct Fourier spectrum subroutines:

NF is returned by the subroutine. The returned value is  $NF = NFFT/2$ .

- NFAC --> The number of difference factors.  
 NFFT --- The minimum extended series length that meets the requirements of the Singleton FFT code. NFFT is returned by FFTLEN; it must be input to PGMS and FFTR; it is computed internally by all of the other subroutines using the FFT.

The extended length of the series is

$$NFFT = N + LAGMAX + K$$

where N is the number of observations in the series.  $K > 2$  is chosen so that NFFT is as small as possible,  $NFFT-2$  is divisible by 4 and has no prime factors greater than 23, and the product of the square-free prime factors of  $NFFT-2$  does not exceed 209. In

-- continued --

general, NFFT will be less than N+LAGMAX+100 for the correlation, Fourier spectrum and autoregressive spectrum subroutines. STARPAC subroutine FFTLEN, when called using the sequence

```
CALL FFTLEN(N+LAGMAX, NDIV, NFFT)
```

with NDIV = 2 when a simple FFT is to be computed and

NDIV = 4 when the covariance function is to be computed,

returns the value of NFFT that is actually used by these subroutines.

NK --> The number of modified Daniell filters to be applied.

NLPPA <-- The vector of dimension at least LAGMAX+1 that contains the LAGMAX+1 values designating the number of lagged product pairs used to compute the autocovariance function at each lag,

$$NLPPA(I) = \sum_{t=1}^{N-i} [\xi(t) \cdot \xi(t+i)] \text{ for } i = 0, \dots, \text{LAGMAX and } I = i + 1,$$

where  $\xi(t) = 1$  if  $Y(t) \neq YMISS$ , and  $\xi(t) = 0$  if  $Y(t) = YMISS$ .

NLPPC <-- The three-dimensional array of dimension at least LAGMAX+1 by M by M that contains the number of lagged product pairs used to compute the cross covariance function for each pair of series at each lag,

$$NLPPC(I,J,K) = \sum_{t=0}^{N-i} [\xi_j(t) \xi_k(t+i)] \text{ for } i = 0, \dots, \text{LAGMAX} \\ \text{and } I = i + 1,$$

where the indices J and K correspond directly to the subscripts j and k, respectively, for  $j = 1, \dots, M$  and  $k = 1, \dots, M$ ; and  $\xi_j(t) = 1$  if  $YM(t,j) \neq YMMISS(j)$  and  $\xi_j(t) = 0$  if  $YM(t,j) = YMMISS(j)$ .

NPRT --> The argument controlling printed output. In each case, when NPRT is not an argument in the subroutine CALL statement the default value is used.

For the correlation analysis subroutines:

If NPRT = 0, the printed output is suppressed.

If NPRT  $\neq$  0, the printed output is provided.

The default value is NPRT  $\neq$  0.

-- continued --



For the Fourier and autoregressive spectrum analysis subroutines:

If NPRT  $\leq$  -1, the spectra are plotted in decibels on a linear scale adjusted so that the peak is at zero.

If NPRT = 0, the printed output is suppressed.

If NPRT  $\geq$  1, the spectra are plotted on a log-linear scale.

The default value is NPRT = -1.

For the periodogram subroutines:

If NPRT  $\leq$  -2, the printed output consists of a page plot of the periodogram versus frequency on a log-linear scale.

If NPRT = -1, the printed output consists of a page plot of the periodogram in decibels on a linear scale.

If NPRT = 0, the printed output is suppressed.

If NPRT = 1, the printed output consists of a vertical plot of the periodogram in decibels on a linear scale. The vertical axis shows the frequency in the left margin.

If NPRT  $\geq$  2, the printed output consists of a vertical plot of the periodogram on a log scale. The vertical axis shows the frequency in the left margin.

The default value is NPRT = -1.

For the integrated periodogram subroutines:

If NPRT = 0, the printed output is suppressed.

If NPRT  $\neq$  0, the printed output is provided.

The default value is NPRT  $\neq$  0.

NW --> The number of different window bandwidths to be used. The default value is four for the Fourier spectrum subroutines and one for the autoregressive spectrum subroutines. When NW is not an argument of the subroutine CALL statement the default value is used.

PER --- The vector of dimension at least NF that contains the NF periodogram values. PER is returned by subroutine PGMS; it must be input to subroutines IPGMP, IPGMPs and MDFLT. The vector YFFT used to input the observed series to PGMS can also be used in place of the vector PER to conserve storage space. (The values in YFFT will be overwritten even when YFFT is not used in place of PER.)

-- continued --

The series periodogram is computed at the harmonic frequencies

$$f_k = k/(NFFT-2) \text{ for } k = 0, \dots, (NFFT-2)/2$$

by a direct Fourier transformation of the series using

$$PER(I) = A(I)^2 + B(I)^2 \text{ for } I = 1, \dots, NF,$$

where

NF is the number of harmonic frequencies at which the periodogram is computed,  $NF = NFFT/2$ ;

A(I) is the real component of the Fourier coefficient,

$$A(I) = (2/(NFFT-2)) \sum_{t=1}^{NFFT} [YCT(t) \cdot \cos(2\pi t(I-1)/(NFFT-2))];$$

B(I) is the imaginary component of the Fourier coefficient,

$$B(I) = (2/(NFFT-2)) \sum_{t=1}^{NFFT} [YCT(t) \cdot \sin(2\pi t(I-1)/(NFFT-2))];$$

YCT(t) is the value of the centered (or tapered) input series at time t. [See arguments YC and YT.]

PERF <-- The vector of dimension at least NF that contains the NF values of the periodogram smoothed by applying a sequence of modified Daniell filters to the raw periodogram. The sequence of filtered series is

$$PERF_{\ell}(I) = \sum_{j=1}^{KMD(\ell)} [h_{\ell}(j) \cdot PERF_{\ell-1}(t_j)] \text{ for } I = 1, \dots, NF \\ \text{and } \ell = 1, \dots, NK,$$

where

KMD( $\ell$ ) is the number of terms in the  $\ell^{\text{th}}$  filter (KMD( $\ell$ ) must be even);

$h_{\ell}(j)$  is the  $j^{\text{th}}$  filter coefficient of the  $\ell^{\text{th}}$  filter, defined by

$$h_{\ell}(j) = [2 \cdot KMD(\ell)]^{-1} \text{ for } j = 1 \text{ and } KMD(\ell)$$

$$h_{\ell}(j) = KMD(\ell)^{-1} \text{ for } j = 2, \dots, KMD(\ell)-1;$$

PERF $_{\ell}$  is the periodogram smoothed using the first  $\ell$  filters (PERF $_0$  is the input periodogram and PERF $_{NK}$  is the series actually returned to the user);

-- continued --

$t_I$  is index of  $PERF_{\ell-1}$  defined by

$$t_I = 2 - [I + (KMD(\ell)/2) - 1 + j] \quad 1 \leq I \leq KMD(\ell)/2$$

$$t_I = I - (KMD(\ell)/2) - 1 + j \quad (KMD(\ell)/2) + 1 \leq I \leq N - KMD(\ell)/2$$

$$t_I = 2N - [I + (KMD(\ell)/2) + 1 + j] \quad N - (KMD(\ell)/2) + 1 \leq I \leq N$$

PERI <-- The vector of dimension at least NF that contains the integrated periodogram values. The vector YFFT used to input the observed series to IPGMS may be used in place of the vector PERI to conserve storage space.

The integrated periodogram is defined by

$$PERI(I) = (N \cdot YVAR)^{-1} \sum_{K=1}^I PER(K) \text{ for } I = 1, \dots, NF$$

where

$$YVAR = (N-1)^{-1} \sum_{t=1}^N (Y(t) - \bar{Y})^2 \text{ with } \bar{Y} = N^{-1} \sum_{t=1}^N Y(t).$$

PHAS <-- The matrix of dimension at least NF by NW that contains the smoothed phase component of the cross spectra between two series, designated j and k, respectively. PHAS(I,L) contains the phase value for the I<sup>th</sup> frequency computed using the L<sup>th</sup> lag window truncation point specified in LAGS.

The estimated phase component of the cross spectrum is

$$PHAS(I,L) = \frac{QSPEC(I,L)^2}{COSPEC(I,L)^2}$$

for  $I = 1, \dots, NF$  and  $L = 1, \dots, NW$ , where

COSPEC(I,L) is the smoothed co-spectra for the L<sup>th</sup> lag point,

$$COSPEC(I,L) = EVEN_0 + 2 \sum_{\ell=1}^{LAGS(L)-1} \{EVEN_{\ell} W_L(\ell) \cos[2\pi\ell(FMIN+\Delta_I)]\};$$

QSPEC(I,L) is the smoothed quadrature spectra for the L<sup>th</sup> lag point,

$$QSPEC(I,L) = 2 \sum_{\ell=1}^{LAGS(L)-1} \{ODD_{\ell} W_L(\ell) \sin[2\pi\ell(FMIN+\Delta_I)]\};$$

-- continued --

$W_L(\ell)$  is the Parzen lag window function for the  $L^{\text{th}}$  window, defined by

$$\begin{aligned} W(\ell) &= 1 && \ell = 0 \\ W(\ell) &= 1 - 6(|\ell|/\text{LAGS}(L))^2 + 6(|\ell|/\text{LAGS}(L))^3 && 1 \leq |\ell| \leq \text{LAGS}(L)/2 \\ W(\ell) &= 2[1 - (|\ell|/\text{LAGS}(L))]^3 && \text{LAGS}(L)/2 < |\ell| \leq \text{LAGS}(L) \\ W(\ell) &= 0 && \text{LAGS}(L) < |\ell| ; \end{aligned}$$

$\Delta_I$  is the frequency increment,

$$\Delta_I = 2 \cdot (I-1) \cdot (\text{FMAX} - \text{FMIN}) / (\text{NF} - 1); \text{ and}$$

$\text{EVEN}_\ell$  and  $\text{ODD}_\ell$  are functions of  $\text{CCOV}$  defined by

$$\text{EVEN}_\ell = \text{CCOV}(\ell+1, j, k) + \text{CCOV}(\ell+1, k, j) \text{ for } \ell = 0, \dots, \text{LAGS}(L)$$

$$\text{ODD}_\ell = \text{CCOV}(\ell+1, j, k) - \text{CCOV}(\ell+1, k, j) \text{ for } \ell = 0, \dots, \text{LAGS}(L).$$

PHI --- The vector of dimension at least  $\text{LAGMAX}$  that contains the IAR coefficients of the order IAR autoregressive model. PHI is returned by the autocorrelation subroutines; it is input or returned by the autoregressive spectrum estimation subroutines depending on the value of IAR.

SPCA <--- The vector of dimension at least  $\text{NF}$  that contains the  $\text{NF}$  values of the autoregressive spectrum. The autoregressive spectrum estimates are defined by

$$\text{SPCA}(I) = S_{\text{IAR}}^2 \left\{ \left| 1 - \sum_{\ell=1}^{\text{IAR}} \text{PHI}(\ell) \cdot e^{i \cdot 2\pi \cdot \ell \cdot (\text{FMIN} + \Delta_I)} \right|^2 \right\}^{-1}$$

for  $I = 1, \dots, \text{NF}$ , where

$S_{\text{IAR}}^2$  is the residual or one step prediction variance of the order IAR autoregressive model,

$$S_{\text{IAR}}^2 = (\text{N} - \text{IAR} - 1)^{-1} \text{N} \left\{ \text{ACOV}(1) - \sum_{\ell=1}^{\text{IAR}} [\text{PHI}(\ell) \cdot \text{ACOV}(\ell+1)] \right\},$$

$i$  is the complex value  $(-1)^{1/2}$ ; and

$\Delta_I$  is the frequency increment,

$$\Delta_I = 2 \cdot (I-1) \cdot (\text{FMAX} - \text{FMIN}) / (\text{NF} - 1).$$

-- continued --



SPCF <-- The array of dimension at least NF by NW that contains the NF values of the Fourier spectrum for each of the NW lag windows. For the autoregressive spectrum analysis subroutines, NW = 1 and SPCF may be dimensioned as a vector of length at least NF. SPCF(I,L) contains the spectrum value for the I<sup>th</sup> frequency computed using the L<sup>th</sup> lag point (see arguments LAG and LAGS). The returned spectrum values are expressed as shown below regardless of the plot option selected.

The estimated Fourier spectrum values are

$$SPCF(I,L) = ACOV(1) + 2 \sum_{\ell=1}^{LAGMAX} \{ACOV(\ell+1) W_L(\ell) \cos[2\pi\ell(FMIN+\Delta_I)]\}$$

for I = 1, ..., NF and L = 1, ..., NW, where

$W_L(\ell)$  is the Parzen lag window function for the L<sup>th</sup> window, defined by

$$\begin{aligned} W(\ell) &= 1 && \ell = 0 \\ W(\ell) &= 1 - 6(|\ell|/LAGS(L))^2 + 6(|\ell|/LAGS(L))^3 && 1 \leq |\ell| \leq LAGS(L)/2 \\ W(\ell) &= 2[1 - (|\ell|/LAGS(L))]^3 && LAGS(L)/2 < |\ell| \leq LAGS(L) \\ W(\ell) &= 0 && LAGS(L) < |\ell| ; \end{aligned}$$

$\Delta_I$  is the frequency increment,

$$\Delta_I = 2 \cdot (I-1) \cdot (FMAX-FMIN)/(NF-1).$$

Note that the modifications necessary for series with missing data are included in the computation of ACOV.

TAPERP --> The total proportion of the input data to be tapered using a split-cosine-bell taper. [See argument YT.] If TAPERP ≤ 0.0, the tapered series is identical to the centered series, YC. If TAPERP > 1.0, a 100-percent taper is applied to the series.

Y --> The vector of dimension at least N that contains the N observations of a time series.

YC <-- The vector of dimension at least N that contains the centered time series. The centered series is

$$YC(t) = Y(t) - \bar{Y} \text{ for } t = 1, \dots, N$$

where  $\bar{Y}$  is the mean of the input series

-- continued --

$$\bar{Y} = N^{-1} \left( \sum_{t=1}^N Y(t) \right).$$

- YFFT <-> The vector of dimension at least NFFT containing the N observations of a time series to be analyzed using the FFT. The length of the vector YFFT must be greater than N to allow for extending the series. Note that the input series is overwritten by the FFT computations.
- YFFT1 <-> The vector of dimension at least NFFT containing the N observations of the first of a pair of series selected from a multivariate time series to be analyzed using the FFT. The length of the vector YFFT must be greater than N to allow for extending the series. Note that the input series is overwritten by the FFT computations.
- YFFT2 <-> The vector of dimension at least NFFT containing the N observations of the second of a pair of series selected from a multivariate time series to be analyzed using the FFT. The length of the vector YFFT must be greater than N to allow for extending the series. Note that the input series is overwritten by the FFT computations.
- YM --> The matrix of dimension at least N by M each of whose M columns contains the N observations of a multivariate time series.
- YMFFT <-> The matrix of dimension at least NFFT by M each of whose M columns contains the N observations of a multivariate time series to be analyzed using the FFT. The first dimension of the array YMFFT must be greater than N to allow for extending the series. Note that the input series are overwritten by the FFT computations.
- YMISS --> The missing value code used within the input series Y to indicate that an observation is missing.
- YMISS1 --> The missing value code used within the input series Y1 to indicate that an observation is missing.
- YMISS2 --> The missing value code used within the input series Y2 to indicate that an observation is missing.
- YMISS --> The vector of dimension at least M that contains the M missing value codes (one for each column) used within each of the M series contained in YM to indicate that an observation is missing.
- YT <-- The vector of dimension at least N that contains the tapered time series. The tapered series is

$$YT(t) = W(t) \cdot YC(t) \text{ for } t = 1, \dots, N,$$

where

-- continued --

$W(t)$  is the split-cosine-bell taper weight used at time  $t$ ,

$$W(t) = 0.5 \cdot \{1 - \cos[\pi(t-0.5)/m]\} \quad 1 \leq t \leq m$$

$$W(t) = 1 \quad m + 1 \leq t \leq N - m$$

$$W(t) = 0.5 \cdot \{1 - \cos[\pi(N-t+0.5)/m]\} \quad N - m + 1 \leq t \leq N$$

with  $m$  computed such that  $2m/N = \text{TAPERP}$  is the proportion of data to be tapered. The importance of tapering is discussed in chapter 5 of Bloomfield [1976].

- Y1 --> The vector of dimension at least  $N$  that contains the  $N$  observations of the first series of a multivariate time series pair.
- Y2 --> The vector of dimension at least  $N$  that contains the  $N$  observations of the second series of a multivariate time series pair.

## E. Computational Methods

### E.1 Algorithms

#### E.1.a Correlation Analysis

The code for computing the autocovariance and cross covariance functions using a fast Fourier transform are based on subroutines written by Jones [1971]. The subroutines which compute the fast Fourier transform and which compute the Fourier transform of real data are those written by Singleton [1969] and the subroutine to compute the cosine transform was written by Richard H. Jones. A discussion of the use of the fast Fourier transform for computing the correlation function can be found on pages 165 to 167 of Bloomfield [1976]. Zeros are automatically appended to the end of the series to meet the requirements of computing the correlation function using an FFT. Correlation function subroutines not using the FFT compute the correlation function directly using the formulas given in §D for arguments ACOV and CCOV.

The autoregressive model coefficients given are the Yule-Walker estimates, computed as described in Appendix A3.2 of Box and Jenkins [1976]. This is not a recommended estimation procedure but rather a means of providing starting values for a maximum likelihood or least squares estimation procedure. Chapter 13 provides subroutines for estimating the least squares values of the parameters of an autoregressive model using these starting values.

#### E.1.b Spectrum Analysis

The transformation used to compute the univariate Fourier spectrum is performed using the algorithm shown on page 311 of Jenkins and Watts [1968], modified to correspond to the definition of the spectrum given for argument SPCF in §D. (The computed spectrum is half that shown in Jenkins and Watts.) The bivariate or cross Fourier spectrum is discussed in chapters 8 and 9 of



Jenkins and Watts. Smoothed squared coherency and phase estimation are discussed in detail on pages 377-380 of Jenkins and Watts. The algorithm used is described on pages 418-420. The confidence interval and significance test for coherency is discussed in Bloomfield [1976] on pages 224-228. The modifications necessary for series with missing data are included in the computation of the covariance function described in §D under arguments ACOV and CCOV. Covariance function values computed using a fast Fourier transform use the FFT code written by Singleton [1968]; a discussion of this technique for computing the covariance function can be found on pages 165 to 167 of Bloomfield [1976]. Subroutines using the FFT automatically subtract the series mean and append zeros to the end of the series to meet the requirements of computing the covariance function using Singleton's code.

The code for computing the autoregressive order selection statistics and autoregressive spectrum estimates is based on subroutines written by Jones [1971]. The coefficients  $\text{PHI}(k)$ ,  $k = 1, \dots, \text{IAR}$  of the autoregressive model are computed from the autocovariance function using the Levinson-Durbin recursive method for solving the Yule-Walker equations discussed in Appendix A3.2 of Box and Jenkins [1976]. The order of the autoregressive model selected by STARPAC is that having the lowest Akaike final prediction error [Akaike, 1974].

Subroutines PGM, IPGM, and IPGMS automatically append zeros to the end of the series to meet the length requirements of the Singleton FFT code [1969].

Subroutine PGMS and FFTR do not automatically center or extend the input series. Centering and extending the series by appending either zeros or the series mean must be done by the user. The resulting computations are wrong if the extension does not comply with the code requirements.

The subroutines for the split-cosine-bell taper and the modified Daniell filter operation were adapted from subroutines TAPER and MODDAN given on pages 116 and 178 of Bloomfield [1976].

## E.2 Computed Results and Printed Output

### E.2.a Correlation Analysis

The Autocorrelation Subroutines. The argument controlling the printed output, NPRT, is discussed in §D. The output from the autocorrelation analysis subroutines includes summary statistics describing the input series; the autocorrelation function estimates and their large lag standard errors; the chi-squared test statistic for white noise and its significance level; the partial autocorrelation function estimates; autoregressive model order selection statistics; and parameter estimates for the selected model. In addition, plots are provided of the autocorrelation and partial autocorrelation function estimates.

The estimated coefficients of the autocorrelation function (acf),  $r(i)$ , are computed with the formula

$$r(i) = \text{ACOV}(i+1)/\text{ACOV}(1) \text{ for lag } i = 1, \dots, \text{LAGMAX}.$$



The estimated standard errors (SE) of the acf are the large lag standard errors discussed on pages 34 to 36 of Box and Jenkins [1976]. The standard error listed at lag  $i$  is computed assuming that the acf is zero for lags greater than  $i$ . The formula is given by

$$SE[r(i)] = SE[r(-i)] = \frac{(N-i) \cdot \left\{ (N-i) + 2 \cdot \sum_{v=1}^q [(N-i-v) \cdot r(v)^2] \right\}^{1/2}}{N \cdot NLPPA(i+1)}$$

for  $i = 1, \dots, LAGMAX$ , where  $q$  is the minimum of  $n - i$  and  $i - 1$ . The user must select the "correct" large lag standard error based on whether the acf estimates are compatible with the assumption that the acf is zero for the selected lag value and beyond. This technique is intended primarily to provide a means of selecting the largest lag value for which the acf is significantly different from zero rather than to provide actual standard error estimates for each acf. The large lag standard error at the selected lag is valid for all acf estimates at lags greater than or equal to the selected lag; the standard error estimates for lag values less than the selected lag are meaningless.

As discussed on pages 64-66 of Box and Jenkins [1976], the estimated partial autocorrelation function (pacf) coefficients,  $p(i)$  for  $i = 1, \dots, LAGMAX$ , are estimates of the  $i^{\text{th}}$  coefficient in an autoregressive process of order  $i$ . Use of the pacf, in conjunction with the acf, to identify an autoregressive moving average process is discussed on pages 174-193 of Box and Jenkins. It should be noted, however, that the method used to compute the pacf [discussed in Appendix A3.2 of Box and Jenkins] is very sensitive to rounding errors. The pacf estimates may not be reliable if the values of the parameters are close to the nonstationarity boundaries.

The order IAR of the autoregressive model selected for the series is that having the lowest Akaike final prediction error [Akaike, 1974]. The modified Akaike information criteria (AIC) is then computed as

$$AIC(J) = N \cdot \text{Log}_e(\text{FPE}_J) \text{ for } J = 1, \dots, LAGMAX,$$

where

LAGMAX is the maximum lag value for which the acvf has been estimated and is therefore the maximum order of autoregressive model which can be selected;

FPE<sub>J</sub> is the Akaike final prediction error, normalized so the minimum final prediction error is one, that is,

$$FPE_J = (S_J^2 \cdot (N+J+1)) / (S_{IAR}^2 \cdot (N+IAR+1));$$

S<sub>J</sub><sup>2</sup> is the residual or one step prediction variance of the order J autoregressive model, computed as

$$S_J^2 = (N-J-1)^{-1} \cdot N \cdot \{ \text{ACOV}(1) - \sum_{\ell=1}^J \text{PHI}(\ell) \cdot \text{ACOV}(\ell+1) \};$$

$S_{\text{IAR}}^2$  is the residual or one step prediction variance of the selected model;

IAR is the order of the AR model which produced the minimum AIC;

N is the number of observations in the series.

The AIC of the selected model order will always be zero.

The partial F-test of the change in the residual sum of squares is a test of the null hypothesis that the order J pacf is zero. The F-ratio is given by

$$F_{1, N-J-1} = (\Delta S_J^2) / S_J^2 = (N-1-J) \cdot \text{PHI}(J)^2 / [1 - \text{PHI}(J)^2]$$

where

$\Delta S_J^2$  is the change in the residual variance due to the hypothesis that  $\text{PHI}(J) = 0$  (1 degree of freedom),

$$\Delta S_J^2 = S_{J-1}^2 - S_J^2;$$

$S_J^2$  is the residual variance of the order J autoregressive model (N-J-1 degrees of freedom).

The significance level of each F-ratio is also listed.

The Cross Correlation Subroutines. The argument controlling the printed output, NPRT, is discussed in §D. The output from the cross correlation analysis subroutines includes summary statistics describing the two input series; the cross correlation coefficient estimates and their standard errors; and a plot of the cross correlation estimates.

The cross correlation function (ccf) estimate for the pair of series stored in the  $j^{\text{th}}$  and  $k^{\text{th}}$  columns of YM is

$$r_{jk}(i) = \text{CCOV}(i+1, j, k) / (\text{CCOV}(1, j, j) \cdot \text{CCOV}(1, k, k))^{1/2}$$

for lag  $i = 0, \dots, \text{LAGMAX}$ ,  $j = 1, \dots, M$ , and  $k = 1, \dots, M$ , where

$$r_{jk}(i) = r_{kj}(-i).$$

The estimated standard errors (SE) of the ccf are computed as

$$SE[r_{jk}(i)] = SE[r_{kj}(-i)]$$

$$= \frac{(N-i) \cdot \left\{ (N-i) + 2 \cdot \sum_{v=1}^q [(N-i-v) \cdot r_{jj}(v) \cdot r_{kk}(v)] \right\}^{1/2}}{N \cdot NLPPC(i+1, j, k)}$$

$$N \cdot NLPPC(i+1, j, k)$$

for lag  $i = 0, \dots, \text{LAGMAX}$  and for series  $j = 1, \dots, M$  and  $k = 1, \dots, M$ , where  $q$  is the smaller of  $N - i$  and  $i - 1$ . This standard error formula assumes that there are no cross correlations between the two series being compared. It does not assume the two series are white noise, although when both series have been prewhitened the summed portion of the equation is approximately zero and the results will approximately equal  $N^{-1/2}$ , which is the standard error assuming white noise. However, the importance of prewhitening before performing cross correlation analysis is clear from the above equation. The presence of autocorrelation in either one or both series can cause a significant increase in the variance of the cross correlation estimates, and, as shown in the example on page 338 of Jenkins and Watts [1968], the cross correlation function estimates can become quite unreliable. Autocorrelation in the input series should be suspected if the standard error estimates computed by the subroutine are not approximately  $N^{-1/2}$ . It is best to use ACF to routinely check for autocorrelation in the input series before performing cross correlation analyses and to prewhiten the series when necessary. The digital filtering subroutines [chapter 10] or the autoregressive model subroutines can be used for prewhitening the input series before cross correlation analysis is performed.

## E.2.b Spectrum Analysis

The Univariate Fourier Spectrum Analysis Subroutines. The argument controlling the printed output, NPRT, is discussed in §D. The output from the univariate Fourier spectrum analysis subroutines consists of four spectrum plots with successively narrower bandwidths. Each spectrum is plotted either in decibels (10 times the base 10 logarithm of the power spectrum), scaled so that the maximum value plotted is zero, or on a logarithmic scale. The 95-percent confidence interval and the bandwidth for each spectrum are shown on the plots.

The bandwidth of a Parzen lag window with truncation point  $T$  is approximately  $1.85/T$  for  $T \ll N$  and a low percentage of missing values. The actual bandwidth is

$$BW = \left\{ \sum_{\ell=-T}^T [W(\ell)^2 \cdot (N - |\ell|)^2 / (N \cdot NLPPA(\ell))] \right\}^{-1}$$

where  $W(\ell)$  is the Parzen lag window function defined in §D under argument SPCF. The bandwidth is indicated by the displayed "B \* W" in the upper-right portion of the plot.



The effective degrees of freedom (edf) in a spectrum estimate using a Parzen lag window is

$$\text{edf} = 2 \cdot \text{BW} \cdot N.$$

For large values of N and a low percentage of missing values this can be approximated by  $3.71 \cdot N/T$ .

A 95-percent confidence interval for the estimated spectrum is

$$\frac{\text{SPCF}(I) \cdot \text{edf}}{\chi^2_{\text{edf}, 0.975}}, \quad \frac{\text{SPCF}(I) \cdot \text{edf}}{\chi^2_{\text{edf}, 0.025}} \quad \text{for } I = 1, \dots, \text{NF},$$

where  $\chi^2_{\text{edf}, \alpha}$  is the  $\alpha$ -percent point function for a chi-square distribution with edf degrees of freedom. Note that when the logarithm of the spectrum is plotted this confidence interval has constant width over frequency although it is not symmetric. In this case, the lower confidence interval limit is

$$\log_{10}[\text{SPCF}(I) \cdot \text{edf} / \chi^2_{\text{edf}, 0.975}] = \log_{10}[\text{SPCF}(I)] + \log_{10}[\text{edf} / \chi^2_{\text{edf}, 0.975}]$$

and the upper confidence interval limit is

$$\log_{10}[\text{SPCF}(I) \cdot \text{edf} / \chi^2_{\text{edf}, 0.025}] = \log_{10}[\text{SPCF}(I)] + \log_{10}[\text{edf} / \chi^2_{\text{edf}, 0.025}]$$

for  $I = 1, \dots, \text{NF}$ .

The width of the confidence interval is indicated by the displayed "C \* I" aligned vertically in the upper-right portion of the plot. The asterisk (\*) separates the upper limit from the lower limit. The user is reminded that the confidence interval applies to individual frequency points, not to the spectrum as a whole.

The Autoregressive Spectrum Subroutines. The argument controlling the printed output, NPRT, is discussed in §D. The output from the autoregressive spectrum analysis subroutines consists of a plot of the autoregressive and Fourier spectra either in decibels (10 times the base 10 logarithm of the power spectrum) scaled such that the maximum value plotted is zero or on a logarithmic scale. The output also includes the autoregressive order selection statistics when the user does not supply the value of the order via argument IAR. The bandwidth and the length of the 95-percent confidence interval for the Fourier spectrum are shown on the plot. The bandwidth is not relevant to the autoregressive spectrum. The Fourier spectrum and its confidence intervals should be used in interpreting the autoregressive spectrum since the confidence intervals for the autoregressive spectrum are not computed.

The autoregressive order selection statistics include the modified Akaike information criteria (AIC); the results of a partial F-test of the change in the residual sum of squares with the addition of the most recent



autoregressive parameter; the order of the autoregressive model selected; and the Yule-Walker estimates of the parameters of the selected model.

The modified Akaike information criteria (AIC) is

$$AIC(J) = N \cdot \log_e(FPE_J) \text{ for } J = 0, \dots, LAGMAX,$$

where

LAGMAX is the maximum lag value for which the autocovariance has been computed, and is therefore also the maximum order autoregressive model which can be selected;

FPE<sub>J</sub> is the Akaike final prediction error, normalized so that the minimum final prediction variance  $S_{IAR}^2$  is one, that is,

$$FPE_J = ((N+J+1) \cdot S_J^2) / ((N+IAR+1) \cdot S_{IAR}^2);$$

$S_J^2$  is the residual or one step prediction variance of the order J model;

$S_{IAR}^2$  is the residual or one step prediction variance of the selected model;

IAR is the order of the model which produced the minimum AIC, i.e., the order having the lowest Akaike final prediction error [Akaike, 1974].

The AIC of the selected model order will always be zero.

The partial F-test of the change in the residual sum of squares is a test of the null hypothesis that the  $J^{\text{th}}$  autoregressive coefficient of the order J model is zero. The F-ratio is given by

$$F_{1, N-J-1} = (\Delta S_J^2) / S_J^2$$

where

$\Delta S_J^2$  is the change in the residual variance due to the hypothesis that  $\text{PHI}(J) = 0$  (one degree of freedom),

$$\Delta S_J^2 = S_{J-1}^2 - S_J^2;$$

$S_J^2$  is the residual variance of the order J autoregressive model (N-J-1 degrees of freedom).

The significance level of each F-ratio is also listed.

The estimated Fourier spectrum and its bandwidth and confidence interval are computed as described above.

The Direct Fourier Transform Subroutines. The argument controlling the printed output, NPRT, is discussed in §D. The output from the periodogram

subroutines consists of a plot of the computed periodogram displayed either in decibels (10 times the base 10 logarithm of the periodogram estimates), scaled so that the maximum value plotted is zero, or on a logarithmic scale. The output from the integrated periodogram subroutines consists of a one-page plot of the integrated periodogram accompanied by 95-percent confidence bands for testing the null hypothesis of white noise. These bands are the Kolmogoroff-Smirnov probability limits applicable to cumulative distribution functions [Hald, 1952].

The Bivariate Fourier Spectrum. The argument controlling the printed output, NPRT, is discussed in §D. The output from the bivariate Fourier spectrum subroutines consists of four spectrum plot pairs (a squared coherency plot and a phase plot) with successively narrower bandwidths chosen by the window-closing algorithm. The 95-percent confidence intervals and the 95-percent significance levels are shown on the coherency plots.

The bandwidth of a Parzen lag window with truncation point T is approximately  $3.71 \cdot N/T$  for  $T \ll N$  and a low percentage of missing values. The actual bandwidth is

$$BW = \left\{ \sum_{k=-T}^T [W(k)^2 (N-|k|)^2 / (N \cdot NLPPC(k))] \right\}^{-1}.$$

NLPPC(k) is the number of lagged product pairs used to compute the covariance function for lag k, that is

$$NLPPC(k) = \sum_{t=1}^{N-|k|} \xi_1(t) \xi_2(t-|k|)$$

where  $\xi(t)$  is an indicator variable:  $\xi(t) = 1$  if  $Y(t)$  has been observed, and  $\xi(t) = 0$  if  $Y(t)$  is missing.

## F. Examples

Autocorrelation Analysis. In the example program of figure F-1a ACF is used to compute and plot the autocorrelations, partial autocorrelations, and autoregressive order selection statistics of the input series Y, where the data used is the series  $X_1$ , a simulated first order autoregressive model with  $\text{PHI}(1) = 0.6$ , listed on page 362 in Jenkins and Watts [1968]. The correlation estimates for this series are given on page 421 of Jenkins and Watts.

Figures F-1b through F-1e show four pages of output from ACF. The first two pages are autocorrelations and the second two pages are partial autocorrelations. Note that for all correlation plots the value of the lag, i, is shown on the left margin and the actual correlation coefficients are shown on the right margin of each of the plots.

Cross Correlation Analysis. In the example program of figure F-2a CCF is used to compute and plot the cross correlations between the first 50 values of the series  $X_1$  and  $X_2$  listed on page 361 of Jenkins and Watts [1968]. The correlation estimates for these series are given on page 420 of Jenkins and Watts.

Figures F-2b through F-2d show the two-part output from CCF. Note that for all correlation plots the value of the lag,  $i$ , is shown on the left margin and the actual correlation coefficients are shown on the right margin of each of the plots.

Univariate Fourier Spectrum Analysis. In the example program of figure F-3a, UFS is used to compute and plot the univariate Fourier spectrum estimates for the first 50 values of the series listed on page 318 of Jenkins and Watts [1968]; the spectrum of this series is shown for various bandwidths on page 270 of Jenkins and Watts [1968].

Figures F-3b through F-3e show the four plots of the power spectrum (in decibels versus frequency) produced by UFS. Each spectrum plotted is computed using a narrower bandwidth than the preceding one. The bandwidth, lag window truncation point and effective degrees of freedom for the spectrum are listed at the top of each page. The bandwidth and the 95-percent confidence interval of the individual spectrum values are shown graphically on each plot.

Autoregressive Spectrum Analysis. In the example program of figure F-4a, UAS is used to compute and plot the univariate autoregressive and Fourier spectrum estimates for the first 50 values listed on page 318 of Jenkins and Watts [1968]. The theoretical and Fourier spectra of this series are shown on page 270 of Jenkins and Watts [1968].

Figures F-4b and F-4c show the two pages of output produced by UAS. The first page lists the autoregressive order selection statistics while the second gives a plot of the autoregressive and Fourier spectra. The bandwidth, lag window truncation point and effective degrees of freedom of the Fourier spectrum are listed at the top of the plot. The 95-percent confidence interval and the bandwidth of the Fourier spectrum are displayed on the plot. The concept of bandwidth does not apply to the autoregressive spectrum and confidence intervals for the autoregressive spectrum are not computed.

Direct Fourier Transform of a Univariate Series and Utility Subroutines. In the example program of figures F-5a and F-5b, TAPER is used to center the input series and to taper 10 percent of the data at the ends of the series (5 percent at each end) and PGMS is used to compute the raw periodogram of the tapered series. MDFLT is then used to smooth the raw periodogram returned by PGMS with three applications of a modified Daniell filter of length eight. PPL (chapter 3) is used to produce a log plot the smoothed periodogram versus frequency. (VPL could also have been used to display the results.) The data used are the Wolf sunspot numbers from 1700 to 1960 as tabulated by Waldmeier

[1961]. The raw and smoothed periodograms of the tapered series are shown on pages 95 and 176, respectively, of Bloomfield [1976].

Figure F-5c shows the raw periodogram output from PGMS while figure F-5d shows the smoothed periodogram. There is no printed output from TAPER or MDFLT.

Bivariate Fourier Spectrum Analysis. In the example program of figure F-6a BFS is used to compute and plot the cross spectrum for the series  $X_1$  and  $X_2$  listed on page 361 of Jenkins and Watts [1968]. The squared coherency and phase spectra for these series are shown on pages 387 and 388 of Jenkins and Watts. Figures F-6b through F-6i show the eight pages of output from BFS.



## Autocorrelation Analysis

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE ACF USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(300)
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C PRINT TITLE AND CALL ACF FOR AUTOCORRELATION ANALYSIS
C
C WRITE (6, 102)
C CALL ACF (Y, N)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I5)
C 101 FORMAT (10F6.2)
C 102 FORMAT ('RESULTS OF STARPAC AUTOCORRELATION SUBROUTINE ACF')
C END
DATA: 100
-2.07 -1.15 0.69 -0.46 -1.49 -0.70 -1.07 -0.69 -0.68 1.27
-1.05 -0.05 -0.84 -0.62 -0.49 -1.29 -0.49 -1.06 -0.38 -0.52
-0.13 1.30 -1.51 -0.43 -1.33 -0.78 0.31 -0.95 -0.90 -0.30
-1.02 -0.93 0.15 1.40 1.22 0.99 0.70 1.70 2.78 1.98
1.39 1.85 2.60 0.51 2.77 1.16 1.07 -0.48 -0.52 0.37
0.00 -1.99 -1.75 0.70 0.73 1.16 0.06 -0.02 1.10 -0.35
-1.67 -1.57 1.16 1.84 3.35 0.40 0.45 1.30 0.93 1.17
-1.74 -1.28 -0.07 1.50 0.53 0.20 -0.42 1.18 0.82 1.50
2.92 1.18 1.23 3.16 0.79 0.68 1.14 1.02 1.02 -0.71
-0.17 -1.50 -0.26 -0.38 0.93 -0.33 -1.12 -2.95 -2.09 -1.11
```

Figure F-1a

Example program using ACF

RESULTS OF STARPAC AUTOCORRELATION SUBROUTINE ACF  
AUTOCORRELATION ANALYSIS

AVERAGE OF THE SERIES = .1450000  
STANDARD DEVIATION OF THE SERIES = 1.8277758  
NUMBER OF TIME POINTS = 100  
LARGEST LAG VALUE USED = 33

AUTOCORRELATION FUNCTION ESTIMATE (ACF)

LAG	1	2	3	4	5	6	7	8	9	10	11	12
ACF	.24	.34	.22	.26	.27	.15	.06	.03	.06	.02	-.04	-.11
STANDARD ERROR	.10	.12	.13	.13	.14	.14	.14	.14	.14	.14	.14	.14
LAG	13	14	15	16	17	18	19	20	21	22	23	24
ACF	-.14	-.10	-.10	-.11	-.12	-.07	.03	.04	.02	-.03	.02	.06
STANDARD ERROR	.14	.14	.14	.14	.14	.14	.14	.14	.14	.14	.14	.13
LAG	25	26	27	28	29	30	31	32	33			
ACF	.09	-.03	-.04	-.06	-.06	-.03	-.09	-.16	-.17			
STANDARD ERROR	.13	.13	.13	.13	.13	.13	.13	.13	.13			

THE CHI SQUARE TEST STATISTIC OF = 79.54  
THE NULL HYPOTHESIS OF WHITE NOISE = 33  
DEGREES OF FREEDOM = 33  
OBSERVED SIGNIFICANCE LEVEL = .0000

Figure F-1b  
Example of ACF output

AUTOCORRELATION FUNCTION ESTIMATE (ACF)



Figure F-1c

Example of ACF output (continued)

PARTIAL AUTOCORRELATION FUNCTION ESTIMATE (PACF)  
AND AUTOREGRESSIVE ORDER SELECTION STATISTICS

LAG	1	2	3	4	5	6	7	8	9	10	11	12
PACF	.94	.06	.02	.17	.09	-.11	-.09	.00	.02	-.06	-.03	-.08
AIC	.00	1.60	3.96	2.59	3.70	4.59	6.32	8.34	10.32	11.94	13.85	15.16
F RATIO	40.28	.39	.05	2.92	.80	1.04	.26	.00	.03	.36	.10	.62
F PROBABILITY	.00	.94	.83	.09	.37	.31	.61	.99	.87	.55	.75	.43
LAG	13	14	15	16	17	18	19	20	21	22	23	24
PACF	-.06	.01	-.02	-.01	-.01	.07	.10	.00	.01	-.05	.02	.01
AIC	16.81	16.83	20.83	22.98	24.94	26.96	27.63	29.72	31.80	33.65	35.72	37.83
F RATIO	.35	.02	.04	.01	.00	.36	.81	.00	.01	.20	.04	.01
F PROBABILITY	.55	.89	.84	.93	.94	.55	.37	.98	.91	.66	.84	.90
LAG	25	26	27	28	29	30	31	32	33			
PACF	-.03	-.10	.00	-.08	-.05	.04	-.06	-.10	-.03			
AIC	39.87	41.10	43.26	44.82	46.78	48.80	50.70	52.03	54.21			
F RATIO	.07	.68	.00	.44	.16	.13	.22	.61	.05			
F PROBABILITY	.80	.41	.97	.51	.69	.72	.64	.44	.83			

ORDER AUTOREGRESSIVE PROCESS SELECTED = 1  
ONE STEP PREDICTION VARIANCE OF PROCESS SELECTED = 1.1600919

YULE-WALKER ESTIMATES OF THE COEFFICIENTS OF THE AUTOREGRESSIVE PROCESS SELECTED

COEFFICIENT NUMBER 1  
COEFFICIENT VALUE .9397

Figure F-1d

Example of ACF output (continued)



PARTIAL AUTOCORRELATION FUNCTION ESTIMATE (PACF)



Figure F-1e  
Example of ACF output (continued)

## Cross Correlation Analysis

```
MAIN PROGRAM:      PROGRAM EXAMPL
                   C
                   C DEMONSTRATE CCF USING SINGLE PRECISION VERSION OF STARPAC
                   C RUN ON CYBER 100/040.
                   C
                   C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
                   C (SEE CHAPTER 1, SECTION D.4)
                   C
                   C N.B. DECLARATION OF Y1 AND Y2 MUST BE CHANGED TO DOUBLE PRECISION
                   C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
                   C
                   C REAL Y1(100), Y2(100)
                   C
                   C READ NUMBER OF OBSERVATIONS
                   C   SERIES 1
                   C   SERIES 2
                   C
                   C READ 100, M
                   C READ 101, (Y1(I), I=1,M)
                   C READ 101, (Y2(I), I=1,M)
                   C
                   C PRINT TITLE AND CALL CCF FOR CROSS CORRELATION ANALYSIS
                   C
                   C WRITE (6, 102)
                   C CALL CCF (Y1, Y2, M)
                   C
                   C STOP
                   C
                   C FORMAT STATEMENTS
                   C
                   C 100 FORMAT (I5)
                   C 101 FORMAT (12F6.2)
                   C 102 FORMAT ('1RESULTS OF STARPAC CROSS CORRELATION SUBROUTINE CCF')
                   C   END
DATA:              50
-0.00 -0.16 -1.07 -1.12 1.30 2.13 2.76 0.56 -0.69 -1.79 -3.82 -2.38
1.00 0.70 -0.15 0.90 0.11 -0.35 -0.73 0.09 -1.63 -0.44 -1.37 -1.71
-1.22 -2.00 -0.22 0.30 1.31 0.71 0.32 0.40 -1.00 -0.94 -1.54 -0.13
1.02 0.02 -0.77 0.11 -0.60 -0.92 -0.09 1.23 1.46 0.61 0.42 2.14
3.18 2.10
0.79 1.12 -1.10 -2.39 -1.75 -0.82 -0.36 1.27 1.75 2.44 0.36 -2.10
-1.93 -1.30 -1.75 -0.34 0.74 0.49 0.70 0.71 0.09 0.59 1.54 0.14
0.55 -1.40 -2.55 -1.66 -0.43 0.50 2.18 -0.24 0.50 -0.18 -1.55 -0.64
-1.09 0.90 -0.66 -0.35 0.40 0.50 0.05 -0.60 0.24 0.50 -1.26 -0.25
0.25 2.18
```

Figure F-2a

Example program using CCF

RESULTS OF STARPAC CROSS CORRELATION SUBROUTINE CCF

CROSS CORRELATION ANALYSIS

	SERIES 1	SERIES 2
AVERAGE OF THE SERIES	-0.996000E-01	-0.996000E-01
STANDARD DEVIATION OF THE SERIES	1.399913	1.202491
NUMBER OF TIME POINTS	50	50
LARGEST LAG VALUE TO BE USED	0	32

CROSS CORRELATION FUNCTION ESTIMATE (CCF)

CCF CORRELATES SERIES 1 AT TIME T WITH SERIES 2 AT TIME T + K.  
 (IF PEAK CORRELATION OCCURS AT POSITIVE (NEGATIVE) LAG  
 THEN SERIES 1 LEADS (LAGS) SERIES 2)

LAG	-29	-26	-27	-28	-29	-30	-31	-32										
CCF	-0.10	.06	.16	.16	.15	.11	.00	-.10										
STANDARD ERROR	.15	.14	.14	.14	.13	.13	.13	.12										
LAG	-13	-14	-15	-16	-17	-18	-19	-20	-21	-22	-23	-24						
CCF	-0.07	-0.05	-0.11	-0.19	-0.09	.00	.16	.17	.09	-.06	-.12	-.14						
STANDARD ERROR	.18	.18	.18	.18	.17	.17	.17	.16	.16	.16	.15	.15						
LAG	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12						
CCF	-0.22	-0.60	-0.33	-0.05	.15	.37	.35	.13	-.05	-.13	.00	.04						
STANDARD ERROR	.21	.21	.21	.21	.20	.20	.20	.20	.19	.19	.19	.19						
LAG	0																	
CCF	-0.02																	
STANDARD ERROR	.14																	
LAG	1	2	3	4	5	6	7	8	9	10	11	12						
CCF	.41	.51	.48	.33	.01	-.10	-.36	-.26	-.04	-.02	.06	.02						
STANDARD ERROR	.21	.21	.21	.21	.20	.20	.20	.20	.19	.19	.19	.19						
LAG	13	14	15	16	17	18	19	20	21	22	23	24						
CCF	.04	.05	.08	.20	.14	.12	-.19	-.27	-.20	-.14	.07	.17						
STANDARD ERROR	.18	.18	.18	.18	.17	.17	.17	.16	.16	.16	.15	.15						
LAG	25	26	27	28	29	30	31	32										
CCF	.19	.10	-0.08	-0.08	-0.10	-.11	-.07	.02										
STANDARD ERROR	.15	.14	.14	.14	.13	.13	.13	.12										

Figure F-2b

Example of CCF output

CROSS CORRELATION FUNCTION ESTIMATE (CCF)

CCF CORRELATES SERIES 1 AT TIME T WITH SERIES 2 AT TIME T + K.  
 (IF PEAK CORRELATION OCCURS AT POSITIVE (NEGATIVE) LAG  
 THEN SERIES 1 LEADS (LAGS) SERIES 2)

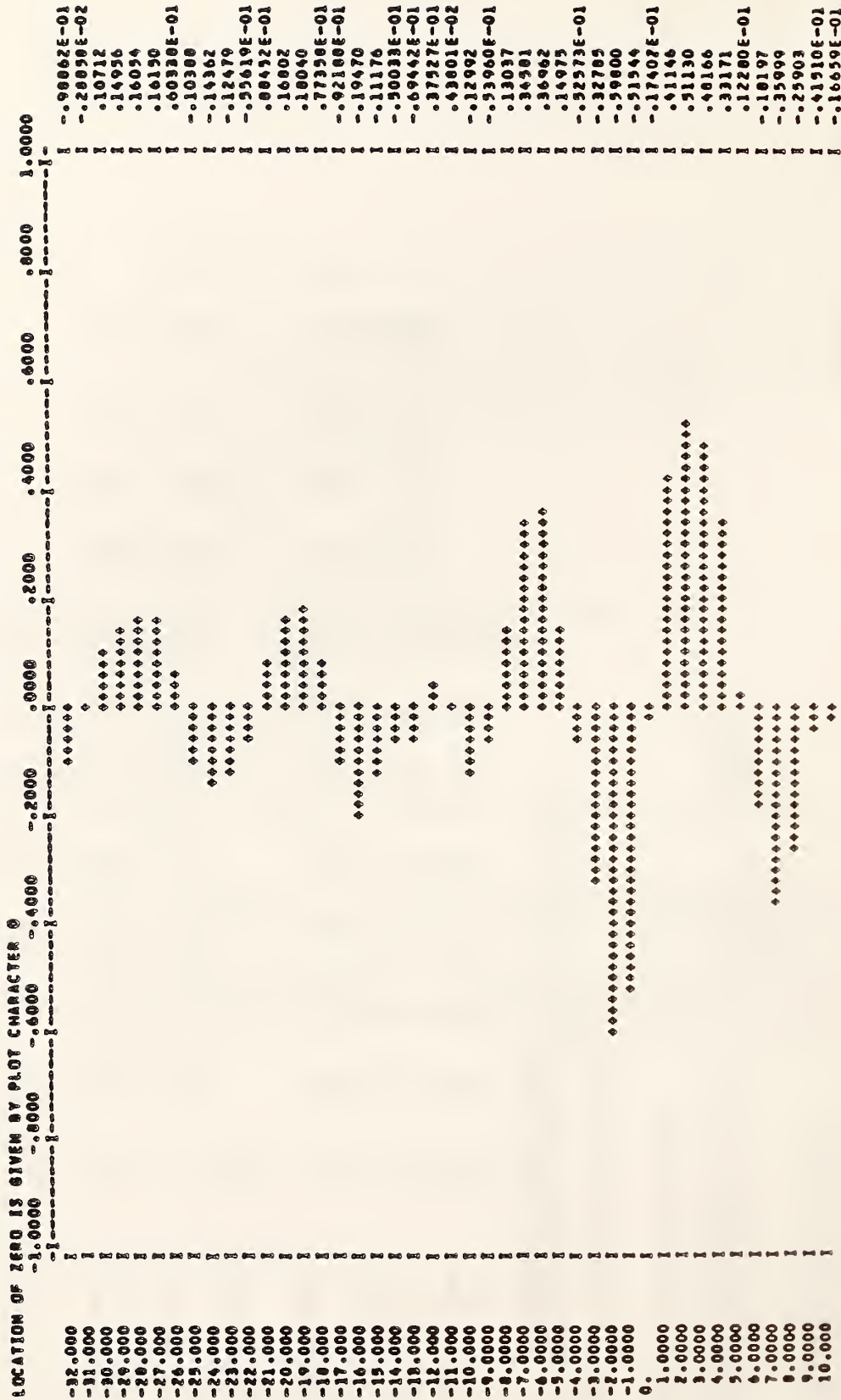


Figure F-2c

Example of CCF output (continued)





## Univariate Fourier Spectrum Analysis

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE UFS USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION 0.4]
C
C N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(300)
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C PRINT TITLE AND CALL UFS FOR UNIVARIATE FOURIER SPECTRUM ANALYSIS
C
C WRITE (6, 102)
C CALL UFS (Y, N)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I5)
C 101 FORMAT (10F6.2)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' UNIVARIATE FOURIER SPECTRUM ANALYSIS SUBROUTINE UFS')
C
C END
DATA: 90
-0.88 -0.12 -0.89 -1.38 -0.07 1.03 2.14 0.35 -1.10 -1.78
-2.76 -1.77 0.98 1.00 -0.70 -1.01 -1.30 -0.85 -0.46 1.63
0.06 -0.17 -1.01 -1.04 -0.66 -1.12 -0.91 -0.71 -0.20 -0.13
0.14 1.59 -0.76 -1.08 -1.77 -1.20 0.45 -0.07 -0.63 -0.35
-0.87 -0.62 0.28 1.90 2.14 1.09 0.31 1.07 2.67 2.44
```

Figure F-3a

Example program using UFS

RESULTS OF STARPAC UNIVARIATE FOURIER SPECTRUM ANALYSIS SUBROUTINE UFS

-- SMOOTHED FOURIER SPECTRUM --  
 (PARZEN WINDOW WITH LAG WIND. TRUNC. PY. 4 / BW. .4685 / EDF. 47)

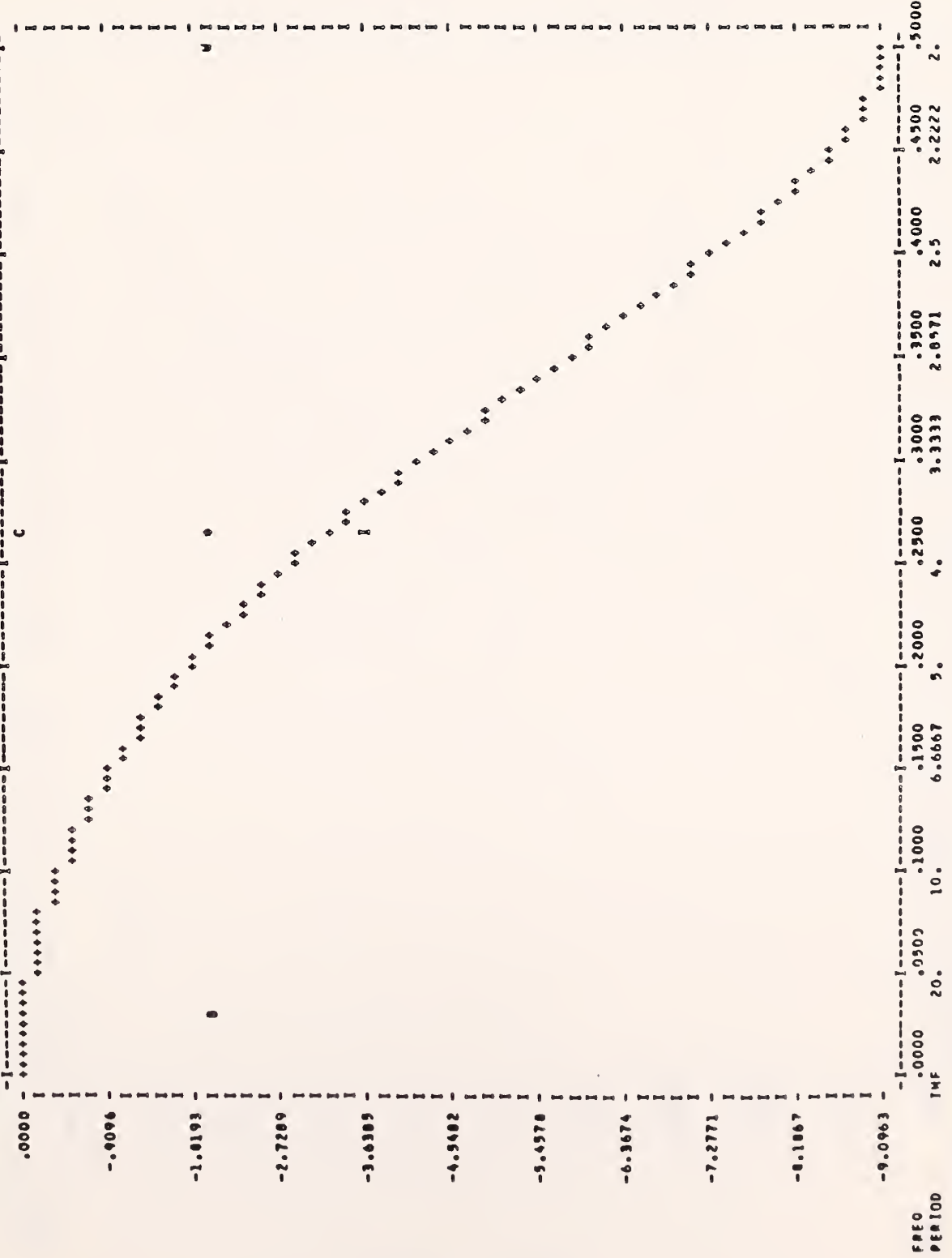


Figure F-3b

Example of UFS output

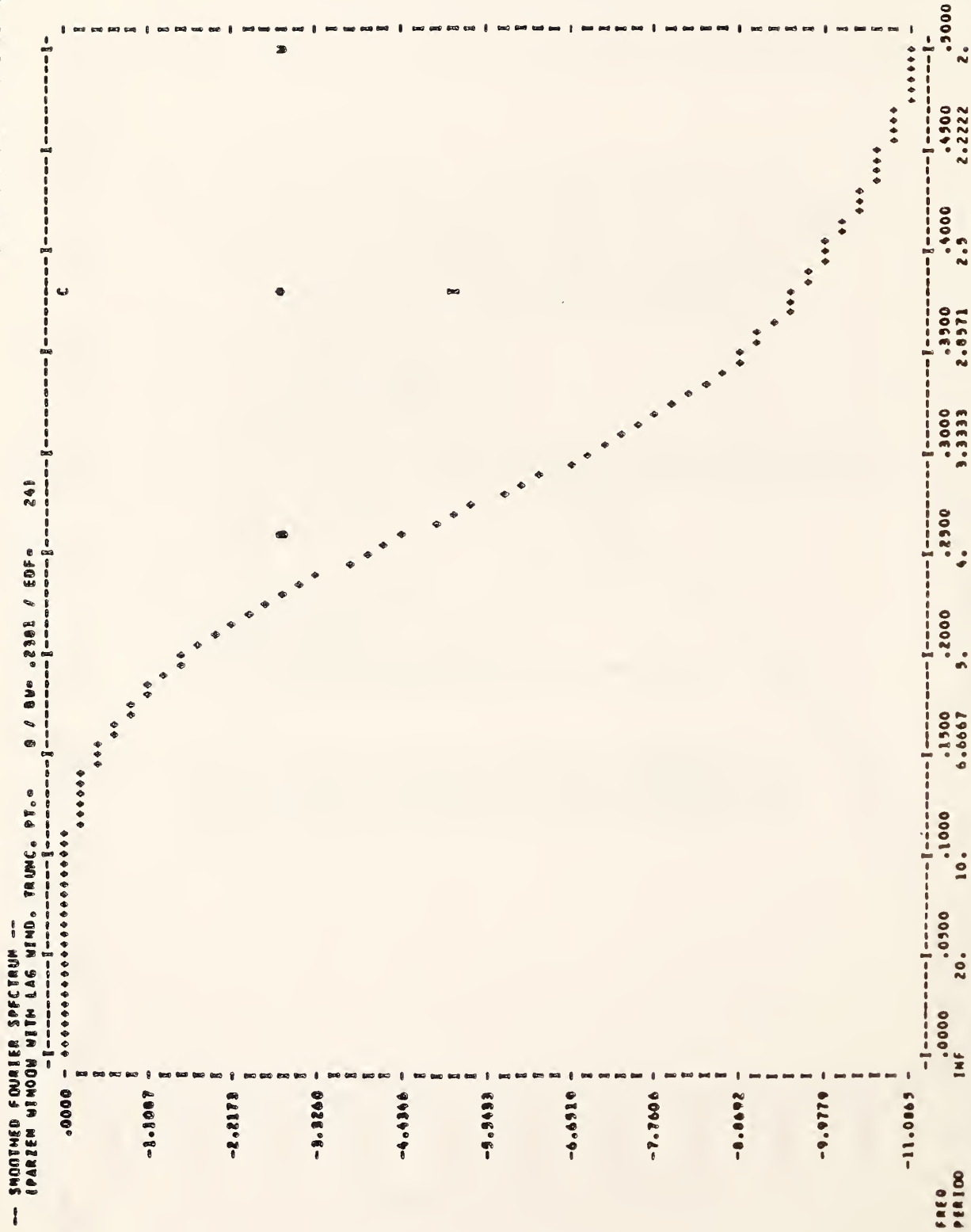


Figure F-3c  
Example of UFS output (continued)



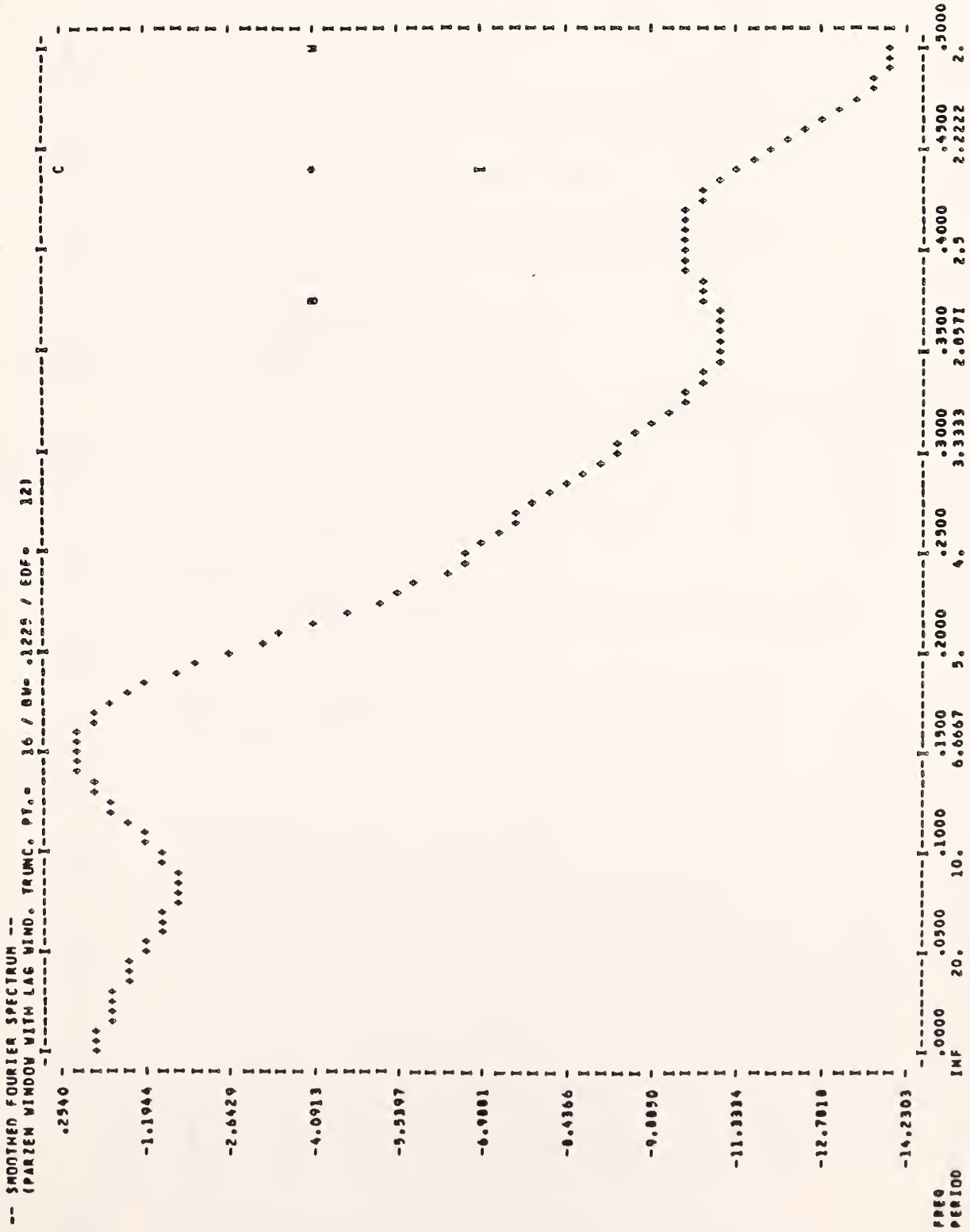


Figure F-3d

Example of UFS output (continued)

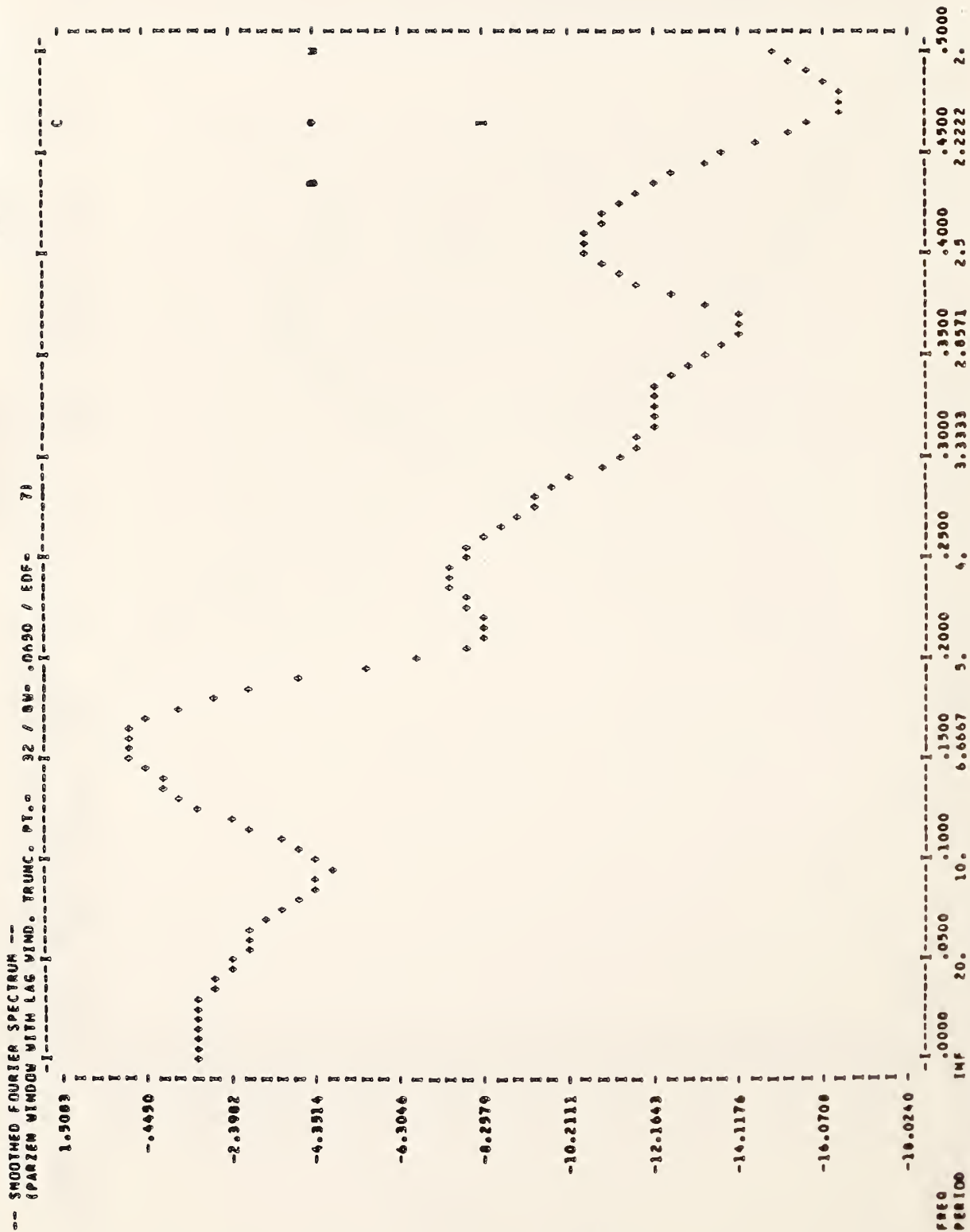


Figure F-3e

Example of UFS output (continued)

## Autoregressive Spectrum Analysis

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE UAS USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 100/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(300)
C
C READ NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C PRINT TITLE AND CALL UAS FOR UNIVARIATE AUTOREGRESSIVE SPECTRUM ANALYSIS
C
C WRITE (6, 102)
C CALL UAS (Y, N)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I9)
C 101 FORMAT (10F6.2)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' UNIVARIATE AUTOREGRESSIVE SPECTRUM ANALYSIS SUBROUTINE UAS')
C END
DATA:      90
-0.88 -0.12 -0.89 -1.38 -0.07 1.03 2.14 0.35 -1.10 -1.78
-2.76 -1.77 0.98 1.00 -0.70 -1.01 -1.30 -0.85 -0.46 1.63
0.06 -0.17 -1.01 -1.04 -0.66 -1.12 -0.51 -0.71 -0.20 -0.13
0.14 1.59 -0.76 -1.08 -1.77 -1.20 0.45 -0.07 -0.63 -0.35
-0.87 -0.62 0.28 1.90 2.14 1.05 0.31 1.07 2.67 2.44
```

Figure F-4a

Example program using UAS

RESULTS OF STARPAC UNIVARIATE AUTOREGRESSIVE SPECTRUM ANALYSIS SUBROUTINE UAS

AUTOREGRESSIVE ORDER SELECTION STATISTICS

LAG	1	2	3	4	5	6	7	8	9	10	11	12
AIC	5.08	.00	1.91	3.54	3.90	5.93	7.30	7.35	9.30	11.37	13.46	15.40
F RATIO	23.53	7.15	.09	.35	1.48	.01	.57	1.68	.10	.01	.02	.14
F PROBABILITY	.00	.01	.76	.55	.23	.93	.45	.20	.75	.91	.89	.71

LAG	13	14	15	16	17	18	19	20	21	22	23	24
AIC	16.80	18.98	20.84	22.94	25.12	27.38	29.94	27.94	29.59	31.88	34.44	36.43
F RATIO	.55	.00	.24	.10	.06	.04	2.37	.24	.45	.12	.00	.32
F PROBABILITY	.46	.96	.63	.75	.81	.84	.13	.63	.51	.74	.94	.58

LAG	25	26	27	28	29	30	31	32
AIC	38.98	41.68	44.55	46.83	49.91	53.00	55.64	59.10
F RATIO	.08	.04	.00	.25	.04	.04	.25	.00
F PROBABILITY	.79	.85	.99	.62	.85	.85	.63	1.00

ORDER AUTOREGRESSIVE PROCESS SELECTED = 2  
 ONE STEP PREDICTION VARIANCE OF PROCESS SELECTED = .00356889

YULE-WALKER ESTIMATES OF THE COEFFICIENTS OF THE AUTOREGRESSIVE PROCESS SELECTED

COEFFICIENT NUMBER	1	2
COEFFICIENT VALUE	.7819	-.3634

Figure F-4b

Example of UAS output



FOURIER SPECTRUM (+) (LAG WIND, TRUNC. PT.= 16 / BW=.1225 / EDF= 12)  
 AND ORDER 2 AUTOREGRESSIVE SPECTRUM (.)

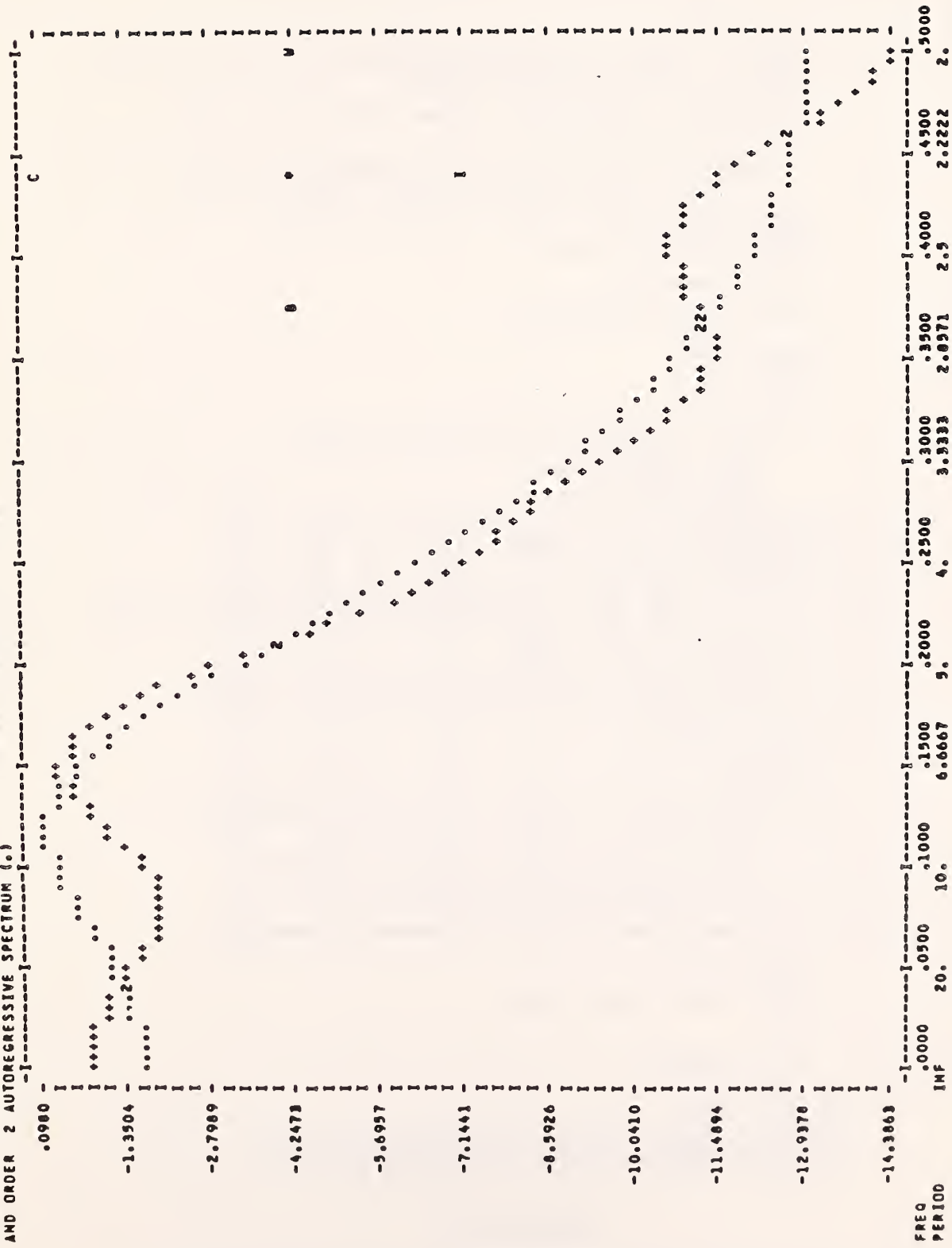


Figure F-4c

Example of UAS output (continued)

## Direct Fourier Transform of a Univariate Series and Utility Subroutines

```
MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE DIRECT FOURIER TRANSFORM SUBROUTINES USING SINGLE PRECISION
C VERSION OF STARPAC RUN ON CYBER 100/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, YFFT, PER, FREQ AND PERF MUST BE CHANGED TO DOUBLE
C PRECISION IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y(600), YFFT(600), PER(300), FREQ(300), PERF(300)
C INTEGER KND(10)
C DOUBLE PRECISION DSTAK(1000)
C
C COMMON /CSTAK/ DSTAK
C
C SPECIFY NECESSARY DIMENSIONS
C
C LDSTAK = 1000
C LPER = 600
C LFREQ = 300
C LYFFT = 600
C
C READ NUMBER OF MODIFIED DANIEL FILTERS TO BE APPLIED
C FILTER LENGTHS
C NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, NK
C READ 100, (KND(I), I = 1, NK)
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C CENTER THE SERIES AND APPLY A TEN PERCENT TAPER TO THE ENDS OF THE DATA
C
C TAPERP = 0.10
C CALL TAPER (Y, N, TAPERP, YFFT)
C
C PRINT TITLE AND CALL PGMS TO COMPUTE PERIODOGRAM OF TAPERED AND
C EXTENDED SERIES.
C
C WRITE (6, 102)
C NFFT = 514
C IEXTND = 0
C NPRT = (-1)
C CALL PGMS (YFFT, N, NFFT, LYFFT, IEXTND, NF, PER, LPER, FREQ,
C 1 LFREQ, NPRT)
C
C APPLY MODIFIED DANIELL FILTERS TO SMOOTH THE PERIODOGRAM
C
C CALL MDFLT (PER, NF, NK, KND, PERF, LDSTAK)
C
C PRINT TITLE AND CALL PPL TO DISPLAY SMOOTHED PERIODOGRAM
C
C WRITE (6, 103)
C ILOG = 1
C CALL PPL (PERF, FREQ, NF, ILOG)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (10I5)
C 101 FORMAT (10F7.2)
C 102 FORMAT ('RESULTS OF STARPAC PERIODOGRAM SUBROUTINE PGMS')
C 103 FORMAT ('RESULTS OF STARPAC FILTER SUBROUTINE MDFLT',
C * ' DISPLAYED USING STARPAC PLOT SUBROUTINE PPL')
C END
```

Figure F-5a

Example program using TAPER, PGMS, MDFLT and PPL

DATA:

```
      3
      8      8
261
 5.00 11.00 16.00 23.00 36.00 58.00 29.00 20.00 10.00 8.00
 3.00 0.00 0.00 2.00 11.00 27.00 47.00 63.00 60.00 39.00
28.00 26.00 22.00 11.00 21.00 40.00 78.00 122.00 103.00 73.00
47.00 35.00 11.00 5.00 16.00 34.00 70.00 81.00 111.00 101.00
73.00 40.00 20.00 16.00 5.00 11.00 22.00 40.00 60.00 80.90
83.40 47.70 47.80 30.70 12.20 9.60 10.20 32.40 47.60 54.00
62.90 85.90 61.20 45.10 36.40 20.90 11.40 37.80 69.80 106.10
100.80 81.60 66.50 34.80 30.60 7.00 19.80 92.50 154.40 125.90
84.80 68.10 38.50 22.80 10.20 24.10 82.90 132.00 130.90 118.10
89.90 66.60 60.00 46.90 41.00 21.30 16.00 6.40 4.10 6.80
14.50 34.00 45.00 43.10 47.50 42.20 28.10 10.10 5.10 2.50
 0.00 1.40 5.00 12.20 13.90 35.40 45.80 41.10 30.10 23.90
15.60 6.60 4.00 1.80 8.50 16.60 36.30 49.60 64.20 67.00
70.90 47.80 27.50 8.50 13.20 56.90 121.50 138.30 103.20 85.70
64.60 36.70 24.20 10.70 15.00 40.10 61.50 98.50 124.70 96.30
66.60 64.50 54.10 39.00 20.60 6.70 4.30 22.70 54.80 93.80
95.80 77.20 59.10 44.00 47.00 30.50 16.30 7.30 37.60 74.00
139.00 111.20 101.60 66.20 44.70 17.00 11.30 12.40 3.40 6.00
32.30 54.30 59.70 63.70 63.50 52.20 25.40 13.10 6.80 6.30
 7.10 35.60 73.00 85.10 78.00 64.00 41.80 26.20 26.70 12.10
 9.50 2.70 5.00 24.40 42.00 63.50 53.80 62.00 48.50 43.90
18.60 5.70 3.60 1.40 9.60 47.40 57.10 103.90 80.60 63.60
37.60 26.10 14.20 5.80 16.70 44.30 63.90 69.00 77.80 64.90
35.70 21.20 11.10 5.70 8.70 36.10 79.70 114.40 109.60 88.80
67.80 47.50 30.60 16.30 9.60 33.20 92.60 151.60 136.30 134.70
83.90 69.40 31.50 13.90 4.40 38.00 141.70 190.20 184.80 159.00
112.30
```

Figure F-5b

Example program using TAPER, PGMS, MDFLT and PPL (continued)

RESULTS OF STARPAC PERIODOGRAM SUBROUTINE PGMS

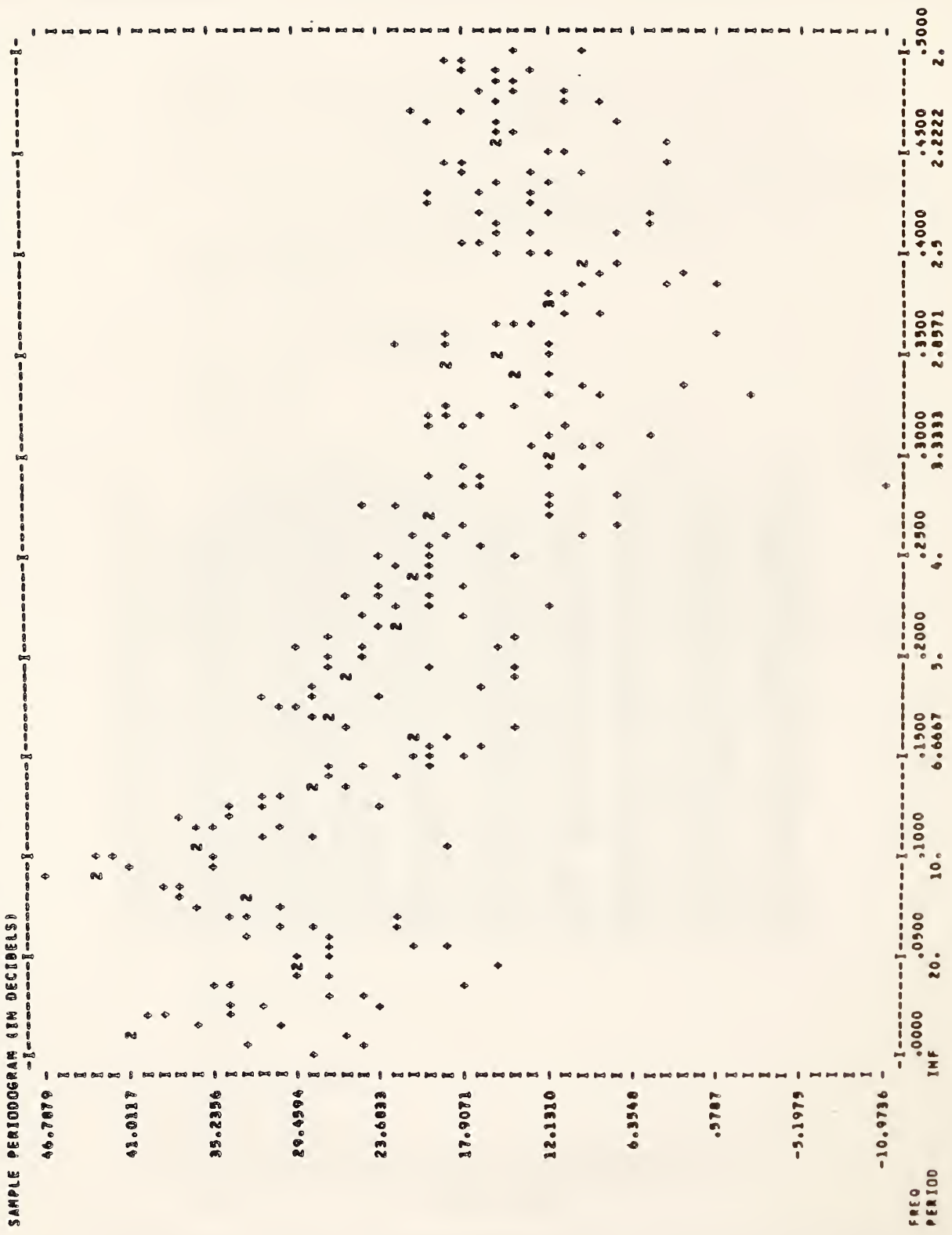


Figure F-5c

Example of PGMS output





## Bivariate Fourier Spectrum Analysis

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE BFS USING SIMLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y1 AND Y2 MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C REAL Y1(100), Y2(100)
C
C READ NUMBER OF OBSERVATIONS
C SERIES 1
C SERIES 2
C
C READ 100, N
C READ 101, (Y1(I), I=1,N)
C READ 101, (Y2(I), I=1,N)
C
C PRINT TITLE AND CALL BFS FOR BIVARIATE FOURIER SPECTRUM ANALYSIS
C
C WRITE (6, 102)
C CALL BFS (Y1, Y2, N)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (I9)
C 101 FORMAT (12F6.2)
C 102 FORMAT ('RESULTS OF STARPAC',
C * ' BIVARIATE FOURIER SPECTRUM ANALYSIS SUBROUTINE BFS')
C END
DATA:
100
-0.88 -0.16 -1.87 -1.12 1.38 2.13 2.76 0.56 -0.69 -1.79 -3.82 -2.38
1.00 0.70 -0.19 0.98 0.11 -0.39 -0.73 0.89 -1.63 -0.44 -1.37 -1.71
-1.22 -2.00 -0.22 0.38 1.31 0.71 0.32 0.48 -1.88 -0.94 -1.54 -0.13
1.02 0.02 -0.77 0.11 -0.60 -0.92 -0.09 1.23 1.46 0.61 0.42 2.16
3.18 2.10 0.37 -0.24 0.97 -0.53 2.44 1.02 -0.53 -2.49 -2.12 -1.04
-0.12 -1.88 -1.50 1.54 3.33 3.08 1.71 0.79 1.55 0.89 -0.89 -1.18
0.89 1.71 3.05 0.15 -1.04 0.12 0.08 0.11 -2.62 -1.28 1.07 3.20
1.92 0.93 -1.88 0.49 -0.98 0.17 1.19 -0.97 -1.68 1.14 -0.67 -0.88
-0.07 0.24 0.95 -2.16
0.79 1.12 -1.10 -2.39 -1.75 -0.82 -0.36 1.27 1.75 2.44 0.36 -2.10
-1.93 -1.30 -1.75 -0.34 0.74 0.49 0.70 0.71 0.09 0.59 1.54 0.14
0.55 -1.40 -2.55 -1.66 -0.43 0.58 2.18 -0.24 0.58 -0.18 -1.55 -0.64
-1.09 0.90 -0.66 -0.95 0.48 0.50 0.05 -0.68 0.24 0.58 -1.26 -0.25
0.25 2.18 2.96 1.56 -0.36 -0.59 -0.12 3.03 2.11 0.78 0.89 -1.45
-0.36 -0.37 -1.39 -4.19 -0.73 -0.98 0.36 0.06 -1.94 -0.08 0.17 1.00
-0.05 0.43 0.15 2.69 0.57 0.29 1.10 0.48 -1.06 -2.28 -2.03 -0.75
1.00 1.71 0.58 1.97 0.99 1.94 2.18 3.14 0.60 0.91 1.35 0.56
0.11 0.00 2.34 1.88

```

Figure F-6a

Example program using BFS

RESULTS OF STARPAC BIVARIATE FOURIER SPECTRUM ANALYSIS SUBROUTINE BFS  
 --- SMOOTHED FOURIER SPECTRUM (SQUARED COHERENCY COMPONENT) (+), 95 PCT. CONFIDENCE LIMITS (.) AND 95 PCT. SIGNIFICANCE LEVEL (-) ---  
 (PARZEN WINDOW WITH LAG WIND. TRUNC. PT. 4 / BW .4637 / EDF 93)

STARPAC 2.005 - DECEMBER 2, 1985

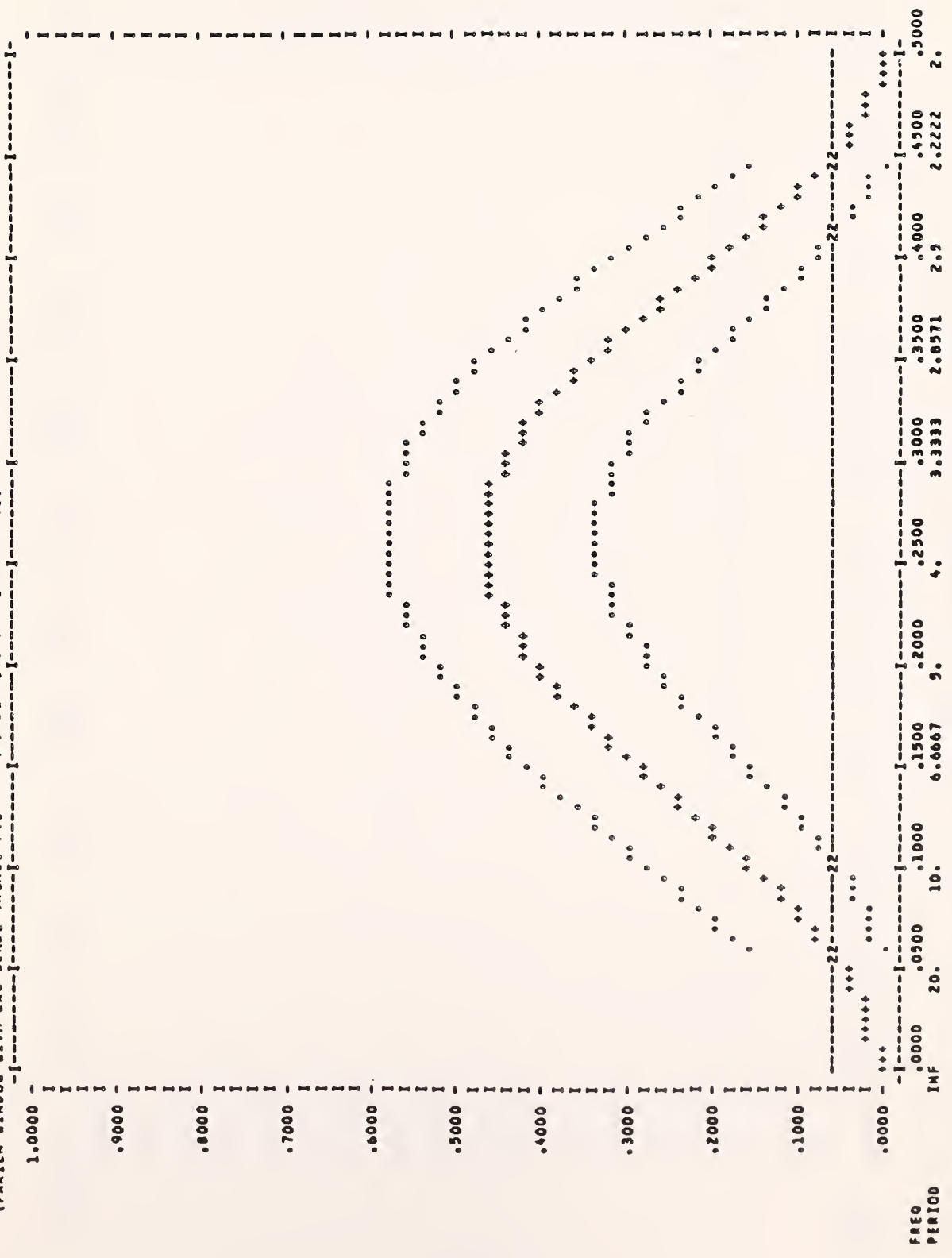


Figure F-6b

Example of BFS output

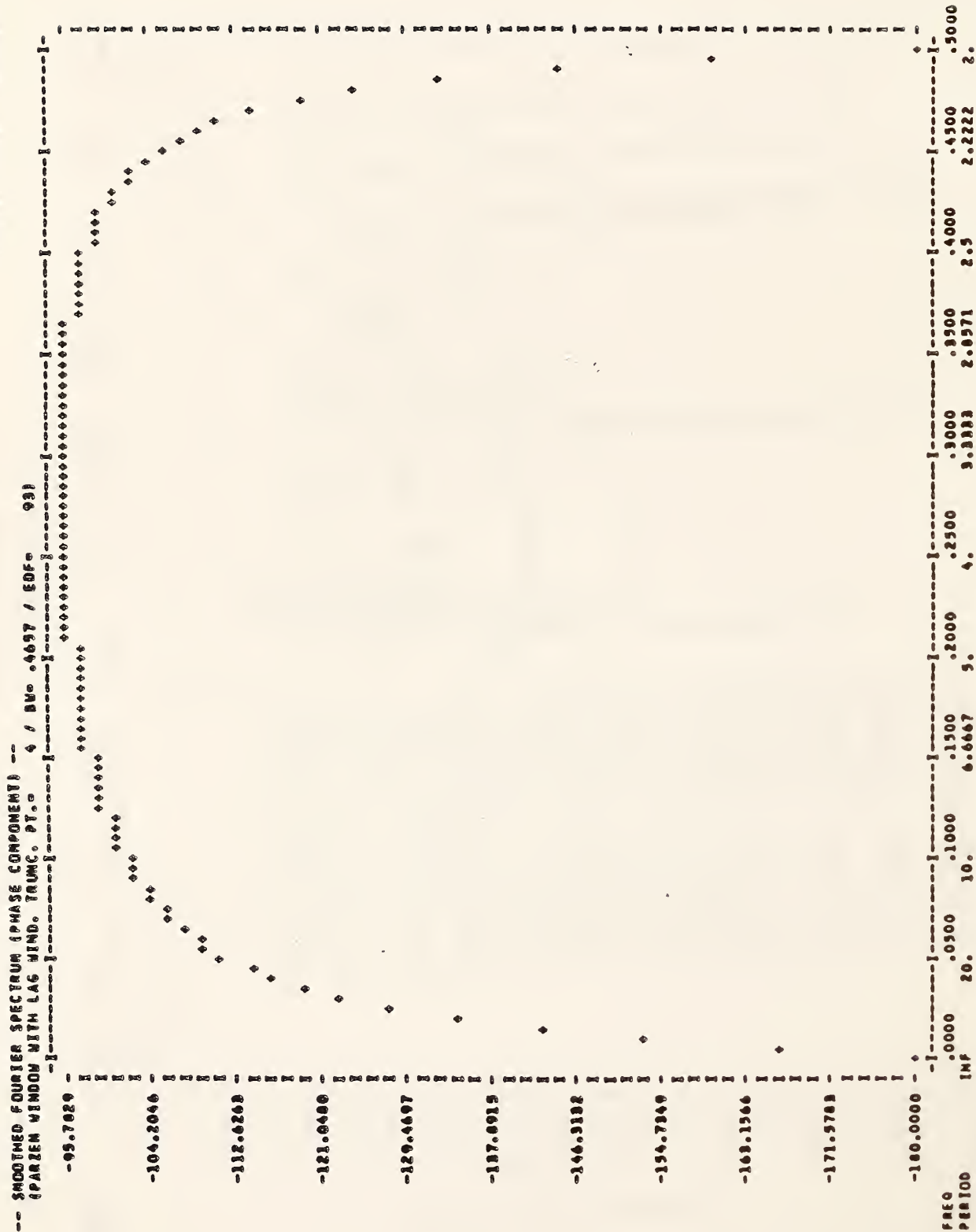


Figure F-6c

Example of BFS output (continued)



STARPAC 2.005 - DECEMBER 20, 1985  
 -- SMOOTHED FOURIER SPECTRUM (SQUARED COHERENCY COMPONENT) (+), 95 PCT. CONFIDENCE LIMITS (.) AND 95 PCT. SIGNIFICANCE LEVEL (-) --  
 (PARZEN WINDOW WITH LAG WIND. TRUNC. PT. 0 / BW = .2349 / EDF = 47)

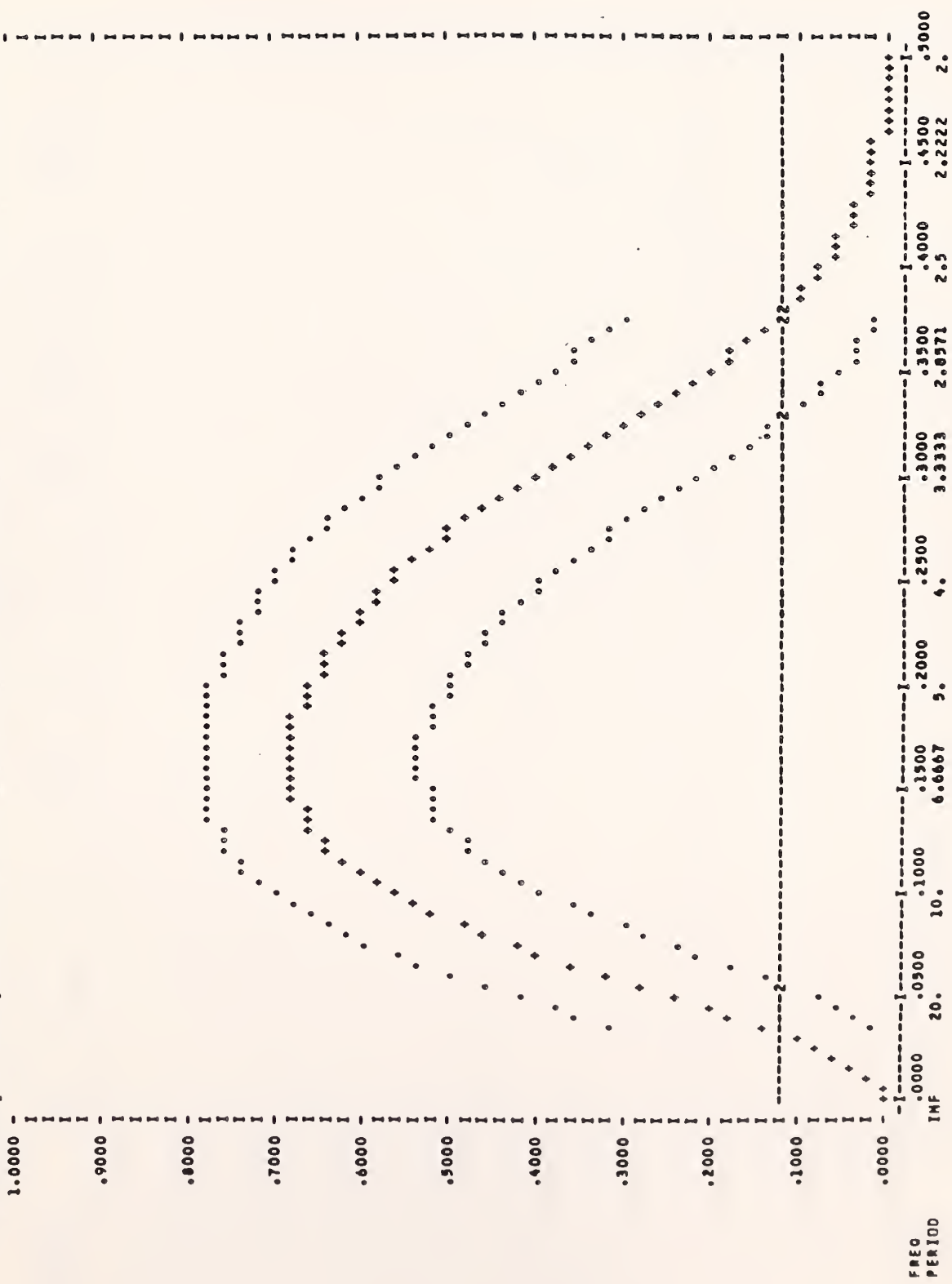


Figure F-6d

Example of BFS output (continued)



STARPAC 2-005 - DECEMBER 2, 1965  
 -- SMOOTHED FOURIER SPECTRUM (SQUARED COHERENCY COMPONENT) (+), 95 PCT. CONFIDENCE LIMITS (.) AND 95 PCT. SIGNIFICANCE LEVEL (-) --  
 (PARZEN WINDOW WITH LAG WIND. TRUNC. PT. = 16 / BW = .1191 / EDF = 24)

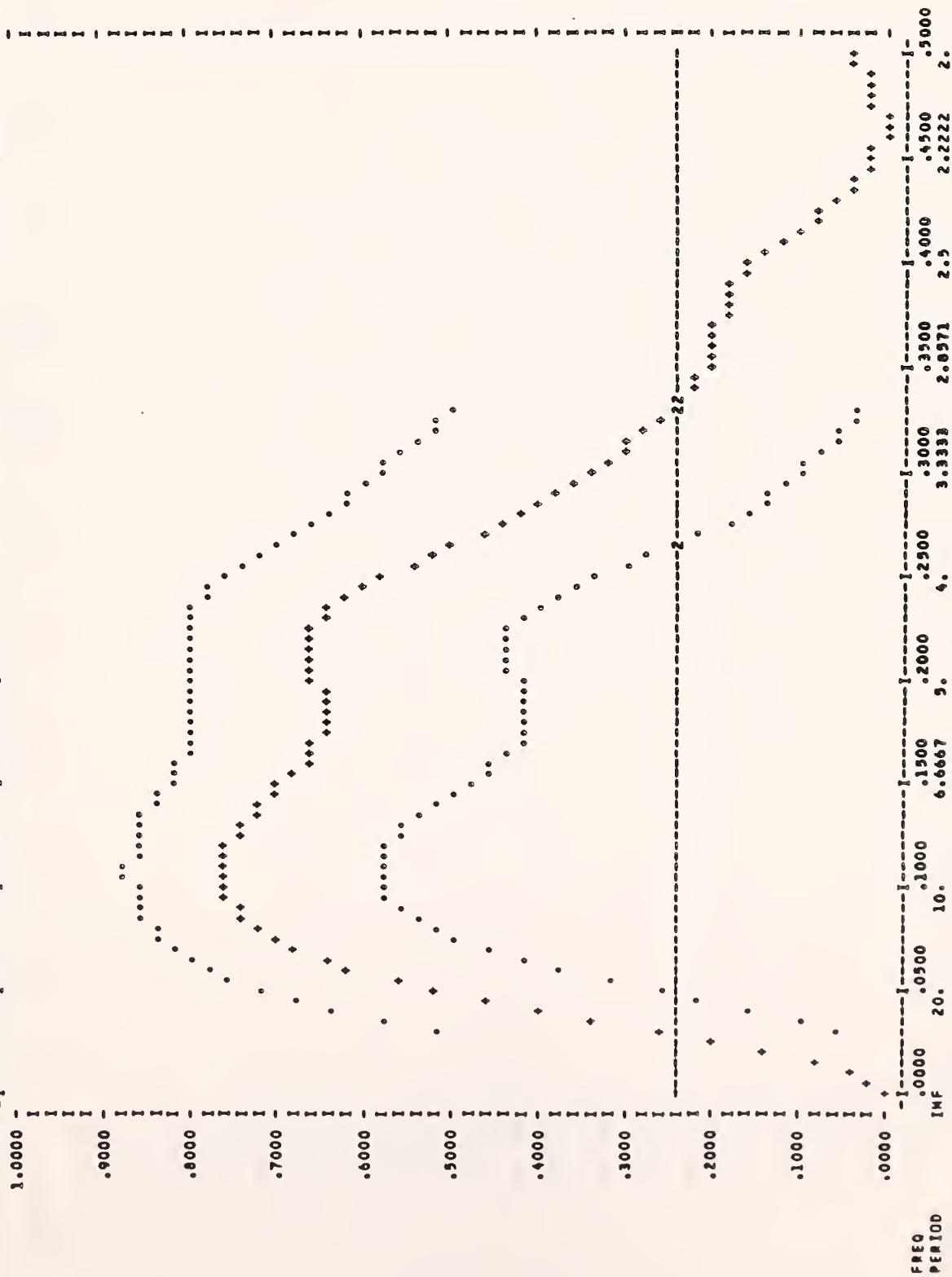


Figure F-6f

Example of BFS output

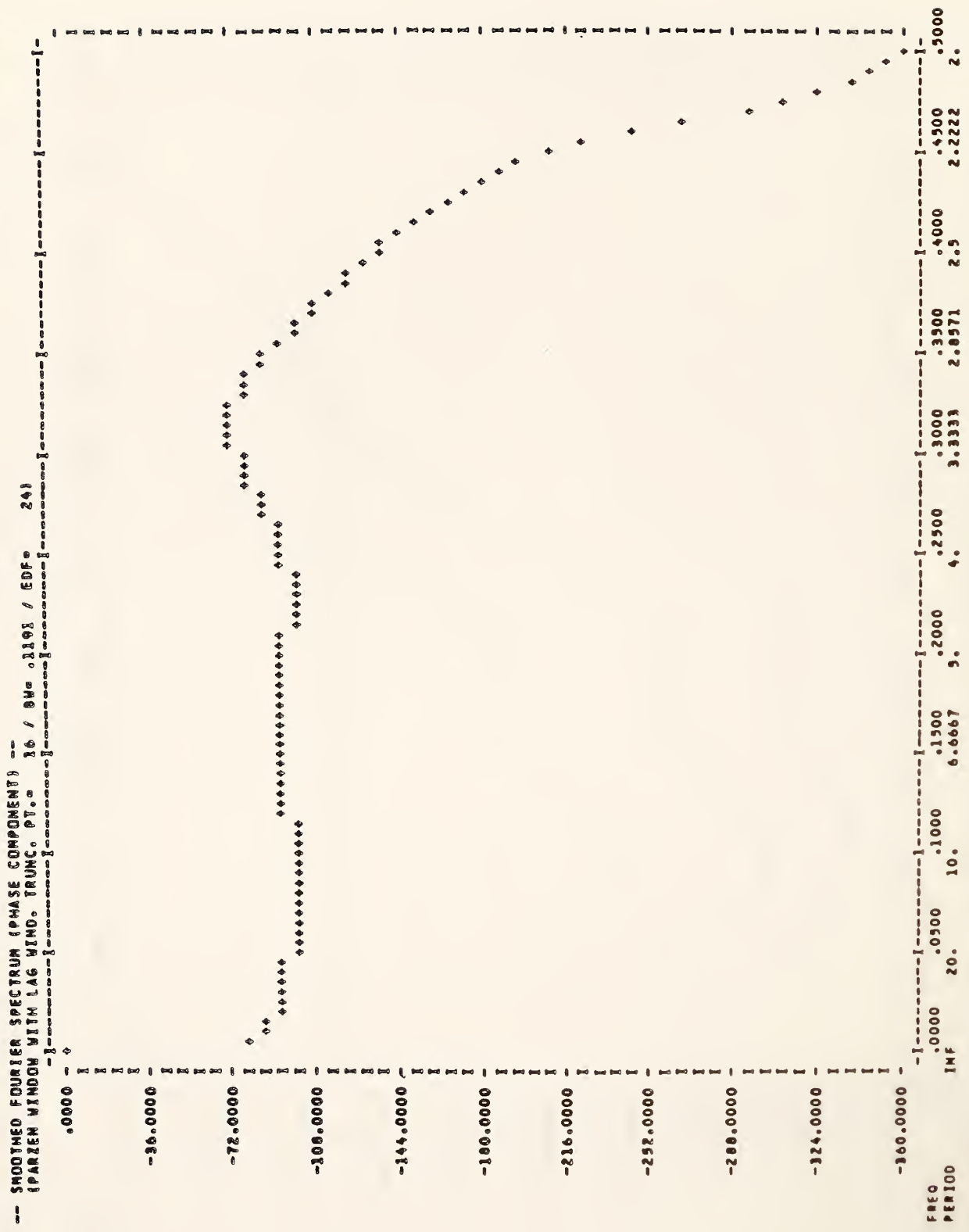


Figure F-6g

Example of BFS output (continued)



--- SMOOTHED FOURIER SPECTRUM (SQUARED COHERENCY COMPONENT) (♦), 95 PCT. CONFIDENCE LIMITS (.) AND 95 PCT. SIGNIFICANCE LEVEL (-) ---  
 (PARZEN WINDOW WITH LAG WIND. TRUNC. PT. = 33 / BW = .0595 / EDF = 32)

STARPAC 2.005 - DECEMBER 2, 1985

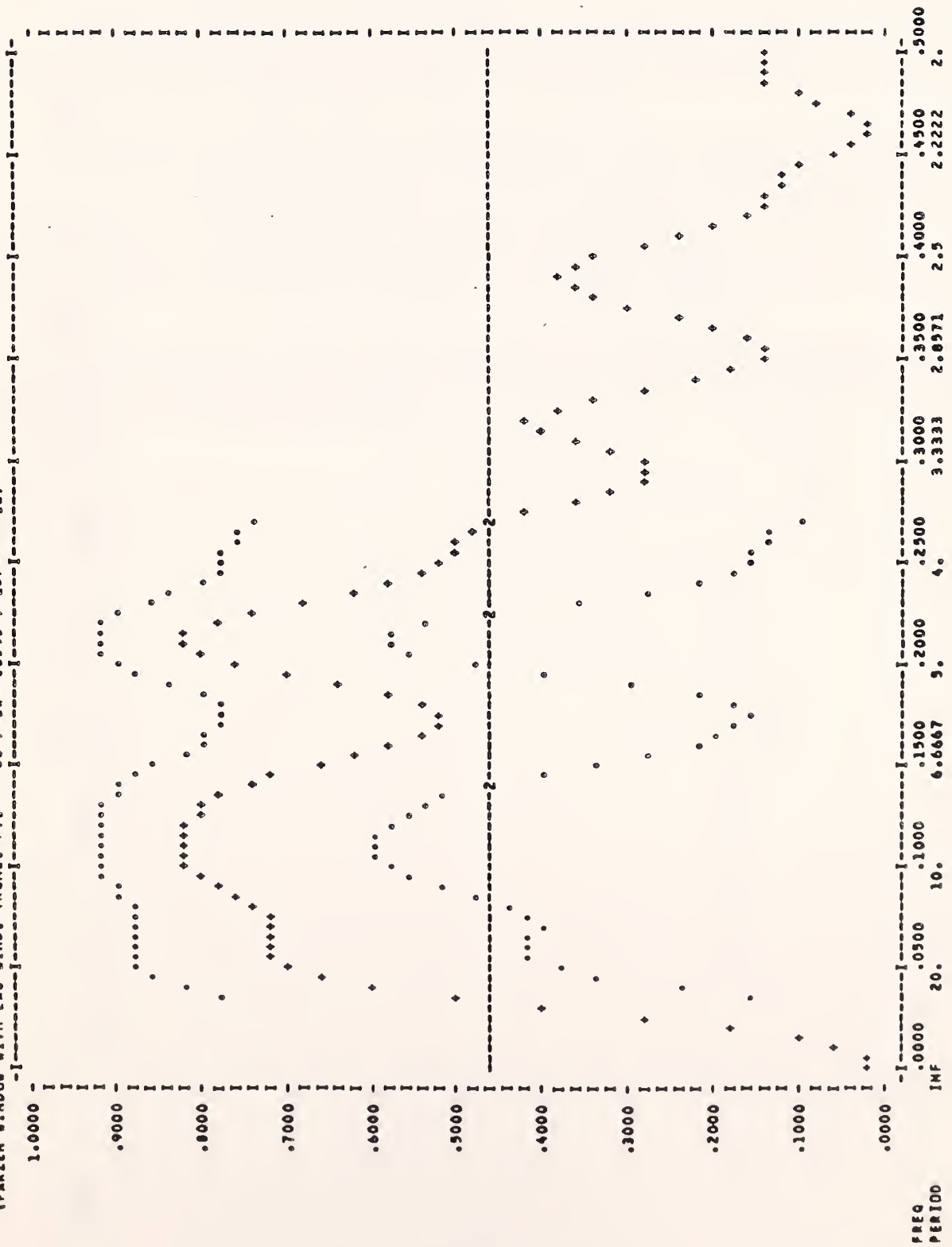


Figure F-6h

Example of BFS output (continued)

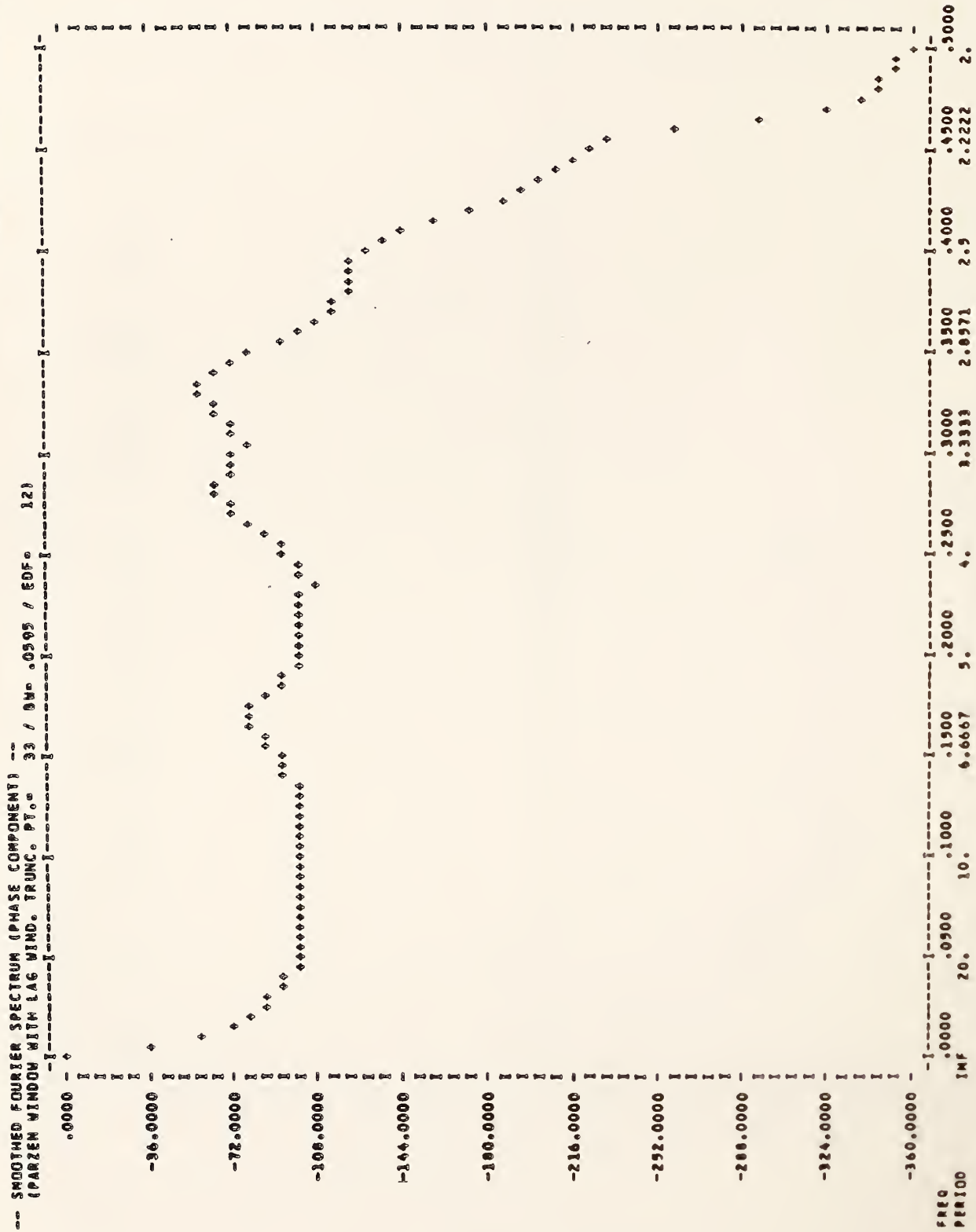


Figure F-61

Example of BFS output (continued)

## G. Acknowledgments

The code for computing the autocovariance and cross covariance functions using a fast Fourier transform are based on subroutines written by Jones [1971]. The subroutines which compute the fast Fourier transform and which compute the Fourier transform of real data are those written by Singleton [1969] and the subroutine to compute the cosine transform was written by Jones.

The transformation used to compute the univariate Fourier spectrum is performed using the algorithm shown on page 311 of Jenkins and Watts [1968]. The algorithm used is described on pages 418-420.

The code for computing the autoregressive order selection statistics and autoregressive spectrum estimates is based on subroutines written by Jones [1971]. The coefficients  $\text{PHI}(k)$ ,  $k = 1, \dots, \text{IAR}$  of the autoregressive model used by the autoregressive spectrum subroutines are computed from the autocovariance function using the Levinson-Durbin recursive method for solving the Yule-Walker equations discussed in Appendix A3.2 of Box and Jenkins [1976].

The subroutines for the split-cosine-bell taper and the modified Daniell filter operation were adapted from subroutines TAPER and MODDAN given on pages 116 and 178 of Bloomfield [1976].





## ARIMA MODELING

A. Introduction

STARPAC contains five user-callable subroutines for AutoRegressive Integrated Moving Average (ARIMA) modeling. Three are for computing the least squares estimates of the parameters of an ARIMA model and two are for computing the minimum mean square error forecasts using these estimates. Both the estimation and forecasting subroutines allow several levels of control of the computations and printed output. The estimation subroutines also 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 respecifying the model for each subset.

Each of the subroutines in this chapter models the input series,  $y_i$ ,  $i = 1, \dots, N$ , with a user-specified general multiplicative ARIMA model using the techniques discussed in Box and Jenkins [1976]. Briefly, this model is defined by

$$\left[ \prod_{j=1}^{\text{NFAC}} (\phi_{p(j)}(B^{s(j)})) \right] \left[ \left( \prod_{j=1}^{\text{NFAC}} (\nabla_{s(j)})^{d(j)} \right) y_i - \mu \right] = \left[ \prod_{j=1}^{\text{NFAC}} (\theta_{q(j)}(B^{s(j)})) \right] a_i$$

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

$N$  is the number of observations in the series;

$\text{NFAC}$  is the number of factors in the model;

$B^s$  is the backward shift operator of order  $s$ , i.e.,

$$B^s y_i = y_{i-s};$$

$\mu$  is the expected value of the differenced series, i.e.,

$$\mu = E \left[ \prod_{j=1}^{\text{NFAC}} (\nabla_{s(j)})^{d(j)} y_i \right],$$

which can be used to allow for a deterministic polynomial trend;

$\phi_{p(j)}(B^{s(j)})$  is the polynomial in  $B^{s(j)}$  of order  $p(j)$ , i.e.,

$$\phi_{p(j)}(B^{s(j)}) =$$

$$1 - \phi_{1,j} B^{s(j)} - \phi_{2,j} B^{2 \cdot s(j)} - \dots - \phi_{p(j),j} B^{p(j) \cdot s(j)},$$

which represents the  $j^{\text{th}}$  autoregressive factor in the model;

$(\nabla_{s(j)})^{d(j)}$  is the backward difference operator, i.e.,

$$\begin{aligned}(\nabla_{s(j)})^{d(j)} &= (1 - B^{s(j)})^{d(j)} \\ &= 1 - d(j)B^{s(j)} + \dots B^{d(j) \cdot s(j)},\end{aligned}$$

which represents the  $j^{\text{th}}$  difference factor in the model;

$\theta_{q(j)}(B^{s(j)})$  is the polynomial in  $B^{s(j)}$  of order  $q(j)$ , i.e.,

$$\begin{aligned}\theta_{q(j)}(B^{s(j)}) &= \\ &= 1 - \theta_{1,j}B^{s(j)} - \theta_{2,j}B^{2 \cdot s(j)} - \dots - \theta_{q(j),j}B^{q(j) \cdot s(j)},\end{aligned}$$

which represents the  $j^{\text{th}}$  moving average factor in the model; and

$a_i$  is the unobservable random noise component at the  $i^{\text{th}}$  observation.

The least squares estimates of the parameters,

$$\phi_{1,1}, \phi_{2,1}, \dots, \phi_{p(1),1}, \phi_{1,2}, \dots, \phi_{p(\text{NFAC}),\text{NFAC}}, \mu,$$

$$\theta_{1,1}, \theta_{2,1}, \dots, \theta_{q(1),1}, \theta_{1,2}, \dots, \theta_{q(\text{NFAC}),\text{NFAC}}$$

are obtained using back forecasts at each iteration as discussed in §E.1.a. The least squares solution is that which minimizes (with respect to the parameters) the sum of the squares of the random noise components,  $a_i$ , i.e.,

$$\sum_{i=-\infty}^N \hat{a}_i^2$$

where carat (^) denotes the estimated quantity. The iterative procedure used is documented in chapter 9.

The user must supply both initial values for the parameters and an array specifying the orders  $p(j)$ ,  $d(j)$ ,  $q(j)$  and  $s(j)$  for each factor  $j = 1, \dots, \text{NFAC}$  in the model. Initial parameter values for the estimation subroutines should be chosen with care since good values can significantly reduce computer time.

Users are directed to §B for a brief description of the subroutines. The declaration and CALL statements are given in §C and the subroutine arguments are defined in §D. The algorithms used and output produced by these subroutines are discussed in §E. Sample programs and their output are shown in §F.

## B. Subroutine Descriptions

### B.1 ARIMA Estimation Subroutines

The simplest of the three ARIMA estimation subroutines, AIME, automatically summarizes the estimated results and a variety of statistics in a five-part printed report described in §E.2.a, and returns the estimated parameters and residuals to the user via the subroutine argument list (level one control). Most ARIMA estimation problems can be solved using AIME.

The other two estimation subroutines, AIMEC and AIMES, provide greater flexibility to the user at the price of more input.

AIMEC, like AIME, also returns estimated parameters and residuals from the fit. In addition, it allows the user to supply arguments to indicate

- a subset of the model parameters to be treated as constants, with their values held fixed at their input values;
- the step sizes used to compute the numerical approximations to the derivative as described in chapter 9, §E.1.b;
- 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
- what sections of the five-part printed report are wanted.

AIMES has all the features of AIMEC and, in addition, returns the following estimated values via the argument list:

- 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 ARIMA Forecasting Subroutines

The simplest of the three ARIMA forecasting subroutines, AIMF, automatically summarizes the estimated results in a two-part printed report. Forecasts are made using N as the origin and extending  $[N/10] + 1$  steps into the future, i.e., observations are forecast for indices N+1, N+2, ..., N+[N/10]+1. Many forecasting problems can be solved using AIMF.

The second forecasting subroutine, AIMFS, allows the user to supply arguments to indicate the number of forecasts to be made, the origins to be

used and the amount of printed output. This subroutine also returns the forecasts and their standard deviations via the argument list.

### C. Subroutine Declaration and CALL Statements

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

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

```
<real> Y(n), PAR(npar)
INTEGER MSPEC (4,npfac)
DOUBLE PRECISION DSTAK(ldstak)
COMMON /CSTAK/ DSTAK
```

#### Subroutine for ARIMA Estimation

*AIME: Compute and print a five-part least squares analysis of the parameter estimates of an ARIMA model; return parameter estimates and residuals*

```
<basic declaration block>
<real> RES(n)
:
CALL AIME (Y, N, MSPEC, NFAC, PAR, NPAR, RES, LDSTAK)
```

===

*AIMEC: Compute and optionally print a five-part least squares analysis of the parameter estimates of an ARIMA model using user-supplied control values; return parameter estimates and residuals*

```
<basic declaration block>
<real> RES(n)
INTEGER IFIXED(npar)
<real> STP(npar), STOPSS, STOPP, SCALE(npar), DELTA
:
CALL AIMEC (Y, N, MSPEC, NFAC, PAR, NPAR, RES, LDSTAK,
1 IFIXED, STP, MIT, STOPSS, STOPP, SCALE, DELTA, IVAPRX, NPRT)
```

===



*AIMES: Compute and optionally print a five-part least squares analysis of the parameter estimates of an ARIMA model 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>
<real> RES(n)
INTEGER IFIXED(npars)
<real> STP(npars), STOPSS, STOPP, SCALE(npars), DELTA
<real> RSD, PV(n), SDPV(n), SDRES(n), VCV(npars,npars)
:
CALL AIMES (Y, N, M, MSPEC, NFAR, PAR, NPAR, RES, LDSTAK,
1  IFIXED, STP, MIT, STOPSS, STOPP, SCALE, DELTA, IVAPRX, NPRT,
2  NPARE, RSD, PV, SDPV, SDRES, VCV, IVCV)

```

#### Subroutines for ARIMA Forecasting

*AIMF: Compute and print the minimum mean square error forecasts obtained using an ARIMA model*

```

<basic declaration block>
:
CALL AIMF (Y, N, MSPEC, NFAC, PAR, NPAR, LDSTAK)

```

===

*AIMFS: Compute and optionally print the minimum mean square error forecasts obtained using an ARIMA model; return forecasts and their standard errors*

```

<basic declaration block>
<real> FCST(nfcst,nfcsto), SDFCST(nfcst)
INTEGER IFCSTO(nfcsto)
:
CALL AIMFS (Y, N, MSPEC, NFAC, PAR, NPAR, LDSTAK,
1  NFCST, NFCSTO, IFCSTO, NPRT, FCST, IFCST, FCSTSD)

```

#### 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.

DELTA --> The maximum scaled change allowed in the parameters at the first iteration, i.e.,  $\delta_0$ . [See chapter 9, §E.1.a.] The default value is 100.0. When  $\text{DELTA} \leq 0.0$  or when DELTA is not an argument of the subroutine CALL statement the default value is used. Normally, the default value should be used. However a smaller value of DELTA may be appropriate if, at the first iteration, the computation of the predicted values 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,

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

where  $\Delta_{\max}^{\text{PAR}(k)}$  is the maximum change allowed for the  $k^{\text{th}}$  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.

FCST <-- The array of dimension at least NFCST by NFCSTO that contains the NFCST forecasts computed from each of the NFCSTO origins.

FCSTSD <-- The vector of dimension at least NFCST that contains the standard deviation of each of the forecasts,

$$\text{FCSTSD}(K) = \text{RSD} \cdot \left( \sum_{j=0}^{K-1} \psi_j^2 \right)^{1/2} \text{ for } K = 1, \dots, \text{NFCST},$$

where

$$\psi_0 = 1 \text{ and } \psi_j = \psi_1 \cdot \psi_{j-1} + \dots + \psi_{P+D} \cdot \psi_{j-P-D} - \theta_j$$

$$\text{with } P = \sum_{j=1}^{\text{NFAC}} p(j); \quad D = \sum_{j=1}^{\text{NFAC}} d(j); \text{ and}$$

$\psi_i$  = the coefficient of  $B^i$  in the polynomial defined by

$$\prod_{j=1}^{\text{NFAC}} (\phi_p(j)(B^{s(j)})) \cdot ((\nabla_s(j))^{dj})$$

IERR ... An error flag returned in COMMON /ERRCHK/. [See chapter 1, §D.5.] Note that using (or not using) the error flag will not affect the printed error messages that are automatically provided even when the user has suppressed the normal printed output.

For AIME, AIMEC, and AIMES:

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 identify 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 variance-covariance matrix is questionable.

IERR = 5 indicates false convergence. [See chapter 9, §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 because of computational difficulties.

For AIMF and AIMFS:

IERR = 0 indicates that no errors were detected and that all the forecasts were computed.

IERR = 1 indicates that improper input was detected.

IFCST --> The exact value of the first dimension of the matrix FSCT as specified in the calling subroutine.

IFCSTO --> The vector of dimension at least NFCSTO that contains the NFCSTO indices to be used as origins, where  $1 < \text{IFCSTO}(J) < N$  for  $J = 1, \dots, \text{NFCSTO}$ . The default value for each element of IFCSTO is N. When IFCSTO(J) is outside the range [1, N] or IFCSTO is not an argument of the subroutine CALL statement the default value is used.

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 §A. The default values are 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:

$$(1) \text{ VCV} = \text{RSD}^2 \cdot (\hat{D}^T \cdot \hat{D})^{-1}$$

$$(2) \text{ VCV} = \text{RSD}^2 \cdot H^{-1}$$

$$(3) \text{ VCV} = \text{RSD}^2 \cdot H^{-1} \cdot (\hat{D}^T \cdot \hat{D}) \cdot H^{-1}$$

where

$$D_{ij} = \hat{a}_i / \partial \beta(j) \quad \begin{array}{l} \text{for } i = 1, \dots, N \\ \text{and } j = 1, \dots, \text{NPAR}; \text{ and} \end{array}$$

$$H = \hat{D}^T \cdot \hat{D} - \left\{ \sum_{i=1}^N \hat{a}_i \cdot (\partial^2 \hat{a}_i / \partial \beta(j) \partial \beta(k)) \right\} \quad \begin{array}{l} \text{for } j = 1, \dots, \text{NPAR} \\ \text{and } k = 1, \dots, \text{NPAR} \end{array}$$

The results of a recent study by Donaldson and Schnabel [1985] 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. 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, 1985].

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.

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 is used.



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, §B.] Also,

$$MBO = \text{maximum} \left\{ \sum_{j=1}^{NFAC} (p(j)+d(j)) \cdot s(j), \sum_{j=1}^{NFAC} q(j) \cdot s(j) \right\}.$$

For AIME, AIMEC and AIMES:

$$LDSTAK \geq 43 + \max\{IS \cdot (N+NPARE), 30+NPARE\} + 2 \cdot NFAC + \max \left\{ \begin{array}{l} IS \cdot 10 \cdot N + 6 \cdot MBO + 606. \\ 94 + 4 \cdot (N+MBO+101) + 5 \cdot MBO + (3 \cdot NPARE^2 + 35 \cdot NPARE) / 2 \end{array} \right\} \cdot P$$

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

For AIMF and AIMFS:

$$LDSTAK \geq 18 + 2 \cdot NFAC + (5 \cdot MBO + 2 \cdot (N+MBO+101)) \cdot P$$

MIT --> The maximum number of iterations allowed. This argument is also used to compute the maximum number of model subroutine calls, ( $2 \cdot MIT$ ). 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 is used.

MSPEC --> The array of dimension exactly 4 rows by at least NFAC columns that contains the orders  $p$ ,  $d$ ,  $q$  and  $s$  for each factor in the model where  $p(j)$ ,  $j = 1, \dots, NFAC$  must be in row 1,  
 $d(j)$ ,  $j = 1, \dots, NFAC$  must be in row 2,  
 $q(j)$ ,  $j = 1, \dots, NFAC$  must be in row 3, and  
 $s(j)$ ,  $j = 1, \dots, NFAC$  must be in row 4.  
 Values of  $p(j)$ ,  $d(j)$ ,  $q(j)$  and  $s(j)$ ,  $j = 1, \dots, NFAC$ , must each equal or exceed zero.

N --> The number of observations.

NFAC --> The number of factors in the model.

NFCST --> The number of forecasts to be computed. The default value is  $[N/10]+1$ . When  $NFCST < 0$  or when NFCST is not an argument of the subroutine CALL statement the default value is used.

NFCSTO --> The number of forecast origins supplied. The default value is 1. When  $NFCSTO < 0$  or when NFCSTO is not an argument of the subroutine CALL statement the default value is used.

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

$$NPART = 1 + \sum_{j=1}^{NFAC} (p(j) + q(j)) .$$

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 AIME, AIMEC and AIMES:

NPRT is a five-digit integer for which the value of the I<sup>th</sup> digit (counting from left to right) is used to control the I<sup>th</sup> section of the output.

If the I<sup>th</sup> digit = 0 the output from the I<sup>th</sup> section is suppressed;  
= 1 the brief form of the I<sup>th</sup> section is given;  
> 2 the full form of the I<sup>th</sup> 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 is 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 §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.

-- continued --

For AIMF and AIMFS:

If NPRT = 0 the printed output is suppressed.

If NPRT ≠ 0 the printed output is provided.

The default value for NPRT is 1. When NPRT ≤ -1 or when NPRT is not an argument in the subroutine CALL statement the default value will be used.

PAR --- The vector of dimension at least NPAR that contains the parameter values. For both the estimation and the forecasting subroutines, parameter values must be ordered

$\phi_{1,1}, \phi_{2,1}, \dots, \phi_{p(1),1}, \phi_{1,2}, \dots, \phi_{p(\text{NFAC}),\text{NFAC}},$

$\mu,$

$\theta_{1,1}, \theta_{2,1}, \dots, \theta_{q(1),1}, \theta_{1,2}, \dots, \theta_{q(\text{NFAC}),\text{NFAC}},$

i.e., the parameter values from the autoregressive factors are first, followed by  $\mu$ , followed by the parameter values from the moving average factors. For all estimation subroutines PAR must contain initial values for the parameters on input and will contain the final values on return. For the forecasting subroutines PAR must contain the parameter values for which the forecasts are to be computed.

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

$PV(i) = y_i - \hat{a}_i \quad \text{for } i = 1, \dots, N.$

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

$RES(i) = \hat{a}(i) \quad \text{for } i = 1, \dots, N.$

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

$$RSD = \left[ \left( \sum_{i=1}^N RES(i)^2 \right) / (N - NPARDF - NPARE) \right]^{1/2}$$

$$\text{where } NPARDF = \sum_{j=1}^{\text{NFAC}} s(j) \cdot d(j) .$$

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

$$|\text{PAR}_\ell(j)/\text{SCALE}(j)| \approx |\text{PAR}_\ell(k)/\text{SCALE}(k)| \quad \text{for } j = 1, \dots, \text{NPAR} \\ \text{and } k = 1, \dots, \text{NPAR},$$

where  $\text{PAR}_\ell(j)$  is the value of the  $j^{\text{th}}$  parameter at the  $\ell^{\text{th}}$  iteration. Values of  $|\text{SCALE}(k)| > |\text{PAR}_\ell(k)|$  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  $\text{PAR}_\ell(k)$ , although normally the default values should be used.

The default values for SCALE are selected by the NL2SOL algorithm [Dennis et al. 1981a and 1981b] and are updated at each iteration. When SCALE is not an argument in the subroutine CALL statement or when the user-supplied value for  $\text{SCALE}(1) < 0$  the default procedure will be used to select scale values. When  $\text{SCALE}(1) > 0$ , values of  $\text{SCALE}(k) < 0$  for  $k = 2, \dots, \text{NPAR}$  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.

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

$$\text{SDPV}(i) = \text{the } i^{\text{th}} \text{ diagonal element of } [(\hat{D} \cdot \text{VCV} \cdot \hat{D}^T)]^{1/2}$$

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

$$D_{ij} = \partial a_i / \partial \beta(j) \quad \text{for } i = 1, \dots, N \text{ and } j = 1, \dots, \text{NPAR}.$$

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,

$$\text{SDRES}(i) = \text{RES}(i) / [\text{RSD}^2 - \text{SDPV}(i)^2]^{1/2}$$

for  $i = 1, \dots, N$ , which is the  $i^{\text{th}}$  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

$$\frac{\max\{ |PAR_{\ell+1}(k) - PAR_{\ell}(k)| / SCALE(k) \text{ for } k = 1, \dots, NPAR\}}{\max\{ (|PAR_{\ell+1}(k)| + |PAR_{\ell}(k)|) / SCALE(k) \text{ for } k = 1, \dots, NPAR\}} < STOPP$$

where  $PAR_{\ell}(k)$  is the value of the  $k^{\text{th}}$  parameter at the  $\ell^{\text{th}}$  iteration. [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

$$\max\{ |PAR_{\ell+1}(k) - PAR_{\ell}(k)| / |PAR_{\ell}(k)| \text{ for } k = 1, \dots, NPAR\}.$$

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 is used.

STOPSS --> The stopping value for the convergence test based on the ratio of the forecasted change in the residual sum of squares at iteration  $\ell+1$ ,  $\Delta RSS_{\ell+1}$ , to the residual sum of squares at iteration  $\ell$ ,

$$RSS_{\ell} = \sum_{i=-\infty}^N \hat{a}_i^2.$$

The convergence criterion is satisfied if certain conditions are met and

$$\Delta RSS_{\ell+1} / RSS_{\ell} < STOPSS.$$

[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  $\ell$  and  $\ell+1$  as measured by  $(RSS_{\ell}^{1/2} - RSS_{\ell+1}^{1/2}) / RSS_{\ell}^{1/2}$ . 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.

-- continued --

The default value is approximately the maximum of  $10^{-10}$  and  $10^{-2 \cdot \text{DIGITS}/3}$ , 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 by the estimation subroutines to approximate the derivative matrix numerically. The procedure used to select the default values is described in chapter 9, §E.1.a. When STP is not an argument of the subroutine CALL statement or when  $\text{STP}(1) < 0$  the default values will be used for all of the step sizes; when  $\text{STP}(1) > 0$ , values of  $\text{STP}(k) < 0$  for  $k = 2, \dots, \text{NPAR}$  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 (as specified by 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.

Y       --> The vector of dimension at least N that contains the series being modeled.

## E. Computational Methods

### E.1 Algorithms

#### E.1.a ARIMA Estimation

The ARIMA estimation subroutines use the NL2SOL software package written by Dennis et al. [1981a and 1981b]. The observations of the series, which are measured with error, are iteratively fit to the ARIMA model by minimizing the sums of squares of the estimated random noise component as described in §A. The back forecasting technique discussed on pages 215-220 of Box and Jenkins [1976] is used to compute the random noise component. Up to 101 back forecasts are computed. The back forecasts are assumed to be negligible when their magnitude is less than 0.01 times the first value of the differenced series (centered about its mean) obtained entirely from the observed data. If, at the last iteration, the 101<sup>st</sup> back forecast is not negligible a warning message is printed.

The iterations continue until the convergence criteria based on the change in the parameter values or in the residual sum of squares are satisfied [arguments STOPP and STOPSS], the maximum number of iterations (or model subroutine calls) is reached [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, §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 a discontinuity in the derivative. 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 discussed in chapter 9 and Dennis et al. [1981a].

### E.1.b ARIMA Forecasting

The ARIMA forecasting subroutines use the techniques discussed in Box and Jenkins [1976], chapter 5. The back forecasting technique discussed on pages 215-220 of Box and Jenkins [1976] is used to compute the random noise component needed for the forecasts. Values of  $a_i$  for  $i$  greater than the forecast origin are assumed to be zero. Up to 101 back forecasts are computed. The back forecasts are assumed to be negligible when their magnitude is less than 0.01 times the first value of the differenced series (centered about its mean) obtained entirely from the observed data. If the 101<sup>st</sup> back forecast is not negligible a warning message is printed.

## E.2 Computed Results and Printed Output

### E.2.a The ARIMA Estimation Subroutines

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

The output from the ARIMA 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 underlined, uppercase phrases. Results which correspond to input or returned subroutine CALL statement arguments are identified by the argument name in uppercase (not underlined).

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 indicated by argument IFIXED.
- The scale values, SCALE.



- The step sizes used to approximate the derivatives numerically, STP.
- The number of observations, N.
- 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 computed using the starting parameter values, RSD.

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 resulting from 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.
- SERIES: the value of the dependent variable,  $Y$ .
- PREDICTED VALUE: the estimated predicted value,  $PV$ , from the fit.
- STD DEV OF PRED VALUE: the standard deviation of the predicted value,  $SDFV$ .
- RESIDUAL: the error estimate,  $RES$ .
- STD RES: the standardized residual,  $SDRES$ .

Section 4 displays the following plots.

- The standardized residuals versus row numbers.
- 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,

$$r_{jk} = VCV(j,k) / (VCV(j,j) VCV(k,k))^{1/2} \quad \begin{array}{l} \text{for } j = 1, \dots, NPARE \\ \text{and } k = 1, \dots, NPARE. \end{array}$$

- PARAMETER ESTIMATES (PAR): the final value of each parameter,  $PAR(k)$ ,  $k = 1, \dots, NPAR$ .
- STD DEV OF PARAMETER ESTIMATES: the standard deviation of each estimated parameter,

$$(VCV(k,k))^{1/2} \quad \text{for } k = 1, \dots, NPAR.$$

- RATIO PAR/SD OF PAR: the ratio of each estimated parameter to its standard deviation,

$$RATIO_k = PAR(k) / (VCV(k,k))^{1/2} \quad \text{for } k = 1, \dots, NPAR.$$

- 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

$$N - \left( \sum_{i=1}^{NFAC} s(j) \cdot d(j) \right) - NPARE$$

degrees of freedom.

- The residual sum of squares at the solution.
- The residual standard deviation at the solution, RSD.
- The residual degrees of freedom,  $N - \left( \sum_{i=1}^{NFAC} s(j) \cdot d(j) \right) - NPARE$ .
- An approximation to the condition number of the derivative matrix,

$$D_{ij} = \hat{\partial} a_i / \partial \beta(j) \quad \text{for } i = 1, \dots, N \text{ and } j = 1, \dots, NPAR$$

(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, page 9.5].

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 ARIMA Forecasting Subroutines

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

The output from the ARIMA forecasting subroutines consists of a summary of the model used to produce the forecasts and, for each origin, a plot and a list of the computed forecasts and a 95-percent confidence interval about the forecasts along with the actual series value when known.

#### F. Examples

The sample programs of this section use the model and data given in table 9.1 of Box and Jenkins [1976]; the model is

$$\nabla_1^1 \cdot \nabla_{12}^1 \cdot y_i - \mu = (1 - \theta_{1,1} \cdot B^1) \cdot (1 - \theta_{1,2} \cdot B^{12}) \cdot a_i \quad \text{for } i = 1, \dots, N.$$

ARIMA Estimation. In the sample program of figure F-1a, AIME is used to compute the least squares estimates of the parameters; figures F-1b through F-1f show the output from AIME.

ARIMA Forecasting. In the sample program of figures F-2a, AIMF is used to compute the minimum mean square error forecasts using the least squares estimates obtained in example 1; figures F-2b and F-2c show the output from AIMF.

## ARIMA Estimation

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE AIME USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 6 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C N.B. DECLARATION OF Y, XM, PAR AND RES MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C INTEGER MSPEC(4,5)
C REAL Y(200), PAR(5), RES(200)
C DOUBLE PRECISION DSTAK(5000)
C
C COMMON /CSTAK/ DSTAK
C
C DEFINE VARIABLES FOR VARIOUS DIMENSIONS
C
C LOSTAK = 5000
C
C READ NUMBER OF FACTORS IN MODEL
C VALUES OF P, D, Q AND S FOR EACH FACTOR
C NUMBER OF PARAMETERS
C STARTING VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, MFAC
C READ 100, ((MSPEC(I,J)), I=1,4), J=1,MFAC)
C READ 100, MPAR
C READ 101, (PAR(I), I=1,MPAR)
C READ 100, N
C READ 101, (Y(I), I=1,N)
C
C COMPUTE LOG OF DATA
C
C DO 10 I = 1, N
C   Y(I) = ALOG(Y(I))
C 10 CONTINUE
C
C PRINT TITLE AND CALL AIME TO PERFORM ARIMA ESTIMATION ANALYSIS
C
C WRITE (6, 102)
C CALL AIME (Y, N, MSPEC, MFAC, PAR, MPAR, RES, LOSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (4I5)
C 101 FORMAT (12F6.1)
C 102 FORMAT ('RESULTS OF STARPAC ARIMA ESTIMATION SUBROUTINE AIME')
C
C END

```

DATA:

2											
0	1	1	1								
0	1	1	12								
3											
0.0	0.4	0.6									
144											
112.0	115.0	132.0	129.0	121.0	135.0	148.0	148.0	136.0	119.0	104.0	118.0
115.0	126.0	141.0	135.0	125.0	149.0	170.0	170.0	158.0	133.0	114.0	140.0
145.0	150.0	178.0	163.0	172.0	178.0	199.0	199.0	184.0	162.0	145.0	166.0
171.0	180.0	193.0	181.0	193.0	219.0	230.0	242.0	209.0	191.0	172.0	194.0
196.0	196.0	236.0	235.0	229.0	243.0	264.0	272.0	237.0	211.0	180.0	201.0
204.0	198.0	235.0	227.0	234.0	264.0	302.0	293.0	259.0	229.0	203.0	229.0
242.0	233.0	267.0	269.0	270.0	315.0	364.0	347.0	312.0	274.0	237.0	278.0
284.0	277.0	317.0	313.0	318.0	374.0	413.0	405.0	355.0	306.0	271.0	306.0
315.0	301.0	356.0	348.0	355.0	422.0	465.0	467.0	404.0	347.0	305.0	336.0
340.0	318.0	342.0	348.0	363.0	435.0	491.0	505.0	404.0	359.0	310.0	337.0
360.0	342.0	406.0	396.0	420.0	472.0	548.0	559.0	463.0	407.0	367.0	405.0
417.0	391.0	419.0	461.0	472.0	535.0	622.0	606.0	508.0	451.0	390.0	432.0

Figure F-1a

Example program and data using AIME



RESULTS OF STARPC ARIMA ESTIMATION SUBROUTINE AIME  
 \* \* \* \* \*  
 \* NONLINEAR LEAST SQUARES ESTIMATION FOR THE PARAMETERS OF AN ARIMA MODEL \*  
 \* USING BACKFORECASTS \*  
 \* \* \* \* \*

SUMMARY OF INITIAL CONDITIONS

MODEL SPECIFICATION

FACTOR	(P	D	Q)	S
1	0	1	1	1
2	0	1	1	12

INDEX	TYPE	ORDER	DESCRIPTION	FIXED	PARAMETER STARTING VALUES (PAR)	SCALE (SCALE)	STEP SIZE FOR APPROXIMATING DERIVATIVE (STP)
1	MU	---	NO	NO	.0000000E+00	DEFAULT	.25934941E-04
2	MA (FACTOR 1)	1	NO	NO	.4000000E+00	DEFAULT	.36440226E-06
3	MA (FACTOR 2)	12	NO	NO	.6000000E+00	DEFAULT	.15779351E-06

NUMBER OF OBSERVATIONS (N) 144

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) .0423E-07

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

RESIDUAL SUM OF SQUARES FOR INPUT PARAMETER VALUES (BACKFORECASTS INCLUDED) .1762

RESIDUAL STANDARD DEVIATION FOR INPUT PARAMETER VALUES (RSD) .3710E-01

BASED ON DEGREES OF FREEDOM 144 - 13 - 3 = 128

Figure F-1b

Example of AIME output

NONLINEAR LEAST SQUARES ESTIMATION FOR THE PARAMETERS OF AN ARIMA MODEL  
USING BACKFORECASTS, CONTINUED

```

ITERATION NUMBER 1
-----
MODEL CALLS          RSD          RSS          REL CHNG RSS          FORECASTED          REL CHNG PAR
                2          .3710E-01          .1761          .9132E-08          VALUE          CHKO          Y          VALUE          CHKO          Y
CURRENT PARAMETER VALUES
INDEX 1          VALUE -.1273109E-03          2          .3979023          3          .6144030
ITERATION NUMBER 5
-----
MODEL CALLS          RSD          RSS          REL CHNG RSS          FORECASTED          REL CHNG PAR
                6          .3709E-01          .1761          .2097E-10          VALUE          CHKO          Y          VALUE          CHKO          Y
CURRENT PARAMETER VALUES
INDEX 1          VALUE -.1362817E-03          2          .3954303          3          .6162601
***** RESIDUAL SUM OF SQUARES CONVERGENCE *****
    
```

Figure F-1c  
Example of AIME output (continued)

NONLINEAR LEAST SQUARES ESTIMATION FOR THE PARAMETERS OF AN ARIMA MODEL  
USING BACKFORECASTS, CONTINUED

RESULTS FROM LEAST SQUARES FIT

ROW	SERIES	PREDICTED VALUE	STO DEV OF PRED VALUE	RESIDUAL	STO
1	.47184909E+01	.47187989E+01	.15934429E-02	-.29942952E-03	-.01
2	.47706846E+01	.47650797E+01	.40331696E-02	.56049371E-02	.15
3	.48288019E+01	.48949850E+01	.24766165E-02	-.12183089E-01	-.33
4	.48598124E+01	.48488634E+01	.2772582E-02	.10948997E-01	.30
5	.47937909E+01	.48224367E+01	.61958994E-02	-.26646202E-01	-.73
6	.49052748E+01	.49269181E+01	.39922604E-02	-.21643391E-01	-.59
7	.49972123E+01	.50177683E+01	.36716059E-02	-.20595607E-01	-.56
8	.49912123E+01	.50097302E+01	.35163796E-02	-.12517946E-01	-.34
9	.49126549E+01	.49087118E+01	.45675520E-02	.39430871E-02	.11
10	.47791239E+01	.47748862E+01	.28521756E-02	.42372828E-02	.11
11	.46443909E+01	.46446312E+01	.28775758E-02	-.24030227E-03	-.01
12	.47706846E+01	.47885746E+01	.30535245E-02	-.17889946E-01	-.48
13	.47449321E+01	.47858724E+01	.42689495E-02	-.40940231E-01	-1.11
14	.4832819E+01	.48096438E+01	.49418218E-02	.26638086E-01	.72
15	.49487599E+01	.4946032E+01	.15629094E-02	.21597355E-02	.06
16	.49032748E+01	.49150647E+01	.91784768E-03	-.97899687E-02	-.26
17	.48283137E+01	.48640774E+01	.48367855E-02	-.35763861E-01	-.97
18	.50039463E+01	.49586485E+01	.35459295E-02	.45297764E-01	1.23
19	.51337984E+01	.50852287E+01	.28596969E-02	.50569772E-01	1.37
20	.51337984E+01	.51183899E+01	.40604817E-02	.17428509E-01	.47
21	.50625950E+01	.50387323E+01	.33128014E-02	.23862737E-01	.65
22	.48903491E+01	.49178407E+01	.25374040E-02	-.27491604E-01	-.74
23	.47301984E+01	.47675322E+01	.10037080E-02	-.31333735E-01	-.85
24	.49416424E+01	.48857129E+01	.22731576E-02	.5592930E-01	1.51
25	.49767337E+01	.49145074E+01	.37748967E-02	.62226322E-01	1.69
26	.50106353E+01	.50169474E+01	.55718067E-02	-.83121034E-02	-.17
27	.51817838E+01	.51806355E+01	.16061142E-02	.51148040E-01	1.38
28	.50937502E+01	.51244951E+01	.37683362E-02	-.30744898E-01	-.83
29	.51474945E+01	.50484648E+01	.48520374E-02	.99029662E-01	2.69
30	.51817838E+01	.52471996E+01	.58878659E-02	-.65416068E-01	-1.79
31	.52933048E+01	.53192418E+01	.23214718E-02	-.25936935E-01	-.70
32	.52933048E+01	.53050808E+01	.23491707E-02	-.11703191E-01	-.32
33	.52149398E+01	.52141344E+01	.31691503E-02	.80131584E-03	.02
34	.50875963E+01	.50649942E+01	.15926116E-02	.22602183E-01	.61
35	.49836066E+01	.49369820E+01	.23923596E-02	.46624627E-01	1.26
36	.5119878E+01	.5128737E+01	.42306701E-02	-.16385872E-01	-.44
37	.51416435E+01	.51287041E+01	.16478504E-02	.12959426E-01	.35
38	.51929569E+01	.51893584E+01	.21257587E-02	.35984238E-02	.10
39	.52626902E+01	.53294867E+01	.27359340E-02	-.66796941E-01	-1.81
40	.51984970E+01	.52323461E+01	.43089262E-02	-.33849107E-01	-.92
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
144	.60684250E+01	.60533938E+01	.47728765E-02	.19031807E-01	.41

Figure F-1d

Example of AIME output (continued)

NONLINEAR LEAST SQUARES ESTIMATION FOR THE PARAMETERS OF AN ARIMA MODEL  
USING BACKFORECASTS, CONTINUED

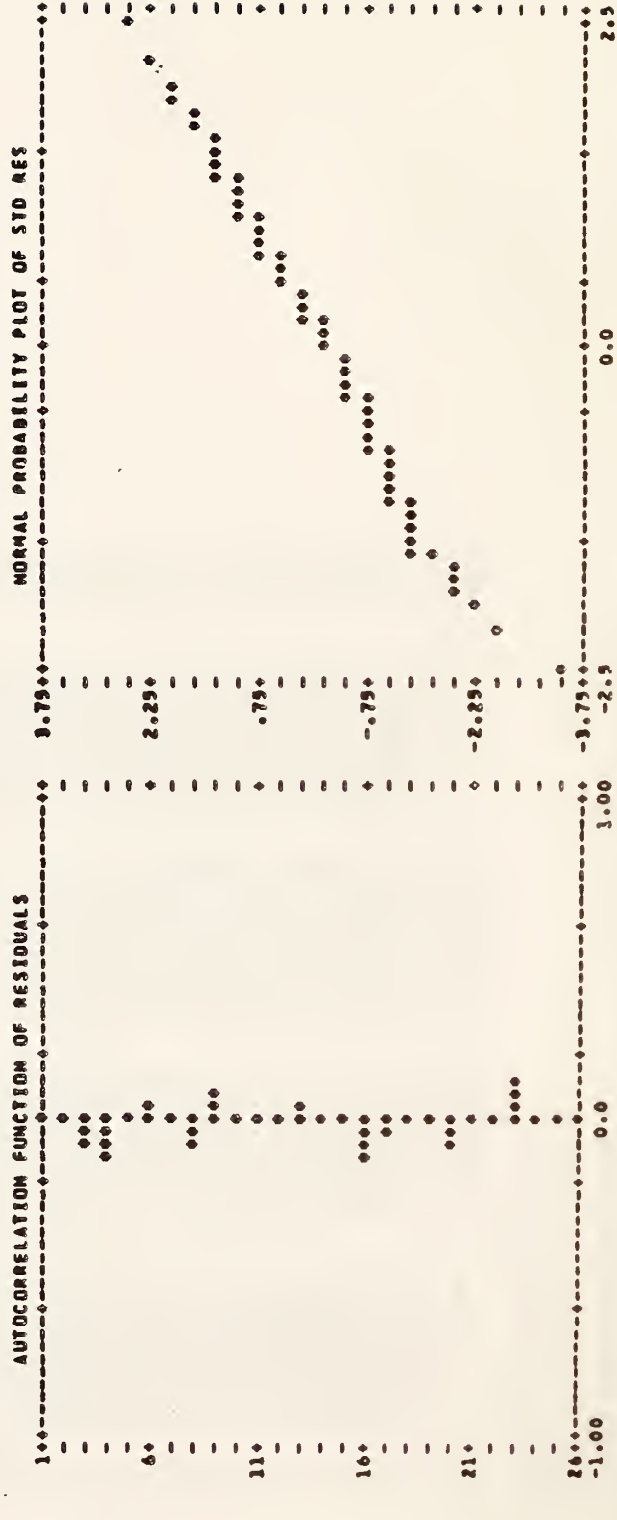
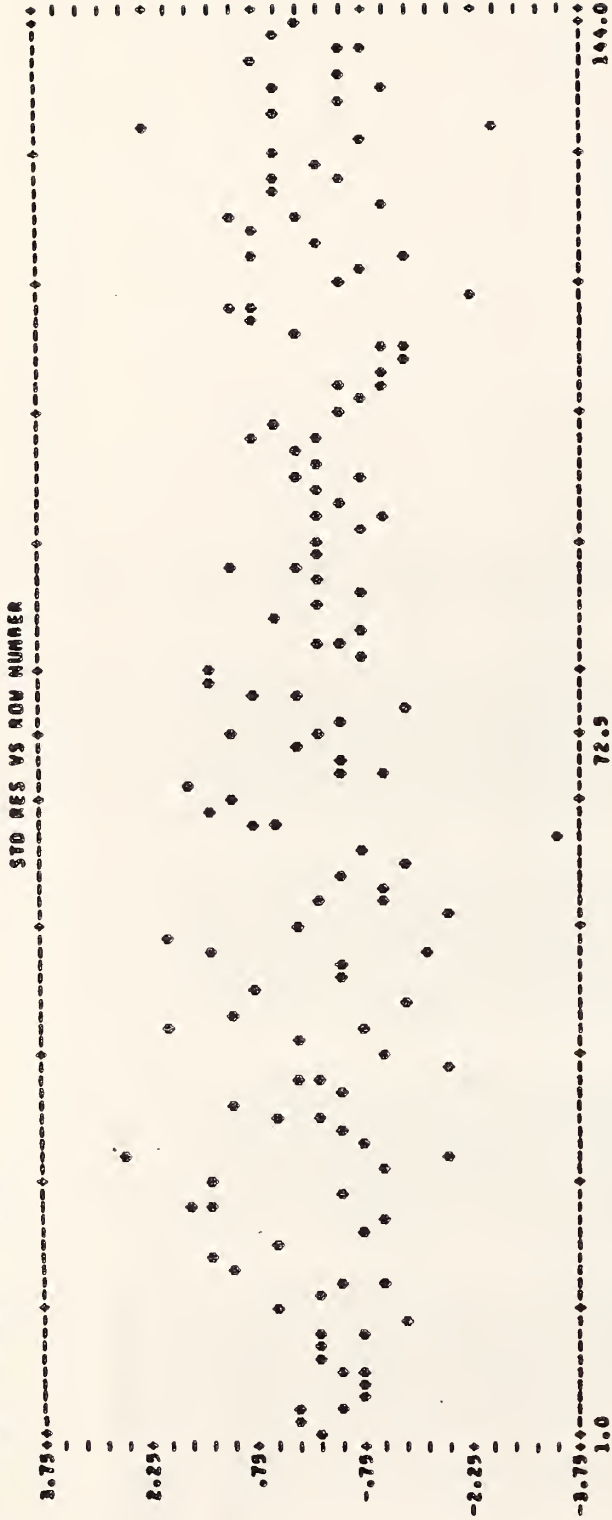


Figure F-1e

Example of AIME output (continued)



NONLINEAR LEAST SQUARES ESTIMATION FOR THE PARAMETERS OF AN ARIMA MODEL  
USING BACKFORECASTS, CONTINUED

MODEL SPECIFICATION

FACTOR	(P	D	Q)	S
1	0	1	1	1
2	0	1	1	12

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

- 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	3
1	.4322997E-06	-.1911819E-06	.1739246E-06
2	-.3790318E-02	.3680507E-02	-.1767576E-03
3	.5079170E-02	-.5657347E-01	.2712668E-02

ESTIMATES FROM LEAST SQUARES FIT

INDEX	TYPE	ORDER	DESCRIPTION	FIXED	PARAMETER ESTIMATES (PAR)	STANDARD DEVIATION OF ESTIMATES (SD OF PAR)	RATIO PAR/(SD OF PAR)	LOWER	UPPER
1	MA	(FACTOR 1)	1	NO	-.13628169E-03	.65746197E-03	-.20728465E+00	-.12248621E-02	.95229868E-03
2	MA	(FACTOR 2)	12	NO	.39543825E+00	.65746197E-03	.60446219E+03	.39434967E+00	.39652683E+00
3	MA	(FACTOR 2)	12	NO	.61628018E+00	.60667182E-01	.10196100E+02	.5158192E+00	.71671674E+00

NUMBER OF OBSERVATIONS (N) 144

RESIDUAL SUM OF SQUARES .1761318 (BACKFORECASTS INCLUDED)

RESIDUAL STANDARD DEVIATION .3709487E-01

BASED ON DEGREES OF FREEDOM 144 - 13 = 131

APPROXIMATE CONDITION NUMBER 92.27663

Figure F-1f  
Example of AIME output (continued)

## ARIMA Forecasting

```

MAIN PROGRAM:      PROGRAM EXAMPL
C
C DEMONSTRATE AIMF USING SINGLE PRECISION VERSION OF STARPAC
C RUN ON CYBER 180/840.
C
C OUTPUT UNIT IS 5 (AUTOMATICALLY EQUATED TO FILE TAPE6 ON CYBERS)
C [SEE CHAPTER 1, SECTION D.4]
C
C 4.9. DECLARATION OF Y AND PAR MUST BE CHANGED TO DOUBLE PRECISION
C IF DOUBLE PRECISION VERSION OF STARPAC IS USED.
C
C INTEGER NSPEC(4,9)
C REAL Y(200), PAR(9)
C DOUBLE PRECISION DSTAK(9000)
C
C COMMON /CSTAK/ DSTAK
C
C DEFINE VARIABLES FOR VARIOUS DIMENSIONS
C
C LDSTAK = 9000
C
C READ NUMBER OF FACTORS IN MODEL
C VALUES OF P, D, Q AND S FOR EACH FACTOR
C NUMBER OF PARAMETERS
C VALUES FOR PARAMETERS
C NUMBER OF OBSERVATIONS
C OBSERVED SERIES
C
C READ 100, NFAC
C READ 100, ((NSPEC(I,J), I=1,4), J=1,NFAC)
C READ 100, NPAR
C READ 101, (PAR(I), I=1,NPAR)
C READ 100, N
C READ 102, (Y(I), I=1,N)
C
C COMPUTE LOG OF DATA
C
C DO 10 I = 1, N
C   Y(I) = ALOG(Y(I))
C 10 CONTINUE
C
C PRINT TITLE AND CALL AIMF TO PERFORM ARIMA FORECASTING ANALYSIS
C
C WRITE (4, 103)
C CALL AIMF (Y, N, NSPEC, NFAC, PAR, NPAR, LDSTAK)
C
C STOP
C
C FORMAT STATEMENTS
C
C 100 FORMAT (4I5)
C 101 FORMAT (12F5.1)
C 102 FORMAT (12F5.3)
C 103 FORMAT ('RESULTS OF STARPAC ARIMA FORECASTING SUBROUTINE AIMF')
C
C END
DATA:
2
0 1 1 1
0 1 1 12
3
0.000 0.399 0.619
144
112.0 119.0 132.0 129.0 121.0 135.0 148.0 148.0 136.0 119.0 104.0 118.0
115.0 176.0 141.0 135.0 125.0 149.0 170.0 170.0 159.0 133.0 114.0 140.0
145.0 150.0 178.0 163.0 172.0 178.0 199.0 199.0 184.0 152.0 146.0 166.0
171.0 180.0 193.0 181.0 183.0 218.0 230.0 242.0 209.0 191.0 172.0 194.0
196.0 195.0 236.0 235.0 229.0 243.0 264.0 272.0 237.0 211.0 180.0 201.0
204.0 188.0 238.0 227.0 234.0 264.0 302.0 293.0 259.0 229.0 203.0 229.0
242.0 233.0 267.0 269.0 270.0 315.0 364.0 347.0 312.0 274.0 237.0 278.0
284.0 277.0 317.0 313.0 318.0 374.0 413.0 405.0 355.0 306.0 271.0 306.0
315.0 301.0 356.0 348.0 355.0 422.0 465.0 467.0 404.0 347.0 305.0 336.0
340.0 318.0 362.0 348.0 363.0 435.0 491.0 505.0 404.0 359.0 310.0 337.0
340.0 342.0 406.0 396.0 420.0 472.0 548.0 559.0 463.0 407.0 362.0 405.0
417.0 391.0 419.0 461.0 472.0 535.0 622.0 606.0 508.0 461.0 390.0 432.0

```

Figure F-2a

Example program and data using AIMF

RESULTS OF STARPAC ARIMA FORECASTING SUBROUTINE AIME  
 \*\*\*\*\*  
 \* ARIMA FORECASTING \*  
 \*\*\*\*\*

MODEL SUMMARY  
 -----

MODEL SPECIFICATION

FACTOR	(P	D	Q)	S
1	0	1	1	1
2	0	1	1	12

INDEX	PARAMETER DESCRIPTION	ESTIMATES
---	TYPE --ORDER	----- (PAR)
1	MU	.00000000E+00
2	MA (FACTOR 1)	.39500000E+00
3	MA (FACTOR 2)	.61500000E+00

NUMBER OF OBSERVATIONS

(N) 144

RESIDUAL SUM OF SQUARES .1761643 (BACKFORECASTS INCLUDED)

RESIDUAL STANDARD DEVIATION 144 - 13 - .3709890E-01  
 BASED ON DEGREES OF FREEDOM 3 - 128

Figure F-2b

Example of AIMF output





## G. Acknowledgments

The subroutines used to compute the least squares solution are those referenced in Dennis et al. [1981]. The algorithm used to select optimum step sizes for numerical derivatives was developed by Schnabel [1982]. The printed output for the ARIMA estimation subroutines has been modeled in part on the linear least squares output used by OMNITAB II [Hogben et al. 1971].



## Appendix A

### CONTINUITY OF VERTICAL PLOTS ON THE CDC CYBER 840 AND 855

Normally, a line printer will automatically provide margins at the top and bottom of each page causing a break in the continuity of a vertical plot extending over two or more pages. However, these automatic page-ejects within a vertical plot can be suppressed by the user on many systems. On the CDC Cyber 840 and 855 model machines this would be done by printing a Q carriage control in column one immediately before the call to the vertical plot routine. Printing an R carriage control will cancel this effect. For example, the sequence

```
        WRITE (6, 100)
100    FORMAT (1H1, 23HTITLE FOR VERTICAL PLOT)
        WRITE (6, 101)
101    FORMAT (1HQ)
        CALL VP (Y, N)
        WRITE (6, 102)
102    FORMAT (1HR)
```

will produce a vertical plot beginning on a new page, without any breaks in continuity and without affecting the automatic page-ejects in the rest of the output. Users of other systems should consult their Computer Center staff for any equivalent method available.





## 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,  $\epsilon_i$ , are not equal, the unweighted least squares estimates remain unbiased but do not have minimum variance. Minimum variance estimates are obtained by using weights  $w_{t_i} = 1/\text{Variance}[\epsilon_i]$  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,  $\eta$ , in the results of a real valued function,  $g(\beta)$ , can be estimated in most cases by evaluating

$$\eta = -\log_{10} \left( \max_{j=-2, \dots, 2} \frac{\{|g(\beta^j) - [a + j \cdot b]|\}}{|g(\beta)|} \right)$$

where

$\beta^j$  is the vector of the NPAR parameters of the function given by,

$$\beta^j(k) = \beta(k) + \beta(k) \cdot j \cdot 10^{-(\text{DIGITS}/2)} \quad \begin{array}{l} \text{for } j = -2, \dots, 2, \\ \text{and } k = 1, \dots, \text{NPAR}, \end{array}$$

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.

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

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

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





## 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.
- 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. (1985). Computational experience with confidence regions and confidence intervals for nonlinear least squares. University of Colorado Department of Computer Science Technical Report CU-CS-302-84.
- 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 NBS 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: 452-483. 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.; Welsch, 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 NBS 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 NBS 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.



U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> (See instructions)		1. PUBLICATION OR REPORT NO. NBSIR 86-3448	2. Performing Organ. Report No.	3. Publication Date January 1987
4. TITLE AND SUBTITLE  STARPAC - The Standards Time Series and Regression Package				
5. AUTHOR(S) Janet R. Donaldson and Peter V. Tryon				
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)  NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)				
10. SUPPLEMENTARY NOTES  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)  STARPAC, the <u>Standards Time Series</u> and <u>Regression Package</u> , is a library of Fortran subroutines for statistical data analysis developed by the Statistical Engineering Division (SED) of 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].  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.				
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) data analysis; nonlinear least squares; STARPAC; statistical computing; statistical subroutine library; statistics; STATLIB; time series analysis.				
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES  300	15. Price  \$24.95





